AN ALGORITHM FOR LOCATING FUNDAMENTAL FREQUENCY (F0) MARKERS IN SPEECH

by

Princy Dikshit
B.E (C.S) July 2000, Mangalore University, India

A Thesis Submitted to the Faculty of
Old Dominion University in Partial Fulfillment of the
Requirements for the Degree of

MASTER OF SCIENCE

COMPUTER ENGINEERING

OLD DOMINION UNIVERSITY
DECEMBER 2004

Approved by:

_____
Stephen. A. Zahorian (Director)

_____
Vijayan K. Asari (Member)

_____
Min Song (Member)

ABSTRACT

AN ALGORITHM FOR LOCATING FUNDAMENTAL FREQUENCY (F0) MARKERS IN SPEECH

Princy Dikshit
Old Dominion University, December 2004
Director: Dr. Stephen A. Zahorian

Speech has been the principal form of human communication since it began to evolve at least one hundred thousand years ago. Speech is produced by vibrations of the vocal cords. The rate of vibration of the cords is called fundamental frequency (F0) or pitch. The objective of this thesis is to locate pitch period cycles on a cycle-by-cycle basis. The complexity in identifying pitch cycles stems from the highly irregular nature of human speech. Dynamic programming is used to combine two sources of information for pitch period marking. One source of information is the "local" information corresponding to the location and amplitude of peaks in the acoustic speech signal. The other source of information is the "transition" information corresponding to the relative closeness of the distance between the signal peaks to the expected pitch period values. The expected pitch period values are obtained from a pitch tracker (YAPT) or from the reference pitch track. The Keele speech database was used for testing purposes.

Over 95% of the identified pitch cycles were within a 1ms deviation of the actual pitch cycles in experiment using clean speech signals. In experiments with noisy speech signals, an accuracy rate of 92% and above was observed for an SNR range of 30db to 5db. In an experiment evaluating the robustness of the algorithm vis-à-vis errors in the pitch track using clean studio quality signals, an accuracy rate of 95% was obtained for an error range of -10% to +60% in pitch. The algorithm generated = 1% extra markers (false positives) for clean studio quality (pitch track error range of -10% to +60%) and noisy speech signals (SNR range of 30db to 5db). The use of the pitch track generated by the ODU pitch tracker (YAPT) for identifying pitch markers gave an accuracy rate of 95% as compared to 93% obtained using the reference pitch track supplied with the Keele database. A preliminary test on telephone quality signals gave an accuracy rate of 63%.

To my mom, dad and sister!

I thank you for your unconditional love, encouragement and support.

ACKNOWLEDGMENTS

I would like to thank Dr. Stephen A. Zahorian for his constant support, guidance and incredibly high patience without which it would not have been possible to complete this thesis.

I would also like to thank Dr. Vijayan K. Asari and Dr. Min Song for consenting to be on the thesis advisory committee.

I would like to thank my Speech Lab colleagues for their help.

This work was partially supported by NSF grant BES-9977260.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER I

INTRODUCTION

1.1 Introduction

Speech has been the principal form of human communication since it began to evolve at least one hundred thousand years ago. Speech is a complex wonder and one of the unique characteristics of the human species. Prior to the twentieth century, there was very little technical understanding of the processes involved. It is only in the last century with advances in acoustical and physiological measurement techniques that researchers have developed a considerable understanding of speech. Recent breakthroughs in desktop computing power and advances in signal processing methods have further accelerated the learning curve for understanding speech. However, many mysteries remain in understanding speech production and perception.

One of the major advances has been in understanding speech production. It has been found that sound in general is produced by vibrations of the vocal cords. The vocal cord vibrations themselves do not contain much information for most languages. However, the sound from the vocal cords is spectrally shaped by the vocal tract according to the content in the speech. This process of speech production can be modeled as a time-varying linear system that is excited by either a quasi-periodic source (for "voiced" sounds, such as all vowels), or a more noise-like source (for "unvoiced" sounds, such as many consonants) [1], [2], [3] and [4].

Speech analysis is the branch of speech science that estimates the parameters of the model for speech production from acoustic measurements of a speech signal. One such very important parameter or feature of the sound is the fundamental frequency for

---

This thesis uses the journal model of IEEE Transactions on speech and audio processing for tables, figures and references

the voiced portions of speech. Fundamental frequency (F0) corresponds to the rate at which the vocal cords vibrate. Determination of F0 from the acoustic signal is difficult for at least three reasons: 1, F0 of the speech signal varies with time, since the vocal cord vibrations are quasi-periodic; 2, The acoustic signal is not always voiced, so any algorithm for automatically determining F0 must first make a voiced/unvoiced decision; 3, The spectral shaping caused by the vocal track sometime makes it difficult to identify the fundamental frequency. The main perceptual attribute of F0 is called pitch, which actually is an overall perceived spectral quality. However, the terms pitch and F0 are often used interchangeably and will be thus used in this thesis.

The main objective of this thesis is to develop an algorithm to locate pitch cycles on a cycle-by-cycle basis. Although a number of algorithms for this purpose are described in the literature, none of the algorithms have proven to be highly accurate. The complexity in identifying pitch cycles stems from the variable and highly irregular nature of human speech. Potential applications of the markings of precise pitch period locations include analysis of jitter, prosody in speech [5], text-to-speech synthesis [6,7], analysis of voice quality and pitch synchronous analysis [8] which finds use in fields as diverse as speaker normalization to spectral analysis. Pitch synchronous analysis has been shown to increase performance of speech recognition, especially under conditions of moderate noise. The present work is intended to provide the detailed cycle-by-cycle identification of pitch periods that would be needed for pitch synchronous analysis and the other potential applications listed.

In the following sections we discuss speech production in detail and also introduce the idea of pitch synchronous analysis, before concluding the chapter with a brief overview of the following chapters.

1.2 Acoustic theory of speech production: Source-filter model

Speech in humans is considered to result from a source of sound energy (e.g. the larynx) modulated by a transfer (filter) function determined by the shape of the

supralaryngeal vocal tract, as shown in Fig. 2. This model is called as the "source-filter theory of speech production" and traces its origins to experiments of Johannes Muller (1848) in which a functional theory of phonation was tested by blowing air through larynges excised from human cadavers. In this model the source of acoustic energy is assumed to be at the larynx with the supralaryngeal vocal tract serving as a variable acoustic filter whose shape determines the phonetic quality of the sound (Fant, 1960).

With the larynx serves as a source of sound energy, voiced sounds are produced by a repeating sequence of events. First, the vocal cords are brought together, temporarily blocking the flow of air from the lungs and leading to increased subglottal pressure. When the subglottal pressure becomes greater than the resistance offered by the vocal folds, they open again. The folds then close rapidly due to a combination of factors, including their elasticity, laryngeal muscle tension, and the Bernoulli effect [2]. If there is a steady supply of pressurized air, the vocal cords will continue to open and close in a quasi-periodic fashion. As they open and close, puffs of air flow through the glottal opening. The frequency of these pulses determines the fundamental frequency (F0) of the laryngeal source and contributes to the perceived pitch of the produced sound. An example of the spectrum of the result of such glottal airflow is plotted at the top left of Fig. 2.

The supralaryngeal vocal tract, consisting of both the oral and nasal airways, as depicted in Fig. 1, serves as a time-varying acoustic filter that suppresses the passage of sound energy at certain frequencies while allowing its passage at other frequencies. The detailed shape of the filter (transfer) function is determined by the entire vocal tract serving as an acoustically resonant system combined with losses including those due to radiation at the lips. A hypothetical filter function for the neutral vowel /˝/ is shown in the center panels of Fig. 2. The formant frequencies, corresponding to the peaks in the function, represent the center points of the main bands of energy that are passed for a specific vocal tract shape. The spectrum of the glottal airflow, which has energy at the fundamental frequency (100 Hz) and at the harmonics (200 Hz, 300 Hz, etc.), is plotted at the top left of Fig. 2.

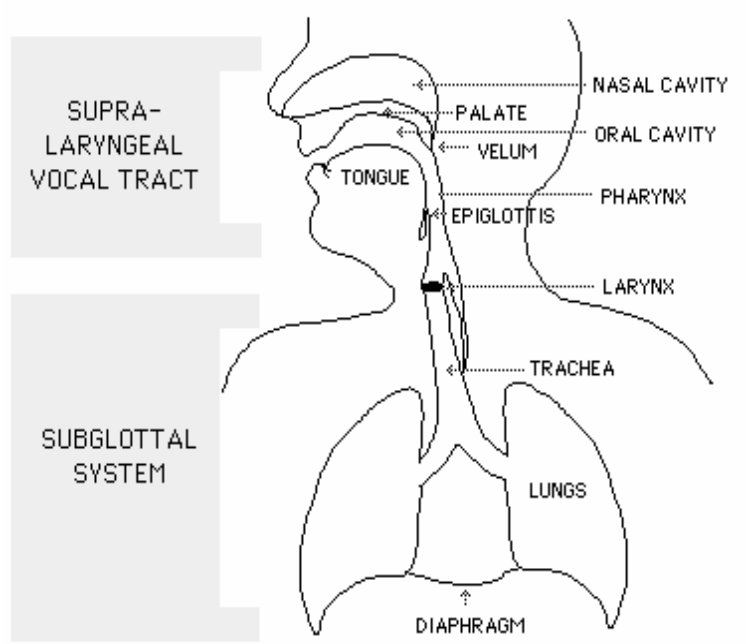


Fig. 1. The human speech production system [9].

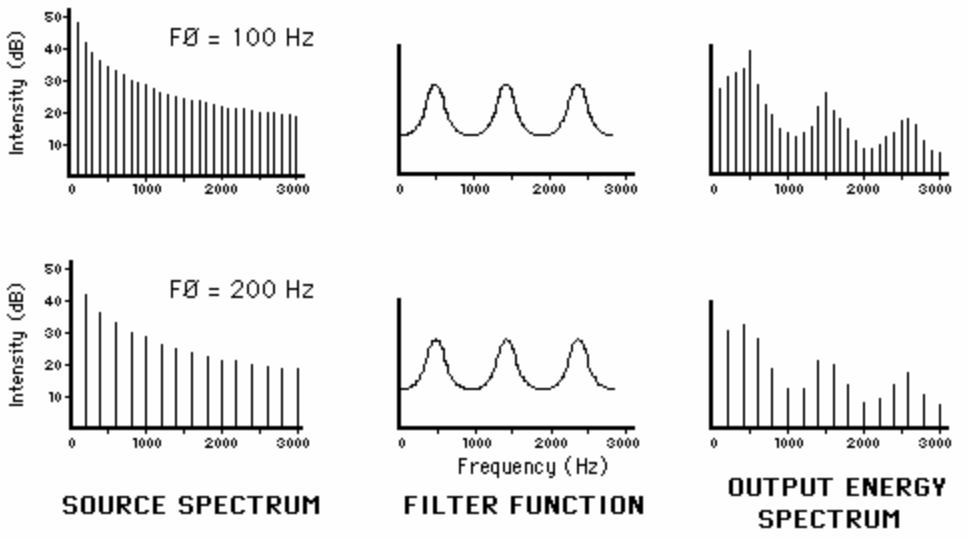


Fig. 2. The source-filter model of speech production [9].

The spectrum at the top right corner of the Fig. 2 shows the spectrum resulting from filtering of the source spectrum with the filter function shown in the center of the figure. Note that the laryngeal source is reshaped by the filter function. Energy is present at all harmonics of the fundamental frequency of the glottal source, but the amplitudes of individual harmonics are determined by both the source amplitudes and the filter

function. The bottom panel of Fig. 2 shows the effect of using a different source function, while retaining the same filter function. In this case, the fundamental frequency of the glottal source is 200 Hz, with harmonics at integer multiples of the fundamental (400 Hz, 600 Hz, etc.). Although not shown in the Fig. 2, if the source was unvoiced, the source spectrum would be continuous, and the output spectrum would also be continuous with the same envelope as for the voiced speech[1].

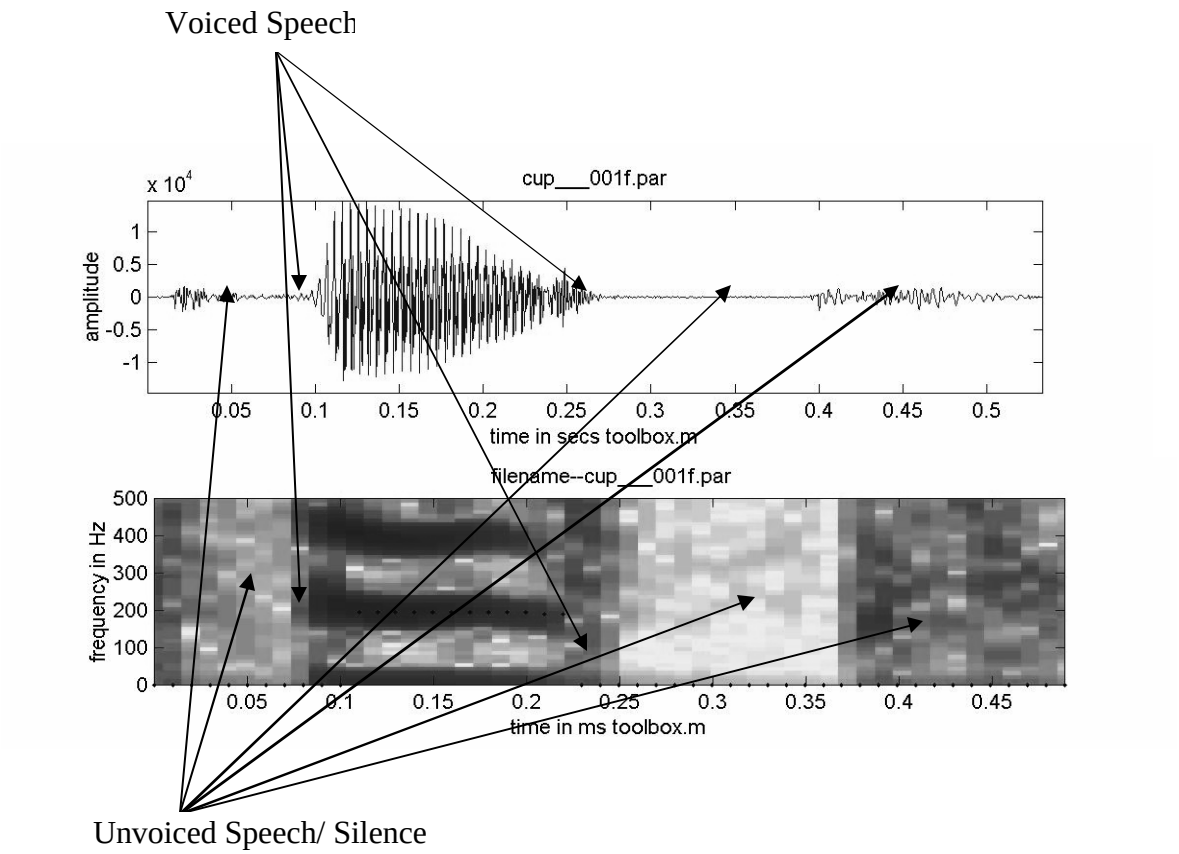


Fig. 3. Illustration of an example of voiced and unvoiced speech. The top panel illustrates the acoustic speech signal and the bottom panel illustrates the spectrogram of the speech signal [10].

Based on the activity of the vocal cords speech can be classified as voiced or unvoiced.

Speech/sound produced when vocal cords modulate airflow from the lungs with rapid openings and closings is said to be voiced (all vowel sounds and parts of many

[1] Note that most of material in section 1.2 up was adapted from a web reference of [9], with only minor changes. This background primer on speech production was only provided here for the convenience of the reader.

consonants). The sound produced when the vocal cords allow a non-periodic turbulent flow, is unvoiced (many consonants such as “s,” “sh,” “z,” etc.). Therefore a sound is voiced when it has periodic components (F0) and unvoiced otherwise. Figure 3 illustrates an example of voiced and unvoiced speech. The top panel in the figure is the acoustic waveform, and the bottom panel is a speech spectrogram. The speech spectrogram, a common tool used in speech analysis, is a time/ frequency/intensity display of speech. Note that the voiced speech interval appears quasi-periodic in the time domain acoustic signal, and as intense (dark) bars in the spectrogram. The unvoiced speech is much more irregular in both time and frequency.

1.3 Pitch tracking and pitch marking

Pitch tracking consists of classifying speech into voiced and unvoiced regions, and for voiced regions determining the fundamental frequency of the vibrations of the vocal chords. Generally pitch tracking involves determining pitch (F0) over a large interval of speech as shown in Fig. 4. The pitch track is represented by the block dots overlaid on the spectrogram. Notice that each dot represents a value of F0 The dots in the voiced region (0.05-0.3 seconds in the figure below) have a non-zero value whereas dots in unvoiced region represent a value of zero frequency implying an absence of periodic component (F0).

Pitch tracking algorithms generally do not identify the temporal location of each vibration of the vocal chords but rather are based on average time spacing between a numbers of vibrations. The averaging, typically due to the use of an autocorrelation type of calculation in the pitch tracking, is used to improve the accuracy of the tracking.

Pitch marking (PM), on the other hand, attempts to locate every vibration of the vocal chords. That is, the beginning and end of each pitch cycle is to be located by timing markers. PM does not involve classifying speech into voiced or unvoiced regions but rather may use such pre-existing knowledge for locating pitch cycle markers. That is, typically, and specifically in the work reported in this thesis, pitch tracking is first

performed and used to help with the pitch marking. Figure 5 shows the markers identified in pitch marking process for an actual speech signal.

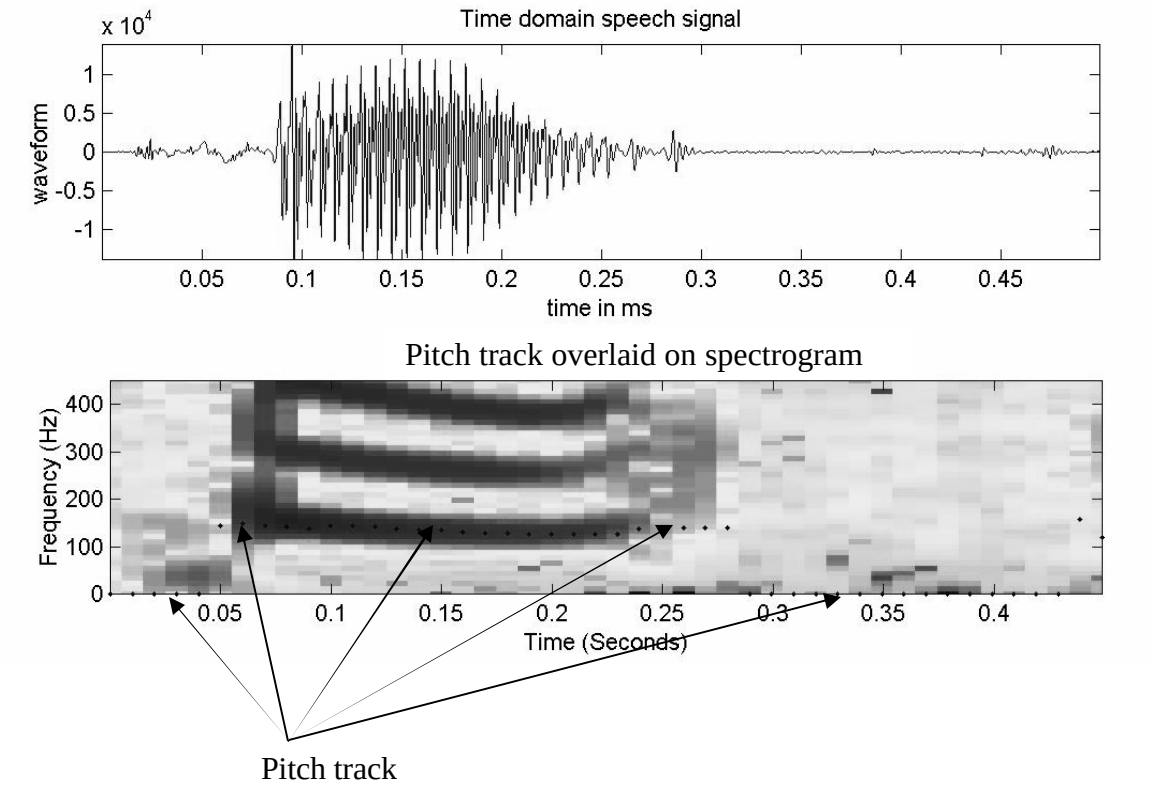


Fig. 4. Illustration of an example of pitch track of a speech signal. The top panel shows the time domain speech signal and bottom panel shows the pitch track overlaid on a spectrogram of the speech signal [10].

Pitch tracking and pitch marking turn out to be extremely difficult problems to solve accurately [11]. The fundamental reason is that speech signal is not really periodic, and it is also highly non-stationary. That is, even over short time intervals on the order of 50 ms the speech signal is often changing in F0, in amplitude and in overall spectral characteristics. Even in normal speech, for some cases, the first harmonic can be much larger than the fundamental, which further pitch tracking and marking difficult. Even the fundamental voiced/unvoiced decision is difficult.

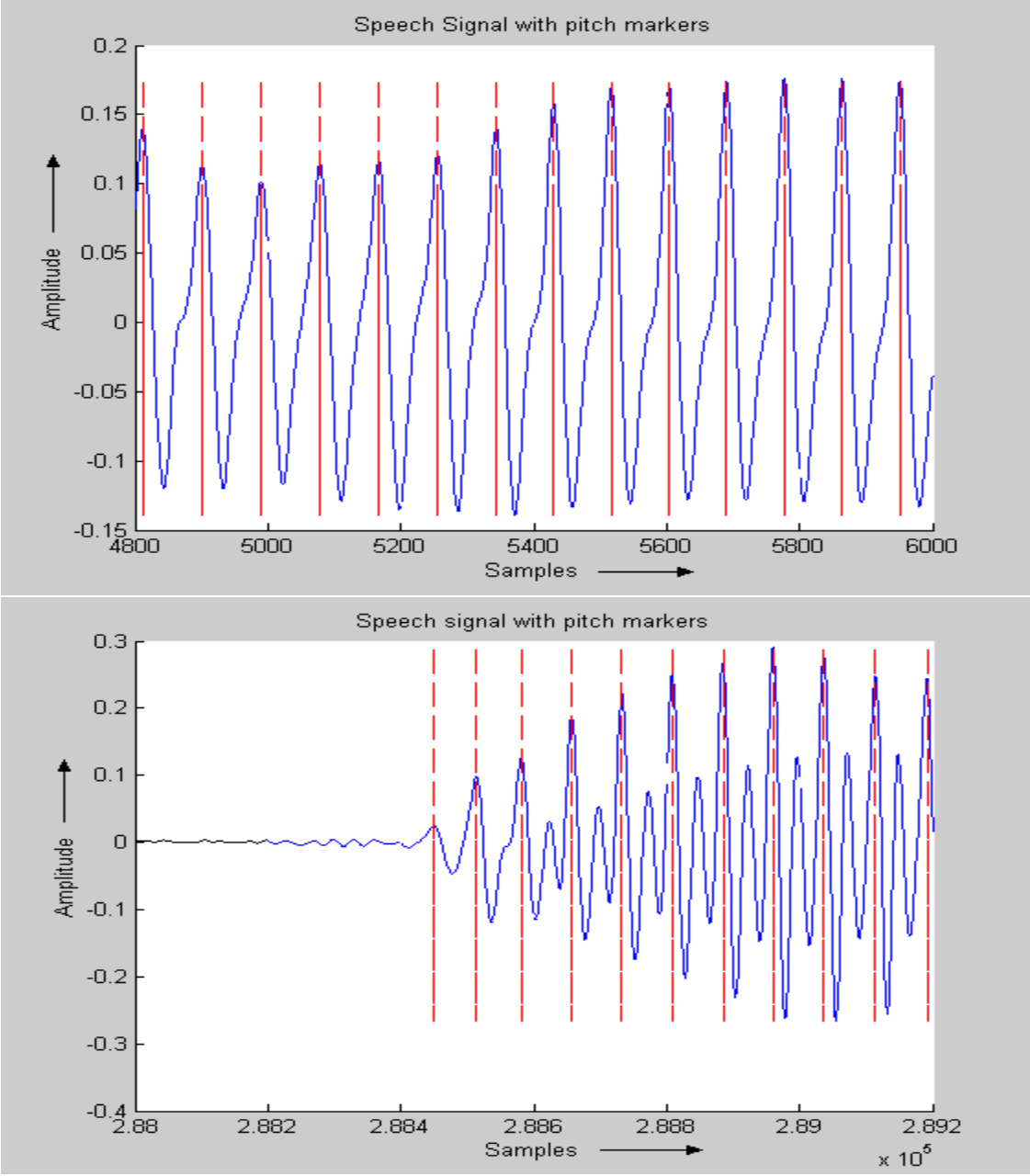


Fig. 5. Illustration of pitch markers identified in speech signal for different regions of speech. The top panel shows markers in speech signal that is purely voiced whereas the bottom panel shows markers in a section of speech signal that has unvoiced and voiced region. Signal in blue represents voiced region and signal in black represents unvoiced signal.

1.4 Pitch synchronous analysis

The speech processing methods or processes that use cycle-by-cycle information about the location of the pitch periods at any point during processing are said to be

performing pitch synchronous analysis. The knowledge of pitch cycles can be used so that the frames needed for short time spectral analysis can contain an integer number of pitch periods. Consequently, smoothing windows (such as Hamming or Hanning) are not needed and better spectral resolution is possible. Spectral analysis and text-to-speech synthesis methods are a few that benefit extensively from such frame based analysis. Other potential applications of the markings of precise pitch period locations include analysis of jitter, prosody in speech [5], and analysis of voice quality and in fields as diverse as speaker normalization to speech recognition.

In text-to-speech synthesis, pitch synchronous analysis [8] comes into play, for example, when the rate of the speech or the length of the speech signal needs to be altered. This can be done by adding or removing individual pitch periods from the speech signal. In order to perform this the pitch periods in the signal need to be identified individually. The speech signal rate or length can be altered by introducing extra copies of pitch cycles or removing certain pitch cycles. Introducing pitch cycles causes an increase in the length of the signal or decrease in the speed of the signal and removing pitch cycles causes a decrease in the length or increase in the speech rate. This type of processing is normally known as time-domain pitch synchronous analysis (TD-PSOLA). There is another form of pitch synchronous analysis called frequency-domain pitch synchronous analysis (FD-PSOLA), which, as the name suggests, works in the frequency domain, but still requires precise pitch cycle identification.

1.5 Objectives of thesis

The basic goal of this thesis is to develop and present an accurate algorithm for pitch marking. The motivation for this work is based on the material presented above in this chapter, such as improved speech recognition under noisy conditions. Another potential application of the pitch marking is the precise identification of the onset of voicing for sounds such as stop consonants (/b/. /p/, /d/, /d/, /t/, /g/, /k/), which is useful in the development of a speech training aid for the hearing impaired.

The specific objectives of the research work are to:

1. Develop an algorithm that has a high accuracy rate in locating the individual speech cycles.
2. Increase the performance of the algorithm using experimental testing, with respect to minimizing errors such as false positives (generation of extra pitch cycles) and false negatives (omission of actual pitch cycles), generally associated with pitch marking algorithms.
3. To document the performance of the pitch marking algorithm mentioned in steps 1 and 2 with respect to several error measures

A summary of the following chapters in this thesis is as follows. Chapter 2 contains a summary of relevant background literature. Chapter 3 gives a detailed description of the pitch-tracking algorithm. Chapter 4 details experiments and results. Chapter 5 gives a summary and suggestions for future work.

CHAPTER II

BACKGROUND--LITERATURE REVIEW

2.1 Introduction

In this chapter, a summary of literature in the field of pitch marking is given.

2.2 Survey of related work

Goncharoff and Gries (1998) [12] present both an algorithm for pitch period estimation and another algorithm for pitch phase estimation. There is extensive use of dynamic programming in both the algorithms. In both cases, unvoiced intervals are treated the same as voiced intervals. Thus, the entire signal is marked with pitch period intervals.

For the case of pitch period estimation, a short-time energy contour is obtained by convolving the squared speech samples with a smoothing window derived from a Hanning window. The computed energy contour has regularly spaced large amplitude peaks in the voiced regions and low-amplitude peaks at irregular intervals in the unvoiced region. Dynamic programming is then applied to a set of candidate pitch estimates, which are defined as the spacings between adjacent energy peaks. Dynamic programming is used to find those candidates such that the sum of the energy peaks along the path is maximum. The pitch period estimates are then interpolated over the entire speech signal.

The pitch period obtained in the first step is then used to mark overlapping frames in the speech signal, with each frame centered about the location of an expected pitch marker and with each frame width equal to twice the maximum expected pitch period. Dynamic programming, with path energy maximization and slope constraints, is then used to determine the most likely pitch pulse location for each frame. The marking

algorithm is claimed to perform well, and to be suitable for techniques such as PSOLA, but no experimental performance analysis is given. As described, the method would seem to work well, provided the pitch tracking is accurate and pitch periods do not deviate too much from cycle to cycle.

In the work by Harbeck et al [5] the task of detecting pitch periods is solved with a search for an optimal path through a space of pitch period hypotheses using dynamic programming (DP). The DP cost function is computed with automatically trained artificial neural networks (ANNs) which combine the outputs of heuristic functions measuring the similarity of adjacent period hypotheses.

The speech signal is normalized to zero mean value, and positive zero crossings are determined in the sections of speech marked as voiced using external frame-based voiced-unvoiced decisions. The zero crossings are identified based on heuristic criteria. After identifying the zero crossings, a search space of period hypothesis for the voiced region is constructed. All the period hypotheses that have their starting points within a distance of I_{max} from the first zero crossing at the beginning of the voiced region are called an *Initial Hypothesis*. Each period hypothesis length lies within the interval [I_{max}, I_{min}], where

$$I_{min} = \frac{1}{G * \frac{100+p}{100}} \text{ seconds} \qquad \text{and} \qquad I_{max} = \frac{1}{G * \frac{100-p}{100}} \text{ seconds,}$$

where G is the pitch level estimate based on a F0-algorithm and p is the permitted deviation of the length of two adjacent pitch periods.

After defining the *Initial Hypothesis* set, all possible paths (consisting of a series of contiguous pitch period hypothesis) are identified that end with a period hypothesis ($P_{i,j}$, zero crossings at i and j, i<j) and then dynamic programming is applied to determine the minimum cost path up to the period hypothesis $P_{i,j}$. A path always begins with at least one period hypothesis from the *Initial Hypothesis* set. The paths obtained, ending with the period hypothesis $P_{i,j}$, are such that the zero crossing at i is the last zero crossing for

all previous period hypotheses ($P_{[i-Imax < k < i-Imin],\, i}$, k<i). This process is continued up to the last period hypothesis of the voiced segment, and the path, with the least cost, ending not more than I_{max} away from the end of the voiced segment, is chosen as the set of final pitch markers.

The cost function used in DP is a function value obtained after combining 14 different primitive cost functions with the help of a neural network. The 14 primitive cost functions are split into two groups of cost functions, such that one group scores the period hypothesis itself (4) and the other group scores the similarity of a period hypothesis with its adjacent period hypothesis (10).

The accuracy of the final pitch markers was determined indirectly by comparing the pitch level G_{est} obtained from the pitch markers with the reference pitch level available for the speech signal. The algorithm was found to have a coarse error rate of 4.75% on a German speech database using a permitted deviation (p) of 45, with a coarse error occurring when the pitch level of a frame differs from the reference pitch level by more than 30 Hz. The algorithm as defined is not computationally efficient because of the use of the ANN and the primitive form of DP method used for determining the least cost path. Although the number of computations and time consumed could be reduced by using smaller values of permitted deviation factor p, it was shown to cause an increase in the coarse error rate thus reducing the accuracy of the pitch markers.

The work by Laprie and Colotte [7] tries to exploit the results of a pitch extraction algorithm for estimating the pitch marks. The method tries to optimize the propagation of the pitch marks using the known pitch values with the help of Dynamic programming (DP). To begin with, all the extrema are extracted in regularly spaced voiced segments (the length of each voiced segment is no more than the smallest pitch period observed from the pitch values). Then an optimal set of pitch marks (extremas) are found using DP.

The first step involves extracting two sets of candidate pitch marks maximas and minimas. The set of candidate pitch marks that have the lowest cost based on DP are chosen as the final set of pitch markers. A total of five peaks are extracted in each segment (size equal to the smallest pitch period), so that each legitimate pitch period has at least some representation in the form of the 5 extremas, even if this results in an actual (large) pitch period represented by more than 5 candidates (segment length smaller than the actual pitch period). The number of candidate peaks extracted in each segment should be large enough that the actual pitch mark is always included), which was proposed to be equal to 5. After identifying the candidate peaks, a process similar to the method mentioned earlier [5] is used to find the final set of pitch marks with a minimum cost path. Here though, instead of using the pitch hypothesis as in [5], candidate peaks (locations) are used. The cost function used for DP in this algorithm is designed to take into account the reliability of two pitch marks with respect to the corresponding pitch period value (obtained from a pitch extraction algorithm) and also the amplitude of the pitch mark. And as in [5], only those paths that end within a distance 'p' of a candidate pitch mark are explored as probable pitch mark sets. Laprie and Colotte also suggest use of correlation function [6, 7] as a part of the DP cost function when the speech signal is corrupted by noise or the estimated pitch track values are inaccurate.

The algorithm is claimed to give good results with all kinds of speech signals and is also claimed to perform equally well on male and female speakers, although no experimental results are given.

The work by Veldhuis [13] also proposes the use of DP to determine the pitch marks. It suggests the use of three different consistency requirements for selecting candidate pitch markers and also as part of the DP cost function. The three proposed consistency requirements are the characteristic-property requirement, the waveform-consistency requirement and the pitch-consistency requirement. The characteristic-property requirement demands that candidate pitch markers be positioned at either higher maxima of the absolute value of the signal or at the first zero crossing before the maximum positive peak or based on some other signal property. On the other hand, the

waveform-consistency requirement tests the similarity between signal portions around adjacent pitch markers, whereas the pitch-consistency requirement selects pitch markers whose distance is close to the estimated pitch period.

The first step of the algorithm requires identifying candidate pitch markers that satisfy the characteristic-property requirement and then computing local costs to quantify this requirement. A set of additional markers is identified in the vicinity (a window range of ±50% of estimated pitch period) of each candidate. The second step involves calculating transition costs quantifying waveform-consistency requirement (normalized cross-correlation function) for the candidate marker and the additional candidate markers identified in its vicinity. If the waveform-consistency requirement value is found to be less than a predefined threshold, then pitch-consistency requirement is used to determine the transition costs. Then DP is used to recursively determine the path for which the sum of the local and transition costs is minimum and thus obtaining final pitch markers.

Experiments conducted to determine the effects of the three requirements showed that the accuracy of the pitch markers is not strongly influenced by the characteristic property. However, it was experimentally determined that pitch markers obtained without using waveform and pitch consistency requirements had less accuracy. Also the algorithm was found to be sensitive to large errors in estimated pitch periods.

2.3 Summary

An overview of selected research works in the field of pitch marking was given. All the works reviewed used DP in conjunction with pitch track information and local amplitude /signal values to identify pitch cycles. A few of the algorithms [6, 7 and 13] even use waveform shape information in the form of correlation or normalized cross correlation function values. Though all the algorithms work well with accurate pitch track values, they were not found to be robust to large errors in the pitch track. The algorithms were also computationally inefficient because DP was used in a very basic fashion.

In the next chapter we describe the algorithm developed to determine the pitch markers on a cycle-by-cycle basis. The algorithm gives a new approach to use of pitch tracking results with DP that is more robust to errors in pitch track and is also more computationally efficient.

CHAPTER III

THE ALGORITHM

3.1 Introduction

In chapter II, a survey of existing algorithms for pitch detection on a cycle-by-cycle basis was given. This chapter presents the complete algorithm developed in this thesis for identifying /marking pitch cycles in the time-domain. The starting assumption is that a pitch track is already available, as well as the acoustic signal. The challenge is to then identify specific pitch cycles in the acoustic waveform, combining information from the waveform (i.e., peaks in the signal) along with the expected period, obtained from the pitch track. The algorithm for combining these two sources of information is centered around dynamic programming to find a lowest cost solution from a set of possible solutions. Note that the pitch tracking itself involves identifying correct pitch period estimates from many possible pitch period values, and thus dynamic programming (DP) is also often used to guide the process, e.g., RAPT [14] and YAPT [15]. It will be shown later in this chapter, how to adapt DP to pitch period detection in the speech signal, a variation/refinement of the pitch tracking problem.

3.2 Dynamic programming

This section gives an explanation of using DP to obtain a solution, applicable to pitch marker detection, [16] but explained in the context of a modified stagecoach problem. A basic working knowledge of DP is assumed as background for this chapter. Nevertheless, an explanation of the basic principles of DP is given below.

A definition of the modified stagecoach problem is as follows:
“A businessman intends to travel by stagecoach from city A in west coast to city J in east coast in the middle of the 19^{th} century in America. The businessman can choose to travel along any of the possible routes, which take him from city A to city J, through cities (B to

I) and five time zones. Stagecoaches run from one city to another city with changes of stagecoach possible only in those cities. The businessman intends to take rest in each city for a day before he changes the stagecoach and begins travel to another city. Rest houses in cities charge him for using their services and stagecoaches cost him 10 dollars for each mile traveled between two cities. He intends to select the route that costs the least."

This is represented in the form of a diagram in Fig. 6.

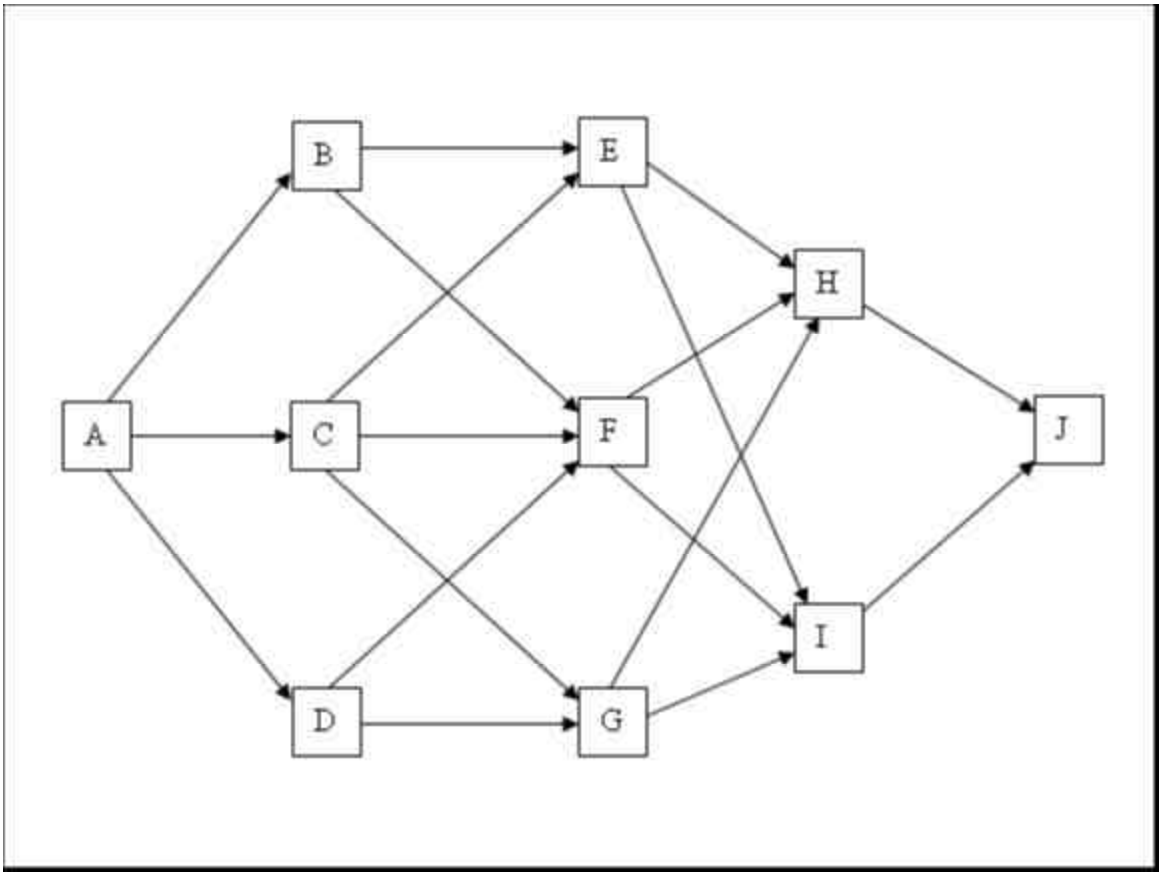


Fig. 6. Illustration of the possible routes in stagecoach problem.

The distance traveled between the cities and the cost of services in the rest house at each city are shown in Table 3.1 and Table 3.2 respectively.

Table 3.1: Representation of distance (miles) between cities in form of a table.

To / From	B	C	D	E	F	G	H	I	J
A	22	19	15	-	-	-	-	-	-
B	-	-	-	16	15	-	-	-	-
C	-	-	-	22	16	22	-	-	-
D	-	-	-	-	19	22	-	-	-
E	-	-	-	-	-	-	16	18	-
F	-	-	-	-	-	-	18	11	-
G	-	-	-	-	-	-	15	13	-
H	-	-	-	-	-	-	-	-	23
I	-	-	-	-	-	-	-	-	29

The table shown above has entries only for the routes existing between two cities as shown by arrows in Fig. 6.

Table 3.2 : Table showing cost of staying at each city.

CITY	A	B	C	D	E	F	G	H	I	J
COST	-	9	7	6	23	45	8	12	23	-

In Table 3.2, cost is not indicated for cities A and J as the businessman starts the journey from city A and ends in city J and hence does not stay at a rest house.

The original stagecoach problem that required finding the shortest route from city A to city J without any constraint of staying at each city can be solved by directly applying principles of DP for the 'distance' variable only. The stagecoach problem as phrased above involves optimizing the total cost inclusive of costs associated with the 'distance' variable (cost of traveling at $10/mile between cities) as well as the 'city cost' variable (cost of staying in each city). This implies combining the two variables for an effective approach to optimize total cost using DP. Let the combined entity be named *TOTAL COST* and defined as the algebraic sum of LOCAL *COST* of the rest house and the *TRANSITION COST* between two cities.

Total Cost = Local Cost + Transition Cost,
where, Transition Cost = distance between the cities * 10.

It should be noted that the above equation gives the total cost of reaching a city from another city. For example, the total cost a businessman incurs if he chooses to travel from city A to city B would be the sum of transition cost from A to B and the cost of staying in rest house at B.

DP is based on the principle that the solution to a large task can be obtained from solutions to a series of subtasks of the large task. For the stagecoach problem, there are a total of 5 stages through which the journey is made. Each stage consists of cities through which the businessman can choose to travel. For example, stage II has three states (cities B, C & D). The businessman can only travel through one city in each stage to reach a city in the next stage. Figure 7 shows the stages and states as described above. Applying DP on the *'Total Cost'* variable gives the minimum cost path of travel from city A to city J, through the different stages outlined in Fig. 7.

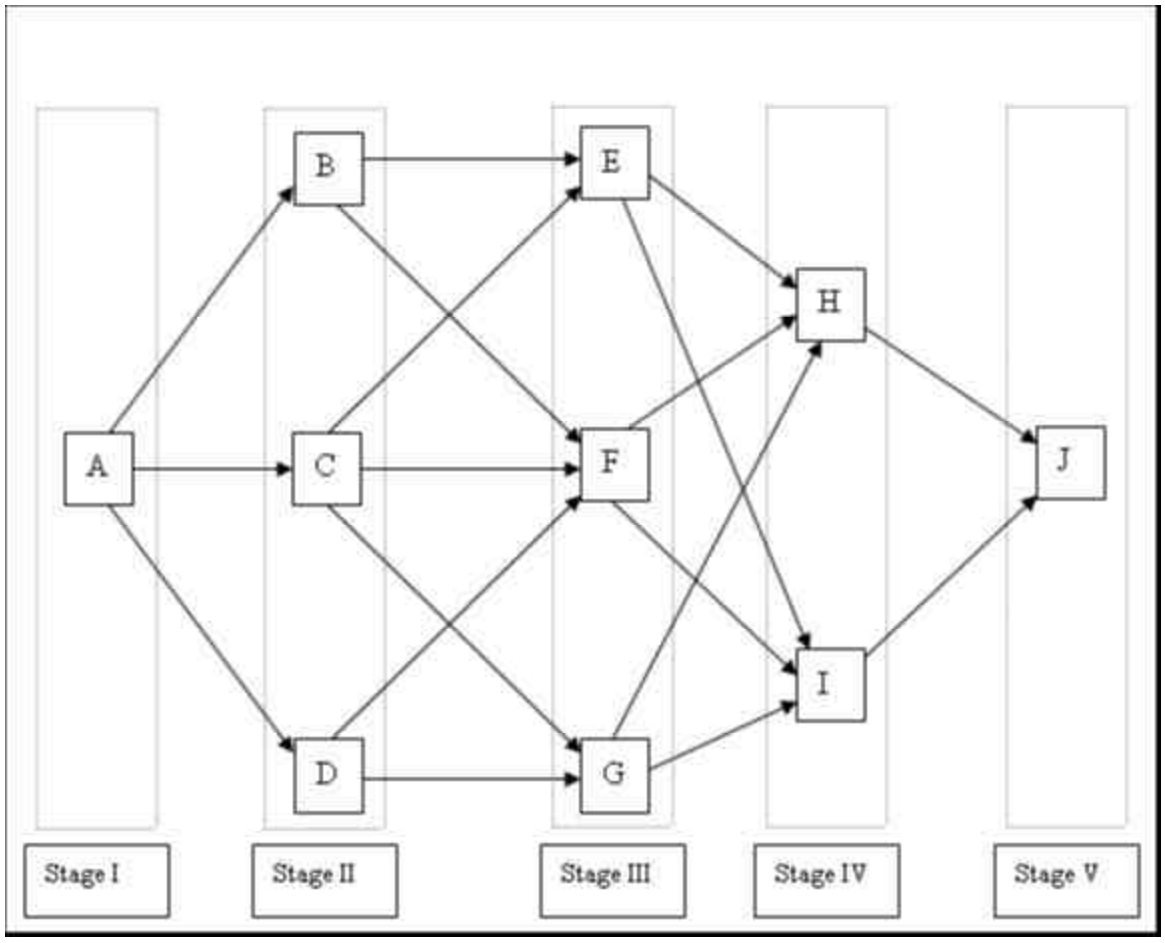


Fig. 7. Depiction of stages and states in the stagecoach problem.

For this example, the lowest cost path could easily be found by evaluating all possible paths (14 in all for this example), evaluation of the cost of each possible path, and then simply finding the minimum cost path. However, for larger more realistic examples, the number of possible paths grows exponentially with the number of states and thus becomes intractably large. Dynamic programming gives a much more computationally efficient method for determining this lowest cost path. In particular if one knows the minimum cost of going through a certain intermediate state, say state E, then in considering paths going forward from state E, one only needs to know this minimum cost path up to E and its cost in leading to and including E. It is not known (nor needed to be known) if the minimum overall cost path from the starting point

to the destination will include stage E. The stages that follow E and the destination which would increase the cost of the path that includes stage E.

Note, however, that it is necessary to determine the minimum cost partial paths to all states at each stage. For example, in the above example, we must compute the partial path costs to E, F, and G at stage III. Even if the lowest cost partial path going forward to stage III is via state E, it may later turn out that the lowest cost final path in fact goes through state F at stage III, since the partial path going forward from state E may be lower than the partial path going forward from state F. The overall DP process consists of finding all the lowest cost partial paths going forward for each state (city) at each stage, from the source to the destination, and also noting the previous state for each partial path (back pointer). When the destination stage is reached (and there is only 1 state for this destination), the lowest cost overall path is found by "back tracking" through all the previous states, making use of the back pointers saved for each state.

To be even more specific, the DP solution to the stagecoach problem is as follows. Assuming the businessman comes to stage III, he has a choice of going through cities E, F, or G of this stage. Assuming he knows (and in fact this is the case), that the minimum cost path of going through city E is 150, the minimum cost path of going through F is 125, and the minimum cost path of going through G is 160. Furthermore, he knows that the minimum cost path through E also requires that he go through city B at stage II, the minimum cost path through F requires that he go through city C at stage II , and the minimum cost path through G at stage III requires he goes through city C at stage II (the back pointers). Then, in determining the possible cities to go through at stage IV, he only needs to consider only the minimum cost paths to each of the cities in stage III, and the costs that will be incurred for travel from cities in stage III to cities in stage IV. Again the minimum costs to each city in stage IV are determined along with back pointers to stage 3 cities. To terminate the solution, note there is only 1 possible city at the final stage. The recursive method just outlined thus provides an answer to the lowest cost path overall, and a back pointer to the lowest cost city at the next to the last stage. Similar back pointers can be used to determine the overall route.

Using the process described above, the optimal path of travel from A-J for the example given is found to be A-D-G-H-J.

The problem of pitch period marker detection can be formulated in a manner similar to the stagecoach problem, as shown in later sections. Thus dynamic programming can be used to help solve the pitch period detection problem.

3.3 Algorithm

This section describes the algorithm to obtain pitch period markers on a cycle-by-cycle basis using prior knowledge of pitch values of the signal. The algorithm works broadly by transforming the pitch period marker detection problem into a form compatible with DP.

As the pitch values for the signal are assumed known, an approximation of the pitch period values can be made. The next step involves identifying actual pitch periods that are closest to the approximate pitch period values using peaks as the cycle markers. Since speech contains an extremely large number of peaks, this would result in a large number of probable peak pairs that may represent a pitch period, and hence a large number of calculations would be needed to identify the peak pair closest to the approximate pitch period. Even if the large numbers of calculations required are overlooked, the pitch values obtained from a pitch tracker may be inaccurate and thus affect the ability to accurately identify the markers. Hence DP was chosen for combining the "local" information (peak amplitudes) and the "dynamic" information (spacing between peaks as compared to spacing expected from the pitch track).

In order to apply DP, it is necessary to know the total number of stages, or more specifically the total number of pitch cycles, over an interval. That is, there appears to be no convenient way to use DP for identifying pitch markers if we do not know the total number of markers that should be found in the speech signal. Thus, in order to use DP, we divide the speech signal into "blocks," each of which is an integer number of pitch periods long (with pitch period determined by the tracking routine) and then into subframes (stages), each of which is expected to have one valid pitch marker. Although

this formulation allows DP to be used and results in fewer calculations, it still has problems when there are large errors in the pitch track, as explained later in the beginning of section 3.3.4. Another reason for using blocks is that the errors in large voiced regions caused by averaging pitch values can be minimized. DP is used to fit the approximate pitch period values to the actual pitch period observed in the speech signal.

To be more specific, the algorithm for pitch marking consists of the steps given below:

1. Pre-processing
2. Block creation process
3. Peak picking process
4. Peak organization into subframes
5. Dynamic programming
6. Post-processing of pitch markers.

The steps involved in the algorithm are also shown in Fig. 8. The first three steps shown in Fig. 8 define the process that allows application of DP to pitch period marker detection.

3.3.1 Pre-processing: determination of signal polarity

Peak identification is an important aspect of this overall algorithm. It was experimentally observed, with the Keele database, that some speech signals appeared to have more prominent positive peaks, whereas other signals had more prominent "negative" peaks. In order to determine the "best" polarity of a speech signal, all voiced regions of speech are identified in the first step. In the second step, all peaks are located with a window size of 1.4*pp (see section 3.3.3 for more about peak picking) in the voiced regions. Then the average peak amplitude over all the voiced regions is calculated. Similarly, another average of the peak amplitudes over all the voiced regions is calculated, but using the inverted speech signal. If the average peak amplitude is found to be greater for the original signal than the inverted signal, the original signal polarity is maintained for the remainder of the processing. Conversely, if the inverted signal has larger average peak values, the signal is inverted and used for all subsequent processing

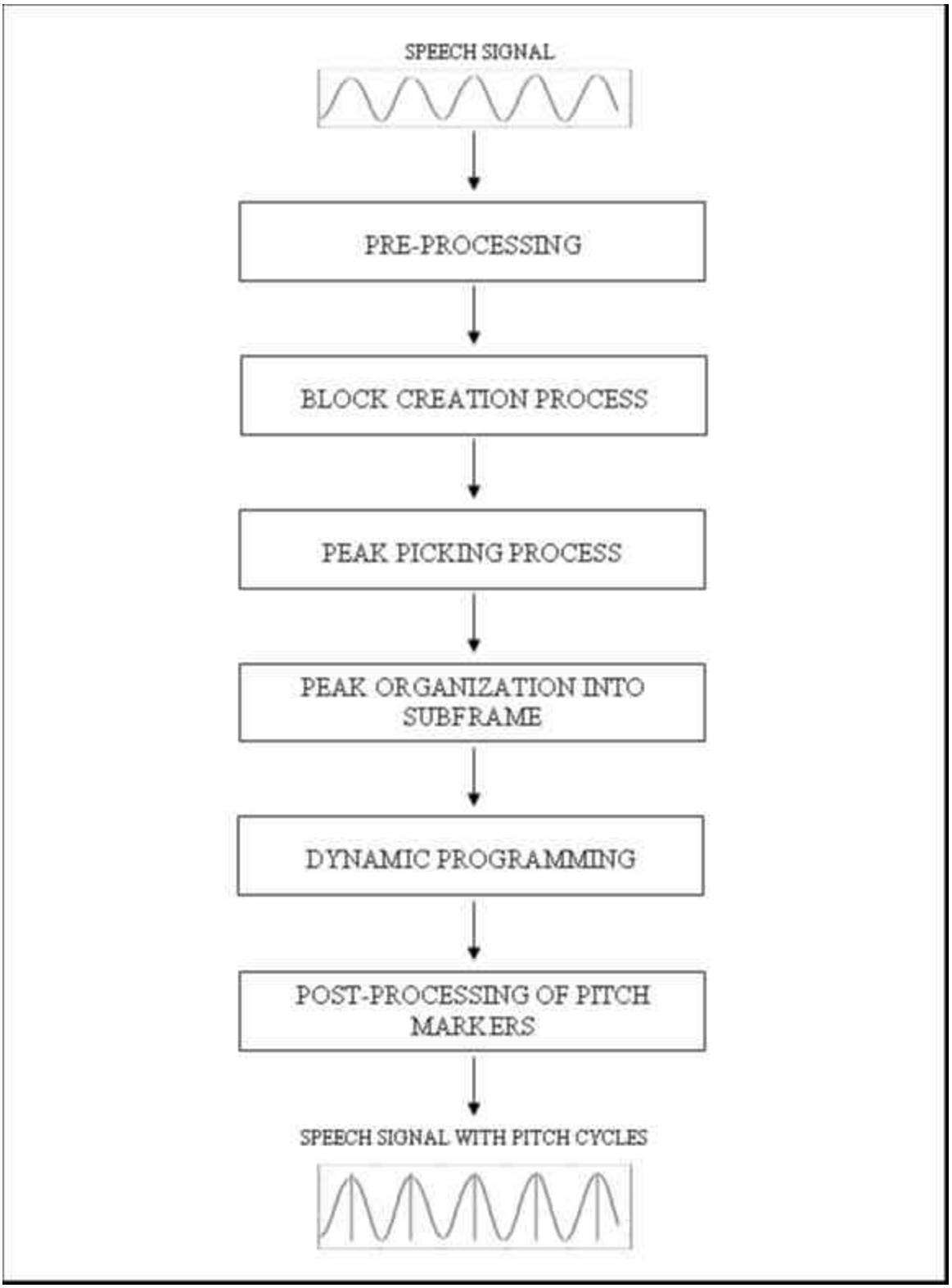


Fig. 8. An outline of the pitch-marking algorithm.

steps. Experimental results related to this signal polarity step are given in section 4.7.1 of chapter IV.

Another preprocessing step was to low-pass filter [17] the speech with an FIR filter of order 100 and cutoff frequency of 1000Hz. This filtering smoothes the signal and simplifies peak peaking. Experimentally, it was found the smoothing improved accuracy by a slight amount.

3.3.2 Block creation process

A 'block' of the speech signal can contain any of the four categories of regions u-u, u-v, v-v or v-u or a combination of them, as was discussed in chapter 1. However, the fundamental assumption is made that blocks are short enough in time (typically about 5 pitch periods, or about 30-60 ms) that at most one transition in voicing is made. Thus we really only consider the four possibilities listed, and of these only the last three categories are of interest as pitch periods are defined only for the voiced regions.

Identification of pitch periods in the case of totally voiced regions (v-v) of speech is generally relatively easy and accurate, as continuity in speech characteristics is maintained in v-v regions, as compared to regions where the speech contains transitions between voiced and unvoiced regions (u-v, v-u). The algorithm is used mainly to identify pitch cycles in voiced regions (v-v) of speech, but does attempt to identify pitch period makers in the voiced portions of (u-v, v-u) regions, and even "examines" portions of the "u" regions for these cases. However, as described in more detail in the experimental section, the (u-v, u-v) regions were the most difficult in terms of locating pitch markers, and it appears those are the regions where the algorithm is most likely to fail. It does not attempt to identify cycles in unvoiced region.

The block creation process is the first among the three steps that make possible application of DP to this problem. This step basically involves dividing the speech signal into smaller chunks called blocks. Division of the speech signal into blocks requires

knowing the size of the block to be created, the starting point for the block, and the pitch values in the region of the block. The algorithm presently uses pitch values obtained from the Old Dominion University developed pitch tracker YAPT [15].

The size of a block is a fixed multiple (typically 5) of the average pitch value over a 60ms long region. A pitch value obtained from YAPT is defined every 10ms, implying the averaging is done over six consecutive pitch values. The block size is a multiple of the average pitch period in order to obtain a minimum of 1-2 accurate pitch cycles in the region, compensating for the averaging done to obtain pitch values. As the block size is dependent on the average pitch period, it varies depending on the pitch values of the region. After obtaining the block size, a block is selected from the speech signal beginning from the location indicated by the starting point. This process is done only when the 60ms long region is totally voiced and also when the 10ms region immediately before the 60ms long region is also voiced. That is, the block is assumed to be totally voiced. This process is as shown in Fig. 9.

It should be noted that on encountering a voiced region of any size, a block is always extracted from the speech signal. The size of the block is always obtained as a multiple of the average of the pitch values of the voiced portions in the region. Thus it is sometimes possible that the size of the block may be greater than the size of the voiced region, and in such situations the block will include some portions of unvoiced regions.

Although the process above is described for v-v transitions in speech, it can also be applied if the region contains unvoiced to voiced (u-v) transition, voiced to unvoiced (v-u) transition or if there are rapid voiced and unvoiced transitions, by making changes to the process as described below:

a) When the region has a u-v transition, then the process is implemented as described except that the starting point, which is at the beginning of the voiced region, is now moved two pitch periods “back” into the unvoiced region.

b) When the region has a v-u transition and the size of the voiced part is found to be less than the block size, then, as mentioned earlier, the block will also include some portions of the signal in the unvoiced region.

c) When the region has rapid voiced and unvoiced transitions, then the block size is obtained as described earlier and the block may also include some unvoiced regions. The starting point is also changed, if the region starts with a u-v type transition, as described in a).

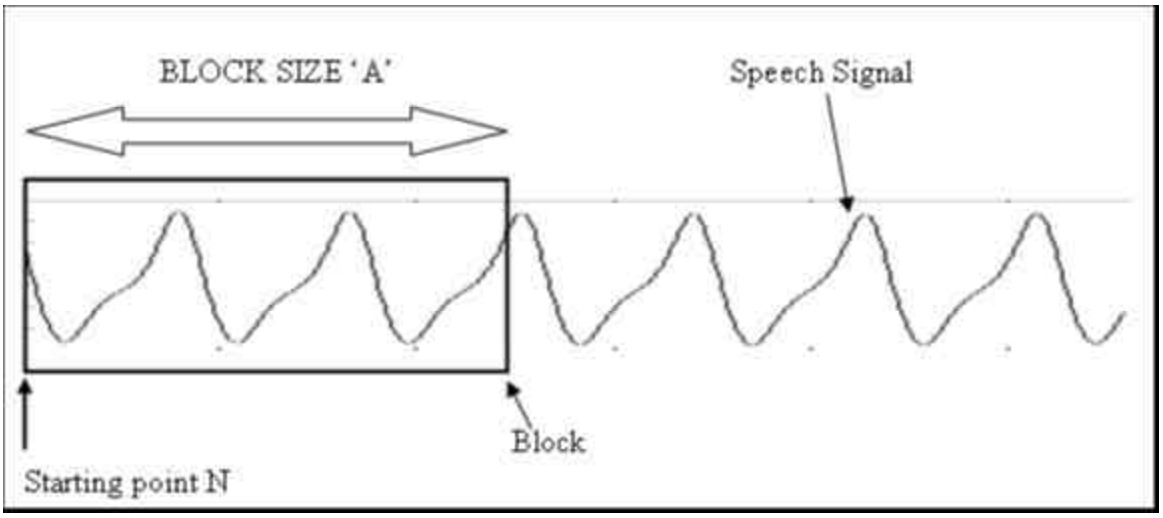


Fig. 9. Depiction of block creation process in voiced region of speech signal.

The starting point for a new block is also dependent on the location of the last pitch cycle in the block of signal contiguous to the signal in the new block. Changes described for u-v and v-u regions in a, b, c above, are aimed at addressing the issue of identification of pitch cycles in these regions as discussed earlier in section 3.

3.3.3 Peak identification process

Identification of pitch cycles requires use of landmarks that help mark the beginning and the end of a pitch cycle. The two widely used landmarks to identify pitch

cycles are peaks and zero-crossings of the signal. The use of zero-crossings as landmarks to identify pitch cycles is plagued with the problem of identifying the zero-crossing point, when the speech signal is not symmetrical with respect to the zero line. As the possibility of this happening cannot be discounted (and was found in practice to occur), this algorithm uses peaks as landmarks to mark the pitch cycles.

The next step after the block creation process is peak identification or peak picking. This step involves locating peaks of the speech signal in the block that would be the most probable candidates to act as landmarks demarcating the pitch cycles. In this process peaks are identified as the candidate markers for pitch cycles/periods using a sliding window method. The result of this step is a collection of the candidate markers for pitch cycles. Before beginning with peak picking the signal in the block is normalized so that the largest signal value is equal to one.

A sliding window of size 1.0 times the average pitch period is used over the speech signal present in the block. As the window is advanced over the block point by point, the middle point of the window is examined to determine if this middle point is the largest values in the window. Whenever this middle value is found to be the largest point in a window, it is considered as a peak and as a candidate marker for delineating the pitch periods. After sliding the window over the whole block, a list of the probable markers delineating the pitch cycles is obtained. Upon identifying the candidate peaks, their normalized amplitude values and locations values are stored.

As the cities in the stagecoach problem were associated with the cost of staying for a day, called as *LOCAL COST*, the candidate markers are also associated with a cost depending on normalized amplitude values. In particular larger peaks are the best candidates for markers, and thus have the lowest costs. The actual local cost of each peak is given by

Local cost = 1 - normalized amplitude.

Figures 10 and 11 show the peaks identified at the end of the peak picking process.

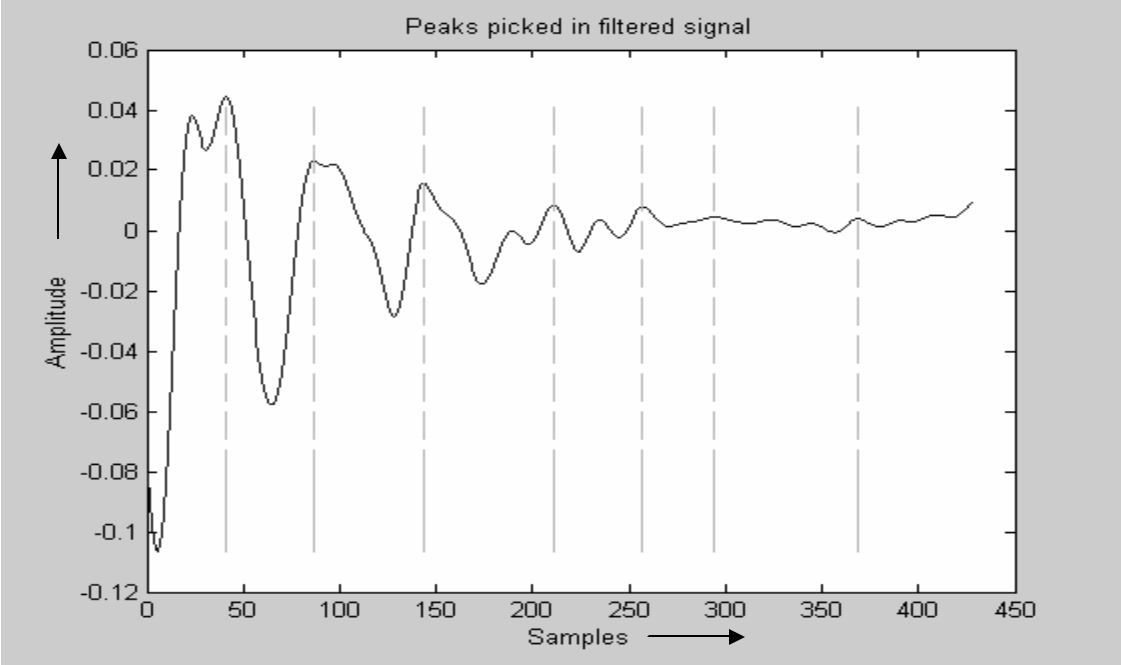


Fig. 10. Candidate peaks identified in peak picking process in region containing v-u transition.

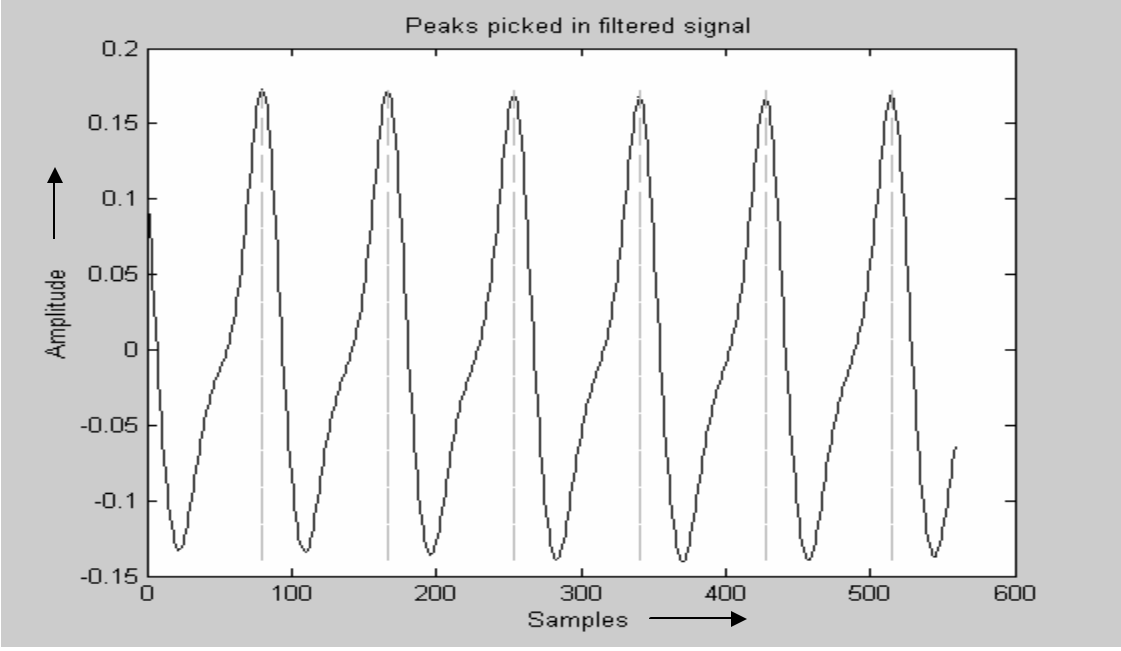


Fig. 11. Candidate peaks identified in peak picking process in region containing v-v transition.

3.3.4 Using dynamic programming

The next step after peak picking involves classification of the peak into subframes, which is analogous to defining stages in the stagecoach problem. This step, although a distinct process as shown as the third step in the Fig. 8, has been combined with the DP stage for better explanation purposes.

The "proper" definition of subframes is critical to enable DP to be used for the finding the lowest cost set of markers, and to find a set of markers that are likely to be the pitch period markers. In particular, in order to use DP, it is assumed that we have a fixed number of stages, and know the number of stages. For the case of pitch marker detection, we assume that each frame contains one valid marker for the beginning of a pitch cycle, and that the end of that pitch cycle occurs at some point in the next subframe. We also assume that the total number of pitch markers (analogous to stages) is equal to the number of subframes. This strategy breaks down, if the actual pitch periods deviate "too much" from the nominal pitch period. Keeping in mind the errors possible in the pitch track and errors caused by the varying nature of pitch cycle lengths, a framework for using subframes has been devised as follows:

The classification into subframes is done by dividing the block into overlapping subframes of size 2*pitch period. Each subframe is assumed to contain at least one beginning marker point for a pitch cycle. Subframes are advanced by one pitch period. Typically a block will contain a total of 4 subframes (call this value K). After dividing block into subframes, the candidate peaks obtained from peak picking are then distributed among the subframes based on their peak locations and the region being represented by each subframe. This may lead to some peaks been present in more than one subframe as there is an overlap of one pitch period between any two consecutive subframes. Each of the subframes contains a fixed number (M) of peaks (typically 10). In subframes that do not have the required number of peaks some fictitious peaks (but with very high local cost) are generated.

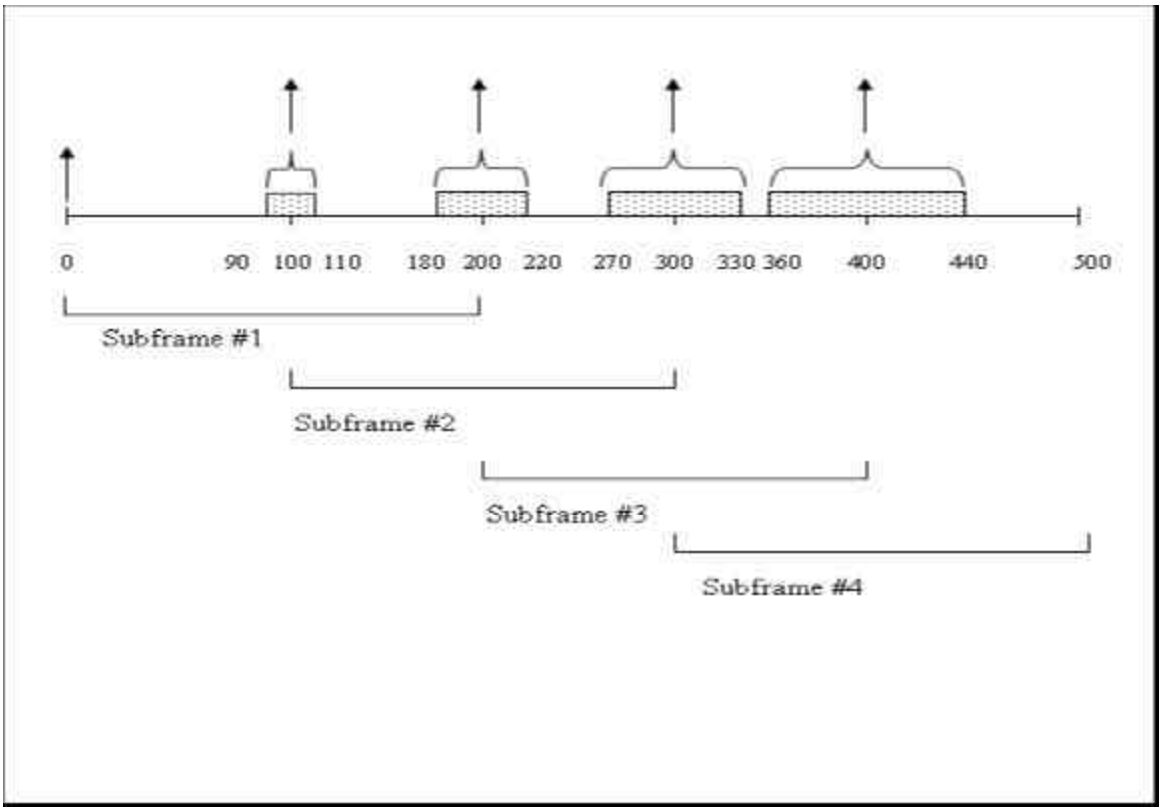


Fig. 12. Illustration of the scheme for use of subframes. The shaded blocks represent the growing regions in which a pitch marker could be assuming there is ±10% maximum error of the assumed actual pitch period, assumed to be 100 sample points. The figure shows as the number of subframes increases the assumptions made in order to be able to use DP may not hold true. This figure assumes that the first actual pitch marker is situated at the beginning of the block.

The framework described above allows the assumptions made earlier to be valid as long as pitch tracking errors are not too "large," as explained below. For the sake of explanation, assume that the error in pitch track and variability in pitch cycle lengths is no less than .9 *pp and no larger than 1.1 *pp, that is the sum of the errors (error in pitch track and the error due to the varying nature of pitch cycle lengths) is not greater than ±10% of the actual pitch period (assumed for explanation to be 100 sample points). As may be understood from the subframe framework, the region in which a pitch marker ought to lie increases in size by 20 sample points with every subframe and can eventually exceed the block size itself. Hence, there is the need to have only a fixed number of

subframes in a block with the size of a block being no greater than 5*pp. This idea is illustrated in Fig. 12.

After determining the peaks and classifying them into subframes, two matrices are formed. One matrix called A stores the *LOCAL COST* values and the other matrix called B stores the *TRANSITION COST* values. Matrix A contains M rows and matrix B contains M*M rows. A has K columns, and B has K-1 columns. The entry in matrix A or the *LOCAL COST* values is equal to 1- normalized amplitude of peak for subframe j. The entries in matrix B are computed as a monotonically increasing function of the absolute difference between a peak location in current subframe and a peak location in the next subframe, as compared to the expected pitch periods. Thus matrix A summarizes a local cost of each candidate pitch marker (large peaks (low cost) are most likely to be the beginning of a pitch cycle). Matrix B summarizes transition costs associated with choosing a certain peak in one frame and a particular peak in the next frame (the transition costs will be lowest for peaks which have a spacing equal to the pitch period as measured in the pitch track. The costs will then increase as the spacing deviate from the expected spacing based on the pitch track).

The transition cost described above is defined as follows:

$$\text{Transition cost} = \left(\frac{\text{Est.pitch period} - (I - J)}{\text{Est.pitch period}} \right)^2$$

where, I = candidate peak location in i^{th} subframe,

J = candidate peak location in $i\text{-}1^{th}$ subframe and

Est.pitch period is the estimated pitch period.

Dynamic programming is then used to find the lowest weighted cost path through the above two matrices. The results of this path are the most likely pitch markers (beginning of pitch cycles) over the interval processed. This process is illustrated in Fig. 13. Only the first K-1 of these markers are retained as the final markers; the last marker is also the starting point for the next block, with a preceding 10ms or longer voiced region. However, if the region next to the $K\text{-}1^{th}$ marker is unvoiced then the starting point for the

next block is set to the beginning of the next voiced region. Thus as stated earlier in section 3.3.2, the starting point of the next block is affected by the last marker of the previous block.

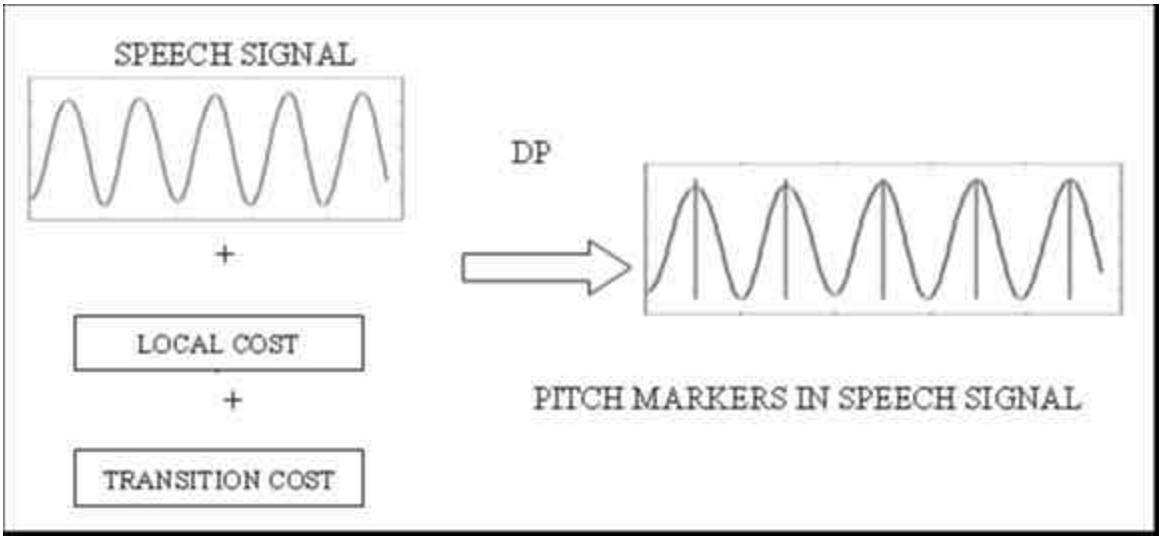


Fig. 13. Illustration of the process for obtaining pitch markers in speech signal using DP.

For regions of signal with v-v transitions in the preceding block and the new block, the analysis for this region is the same as that described above, except while determining the transition costs, the last marker from the previous block is used to compute transition costs to the marker candidates in the first subframe of the new block. For the remaining types of transitions such as u-v and v-u, the analysis is as described without any changes.

The fifth and the final step of post-processing as shown in Fig. 8, has also been combined with this section. This step is used to vet the list of markers obtained from DP, if they are obtained from blocks which contain some unvoiced regions or if the block had u-v and v-u type transitions. Markers that are found in unvoiced region are retained only if they have large amplitude values whereas markers in voiced region are retained irrespective of their amplitude. Figures 14, 15, 16 and 17 show examples of the final set of markers obtained for v-v, u-v, v-u and u-v-u-v regions of speech.

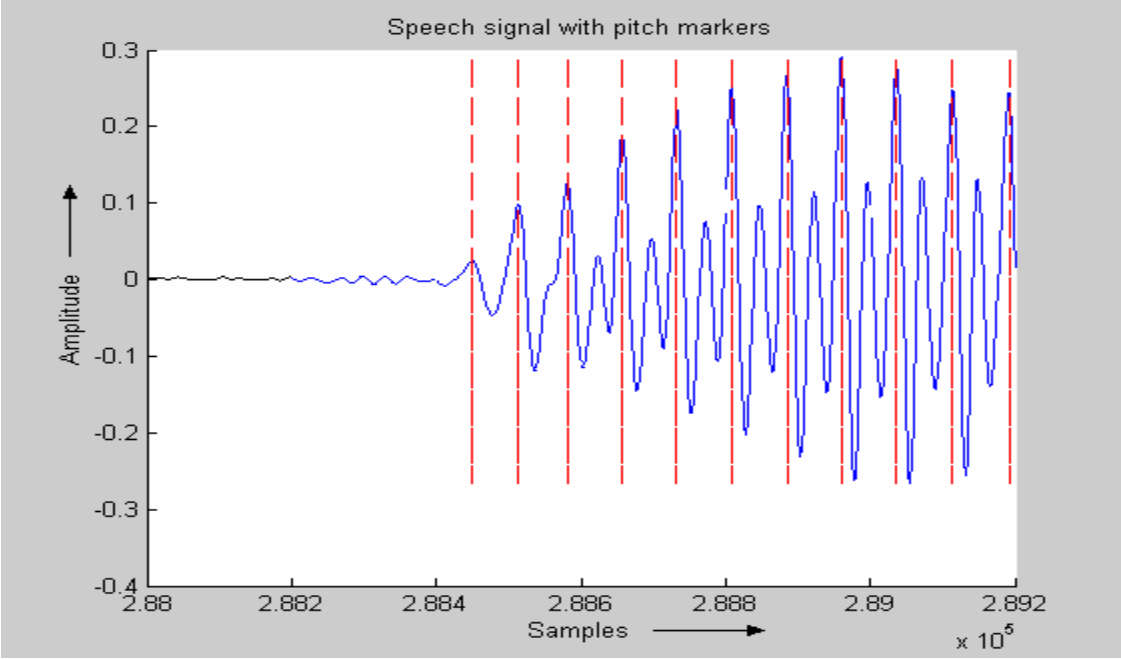


Fig. 14. Pitch markers obtained in a region of speech with u-v transition. Signal in blue represents voiced region and signal in black represents unvoiced region.

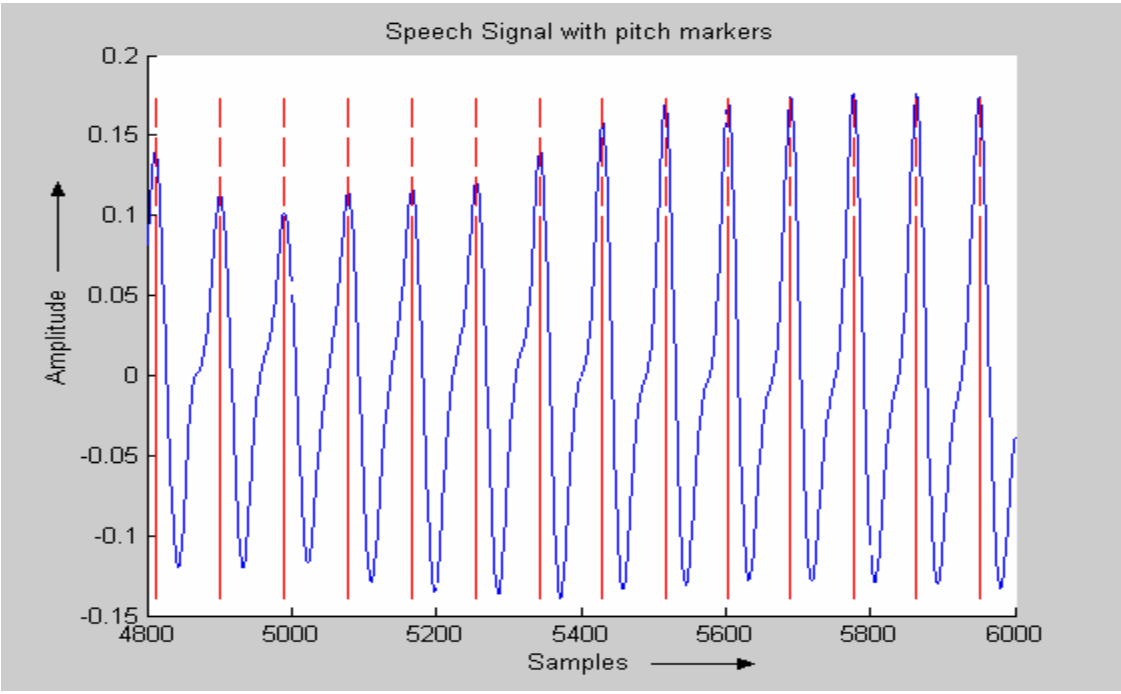


Fig. 15. Pitch markers obtained in a region of speech with v-v transition.

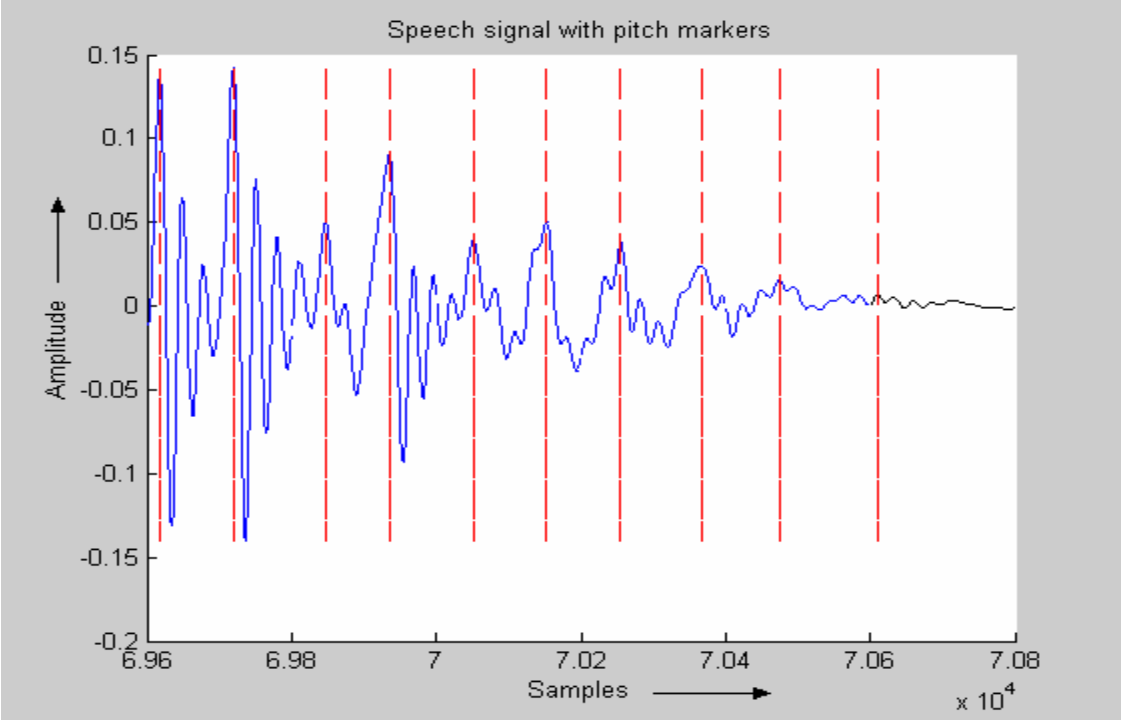


Fig. 16. Pitch markers obtained in region of speech with v-u transition. Signal in blue represents voiced region and signal in black represents unvoiced region.

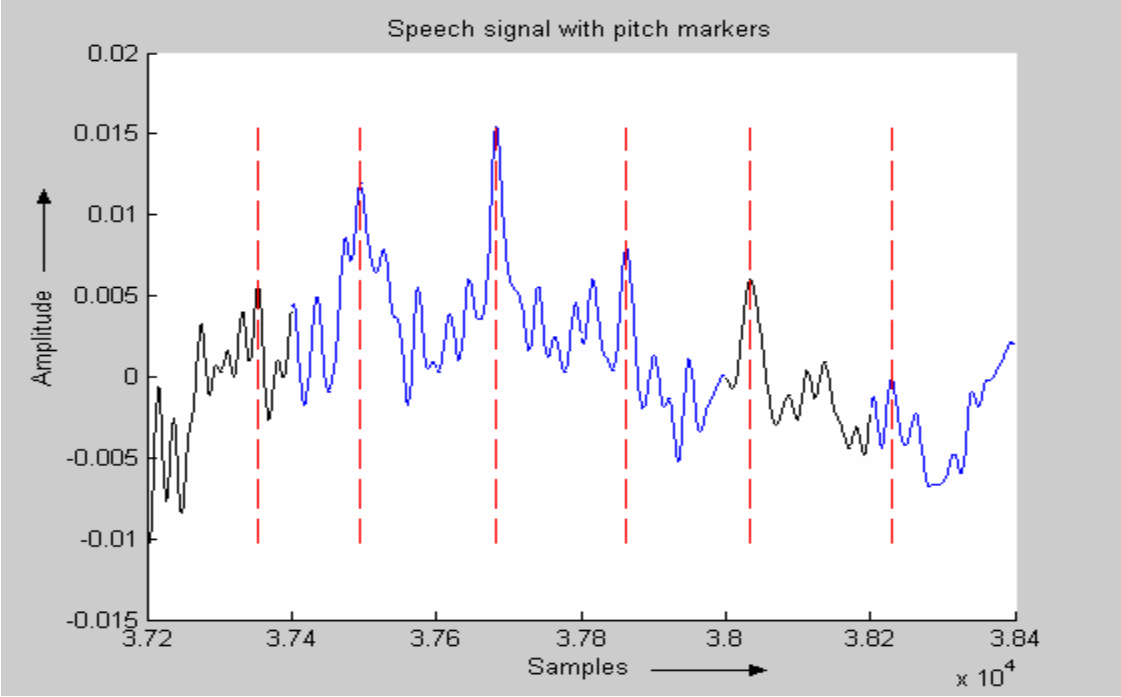


Fig. 17. Pitch markers obtained in region of speech with rapid unvoiced to voiced to unvoiced to voiced (u-v-u-v) transitions.

The processes described in sections 3.3.2-3.3.4 are repeated until the last voiced region of the speech signal has been processed.

3.4 Summary

This chapter gives the detailed explanation of the pitch period detection algorithm. The algorithm has been supported by figures and tables. Dynamic programming is used to accurately determine the lowest cost (or best fitting) individual pitch cycles.

The following chapter gives an experimental validation and evaluation of the algorithm with the help of the laryngograph signals supplied along with the Keele database.

CHAPTER IV

EXPERIMENTS, EVALUATION AND ANALYSIS

4.1 Introduction

In the previous chapter, an algorithm to identify pitch markers in the speech signal was introduced. In this chapter, experiments, evaluation and analysis are presented, all of which illustrate the accuracy of speech markers obtained using the algorithm described in chapter III. The analysis of the accuracy of the speech markers, in addition to the algorithm described in the previous chapter for locating the markers in the acoustic signal, requires an algorithm for locating control markers and a set of error measures. In the present chapter, the algorithm for locating reference markers in a control signal is defined and also various error measures are discussed, which are used to quantify the accuracy of the speech markers. Several experiments and results are presented. First, however, the Keele speech database (used for experiments) and the importance of selecting it are discussed.

4.2 Database description

The Keele speech database provided by Keele University, UK, consists of studio quality and telephone quality speech signals. Each set contains a total of 10 sentences, spoken by 5 different male and 5 different female speakers. The studio quality and telephone signals are made from the same original recordings; the telephone set was obtained by passing the original recordings through telephone lines and recording the telephone signal outputs. Thus the studio and telephone signals have the same pitch track. For each speech signal in the database, another simultaneously recorded signal called the laryngograph signal is also provided. The laryngograph signal is a representation of the closures and openings of the vocal cords and hence is closely associated with fundamental frequency. The inclusion of the laryngograph signal in the

Keele speech database was a major factor for selecting the database for testing purposes, as it provided a means of extracting reference markers. Each signal in the database (speech or laryngograph) is approximately 30s long and sampled at 20 kHz. The database also includes a reference pitch track of each speech signal, which was used extensively in the pitch marking process.

Another well-known speech database, the MOCHA database from the Center for Speech Technology Research at the University of Edinburgh, contains laryngograph signals along with speech signals, but the average length of the signals is very small, about 300ms long, as compared to approximately 30 seconds for each sentence in the Keele database. As a longer duration signal provides a better opportunity for testing, the Keele database was chosen as the main test database, although some initial testing was done on the MOCHA database too.

4.3 Algorithm for extracting reference markers

This section describes the algorithm for extracting reference markers from the laryngograph signal. Initially, the algorithm described in chapter III was used to obtain markers in the control signal. However, extensive visual examination of the laryngograph waveform with markers superimposed revealed locations with errors. As the reference markers need to be as accurate as possible, a new (simpler) method was designed for detecting reference markers, which was arrived at after extensive empirical testing. It is important to note that since the laryngograph signal is considerably different than the acoustic signal, it is not surprising that different methods were needed for locating pitch markers in these two signals.

The new method, described here, is based primarily on simple peak picking, but with intervals determined by the reference pitch track. Although pitch cycles are clearly identifiable in the laryngograph signal, some signal processing is still required in order to accurately locate pitch cycles in the laryngograph signal. This is done by first taking a first order difference of the laryngograph signal, as this creates much more prominent

peaks for each pitch cycle. The first order differencing also removed the large very low-frequency components often observed in the laryngograph signal. In the remainder of this work, the first order differenced signal is called the control signal. Figure 18 shows the laryngograph and the control signal. The reference pitch track, supplied with the Keele database, was used to guide the extraction of reference markers.

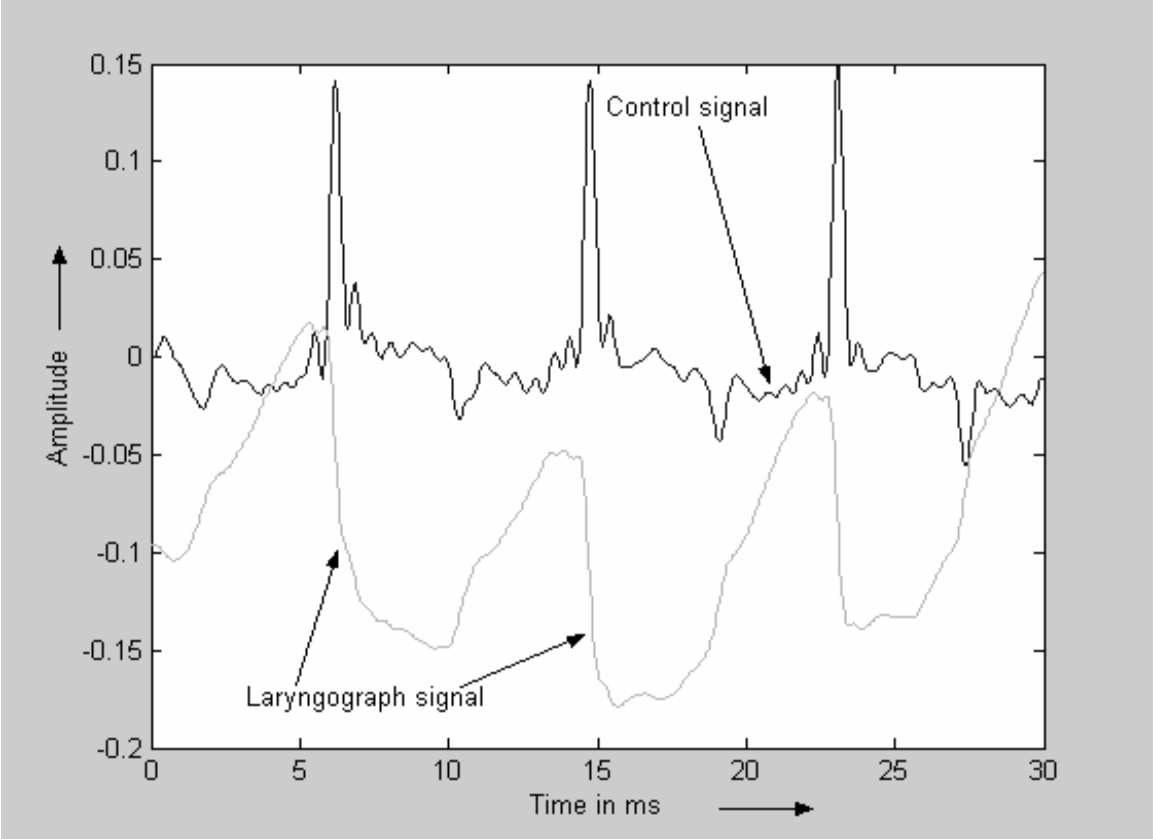


Fig. 18. Illustration of the laryngograph signal and the derived (by first-order differencing) control signal. As can be observed the peaks are much more prominent in the control signal.

The processing is performed on non-overlapping blocks of the signal, with each block approximately 5 nominal pitch periods long. The same process of block creation mentioned in chapter III is used to divide the control signal into blocks. Also, as mentioned in the previous chapter, there are fours kinds of blocks: (1) blocks for which the reference pitch track indicates the signal is voiced in the entire interval. (2) Blocks for which the reference pitch signal indicates the signal makes an unvoiced to voiced

transition someplace in the block. (3) blocks for which the reference pitch signal makes a voiced to unvoiced transition someplace in the block. (4) blocks for which the reference pitch track is completely unvoiced. These blocks are processed differently, as discussed later. The basic assumption, based on extensive examination of the signals, is that voiced blocks (type1) have pitch cycles throughout the block, that voiced to unvoiced blocks (type 3) and unvoiced to voiced blocks (type 2) may have some valid pitch cycles in both the unvoiced and voiced regions. Unvoiced blocks are assumed to have no valid pitch cycles and therefore do not need to be examined for pitch markers.

As mentioned previously, the algorithm for locating reference pitch markers is based on simple peak picking, a process similar to one described in chapter III except that different sizes of moving windows are used.

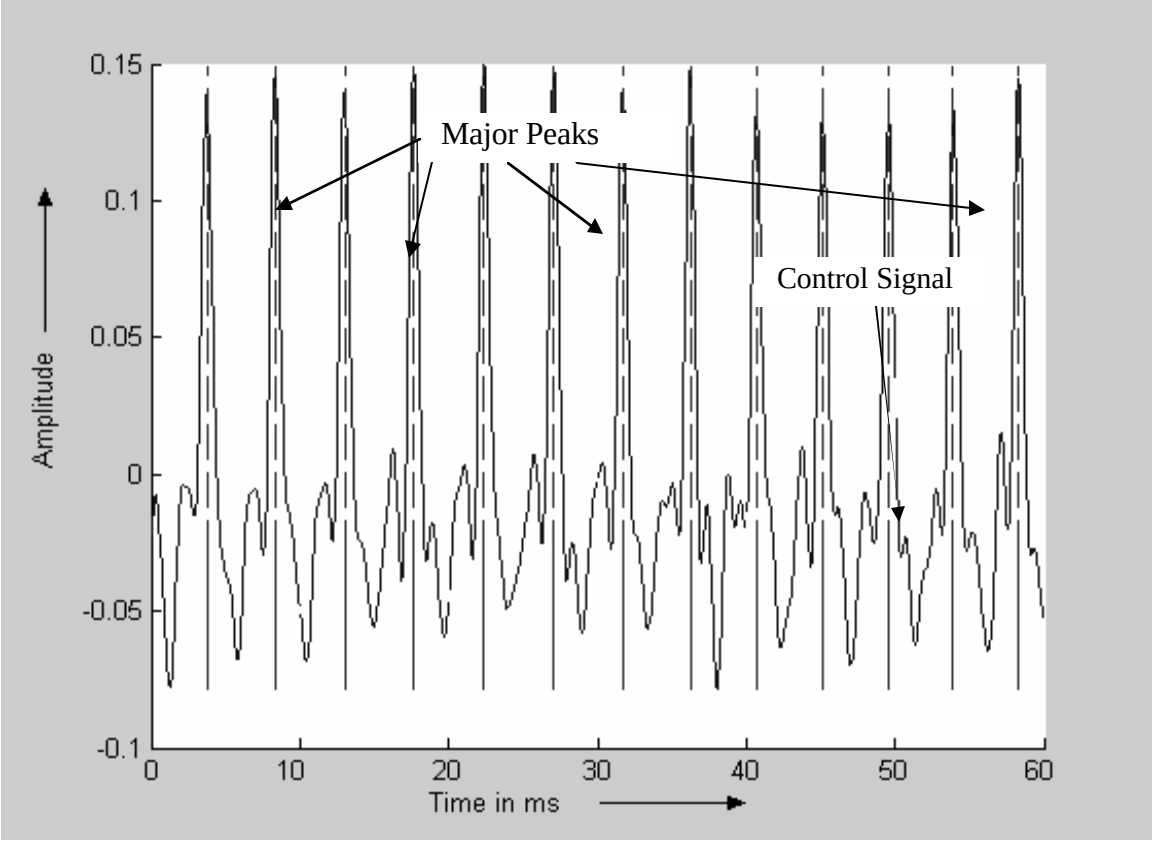


Fig. 19. Major peaks (indicated by vertical lines) in the control signal (first-order differenced laryngograph signal) obtained using ± 0.5*pp wide window.

The algorithm for locating pitch markers in the control is as follows for each block:

1. When the signal in the block is entirely voiced, all peaks are identified that are the largest in amplitude over an interval that is 0.5 times as long as the pitch period (pp) in either time direction (that is, the largest peak in each pp interval with the peak at the center). These peaks are labeled as definitive pitch markers in the control signal and are called major peaks, as shown in Fig. 19.

2. When the signal in a block makes an unvoiced to voiced transition, peaks are found as described in step 1, but with the additional constraint that the normalized amplitude be greater than 0.0075, in order to consider a peak as pitch marker. The normalization is computed with respect to the highest amplitude signal in the block. It was experimentally observed that peaks below 0.0075 in normalized amplitude generally do not appear to correspond to pitch cycles.

3. When the signal in a block makes a voiced to unvoiced transition, peaks are located only for the voiced region, and using the method described in step 1. No peaks are located in the unvoiced region.

4. Since it was observed that in some cases “valid” peaks are missed, as shown in Fig. 20, an additional step of processing is used to attempt to identify the missing peaks. This additional step requires performing peak picking again using a moving window of width ± 0.25*pp, rather than ± 0.5*pp, as in step 1. This peak picking process locates (again) all the peaks found in step 1, plus some additional peaks. The peaks found from step 1 and again "discovered" here are kept as definite pitch markers. The additional peaks found are considered as potential pitch markers and require further processing (described in next step) to be considered as definite pitch markers. These additional peaks are called minor peaks.

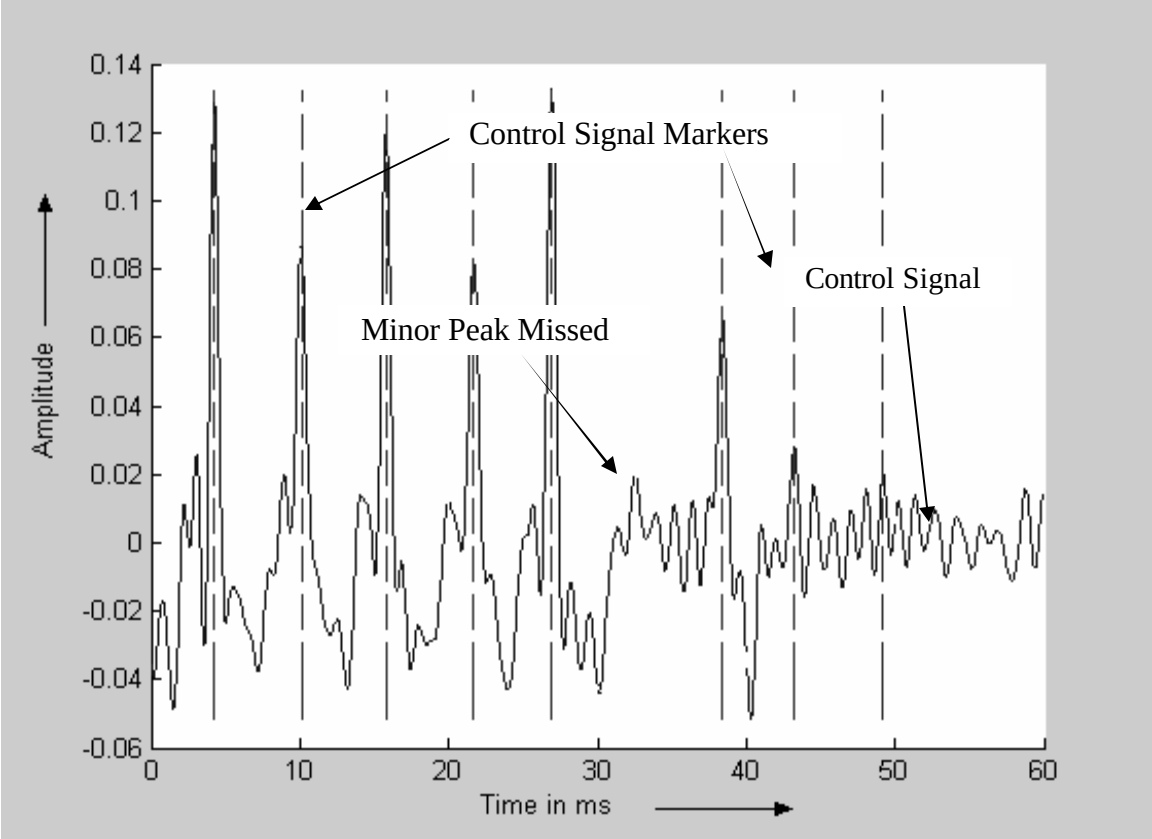


Fig. 20. Illustration that one of the probable markers, shown as a minor peak, has been missed. It also shows major peaks in the control signal (first-order difference of the laryngograph signal) obtained using ± 0.5*pp wide window.

5. The potential pitch markers obtained in step 4 are evaluated as follows:

   i. In this step minor peaks are identified that lie between two consecutive major peaks (located at T_i and T_{i+1}), which implies identifying peaks to the right of T_i and to the left of T_{i+1}. As this step is applied only on pairs of consecutive major peaks, it is not possible to identify minor peaks to the left of T_1 and to the right of T_n using this step. It should be noted that T_1 is the first major peak in a block, and T_n is the last major peak in a block.

   ii. Those minor peaks from step i, whose amplitude is greater than a certain fraction of both surrounding major peaks (typically more than 2/9) and also whose amplitude is greater than a certain percentage of the of the average

signal amplitude in the block (typically greater 2.5% are also labeled as pitch markers. For the case of very low average amplitude regions (<0.05, on as full-range scale of approximately +-1.0) of the signal, the threshold value of 2.5% is changed to 1.5%.

iii. This step is used to identify minor peaks to the left of T_1. As T_1 represents the location of the first major peak in the block, the block is searched from the beginning up to T_1.

iv. Those minor peaks from step iii, whose amplitude is greater than a certain fraction of the neighboring major peak T_1 (typically more than 2/13) and also whose amplitude is greater than a certain percentage of the of the average signal amplitude in the block (typically greater 0.75% are also labeled as pitch markers. For the case of very low average amplitude regions (<0.05) of the signal, the threshold value of 0.75% is changed to 0.6%.

v. Step iii is performed again to identify minor peaks to the right of T_n with the search range extended from T_n up to the end of the block.

vi. Those minor peaks from step v, whose amplitude is greater than a certain fraction of the neighboring major peak T_n (typically more than 2/13) and also whose amplitude is greater than a certain percentage of the of the average signal amplitude in the block (typically greater 1% are also labeled as pitch markers. For the case of very low average amplitude regions (<0.075) of the signal, the threshold value of 0.75% was used instead of 1%.

6. The peaks that have been labeled as pitch markers at the end of steps 1 and 5 are the final markers for the control signal, as shown in Fig. 21.

The control markers obtained at the end of this process, on visual inspection were found to be very accurate. Note that the various thresholds mentioned were found based on extensive empirical testing.

4.4 Error estimates

This section describes the process used for error estimation and various measures used to quantify the error.

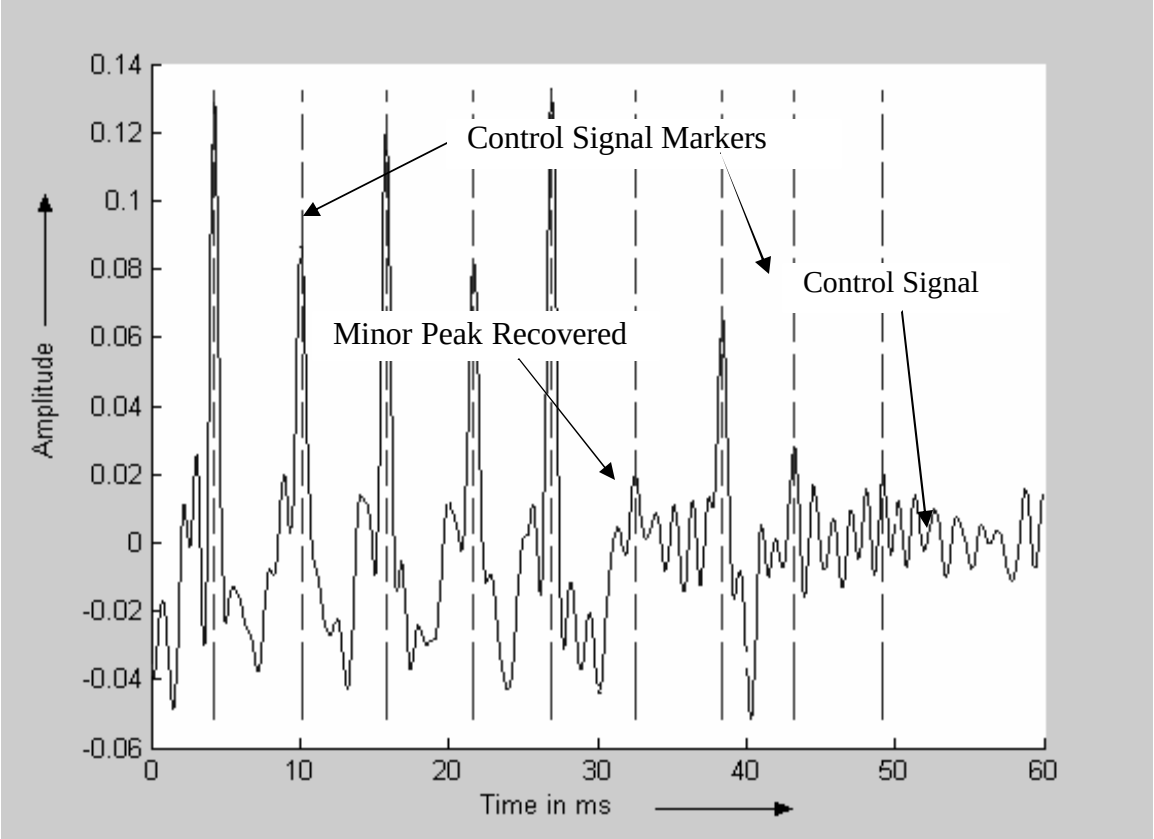


Fig. 21. Results obtained after using both ± 0.5*pp wide window as well as ± 0.25*pp wide window. Note that a pitch marker is now placed at the missing minor peak.

4.4.1 Error estimation process

After obtaining the speech as well as the control markers, a comparison is done to determine the accuracy of the speech markers. This error estimation process is as described below:

1. The first step involves locating speech markers closest to each pair of consecutive control markers (control markers located at C_i and $C_{i+1,}$ speech markers located at S_i and S_{i+1}).
2. In the second step the difference between the speech markers from step 1, $T_S = Si_{+1} – S_i$ is computed and also the difference between the control markers, $T_C = C_{i+1} - C_i$, is computed. Again another difference is computed between T_S and T_C. The difference between the differences, $TD = T_S – T_C$, is considered to be the error in the speech markers. If no speech marker is found close to either or both of the control markers, a default value (typically average pp) is used to define the error. Note that the difference of the differences is used as an error measure, rather than absolute differences between control markers and their corresponding speech markers, [(C_i , S_i) , (C_{i+1} , S_{i+1})…], since processing can introduce some delays between the two signals (control and speech).

This two step process is performed for each pair of consecutive control markers or in a sense for each pitch period. The error T_D in effect reflects the closeness of the pitch period between the speech markers S_{i+1}, S_i, to the true pitch period corresponding to control markers C_{i+1} , C_i.

4.4.2 Error measures

Although an error estimate is already available in the form of error sequence T_D, on its own it is not very helpful as an analysis tool because of its huge size (a large signal will have thousands of periods and hence T_D will be a very large sequence representing errors in each period) and, therefore, there is a need to present the results in a simple and readable form. This section describes

the error measures used to quantify the accuracy of speech markers in a very compact form.

The error measures used to evaluate accuracy of speech markers are as follows:

a. Avg_err: This measure gives an average of the error values of the sequence T_D, in sample points.

$$\text{Avg\_err} = \frac{1}{N}\sum_{i=1}^{N} T_D(i),$$

Where,

T_D is the error sequence representing error in each of the pitch periods,

and, N is the total number of pitch periods in the control signal.

b. Abs_avg_err: This measure gives an average of the absolute error values of the sequence T_D, in sample points.

$$\text{Abs\_avg\_err} = \frac{1}{N}\sum_{i=1}^{N} |T_D(i)|,$$

Where,

T_D is the error sequence representing error in each of the pitch periods,

and, N is the total number of pitch periods in the control signal.

c. One_ms_Accuracy: The error values in T_D which were found to be less than or equal to 1 ms / (20 sample points at 20 kHz sampling rate) expressed as a percentage of the total number of pitch periods in the control signal.

$$\text{One\_ms\_Accuracy} = \frac{1}{N}\sum_{i=1}^{N} O\_m\_A(i),$$

$$\text{And, } O\_m\_A(i) = \begin{cases} 1 & \text{if } |T_D(i)| \le 1\text{ms, else } \mathbf{0}. \end{cases}$$

Where,

T_D is the error sequence representing error in each of the pitch periods,

and, N is the total number of pitch periods in the control signal.

d. Pt3_ms_Accuracy: This measure gives the percentage of the total number of pitch periods in control signal for which the Abs_avg_small_err is ≤ 0.3ms. Note that the "threshold" of 6 sample points is arbitrary, but was used as a measure of very small errors.

$$\text{Pt3\_ms\_Accuracy} = \frac{1}{N}\sum_{i=1}^{N} P3\_m\_A(i),$$

$$\text{And, } P3\_m\_A(i) = \begin{cases} 1 & \text{if } |T_D(i)| \le 0.3\text{ms, else } \mathbf{0}. \end{cases}$$

Where,

T_D is the error sequence representing error in each of the pitch periods,

and, N is the total number of pitch periods in the control signal.

e. Extra_speech_markers: This measure gives a count of the extra-"unwanted" speech markers that were identified by the algorithm. That is, these are the markers in the speech signal for which there appears to be no corresponding marker in the control signal.

f. Esm: This measure is same as error measure Extra_speech_markers but expressed as a percentage of the total number of pitch periods in the control signal.

The error measures can be divided into two groups. One group gives the performance in terms of errors, error measures a, b, e and f, thus should have values as low as possible, whereas the other group gives performance in terms of accuracy, error measures c and d, and thus should have values as high as possible. Ideally error measures a, b, e and f should have values close to zero and error measures c and d should have values close to 100.

At this juncture it is worth mentioning that a mock test was conducted using the control signal, acting as speech as well as reference signal at the same time, to determine the reliability of the error estimation process. As expected the values of error measures a,

b, e and f were either zero or negligibly small. Also error measures c and d were approximately 100%. This demonstrated the reliability of the error estimation process.

4.5 Experiments

In general the experiments used the following settings of control parameters while estimating speech markers, unless mentioned otherwise:

a) Block Size: 5*pp long

b) Frame Size: 2*pp.

c) Moving Window Size: 1*pp long with peak at the center of window.

d) Local to Transition Cost Weight Ratio: 0.75.

e) Pitch Track: ODU (YAPT) pitch track.

Most of the parameter values used for estimating control markers are the same as those listed above except for the following:

a) Moving Window Size: 1*pp long for Major peaks and 0.5*pp long for Minor peaks.

b) Pitch Track: Reference pitch track.

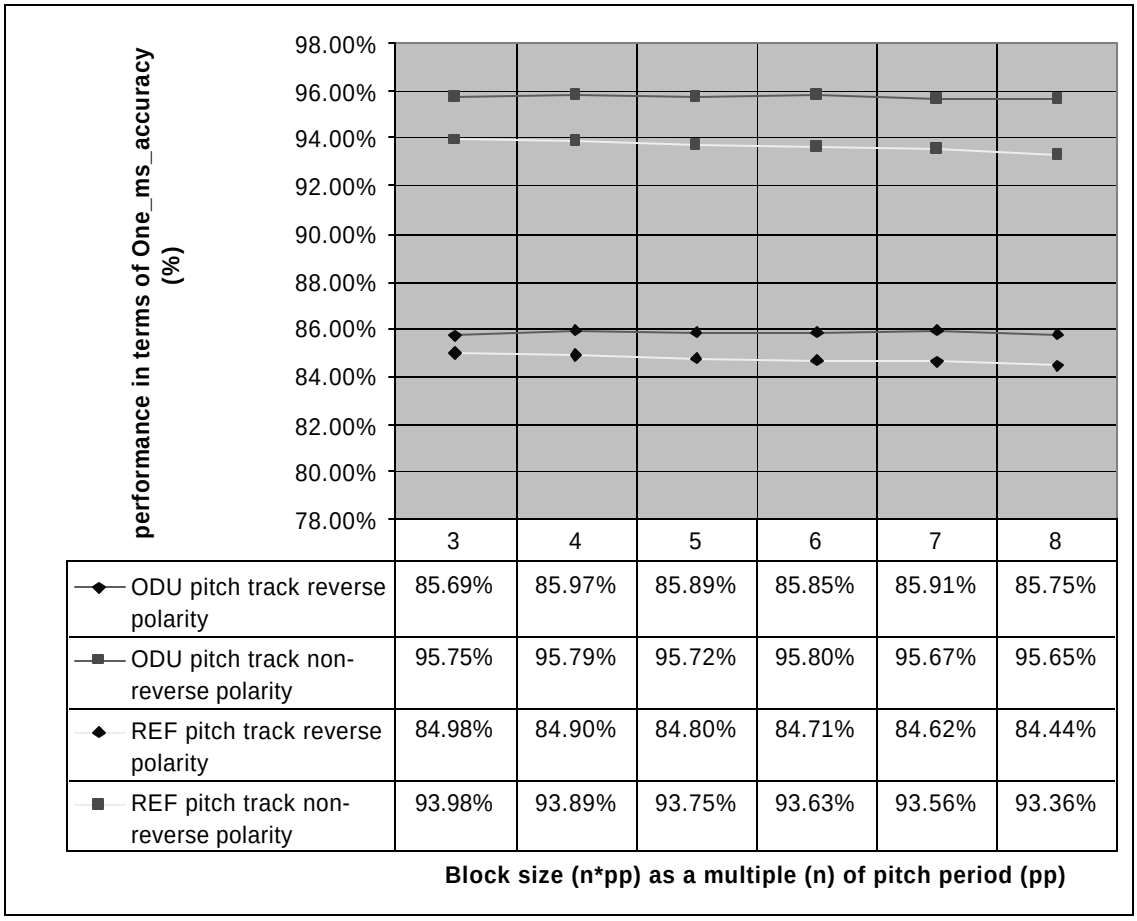
4.5.1 Experiment – I: Effect of signal polarity

This test determines the effect signal polarity (discussed in section 3.3.1 of chapter III under Peak Identification Process) has on the accuracy of speech markers, when the block size in speech marker algorithm is varied in size from 3*pp-8*pp. The experiments were conducted using both the reference pitch track and the ODU (YAPT) pitch track. The results obtained are shown in Fig. 22.

As observed from Fig. 22, there is a considerable drop in performance when a signal with reverse polarity (indicated by tilted rectangle points) is used for estimating speech pitch markers. It was also observed that the algorithm performs better when using the YAPT pitch track. Another observable fact from the figure is that the use of blocks of

different sizes has no perceptible difference in performance across each of the 4 different subtests shown in Fig. 22. This experiment shows that use of the "incorrect polarity speech signal (when the speech signal has prominent valleys) as opposed to the "correct" polarity speech signal (speech signal that has prominent peaks) plays a critical role in the accuracy of the speech markers.

The process of changing polarity of speech signals, so that signals always have prominent peaks (maximas) requires the use of the polarity checking algorithm (section 3.3.1 chapter III) to identify the polarity and correct as needed.

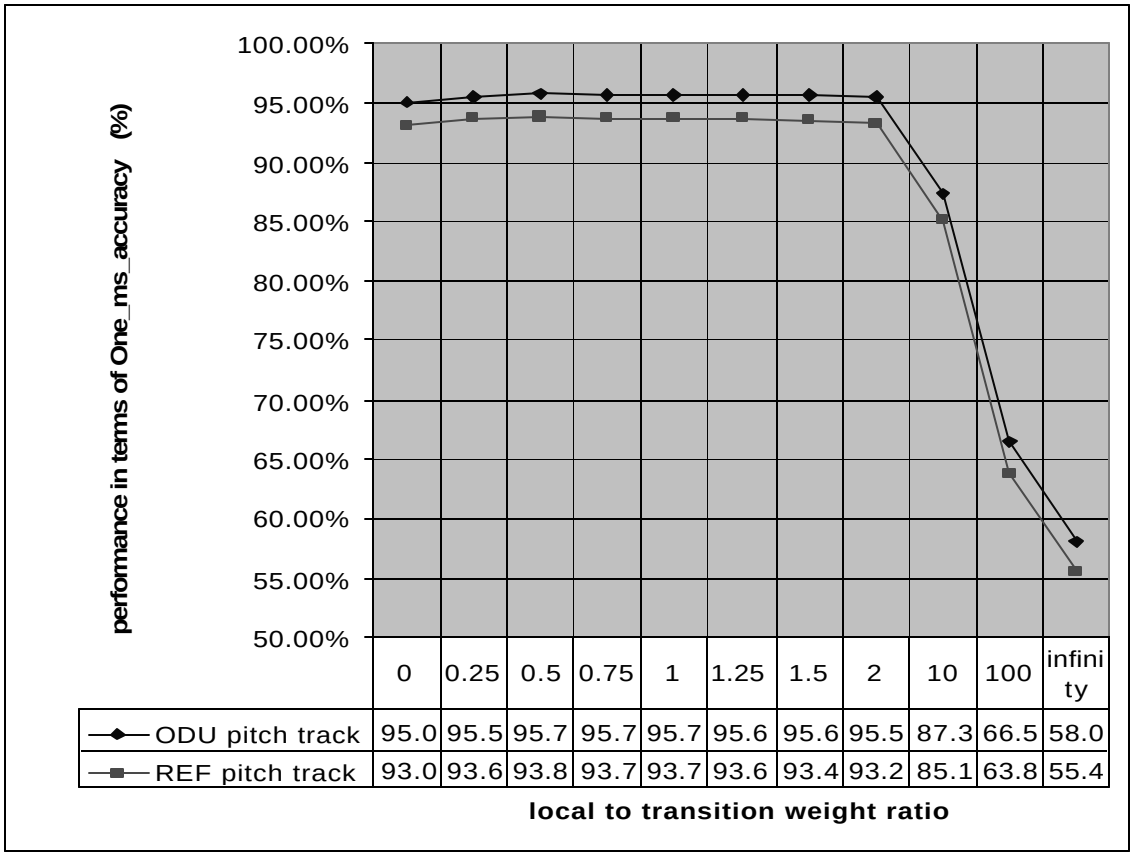


	3	4	5	6	7	8
ODU pitch track reverse polarity	85.69%	85.97%	85.89%	85.85%	85.91%	85.75%
ODU pitch track non-reverse polarity	95.75%	95.79%	95.72%	95.80%	95.67%	95.65%
REF pitch track reverse polarity	84.98%	84.90%	84.80%	84.71%	84.62%	84.44%
REF pitch track non-reverse polarity	93.98%	93.89%	93.75%	93.63%	93.56%	93.36%

Fig. 22. Effect of signal polarity on the accuracy of the speech markers. Results were obtained using blocks of different sizes (3*pp-8*pp) and different pitch tracks.

4.5.2 Experiment –II: effect of local to transition cost (LTC) weight ratio

This experiment shows the effect of the ratio of local to transition cost weights used in the DP has on the overall accuracy of speech markers. Note that a low ratio of LTC implies that the estimated pitch track has a large influence on the selection of markers, and a high ratio of LTC implies that the estimated pitch track is determined mainly from the peak picking.



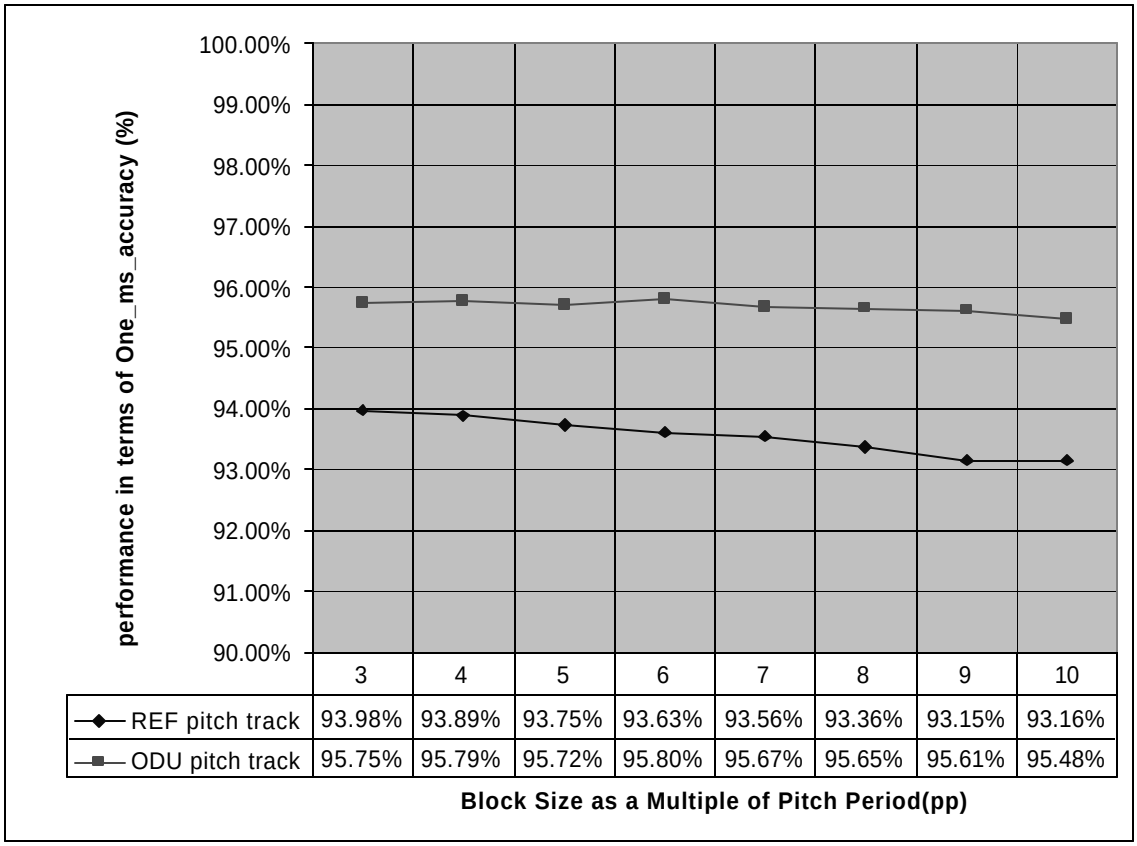
	0	0.25	0.5	0.75	1	1.25	1.5	2	10	100	infinity
ODU pitch track	95.0	95.5	95.7	95.7	95.7	95.6	95.6	95.5	87.3	66.5	58.0
REF pitch track	93.0	93.6	93.8	93.7	93.7	93.6	93.4	93.2	85.1	63.8	55.4

Fig. 23. Effect of local to transition cost weight ratio while using reference and ODU pitch track.

In the extreme, an LTC of 0 implies the peak heights are not used at all in the pitch marker, and an LTC of infinity means that the estimated pitch values are not used at all in the pitch marker. Results are shown for the one ms accuracy as a function of LTC. As shown in Fig. 23, ratios of LTC up to 2 have no major affect on the one ms accuracy, which remains stable around 95% when using the ODU pitch track and 93%

when using the supplied Keele Reference pitch track. The performance however decreases rapidly beyond a weight ratio of 2, indicating that transition cost (closeness of distance between markers to estimated pitch period) has a significant influence on the performance.

4.5.3 Experiment –III: effect of block size



	3	4	5	6	7	8	9	10
REF pitch track	93.98%	93.89%	93.75%	93.63%	93.56%	93.36%	93.15%	93.16%
ODU pitch track	95.75%	95.79%	95.72%	95.80%	95.67%	95.65%	95.61%	95.48%

Fig. 24. Performance as a function of block size while using reference and ODU pitch track.

The experiments reported in this section reconfirm some of the previously observed results of experiments I and II, as well as provide some additional information about the operation of the marking algorithm. In these experiments, performance is tested as a function of block size for the supplied reference pitch track and the ODU-computed pitch track. Results obtained are shown in Fig. 24. As can be observed, the block sizes have very little effect on performance and also the algorithm works better when using the ODU computed pitch track (YAPT) as compared to the supplied reference track.

4.5.4 Experiment –IV: robustness of algorithm to errors in pitch track

This experiment was conducted to determine the robustness of the algorithm to errors in the pitch track used for estimating speech markers. This experiment uses the reference pitch track instead of the ODU pitch track for estimating speech markers as, presumably, this supplied track is a better standard. The tests were conducted using different block sizes (3*pp-7*pp).

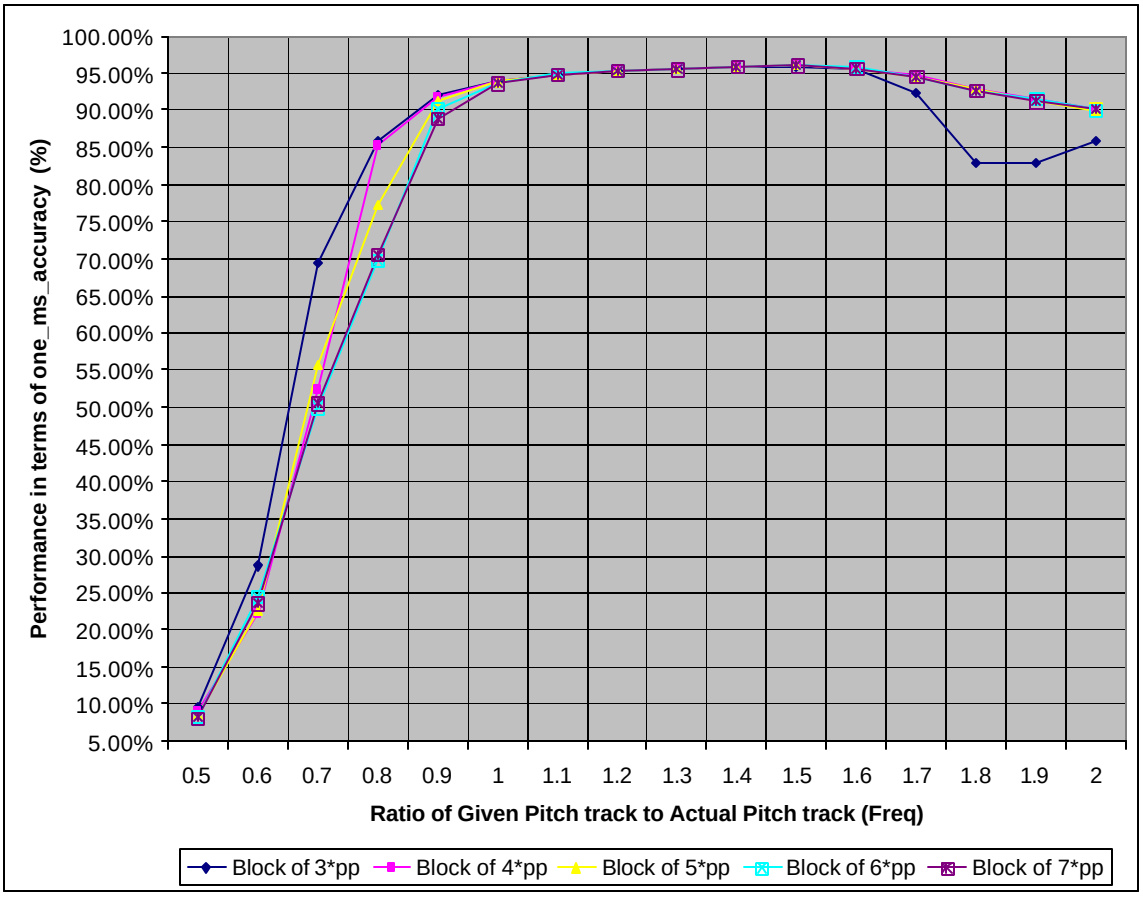


Fig. 25. Performance of algorithm as a function of error in pitch track for different blocks of size.

As observed from the Fig. 24, blocks of sizes 4*pp-7*pp all have the same performance throughout the error range. The best performance is seen when the error in the pitch track, expressed as a ratio of the pitch track used for estimating speech markers and the reference pitch track, varies from 0.9-2. Beyond this range the performance

decreases. Also observed from Fig. 25, a relatively better performance is obtained for a block size of 3*pp as compared to blocks of other sizes, for an error range of 0.5-0.8. In summary, the algorithm is found to have a pitch-tracking error tolerance range of -10% to +100% (in frequency domain) or -50% to +11.1% (in time domain or pitch period).

It should be noted that this experiment shows the performance of algorithm when there is uniform error in pitch track used for estimating speech markers. This experiment does not reflect performance when there is non-uniform error in the pitch track or when the error in the pitch track is because of the sudden variations in the pitch period.

4.5.5 Experiment –V: effect of moving window size

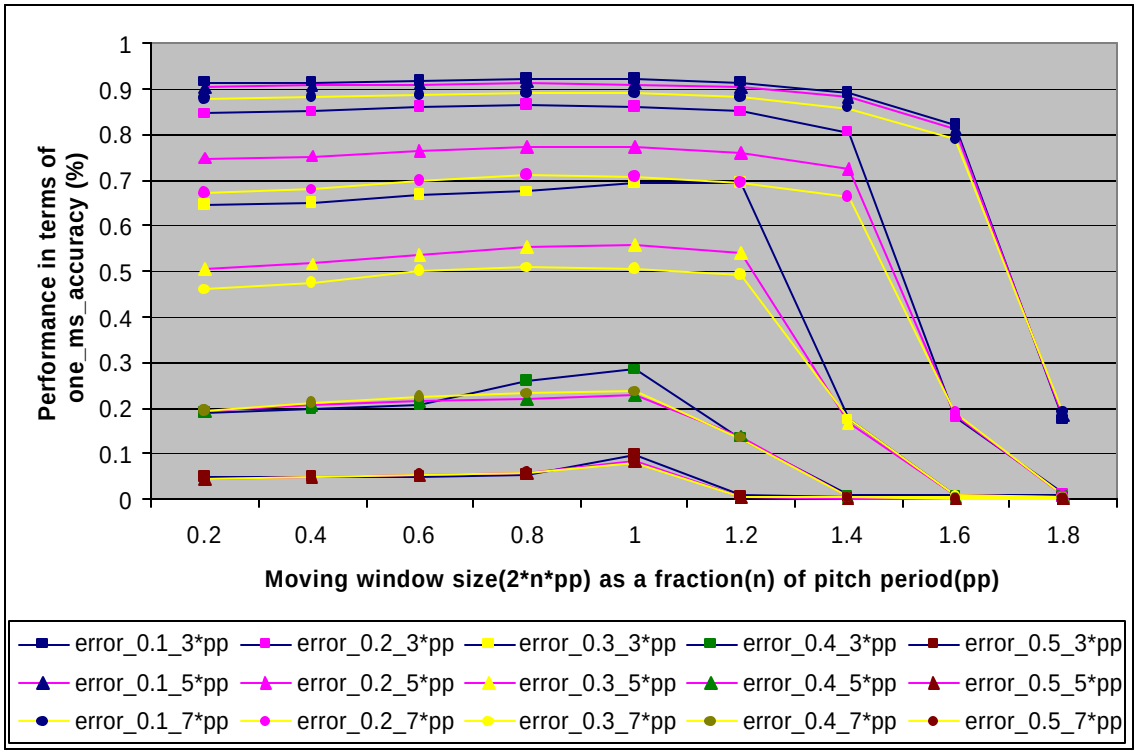


Fig. 26. Effect of different Moving Window sizes on speech marker accuracy. Results were obtained using erroneous pitch track values and different block sizes.

Another experiment was conducted to study the affect of size of the Moving Window (used in peak picking, section 3.3.3, chapter III). This was studied with respect to errors in the pitch track and block size. The size of the moving window (used in peak

picking) was varied from 0.2*pp to 1.8*pp in steps of 0.2*pp. The results obtained are as shown in Fig. 26. The notation error_x_y*pp used in Fig. 26 implies speech markers were computed using pitch track with x% error in pitch track and blocks of size y*pp .

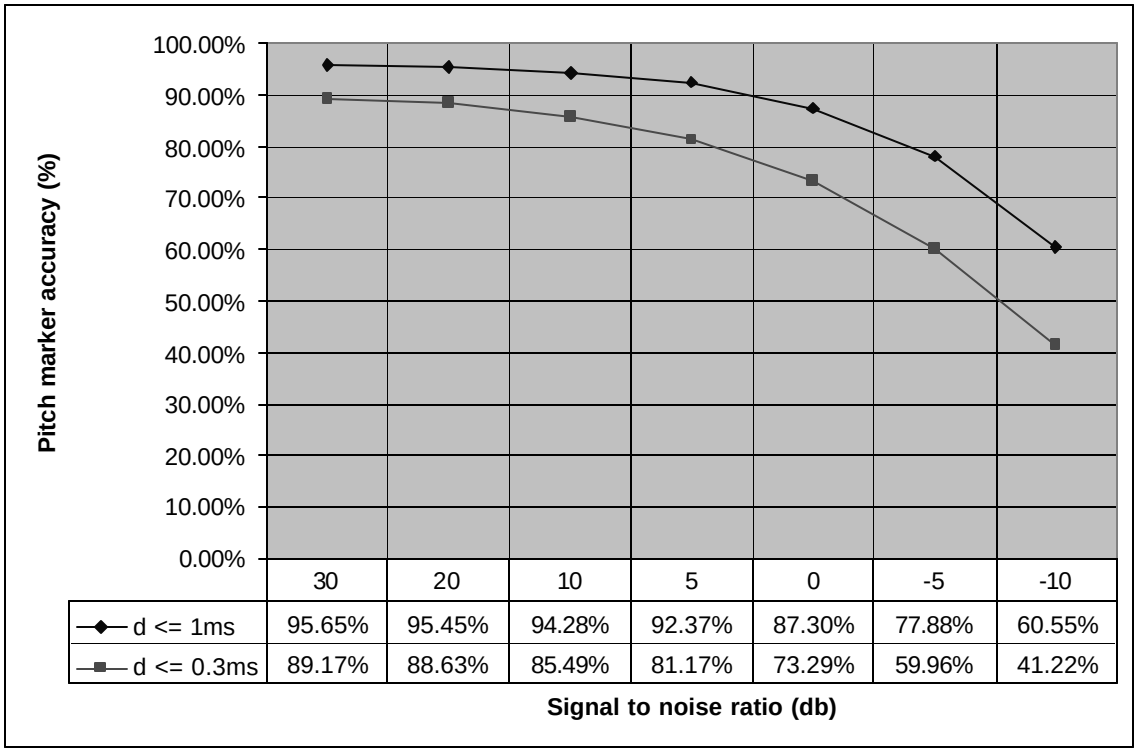
The error in the pitch track was varied in the range of -50% to -10% and the block size was varied from 3*pp to 5*pp to 7*pp. This results in a total of 15 different combinations of error and block size. On studying each combination for the effect of different window sizes, it can be concluded that in general with increasing window size the performance decreases sharply after a certain breaking point. The performance decreases in the window size range of 1.2*pp-1.8*pp depending on the error in the pitch track; the greater the error in the pitch track the lower the window size value when the performance begins to decrease. Although the performance decreases beyond a certain threshold point of window size, it has no affect on performance for most of the remaining range (generally, 0.2*pp-1.4*pp). Another pattern that can be observed is that as the error in pitch track increases, performance decreases and block size has more or less no effect on performance.

4.5.6 Experiment –VI: robustness of algorithm to noise

This experiment was conducted to determine robustness of algorithm to noise in speech signal. White Gaussian noise of various levels was added to the speech signal to test the robustness to different SNRs. The ODU (YAPT) generated pitch track was used for computing pitch markers in noisy speech signals. It should be noted that pitch track was computed from the noisy speech signal every time noise was induced in the signal and then applied for computing pitch markers in the noisy speech signal. The experiment was conducted using blocks of size 5*pp. The results obtained are as shown in Fig. 27.

The Fig. 27 shows a plot of performance of the algorithm in terms of the error measures One_ms_Accuracy (deviation‘d’ = 1ms) and Pt3_ms_Accuracy (deviation‘d’ = 0.3ms) for different SNRs (30db to -10db). As can be observed from the graph the algorithm shows an accuracy rate of 92% and above for the SNR range of 30db to 5 db in terms of the error measure One_ms_Accuracy. And when considering deviation of the

pitch markers to be less than 0.3ms or error measure Pt3_ms_Accuracy, the performance ranges from 89% to 81% for the SNR range of 30db-5db respectively. There is a large drop in performance for speech signals with SNRs < 5db.



	30	20	10	5	0	-5	-10
d <= 1ms	95.65%	95.45%	94.28%	92.37%	87.30%	77.88%	60.55%
d <= 0.3ms	89.17%	88.63%	85.49%	81.17%	73.29%	59.96%	41.22%

Fig. 27. Performance vis-à-vis noisy speech. The accuracy is shown in terms of two error measures – One_ms_Accuracy and Pt3_ms_Accuracy.

Another test conducted using pitch track obtained from noise free signals and used to find pitch markers in signals corrupted by noise gave near identical results to values shown in Fig. 27.

4.6 Summary

In this chapter we discussed algorithms to generate control markers from the control signal (Control Marker Algorithm) and then to compare the control and speech markers (Error Estimation Process). Also we discussed various error measures used in

error analysis. And lastly, experiments and their analysis were discussed. The algorithm was found to perform well even when there were large errors in pitch track. It was found that the moving window size (window used in peak picking) and block size had no measurable effect on the performance of the algorithm. The algorithm showed sensitivity to signal polarity of the speech signal. It was found that signals which had relatively large maximas compared to minimas showed improved performance. Another important result observed was robustness of algorithm in identifying pitch markers to noise in speech signal.

In the next chapter we conclude with a summary and suggestions for future work.

CHAPTER V

SUMMARY AND FUTURE WORK

5.1 Summary

The research in this thesis was performed with the aim of developing a pitch marking algorithm, which can be used in improving the speech recognition rate for automatic speech recognition systems. The algorithm developed identifies pitch cycle markers which in turn are used to identify the individual pitch cycles. The algorithm uses dynamic programming to combine two sources of information for the pitch marker. One source of information used is the "local" information corresponding to the location and amplitude of peaks in the acoustic speech signal. The other source of information used is the "transition" information corresponding to the relative closeness of the distance between the peaks to the expected pitch period values. The expected pitch period values are obtained from a pitch tracker (YAPT). The Keele speech database was used for testing purposes.

Several experiments were performed to study the effect of parameters like block size, frame size, signal polarity and moving window size on the performance using clean studio quality signals. It was found that none of the parameters except signal polarity affected the performance of the algorithm, at least for a broad range of parameter values. An accuracy rate of approximately 95% (error measure: One_ms_Accuracy) was observed for those experiments. The use of signals with reverse signal polarity (more prominent minima than maxima) was observed to cause a drop in performance by 10% from 95% to 85% (error measure: One_ms_Accuracy), indicating the importance of choosing the correct signal polarity (more prominent maxima than minima). In another experiment with noisy speech signals, an accuracy rate of 92% and above (error measure: One_ms_Accuracy) was observed for the SNR range of 30db-5db. The algorithm was also tested for its robustness vis-à-vis errors in the pitch track using clean studio quality signals and was found to be highly robust with a pitch track error range of -10% to

+60%. The algorithm generated = 1% extra markers (false positives) for clean studio quality (pitch track error range of -10% to +60%) and noisy speech signals (SNR range of 30db to 5db). A preliminary test on telephone quality signal gave an accuracy rate of 63%. The use of the ODU pitch tracker YAPT generated reference pitch track for identifying pitch markers gave an accuracy rate of 95% as compared to 93% obtained using reference pitch track from the Keele database. This better performance of pitch marking using the ODU pitch tracker versus the supplied reference pitch track was not expected, and remains as one of the surprising results of this work.

5.2 Future work

Although the algorithm showed very high performance, it is still far from perfect. There are a number of areas that have a great scope for future work.

1. Even though the algorithm performed very well with studio quality and noisy signals, it showed very poor performance with telephone quality speech. So, further work needs to be done to improve the accuracy rate with telephone quality signal.
2. A further improved algorithm needs to be developed that can tolerate both pitch doubling and pitch halving error types. These two types of errors are common in pitch tracking. The algorithm presented in this thesis works in a much narrower error range.
3. Pitch marking could be evaluated in applications like pitch synchronous analysis, spectral analysis et al.

REFERENCES

[1] R. W. Schafer and J. D. Markel, "Speech Analysis," IEEE PRESS, New York, 1979.

[2] D. B. Fry, "The Physics of Speech," Cambridge University Press – Cambridge Textbooks in Linguistics, Cambridge, 1979.

[3] G. J. Borden and K. S. Harris, "Speech Science Primer: Physiology, Acoustics, and Perception of Speech," Williams and Wilkins, Baltimore, 1980.

[4] I. R. Titze, "Principles of Voice Production," Prentice Hall, New Jersey, 1994.

[5] S. Harbeck, A. Kießling, R. Kompe, H. Niemann and E Nöth, "Robust pitch period detection using dynamic programming with an ANN cost function," Proc. EUROSPEECH, Madrid, vol. 2, pp. 1337-1340, September 1995.

[6] V. Colotte and Y. Laprie, "Higher precision pitch marking for TD-PSOLA," XI European Signal Processing Conference (EUSIPCO), Toulouse, 2002.

[7] Y. Laprie and V. Colotte, "Automatic pitch marking for speech transformations via TD-PSOLA," European Signal Processing Conference (EUSIPCO), Rhodes, 1998.

[8] E. Moulines and F. Charpentier, "Pitch-Synchronous Waveform Processing Techniques for Text-To-Speech Synthesis Using Diphones," Speech Communication, vol. 9, pp. 453-467, 1990.

[9] http://www.haskins.yale.edu/haskins/HEADS/MMSP/acoustic.html.

[10] K. Kasi, "Yet another algorithm for pitch tracking," Masters Thesis, Old Dominion University, Norfolk, VA, 2002.

[11] W. Hess, "Pitch Determination of Speech Signals," Springer Series in Information Sciences, volume 3, 1983.

[12] V. Goncharoff and P. Gries, “An Algorithm for Accurately Marking Pitch Pulses in Speech Signals,” Proc. IASTED’98 International Conference on Signal and Image Processing, Las Vegas, NV, October 1998.

[13] R. Veldhuis, “Consistent pitch marking,” International conference on Speech language Processing, vol.3, pp. 207-210, 2000.

[14] D. Talkin, “A Robust Algorithm For Pitch Tracking”, in Speech Coding and Synthesis, Elsevier, Amsterdam, pp. 495-518, 1995.

[15] K. Kasi and S.A. Zahorian, “Yet another algorithm for pitch tracking,” International Conference on Acoustics, Speech and Signal Processing, 2002.

[16] D. K. Smith, “Dynamic Programming: a practical introduction,” Ellis Horwood Series in Mathematics and Its Applications, London, 1991.

[17] J. G. Proakis, D. G. Manolakis, “Digital Signal Processing 3ed,” Prentice Hall, New Jersey, pp. 614-630, October 1995.