SPEECH FUNDAMENTAL PERIOD ESTIMATION USING PATTERN CLASSIFICATION

IAN HOWARD

5/06/1991

V2.0

POST-VIVA EDITION

8/2/92

DEPARTMENT OF PHONETICS AND LINGUISTICS
UNIVERSITY COLLEGE LONDON

A thesis submitted for the degree of Doctor of Philosophy in the University of London

ABSTRACT

The problem investigated concerns the robust estimation of fundamental period, only on the basis of representative speech pressure waveforms. The work has involved the design and development of a set of algorithms. The main intended application is in pattern processing acoustic and cochlear implant hearing aids. Essentially the task is to infer from the acoustic evidence available the points in time at which vocal fold closures occur. Its accomplishments both gives fundamental period information on a cycle-by-cycle basis and provides information concerning whether voicing is present.

The task of detecting the point of closure of the vocal folds is formulated as a pattern recognition problem, and the pattern recognition technique employed uses the multi-layer perceptron (MLP). The first system configurations investigated were based on a pre-processing of the speech pressure waveform by a wide-band filterbank analyzer. This gave an input to the classifier which consisted of a set of adjacent time frames from the output of the filterbank. The output from the classifier was defined as being in one of two classes. In the first there is a period epoch marker at a given output frame, in the second there is not. This first classifier was trained to generate an output which signified the presence of a vocal fold closure at the centre of its input window. The fundamental periods between successive vocal-fold closures defined by these epoch markers, are given the name Tx. The labelling of both training and test data was performed semi-automatically by means of an algorithm that makes use of the output of a laryngograph.

Developments of this first approach were then explored. These were primarily directed towards methods for reducing the training time for the MLP and improving the time resolution of the fundamental period estimates. Different pre-processing stages were investigated and these included direct operation on the speech pressure waveform and the use of a simplified auditory filterbank. Methods to reduce the computation load required for practical implementation were examined and these resulted in a system using a low-order filterbank together with a smaller MLP network. The last configuration was of practical interest because it had a processing load small enough to

be run in real-time on a portable DSP system. A real-time system was implemented in conjunction with Mr. John Walliker. First patient results using this system are reported following perceptual assessments made by Dr. Andrew Faulkner.

A number of objective assessment techniques were developed and used to permit quantitative comparisons between fundamental period estimation algorithms to be carried out. These involved both quantitative comparisons between frequency contours and between time excitation epoch markers. Using these comparisons, various different configurations of the MLP-Tx algorithm were evaluated over a wide range of speakers and environmental conditions. The performance of the MLP-Tx algorithm was also compared against that of established fundamental frequency estimation algorithms, and its performance in competing noise was found to be better than that obtainable by the use of the peak-picking approach previously employed.

ACKNOWLEDGEMENTS

The author would like to express his gratitude to Professor Adrian Fourcin for providing the opportunity to work in his department and for suggesting a suitable area in which to study for the degree of Doctor of Philosophy.

The speech filing system used for speech data handling (SFS) was written by Dr. Mark Huckvale, in conjunction with whom the pattern processing system (PPS) was developed. Many thanks are also due to Dr. Mark Huckvale for reading and commenting on the chapter on pattern recognition techniques.

The peak-picker algorithm used for some of the comparisons was written by Dr. David Howard and Andy Eaton.

Sarah Palmer helped with some of the recordings.

Remi Brun constructed the constant distance microphone mounting device, and also helped with some of the recordings.

Julian Daley and John Walliker built the preamplifier and high pass filter board for the Knowles microphone used for some of this work.

The real-time hardware implementation of the MLP-Tx algorithm on a TMS320C25 digital signal processor was carried out in conjunction with John Walliker. The author was responsible for training the real-time system using a software simulation and the assembly language program and hardware for the TMS320C25 were both developed by John Walliker.

Peter Bocherby provided some of the computational facilities used.

The author would also especially like to thank Dr. Andrew Faulkner and Dr. Stuart Rosen for reading and commenting on the draft manuscript.

The author would also like to thank his other colleagues and the staff in the Phonetic and Linguistics department for their help, and the many speakers who gave up their time to be recorded for the training, evaluation and testing databases.

Finally the author would like to thank Professor Eric Ash for his initial help in providing the chance to study for a PhD at University College London.

This work was initially supported by the Medical Research Council of Great Britain, then carried out by the author in his own time and finally with support from the Laryngograph Trust.

TABLE OF SYMBOLS

A/D	Analogue to digital
ANN	Artificial neural network
BW	Bandwidth
C_i	Covariance matrix
exp	Natural exponent
dB	Decibel
$\delta y/\delta x$	Representation of a partial differential
D_i	Decision boundary i
DL	Difference limen
Dx	Frequency histogram of Fx values
E	Total error in MLP output
E_p	Error in MLP output, for pattern p
EPI	External Pattern Input group at UCL
ERB	Equivalent rectangular bandwidth
FT	Fourier transform
F_j	Squashing function in MLP
Fx	Larynx fundamental frequency
Hz	Frequency in Hertz
I_{pi}	Input pattern p for node i in MLP
L_{ij}	Loss matrix for Bayesian classifier
Ln	Natural logarithm
Lx	Output waveform from laryngograph
M	Number of pattern classes
M_i	Mean vector
MLP	Multi-layer perceptron
MLP-Tx	Name of MLP algorithm for fundamental period estimation
ms	Milliseconds
NET_{pj}	Weighted sum of inputs at node i for pattern p
O_{pj}	Output pattern p at node j

π	3.141592654, the natural constant
PPS	Pattern processing system, used at UCL
$p(W_i\|X)$	Conditional probability of pattern class W_i given data vector is **X**
$p(X\|W_i)$	Conditional probability of data vector **X** given pattern class W_i.
R_j	Conditional average risk in Bayesian classifier
RNID	Royal National Institute for the Deaf
ROC	Receiver Operating Characteristic
SFS	Speech Filing System, used at UCL
SNR	Signal to noise ratio
T_{pi}	Target pattern p for output node i in MLP
Tx	Larynx fundamental period value
UCL	University College London
VOT	Voicing onset time
W_i	Pattern class i
W_{ij}	Weight between nodes i and j in MLP
X	Data vector
α	Momentum term for MLP
σ_{pj}	Error term for MLP
Γ	Learning rate constant for MLP
Δ	Representing a small change

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1: THE AIMS OF THE WORK

1.1 AIMS OF THE WORK

1.1.1 Speech fundamental period estimation

This work is concerned with the design and development of an algorithm (MLP-Tx) that can perform speech fundamental period estimation. The algorithm has been specifically developed to perform fundamental period estimation for a signal processing hearing aid designed by the EPI group at UCL which seeks to provide from the acoustic speech signal an output which corresponds to that from the laryngograph (Walliker et al., 1986; Rosen et al., 1987).

The novel aspect of the work stems from the fact that the task is formulated as a pattern recognition problem and the MLP-Tx algorithm, based on a multi-layer perceptron pattern classifier, was trained-by-example to perform the required task. The data that was used to train the algorithm was composed of speech that was semi-automatically labelled for fundamental period location using a pair of algorithms that made use of the output of a laryngograph, which was recorded simultaneously with the speech. The fundamental periods were delineated in terms of the closure of the vocal folds as a function of time, as defined by the location of the maximum positive differential in the output of the laryngograph.

One of the strengths of the MLP-Tx algorithm is that the fundamental period estimates are made on a cycle-by-cycle basis. Consequently irregularities in vocal fold vibration can be detected by the algorithm, whereas many other algorithms would tend to smooth the period values. Creaky voice can be dealt with effectively using the MLP-Tx algorithm, whereas many other algorithms treat this important larynx excitation as being unvoiced due to its intrinsic irregularity.

Another strength of the MLP-Tx algorithm is that it operates robustly in the presence of noise.

The MLP-Tx algorithm is also suitable for real-time implementation because of the simple uniform structure of the MLP, and the inherently small (about 10ms) input to output delay. This small delay is important to prevent a lack of synchronization between the speech signal and gestures and lip movements made by a speaker. A practical real-time implementation for use in hearing aids for the profoundly hearing impaired has been carried out in conjunction with John Walliker (Howard & Walliker 1989; Walliker & Howard, 1990).

There are many other applications for fundamental period estimation algorithms, some of which are discussed below, since this an important component in the description of vocal fold vibration. There is a genuine need for algorithms that are robust and will operate in real environmental conditions.

It is important to be able to assess the performance of a fundamental period estimation algorithm, so that improvements can be evaluated and so that its performance with respect to other techniques can be gauged. To this end some work on speech fundamental period estimation algorithm comparison techniques was also carried out.

1.1.2 Applications of speech fundamental period estimation

Cochlear implants

One application for algorithms that can perform speech fundamental period estimation is in signal processing hearing aids (Fourcin et al., 1983). Such hearing aids are of value to the profoundly deaf. They operate by extracting basic elements of speech and presenting this reduced representation in a suitable format to the patient either acoustically or directly onto the cochlea by electrical stimulation. It has been shown that presenting voice fundamental period feature alone can be more beneficial to some patients than presenting an amplified version of the whole speech signal (Rosen et al., 1987). The main reason for this is that the auditory systems of this class of patients has a very restricted channel capacity which is less than that required to encode adequately the whole speech signal. Consequently presentation of the whole speech signal maybe

confusing and even painful, whereas a basic fundamental period (frequency) signal can be made much easier for the patient to interpret usefully.

In the case of single channel cochlear implants, the information from the pattern element extractor sent into the hearing system of the patient must all be present in the one signal (unlike in multi-channel implants). This is the approach adopted by the EPI group at UCL (Walliker et al., 1986). The elements used by the EPI group are chosen to be simplified aspects of speech that, when suitably coded, are matched to the residual discriminative abilities of the patients (Fourcin, 1979; Faulkner, Ball & Fourcin, 1990; Walliker et al., 1985).

Speech fundamental frequency is a particularly useful feature to present to the profoundly deaf because it is almost completely invisible to the lip-reader and consequently it provides an aid to lipreading and the development of voiced speech production. There is particular benefit to be derived from the use of an algorithm that can perform fundamental period estimation on a cycle-by-cycle basis for such signal processing hearing aid applications. This is because preservation of the irregularity present in the original speech excitation is beneficial because the patient can then hear creaky and other irregular voice characteristics (Abberton et al., 1985). For these applications the algorithm must run in real-time with a processing delay between input and output should be as small as possible, and no more that a maximum of 40ms, or the signal looses its usefulness (McGrath & Summerfield ,1985). The MLP-Tx algorithm is therefore particularly suitable for such applications, because it possesses both of these qualities.

Speech Coding

The transmission and storage of speech by electronic (and optical) means is very widely carried out in modern society. The cost of the transmission and storage of a signal is clearly dependent on the data-rate of the signal. For example the greater the data-rate required to specify a speech signal, the fewer the speech channels that can be carried down one telephone transmission cable. Consequently there is great interest in

techniques in speech processing that will reduce the data-rate necessary for communication by speech, since it reduces costs.

Much work has been carried out in the field of speech coding with the goal of reducing the data-rate needed for speech transmission. For example especially important contributions have been made by Dudley (1939), Gabor, (1947), Lawrence, (1953), Schroeder (1966), and Gold & Rader, (1967). In essence, such schemes work by taking advantage of the observation that there is a correlation between adjacent time samples of the speech signal. Consequently it is not necessary to transmit each original data sample, as would be the case if all the data samples were un-correlated.

Many coding schemes that aim to reduce the data-rate in the transmission of speech assume a source-filter model of speech production and require the determination of speech fundamental frequency (Willis, 1829; Fant, 1960). Such schemes can be considered as speech analysis/synthesis systems, and include Gabor's system, Dudley's vocoder, the channel vocoder (Flanagan, 1972) and copy synthesis (Lawrence, 1953).

Speech and Speaker Recognition

Another application of information relating to speech excitation is in automatic speech recognition. The incorporation of such information into the recognition process has been demonstrated as beneficial to the recognition task (Atal, 1974; Rosenberg & Sambur, 1975). In addition, fundamental period information has been shown to be useful in speaker recognition, by both man and machine (Atal, 1972; Abberton, 1974; Abberton, 1976).

Glottal-synchronous speech analysis

The individual identification of speech fundamental period is also useful in providing a means to carry out glottal-synchronous analysis of speech. The idea behind this technique is that when, for example, performing a short-time spectral analysis of the speech, the window for the analysis is selected on a period-by-period basis to include

input over only one period, rather than using a fixed window size for the analyses. It has been found that such an approach can give better estimates of the vocal tract transfer function than using fixed window analysis (Hunt & Harvenberg, 1986; Pearce & Whitaker, 1986; Hunt, Zwierzynski & Carr, 1989). To carry out this task requires the identification of the vocal fold closure points, and this is on of the tasks performed by the MLP-Tx algorithm.

1.2 ORGANIZATION OF THE THESIS

The body of the thesis is organized in the following manner:

Chapter 2 provides a brief discussion of acoustic, articulatory and phonetic descriptions of the speech signal. Properties of the voice source are then examined, with particular reference to their correlates with the output of a laryngograph.

Chapter 3 explores some of the issues involved in speech fundamental frequency and period estimation. The basic requirements for algorithms that are to perform such a task are discussed and these include the required frequency resolution for a given application and the benefits of average or cycle-by-cycle estimates. A description of the different approaches used in fundamental frequency estimation is then given.

Chapter 4 gives a more detailed discussion of the operation of some established techniques used for speech fundamental frequency and period estimation. Among those mentioned are the Gold-Rabiner algorithm, the SIFT algorithm, Auto-correlation analysis and the Cepstrum algorithm. Finally, the use of the laryngograph for fundamental period estimation is introduced.

Chapter 5 describes the detailed implementation a fundamental period estimation system using a laryngograph. This involves an automatic analysis of the laryngograph waveform followed by an interactive analysis of the laryngograph and speech waveforms. This was the reference used in this thesis to provide fundamental period excitation epoch markers used both to train and test the MLP-Tx algorithm.

Chapter 6 describes some standard techniques for assessing the performance of speech fundamental frequency estimation algorithms. In addition, some methods newly developed for the present work are discussed and explained. Details of how both established and new sets of comparisons were implemented is then given.

Chapter 7 provides a brief overview of pattern recognition. Some classical approaches to pattern classification are discussed and these include the Nearest neighbour and k-means classifiers, and classification based on the use of likelihood functions, such as the Bayes' classifier. The more recent field of artificial neural networks is then introduced. There is then a more in-depth description of a currently popular connectionist technique, the multi-layer Perceptron (MLP). Methods for reducing training times for the MLP are then described.

Chapter 8 formulates speech fundamental period estimation as a pattern recognition problem. The basic idea is discussed and qualitative results to some initial experiments are given.

Chapter 9 investigates the problem of speech fundamental period estimation using pattern classification in greater depth. Issues concerning input pre-processing and output post-processing are discussed. The requirement for the appropriate databases, to train and test the MLP-Tx algorithm, are examined. These issues are experimentally investigated, and quantitative results are given for different system configurations. A final set of experiments was then carried out on a different set of test data, in which three of the best MLP-Tx algorithm configurations were compared with established fundamental frequency estimation techniques.

Chapter 10 takes a closer look at the operation of some of the MLP networks. The patterns of MLP weights are displayed and the intermediate node activations are given for some of the algorithms during the process of detecting a fundamental period epoch marker. The sources of error in the MLP-Tx algorithm are examined, to give a basis for future improvements.

Chapter 11 considers the problems involved of running the MLP-Tx algorithm in real-time using a digital signal processing chip. Computational complexity is then considered and a reduced computation algorithm is described. The effects of quantizing the weights and using a look-up table for the sigmoid non-linearity are investigated. Perceptual evaluation by Dr. Andrew Faulkner are then described briefly with normal and profoundly deaf listeners of the real-time MLP-Tx algorithm in its intended role in the EPI signal processing hearing aid. The results obtained demonstrate that the real-time MLP-Tx algorithm outperforms the peak-picker algorithm in the presence of pink noise.

Chapter 12 then gives a brief review of the main points emerging from the thesis.

The appendices contain other material that is appropriate to be included for reference purposes, but constitutes extra information that would be rather too much effort to read to warrant putting in the main body of the thesis. There is an analysis of the command line arguments and the computer output generated at each stage of processing, to enable other researchers to use the software developed in the course of this thesis. A description of the pattern processing system used to train and test the MLP classifier in appendix A.1. In appendix A.2, there then follows an analysis of the computer programs that were written for this work, as well as other existing SFS programs, and their subsequent use in the training and testing of the MLP-Tx algorithm. The reading passages, used to train and test the MLP-Tx algorithm, are given in appendices A.3 and A.4. Appendix A.5 shows the questionnaire filled in by all the speakers. Appendix A.6 shows the frequency histograms for the training and two sets of testing data. Appendix A.7 gives a list of the speakers, the passages and the recording conditions used in the testing and training data. Appendix A.8 gives quantitative results from preliminary tests of the MLP-Tx algorithm, in which different configurations of the MLP were investigated.

CHAPTER 2: THE PRODUCTION AND DESCRIPTION OF SPEECH

2.1 SPEECH PRODUCTION

2.1.1 Introduction

This chapter provides a basis for the subsequent discussion of the speech related problems encountered as different stages of the work in this thesis. Firstly there is a general discussion of the origin and nature of the speech signal. There then follows articulatory, phonetic and mathematical descriptions of speech. Finally, voiced speech is discussed together with its relationship to the output from a laryngograph.

2.1.2 The speech signal

Speech provides human beings a means for the transmission of a complex message using sound. It is a signal that is very resistant to interference. Speech may still be intelligible even when the signal is distorted or heavily contaminated with interfering noise, although the quality of the speech will be reduced by such a process.

2.1.3 Origins of speech

The development of spoken language in humans was limited by constraints of evolution (Borden & Harris, 1980). Speech communication must be consistent with the available broadcast facilities (the speech centres in the brain and human vocal apparatus) and decoding system (the human auditory system). The organs of the body used for the production of speech, the vocal organs and respiratory apparatus, were originally evolved to permit breathing of air and the chewing and swallowing of food. However in the course of evolution, they have also been used to provide a means of communication using sound.

The use of speech as a means of communication is only possible because the code for the signal, that is the language system, is known to both speaker and listener. This

system determines the important sound contrasts and prosody.

2.1.4 The hierarchical nature of the speech signal

The hierarchical nature of the speech signal arises from its structured generation process (Borden & Harris, 1980). Some of the stages are illustrated in figure 2.1. Within the human brain, the speech centres contain information concerning the generation of speech. The phonological system used, the grammar and syntax of the language and the vocabulary are all implicitly represented. A possible description of the processes involved in speech production could be as follows: Let us suppose the top of this structure involves a cognitive level of representation where different system activity relates to different "ideas". The first step in speech generation involves a process which effectively arranges ones thoughts into the desired linguistic form and selects appropriate words and phrases to describe one's intended message. In addition these units must then be put into the correct temporal order as required by the grammar of the language. Then consideration to the different sound contrasts necessary for the given language and accent must be made. This could be thought of as corresponding to a phonemic level of processing. The message must next give rise to the signals necessary to control the muscles in the vocal apparatus. Finally, the physical behaviour of air in the vocal apparatus gives rise to an acoustic disturbance that radiates from the lips, and/or nose, carrying the message. The overall result of this coordinated activity is the radiation of sound from the speaker, a small part of which finally reaches the listener. Thus in the speech production process, there is a transformation between a linguistic to a physiological to an acoustic representation of the message. These successive layers form a hierarchically organised structure which can be used as a basis for similarly structured computer-based analysis of speech, as described in the next section.

Reception of the speech sounds in the listener results in processing with a reverse effect. There is a transformation from information in sound, to movement in the eardrum to nerve impulses in the auditory nerve and then finally activity in the higher centres in the brain.

2.1.5 Descriptions of the speech signal

There are several different ways in which one can describe speech. One may use the ideas of information theory and consider speech from the point of view of its information content (Shannon, 1968). Alternatively one may characterize speech as a signal which somehow carries the message information and look at properties of the acoustic speech waveform using parametric descriptions of the acoustic waveform (Rabiner & Schafer, 1978). In addition, one may adopt the approach of phoneticians and describe speech in terms of phonetic sound qualities which are related to the actions of the articulators in the vocal apparatus (Wells & Colson, 1971).

2.1 Descriptions of speech

2.2.1 Articulatory Levels of Description

One also can describe speech at the articulatory level, in terms of the behaviour of the anatomy of the vocal tract (Wells & Colson, 1971). The vocal apparatus, a cross-section through which is given in figure 2.2, provides a means by which nerve impulses from the brain may give rise to the acoustic speech signal. The final speech pressure waveform that is radiated at the lips and nose will depend upon the nature of the excitation and also the position of the articulators. Because the vocal tract transfer function and the excitation are both a function of time, the spectrum of speech is not stationary. By controlling the action of both the articulators and the vocal folds simultaneously, the brain may thus generate a signal in which the underlying message has been suitably coded for acoustic transmission.

The vocal apparatus is a complex sound generator. For voiced speech production, the larynx is the source of the sound and the vocal tract is a time-varying acoustic filter which modifies the laryngeal excitation depending on the position of the articulators. Voiced speech excitation is discussed in more detail in a later section. For voiceless excitation, the sound source is due to turbulent airflow at a point of construction in the vocal tract, and the location of this point is again dependent upon the position of the

articulators. Frication occurs only when the flow of air through constrictions in the vocal tract exceeds a certain critical value. Above this value, determined by the Reynolds number for air, the flow of air becomes turbulent. This turbulence gives rise to an acoustic disturbance that is noise-like in character. That is, un-correlated and with a flat spectrum.

The power needed to generate the sound largely comes from the breathing mechanism; the sources of air are often referred to as the air streams. The most common air stream due to exhaling from the lungs is known as pulmonic egressive. In addition there are oral and pharyngeal air-streams due to air movement caused by the action of the mouth and pharynx respectively. The respiratory system can be controlled by the brain so that breathing fits in to suit the speech. Mainly exhaled air is used for speaking, and expiration may last over 10 seconds in some cases.

2.2.2 The vocal tract

The vocal tract consists of two irregular tubes. There is a passage that connects the larynx to the pharynx, to the mouth and then to the outer air. In addition, when the soft-palate is lowered, there is another passage between the larynx to the nostrils to the outer air. The acoustic behaviour is the result of reflections and standing waves in these tubes and is dependent on the natural frequencies of vibration and damping within the system.

The dimensions of the vocal tract determine its resonant, or formant, frequencies. The relationship between these resonances is known as the formant structure. The vocal tract can be controlled by will to generate changes in this formant structure that are perceptibly different to a listener by the action of different articulators. Formant structure is important because it provides one means to distinguish sounds.

The articulators are the parts of the vocal tract that can be moved to alter the sounds that can be produced. The tongue can be moved up, down, backwards and forwards in order to change the effective length and cross-sectional area of the vocal tract. In

addition, the opening at the lips can be altered, the soft-palate can be opened and closed, and the jaw can be raised and lowered. The vowel systems in languages exploit all of these methods to change the formant structure.

The motion of the articulators is constrained by their anatomy and the muscles that move them. Consequently they can only move at a limited rate from one position to another. As a result of this the present location of the articulators will have some effect on their future position. These effects manifest themselves in the speech signal as assimilation effects.

2.3 PHONETIC LEVELS OF DESCRIPTION

A description of speech that is related to the articulatory descriptions is one based upon the phonetic qualities of speech (Wells & Colson, 1971; O'Connor, 1973; Ladeford, 1975). The field of phonetics is the study and description of speech sounds. It is concerned with what sounds we produce and how we produce them.

Phonetic descriptions are based on perceptible differences in the way the vocal tract of the speaker is used to produce speech sounds. Most languages, including English, can be described in terms of a set of distinctive sound units that are known as phonemes. A table of the phonemes of English, together with examples of them, is given in table 2.1.

A phonetician can write down a representation of speech sounds using a phonetic transcription, which consists of a set of symbols. At the segmental level these symbols indicate the place and manner of articulation as well as the presence or absence of voicing. The manner of articulation refers to the kind of articulation used, for example nasal, rolls, plosive, lateral, affricate. A description of the setting of the lips is also important and it is required to know their rounding, spreading and protrusion. Suprasegmental aspects of speech, such as the intonation of an utterance has a linguistic component that may be described in terms of a fall, rise, rise-fall, fall-rise, etc.

2.3.1 Phonemes

The important point about phonemes is that they are sound units that are contrastive with respect to one another and can be used to discriminate between words. A phonetician shows that two sounds (allophones) are phonemes by finding what is known as a minimal pair to demonstrate that a contrast exists between them. This is a pair of different words that are distinguished on the basis of the phoneme under investigation. The contrastiveness of a particular pair of sounds depends upon the given language and even the dialect. Consequently a given phonemic transcription system may not be suited for transcribing other languages. Phonemes can themselves be classified into vowels, diphthongs, semivowels and consonants.

2.3.2 Allophones

A phoneme has variants known as allophones. The allophones of a phoneme constitute a set of sounds that do not change the meaning of a word, are similar to each other and occur in phonetic contexts different from one another (Ladefoged, 1975).

The allophones belonging to a given phoneme may either be arranged into complementary distribution or in free variation. If two allophones are in complementary distribution, this refers to the fact that the particular allophone used is dependent on the context (that is, the neighbouring phonemes). If two allophones are in free variation, the particular allophone used is freely selected and not dependent on context. Sounds that are in complementary distribution or free variation are only said to represent the same phoneme if they are phonetically similar. That is, they must have most of their phonetic features in common and they must sound similar to native speakers of the language.

There are various effects that occur in continuous speech. Two of these are assimilation and elision. Assimilation is a phenomenon whereby a phoneme consonant changes so that is has, for example, the same place of articulation as the following consonant. This makes the production of the sounds easier, since it requires less articulator movement

than would otherwise be needed. Another related phenomenon is elision, whereby a phoneme in an utterance is missed out, again to facilitate speech production by simplifying the required articulations.

It is valuable to make some brief general statements concerning the acoustic properties of certain categories of speech sound, as an aid in understanding the problems involved in speech fundamental period estimation.

2.3.3 Consonants

Consonants constitute the sounds that are not vowels and are differentiated by place of articulation (bilabial, labiodental, alveolar, dental, velar, paleto-alveolar, post-alveolar) their manner (plosive, fricative, affricate, nasal, continuant) and whether or not they are voiced. The differentiation between vowels and consonants must be made in terms of the relationship of the sounds in a language system and cannot be done solely on the basis of acoustic characteristics.

Plosives are transient non-continuant sounds and are characterised by three distinct phases. Firstly there is an approach phase, during which the appropriate articulators move towards their target positions. Secondly there is a hold phase, where the vocal tract is blocked off by closure of the articulators. Finally there is the release phase, when the articulators separate again. After the plosive release there may be a voiceless excitation due to the release of breath, and this is known as aspiration. Therefore plosives give rise to a brief transient burst of noise, as released air flows through the constriction. Thus a plosive is characterised by a short silence typically followed by a short noise burst when the stop is released. The length of the silence depends on the tempo of the utterance. It is shorter in voiced sounds than unvoiced sounds. However, the main difference between voiced and unvoiced plosives is that in the former the vocal folds vibrate during the closure as the pressure builds up, whereas in the latter case they do not. Often a small amount of low frequency energy can still radiate through the walls of the throat during the closure in a voiced plosive.

In an affricate, there is a plosive followed by a homorganic fricative. The latter is a fricative with friction occurring at the point of release of the plosive.

Nasal consonants involve the lowering of the soft-palate and a complete closure in the oral cavity so that air can only escape via the naso-pharynx. When the nasal passage is open, the closed oral cavity serves as a resonant cavity that traps acoustic energy at its natural resonant frequencies. The effect of this is to add an anti-resonance to the transfer function of the vocal tract, and results in the removal of energy from the radiated speech at the frequency of this anti-resonance (Flanagan, 1972). Since the oral opening of the vocal tract is closed off during a nasal, nasals consequently are of lower intensity than oral consonants. Different nasal consonants are differentiated by the place at which the obstruction of the oral tract takes place.

Fricatives are consonants in which there is turbulent air flow at a narrow region in the vocal tract, giving rise to noise-like acoustic excitation at the point of the narrowing. The location of the point of the narrowing determines which fricative is produced. This noise source is filtered by the action of the resonance of the oral cavity forward of the constriction and the anti-resonance of the oral cavity behind the constriction. Due to their noise-like excitation, fricatives are characterized as having non-periodic waveforms with significant energy at high frequencies (that is above a few kHz, which is not the case for vowels). In voiced fricatives, the point of constriction in the vocal tract is the same as for their unvoiced phoneme counterparts. However, there is also voiced excitation due to vocal fold vibration.

2.3.4 Vowels

Vowels are voiced sounds that are characterized by a lack of constriction of the vocal tract (it should be noted that whispered speech can be still treated as voiced phonemically, even though there is no vocal fold vibration but turbulence at the glottis instead). It is essentially the cross-sectional area of the vocal tract that determines its resonant frequencies and consequently the vowel quality that is produced. The dependence of the cross-sectional area of the vocal tract on the location in the vocal

tract is known as the area-function of the vocal tract. For vowel sounds there are no obstructions of the vocal tract, although the area-function depends mainly on the position and attitude of the tongue, and also to a lesser extent on the position of the jaw, soft-palate and the rounding of the lips. The vertical position of the tongue is often described in terms of height of the tongue, where a CLOSE tongue position represents the highest the tongue can be raised, whereas a OPEN tongue position is the furthest down it can be placed. The horizontal position of the tongue is described as FRONT, CENTRE or BACK, depending upon whether the tongue is forward in the mouth, midway or back in the mouth.

From the production point of view, vowels are more difficult to describe than consonants because the shape of the vocal tract cannot be as easily identified.

The auditory quality of a vowel is usually described by ear with respect to a reference set of vowels, known as the cardinal vowels. The quality of these vowels is independent of language and the cardinal vowel system provides a classification scheme on the basis of perceptible difference between a given vowel and the reference set. The cardinal vowels consist of a set of vowels that provide a coverage of all the possible vowels that can be produced. Thus they constitute a sampling of vowel space along the dimensions of open to close and front to back. In addition to tongue position, vowels may have different amounts of lip rounding.

In the case of diphthongs, the vocal tract area function changes smoothly between those of the appropriate two vowels. In all other respects, a diphthong has the features of an ordinary vowel.

Semivowels are a group of phonemes that are difficult to characterize. Their acoustic properties are similar to vowels and they are generally characterized by a gliding transition of their area-function between those of the adjacent phonemes. Consequently they are strongly influenced by their context. The distinction between semivowels and vowels is made linguistically with reference to their behaviour in a syllable, and not only on acoustic grounds.

2.3.5 Intonation

The most important function of speech fundamental frequency is as the carrier of intonation. Intonation is the temporal pattern of perceived pitch and it has two different purposes. It can convey grammatical information that forms part of a language system. As such, it is mainly the relative change in intonation that is important. For example, it can be used as a means of encoding stress into an utterance, which provides a means of emphasizing certain words. In addition, it can also convey information relating to the emotional state of the speaker. The fundamental frequency contour is important for the intelligibility and naturalness of the utterance (O'Connor & Arnold, 1961). In tone languages (such as Chinese) fundamental frequency changes produce lexical meaning contrasts.

2.5 DIGITAL REPRESENTATIONS OF THE SPEECH WAVEFORM

Speech propagates through the air as an acoustic pressure waveform. For the purposes of computer speech analysis, it is necessary first to convert it in a different form and this usually takes the shape of amplitude measurements of the speech pressure at regular time intervals (Rabiner & Schafer, 1978). The conversion of the acoustic speech waveform into a digitized speech pressure waveform involves firstly converting acoustic pressure variations in the air to electrical fluctuations using a pressure microphone (It is also possible to use a velocity microphone which responds to the velocity of the air rather than the pressure, but this type of microphone is less common). The output from the microphone is then low-pass filtered and then sampled at a uniform rate by means of an analogue-to-digital (A/D) converter, which converts the amplitude measurements to a number. It is necessary to ensure that the bandwidth of the signal to be sampled is less than half the sampling frequency, otherwise aliasing will occur and this is prevented by the low-pass filter (Nyquist, 1928). If the sampled data is aliased, then it will not be possible to reconstruct the original waveform from it, because it no longer uniquely represents the original waveform. It is also important that the resolution of the A/D converter is sufficient for the application, because the process of quantization of the continuously valued input signal into a set of discrete levels introduces uncertainty

in the signal representation that can be considered as additive noise (Rabiner & Schafer, 1978).

A description of speech in terms of the sampled representation of the speech pressure waveform is a very general representation that is only concerned with preserving the wave-shape of the signal by the appropriate choice of sampling frequency and levels of quantization. Such a description involves no other a priori knowledge particular to the characteristics of speech.

2.5.1 Parametric models

Parametric models of speech are more abstract that this and are concerned with representing the signal in terms of the output from a production model (Fant, 1970; Flanagan, 1972). In a simplest case of such a model, speech production is represented as an excitation source driving a time-varying linear filter that represents the acoustic effects of the excitation spectrum, vocal tract, and radiation effects at the lips. For voiced speech, the excitation source in this model must mimic the excitation due to the repeated opening and closure of the vocal folds. For voiceless excitation, it must mimic the noise-like excitation due to turbulent airflow in the vocal tract. In more sophisticated models, the effects of the excitation spectrum, vocal tract and lip radiation can be represented separately. In both cases, the time-varying linear filter must account for the resonances of the vocal tract, which are known as the formants. For simple purposes the vocal tract can be approximately modelled as two tubes. This production model is useful for the generation of synthetic speech as well as a model for speech analysis. For synthesis of voiced speech it is the first three resonances that are most important (Holmes, 1988).

2.5.2 Acoustic variability of speech

Different speakers will have different larynx sizes, vocal tract sizes, phonetic and linguistic upbringing, speech habits, emotional states and vocal fold characteristics. All these factors affect the speech produced in different ways. Consequently there will be

a large difference in the acoustic realizations of utterances for different speakers (cross-speaker variabilities). In addition, variabilities also arise because of differences that occur in a given speaker as a function of time (occasion-to-occasion variability). An example of speech variability in short-term acoustic representations is demonstrated by the fact that the first two formants for different speakers for the same vowels overlap, as shown in figure 2.3 (Peterson & Barney, 1952).

2.2 VOICED EXCITATION

There now follows a more in-depth description of voiced speech excitation, because this area is of particular interest to speech fundamental period estimation.

The basic acoustic function of the larynx is to act as the sound source during voiced speech production. A cross-section through the larynx is shown in figure 2.4, and front and back views are shown in figure 2.5. Its action gives rise to a glottal wave which acts as a carrier for the speech message imparted by the effects of the vocal tract. In addition, the characteristics of the voice source are important because it contributes to the means by which the physical, psychological and social characteristics of the speaker can be conveyed.

2.2.1 Vocal Fold Vibration

Voiced excitation occurs when air flows between the vocal folds causing them to vibrate and the main peak of excitation results from their closure. The result of vocal fold vibration is thus a modulation of the air flow that passes into the vocal tract and constitutes a quasi-periodic acoustic excitation.

The vibration of the vocal folds that characterises voiced speech is complex. The vibrating system is three dimensional, and consequently its motion is more complicated than simple harmonic motion. It is a vibrating system that has different modes of oscillation. In normal voice, the vocal folds constitute a thick shelf across the larynx (figure 2.4) all of which moves periodically together and then apart again. In other

modes of vibration, the vocal folds can be thinned out at the edges. This results in a lighter vibrating section and consequently a higher frequency of vibration.

2.2.2 Mechanism of vocal fold vibration

The mechanisms involved in vocal fold vibration can be understood by considering the following sequence of events, which follows what is known as the myo-elastic theory of phonation (Van den Berg, 1957). Air from the lungs during exhalation is the main airstream used in phonation (known as the pulmonic airstream). The laryngeal muscles can cause the vocal folds to close, thus blocking the air passage. If this happens during exhalation there will be a build up of air pressure below the vocal folds, which will eventually force them apart. After this happens, there are two mechanisms involved in bringing them back together again. Firstly the muscle fibers and ligaments in the vocal folds are elastic, and after the vocal folds have been forced out of position, they spring back to their resting position. Secondly, as air flows through the constriction in the vocal folds, its velocity increases and consequently its pressure decreases, due to the Bernoulli effect. When the air pressure between the vocal folds drops, the external pressure tends to force the vocal folds together. There is positive feedback in this mechanism, because the closer the vocal folds get, the faster the air flow and the greater the pressure drop will be. Therefore, the vocal folds are accelerated together, resulting in a strong impulse excitation of the vocal tract as they snap shut. After this, the pressure then rapidly returns to normal atmospheric, and because of the constriction the sub-glottal pressure starts to rise again. Thus the cycle repeats itself. The overall effect is that successive puffs of air enter the vocal tract just above the larynx.

The frequency of vibration of the vocal folds depends upon the sub-glottal pressure and their resistance to movement. The resistance to movement of the vocal folds depends on their mass, length and tension. The effective length of the vocal folds can be adjusted by means of the thyro-arytenoid muscles and crico-thyroid muscles (see figure 2.5). The latter changes the angle between the thyroid and cricoid cartilages thus stretching and lengthening the vocal folds. Since all of the parameters affecting vocal fold vibration rate are controlled by the action of muscles in the larynx and air pressure

and flow, the speaker is able to alter the vibration rate at will.

2.2.3 Laryngographic descriptions of Voiced speech

A device of particular value in the analysis of voiced speech excitation is the laryngograph (Fourcin & Abberton, 1971). A description of the laryngograph and its relationship to vocal fold vibration is of particular importance here because it forms a fundamental part in the training and testing of the fundamental period estimation algorithm which is the subject of this thesis.

A laryngograph operates by measuring the conductance across the larynx at the level of the vocal folds. This is achieved by placing two electrodes across the larynx with a small alternating voltage at several MHz across them. Movement of the vocal folds causes a change in the conductance which is subsequently detected.

The output waveform from the laryngograph thus gives a measure of vocal fold activity and is temporally much simpler than the corresponding speech pressure waveform. The point of closure of the vocal folds, which gives rise to the main peak in excitation, can be easily determined from the laryngograph waveform. The manifestation of the closure of the vocal folds in the laryngograph output signal is well agreed upon (Fourcin, 1974). The point of closure is usually taken as the point of maximum gradient in the closing phase of the laryngograph signal. Agreement on the opening point is, however, less well accepted. This is because as the vocal folds open, they "peel apart" from below and the corresponding effect in the laryngograph waveform is difficult to define as a specific distinct event. Figure 2.6 shows the relationship between vocal fold vibration and the laryngograph waveform for normal modes of laryngeal activity.

2.2.4 Laryngograph signals for different voice qualities

There now follows a description of the characteristics of the laryngograph waveform for different voice qualities. According to Hollein (1972) there are three major vocal registers; modal (normal), falsetto, and vocal fry (creak).

2.2.5 Normal voice

Normal voice is characterised by regular vibration of the vocal folds, without any frication. It is used over most of the speaker's frequency range. This is typically about 90-200Hz for male and 150-310Hz for women.

With normal voice the whole body of the vocal folds vibrates, giving characteristically relatively long vocal fold closure times. The brief velocity peak of the vocal folds that occurs as they snap shut gives an excitation with significant high frequency components, which results in a well defined set of formant frequencies. The speech pressure waveform for normal voice and the corresponding output from a laryngograph are shown in figure 2.7.

2.2.6 Breathy voice

Breathy voice may be characterised by incomplete closure of the vocal folds, and by greater pulmonic airflow than in normal speech. The vocal folds vibrate but do not necessarily make contact, although lack of contact only happens during very breathy voice. The closure points as observed by means of a laryngograph are smoother, because full closure is not made. Also, the open phase is much longer than normal. This results in greater sub-glottal damping of the vocal tract, and the vocal tract resonances are therefore less well defined than with normal speech. There is also noise generated by turbulence at the glottis, which shows up in the speech pressure waveform. A more extreme case of this aspiration occurs in the case of whispered speech, when there is strong air turbulence at the glottis and the vocal folds do not meet. The speech pressure waveform for breathy voice and its corresponding output from the laryngograph is shown in figure 2.8.

2.2.7 Creaky voice

A special case of vocal fold vibration is that of creaky voice. It generally occurs at the end of utterances with falling intonation and it is characterised by laryngeal vibrations

of unusually large duration. Sometimes these are alternated with shorter duration cycles, giving a short cycle followed by a long cycle. The irregularity is perceived as a creaky voice quality. The speech pressure waveform shows clear evidence of vocal tract excitation at each closure, and since the cycle time is large, each excitation of the vocal tract has time to die down a long way before the next excitation occurs, and consequently the excitation points are well defined. There is a tendency for speakers to use creaky voice quality if they want to go down to a low pitch that is below the bottom end of their normal frequency range. The speech pressure waveform for one example of creaky voice and the corresponding output from the laryngograph is shown in figure 2.9.

2.2.8 Falsetto voice

Falsetto voice occurs when only the top edge of the vocal folds vibrates which results in damping of the vocal tract by the sub-glottal system much sooner after the excitation point than in the case of normal voice. This results in a much temporally simpler speech pressure waveform than with normal voice quality. There is a tendency for the speaker to make use of a falsetto voice to reach fundamental frequencies that are above their normal range. The speech pressure waveform for an example of falsetto voice and its corresponding output from the laryngograph is shown in figure 2.10.

2.2.9 Mixed excitation

In some cases both fricative excitation and voicing occur at the same time. This is known as mixed excitation. Because of the pulsatile nature of the air flow via the vocal tract in this condition, the frication occurs in bursts synchronously with the glottal air flow pulses. Figure 2.11 gives an example of mixed excitation in a voiced fricative.

2.2.10 Problems in using the laryngograph

There are several limitation in using electro-glottography in general to estimate the operation of the vocal folds (Colton & Conture, 1990). These range from problems in

obtaining good quality laryngograph signals with some speakers to cases where there are discrepancies between the speech and laryngograph signals.

Only a small fraction of the current from the laryngograph electrodes passes through the vocal folds. As a consequence of this, the laryngograph waveform (known as Lx) is strongly affected by gross larynx movements, blood flow through the neck and the contraction of the extrinsic laryngeal muscles. Figure 2.12 shows a large excursion in the laryngograph waveform that often occurs as a speaker prepare to phonate that has no corresponding acoustic excitation. By high pass filtering this composite signal within the laryngograph, the faster fluctuation due to vocal fold vibration can be emphasized (Colton & Conture, 1990).

2.2.11 Discrepancies between the speech signal and the laryngograph signal

There are circumstances where the laryngograph does not always give a strong indication of voicing when observation of the speech pressure waveform indicated that voicing is indeed present (Howard & Lindsay, 1988). This happens when the vocal folds vibrate without making firm contact and are "flapping about in the breeze" (Childers & Larar, 1984). This mainly occurs towards the end of unstressed voiced segments, when the vocal folds are still vibrating but no firm closure is made. Consequently there is little change in the impedance across the larynx and therefore little fluctuation on the laryngograph waveform. This phenomenon occurs more frequently in the case of female speakers than for male speakers. Figure 2.13 shows the case when there is evidence of vocal excitation in the speech pressure waveform, but little evidence for it in the laryngograph waveform. Conversely, there are occasions when there is laryngograph activity, but no speech pressure waveform, such as during a hold in a plosive. In this case the acoustic excitation occurring at the vocal folds is attenuated by the closure, and consequently there is little or no speech output. Figure 2.14 illustrates this phenomenon.

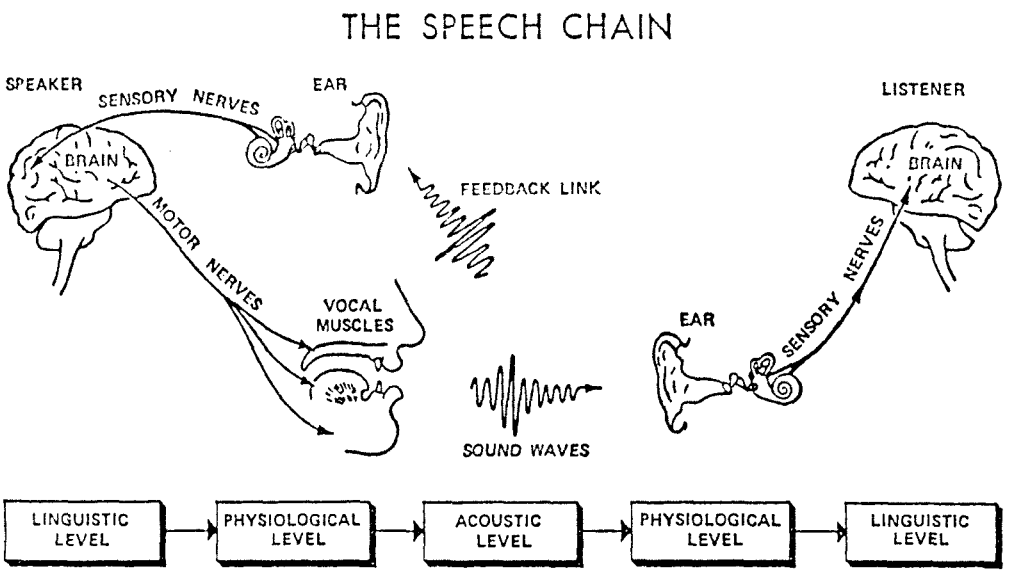


Figure 2.1 The speech chain.

This shows the stages in the generation of a message within the brain of a speaker to its transmission using sound, and then its reception in the brain of a listener (visual information, such as speaker gestures and lip moments, can also contribute to the communication process, but is not shown here). The message is shown to start as activity corresponding to a linguistic level within higher centres in the speaker's brain. Suitable nerve signals are then generated to control the vocal apparatus. This results in the broadcast of an acoustic speech wave which travels to the listener. The sound is then analysed by the ear (more particularly the cochlea) and nerve signals then convey the information to higher centres in the listener's brain, where their linguistic significance is interpreted.

(Taken from Denes & Pinson, 1973).

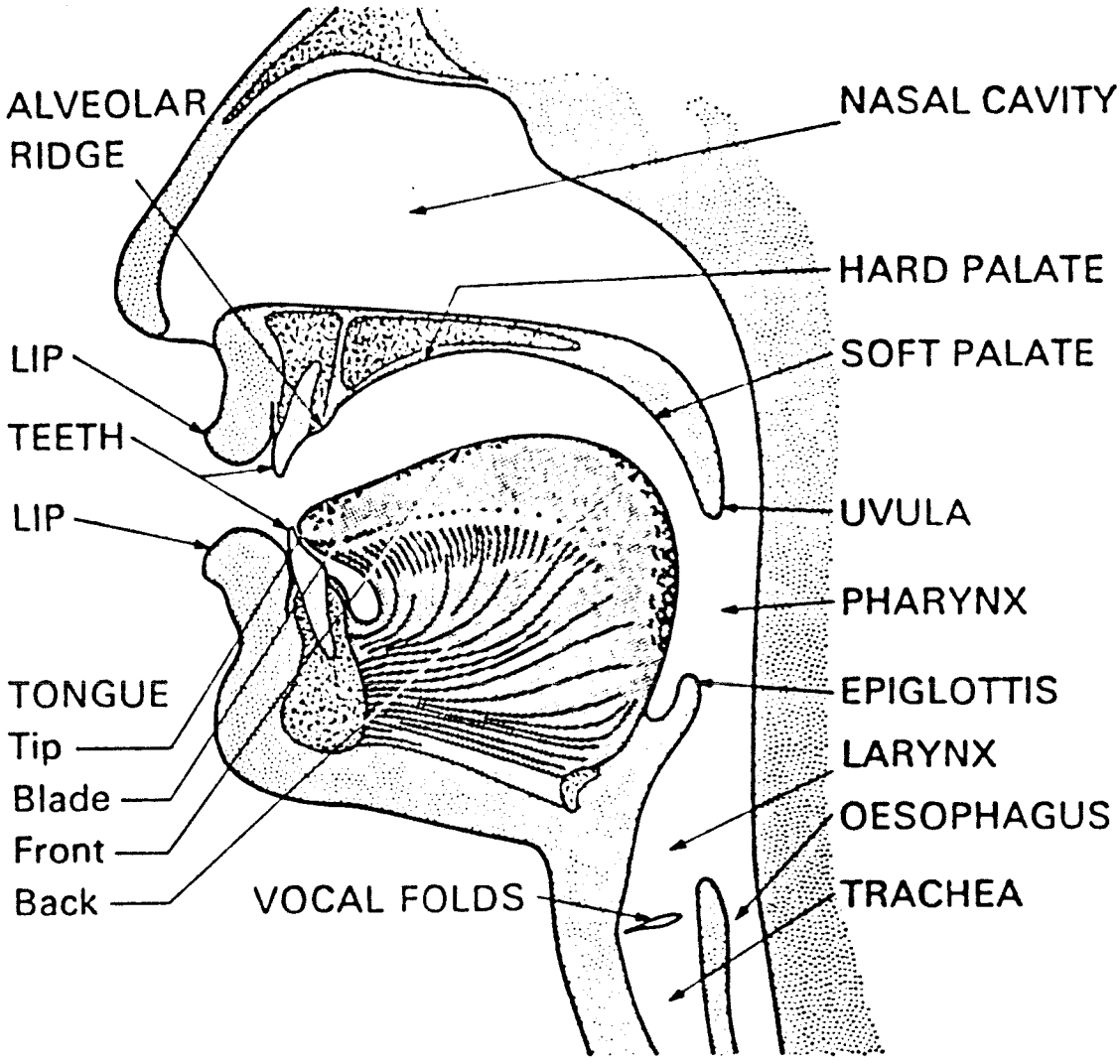


Figure 2.2 Cross-section through the human vocal tract.
The position of the articulators is shown.
(Taken from Wells & Colson, 1971).

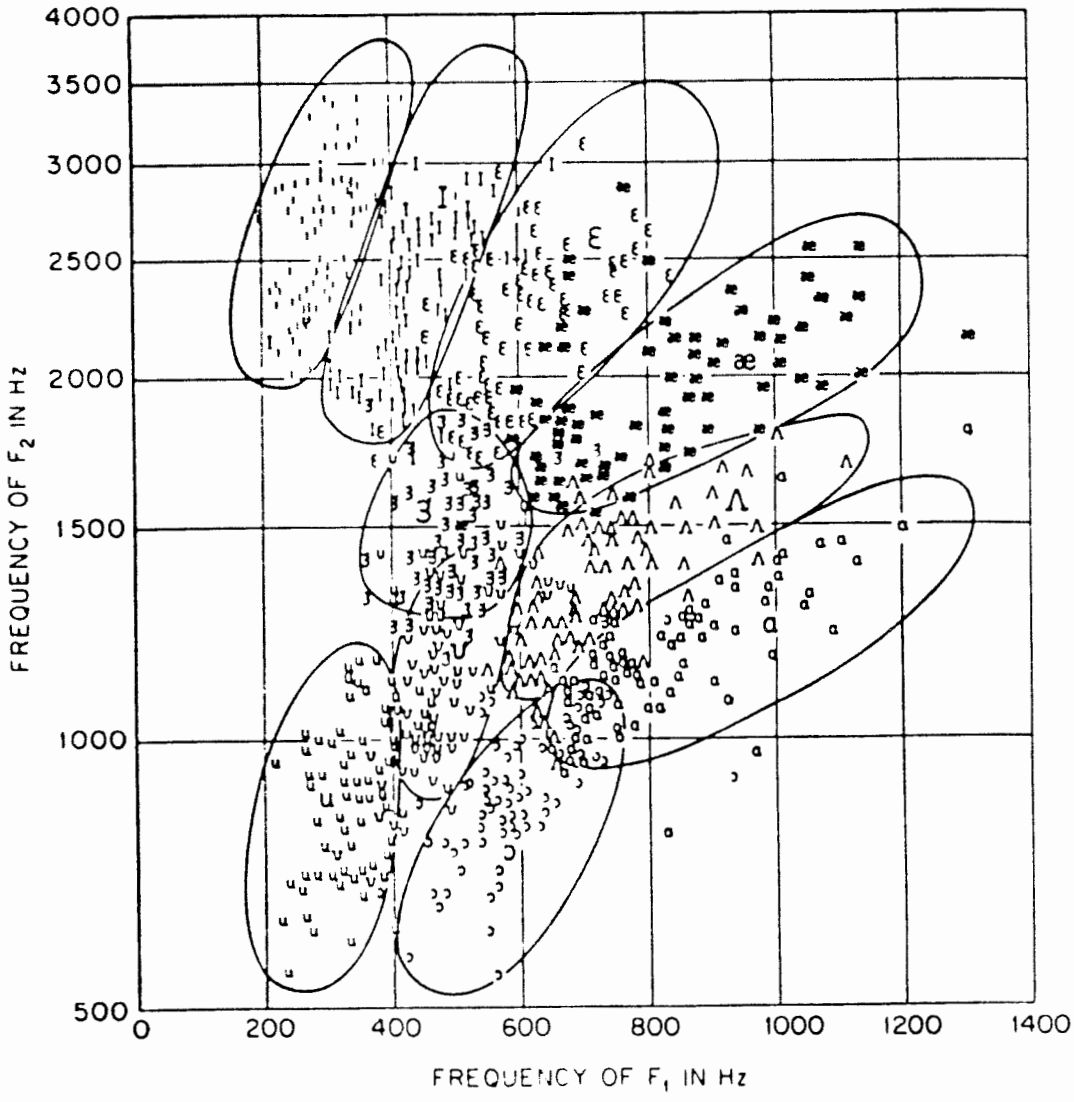


Figure 2.3 Variability of formant frequencies across speakers.

Figure shows the overlap between the first two formant frequencies of different vowels for different speakers.

(Taken from Peterson & Barney, 1952).

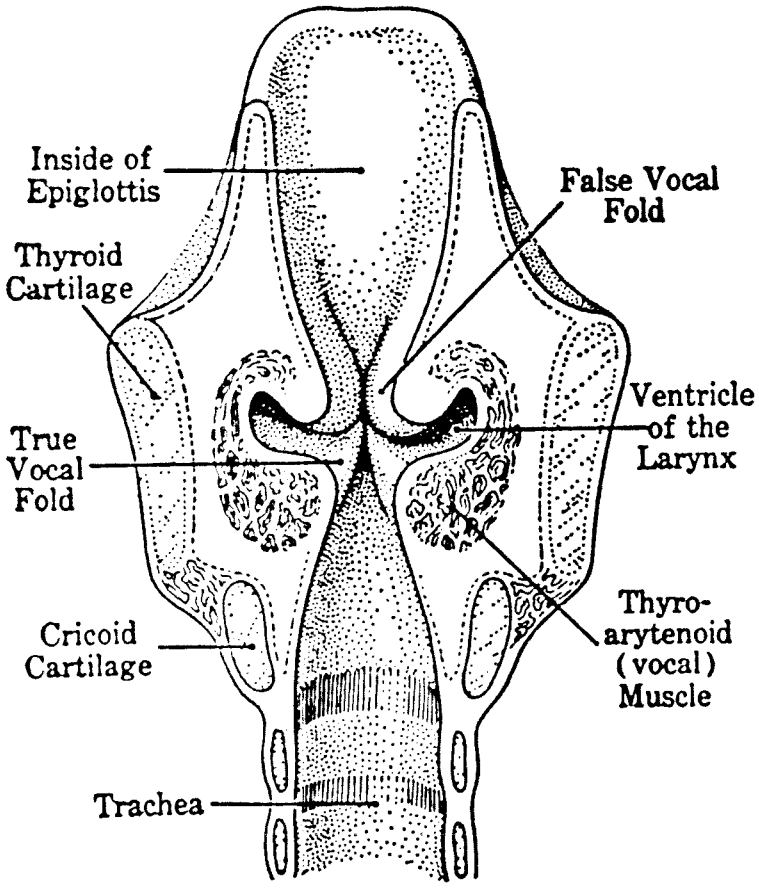


Figure 2.4 Cross-section through the larynx.
The vocal folds can be clearly seen.
(Taken from Borden & Harris, 1980).

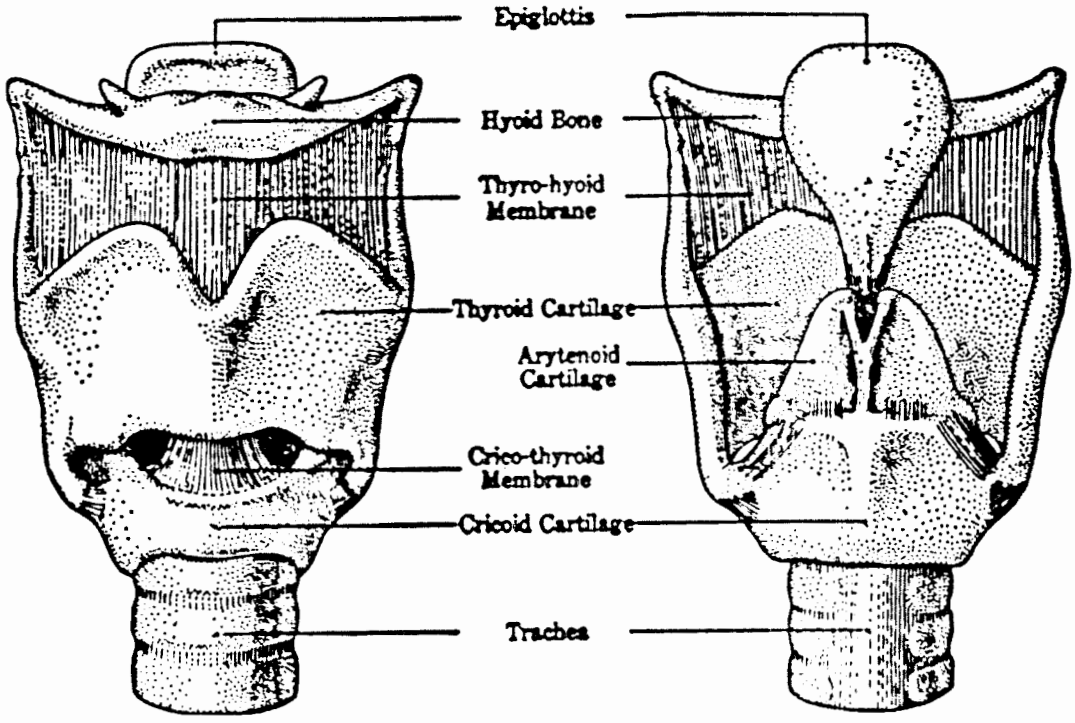


Figure 2.5 Front and rear views of the larynx.
(Taken from Borden & Harris, 1980).

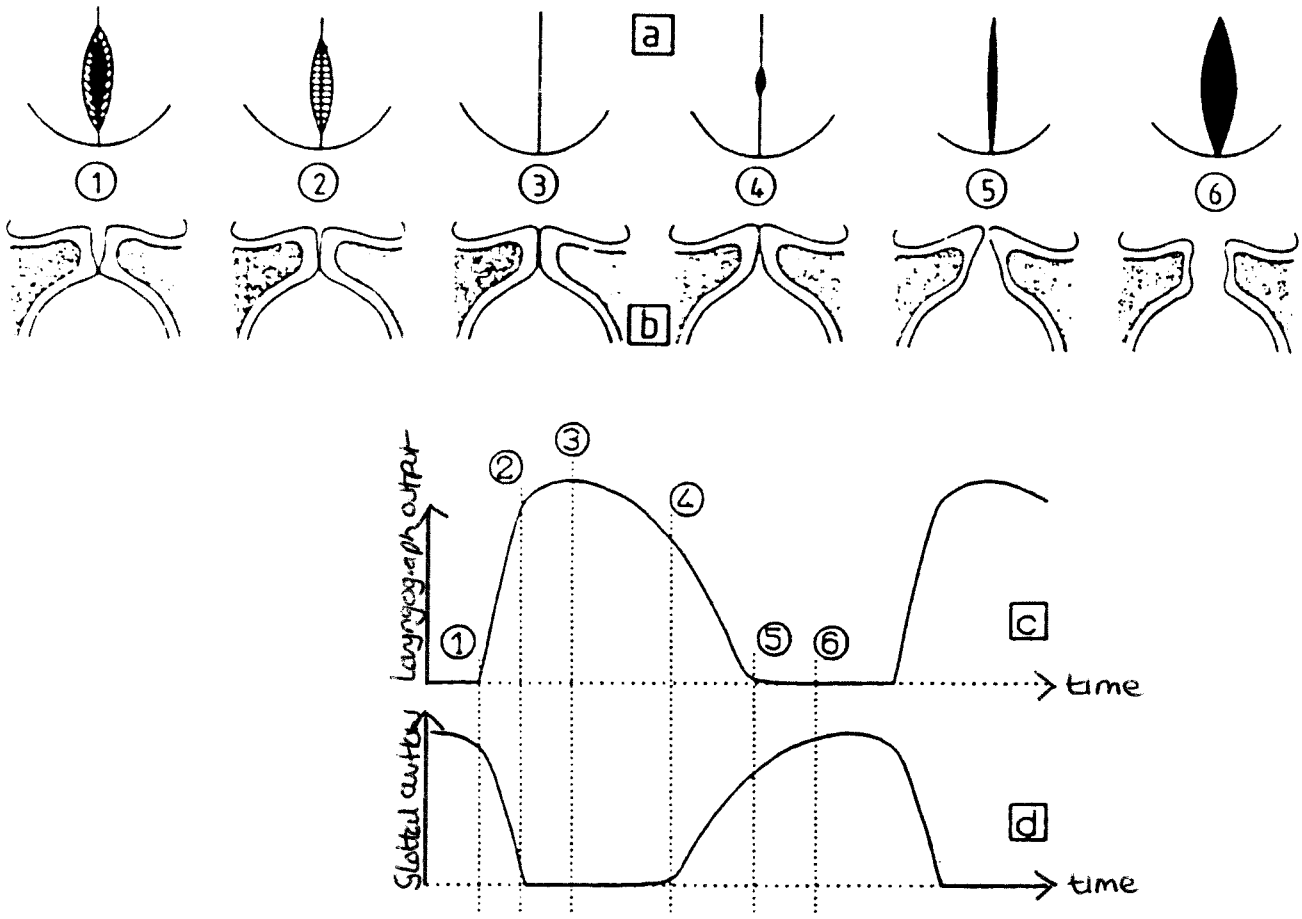


Figure 2.6 The relationship between vocal fold motion and the laryngograph waveform, during normal speech.

Six key stages in a complete period are shown. Diagrams (a) shows the view of the vocal folds from above. Diagrams (b) show a cross-section of the vocal folds. The corresponding effect in the laryngograph waveform is shown in diagrams (c). Diagram (d) shows the corresponding glottal air flow. The marked points are as follows:

(1) is the point of closure at a single point.

(2) is the instant when complete closure has been made over the length of the glottis, but not over the vertical plane.

(3) is the point of maximum closure.

(4) is the point at which opening begins.

(5) is the instant at which the entire length of the glottis is open.

(Taken from Hess, 1983; Base on Lecluse, 1977).

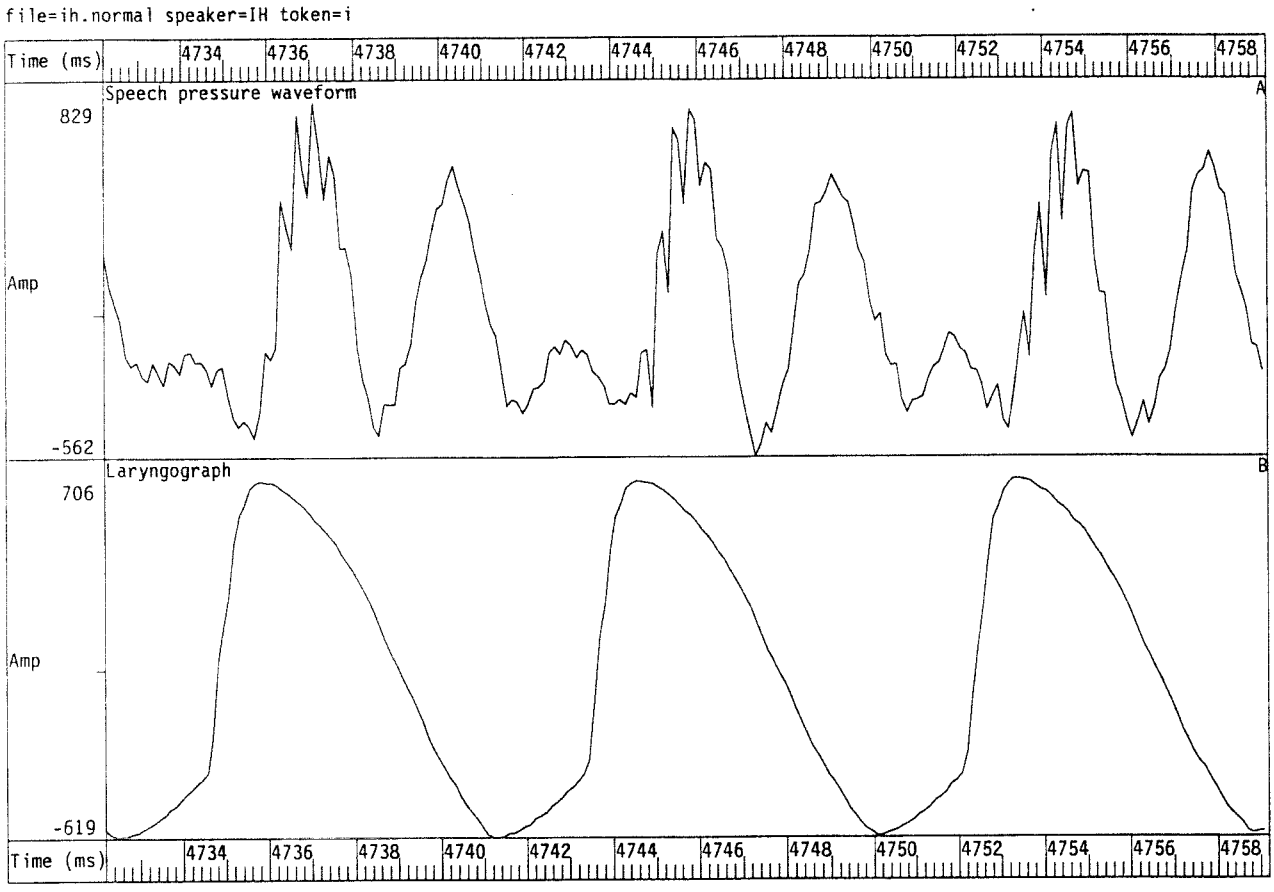


Figure 2.7 Speech pressure waveform and laryngograph waveform for an example of normal speech.

The laryngograph waveshape is similar to that shown in figure 2.6, except high-pass filtering present in the laryngograph has resulted in sloping of the horizontal sections of the waveform.

The utterance is the vowel /i/ spoken by a male.

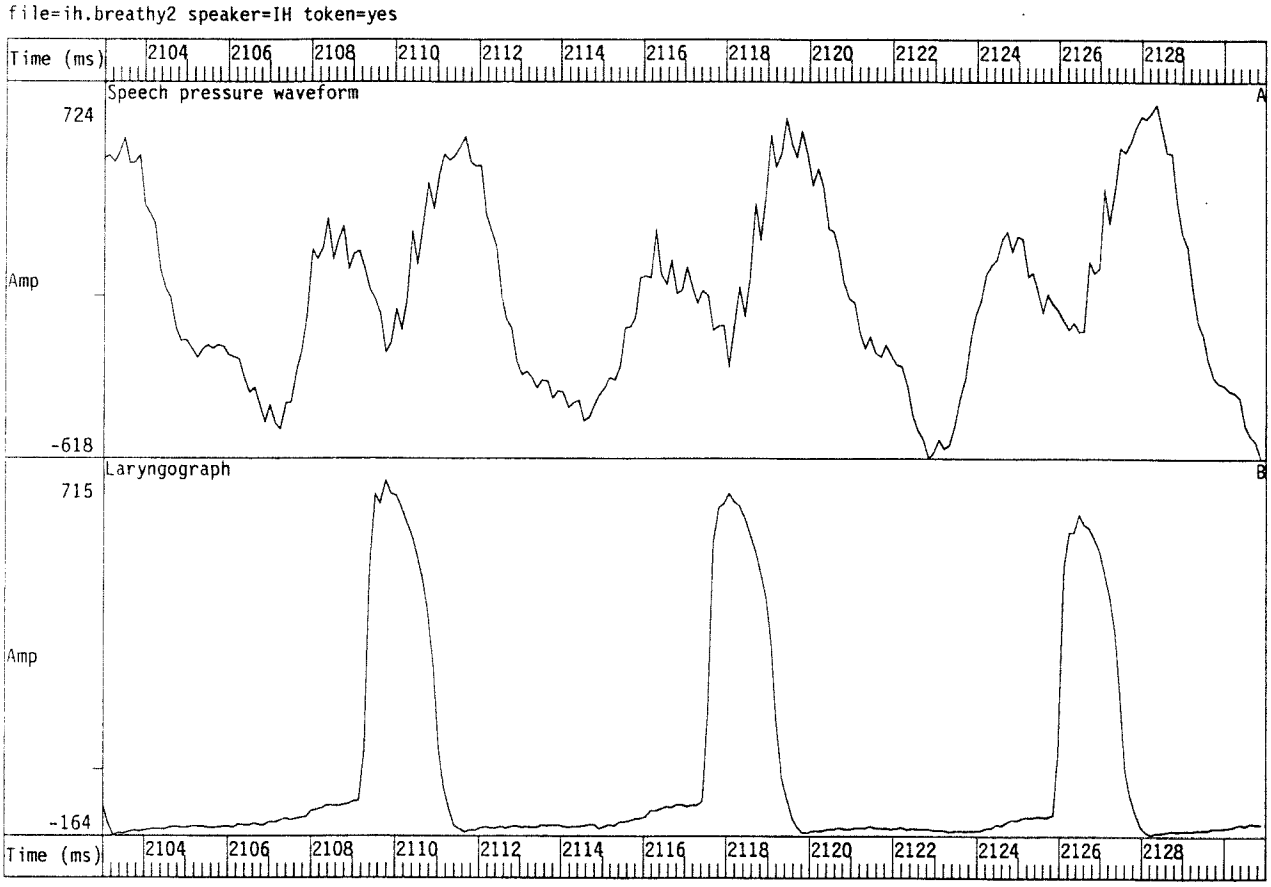


Figure 2.8 Speech pressure waveform and laryngograph waveform for an example of breathy voice quality.

It can be seen that the vocal folds maintain firm closure for a smaller proportion of the period than in the case of normal voice quality. Consequently the laryngograph waveform is positive for a smaller portion of the overall cycle. The utterance is the vowel /i/ spoken by a male.

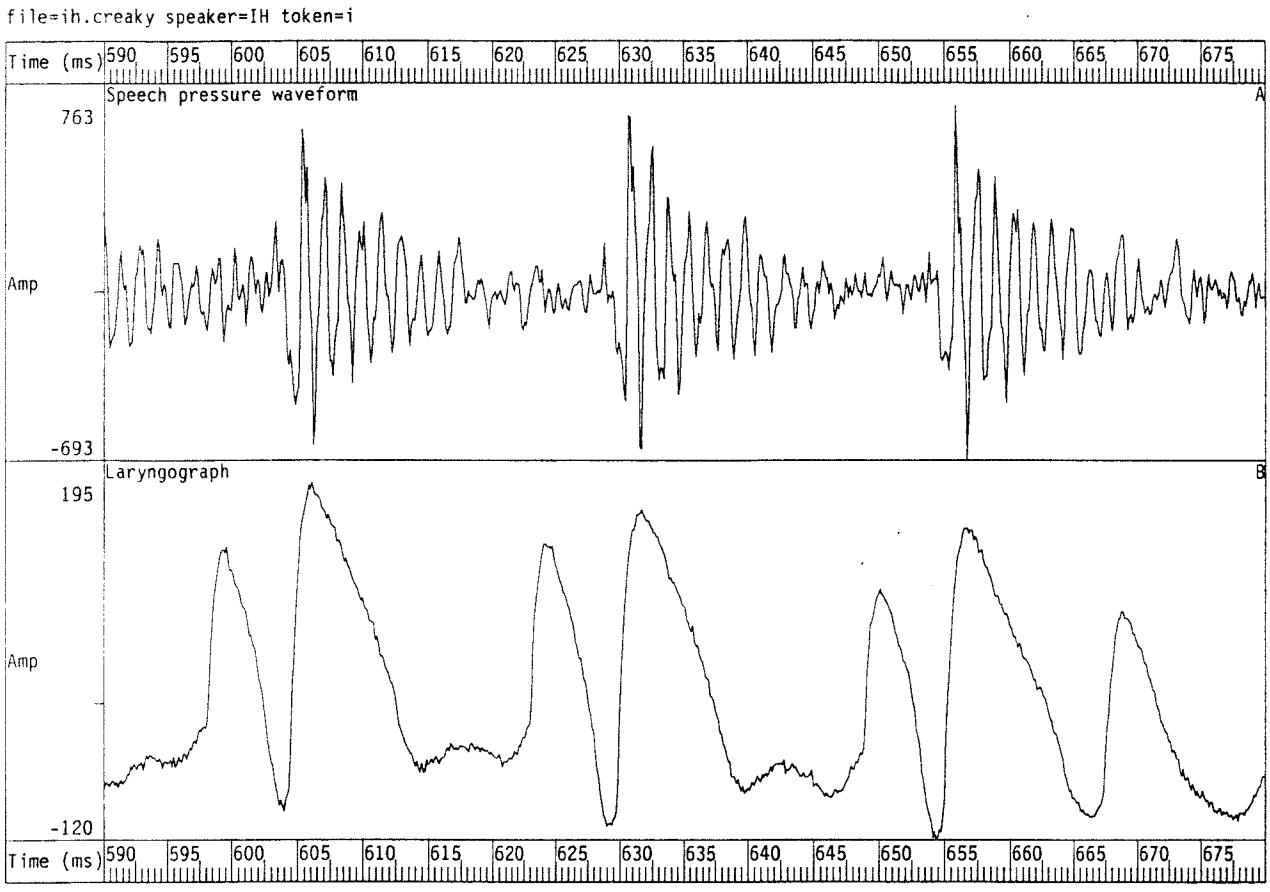


Figure 2.9 Speech pressure waveform and laryngograph waveform for an example of creaky voice quality.

In this case, the vocal fold closures occur irregularly, sometimes with a long closure followed by a shorter closure. The utterance is the vowel /i/ spoken by a male.

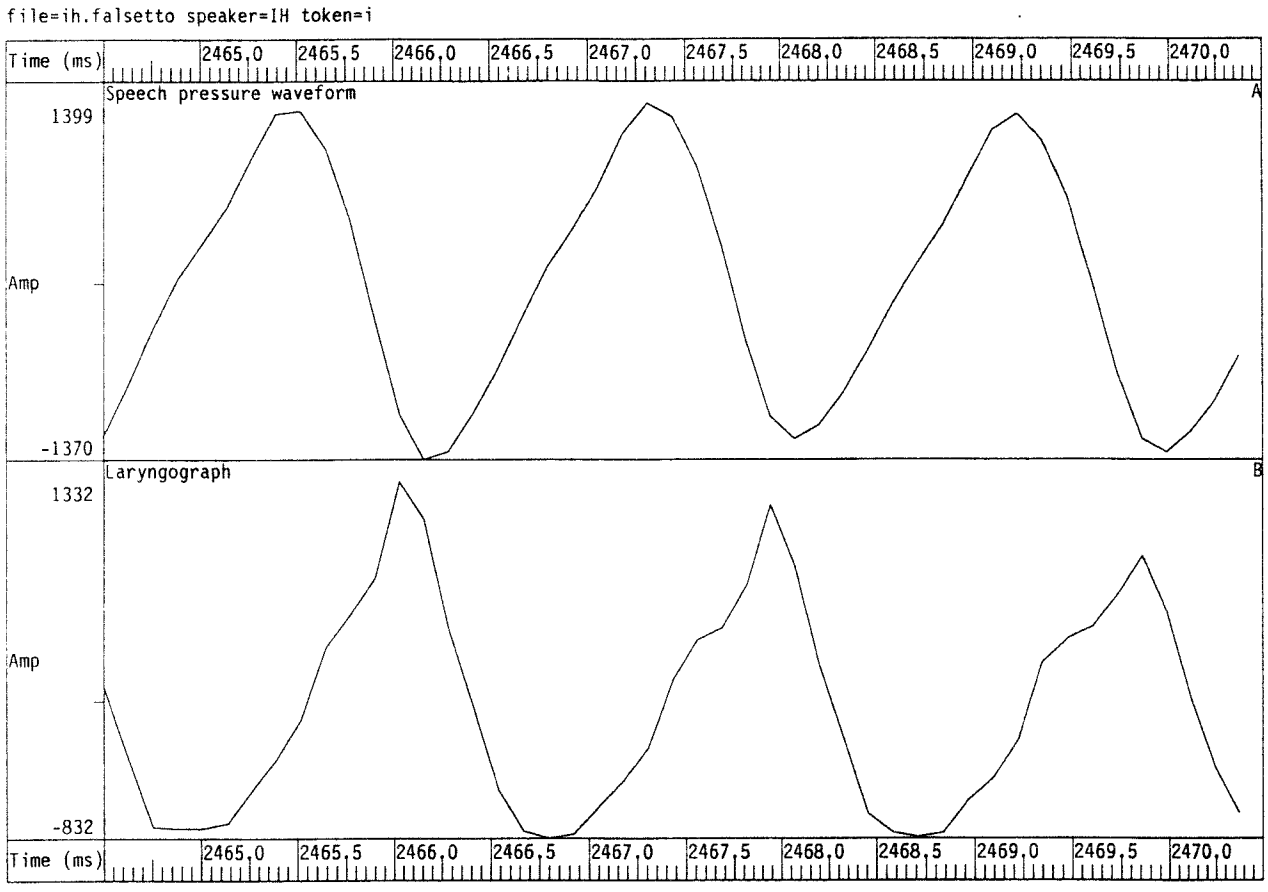


Figure 2.10 Speech pressure waveform and laryngograph waveform for an example of falsetto voice quality.

The utterance is the vowel /i/ spoken by a male.

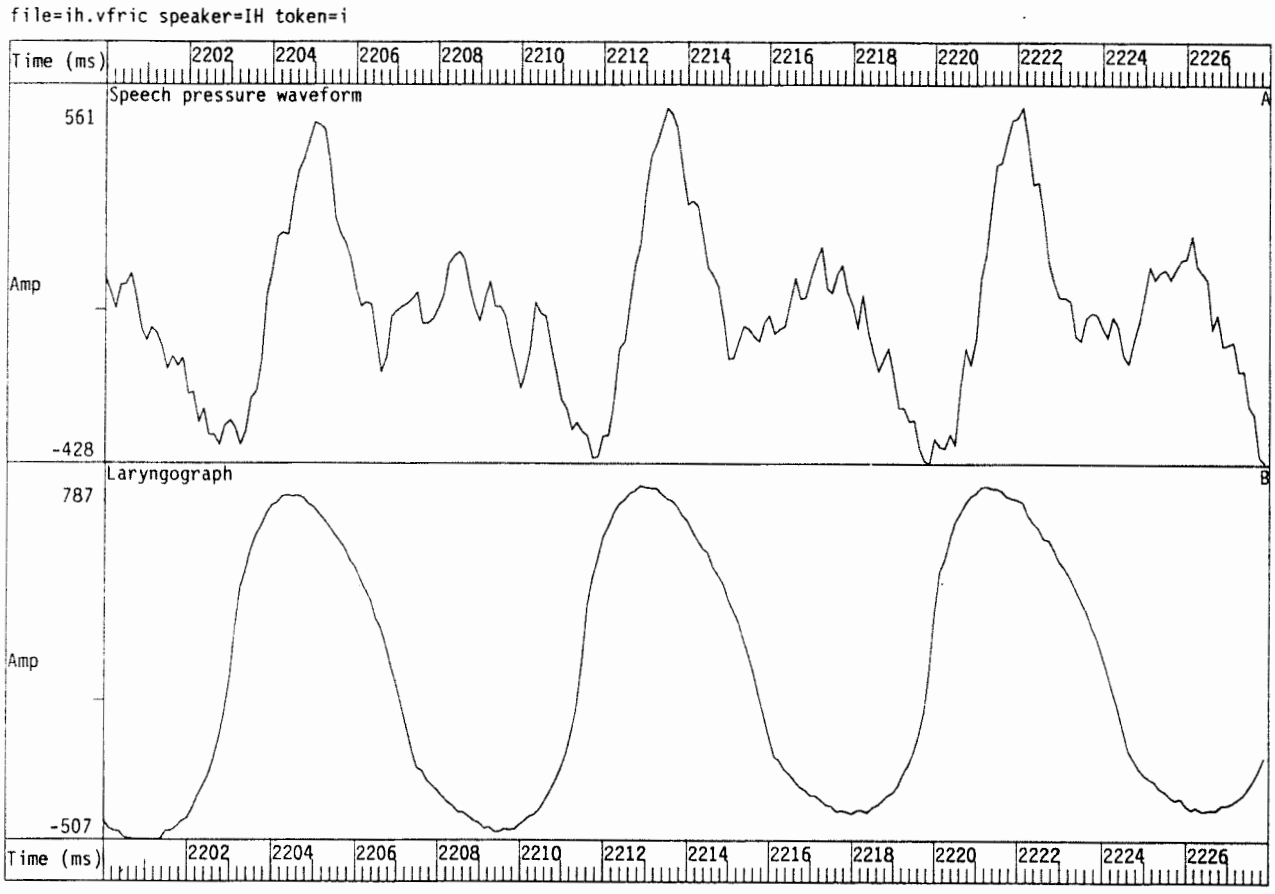


Figure 2.11 Speech pressure waveform and laryngograph waveform for a voiced fricative.

There is fricative excitation in addition to the quasi-period excitation due to vocal fold vibration. It can be seen that the frication occurs synchronously with the vocal fold vibrations. The utterance is the voiced fricative /z/ spoken by a male.

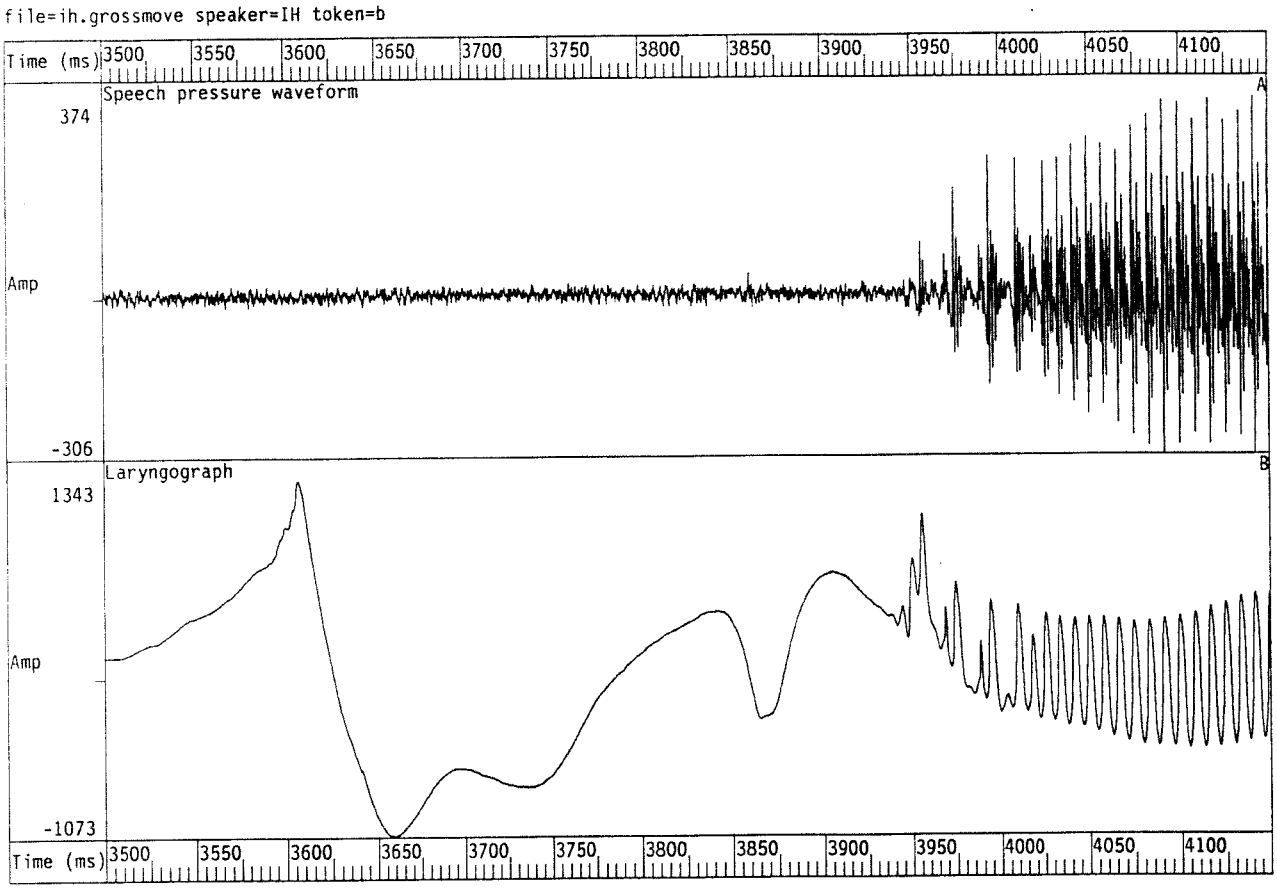


Figure 2.12 Unwanted excursion in laryngograph output waveform.

It can be seen that prior to phonation there are spurious excursions of the laryngograph waveform that have no acoustic significance. The utterance is the onset of the vowel /i/ spoken by a male.

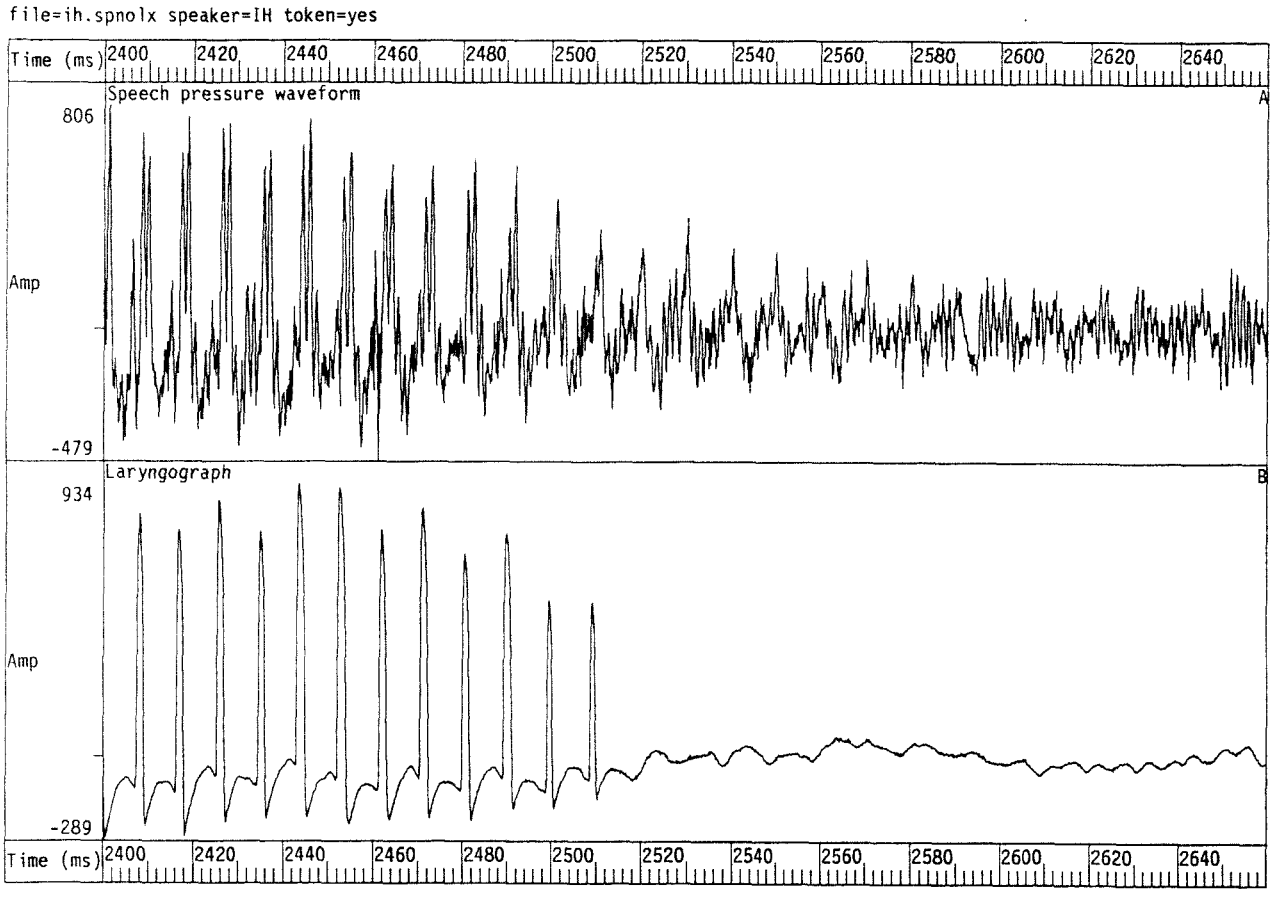


Figure 2.13 Evidence of vocal fold vibration in the speech pressure waveform, but little in the laryngograph signal.

This situation arises when firm vocal fold contact is not made, but the vocal folds are still vibrating. The section shown is the end of the utterance "yes" spoken using a breathy voice quality by a male.

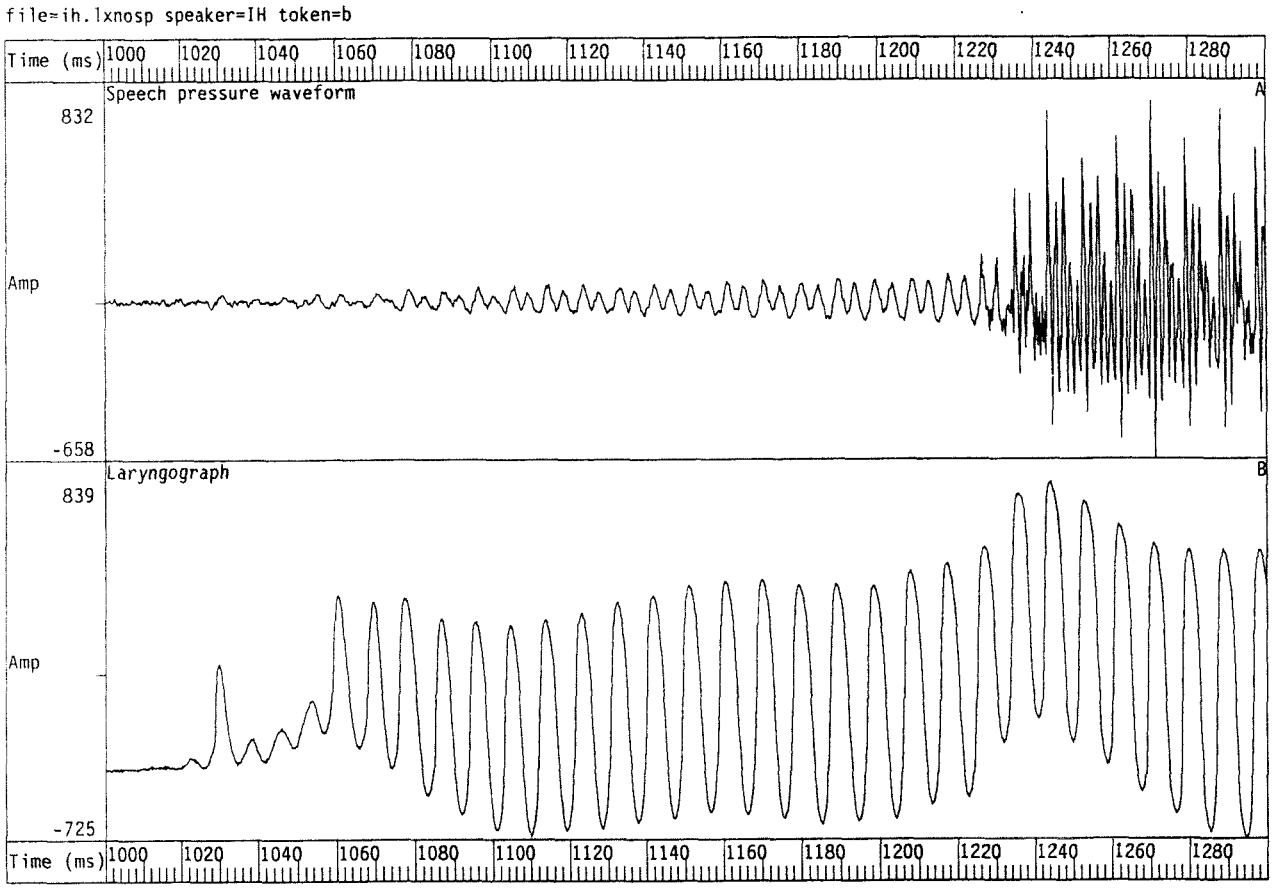


Figure 2.14 Evidence of vocal fold vibration in the laryngograph waveform, but only a small amount in the acoustic speech pressure waveform.

This occurs when there is a block in the vocal tract, such as in the case of the hold stage in a plosive, but there in still sufficient air flow through the larynx to maintain vocal fold vibration (this air flow results in an increase of air pressure behind the constriction). The section shown is the lead up to the plosive /b/, spoken by a male.

CHAPTER 3: ISSUES IN SPEECH FUNDAMENTAL FREQUENCY AND PERIOD ESTIMATION

3.1 INTRODUCTION

This chapter explores some of the issues and problems involved in the estimation of speech fundamental frequency. Firstly there is a discussion of what is meant by the terms fundamental frequency, fundamental period and pitch. Some aspects of human pitch perception and their relationships to the requirements of algorithms that estimate speech fundamental frequency are then discussed. Finally, there is a brief introduction to the basic approaches to speech fundamental frequency estimation by machine.

3.1.1 Fundamental frequency and pitch

Before entering into an in depth discussion of the problems involved in estimating speech fundamental frequency, it is necessary to precisely define the problem. It is also enlightening to investigate the relationship between the parameters fundamental frequency, fundamental period and pitch.

The automatic estimation of fundamental frequency of voiced speech excitation is often misleadingly referred to as pitch analysis. Pitch properly refers to a percept rather than a parameter of speech production (McKinney, 1965), although the term pitch is often used in current technical literature to express both fundamental frequency and fundamental period. Pitch is a subjective phenomenon whereas fundamental frequency is open to physical measurements. There is a relationship between pitch and frequency, but it is rather complex, although pitch is correlated with the physical feature of fundamental frequency. Thus, when one is considering speech at the acoustic level, it is preferable to use the concept of fundamental frequency. It is also useful to distinguish between fundamental period estimation, implying a period-by-period estimation process, and fundamental frequency estimation, which result from short-term analyses.

3.1.2 Approaches to speech analysis

Zwicker, Hess & Terhardt (1967) looked at the problem of speech analysis from three different points of view. All of these are related for stationary periodic signals, but not for real speech.

1] Speech can be considered from the production viewpoint, and it can be analysed using knowledge about the way in which it was generated. The parameters that are estimated using such an approach are related to the control parameters of the speech production process. At the lowest level of speech production, one can define the larynx fundamental periods as the time between successive vocal folds closures. Similarly the larynx fundamental frequency can be defined as rate of the vocal fold vibration.

2] From a perceptual viewpoint, speech may be analysed in a fashion that is similar to the processing that is believed to occur in the human auditory system. In the case of a human listener or by using a model of human pitch perception, the perceived pitch of a speech stimulus can be defined as the frequency of a pure tone that evokes the same perceived pitch.

3] From the signal processing viewpoint, speech analysis does not necessarily take account of speech production or speech perception, but seeks to describe the signal in some mathematically optimal way. If the speech production process is not taken into account, the fundamental frequency and the fundamental period of the speech can be defined in terms of the minimum repetitive period of the signal, or corresponding to the common sub-multiple of a set of harmonics. This task can be carried out using digital signal processing techniques, such as auto-correlation.

3.1.3 Simplified model of speech excitation

A simplified model of the excitation of voiced speech sounds was described by McKinney (McKinney, 1965). In this model, a volume velocity glottal excitation function ug(t) excites a passive linear system. This is illustrated in figure 3.1. The

supra-glottal system transfer function represents the characteristics of the vocal tract and radiation at the lips. The glottal wave is often modelled as a pulse train. However, in this model ug(t) will be considered to be due to the sum of a pulse train pg(t) and a slowly varying function vg(t). The latter term is required because the volume velocity at the glottis does not always go to zero during each cycle of vibration. The function pg(t) will be called the excitation pulse function. Each individual excitation pulse has an associated time of occurrence, its excitation pulse time. In order to make this coincide with the principal excitation of the formant resonance in the vocal tract, the excitation time is defined to occur at the time when the excitation pulse function reaches a zero value at the end of each glottal cycle (see figure 3.1). This time is also the instant of glottal closure, and corresponds to the maximum positive gradient in a laryngograph waveform.

3.2 FUNDAMENTAL PERIOD, FUNDAMENTAL FREQUENCY AND PITCH

3.2.1 Definition of fundamental period

Hess (1983) states that there are three possible ways to define T_0, the speech fundamental period.

1] There is a long term definition, whereby T_0 is the period duration of a signal that is strictly periodic.

2] There is a short-term definition, in which case T_0 is due to the average elapsed time between successive excitations, somehow averaged over a specified short-term window.

3] There is a period-by-period definition, where T_0 is the elapsed time between two successive period markers.

Definition 1] cannot be applied to speech, because it is a quasi-periodic signal and this definition only applies for stationary signals. Definition 2] implies a short-term analysis of the speech signal, whereas 3] can be achieved by means of time-domain analysis of

the speech signal. In each case, the associated fundamental frequency F_0 to a fundamental period T_0 is defined as

$$F_0 = 1/T_0$$

3.2.2 Period-by-period or average measurements

Hess and Indefry (1987) discuss several basic approaches to estimating the fundamental period and fundamental frequency values of speech. Their analysis is as follows:

Method 1: Ideally an algorithm is required that can locate individual laryngeal cycles as accurately as possible. Such an algorithm will then be able to measure the natural fluctuations in vocal fold vibration. By detecting the "event" points of glottal closure it is possible to generate cycle-by-cycle fundamental period estimates, that are the times between successive points. In this case the period estimates are in correct phase; that is to say, a period is defined with its start located at one excitation point an its end at the next excitation time. Most algorithms operating on the acoustic speech waveform are unable to perform this function. However, laryngograph-based analyses can quite easily follow this definition.

Method 2: The next best approach, in terms of retaining information concerning the excitation, is to use an arbitrary repetitive point in the speech waveform and calculate the successive period spacing between these points. Most time-domain fundamental period estimation algorithms operate in this manner. This again leads to period-by-period measurements. In this case, the repetitive point may not correspond to the point of excitation in the speech waveform, and its location relative to the excitation point may change depending upon the wave-shape. Consequently, the period estimates may not be in-phase with the excitation points, as there were in the previous case.

Method 3: Another method involves determination of the average length of several successive periods. This operation is implicitly carried out by algorithms that use short-term analysis, such as auto-correlation. The inherent smoothing with this approach

results in the loss of fine perturbations in the fundamental period values that occur in speech.

Method 4: Finally fundamental frequency can be determined from a short-term frequency representation of the signal. Again, the window of analysis is required to contain at least one period, which gives a minimum window of around 20ms. The detailed nature of the method varies from technique to technique. This approach also results in smoothed frequency estimates.

3.2.3 The perception of spectral and virtual pitch

There is now a brief discussion of the human perception of pitch. This section is included because it is the limitations of the human auditory system and the perception of pitch that provide the ultimate limit on the performance necessary for a speech fundamental period (or frequency) estimation algorithm for general use.

Pitch perception has been investigated by many researchers for a long time. Many of the earlier theories of pitch perception relate to stationary complex sounds. At present, little is known about the perception of non-stationary sounds with changing fundamental frequency of excitation.

In early research, it was believed that the fundamental harmonic played the dominant part in the perception of pitch. However, Schouten (1938) showed that the phenomenon of pitch perception is not only evoked by the fundamental harmonic (at least not over the range of normal speech), an that the pitch of a harmonic complex remains the same when the fundamental harmonic is removed.

In an attempt to explain this phenomenon, de Boer (1956) proposed that this is not due to non-linear reconstruction of the fundamental harmonic within the ear, and that the perception of pitch is due to a pattern matching process. Subsequent work developed this idea further. In these theories, each harmonic evokes a spectral pitch corresponding to its fundamental frequency. All the spectral pitches then contribute to an overall pitch.

This is knows as the residual periodicity (Goldstein, 1973) or the virtual pitch (Terhardt, 1974).

3.2.4 Some important models of pitch perception

Three models that represent different approaches to pitch perception are now described. All models are characterized by a peripheral analysis that is characterized by a frequency analysis and a stage in which low pitch is estimated. However, the final pattern recognition stage is different in each model.

1] Wightman's pattern transformation model (1973).
There are three stages of processing in this model. Stage 1 is a limited frequency resolution power spectrum analyzer which is an approximation to frequency analysis performed by the peripheral auditory system. Stage 2 consists of a Fourier transform, which is assumed to be realised by means of a specially wired network of neural elements. Stage 3 is then a pitch estimator that operates by finding the positions of maximal activity in the output patterns from stage 2.

2] Goldstein's optimal processor model (1973).
In this model the processor is believed to make an optimal estimate of fundamental frequency on the basis of the noisy representations of the harmonics that are resolved. Under the assumption that the input stimulus is periodic and that adjacent harmonics are present, the model calculates the harmonic numbers and makes use of this information to estimate the fundamental frequency.

3] Terhardt's learning matrix model (1974).
This model is centres on a learning matrix that uses spectral-pitch and lowest spectral-pitch cues as its input (the term spectral-pitch refers to an estimate determined from peak in the short-term spectrum of the signal). The model operates in two phases: The first is a learning phase, which is assumed to be part of the childhood learning process in which a subject acquires the ability to recognize speech. In this phase, the correlations between the two input signals make their impression on the learning matrix.

The second is the recognition phase, in which the learned system generates its pitch estimates. During this phase of operation, the previously impressed traces in the learning matrix can be evoked by similar input stimuli to provide a virtual low pitch. Any given stimulus generates an number of such virtual pitch cues and the strongest determined the final pitch estimate.

A single sinusoidal tone evokes a spectral pitch. A signal such as speech is not a single tone, but rather a complex tone. If we assume for the moment that it will have many harmonics, each of which has it associated spectral pitch. The individual spectral pitches due to the harmonics are then centrally combined to give rise to the sensation of virtual pitch. This is the perceptual equivalent of fundamental frequency.

A definition of spectral and virtual pitch based on a quote by Terhardt (1972a) is as follows:
A single sinusoidal tone evokes a sensation known as the spectral pitch, which is related to the greatest place of excitation in the organ of Corti. The spectral pitches due to the partials associated with a complex sound can be individually perceived by a subject, provided that he makes a conscious effort to do so, unless the difference in frequency between the partials does not fall below a certain level. In addition to the spectral pitches, a stimulus generally evokes a dominant global pitch. In the case of harmonic sounds, this corresponds to the fundamental frequency. This due to a completely different phenomenon to that which evokes spectral pitch, and is known as the virtual pitch.

3.2.5 The pitch of speech

For most purposes, it can be assumed that the pitch and fundamental frequency of speech sounds correspond to each other. However, this is only true if fundamental frequency is defined as the reciprocal of fundamental period. This definition of fundamental frequency only corresponds to the largest common divisor of the partials in the case of strictly periodic signals. The definition of pitch by Terhardt (1979b) provides a good way to combine the temporal properties of the stimulus with its

perceived pitch. He states that "The extraction of fundamental frequency is in some respect equivalent to extraction of virtual pitch. In a strict sense, however, the frequency which corresponds to virtual pitch, and the fundamental frequency defined as the largest common divisor of the partials) are in general not identical. ...Hence in the analysis of auditory signals such as speech and music actually the extraction of fundamental frequency is not the real aim but rather extraction of the frequency which corresponds to the virtual pitch".

3.2.6 Difference limens for changes in frequency

The smallest detectable change in the frequency of a stimulus is known as the frequency different limen (DL) for frequency change. For synthetic speech stimuli the fundamental frequency DL has a value of about 0.3% to 0.5% of the fundamental frequency for the fundamental frequency range of male voice; that is over about 40Hz-150Hz (Flanagan & Saslow, 1958). This is less than the difference limen for a pure tone within the same frequency range, which correspond to about 3Hz (Zwicker & Feldkeller, 1967).

Even if changes in fundamental frequency are audible, there are not necessarily linguistically significant. The DL for linguistic significance is an order of magnitude larger than the DL for audibility (McKinney, 1965). This is not that surprising if one considers the fact that if the change is important, then it makes sense that is should be easy for the auditory system to detect.

3.2.7 The precision of speech production

Hess (1983) states that unless the output from a speech fundamental frequency estimation algorithm is to be used in synthesis applications (in which case the result is presented to the ear), or for scientific investigations into vocal fold vibration, there is no need to estimate speech fundamental frequency to a higher accuracy than it can be produced by the vocal apparatus. Various researchers have carried out measurements of the cycle-by-cycle changes in location of the glottal pulses. Gill (1962) found that there are more variations in wave-shape than in length of the glottal excitation.

Lieberman (1963) found that for successive periods, there was a relative difference of more than 1% for 30% of all periods and there was a difference of more than 3% for 10% of the periods. Similar results were found by Hollein et al. (1973) and Horii (1979). Horii found that the mean value of the jitter (the absolute difference in time) between two successive glottal pulses had a value of 51 microseconds at 98Hz and 24 microseconds at 298 Hz. In addition, for 10% of the periods in the data used, the jitter exceeded 100 microseconds.

These perturbations in the excitation are large compared to the frequency DLs for steady-state stimuli, and are audible to a listener. They cannot be individually distinguished, but contribute to the sensation of naturalness (Schroeder & David, 1960). Their effect is quite different from that of quantization noise, as has been observed in the context of speech synthesis (Holmes, 1976).

3.3 PROBLEMS IN SPEECH FUNDAMENTAL PERIOD AND FREQUENCY ESTIMATION

3.3.1 Basic difficulties

The determination of speech fundamental frequency is a difficult problem for many reasons. Speech is a non-stationary signal. That is to say, its characteristics change greatly as a function of time. One reason for this is that the shape of the vocal tract can change rapidly even within the space of a single fundamental period. In addition, the vocal tract can give rise to a wide variety of speech sounds, with a multitude of different temporal structures. The glottal excitation of the vocal tract is often only quasi-periodic. This is particularly true in the case of creaky voice. In addition there are acoustic interactions between the excitation from the vocal folds and the vocal tract.

3.3.2 Requirements for fundamental frequency estimation algorithms

There have been many suggestions as to how the ideal fundamental frequency algorithm should perform (Rabiner et al., 1976). It must be free from gross errors, which occur

when the frequency or period estimates deviate substantially from their true values. It must be able to retain the irregularity that exists in the vocal fold vibration. The fundamental period or fundamental frequency values should be as accurate as possible. The algorithm must be able to respond rapidly enough to changes in the excitation period. There should be no voicing determination errors. The measurements should be robust over different speakers, noise and environmental conditions. The algorithm should ideally require as little computation as possible, because this makes it easier (and possibly cheaper) to implement in real-time and for non-real time applications it will need less computer time to run (although this is becoming less important as time goes on, because of improvements in computer technology).

The requirements for a fundamental frequency or period estimation algorithm are all dictated by characteristics of the speech production, speech perception, and the particular application for which the algorithm is intended. The human ear is capable of detecting sounds over a wider frequency range than the vocal apparatus can produce, and can detect changes in frequency that are far smaller than the smallest frequency perturbations that a speaker can intentionally generate.

3.3.3 Sources of gross errors in fundamental period and period estimation

There are various reasons why a particular algorithm may generate gross errors. Firstly, when there are adverse signal conditions, which can occur when there is a strong first formant, a rapid change in articulator positions or in the case of band-limited or noisy speech. Secondly, when there is inadequate algorithm performance, perhaps because the analysis window is too small in a short-term algorithm, or because of the absence of some feature used in the estimation process. Thirdly, because the algorithm is unable to deal satisfactorily with creaky voice. In this case, the inherent averaging in some algorithms may cause erroneous output to be generated.

In addition difficulties can arise due to the recording conditions. Quite often the speech signal is degraded by amplitude and phase distortions, and background noise is almost always present to some extent. It is particularly difficult to get algorithms to operate

well over telephone lines, because of phase and amplitude distortions, fading, and break-through from other signals.

Strong first formant in vicinity of second harmonic

Gross errors can arise when there is a strong first formant in the vicinity of the second harmonic, which results in its amplitude becoming significant or greater than that of the fundamental harmonic. This can lead to what are known as "doubling" errors, because this leads to a significant second peak in each period, which time-domain algorithms sometimes confused with the main peak. This is illustrated in figure 3.2. For comparison, a temporally simple speech pressure waveform is shown in figure 3.3. Frequency-domain and short-term algorithms face a similar problem with this class of signals, because the second harmonic dominates the short-term spectrum. In this thesis, gross errors that exceed the true values are known as chirp errors.

The complementary type of errors to chirp errors are defined in this thesis as drop errors. In a time-domain algorithm they will occur whenever it misses out a period marker, giving the impression that the period is longer than it truly is. This situation can arise when there are rapid envelope changes in the speech waveform, and it is especially associated with voiced sounds made with articulations that result in obstruction of the vocal tract, such as the sound /r/. It can also occur due to missing secondary excitations in creaky voice quality speech or during diplophonic voicing (which is the tendency to generate pairs of pulses that can occur during even normal voice).

3.3.4 The required operating frequency range

The range of possible fundamental frequencies for human speech is wide. For an arbitrary utterance, the range over a large population of subjects can lie between 33Hz to 3100Hz by Moerner, Fransson & Fant, 1964. However, another investigation due to Catford,1964 (that did not include creaky voice) confined the range to between 70Hz and 1100Hz. For the purposes of singing, a somewhat wider range is required. Hess,

1983 gives the range of 50Hz to 1800Hz to cover a bass to a soprano.

For an individual speaker, the distribution of fundamental frequency depends upon the experimental conditions. It is particularly relevant whether the speech was taken from conversation or from read text. The frequency distributions from read text rarely exceed an octave range. Provided the distribution is plotted on a logarithmic scale, this fundamental frequency distribution comes close to a normal distribution (Risberg, 1961; Schultz-Colson, 1975)

Algorithms that perform speech fundamental frequency estimation usually restrict their operation to a sub-range of the possible fundamental frequency values. A good working range for an algorithm is between 50Hz and 800Hz, because this covers the range of most adult conversational speech (Hess, 1983).

3.3.5 Required measurement resolution and accuracy

The accuracy and resolution requirements for a fundamental frequency algorithm are determined by its intended applications. The human auditory system is more sensitive to changes in absolute frequency at low frequencies, and in general the noticeable difference in frequency is proportional to frequency. The difference limen with respect to the fundamental frequency (DL) for human listeners perhaps represents the ultimate required performance, which is typically 0.3-0.5% resolution of the fundamental frequency for steady state harmonic sounds. Most algorithms do not meet this specification. However, for most applications, less accuracy can be tolerated.

The difference limen for linguistic significance is greater than for that of perception (McKinney, 1965). Thus for prosodic analysis, an accuracy of a few percent may be adequate.

The required frequency (or time) resolution required is dependent upon the required application of the algorithm. For intonation training, a resolution of 3-4% will suffice (for example in a Voicscope, Abberton & Fourcin, 1973). There are also limits on the

resolution of fundamental frequency values that can be displayed with such schemes, due to the limited number of pixels available for the graphics display.

Consideration to human frequency difference limens suggest that a frequency resolution of 0.3%-0.4% of the fundamental frequency value would be ideally required by a fundamental frequency or period estimation algorithm.

Requirements for profoundly deaf EPI patients

The required frequency resolution for the profoundly deaf patients for whom high technology signal processing hearing aids are intended is only about 1% of the fundamental frequency values within the male frequency range and poor above about 200Hz, which is several times worse than for normal listeners.

3.3.6 Accuracy limitations due to time quantization of sampled signals

There is an intrinsic accuracy limit in time-domain fundamental frequency estimation algorithms that operate using sampled digital signals which is due to the time quantization of the input signal. This introduces uncertainty into the location of an event in time. For example, at a sampling frequency of 10kHz, it is only possible to locate a time event to 1/10000 = 100 microseconds. For a fundamental frequency of 100Hz, this corresponds to an accuracy of 1%. At higher fundamental frequencies, this percentage error increases still further. Even at 100Hz, this error is greater than the auditory DL for frequency change. The same problem arises for short-term analysis algorithms that operate in the lag domain (for example auto-correlation, cepstral analysis, etc).

There is a similar problem in the case of frequency-domain analyzers. In this case, a sampling rate of 10kHz and an analysis window of 100ms (which is very long for the short term analysis of speech) gives rise to a frequency resolution of 10Hz. Consequently, in this case it is the lower frequencies that give a proportionally larger quantization error. Thus there is a 10% error at 100Hz, and a 2% error at 500Hz. With

regard to this accuracy issue, Hess and Indefry point out (1987) that to reduce sampling accuracies to 0.5% up to the fundamental frequency of 500Hz requires a sampling period of 10 microseconds.

Many algorithms use interpolation at their outputs to improve the time or frequency resolution of their estimates. Interpolation can easily be carried out in the case of frequency-domain algorithms and those employing short-term analysis. Interpolation is more difficult to use in time-domain algorithms, although the accuracy of location of peaks and zero-crossings can be increased using interpolation. Another approach to reducing quantization errors is by smoothing the frequency estimates, although this approach is not always guaranteed to improve accuracy.

3.3.7 Required maximum rate of change of speech fundamental period

In regularly excited speech (not creak), the maximum rate of change of period length is typically taken to be a 10% to 15% change between successive periods (Reddy, 1967).

The maximum rate of change of frequency of the normal voice source was found to be about 1% per millisecond by Sundberg (1979). However, in voice qualities such as creaky, as well as in pathological speech, there can be much larger change per period than this figure suggests.

The maximum rate of change on fundamental period usually presents no problems to time-domain analyzers, because they operate on a period-by-period basis. However, they do put an upper time window limit on short-term analysis procedures of around 20ms -30ms.

3.4 CATEGORIZATION OF SPEECH FUNDAMENTAL FREQUENCY ESTIMATION ALGORITHMS

3.4.1 Preliminary classification

McKinney (1965) states that a 'pitch' determination algorithm can be essentially decomposed into three stages. These are the pre-processor, the basic extractor and the post-processor, as illustrated in figure 3.4. The main task of the measurement is performed by the basic extractor stage. The main function of the pre-processor is one of data reduction, and the emphasis of features in the input speech to facilitate the operation of the basic extractor. The post-processor combines many functions, such as error correction and the generation of output in the desired format.

3.4.2 Types of algorithm

The techniques that have been developed to determine speech fundamental frequency are broadly classified into four main groups by Hess, 1983; Those that operate in the time-domain, those that operate over some short-term window of the speech, which he calls short-term analysis, those which are hybrids of the first two, and finally those that operate by direct measurement of vocal fold activity. The is often no clear-cut distinction between the first two types. It is important to understand what is meant by the terms short-term, time-domain and frequency-domain.

Time-domain algorithms employ direct measurements on the speech signal and involve looking for temporal features in the speech pressure waveform (or in the filtered waveform), such as local maxima and minima.

Short-term analysis procedures use some form of transformation of the data within a short (for example, 20ms) time window. The nature of the transformation depends on the particular method used. The estimate obtained with such an approach consists of a sequence of average fundamental period or frequency values obtained over the input interval.

Frequency-domain algorithms make explicit 'frequency' estimates. There may be a frequency-domain interpretation to certain short-term operations which are implicit. For example, the auto-correlation technique can be implemented via a frequency-domain representation.

The time-domain refers to analyses which use the same time base as the input speech signal. A time-domain analyzer gives rise to an output signal that consists of a series of excitation markers that delineate period boundaries. Time-domain operation thus generally presumes the local definition of fundamental period and gives rise to a period-by-period fundamental period estimates.

The next chapter will examine some time-domain, short-term and laryngeal algorithms in more detail.

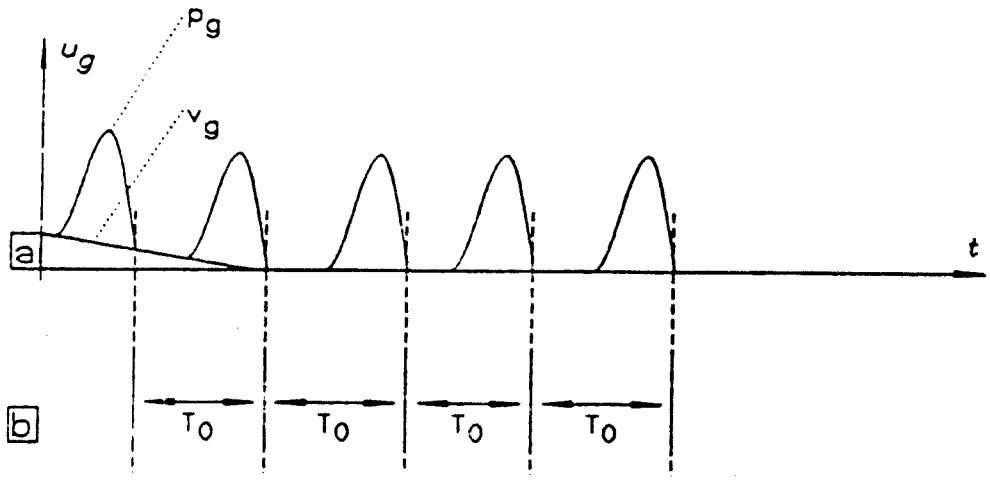


Figure 3.1 Diagram showing voice source parameters.
This illustrates; a) the excitation signal, and b) the corresponding period durations. (After McKinney, 1965).

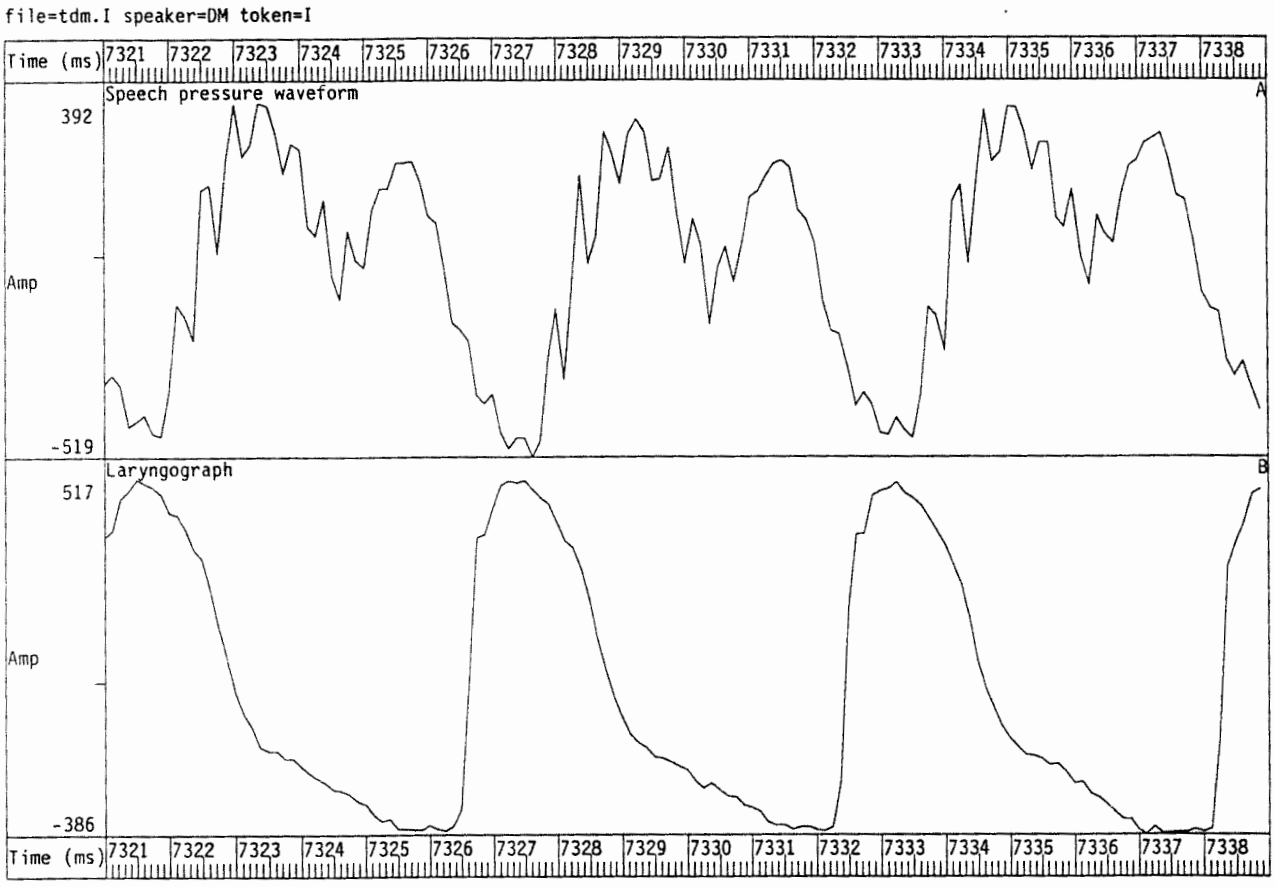


Figure 3.2 Speech pressure waveform exhibiting two peaks per fundamental period. The speech is shown in trace A. The corresponding laryngograph waveform in shown in trace B. This situation arises when the first formant coincides with the second harmonic in the excitation spectrum. This situation can lead to "doubling error" in simple fundamental period estimation algorithms. The speech is the vowel /I/ from a male subject.

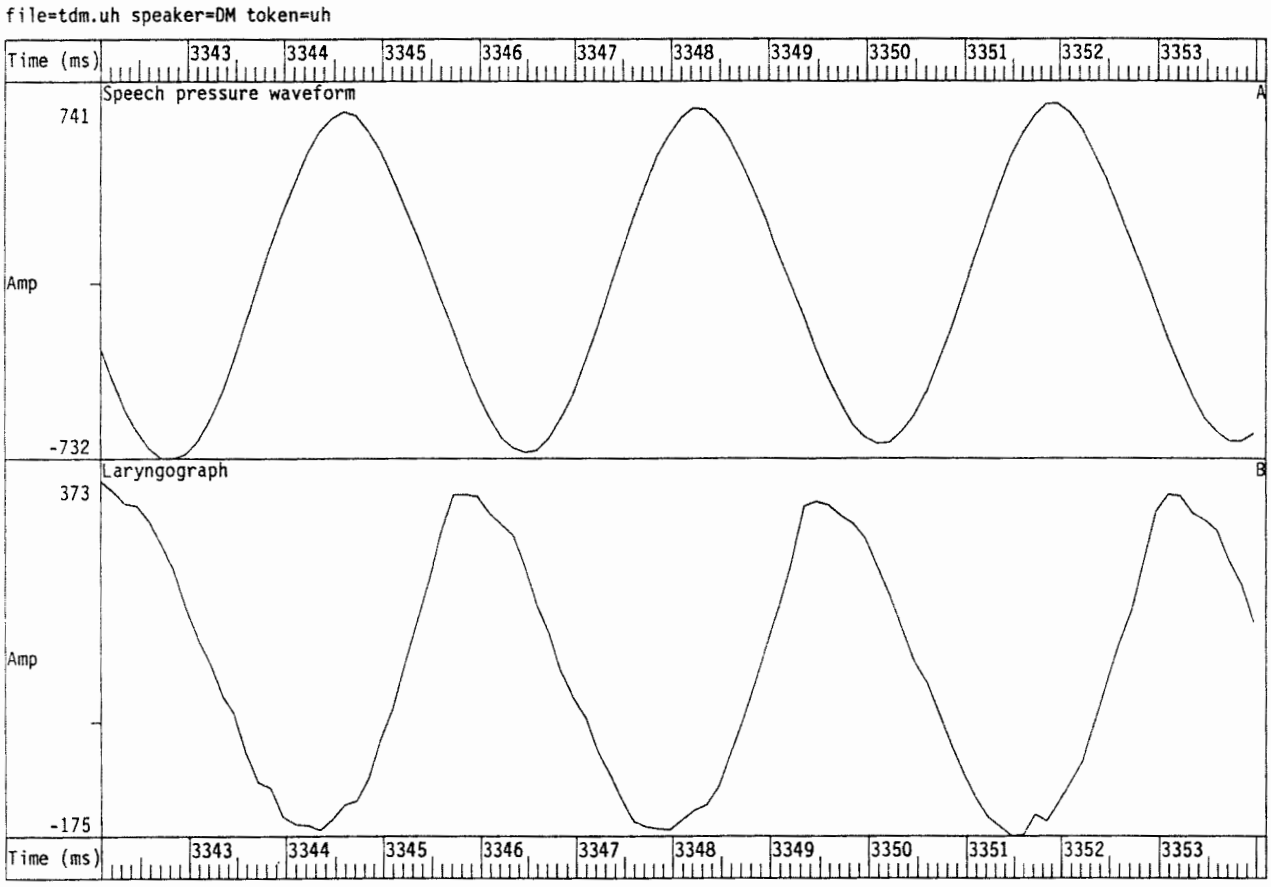


Figure 3.3 Temporally simple speech pressure waveform.

Speech is shown in trace A. The corresponding laryngograph waveform in shown in trace B. It is relatively would be easy to determine the fundamental period of the speech in this case, even with a simple fundamental period estimation algorithm. The speech is the vowel "u", as in the word "but", from a male subject.

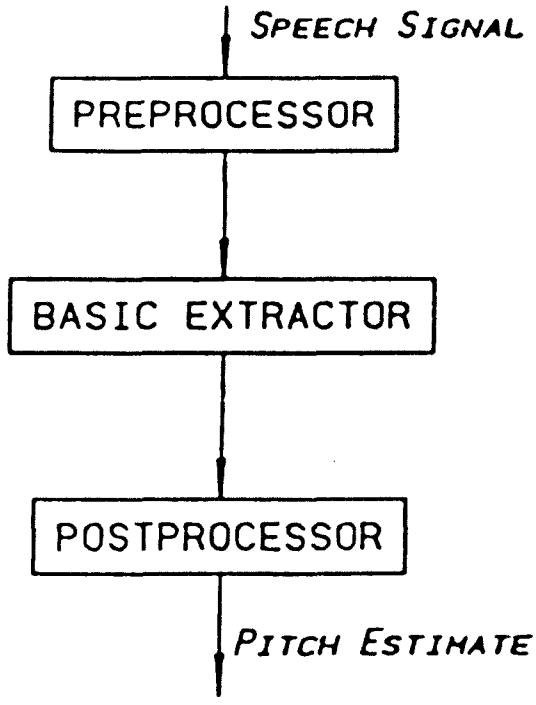


Figure 3.4 Block diagram illustrating the basic stages involved in speech fundamental frequency/period, estimation.

The pre-processing stage is involved with data reduction and extraction of important features of the speech signal. The basic extractor essentially performs the main task estimation of period or frequency. Finally, the post-processing stage converts the output from the basic extractor into a desirable format and may also perform error correction and smoothing of the raw estimates.

(Taken from Hess, 1983; After McKinney, 1965).

CHAPTER 4: ESTABLISHED METHODS OF SPEECH FUNDAMENTAL FREQUENCY/PERIOD ESTIMATION

This chapter discusses the different approaches to speech fundamental period estimation. These are described under the different section headings of time-domain algorithms, short-term analysis algorithms and laryngeal devices, and some examples of particular types of each method are given. Because of the very large number of different methods that exist, they are only discussed at length to illustrate particular principles, if they are well established, or are used later in this thesis in comparison tests.

4.1 TIME DOMAIN SPEECH FUNDAMENTAL PERIOD ESTIMATION

4.1.1 Introduction

The main strength of time-domain speech fundamental period estimation is its ability to make cycle-by-cycle measurements. This enables such algorithms to deal satisfactorily with irregular voice qualities (such as creaky voice) and with rapid changes in the frequency of vocal fold vibration. However, the price to be paid is that such algorithms can be more sensitive to noise. There are two main reasons for this. Firstly, the local key feature used by a typical algorithm, such as a signal peak, may be much more affected by noise than the gross overall waveform shape. In addition, the measurement is not averaged over several cycles, as is the case with short-term analyses. Of course, the estimates from a time-domain algorithm can be averaged afterwards (which if performed appropriately reduces noise), but this sacrifices the cycle-by-cycle estimation for greater noise resistance.

Hess (1983) states that periodicity manifests itself in the speech waveform in various ways. One possible feature is the presence of a fundamental. Another is the existence of a structural pattern that approximately repeats each period. This pattern often exhibits high amplitudes at its onset and lower amplitudes towards its end. This follows from consideration of the source-filter model of speech production in which the vocal tract can be considered as a passive linear system driven by impulse excitations (Fant, 1960).

This model implies that a speech period is composed of a sum of exponentially decaying sinusoids. Additionally it follows that there may be abrupt changes in the waveform of the speech signal (or its time derivatives) at the excitation points.

Time-domain fundamental period algorithms generate estimates in the form of a sequence of period markers that somehow utilize these characteristics. Hess (1983) sub-classifies time-domain algorithms in terms of the principle by which they detect the periodicity. These are shown in figure 4.1. One extreme approach to this involves the use of structural analysis. Algorithms of this class use a complex procedure to find a valid fundamental period from a set of basic measurements that are made directly on the speech waveform. The other extreme approach involves complex pre-processing of the speech signal to extract the fundamental harmonic, followed by a simple measurement and validation procedures on the temporally simplified waveform. Thus, in the former, case simple measurements are made and the burden of the task falls on a complex validation procedure that is then used to find the period estimate. In the latter case, the main burden of the work is allocated to the pre-processing whereas the validation procedures are relatively simple.

There are also algorithms that operate somewhere in the middle ground between there two extremes by using some temporal simplification and some analysis of temporal structure.

In addition there are also multi-channel algorithms, which typically involve the operation of a set of simple measurements on the output of each channel and use an overall control unit to select the appropriate output.

4.1.2 Fundamental harmonic extraction

The main characteristic of time-domain algorithms that employ first harmonic extraction is that they usually require the first harmonic to be present in the speech signal (unless it can be recreated by non-linear distortions within the algorithm). In addition they are sensitive to low frequency distortions of the signal.

In this class of algorithms, the first harmonic is first emphasised relative to the higher harmonics using linear and/or non-linear filtering. These systems fail drastically whenever the pre-processor does not sufficiently suppress the amplitudes of the higher harmonics. Under such circumstances, more than one period marker may be generated per speech period, giving rise to "chirp" errors (classes of errors are discussed fully in chapter 6).

After the preprocessor, the temporally simplified waveform is then fed to a low complexity basic extractor. A basic extractor is typically some kind of threshold analyzer. Such an extractor generates a marker whenever the pre-processed signal crosses a threshold (which may be zero). A slightly more sophisticated variant of this type of extractor makes use of hysteresis, such that a marker is only generated after two thresholds have been crossed in a given sequence. Figure 4.2 illustrates these basic extractor types.

To achieve an accurate period estimate, it is important for the basic extractor to locate the period markers accurately. For a non-zero threshold extractor systematic "fine" errors can arise due to the movement of a marker as the signal level changes, even if its frequency remains constant (Hess, 1983). Such fine errors can also arise during a formant transition, which can give rise to a phase change in the first harmonic (this is illustrated in figure 4.3).

Analysis of zero-crossing extractor to avoid gross inaccuracies

For the simple case of a zero-crossing basic extractor, McKinney (1965) gave an analysis showing the necessary relationship between the amplitude of the fundamental harmonic and the higher harmonics which must be maintained to avoid additional unwanted markers being generated. This is given by:

$$A(1) \Big/ \sum_{k=2}^{k=m} A(k) > 1$$

where the amplitudes of the harmonics are denoted A(m), and m is the harmonic number. This relationship corresponds to the worst case when the phases of the higher harmonics are all 180 degrees out of phase with the fundamental. To meet this equality, the speech must be suitably pre-processed before it reaches the basic extractor.

Linear pre-processing

Linear filtering can have the effect of reducing the amplitude of the higher harmonics in accordance with its amplitude frequency response. To meet the necessary amplitude relationship between the fundamental and other harmonics (for example, the second), a substantial amount of low-pass filtering of the input signal is required. In the case of (non-zero) threshold extractors, larger amplitudes for the higher harmonics can be tolerated than for the zero-crossing extractor. Hess (1983) claims one needs 6dB/octave less attenuation from the low-pass filter if a threshold analyzer with hysteresis is employed, and in such a case the low-pass filter must reduce the amplitude of the harmonics by about 18dB/octave. A suitable filter could therefore be constructed by cascading three integrators, each of which provides 6db/octave attenuation.

Unfortunately, problems arise with this type of pre-processing when a large operational frequency range is needed, because high frequency signals become so attenuated that the system must deal with a very large dynamic range. If the frequency range exceeds 3 octaves, this can give rise to a 54dB dynamic range variation solely due to changes in signal frequency, quite apart from the 30dB or so variation of speech that occurs anyway. An automatic gain control may be of some value, but it cannot overcome all of the problems associated with a large dynamic range. However this problem can be avoided if the analysis is restricted to a narrow range of operating frequencies (Baronin, 1974).

Tracking filters

One way around the difficulties encountered in sufficiently suppressing the higher harmonics in a speech signal for processing by a basic extractor is to dynamically select

appropriate fixed sub-bands, or by use of an adaptive filter that tracks the fundamental frequency.

One of the earliest and simplest algorithms that employed several sub-ranges was due to Dempsey et al. (1953), and it employed three low-pass filters with different cut-off frequencies. The appropriate filter was then selected manually. Other systems adopt the same basic approach, but use two sub-ranges, one for men and the other for women speakers (Riesz & Schott, 1946; Dawe & Deutch, 1955).

It is clear that the task of manually changing the operating band is an undesirable characteristic of these algorithms. To avoid this manual intervention, schemes have been developed that employ variable cut-off filters that can be automatically tuned to track the input fundamental frequency. There are two configurations of such schemes; those that operate in open-loop mode and those that operate in closed-loop mode.

A pre-processing scheme adopted by Peterson & Peterson (1968) was adopted to circumvent the problems of using a single low-pass filter in conjunction with a simple basic extractor. The speech is first optionally half-wave rectified and high-pass filtered and then fed into a bank of low-pass filters. Only the outputs from the low-pass filter with cut-off frequencies greater than the fundamental frequency are automatically selected, using the principle that the lowest frequency present corresponds to the correct period estimate, so that only the fundamental harmonic reaches a basic extractor.

There are various other algorithms that adopt the same basic approach as this one, such as the algorithms due to Hollein (1963) and Dibbern (1972). Care has to be taken with these algorithms so that the transients that occur as the different channels are selected do not interfere with the period estimates that are generated.

Open loop systems typically use a simple crude fundamental frequency estimator to tune the tracking filter in the main estimator (Peterson, 1952; Miller, 1953; Barney, 1958; Martynov, 1958; Yaggi, 1962). The problem with this approach is that errors in the crude estimate can result in failure of the system. Consequently with respect to gross

errors, the overall system only works as well as the crude estimator. However, when the main filter does track the fundamental correctly, the period can be very accurately estimated.

Closed-loop systems make use of the output from the main tracking filter to obtain an estimate of the frequency range, which is then used to alter the main tracking filter (Riesz, 1952; Feldman & Norwine, 1958; Sapozhkov, 1963; Yasuo, 1962; Pirogov, 1963; Peckham, 1979). Difficulties arise with this approach because there are often many stable states for the system, only one of which corresponds to correct operation. In particular the higher harmonics are sometimes mistakenly tracked instead of the fundamental. This leads to (chirp) errors that can persist for the duration of a voiced segment of the speech.

Non-linear pre-processing

Pre-processing is sometimes performed using non-linear functions which can sometimes have beneficial effects. For historical reasons related to ease of implementation, half-wave and full-wave rectification has received much attention in the past (McKinney, 1965). Full-wave rectification is effective in increasing the level of the first harmonic relative to the level of the higher harmonics, unless the signal is almost sinusoidal. In this case, the effect is unwanted because it effectively doubles the periodicity of the signal. Put another way, full-wave rectification is beneficial, unless it is not needed in the first place, in which case it is detrimental. Half-wave rectification does not double the periodicity of sinusoidal signals, but its effects on asymmetric speech signals is not generally beneficial.

4.1.3 Structural analysis

Another approach to time-domain fundamental period estimation involves the temporal structure of the speech waveform. This general approach is further sub-divided by Hess (1983) into the analysis of extrema and envelope modelling (as shown in figure 4.4).

Envelope modelling

The idea behind envelope modelling follows from the earlier observation that the vocal tract can be considered to be a passive linear system excited impulsively. Therefore, after each impulsive laryngeal excitation, the speech signal will be characterised by a set of exponentially decaying sinusoids. In schemes that adopt envelope modelling, the envelope of a speech period is considered to be a decaying exponential function that starts from a maximum value at each excitation point. After an excitation, the envelope of the model gradually decays away. When the speech signal exceeds the modelled envelope function, a new excitation is assumed to have occurred and the envelope model is reset ready so that it is ready detect the next excitation point. Much early work was carried out on algorithms of this type using analogue hardware (Vermeulen & Six, 1949; Gruenz & Schott, 1949; Dolansky, 1954,1955; Anderson, 1960; Filip, 1967,1969).

This class of algorithm is often implemented using a cascade of identical stages, each of which suppress the secondary peaks and enhance the principal peak (due to the excitation). Secondary peak suppression is performed using a peak detection circuit and primary peaks are then enhanced by subsequent differentiation. The main problem associated with this approach is to find the appropriate time-constants to fit the decay of the speech signal. If the time constant of the peak decay section is too short, then it will fail to suppress the secondary peaks. Conversely, if it is too long, primary peaks will be suppressed. This is particularly true when there is a rapid overall envelope changes in the speech which can give rise to a large principal peak followed by a smaller one.

Peak Picker

The peak-picker (Howard, 1986) is an example of a simple envelope modeler, based upon earlier work by Gruenz & Schott, 1949. A version of this algorithm used later in this thesis, is used here for comparisons, and it is a software implementation of a small battery powered device developed as part of the External Pattern Input (EPI) group cochlear implant prothesis, at University College London. With the use of algorithm

like this, that makes use of peaks in the speech pressure waveform, it is important to appreciate that the speech waveform is generally not symmetrical about zero (Anderson, 1960), and the biggest peaks are usually those that correspond to positive pressure. Therefore, to achieve the highest level of performance, the speech must be polarised such that the major peaks are positive going. For live operation from speakers, the polarity only needs to be set-up once for a given microphone, but recorded or broadcast speech may generally be of either polarity. Consequently for such operation, the polarity will need to be checked. This is typically carried out by using the peak-picker with and without speech inversion, and then selecting the polarity that gives the best results.

The operational waveforms in the peak-picker are shown in figure 4.5. Firstly the input speech is filtered, using a 4-pole Butterworth low-pass filter with a cut-off frequency of 450 Hz, which temporally simplifies of the speech pressure waveform by removing all but the first few harmonics. The next stage consists of a logarithmic amplifier. This helps minimise the effects of the large rapid amplitude variations in the level of the input speech. Next is the first peak-picker stage that consists of a function that has an output that follows the input only if the input is greater than the output. Otherwise the output decays linearly with time. This has the effect of suppressing secondary peaks in each cycle. A differentiator is then used to emphasize the discontinuities that occur at the onset of the primary peaks. The previous two steps are then repeated, to increase the suppression of any secondary peaks. If the algorithm has succeeded there is then only one peak left per period. A simple threshold is then used to locate these peaks. This threshold is typically set by iteratively adjusting the value and observing the output from the device. A cleaned-up pulse output is then produced using a monostable circuit.

The peak-picker operates on a period-by-period basis and is thus able to retain irregularity in the laryngeal excitation. In addition, the input to output delay is relatively small, making it well suited for real-time applications in pattern processing hearing aids. Its main weakness is that its performance in noise and reverberant environmental conditions is inadequate for many applications. The peak-picker is quantitatively compared against other algorithms in chapter 9.

Analysis of extrema

The analysis of extrema relies on what are effectively expert systems to validate a set of measurements of the speech signal which are chosen to characterise its temporal structure, and at the same time discard unwanted information. For example, maxima and minima are often used as temporal features of the waveform. These markers are then subject to tests to ascertain whether or not a given marker could constitute a period marker. The marker validation procedure often needs to operate over a large time window, which results in a long inherent delay between the input and output that makes such algorithms unsuitable for real-time operation.

Peak detection and global correction

One algorithm that employs analysis of maxima and minima is due to Reddy (1966,1967). This algorithm was originally designed to be incorporated in a speech recognizer and operates on blocks of 25ms duration. The first step involves locating the local maxima and minima within a block. These markers are then subject to a set of tests to eliminate markers that do not correspond to period markers. This involves the following: The absolute maximum in the block is first found. Maxima are then labelled significant if they are positive, if they are at least 2.5ms away from other significant maxima, if they are greater than 0.9 times the local absolute maximum, or if their amplitude is such that they lie above a linearly extrapolated line arising from the previous two significant maxima. If both of the last two tests fail, then a maximum may still be labelled significant if it constitutes a local maxima over 13.5ms. A similar procedure also applies to the definition of significant minima. The next step is then to label a maximum as significant a peak only if a significant minimum lies within 3.5ms. These significant peaks usually correspond quite well to period markers, but there are occasions where mistakes occur. Reddy then employs a global error correction. This algorithm operates by estimating the regularity of the markers and removing and adding them whenever necessary. This is achieved by predicting the current period on the basis of the past few periods. If the current period is greater or less than this by 12.5% then there is assumed to be an error. However, only certain classes of errors can then be

corrected. For example, if there was another maxima near the principle peak, this can sometimes be the valid marker that was originally missed, and changing the period marker to this one corrects a "hop" error (an error due to a misplaced marker). Also, it is sometimes possible to remove extra markers which give rise to a "chirp" error (an error due to an extra unwanted marker). Similarly it is sometimes possible to insert another marker at a previously discounted maximum to avoid a "drop" error (an error due to missing a marker). Unfortunately, when the global correction routine has to deal with too many earlier errors, it can fail totally (Liedtke, 1971).

Pitch Chaining

Another algorithm that operates by direct analysis of the structure of the speech pressure waveform is due to Schaefer-Vincent (1983) and is known as pitch chaining. It operates by first extracting a skeleton of maxima and minima values from the time-sampled speech pressure waveform. All possible combinations of three maxima and all possible combinations of all minima are then subject to analysis by an expert system, to determine whether or not they represent the markers that define two adjacent period values (see figure 4.6). The tests include bounds on period values, a bound on the ratio of the period values and bounds on the relationship between the magnitude of the extrema. If the extrema pass the test conditions, they are then considered to be valid period twins. The algorithm then attempts to fit together the latest period twin with ones previously found to form a series of "chains". If a new period twin cannot be added onto an old one, which will happen if there are no old ones, or there is no coincidence in the period values, a new chain is started. Thus the "chains" correspond to the different ways in which the period twins can fit together. Consequently, a chain will generally have many branches from it, corresponding to possible different sequences of period twins. When a chain exceeds a preset length, its fundamental frequency values calculated and final values are output as frequency values in 10ms frames, and all earlier chains are deleted. It is to be noted that the final stage in this algorithm involves frequency averaging. Consequently this algorithm does not give cycle-by-cycle estimates of fundamental frequency, although it could - but not without a long delay.

Not all structural analysis algorithms use maxima and minima as primary operational features. For example, the algorithm due to Miller (1974, 1975) employs what he calls the excursion cycle, which is defined as the sum of all the samples between two consecutive zero-crossings of the signal. These markers are then subject to tests similar to those used by Reddy (1966, 1967) described above.

Gold-Rabiner Algorithm

Gold (1962) observed that algorithms that are based on the regularity of the signal tend to fail whenever the signal becomes irregular, such as when there are rapid changes in fundamental frequency or sound quality, or with irregular voice qualities like creak. Conversely, algorithms based on peak detection tend to fail when the speech does not exhibit strong peaks, which can happen, for example, in the case of nasals, back vowels or speech with falsetto excitation. Gold claimed that in order to achieve good performance with a structural analysis algorithm on both of these kinds of signal, it was better to use more than one "rule" to interpret the extrema of the speech. A first attempt to implement this principle was carried out using three individual basic period extractors that operated in parallel and whose outputs are then fed into a relatively simple combiner algorithm (Gold, 1962). A more sophisticated version of this algorithm, which is possibly the best known analysis of the structure algorithm, was developed by Gold and Rabiner (1969). The schematic diagram for this algorithm is shown in figure 4.7.

The first stage consists of pre-processing using a low-pass filter with a 900Hz cutoff frequency. This has the effect of reducing the higher formants which can impair the accuracy of marker placement. The second stage results in the generation of a set of six pulse trains which depend on the maxima and minima of the pre-processed waveform; their relationship to the waveform is shown in figure 4.8. This results in the generation of pulses of height m_1 and m_3 at the positive peaks, pulses of height m_4 and m_6 at the negative peaks, and a peak-to-peak measurement pulses of height m_2 and m_5. This set of measurements were arrived at by the authors by considering two extreme cases of input waveforms, as shown in figure 4.9. The first of these is a pure sine wave. The second is a waveform that contains a fundamental and a strong second harmonic.

It can be seen that in the case of the sine wave, measurements m_1, m_2, m_4 and m_5 exhibit the appropriate periodicity, whereas in the case of the other waveform only m_3 and m_6 are appropriate.

These six pulse trains are then fed into six identical simple period estimators. These operate by following peaks for a holding interval, and then exponentially decaying, as shown in figure 4.10. The time constant and holding times are both adapted on the basis of previous period estimate from the given detector.

The six period estimates from the six basic detectors are then subject to an evaluation procedure. This is achieved by first forming a 6X6 matrix, as a function of time from the period estimates, with the columns representing the individual detectors and the rows representing the period estimates. The row represent the direct estimates from the basic extractors. Rows 2 and 3 are the estimates from the previous two periods, whereas the other three rows represent the sum of estimates from the first and second, the second and third and the first and third rows respectively. The reason for including the sums is that the individual detectors are biased towards generating output periods due to the second and third harmonics and under these conditions it is the last three rows that will give the correct estimates. The coincidence of values in the matrix then calculated, and the value that occurs the most frequently is taken as the period estimate.

Another similar algorithm is due to Tucker and Bates (1978). In this algorithm, the speech is first subject to centre clipping (which involves setting small amplitudes to zero and is discussed later in conjunction with auto-correlation) which has the effect of reducing the effect of the formant structure whilst maintaining the periodicity of the signal. (Sondhi, 1968). Each peak that survives this process is then characterised in terms of five features, which are then used to form an overall period estimate in a similar way to in the Gold-Rabiner algorithm.

4.1.4 Simplification of temporal structure

In between the two extreme approaches of extraction of the fundamental harmonic and

direct structural analysis of the speech waveform, one can adopt a compromise and make use of both principles; perform a degree of temporal simplification and then perform a simpler structural analysis of the waveform. Hess (1983) sub-classifies such schemes into those that perform inverse filtering and those that perform epoch detection, as shown in figure 4.11. The idea behind inverse filtering is that by passing the speech waveform through a filter that constitutes the inverse response of the vocal tract (the inverse filter), one can estimate the excitation signal. Epoch detection, on the other hand, relies on the detection of the discontinuities in the differential of the speech waveform that occur whenever the vocal folds snap together.

Inverse filtering

According to the source filter model for speech production (Fant, 1960), one can consider speech production to be due to the convolution of the excitation and the impulse response of the vocal tract, that is:

$$x(n) = p(n) * h(n)$$

where x(n) represents the sampled speech signal, p(n) represents the sampled excitation time function and h(n) represents the vocal tract impulse response. In the frequency-domain (the z-domain) this becomes;

$$X(z) = P(z).H(z)$$

where X(z) is the z-transform of the speech waveform, P(z) is the z-transform of the excitation and H(z) is the z-transform of the vocal tract impulse response. From this equation, it can be seen that if the z-transform of the speech is divided by the z-transform of the vocal tract response, the result will be the z-transform due to the excitation. Therefore, if the inverse filter 1/H(z) can be found, it can be used to remove the effects of the vocal tract from the speech waveform.

Fant (1970) stated that for vowels and sounds that are not nasals, the vocal tract transfer

function H(z) can be modelled using an all-pole filter, and this implies that the inverse filter is an all-zero filter with a state equation of the form;

$$y(n) = d_1x(n) + d_2x(n-1) + ... + d_kx(n-k)$$

There are various ways to determine the coefficients of the inverse filter. However, conventional methods of formant analysis (for example involving spectrographic analysis) require much effort (Flanagan, 1972). A popular approach is based upon the technique of linear predictive coding, or LPC analysis as it is often known. This technique proposes that one cam predict the current sample x(n) of a signal from the past signal to within a certain error limit e(n), that is

$$x(n) = a_1x(n-1) + a_2x(n-2) + ... + a_kx(n-k) + e(n)$$

The form of this state equation is the same as that of the inverse filter. For this to constitute a useful model of speech production, it is important that the error term be small and the coefficients a_1, a_2, ..a_k be known. Because the vocal tract alters its response as a function of time, a given set of coefficients are only relevant over short-time intervals, and consequently must be estimated using short-term analysis. The optimization of coefficients is performed by minimizing the energy of the error signal over a frame (which is typically 10-30ms in length). This involves formulating the error as a function of the predictor coefficients and then solving a set of linear equations (Markhoul, 1975; Markel & Gray, 1976). Solving the predictor equation for the error gives;

$$e(n) = x(n) - a_1x(n-1) - a_2x(n-2) - ... - a_kx(n-k)$$

and hence provides an estimate of the inverse filter. If the predictor was able to predict completely the input signal x(n), the error signal would always be zero. It follows from linear filter theory that in this case the speech signal would consist of a sum of decaying sinusoids. Of course, this situation only arises in speech production between excitation points (the only time the system is stationary). Since the analysis assumes impulsive

excitation, there is a peak in the error signal at each excitation point corresponding to the excitation (which can be used to locate the excitation point).

The LPC residual signal is directly used by some time-domain fundamental period algorithms (Atal & Hanauer, 1971; Strube, 1974). The LPC prediction error for some vowels is shown in figure 4.12. In addition, it is also used as a pre-processed input to short-term algorithms, such as the SIFT algorithm (Markel, 1972) that is discussed in a later section.

There are various problems associated with inverse filtering. The LPC analysis is carried out by performing a minimization of the error signal. Unfortunately, this minimization does not always preserve the excitation signal, which can sometimes also be lost by the procedure (Gold, 1977). Another problem occurs when the first formant coincides with the fundamental harmonic in the signal. In this case, by removing the effect of the formant there is a tendency to cancel out the fundamental and produce an inverse filtered waveform without it.

Epoch detection

The term epoch was first introduced by Ananthapadmanabha & Yegnanarayana (1975, 1979). The idea behind epoch detection is that it is possible to detect isolated events or "epochs" that arise at the moment of vocal fold closure. The first task of this class of algorithm is therefore to emphasise features which correspond to the vocal fold closure points in the speech signal.

One of the first algorithms of this kind was due to Smith (1954, 1957). The initial pre-processing employed a bank of 32 second order bandpass filters, the outputs from which are full-wave rectified and smoothed. This has the effect of performing amplitude demodulation of the different frequency bands by means of envelope detection. The smoothed outputs are then summed. There is sufficient phase coherence between the different channel envelopes because all the resonances of the vocal tract are simultaneously excited each time the vocal folds snap together and consequently they

add synchronously after the excitation point and then decay away. The point of inflection of this resulting waveform is then used to mark the onset of successive periods.

This system was also implemented by Yaggi (1962,1963) using a channel vocoder. In this case 18 channels were used ranging from 70Hz to 4kHz, employing bandwidths between 130Hz to 390Hz for the lowest to highest channels respectively. A block diagram of this system is shown in figure 4.13. The channel outputs were again full-wave rectified and first order low-pass filtered. Figure 4.14 illustrates their outputs.

This scheme still operates if the fundamental harmonic is absent, and the coincidence of the first formant with the fundamental harmonic which can cause the failure of inverse filtering similarly presents no difficulties.

A slightly different algorithm is due to Rader (1964) using Hilbert filters rather than envelope detectors.

The algorithm due to Ananthapadmanabha & Yegnanarayana (1975, 1979) functions by detecting the discontinuity in the speech waveform that occurs as the vocal folds snap together. They define an epoch as follows:

"Let f(t) be a function defined over the interval (a,b), and zero outside the interval. Also let f(t) posses continuously differentiable derivatives in the interval (a,b). Then the point of discontinuity of the lowest ordered derivative will be regarded as an epoch..."

The response of this system to speech is shown is figure 4.15. Problems arise in practice when there is more than one epoch per period. In addition, the algorithm does not function well when the input signal has weak discontinuities, such as in the case of falsetto voice and voiced fricatives.

4.1.5 Multi-channel analysis

Many algorithms, including some that have already been discussed in this chapter, are examples of multi-channel analyzers. We shall consider three types of multi-channel analyzers (after Hess, 1983).

1] Main channel and auxiliary channel principle. In this configuration, a crude auxiliary channel is used to adapt the operation of a main channel. This is the approach adopted by open-loop tracking filter systems, where the auxiliary channel sets the context for the operation of a more accurate main period estimator.

2] The sub-range principle. In this configuration, there are several similar or identical algorithms that are individually optimised for operation over different frequency sub-ranges. Their outputs are then somehow combined together to generate an overall estimate.

3] The multi-feature principle. In this type of system, the channels are independent and each either process different parameters of the input signal, or process similar features in different ways. Not all parts of such a system have to be independent. For example, common pre-processing may be employed. Of course, there must again be a common data fusion stage, where the different signals are combined together to estimate the fundamental period value.

There are two main problems with multi-channel algorithms. The first is how the results from the separate channels are combined together. In most cases, at a given time all channels will generate some kind of period estimate, although not all will be useful. Quite often some basic decision rule is applied, such as the selection of the estimate that corresponds to the longest fundamental period, or by choosing the period estimate that has the highest number of occurrences.

The other problem relates to how the period markers obtained from different channels should be synchronized, because markers from different channels can differ in phase (that is to say, a different channel may use a different point in each speech cycle to mark the period). This phase relationship between markers from different channels must

be taken into account. This problem becomes simple if one does not require the phase information, in which case some kind of averaging of separate channel estimates can then be used.

4.2 SHORT-TERM SPEECH FUNDAMENTAL FREQUENCY ESTIMATION

4.2.1 Introduction

Short-term fundamental frequency estimation algorithms differ from time-domain algorithms in that some kind of transformation is applied to the speech signal, and it is this that is subject to measurement to estimate the fundamental frequency or fundamental period.

As a result of the way in which evidence is combined over the observation window, short-term approaches give rise to fundamental frequency estimates that correspond to the average value over the analysis window. Therefore short-term analysis techniques are unable to perform estimation on a period-by-period basis, and they generally do not make use of phase information. This does have the advantage that they are not sensitive to phase distortions that may adversely affect some simple time-domain techniques. In addition, because they make use of evidence from all the data within the input window, such techniques are generally robust in the presence of unwanted noise and signal corruption. They often require substantially more computation than time-domain techniques, although with fast modern digital computer technology this does not constitute as much as a problem as it did in the past.

Short-term analyses include spectral analysis of the speech signal, which operate by transforming the input to the frequency-domain, and lag-domain analyses, such as auto-correlation. A sub-classification of short-term analyses is shown in figure 4.16.

4.2.2 The principle of short-term analysis

As previously stated, speech is a non-stationary signal. Principal characteristics of the

speech signal, including its periodicity, change as a function of time. Consequently, in order to usefully estimate such parameters, it is their momentary values that are of interest, rather than their long time average values. The short-term value xs(n,q) of a sampled signal x(n) is typically estimated by first multiplying a section of the signal with a time-function w(k) known as a window.

$$xs(n,q) = x(n) \,.\, w(n-q)$$

The window function, represented here as w(k) is non-zero only over a short interval. For example, in the case of a rectangular window:

$$w(k) \quad \begin{cases} = 1 \text{ for } k = -k/2 \text{ to } k/2 \\ \\ = 0 \text{ otherwise} \end{cases}$$

The analysis is then carried out on the windowed signal xs(n,q) for all the data points q for which the result is required. The parameter extracted at point n is known as a frame.

The window length k is selected to be long enough so that there are sufficient samples within the window to reliably estimate the parameter, and short enough such that the parameter does not change too much over the window. To ensure several periods (at least two) fall within a frame at the lowest frequency, a window around 20-50ms in duration is typically used. The shape of the window also influences the results obtained from the analysis, since the multiplication of the input signal by a window has the effect in the frequency-domain of convolving the frequency response of the window with the spectrum of the signal. A full discussion of such issues appear in several textbooks (Rabiner & Schafer, 1978; Oppenheim & Schafer, 1975; Hamming, 1980)

Characteristics of short-term analyzers

The basic operational steps in different kinds of short-term analysis algorithms are

generally similar. A typical processing scheme is shown in figure 4.17, in this case for a frequency-domain analyzer. Sometimes the signal is initially pre-processed to reduce its temporal complexity (using low-pass filtering, centre clipping or inverse filtering). Then the signal is divided up into time frames and the specified short-term transformation is then carried out on each frame, the effect of which is to generate a signal containing a peak or peaks, the location(s) of which are determined by the fundamental period or frequency. In addition, the degree of periodicity is usually related to the height of the peaks. The next step involves using some algorithm to identify the location of the peaks. The peak height is often compared with a preset threshold so that a voiced/unvoiced decision can be made. Finally, there is sometimes a post-processing stage that may perform interpolation around the peaks to improve their location accuracy and attempt to correct any errors that were made.

Problems with irregular speech excitation

Because the function of a short-term algorithm is essentially to detect the periodicity of the input signal, these algorithms run into difficulties when the signal contains irregular periods. There are two ways in which an algorithm can fail under these conditions.

1] If the analysis frame is short, then a frame may only contain irregular periods (this is also the case if the signal is all irregular) and consequently the algorithm may consider the speech unvoiced, and generate no period estimate.

2] If the analysis frame is wide, then the contribution a few irregular periods to the analysis will be small and consequently there will be little or no indication of irregularity in the frame estimate. Although this does not give a good indication of what is really happening in the signal, this situation is still preferable than a failure to detect voicing.

Computational considerations

The majority of the computational effort with this type of algorithm is taken up by the

short-term transformation. Hess (1983) stated that most short-term transformation of an input vector **x** (containing all the input signal samples within a frame) can be viewed as a matrix multiplication with a transformation matrix **W** that results in an output vector **X**; that is

$$\mathbf{X} = \mathbf{W}\mathbf{x}$$

One important observation concerning this operation is that the computation will, in the general case using a direct implementation, increase with the square of the number of samples in the input vector. This is a good reason for keeping the input window as small as it can be. However, in the case of spectral transformations the number of samples in the input window determines the resolution of the output frequency estimates; the fewer the input samples, the lower the spectral resolution. To circumvent this trade-off, it is quite often better to perform output rather than input interpolation to increase the resolution of the frequency estimates, because it requires less computation.

There are various ways to reduce computation. For example, it may be possible to re-implement the transformation in a less computationally intensive way. For example, a Fourier Transform can be computed using the FFT (Fast Fourier Transform), due to Cooley & Turkey (1965). Alternatively, it may be possible to avoid multiplication, which is often a computationally expensive operation. This is the case with the AMDF algorithm that operates in a similar way to auto-correlation (Dubnowski et al., 1976). Both algorithms are discussed in the next section.

4.2.3 Lag domain analysis

One of the earliest forms of short-term analysis employed correlation techniques. Correlation provides a measure of the similarity between two input signals. A special case that is of particular interest for fundamental period estimation arises when the input signal is correlated with itself, known as auto-correlation. In the case of the sampled input signal x(n), the auto-correlation r(d) is defined as:

$$r(d) = \lim_{N \to \infty} \frac{1}{2N+1} \sum_{n=-N}^{n=N} x(n)x(n+d)$$

Where the parameter d represents the lag between the input signal and a delayed version of itself. The auto-correlation function has several important properties. Firstly, in the case where the input signal is periodic with period P samples, the auto-correlations is also periodic over the same interval; that is

$$r(d) = r(d+P)$$

In addition, the auto-correlation is an even function and attains a maximum value at lag d=0 at which point the value r(0) corresponds to the power in the input signal. So far we have considered only the long-term auto-correlation. In order to perform short-term analysis, it is necessary to modify the definition slightly to include a window function. One possible definition after Rabiner (1977) is as follows:

$$r(d,q) = \sum_{i=0}^{N-1} [x(q+i)w(i)][x(q+i+d)w(i+d)]$$

where q is the starting sample for the short-term analysis and the window w(i) forces sample value to zero outside the interval $0 <= i <= N$. An example of the auto-correlation of voiced speech is shown in figure 4.18.

The period of a periodic input signal can be found by locating the first peak (from the origin) in the short-term auto-correlation, the lag of which corresponds to the fundamental period value T_0. Unfortunately, the results obtained using the auto-correlation directly (or on low-pass filtered) speech are rather poor, because the formant structure of the speech can affect the location of the major peak (Schroeder, 1970).

Centre-clipping

To avoid this problem, it is desirable to pre-process the signal to suppress the effect of the formants. Such techniques are sometimes known as "spectrum flatteners" because their effect is to remove the prominent peaks in the spectrum of the signal due that are to the formant resonances. One such technique is known as centre clipping, and this makes use of the instantaneous non-linearity shown in figure 4.19 (Sondhi, 1968). Its effect is to set all signal values below a pre-set percentage of the short-term peak to zero. The effect is shown in figure 4.20. It can be seen that its effect is to remove many of the smaller peaks in the signal that arise to the dampened resonances of the vocal tract. As a result, the auto-correlation of a centre-clipped signal contains considerably fewer extraneous peaks, and such a scheme performs much better in speech fundamental period estimation (Dubnowski et al., 1976; Rabiner et al., 1977).

As well as centre-clipping, Sondhi (1968) also proposed spectral flattening by means of the use of a set of bandpass filters. The outputs from a bank of bandpass filters (with band-widths around 100Hz) are divided by their short-term envelopes, and the channels once again added together. Sondhi stated that this system was inferior to centre clipping.

The SIFT Algorithm

Another approach to spectral flattening is to employ an inverse filter. A system adopting this approach is due to Markel (1972) and is known as the SIFT (Simplified Inverse Filter Transformation) algorithm. A schematic diagram for this scheme is shown in figure 4.21.

The first step is an initial low-pass filtering at 900Hz and subsequent decimation of the speech from a 10kHz to a 2kHz sampling rate in order to reduce the subsequent processing load. The next stage involves explicitly calculating a linear filter to approximate the inverse vocal tract and excitation source responses using the auto-correlation method of LPC analysis. A relatively low order (fourth) filter is sufficient, because there will generally be at most two formants in the 0-1kHz range. This inverse filter is then used to temporally simplify the speech pressure waveform. Auto-

correlation is then used to estimate its fundamental period. Because the sampling rate at this point is rather low (2kHz), it is then necessary to employ interpolation around the peak in the auto-correlation function to increase the resolution of the period estimate. This algorithm suffers from the limitations associated with the inverse filtering process, and there are problems when the speakers frequency range is high (for example in the case of children) because the spectral flattening tends to fail when there is no more than one harmonic in the 0-900Hz range.

Average Magnitude Difference Function (AMDF)

Another function that measures the similarity between a signal and a delayed version of itself is the average magnitude difference function (Ross et al., 1974). This is defined as;

$$AMDF(d) = (1/k).\sum_{n=q}^{q+K-1} | x(n) - x(n-d) |$$

This function exhibits a strong minimum when the lag d becomes equal to the fundamental period T_0, and has some similarities to the auto-correlation function. Because the AMDF function does not employ multiplications, it is computationally less demanding than auto-correlation, although it is more susceptible to changes in input signal intensity (Hess, 1983).

There are other approaches to fundamental period estimation that use distance functions other than the AMDF. For example, Nguygen & Imia (1977) and Sanchez (1977a,b) use a more general function of which the AMDF is one case. Ney (1982) used a generalized distance function in conjunction with dynamic programming. One disadvantage of this last technique is that it requires all the speech samples within a voiced segment to be present before the optimum estimate can be calculated.

4.2.4 Frequency-domain analysis

Frequency-domain algorithms operate on some kind of spectral representation of the signal. They can take advantage of the harmonic structure of the excitation, and in some algorithms the fundamental does not even have to be present for such schemes to function. A valuable feature of frequency-domain analyzers is that the resolution of their frequency estimates can be increased relatively easily by means of interpolation.

The simplest frequency-domain analysis of speech fundamental frequency would simply involve searching the short-term spectrum for the first harmonic. This approach will, of course fail if it is weak or not present. In addition, the accuracy will be low because the relative frequency resolution is worse the lower the frequency. To get around these problems, algorithms more often measure the spacing between adjacent harmonics, or compute weighted averages of the higher harmonic frequency values.

Harmonic product spectrum

One technique that combines together estimates from all the harmonics is based on the principle of spectral compression, and uses the logarithmic harmonic product spectrum (Schroeder, 1968; Noll, 1970). The short-time log power spectrum is compressed along the frequency axis by integer factors, and the individual compressed versions are then added together. This operation is defined by

$$P(m) = \sum_{k=1}^{K} \log |X(km)|^2 = 2\log \prod_{k=1}^{K} |X(km)|$$

where X(m) represents the input spectrum and P(m) represents the log harmonic product spectrum which is the sum of K frequency compressed versions of the input spectrum. For voiced speech, harmonics of the fundamental frequency coincide and add, whereas this does not generally happen at other frequencies. This results in a peak at F_0 which becomes sharper as the number K increases. This is illustrated in figure 4.22. One strength of this algorithm is that it has been found especially resistant to noise, because the contribution in the input spectrum X(m) due to noise does not add constructively

after the stages of compression. Another strength is that the fundamental harmonic does not need to be present with this algorithm.

Frequency and Period histograms

Another scheme, proposed by Schroeder (1968) involves building up histograms of the frequencies of the peaks present in the short-term spectrum, which occur at harmonics of the fundamental frequency. The frequencies of the peaks are then divided by two, and added once again to the histogram. The procedure is repeated with compression factor of 3,4 etc. This results in a peak occurring at the fundamental frequency, which can then be detected.

Harmonic pattern matching

An algorithm due to Martin (1981,1982) employs the principle of harmonic pattern matching by means of applying a comb filter to the short-term amplitude spectrum of the input signal. This involves searching for values of the spectrum that are situated at harmonic frequencies, such that their sum is a maximum over a given frequency interval. The fundamental corresponding to the harmonic structure that has the largest sum is considered to be the fundamental frequency of the signal. Thus, expresses mathematically, a spectral comb C(m,p) is defined as a series of impulses:

$$C(m,p) = C(k) \text{ if } m = k;\ k = 1,2,3...M$$
$$= 0 \text{ otherwise}$$

where p is the trial fundamental frequency (This value of p is determined by the spectral analysis). For each p the input spectrum A(m) is multiplied by the comb C(m,p) and the resulting components are added up to give the harmonic estimator function $A_C(p)$ as follows:

$$A_C(p) = \sum_{k=1}^{N/2p} A(kp)C(kp,p)$$

The value of p at which A_C is maximal is taken as the estimated fundamental frequency F_0. In practice, the comb "teeth" must be weighted to avoid octave errors (that is, errors where the estimate is wrong by a factor of 2).

Another group of algorithm that have similarities to harmonic compression are those that adopt the principle of maximum likelihood period estimation (Wise et al., 1976; Friedman, 1977). Both approaches make use of a comb filter that enhances the harmonic structure and is optimally matches to the signal. However, in the latter cases, the problem is formulated (in the lag domain) with respect to the time waveform of the signal, rather than its spectrum.

Psychoacoustically-based fundamental frequency estimation

There are various algorithms that apply models of pitch perception to the estimation of speech fundamental frequency. One such model is based on a the model by Goldstein et al. (1973) and was used by Duifhuis, Willems & Sluyter (1978, 1979, 1982). The overall procedure is somewhat similar to that proposed by Martin (described above). The algorithm uses what is described as a "harmonic sieve", which is a spectral comb of finite resolution that filters out all except harmonically related frequency values from an input spectrum. The estimated fundamental frequency F_0 is computed as the maximum likelihood estimate from all the peaks that pass through the sieve. Another similar algorithm is due to Terhardt (1972a,b; Terhardt et al., 1982a,b).

Cepstrum Processing

The cepstral technique (Noll 1964,1967) is a special case of what is known as homomorphic filtering (Oppenheim & Schafer, 1975). The idea behind the cepstrum is to separate out from the speech waveform the effect of the excitation source and the response of the vocal tract. Thus we wish to undo the convolution

$$x(n) = p(n)*h(n)$$

where x(n) represents the speech signal, p(n) represents the impulse response of the vocal tract, s(n) represents the excitation signal and * denotes convolution. In the frequency-domain, this becomes

$$X(w) = P(w).H(w)$$

where

$$X(w) = F\{x(n)\}$$
$$H(w) = F\{h(n)\}$$
$$P(w) = F\{p(n)\}$$

where F{ } denotes the Fourier transformation. By taking the logarithms of the power spectra, this relationship becomes additive. That is;

$$\log[|X(w)|^2] = \log[|P(w)|^2] + \log[|H(w)|^2]$$

The voiced excitation signal manifests itself in the log power spectrum of the speech as a high frequency cosine-like ripple due to the harmonics, whereas the vocal tract response gives rise to a low frequency ripple (Noll, 1967). This is illustrated in trace a) in figure 4.23. By calculating the inverse Fourier transform of the log power spectrum, one then gets back to a time-domain signal known as the **cepstrum** of the input signal, in which the temporal effects of the vocal tract and excitation are separate. Thus the cepstrum exhibits a strong peak at a quefrency (which is the term used to denote time in the cepstral domain) equal to the fundamental period duration T_0 of the input signal (see trace b) in figure 4.23). As in the case of auto-correlation, the height of this peak relates to the periodicity of the input signal, and if it falls below a preset threshold level, the input is assumed to be unvoiced. Implementations of cepstral speech fundamental period estimation often additionally employ a set of rules (known as the Noll rules) to locate the appropriate peak in the cepstrum, which involves adapting the

threshold on basis of past period estimates, as well as checking for possible period doubling and halving conditions. These rules reduces the number of gross errors generated by the technique.

To summarise, the cepstrum is the spectrum of the logarithm of the power spectrum of the speech pressure waveform and for voiced speech, the cepstrum has a peak, the location of which corresponds to the fundamental period. Of course, all the analysis must be performed on a short-term basis for reasons previously discussed. A schematic diagram for cepstral period estimation is given in figure 4.24.

At the time it was first published, the cepstral technique constituted a breakthrough, since it was much more reliable than many other approaches. Consequently, for a long time it was adopted as a reference against which other algorithms have been compared (Liedke, 1971; Markel, 1972; Moorer, 1974; Martin, 1981, 1982). However, the cepstral technique requires that there be many adjacent harmonics in the input signal; otherwise there will not be periodic ripples in the log power spectrum of the input signal. For example, the cepstral technique cannot estimate the fundamental frequency of a sinusoidal signal. On the other hand, the cepstral technique is quite able to deal with a strong formant structure in the input signal. Therefore, the cepstrum behaves in a complementary way to auto-correlation, which experiences difficulty with strong formants but is able to deal with pure sinusoids.

4.3 LARYNGEAL MEASUREMENT OF SPEECH FUNDAMENTAL PERIOD

4.3.1 Introduction

Laryngeal devices operate by attempting to estimate vocal fold activity by direct measurement. When this approach is successful, it leads to a signal that itself is temporally simpler that the speech pressure waveform. Consequently, the fundamental period of voiced speech can be estimated from such signals using relatively simple extractors. There are two basic approaches to estimating the vocal fold activity and these involve either using contact microphones or an electro-glottograph. Photographic

techniques for the examination of vocal fold vibration using strobe lights have also been used, but they are not practical for everyday use. They do, however, provide useful insight into the interpretation of electro-glottographic signals.

Contact microphones

Contact microphones are sensitive to vibrations as well as acoustic pressure variations, and when such a device is placed on the neck in the vicinity of the larynx, the detected signal reflects vocal fold movement and response of the body to the acoustic disturbance in the trachea. McKinney (1965) made several observation concerning the use of contact microphones. To achieve good results, it is necessary that an airtight seal is maintained around the microphone, and that the output waveform changes according to its location and the neck of the used. Contact microphones are also sensitive to movement of the speaker. However, the output waveform is temporally simple and relatively unaffected by the movement of the articulators, although the waveform shows little relationship to the excitation derived from inverse filtering or glottal photography. Overall, McKinney decided that such microphones were not of great value in speech fundamental frequency estimation.

Electro-glottograph

An electro-glottograph operates by measuring the conductance across the neck at the level of the vocal folds, which provides an estimate of vocal fold contact and therefore vocal fold movement. Again, the output waveform is well defined and temporally simple, and is even less influenced by the action of the articulators than contact microphones. One such device is the laryngograph (Fourcin & Abberton, 1971).

4.3.2 The laryngograph

As stated previously in chapter 2, the laryngograph works by measuring the conductance across the larynx at the level of the vocal folds. The output waveform from the laryngograph thus gives a direct measure of vocal fold activity and is temporally much

simpler than the corresponding speech pressure waveform. The point of closure of the vocal folds, which gives rise to the main peak in excitation, can be easily determined from the laryngograph output waveform (Lx). Therefore, by means of a relatively simple time-domain fundamental frequency estimation algorithm, a good estimate of speech fundamental frequency can be obtained. One big advantage of the laryngographic technique is that it is easy to perform period-by-period estimation and consequently the smearing of fundamental frequency values across time is avoided.

Hess argues that the laryngograph can form the basis of an ideal instrument for speech fundamental period estimation because it is robust, reliable, does not interfere with articulation and is essentially immune to environmental noise (Hess,1983; Hess & Indefry, 1984). However, it is important to appreciate the limitation of the laryngograph in this application (as discussed in chapter 2).

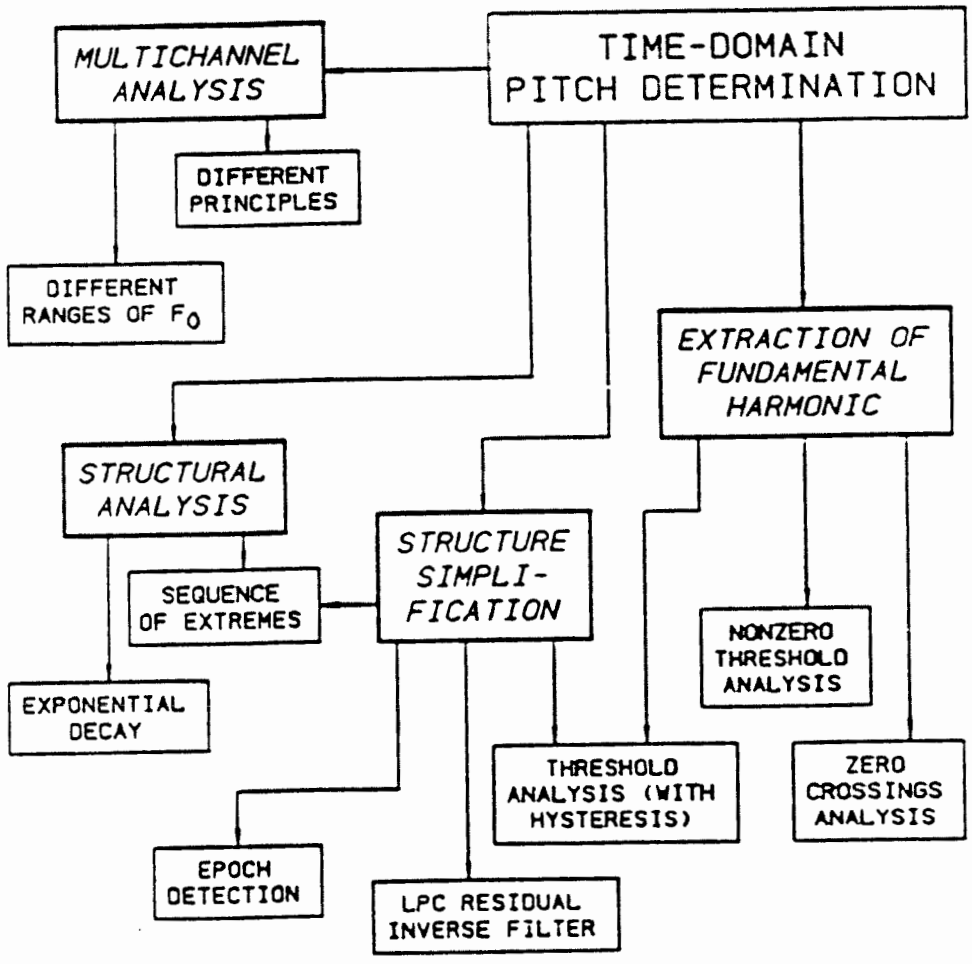


Figure 4.1 Classification of time-domain fundamental period estimation algorithms. These approaches are explained in the main text.
(After Hess, 1983).

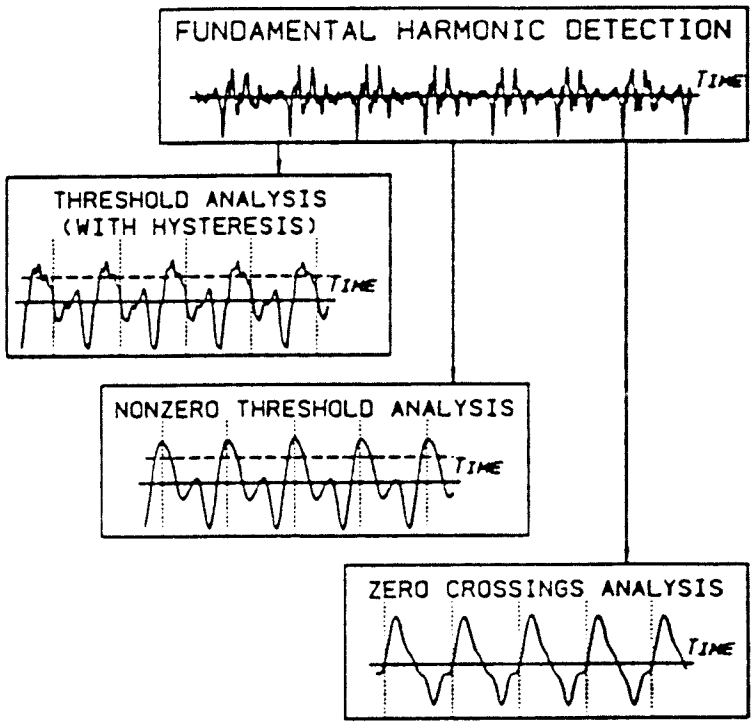


Figure 4.2 Sub-classification of fundamental harmonic detection techniques using threshold analyzers.

These approaches are explained in the main text.

(After Hess, 1983).

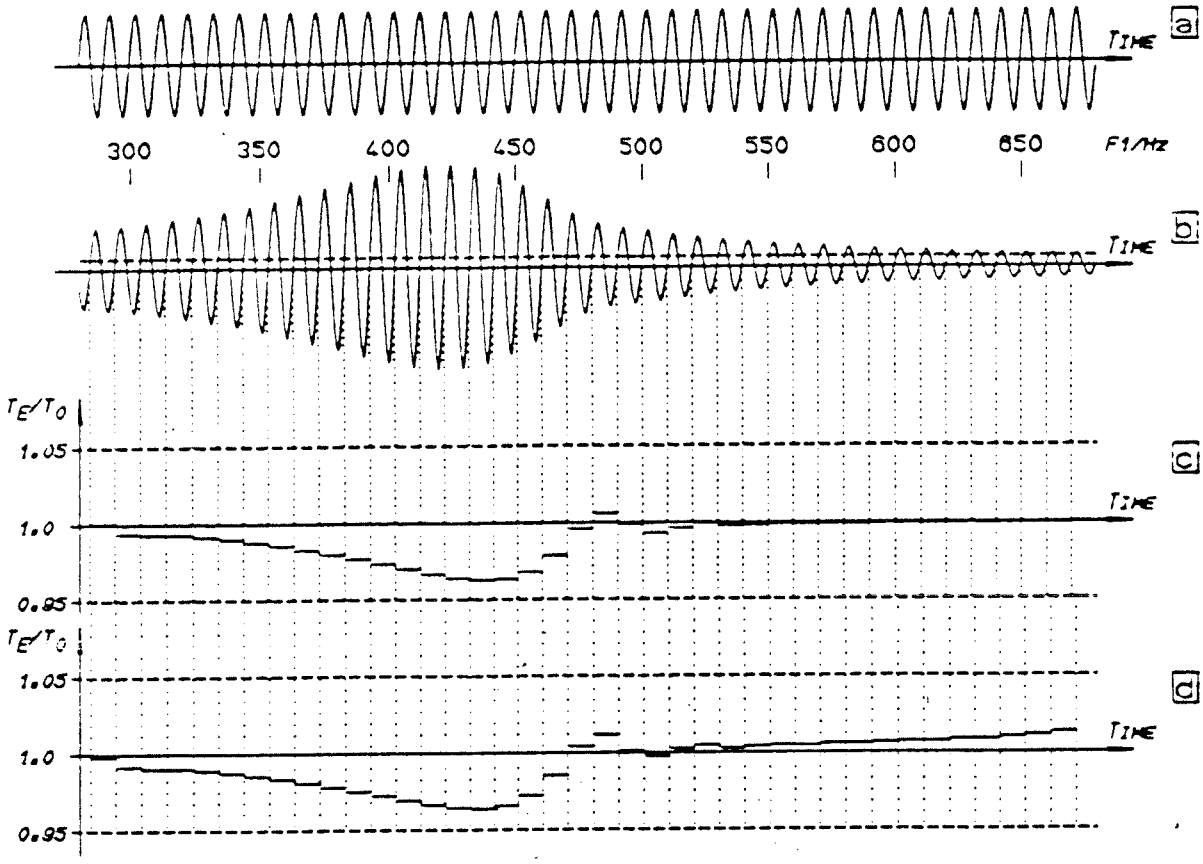


Figure 4.3 Effect of rapid changes in formants on the estimated fundamental frequency derived using a threshold analysis.

Trace a) shows an input sinusoid. This is fed into a 2nd order time-variant filter, which simulates a formant transition between 300Hz and 650Hz in 90ms, and its output is shown in trace b). The ratio of estimated fundamental period to true fundamental period for a zero-crossing analyzer and a threshold analyzer (operating at 10% of the signal peak) are shown in traces c) and d) respectively. It can be seen that there is a deviation of several percent in both cases.

(After Hess, 1983).

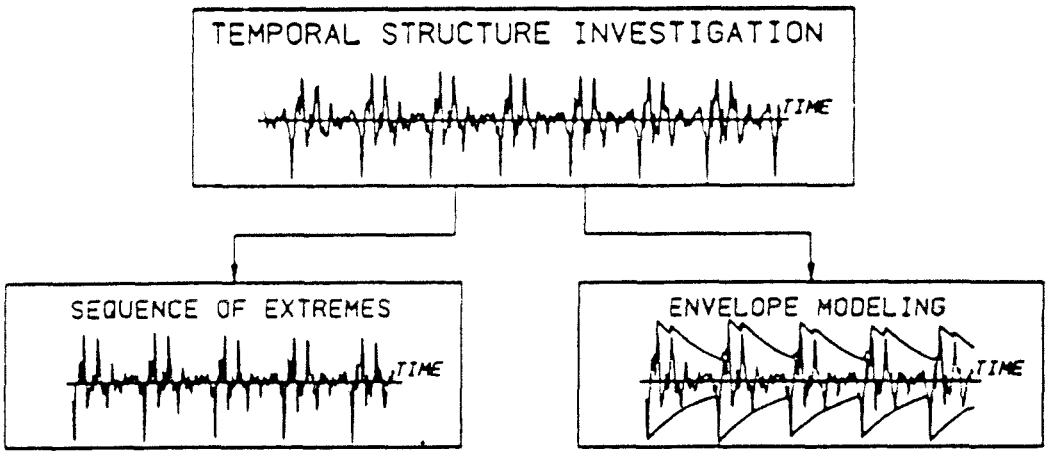


Figure 4.4 Sub-classification of structural analysis fundamental period estimation techniques.

These approaches are explained in the main text.

(After Hess, 1983).

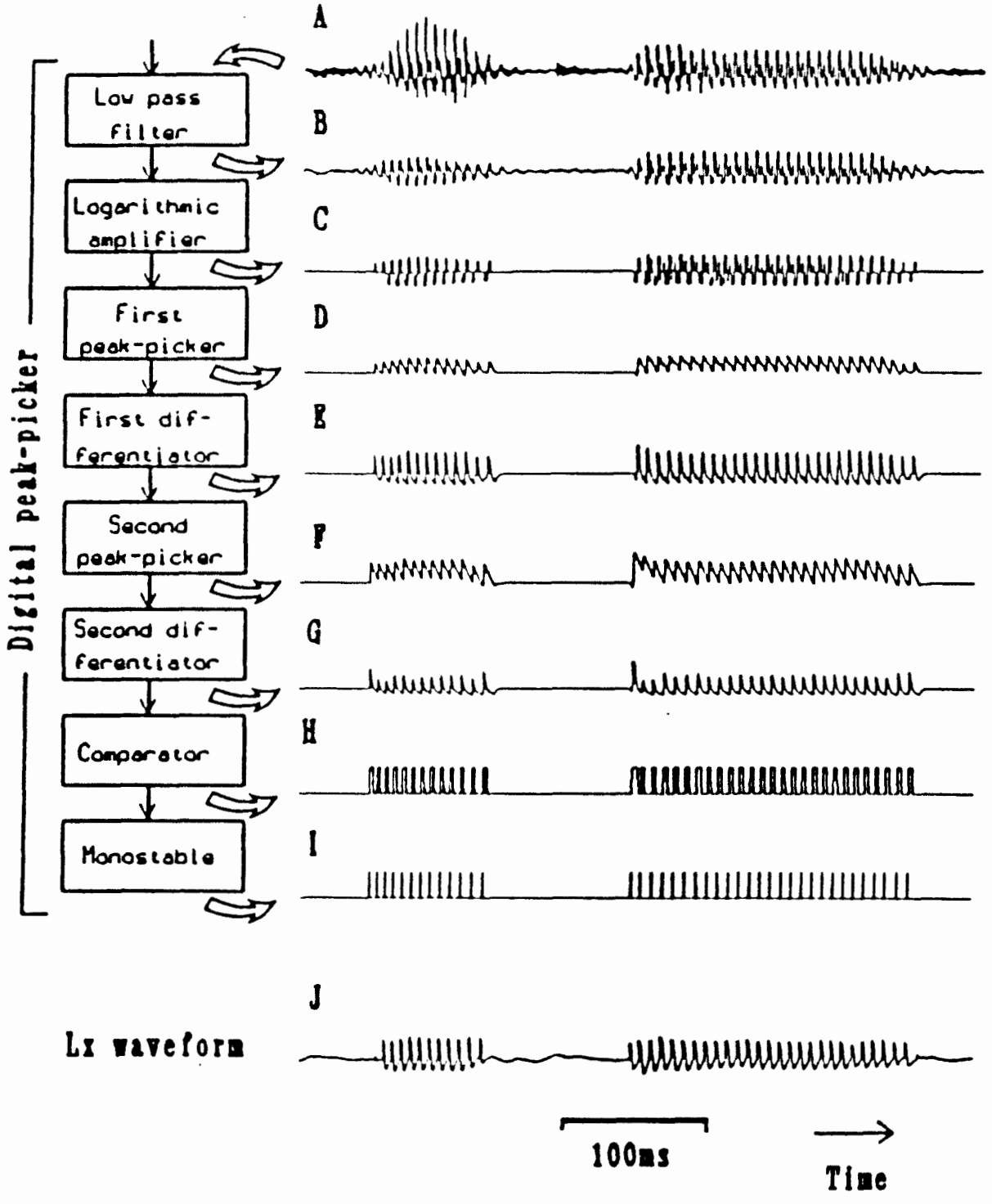


Figure 4.5 Operational waveforms in the peak-picker.

The stages of processing and the operational waveforms generated at each stage are shown. The output of a laryngograph is shown in the bottom trace. Full details appear in the main text.

(After D M Howard, 1986).

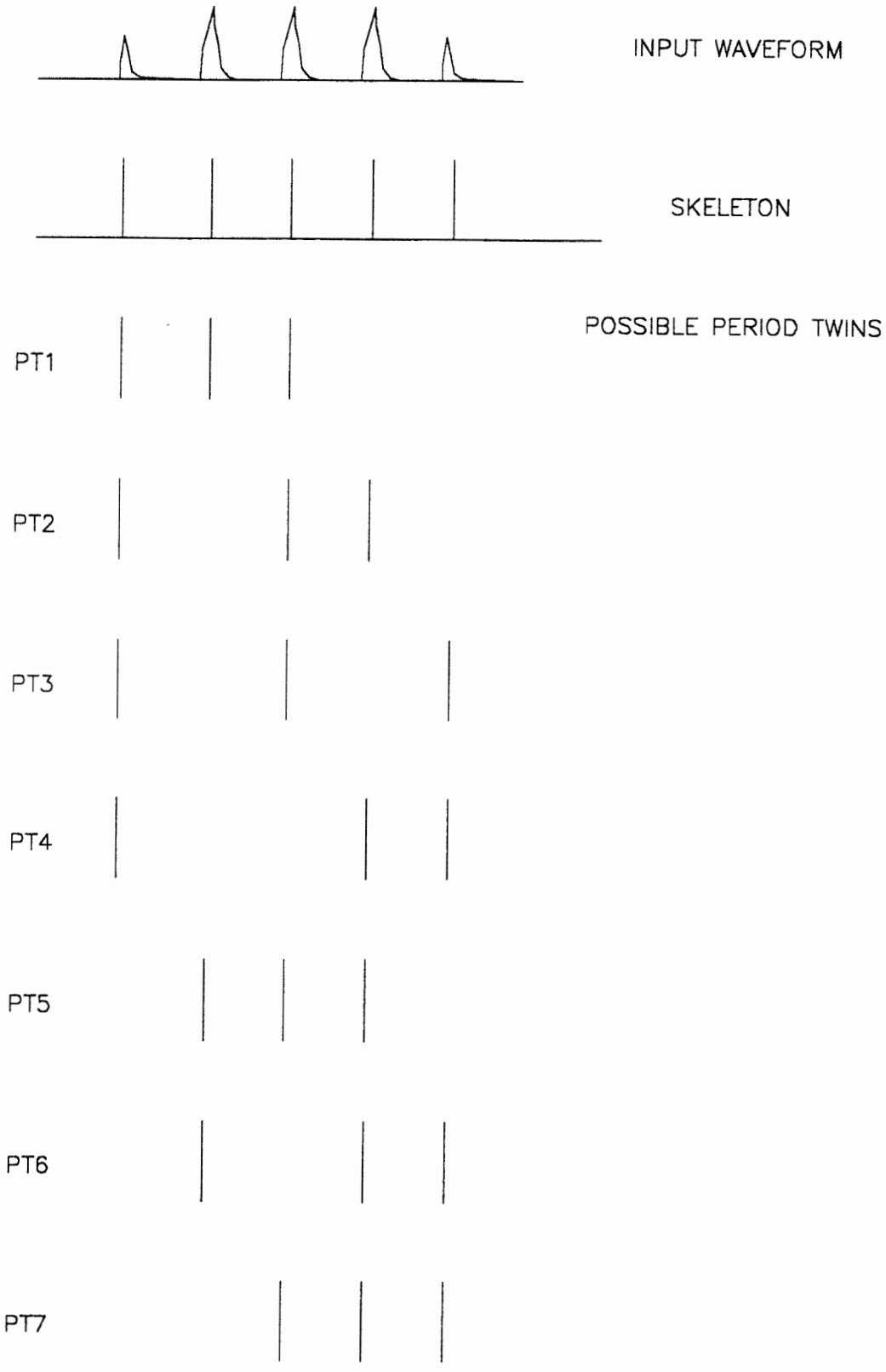


Figure 4.6 Generation of period twins in the pitch chaining algorithm.
The first stage of operation in the generation of a skeleton of extreme values of the input speech waveform. This is illustrated in the first two traces in the diagram for a simple example waveshape. The next stage is the generation of all possible sets of three period markers (the period twins), illustrated in the remaining traces. These are then subject to a set of conditions to remove unlikely candidates before a period estimate is made. (Algorithm after Schafer-Vincent, 1983).

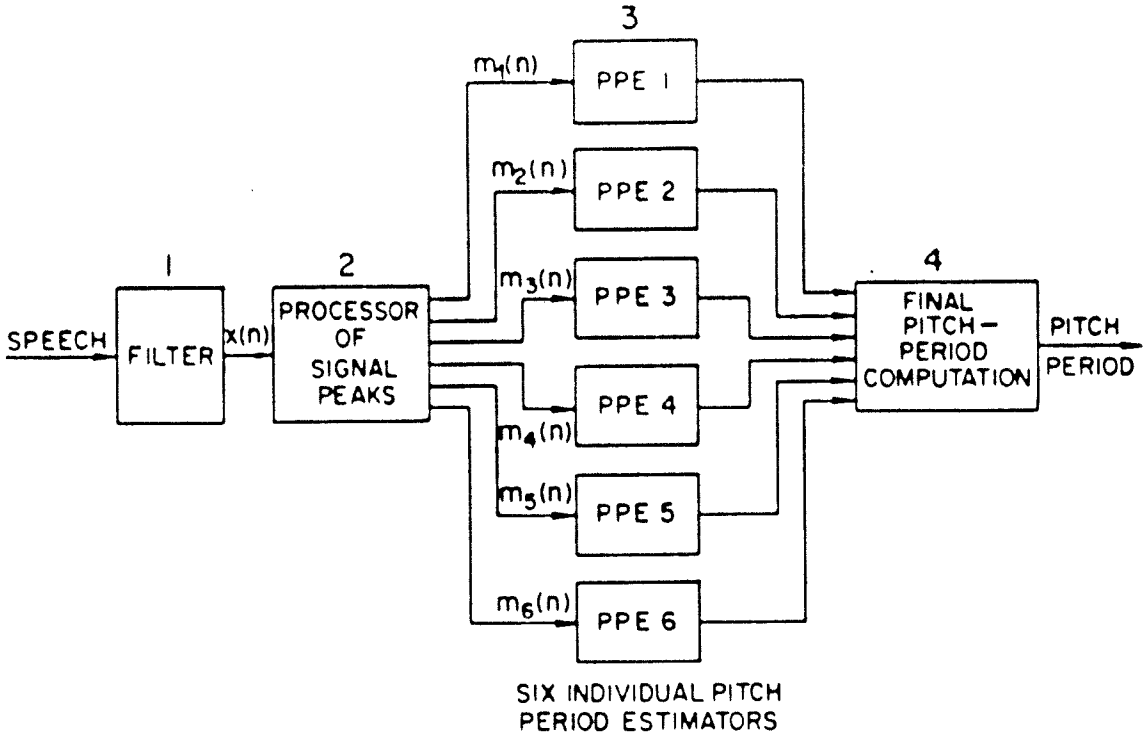


Figure 4.7 Schematic diagram for the Gold-Rabiner algorithm.

The input speech is first low-pass filtered (state 1), and then the signal peaks are examined (stage 2). Six different measures from the previous stage are then fed into a set of six period estimators (stage 3). Finally the outputs from the basic extractors are examined and the period value is estimated (stage 4).

(After Gold & Rabiner, 1969).

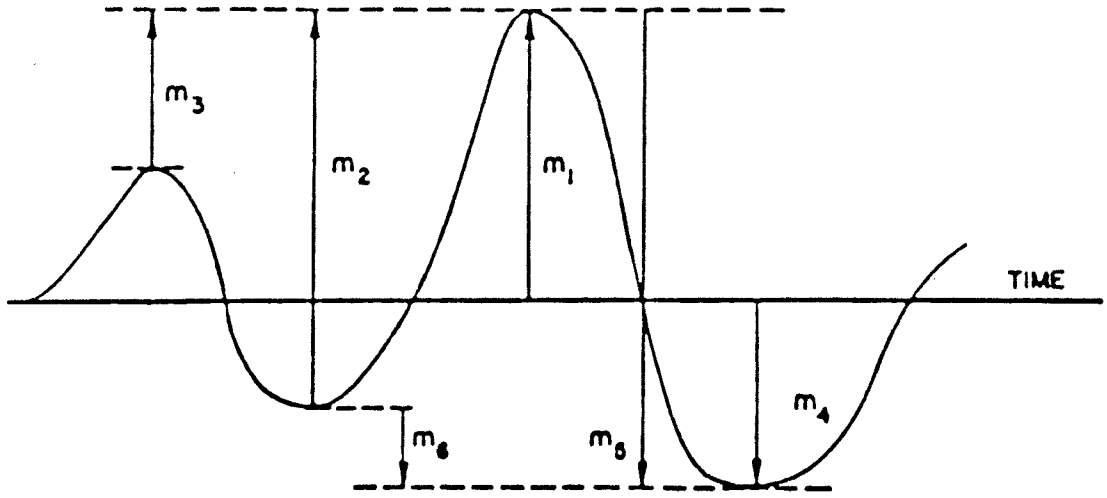


Figure 4.8 Relationship between measurements m_1 - m_6 and the corresponding features of the waveform used in the Gold-Rabiner algorithm.
These signal measurements are explained more in the text.
(After Gold & Rabiner, 1969).

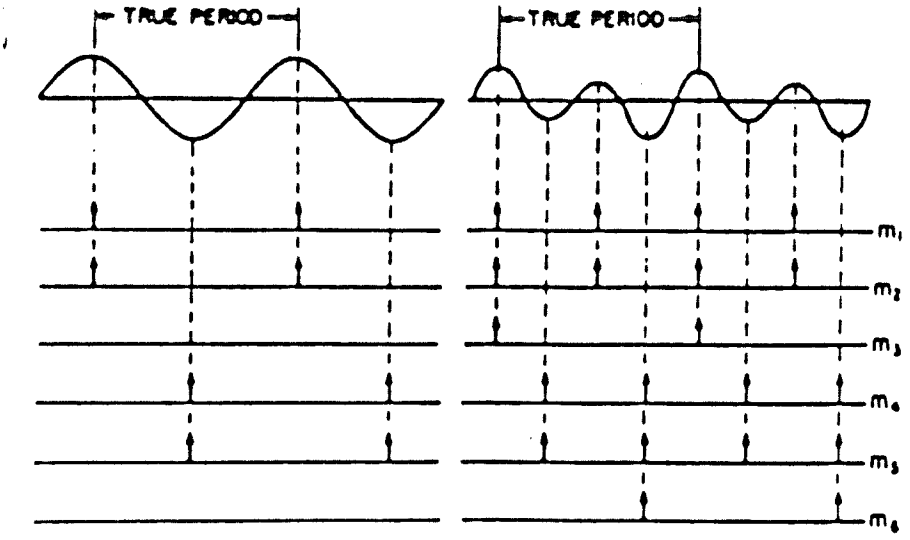


Figure 4.9 Behaviour of basic measurements in the Gold-Rabiner algorithm for two simple waveforms.
The first case shown is a pure sine wave and the second is a waveform with a strong second harmonic.
(After Gold & Rabiner, 1969).

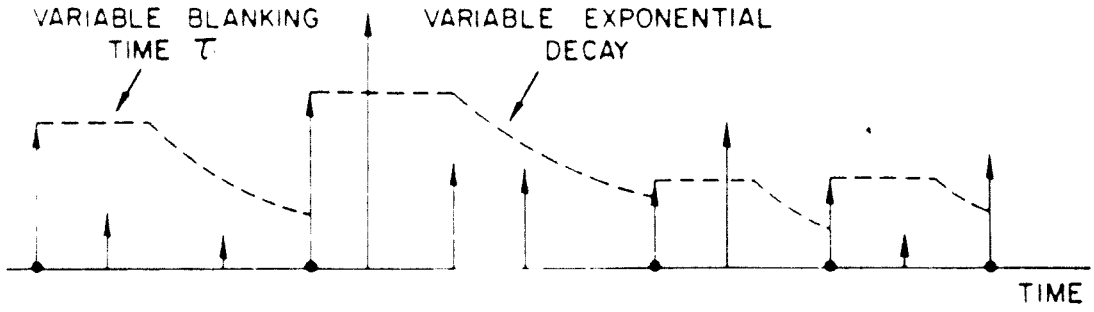


Figure 4.10 Operation of the basic extractor used in the Gold-Rabiner algorithm. It operation is such that a large measurement pulse can mask preceding smaller ones by means of a blanking interval and an exponential decay. Further details appear in the text.
(After Gold & Rabiner, 1969).

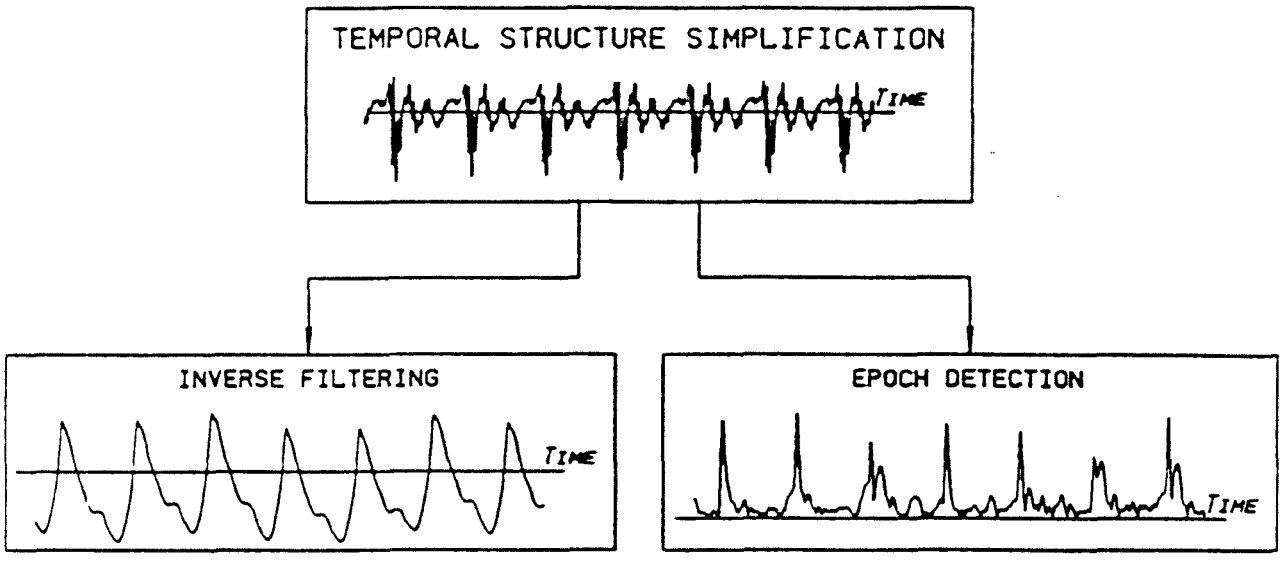


Figure 4.11 Sub-classification of temporal simplification techniques.
These approaches are explained in the main text.
(After Hess, 1983)

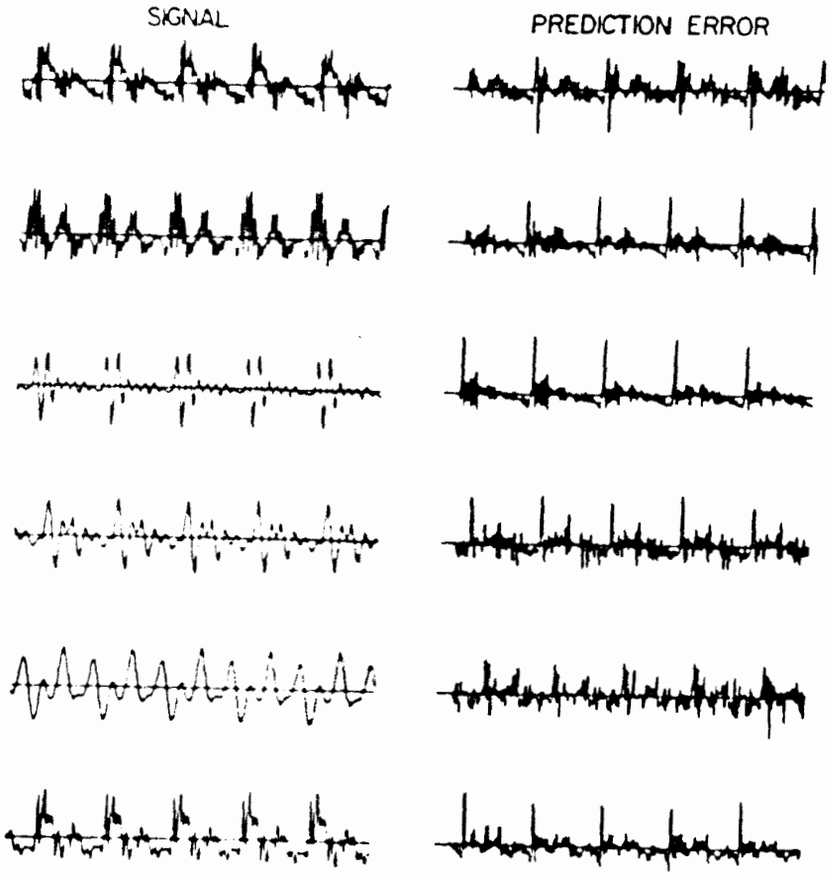


Figure 4.12 Speech signals and their corresponding LPC prediction error.
The input differentiated speech (from top to bottom) corresponds to the vowels (i, e, a, o, u ,y).
(After Strube, 1974).

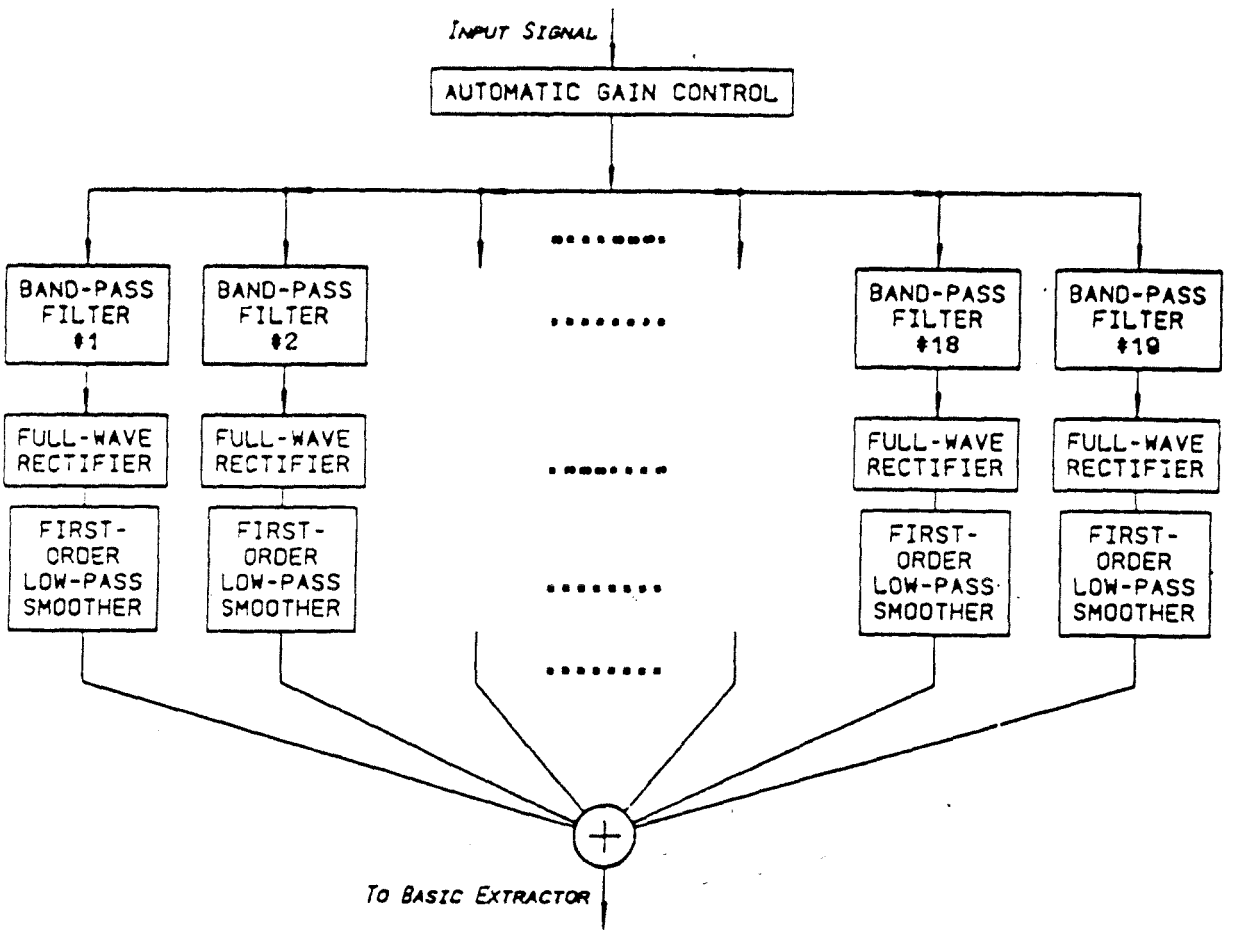


Figure 4.13 Schematic diagram for multi-channel epoch detector used by Yaggi. (After Yaggi, 1962).

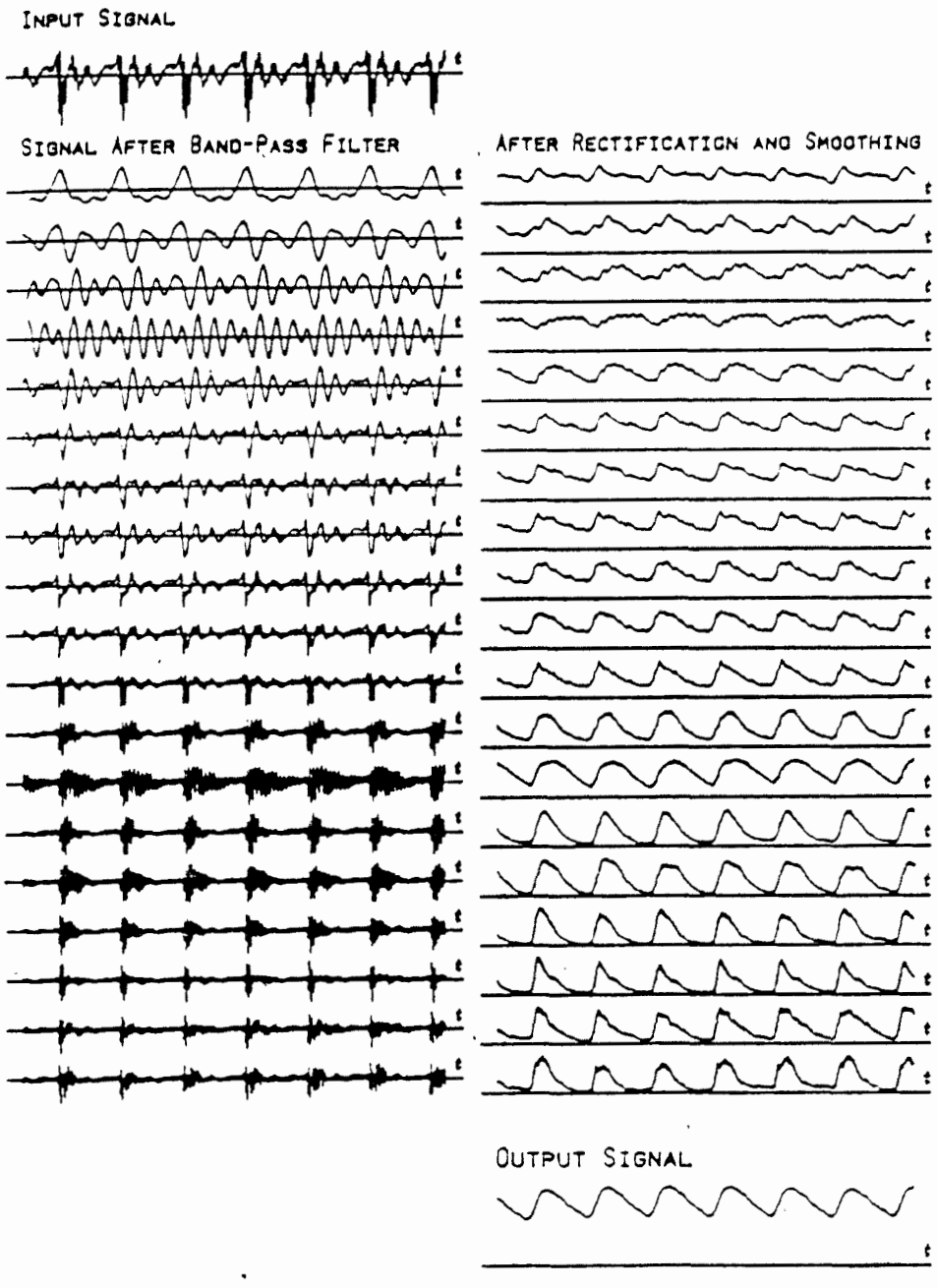


Figure 4.14 Operational waveforms from multi-channel epoch detector used by Yaggi. The bandpass filters covered the range of 120Hz to 3.8kHz and their bandwidths varied between 114Hz to 442Hz, from top trace to bottom trace respectively. The speech is the vowel "a" from "algorithms" spoken by a male subject.
(After Yaggi, 1962).

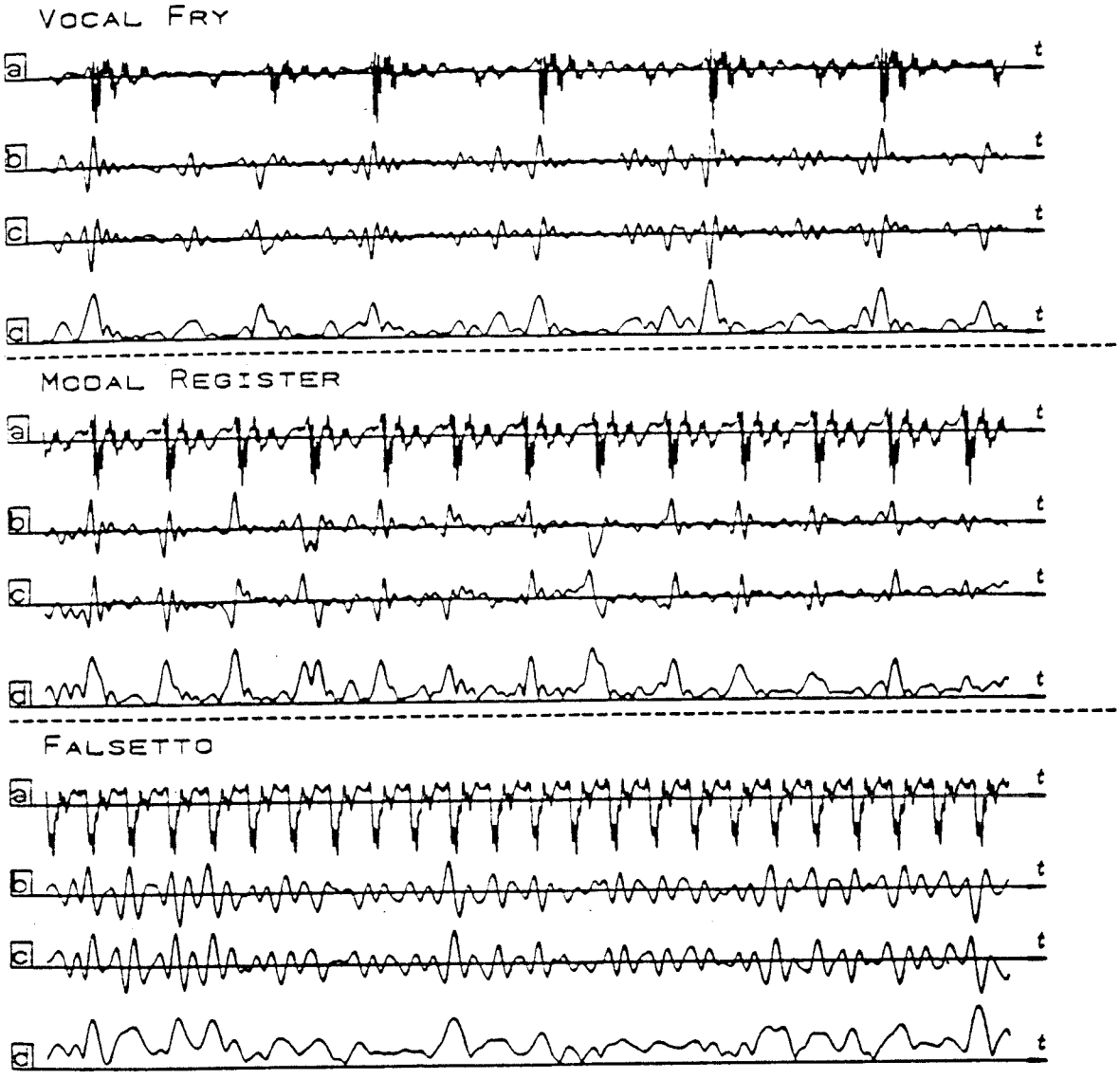


Figure 4.15 Output waveforms from epoch detector based on the identification of discontinuities in the speech waveform.

Example of the responses obtained in the case of vocal fry (creak), modal register (normal voice) and falsetto voice. In each case the speech pressure waveform appears in trace A, and the output from the epoch filter is shown in trace D (traces B and C show intermediate results used to compute the final output and represent the bandpass filtered input and the 90 degree phase-shifted bandpass filtered input respectively). The speech token is for the vowel /E/ produced by a male subject. Notice that the system only operates usefully when the glottal closures are well defined.

(Taken from Ananthapadmanabha & Yegnanarayana, 1979).

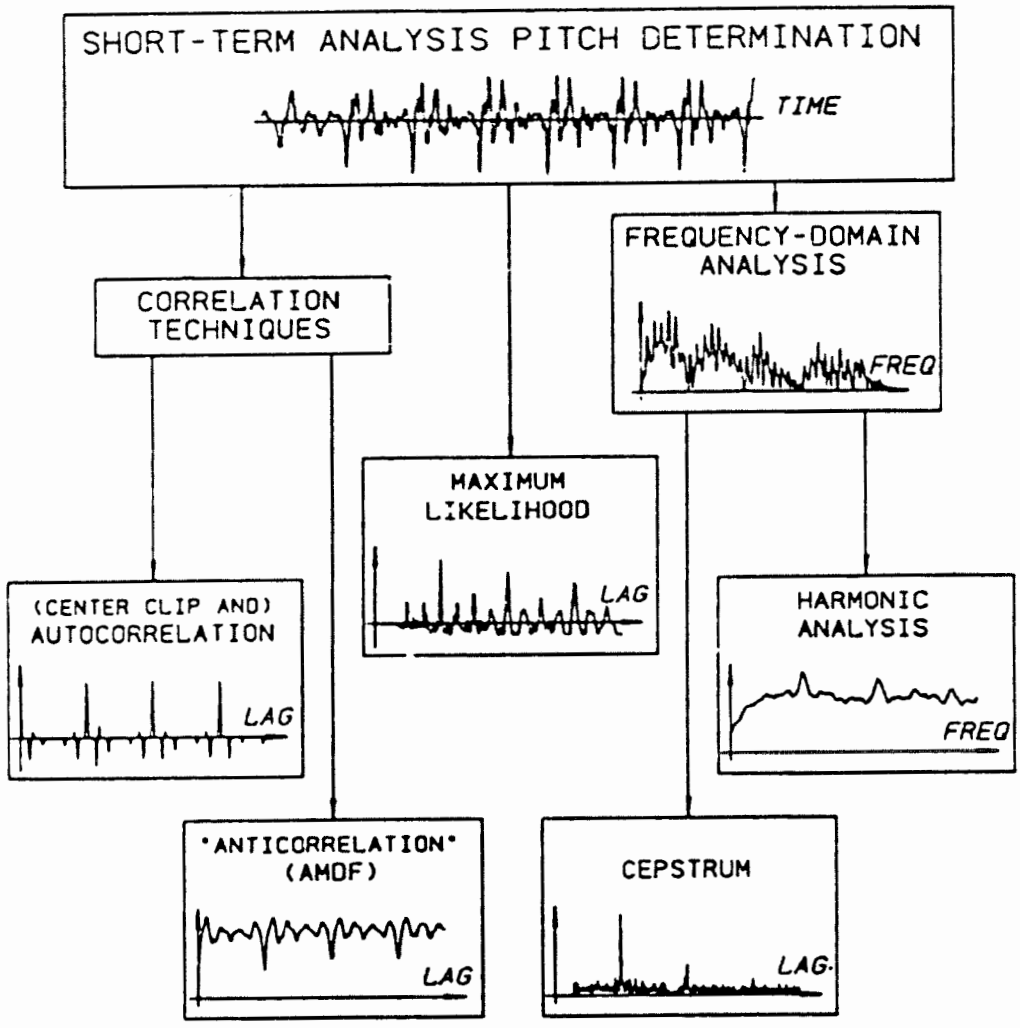


Figure 4.16 Sub-classification of short-term fundamental frequency estimation algorithms.

These approaches are explained in the main text.

(Taken from Hess, 1983).

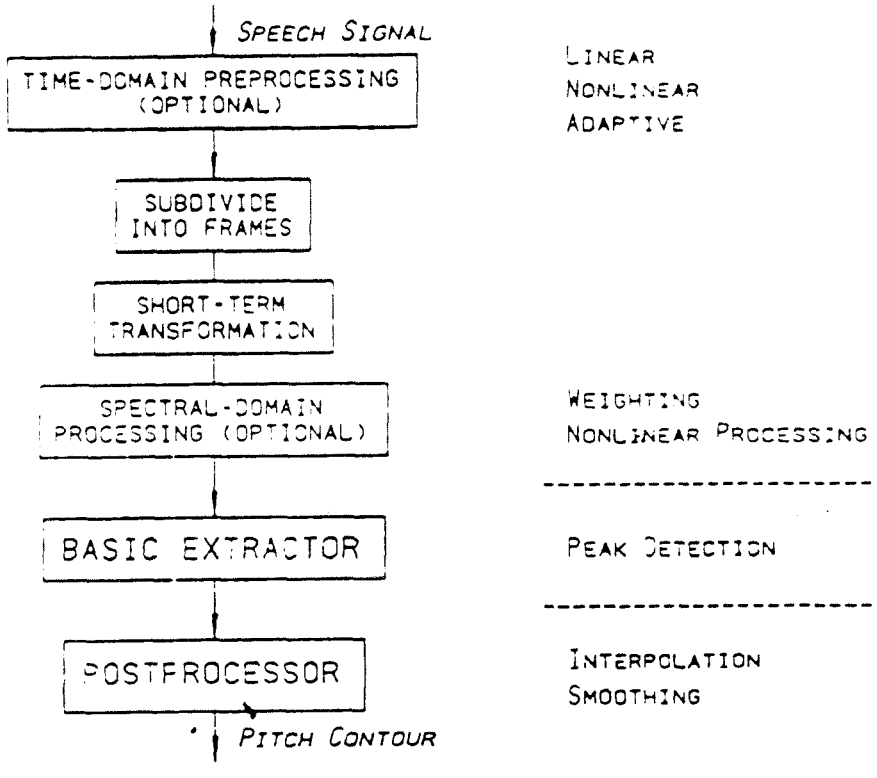


Figure 4.17 Schematic diagram for a frequency-domain fundamental frequency analyzer. Possible different stages of operation are illustrated. These consist of pre-processing stage, the division of the input into frames, followed by the computation of a short-term transformation on each frame. Some kind of operation on the spectrum is then performed, followed by the detection of a principle peak. Finally interpolation and smoothing are carried out.
(Taken from Hess, 1983).

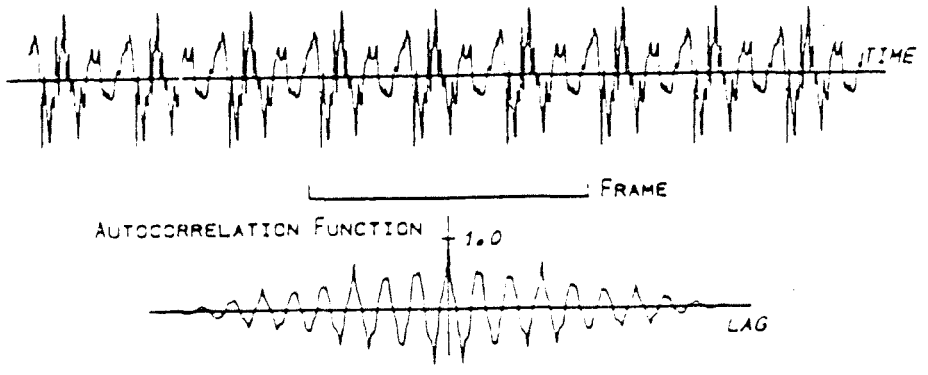


Figure 4.18 Voiced speech and its corresponding short-term auto-correlation.
The third peak (the largest non-zero offset peak) from the origin represents the fundamental period.
(Taken from Hess, 1983).

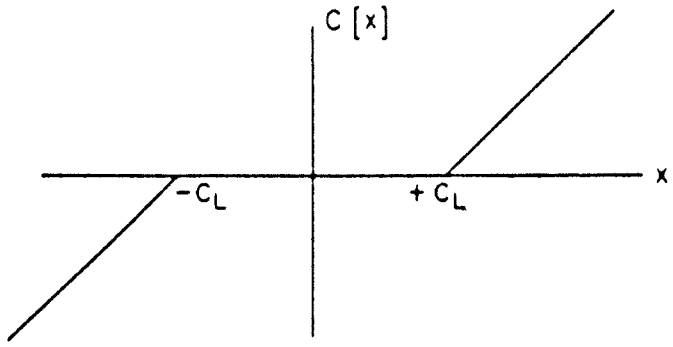


Figure 4.19 Function used to centre-clip speech.
(Taken from Rabiner & Schafer, 1978).

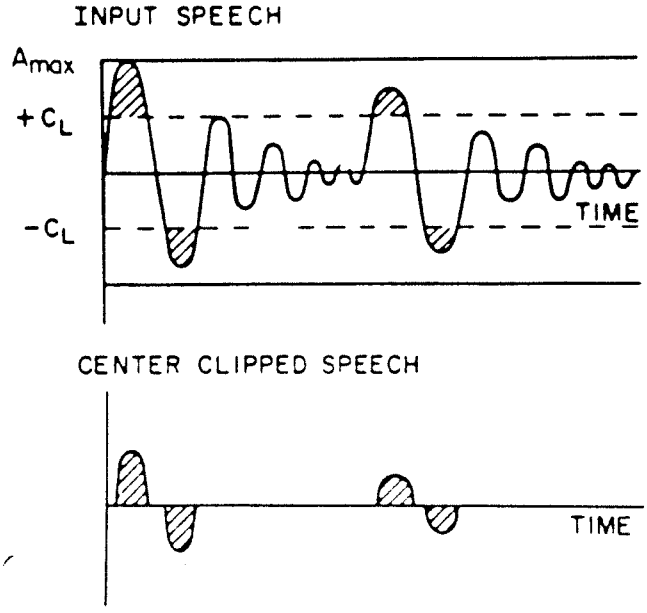


Figure 4.20 Effect of centre clipping on a simplified speech waveform.
(Taken from Sondhi, 1968).

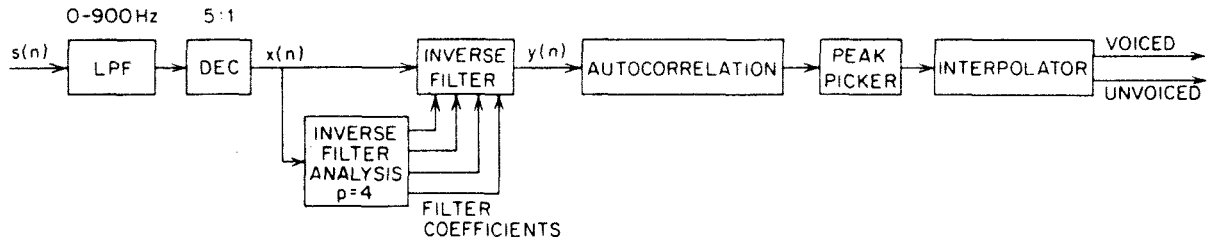


Figure 4.21 Block diagram of the SIFT algorithm.

Firstly, the input is low-pass filtered and decimated. An LPC inverse filter is then used to simplify the input speech. Autocorrelation is then used to determine the fundamental period value and interpolation is used to increase the resolution of the estimate. (Taken from Rabiner & Schafer, 1978; After Markel, 1972).

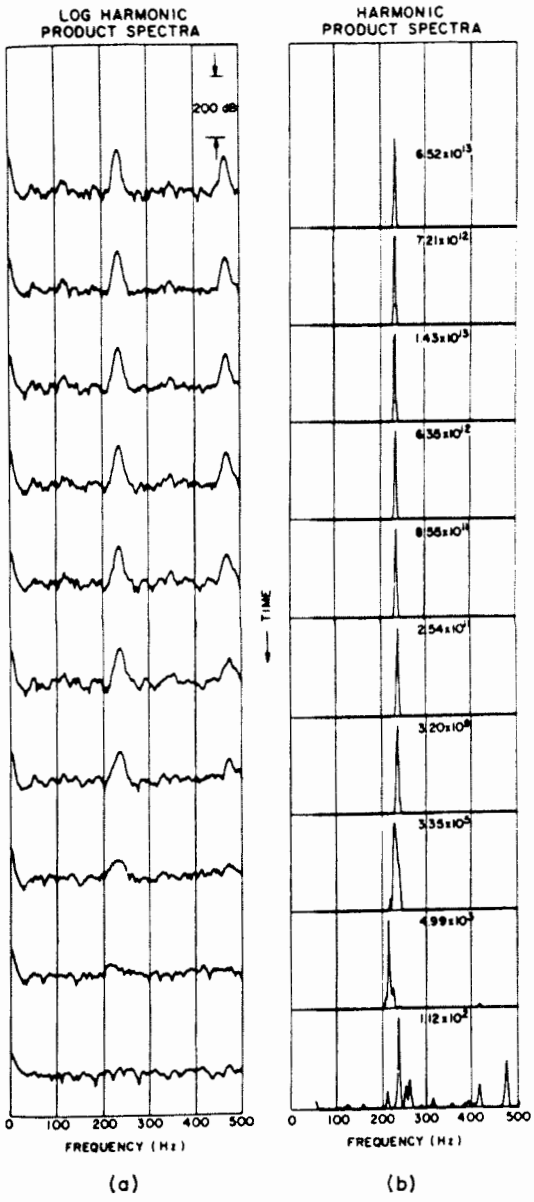


Figure 4.22 Example of the spectral compression to estimate fundamental frequency. A series of log harmonic product spectra are shown in a) and their harmonic product spectra are shown in b).
(Taken from Noll, 1970)

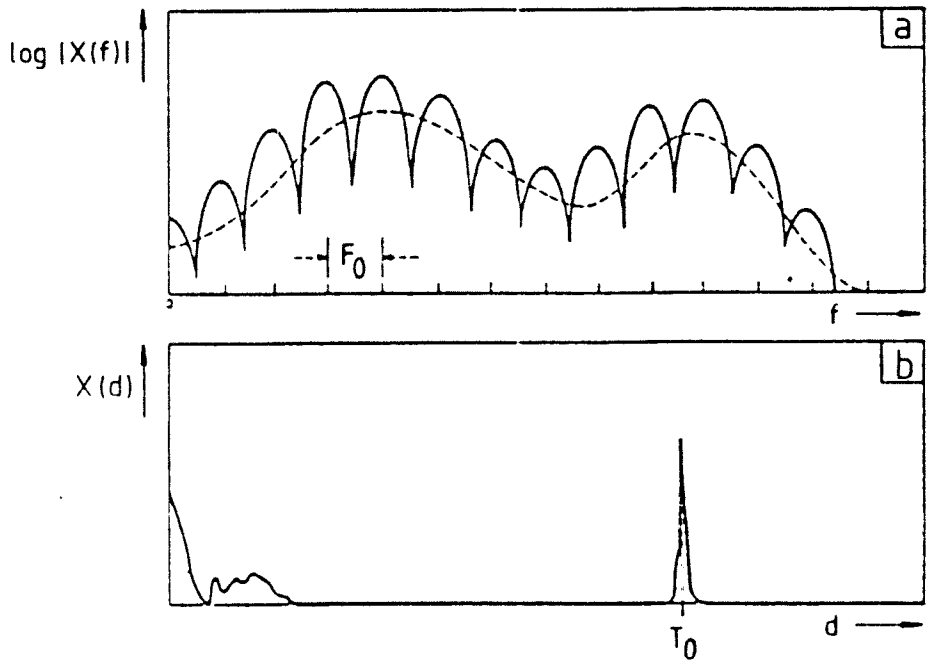


Figure 4.23 Logarithmic power spectrum of voiced speech and its corresponding cepstrum.

Trace a) show the log power spectrum exhibiting cosine-like ripples due to the excitation harmonics. The overall spectral shape is due to effect of the vocal tract resonances. The cepstrum is shown in b). There is a well defined principle peak corresponding to the fundamental period.

(Taken from Noll, 1967)

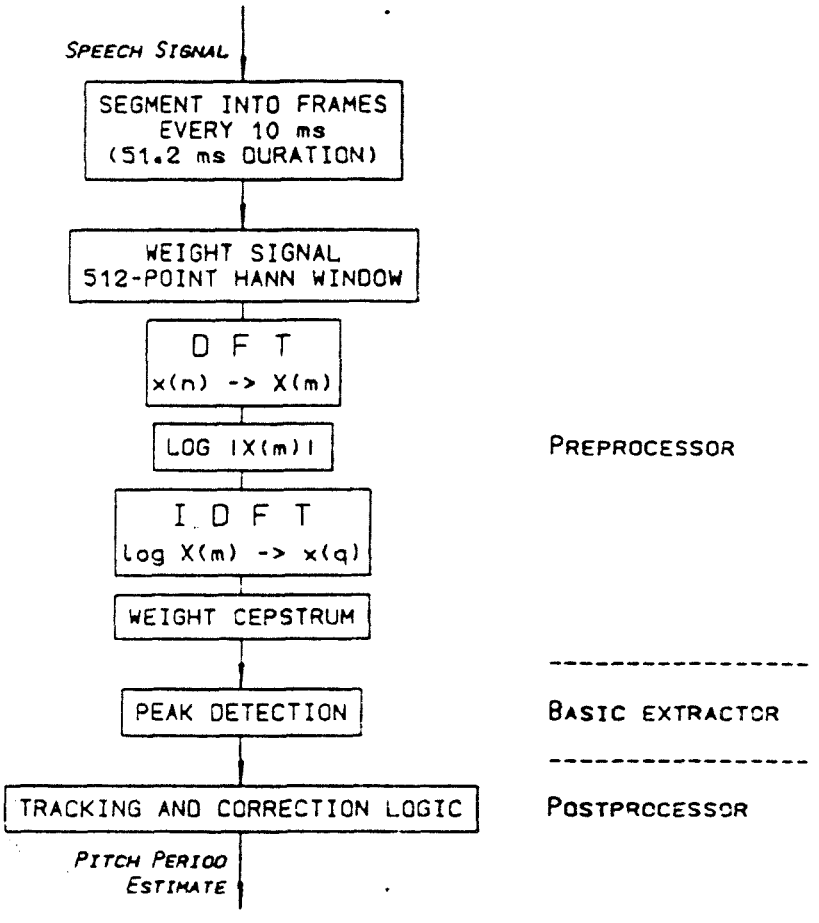


Figure 4.24 Block diagram of the cepstrum algorithm.

This involves dividing the input up into frames, calculating the log power spectrum for each frame and then calculating the inverse Fourier transform of the log power spectrum. The peak in the cepstrum is then located which the fundamental frequency estimate in that frame.

(Taken from Noll, 1967).

CHAPTER 5: FUNDAMENTAL PERIOD ESTIMATION USING THE LARYNGOGRAPH

5.1.1 Introduction

The fundamental period epoch markers used in this thesis for the training and testing of the MLP-Tx algorithm were obtained using a system that uses the laryngograph signal (Lx). This involved two separate algorithms that were implemented in software as two different programs. The first consisted of an initial automatic analysis that generated a first approximation to the period markers. The second was an interactive program which provided the user a means to alter and correct the first set of period markers. Full operational details for both of these programs are provided in appendix A.2.

5.1.2 Automatic reference fundamental period estimation

The automatic reference fundamental period estimation algorithm defines the output in terms of the point of maximum positive gradient in the laryngograph waveform. It is acknowledged that this point corresponds to the point of closure of the vocal folds and in addition this point is more uniquely defined in each laryngograph cycle than other features such as the cycle maximum and minimum (Hess & Indefry, 1984). Figure 5.1 shows a typical example of the laryngograph signal and its corresponding differentiated signal. There are occasions when the differentiated signal is not well defined, as illustrated in figure 5.2. It is very important that the period marker values estimated by the reference algorithm are accurate reflection of the true period marker values for the speech. In addition, it is important that there are as few false period markers and missing period markers as possible. The algorithm used to automatically determine closures constitutes a simple expert system which then checks the validity of the initial estimates. The proximity of a period marker to adjacent markers is tested. In addition, a valid period marker can only occur after a local minimum in the laryngograph waveform, and before a local maximum. Another program was also written to permit manual checking of the automatic period marker labelling and permit changes to be made if necessary. The various processing stages of the first program are now described

in detail. A flow-chart for this program is shown in figure 5.3.

Stage 1:

The laryngograph signal is read in from a SFS file (Huckvale et al., 1988) and then bandpass filtered between 40Hz and 3kHz with a 251 point linear phase FIR filter, and the time-offset associated with it then corrected. This removes unwanted low frequency noise from the signal as well as high frequency noise.

Stage 2:

The bandpass filtered laryngograph signal is then differentiated. The laryngograph signal is then tested for correct polarity by separately summing the values of all the positive and negative peaks in the differentiated laryngograph signal. If the sum of positive peaks exceeds the sum of negative peaks, then the laryngograph waveform is assumed to be of correct polarity; otherwise it is inverted. The correct polarity bandpass filtered signal and the differentiated signal are then retained for future analysis. In addition, both these signals are written back to the SFS file to enable further analysis using an interactive program at a later stage.

Stage 3:

It has been observed that there is a tendency for the laryngograph waveform to die away rapidly towards the end of voiced segments (See chapter 2). Under these circumstances, it is very difficult to detect the excitation points, because the peaks in the differentiated laryngograph signal become very small and comparable in size to the background noise. As a consequence to this, the threshold that best detects the period markers is as low as the background noise on the laryngograph waveform will permit. Therefore, it is the characteristics of the noise that essentially determine the threshold.

The level of the background noise in the differentiated laryngograph signal is estimated by consideration of voicing-free regions of the laryngograph signal that have been previously labelled by hand. The mean μ_{dlx} and standard deviation σ_{dlx} of the instantaneous amplitudes of the differentiated laryngograph signal within the no-voicing regions are then estimated, giving a simple statistical description of the background

noise for the no-voicing conditions.

The use of the mean and the standard deviation is based upon the assumption that the noise is Gaussian and white. In this case, one can calculate the probability that the signal will exceed a given value in terms of the signal mean and standard deviation. The probability that the noise will exceed a given threshold corresponds to the area under the Gaussian curve for values greater than the threshold. Ideally it is required that the threshold used should give no false T_0 markers within the voice-free region. If we specify that we want no more than one false period marker in 20 seconds of speech (a typical length for a sentence) this leads to one error out of 160000 samples (using an 8kHz sampling rate). This correspond to a probability of error of around 10^{-5}. Writing the threshold as

$$\text{threshold} = A\sigma_{dlx} + \mu_{dlx}$$

Then the probability of an error is given by P_e, where

$$P_e \approx (1/2)\text{erfc}(A/1.414)$$

Where erfc(x) is the complementary error function for x. For a value of P_e of 10^{-5} this leads to a value of A of around 4.

Stage 4:

The generalized maxima that exceed the threshold value and occur more than a pre-defined minimum period value (±0.5ms) from other maxima are then calculated. This avoids the detection of multiple period markers around the point of maximum gradient that can otherwise occur if the signal is very noisy. This puts a limit on the maximum operating frequency of 500Hz, which is sufficient for the current application. If the threshold has been well chosen by the previous stage of processing, the generalized maxima will constitute the final period markers, with only a few exceptions.

Stage 5:

A potential period marker is then rejected if it does not have a local minimum (within 20ms) in the laryngograph waveform preceding it and a local maximum in the laryngograph signal following it. If more than one marker shares the same maximum and minimum in a laryngograph cycle, only the period markers with the largest local maximum in the differentiated laryngograph waveform is retained.

Stage 6:
Potential period markers are also rejected if they are separated from other period markers by a predetermined range of 20ms or more. This has the effect of removing any spurious period markers. It also limits the lowest operating frequency to 50Hz, which is again sufficient for the current application.

stage 7:
The period marker values are finally written out to another item in the SFS file.

5.1.3 Interactive fundamental period estimation algorithm

An additional interactive program provides a means to hand-correct regions of speech and laryngograph erroneously labelled with fundamental period markers by the automatic period estimation program. It also provides the means to label sections of speech as ambiguous, so that they can be ignored and not be used for training and testing of the MLP-Tx algorithm. This latter point is important, because occasionally there are regions of speech and laryngograph signals which are difficult to interpret, and it is consequently better to exclude them from future analysis rather than permit the possibility of using falsely labelled speech data. To deal with these discrepancies, there is an option on the interactive laryngograph analysis program to permit the user label certain regions as "rejected" so that later they will not be used for training the MLP-Tx algorithm or in performance evaluation tests. Figure 5.4 shows a typical operating window seen by the user whilst using the program.

The program operates by displaying the speech pressure waveform, the bandpass filtered laryngograph waveform, the differentiated laryngograph waveform, and the preliminary

estimate of the period markers generated by the automatic analysis program. In addition there is another trace representing the output from the interactive program, which consists of a set of period markers that are initialized from the automatic analysis program. The program lets the operator zoom in and examine the waveforms in the required detail. In addition, the user may select a new threshold on the differentiated laryngograph waveform, and re-run the analysis over the selected region. In this way, period markers may be added or removed as desired.

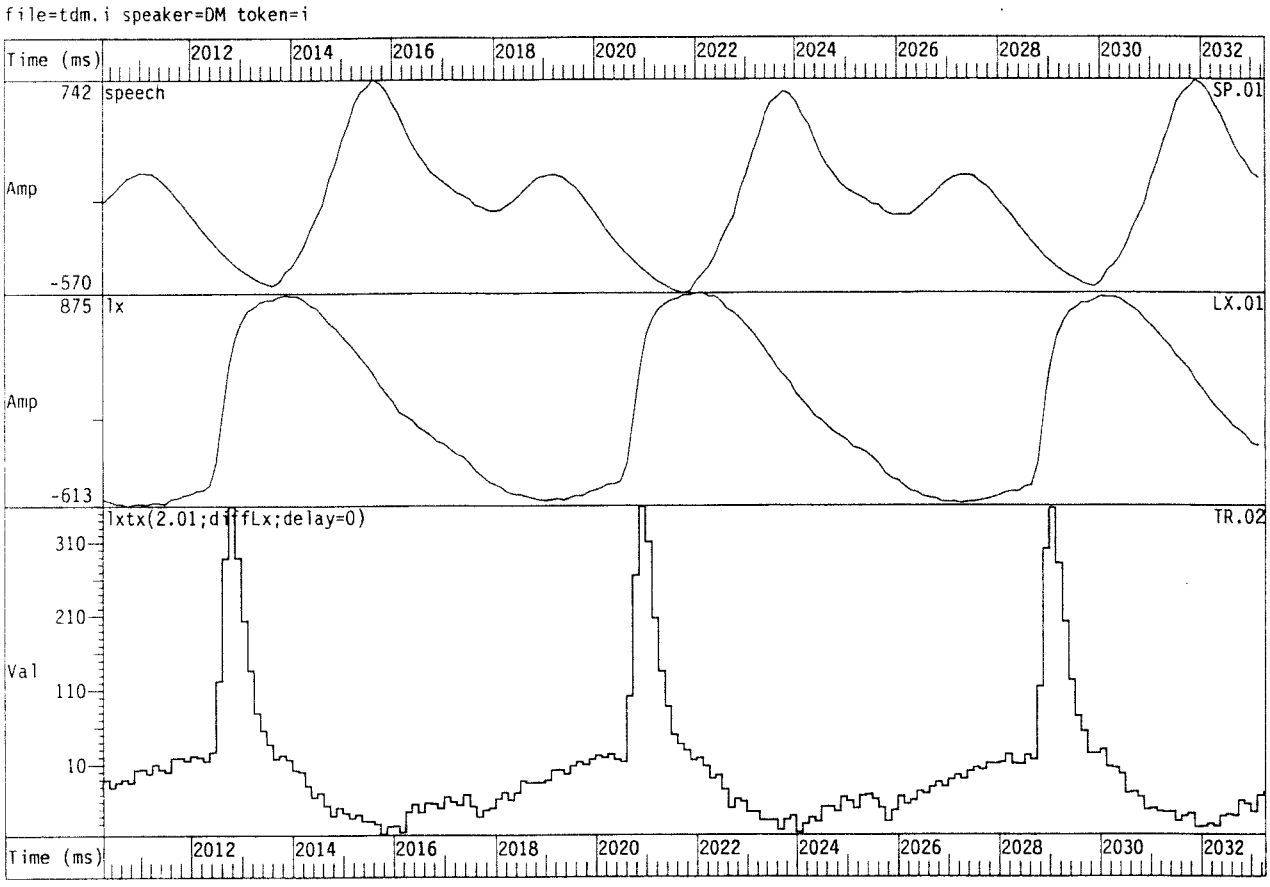


Figure 5.1 Laryngograph waveform exhibiting single well-defined peak differential per excitation period.

The speech pressure waveform, corresponding laryngograph and differentiated laryngograph waveforms are shown. The utterance is the vowel /i/ spoken by a male subject.

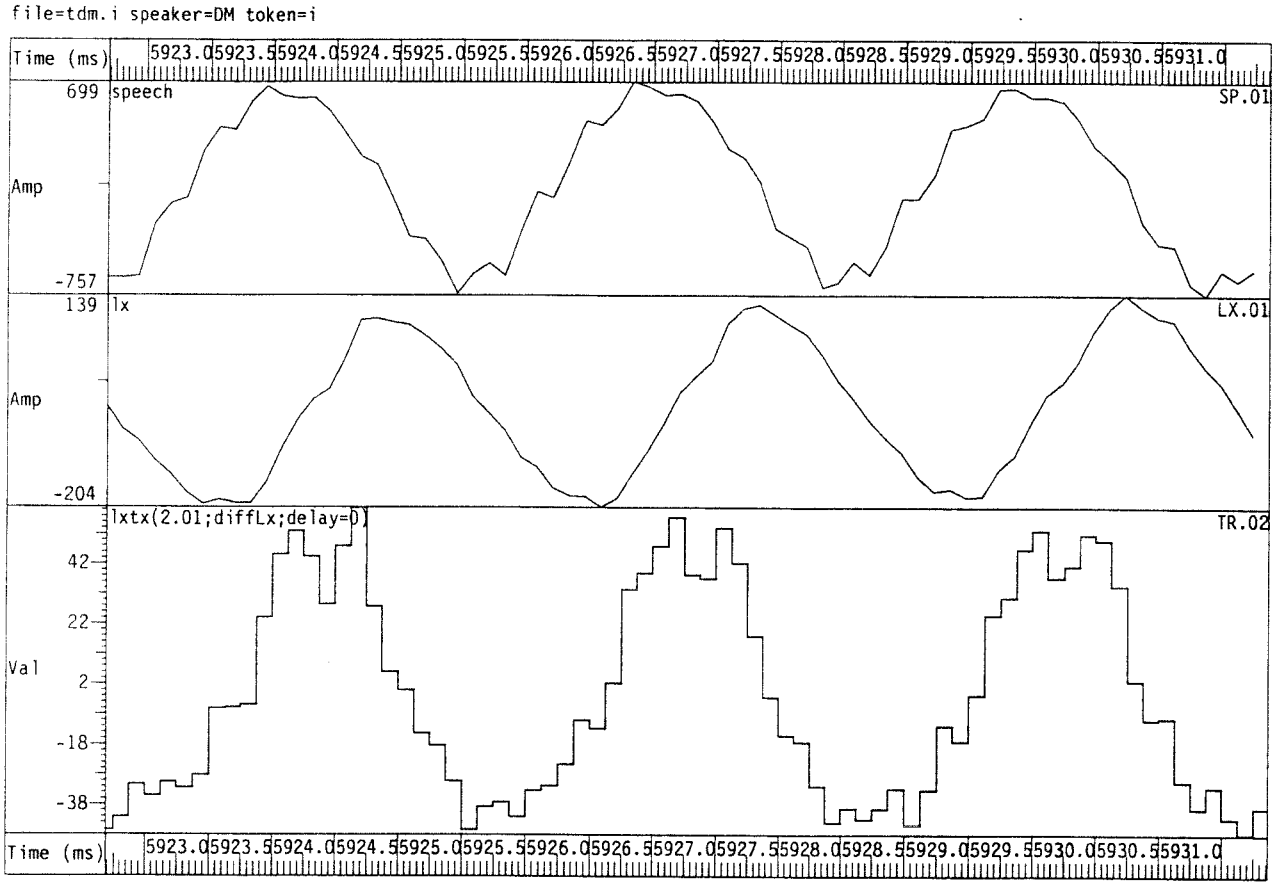


Figure 5.2 Laryngograph waveform exhibiting poorly defined peak differential per excitation period.

The speech pressure waveform, corresponding laryngograph and differentiated laryngograph waveforms are shown. The utterance is the vowel /i/ spoken by a male subject.

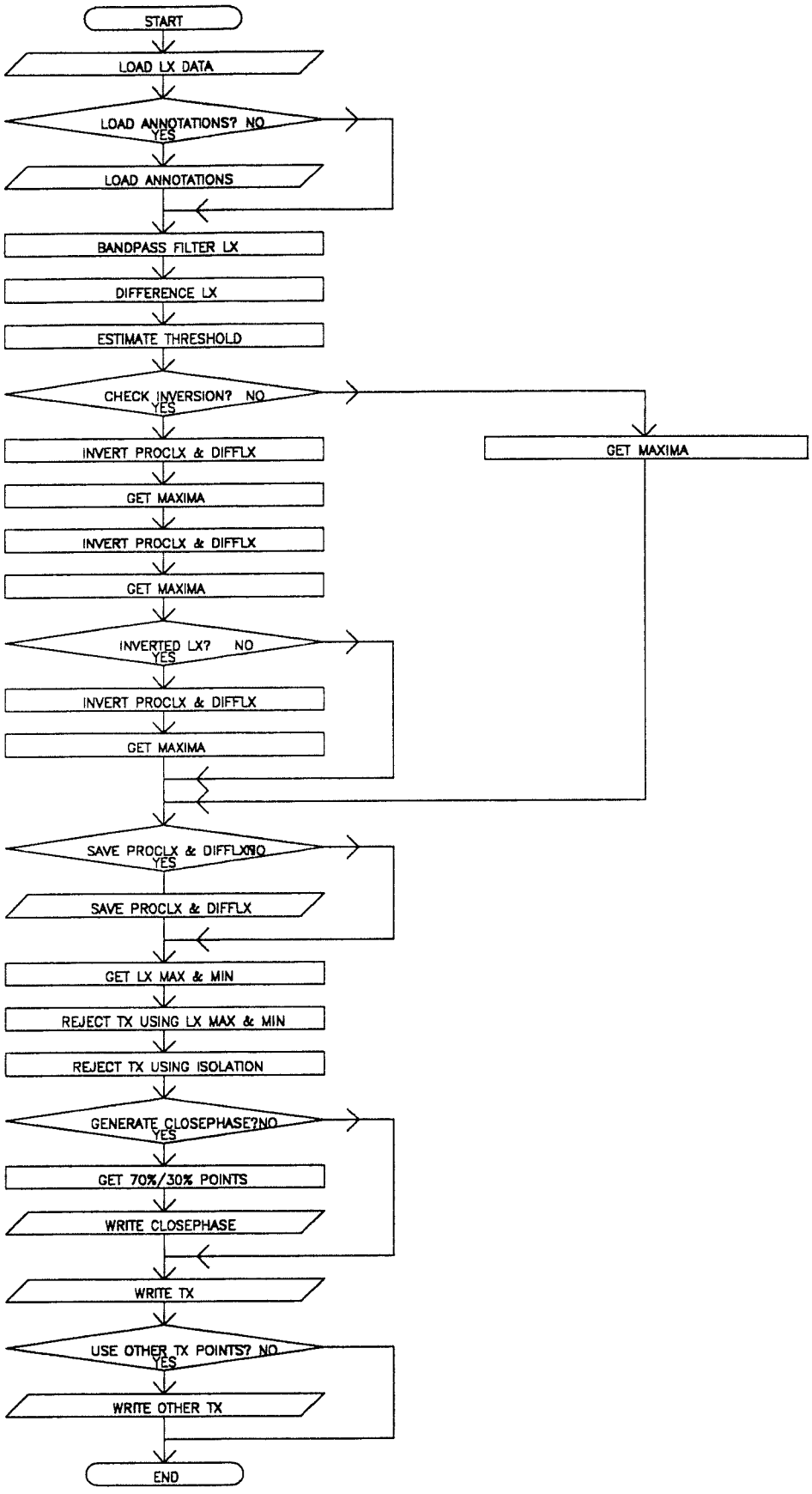


Figure 5.3 Flow-chart of the operation of the automatic fundamental period estimation program.

This program (lxtx) was used to automatically determined the period markers on the basis of the laryngograph waveform.

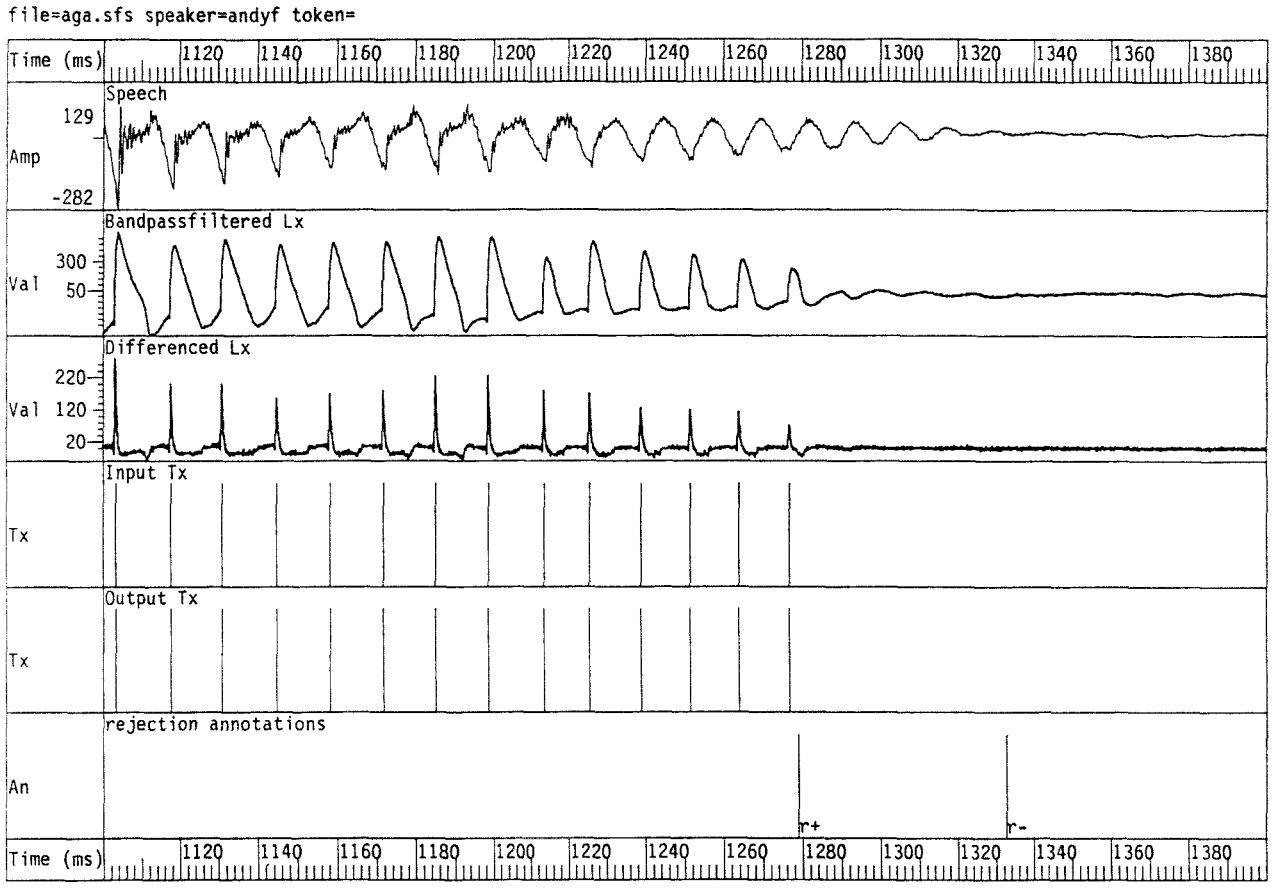


Figure 5.4 Typical operator's view using the interactive period marker estimation program (lxia).

The top trace shows the speech waveform. Below it are the band-pass filtered and differentiated laryngograph waveforms. Next are the period markers from the automatic analysis, followed by the period markers from the present analysis. In the bottom of trace the rejection annotation labels are displayed, which can be set to label certain regions of the signal unreliable and not suitable for future analysis. The utterance is the end of /aga/ spoken by a male subject.

CHAPTER 6: TECHNIQUES FOR COMPARING FUNDAMENTAL FREQUENCY/PERIOD ESTIMATION ALGORITHMS

6.1 INTRODUCTION

6.1.1 Organization of this chapter

This chapter investigates some established techniques for evaluating fundamental period and frequency estimation algorithms. Some of the standard approaches to the evaluation of fundamental frequency algorithms are discussed. The implementation of some frequency contour comparisons are then described in detail. Two newly developed measurements specifically for the comparison of period-by-period (time-domain) algorithms are then introduced and details of their practical implementation are described. Finally, some difficulties in making comparisons are considered.

6.1.2 The need for quantitative comparisons of performance

To assess the performance of any fundamental frequency or period estimation algorithm, it is necessary to have evaluation techniques. In general these involve some kind of comparison between the results from the test algorithm and those obtained from a reference algorithm that represents "ideal" performance, or is an accepted standard.

It is desirable to be able to quantify the performance of an algorithm for several reasons. Firstly, one often wishes to know how well a new technique compares with established techniques, over a given range of conditions. Without this information, there is no way of knowing whether or not a new algorithm is of practical value.

Secondly, evaluation techniques provide the means to test modifications to an algorithm. If the performance of an algorithm can be estimated, the effect that alterations of parameters have on its performance can be monitored.

Although the assessment of fundamental frequency and fundamental period estimation

algorithms is an important issue, little work has been reported on quantitative comparisons in the literature. The best known study to compare speech fundamental frequency algorithms is due to Rabiner et al. (1976). Evaluations were also made by Reddy (1967), Tucker & Bates (1978), Friedman (1979) and Howard el al. (1986). A subjective study was carried out by McGonegal et al., (1977) which involved perceptual assessment of speech re-synthesised from the estimated frequency contours.

6.1.3 Fundamental frequency and fundamental period comparisons

There is a basic difference in format between fundamental frequency estimates and fundamental period estimates. Fundamental frequency estimates typically constitute a set of frequency values at regular time intervals, determined by the frame-rate of the algorithm. Fundamental period estimates typically consist of a sequence of time values corresponding to the location in time of successive period markers. If comparisons are made **directly** on such estimates, it is clear that different techniques will be required in the two cases.

In chapters 3 and 4 it was explained that fundamental period estimation provides a more complete description of vocal fold vibration than the sampled fundamental frequency, because the former has the capability to retain period-by-period irregularities. Similarly, direct comparisons on period markers are desirable because they can operate on a marker-by-marker basis.

Of course, it is always possible to estimate the frequency contour associated with a sequence of fundamental period markers. However, such a conversion discards the timing information concerning precise marker location in each speech cycle (which is important when the markers are used for period synchronous analysis; see chapters 3 and 4). In addition, the conversion procedure may smear the effect of irregularities in the excitation, especially if the frequency contour is smoothed before sampling at the desired frame rate (which is of course necessary to prevent aliasing of the contour).

6.2 ESTABLISHED COMPARISON TECHNIQUES

6.2.1 Frequency contours

A frequency contour is simply a plot of the estimated fundamental frequency against time. A logarithmic frequency scale is often used, because it better reflects the perceptual importance of frequency changes. In this thesis, the frequency estimates are plotted using a sampling rate around 100 Hz. Some examples of frequency contours are shown in figure 8.11.

If the frequency values are derived from period values, the frequency point can be plotted synchronously with these period values. However, if frame-by-frame comparisons are to be made, it is necessary that both contours are defined at the same frame rate, so this approach cannot in general be adopted.

6.2.2 Visual comparison of frequency contours

Perhaps the simplest and most widely used technique for speech fundamental frequency estimators involves the visual inspection of the frequency contour. The contour can provide useful insight into the operation of the algorithm. If the frequency contour is displayed together with other information, such as the speech pressure waveform, the output of a laryngograph, or a reference frequency contour, the basic operation of the algorithm can be quickly assessed. Gross errors and voicing determination failure show up quite well. However, errors due to small differences between contours are more difficult to judge.

6.2.3 Frequency histograms

Another simple technique for presenting the results from an algorithm involves the generation of histograms of the frequency values (the reciprocal of the fundamental periods). Abberton, Howard & Fourcin, (1989) used 128 bins in the histogram organized on a logarithmic frequency scale covering the range of 30 - 1000 Hz. Again one may visually compare results obtained for the reference and test algorithms. These plots are given the name Dx at UCL. Frequency histograms are shown in appendix A.8

In addition to visually assessing the form of the frequency histograms, it is also sometimes useful to calculate the mean, mode and median frequency values in the associated data.

6.2.4 Problems with subjective measurements

The visual assessment of frequency contours is subjective. That is, it depends upon the opinion of an observer, and different observers can give different results. This is clearly undesirable, because it introduces uncertainty and bias, and makes the results difficult to repeat. In addition, although it may be possible to **rank** a set of results from different algorithms, it much more difficult to come up with an absolute performance figure for a given result.

Another problem with subjective comparisons is that they are not automatic, and may require a large amount of human observation time. Automatic assessment is more desirable because it reduces the human effort, making it possible to run very large comparison experiments that simply would not be practical otherwise.

Therefore, rather than use subjective comparisons, it is better to precisely define a set of measurements in terms of mathematical operations on the estimates. This requires that the measurements really do relate to the important aspects of algorithm performance, which will be different for different applications. For example, in a lip-reading task supplemented by real-time speech period estimation from a set of different algorithms, Rosen et al. (1982) found that the inherent time-delay due the algorithms was a dominant factor in their performance ranking.

6.2.5 Quantitative comparison of frequency contours

Rabiner et al. (1976) defined four types of errors that can occur. These are also illustrated in figure 6.1 (the gross error criterion is discussed in the next section).

Gross and fine errors

1) Gross frequency errors. These are due to dramatic failures of the algorithm, and can include instances in which the frequency values obtained fall outside the normal range of values that occur in speech.

2) Fine frequency errors. These constitute small deviations of the frequency estimates from their true values.

3) Voiced-to-unvoiced errors. These occur when the test algorithm fails to generate an output to signify that the input speech was voiced when it was voiced.

4) Unvoiced-to-voiced errors. These occur when the test algorithm indicated that the input speech is voiced when in fact it was not voiced.

The last two types constitute errors in the determination of voicing of the speech signal which may be important if the fundamental frequency algorithm is providing information needed to distinguish voiced from voiceless sound, as in the case of the EPI signal processing hearing aid.

Rabiner et al. defined a frame-by-frame error as the difference in frequency between the frame value from the test algorithm and the frame value from the reference algorithm, for those frames in which both algorithms indicate voicing is present. Thus

$F_{error}(n) = F_t(n) - F_r(n)$, if $F_t(n)$ and $F_r(n)$ are non-zero.

where $F_{error}(n)$ is the error in frequency at frame n between the value of test frequency contour $F_t(n)$ at frame n and the value of the reference contour $F_r(n)$ at frame n. $F_r(n) = 0$ is the conditions for unvoiced speech. If the value of $F_{error}(n)$ exceeds a given bound, then the error at frame n is considered to constitute a gross error. Rabiner el al. (1976) used the criterion whereby a gross error constituted a difference of greater than 1ms between the test and reference fundamental period estimates. Tucker & Bates (1978) defined a gross error as one in which the difference exceeded 10%, Reddy (1967) used a 12.5% difference and Friedman (1979) used a 25% difference. In this thesis, a

difference of greater than 10% was considered to constitute a gross error, because it corresponds better to the logarithmic scale used to represent frequency contours.

If the value of $F_{error}(n)$ is less than this bound, then it is considered to constitute a fine error (frames with a zero $F_{error}(n)$ value are also classified as fine errors). The mean and standard deviation of the fine errors are then calculated and this gives an indication of the accuracy with which the test algorithm generated its frequency estimates.

Reddy (1966) used the terms hops, holes and chirps to describe a time-domain algorithm misplacing, missing or adding an additional marker in its estimates of speech fundamental period. A further classification of gross errors that is used in this thesis is based on the last two descriptions. A "chirp" error is defined to be a gross error in which the test frequency value exceeds the reference value by more than the preset amount, and a "drop" error is a gross error in which the test algorithm gives a frequency falling below the value of the reference by more than a preset amount. This sub-classification of gross error provides more information to the nature of the performance of the test algorithm.

Voicing transitions errors

In addition to determining distances between the frequency contours, Rabiner el al. also define unvoiced-to-voiced errors whenever

$F_r(n)$ is equal to 0 and $F_t(n)$ is not equal to 0

That is, when the test algorithm indicates voicing but the reference does not, there is an unvoiced to voiced error. Similarly, voiced to unvoiced errors occur whenever

$F_r(n)$ is not equal to 0 and $F_t(n)$ is equal to 0

That is, when the test algorithm indicates no voicing, but the reference indicates voicing, there is an voiced to unvoiced error (such measures are also of value in the evaluation

of voiced/voiceless detectors).

6.2.6 Implementation of frequency contour comparisons

A set of frequency contour comparison metrics were implemented by the author. The use of this program and the output generated is given in appendix A.10. The details of the comparisons implemented are now explained.

Check frame rates

The first stage in the comparison is to check that the two frequency contours are specified at the same frame-rates. Naturally one cannot make comparisons if the two sets of frequency values are not defined at the same points in time. A standard rate of 100Hz was used for all the comparisons in this thesis.

Estimate time difference between test and reference contours

Secondly, the time-delay between the reference and the test frequency contours must be known. It makes no sense to compare frames if they correspond to different times of the input speech. If the time delay is known a priori, it can simply be entered directly to align the two contours. If not, then it must be calculated. To perform this task, the standard deviation of the values corresponding to the differences between two frequency contours is calculated over a range of delays (±500ms was used). The minimum point in this time function is then located, which usually corresponds to the best time alignment of the two contours. This time function for a 20 second piece of evaluation data is shown in figure 6.2. The minimum is typically well defined. This procedure can reliably align the test and reference contours, unless the test contour contains a large number of gross errors, or the search range used is too large. To be sure that the time alignment was always correct, the time alignment for all the data files known to have the same delay were examined. The modal value of the delay was then used to re-align the test and reference data in those few cases in which the algorithm had failed.

Calculation of errors in voicing determination

Since a voiced-to-unvoiced error occurs whenever the test frequency contour indicated the absence of voicing and the reference indicated voicing, the maximum number of such errors is equal to the number of voiced frames in the reference frequency contour. Hence the voiced-to-unvoiced errors were expressed as a percentage of the voiced frames in the reference frequency contour.

Since a unvoiced-to-voiced error occurs whenever the test frequency contour indicated voicing and the reference indicated the absence of voicing, the maximum number of such errors is equal to the number of unvoiced frames in the reference frequency contour. Therefore the unvoiced-to-voiced errors were expressed as a percentage of the unvoiced frames in the reference frequency contour.

Calculation of gross errors

A gross error occurs whenever both the reference and test frequency contours indicated voicing, and the frequency value of the test frequency contour deviates by more than ±10% of the frequency value of the reference frequency contour. These errors were then expressed as a percentage of the number of frames in the test and reference frequency contours that were both voiced at the same time. It was a trivial task then to classify the gross errors into those in which the test value is greater than the reference value and those that are less than the reference value. This then gives an indication to whether the errors are 'chirps', which occur when there is a local false rise in frequency, or 'drops', which occur when there is a local false drop in frequency.

Calculation of fine error statistics

A fine error occurs whenever both the reference and test frequency contours indicated voicing, and the test frequency contour deviates by less that ±10% from the reference frequency contour. These errors were then expressed as a percentage of the number of frames in the test and reference frequency contours that were both voiced at the same

time. The mean and standard deviation of the fine errors were then calculated.

Calculation of contour statistics with respect to different labels

The reference frequency contour used for the comparisons could also be annotated with labels to indicate the different sound types, or its local reliability. These annotation labels could then be used by the comparison program so that the metrics described above could be calculated for each label class. In this way it is possible to determine the performance of the test frequency contour with respect to different types of input sounds. More importantly, this facility was used with labels that indicated when the reference frequency contour was reliably defined, so that meaningful statistics could be generated.

6.3 NEW COMPARISON TECHNIQUES

6.3.1 Advantage of period marker comparisons

Since the work in this thesis was mainly concerned with speech fundamental period estimation, it was felt worthwhile to consider comparison techniques that could be specifically used for this type of fundamental period determination algorithm.

There is advantage to be gained from making comparisons directly on the period markers, because the process of converting the period markers to a fixed rate frequency contour inevitably loses information concerning the speech excitation. In particular, comparisons in terms of the period markers can give information concerning the absolute location of the excitation point in a speech cycle. This information will not be present in a frequency contour sampled at a fixed rate. Also, the conversion of the period markers to frequency values loses information concerning the type of errors that occurred in the fundamental period estimation algorithm. For example, if an extra period marker is inserted, this constitutes a fundamental period chirp error. However, after converting the period markers to a frequency contour, the effect of a single period marker insertion may well be smeared out by any smoothing process that is employed

over the given frequency frame and the insertion error may only appear as a fine error in the frequency contour comparison (although this depends on the averaging window size and the definition of a gross error).

6.3.2 Period marker comparison metrics

Given a set of test fundamental period markers and a set of reference fundamental period markers for the same piece of input speech, we wish to quantify the difference between them. To be more specific, one wishes to know whether the correct number of test period markers are present and located accurately. This calculation is not easy to perform directly, as was in the case of frequency contours, because the two sets of period markers are not generally defined at the same points in time. In addition, there will also in general be a gross time difference between the test and reference period markers. The required measurements between a set of reference period markers and test period markers are illustrated in figure 6.3.

Hits, misses and false alarms

The period marker comparisons used in this thesis operate by firstly finding the correspondence (if any) of a reference period marker to a test period marker. That is to say, for a given period marker in the reference set, the period marker in the test set that corresponds to the same speech excitation is determined (the procedure to perform this operation in explained shortly). After this operation has been carried out, it is then possible to say whether or not there is a test period marker that corresponds to a given reference period marker. If there is, this constitutes a 'hit' and if there is not, then this constitutes a 'miss'. For all the 'hits', it is then possible to calculate the deviation (in time) between the given reference period marker and its corresponding test period marker. This 'jitter' is a measure of the location accuracy of the test period marker, and for a given piece of speech, the mean and standard deviation can be calculated to give an indication of overall test period marker location accuracy.

Absolute marker jitter and period jitter

In addition to this 'absolute time difference jitter' between the test and reference period markers, it is also possible to calculate the difference between the local period values as determined from the test and reference data sets. This measure has the advantage that it cancels out any slowly varying time shift between the reference and test period markers that would show up in absolute time jitter measurements. Such an effect will occur when a speaker is moving his head relative to a microphone whilst using the laryngograph to generate the reference period markers. This will interfere with the absolute measurements, but not affect the period difference measurements. It is often the time difference between successive period markers that is required of a fundamental period estimation algorithm, rather than the absolute time locations, because it is this that relates to the fundamental frequency.

6.3.3 Dynamic programming alignment of test and reference period markers

Comparisons between the test and reference period markers rely on being able to determine the correspondence between a period marker in the reference set and the test set. As mentioned briefly before, there will often be an overall time-shift between the test and reference period markers, arising from the different operation lags between different algorithms. In addition, test pulses may be missing in certain locations and present in others. Even when the test pulses are present, they will not always occur at exactly the same time as the corresponding reference period markers. There will also often be false pulses in the test period marker set. Clearly an algorithm that is to find the correspondence between the two sets of markers must be robust enough to take all of these difficulties into account.

Constant time shift alignment

Before discussing the full alignment problem, let us briefly consider what one would do in the case of two sets of markers between which there was only an single overall time-shift. The simplest operation that would indicate the lag between the two sets of markers would be the cross-correlation function. That is, the point of maximum coincidence as a function of time lag between the two sets of markers could be used to

identify the time lag. In the case of period markers, which can be represented in time as a sequence of impulses, the multiplications in the cross-correlation can be replaced by a logical AND function. It will be appreciated that if there is only a constant offset between the two sets of markers, after the alignment has been performed, all 'hits' will align completely between the two sets of period markers, and so they can be identified by processing the list reference period markers and looking for a test period marker at exactly the same time location. The cross-correlation function for two sets of period markers for a 20 second piece of speech is shown in figure 6.4.

Dynamic time-warping alignment

The same basic principle can be extended using dynamic programming to find the correspondence between the reference and the test period markers, when each test marker is independently shifted in time from the reference period marker location. Initially the best constant delay alignment is performed. Then each period marker in the reference set is individually correlated (or rather ANDED) with the test period markers over a suitable range around its original location of ±10ms, and the coincidences recorded whenever they occur. This is illustrated in stage 1 of figure 6.5. This process results in a matrix; for each reference period marker the coincidence of test period markers is given for a range of lags. The next stage of operation involved finding the best 'path' through this matrix, using a dynamic programming procedure. This is illustrated in stage 2 of figure 6.5. The optimum path is defined as that which minimized deviation and maximizes the number of 'hits' that are detected. Figures 6.6 and 6.7 show the time warp path found for some real data.

After the non-constant alignment has been carried out, the 'hits' are then identified. The jitter between absolute marker locations is then determined, which corresponds to the deviation at which a given 'hit' occurred.

The false alarms are then identified. A distinction is made between those which occur outside voicing regions and those which occur within voicing regions. In this case, a voicing region is defined as a region within 20ms of any reference period marker. It

is believed that this classification of false alarms is valuable, because the false alarms during voicing often correspond to chirp errors.

The number of misses is also computed, and this is simply the difference between the number of reference pulses and the number of hits.

6.4 PROBLEM ARISING WITH COMPARISONS

6.4.1 The basic problem

To make valuable comparisons between different algorithms, it is important to bear in mind that the metrics that we have discussed so far constitute a set of interdependent measurements relating to the performance of a algorithm. For example, in the case of the frequency contour comparisons, to give an overall rating of a particular algorithm involves consideration of its voicing determination performance as well as the gross errors and fine errors it generates. If the criterion of detection of voicing for an algorithm is altered, the relationship between the hits and false alarms changes. In addition, the number of gross errors and the statistics for the fine errors may change. Therefore, for a given algorithm, a different set of numbers (for voicing errors, unvoiced errors, gross errors, etc) may be generated depending upon the setting of the voicing detection threshold. In the evaluation of such an algorithm, we are interested in its inherent performance, not its performance for an arbitrary threshold value. One solution to this problem is to set the threshold for all algorithms (whenever possible) to give similar voicing errors. This is now discussed in somewhat more detail.

6.4.2 Relationship between hits and false alarms

As in the case of any detection system, there is a compromise between the number of 'hits' and 'false alarms' made by a fundamental period estimation system. To appreciate this point, consider a fundamental period estimation system which is essentially composed an epoch detector followed by a comparator (for example the MLP-Tx

algorithm, as explained in chapters 8 & 9). The output from the first stage is a multi-level time-waveform, and it is the task of the comparator to convert this into a time value that indicates the location of the local peak of the time waveform, provided it is greater than a pre-set threshold value. Therefore, the comparator must distinguish inputs that are greater or less than a preset threshold.

The magnitude of the output from the pattern processor is related to the similarity of the input to a speech excitation candidate. In cases where there is a well defined excitation in the input speech, there will be a well defined response from the pattern processor. This is illustrated in the numerous plots of MLP output waveforms (for example, see figure 10.1). However, there will be occasions when the excitations are not so well defined (for example, figure 10.6). For example, the amplitude of the speech may be very low. Thus there will be situations where there is only a small output from the pattern processor, as opposed to a well defined output. Conversely, when no input excitation presented to the system, and there is no input noise, the output from the pattern processor will be low. However, noise will inevitably be present, especially in real operating conditions, and consequently sometimes there will be input noise that will result in an output pulses from the pattern processor. Therefore, the pattern processor output can be characterized by two conditional probability distributions: The output pulse height distribution conditional on there being an input excitation present, and an output pulse height distribution conditional on there being no input excitation present. The task of the comparator is then to divide this graph into peaks below and above the threshold. It is apparent that if there is any overlap between the two conditional distributions, then errors will inevitable be made in this process. Whether these errors are largely misses or false alarms depends upon the threshold. If a low threshold is used, then all the events will be detected, but many false alarms will also be made. If a harsh threshold is used, then few false alarms will be made, but many true events will be missed.

6.4.3 Receiver operating characteristic (ROC)

If one wishes to characterize the performance of the fundamental period estimation

system, it must first be appreciated that the performance of the fundamental detection process depends on the pattern classification section, while the threshold in the comparator merely affects the trade off between the hits and false alarms. If a set of hits and false alarms are obtained by testing the system at a number of different thresholds, a plot of these points is known as the receiver operating characteristic for the detector, and it is the position of this curve that characterises the performance of the detector (Levine & Schefner, 1981).

A measurement used in this thesis to characterise ROCs is the equal error criterion, that is the equal error hit and false alarm rate. This corresponds to the percentage hit rate that occurs when there are as many misses as false alarms.

As mentioned previously, a serious problem with comparisons between different algorithms is that they may operate at different points on their respective ROCs. That is to say, an algorithm may be set to respond to all excitations, and consequently generate many false alarms, whereas the others may give few false alarms, but miss many excitations as well. Consequently one cannot simply compare the hits and false alarm rates and decide which algorithm is best.

One possible solution would be calculate the ROC for each algorithm, which would then enable a comparison of the 'quality of detectors' to be made. In order to do this, the hits versus false alarms for each device over a range of thresholds would have to be calculated. At best this would be computationally expensive. In some cases, it may not be possible. For example, in a piece of hardware, such a threshold may not be accessible.

In some situations, it may not be necessary. For example, if one device gives a higher 'hit' rate and a lower 'false alarm' rate, then it is the better detector, since both measures indicate more desirable performance.

6.4.4 Setting 'hit' rate of test and reference algorithms to the same values

If it is possible to adjust one of the devices or algorithms, then all that needs to be done is to set it such that it has the same hit rate (or false alarm rate) as the other device. The performance of the other measure then indicates which of the two incorporates a better detector. Unfortunately, this procedure would involve running one algorithm for a range of thresholds and finding the hits and false alarms for each run, and this is very time-consuming.

6.4.5 Setting period marker count to the same as the reference period marker count

Another less computationally intensive normalization is to set both algorithms to generate the same number of output period markers (given by the number in the reference algorithm). This does not require running the computationally intensive period marker comparison algorithms, and requires much less processing. This technique was adopted to normalise the different fundamental period estimation algorithms in the final testing experiments.

It should be noted that some measures of algorithm performance are not directly affected by the operating point of the algorithm, because they scale with the number of detected excitation points. For example, the number of voiced false alarms is expressed as a percentage of the total detected period markers. Consequently, both are reduced by using a strict threshold. However, in most cases it is desirable to operate all algorithms at around the same percentage hit level.

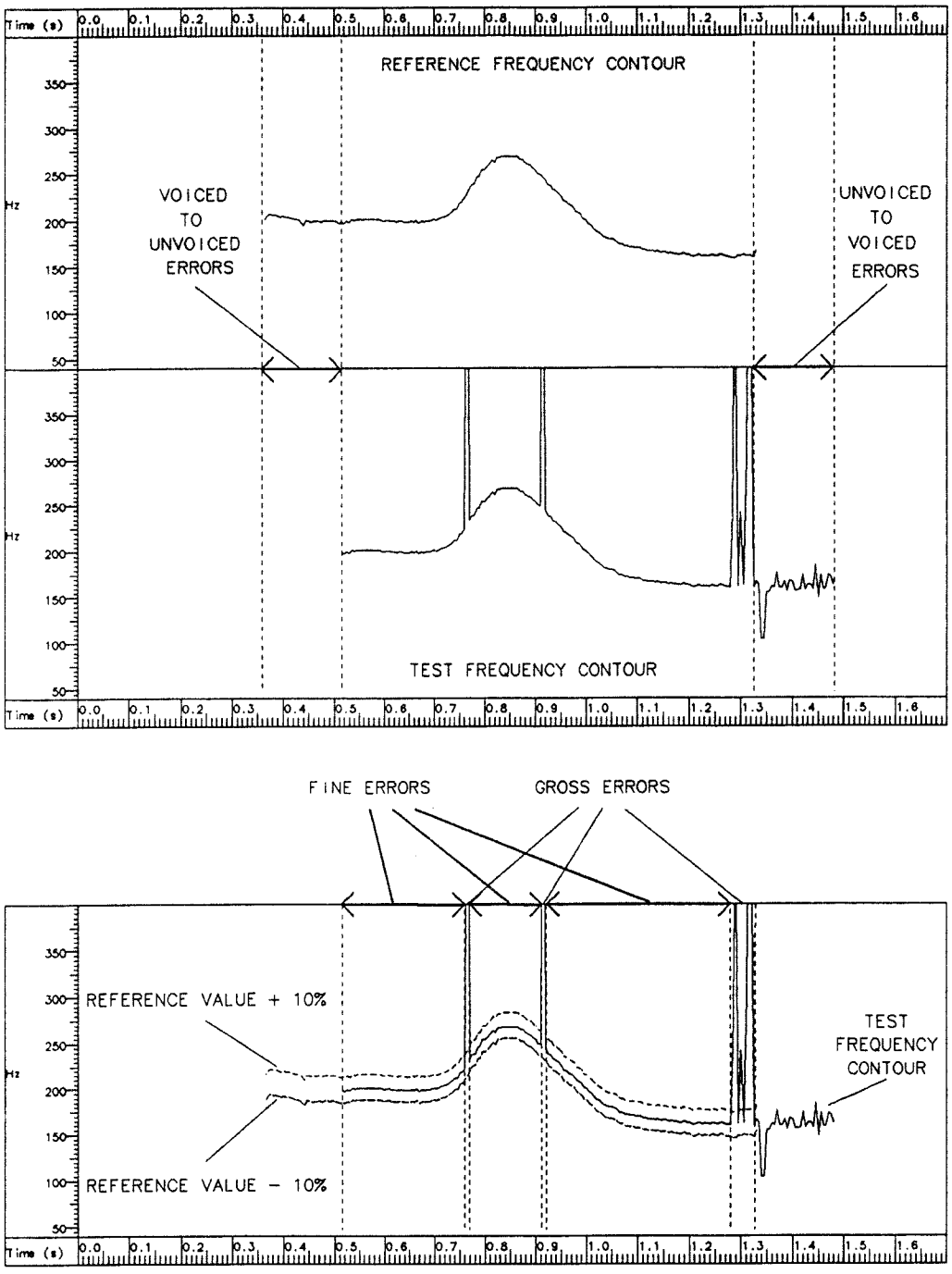


Figure 6.1 Illustration of types of errors used to quantify a test frequency contour. Upper box: Comparison between two frequency contours showing examples of voiced-to-unvoiced errors and unvoiced-to-voiced errors.

Lower box: Comparison between a test frequency contour and a reference contour to illustrate gross errors. Whenever the frequency values in the test contour deviates by more than 10% from those in the reference contours (a limit shown by the two sets of dotted contours) there is a gross error. In this examples, the gross errors exceed the reference values, and are consequently chirp errors.

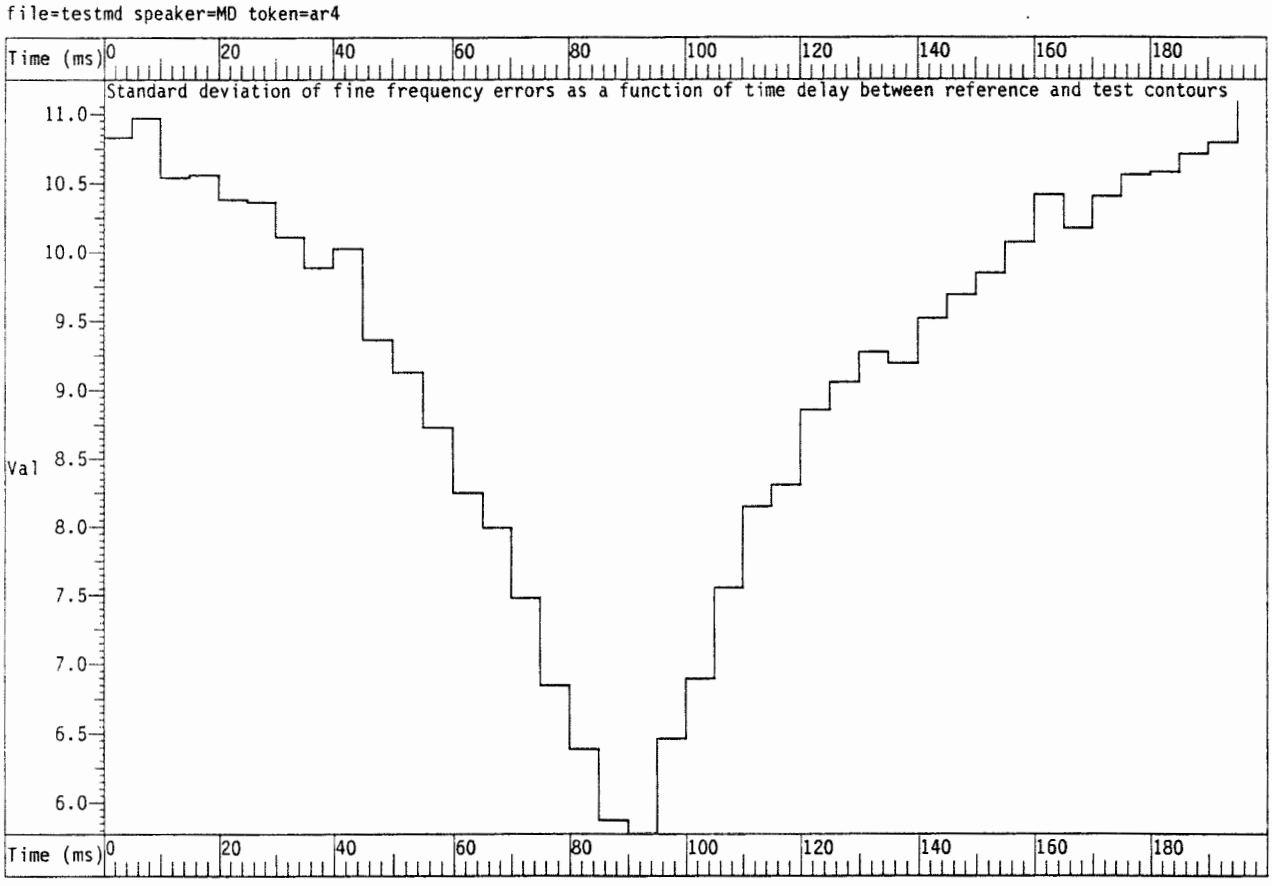


Figure 6.2 Effect of relative time delay on standard deviation of the fine frequency differences.

This function typically exhibits a well defined minimum and provides a means to align the two contours. This result corresponds to a 12 second section of female speech.

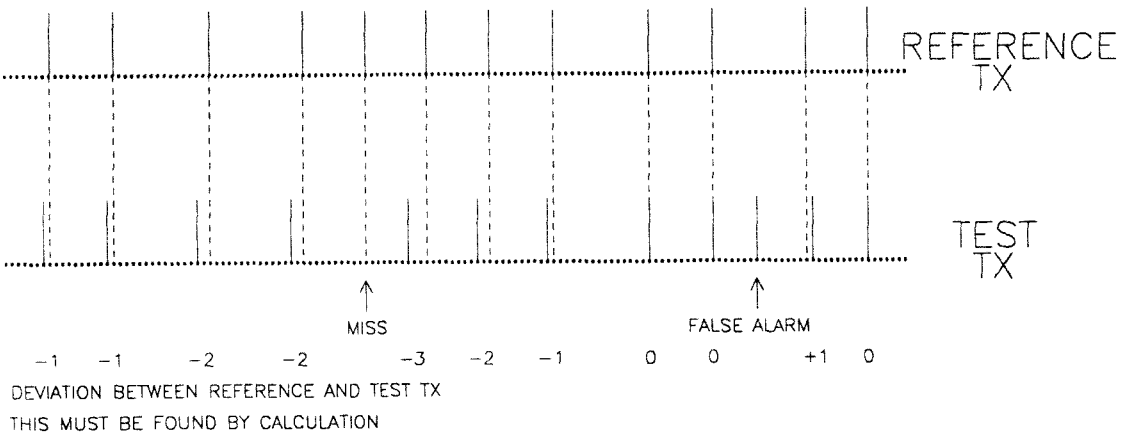


Figure 6.3 Illustration of a required information from period marker comparisons.

It is enlightening to know the number of hits, misses and false alarms generated by a test algorithm, and examples of these are shown. For the hits, the deviation between the test marker and reference marker gives an indication of the accuracy of the algorithm under evaluation.

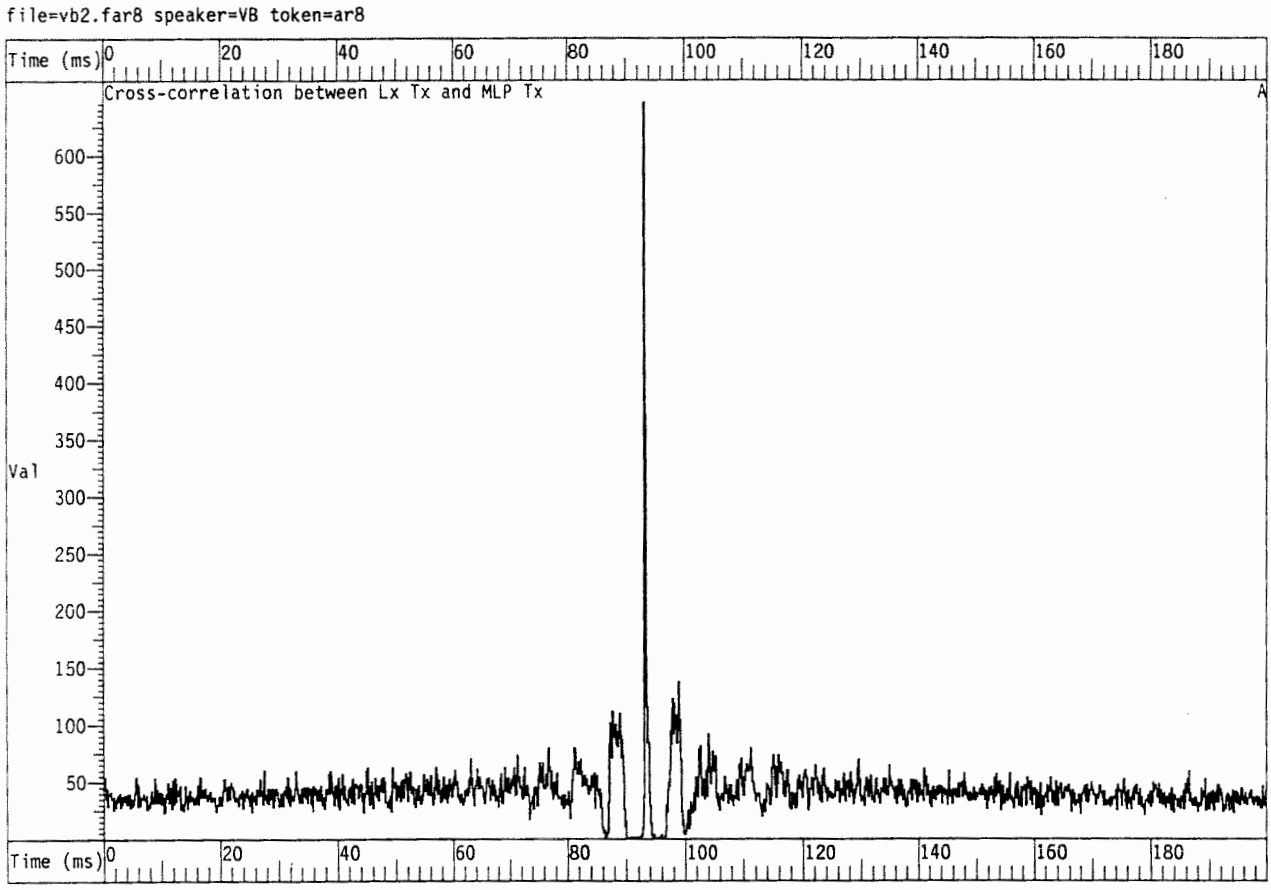


Figure 6.4 Plot of the cross-correlation between reference period markers and MLP-Tx test period markers.

The reference markers were derived from the laryngograph. It can be seen that in this example (which is typical for the test markers generated from the MLP-Tx algorithm) the peak is well defined, and its location provides information relating to the overall constant time-shift between the test and reference markers. This result corresponds to a 15 second section of female speech.

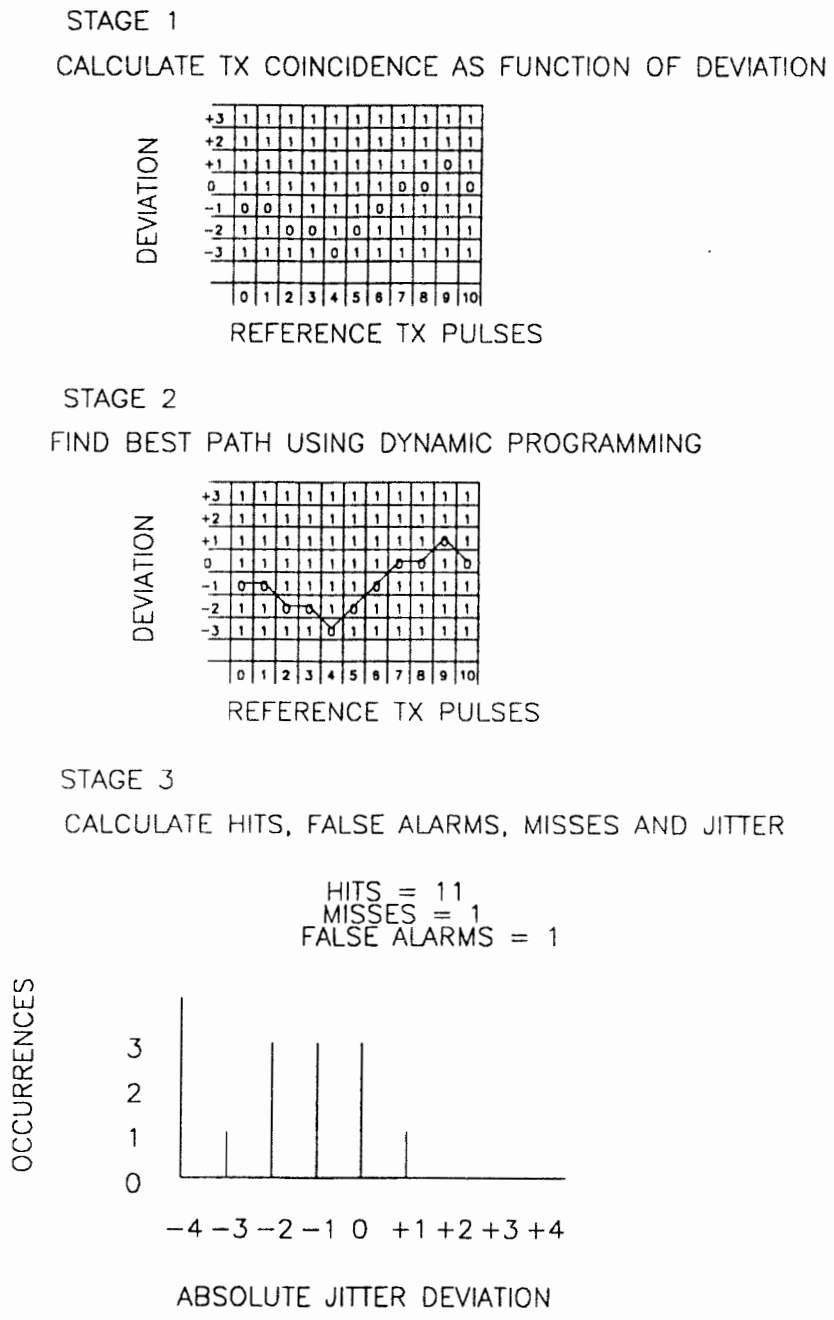


Figure 6.5 Illustration of the operational stages in the fundamental period marker comparisons.

Stage 1 involved determining the local coincidences of a period marker against test period markers. The results from this is shown in the first matrix. The next stage involved finding the best path through the deviation matrix, so that it is possible to assign corresponding reference markers and test markers to each other. After this has been performed, the number of hits is simply given by counting the reference markers that have corresponding markers, and the false alarms are the test markers without a corresponding reference marker. The local time difference (the jitter) between the test and reference markers is the warp value associated with each hit. Results of these operations are illustrated in stage 3 of the figure.

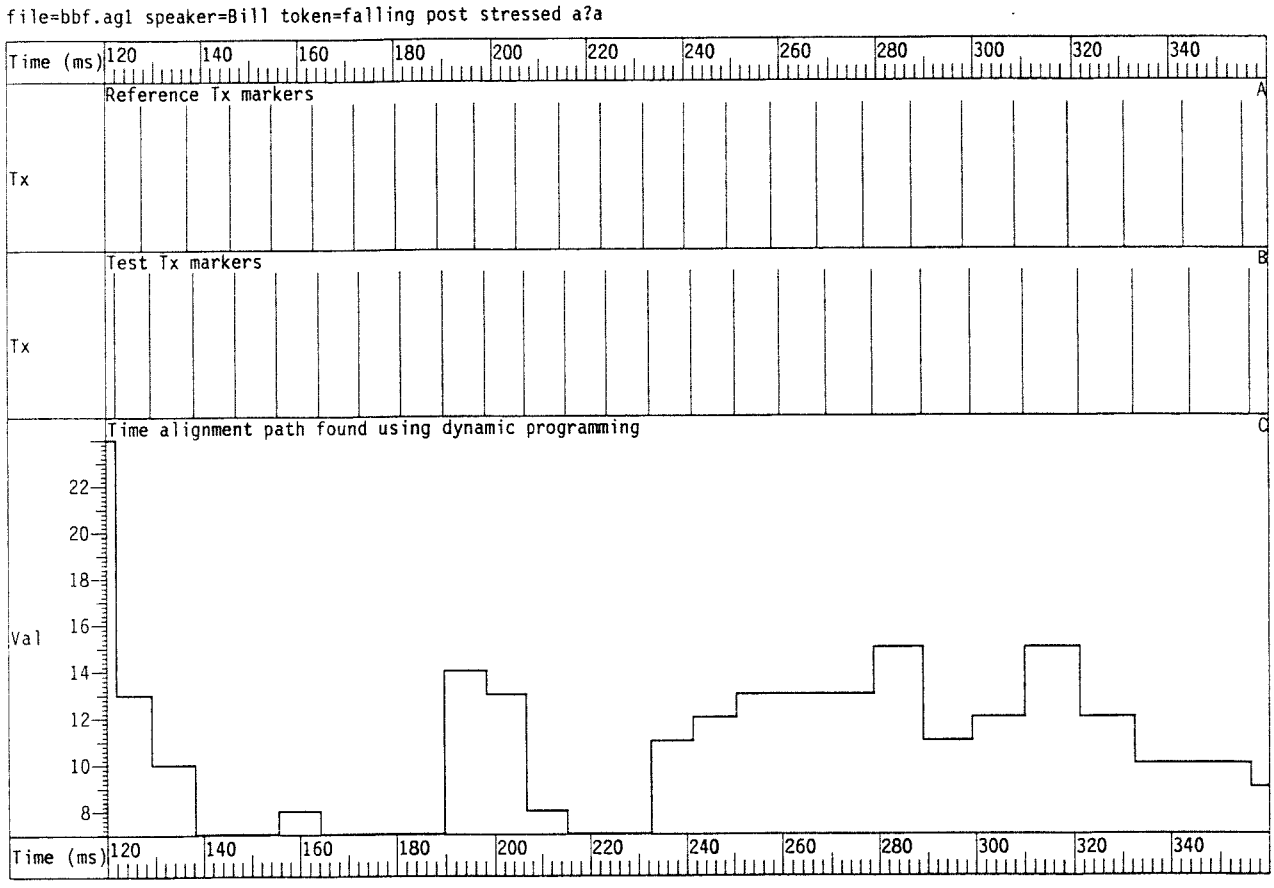


Figure 6.6 Illustration of the time-warp path resulting from real period marker data. Traces A and B show the reference and test markers respectively. Trace C shows the warp between the markers. It is plotted so that the value of the warp between a reference and test marker is defined from the last reference marker until the current marker. The speech in from a male subject.

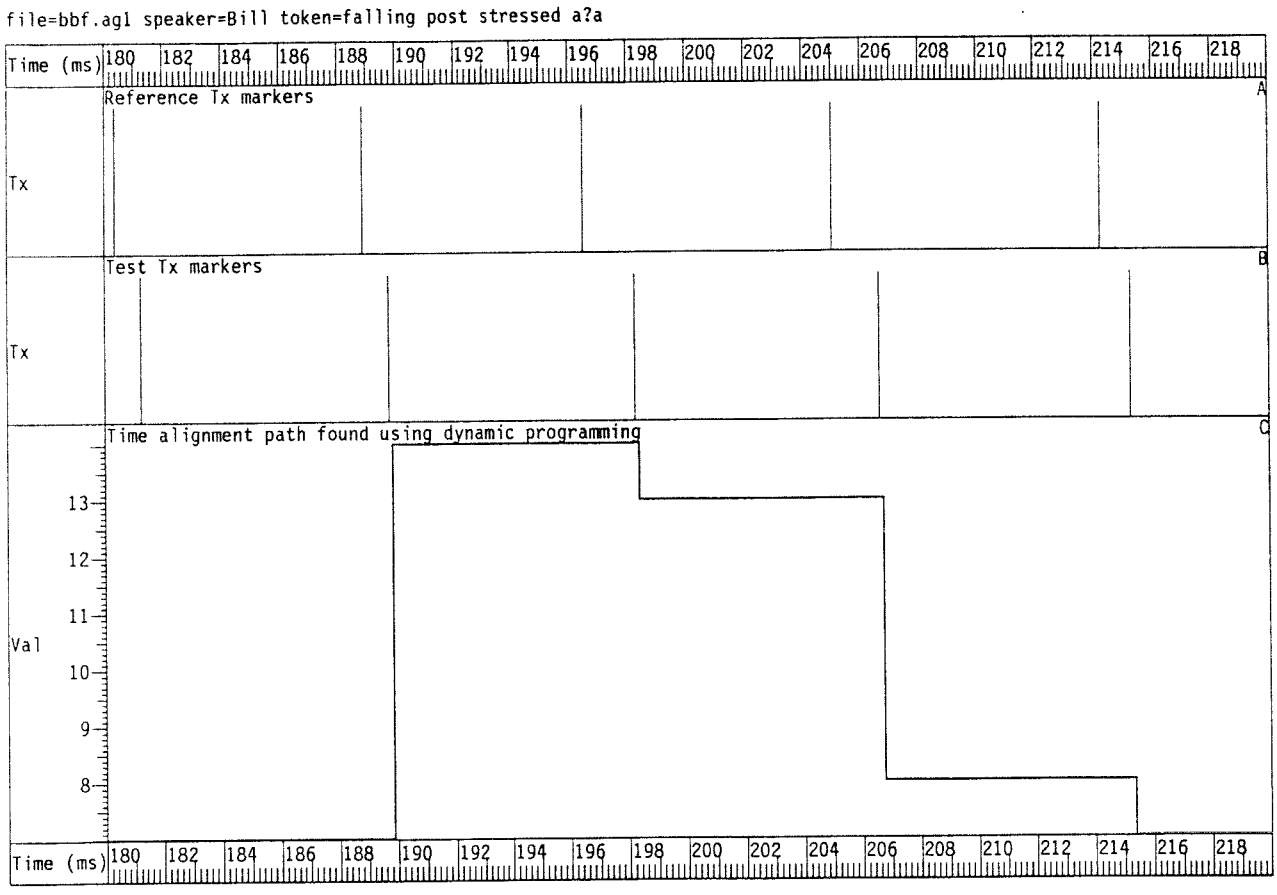


Figure 6.7 A close-up of the time-warp path shown in figure 6.6

CHAPTER 7: PATTERN RECOGNITION TECHNIQUES

7.1 BASIC CONCEPTS IN PATTERN RECOGNITION

7.1.1 Introduction

This chapter gives a brief overview of pattern recognition. First some of the basic issues and definitions involved in the recognition of patterns by computer are discussed, and then some classical techniques are described. This is then followed by a discussion of the more recent field of artificial neural networks, leading up to a description of the multi-layer perceptron (Rumelhart et al., 1986). This is considered in detail, because it is the pattern recognition technique that is employed in this thesis. It was chosen because it has been shown to perform well in many problems, including many in speech analysis (Boulard & Wellekens, 1987; Peeling & Moore, 1986; Huang & Lippmann, 1987; Elman & Zipser, 1987).

7.1.2 Definition of a Pattern

A pattern can be considered to be an ordered array of elements. It is typically generated as the result of the representation of a set of measurements or descriptions of an organized discrete phenomenon or event. Mathematically a pattern can be represented as a multi-dimensional vector with the components in the vector corresponding to the elements in the pattern. **Pattern recognition** is the process by which such vectors are classified into different categories. The system that performs this function is called a **pattern classifier**.

7.1.3 Supervised and unsupervised pattern recognition

Some of the basic issues in pattern recognition will now be considered in more detail. Pattern classifiers operate in two distinct modes, training (learning) mode and recognition mode (Tou & Gonzalez, 1974).

In training mode, a set of patterns vectors are made available. The goal of the training is to determine a decision boundary that will indicate the class membership of pattern vectors. The use of training material that is representative of the task is important to achieve good performance of the final system.

In recognition mode, an input vector is presented to this trained classifier and its output class is then estimated.

Approaches to pattern recognition can be sub-divided into those which are supervised and those which are unsupervised, depending upon how their training is carried out. Supervised training involves the use of a "teacher" to indicate explicitly the class of the training examples. In unsupervised training, there is no teacher present, so there is no explicit way in which a classifier can know the class of a given input pattern. In this case, the classifier must apply some general principle to an input pattern, and group it with other input patterns with similar characteristics. The classifier typically adapts its response so that patterns that are similar become clustered together, whereas patterns that are grossly different form separate clusters

In this thesis supervised pattern recognition is used because of the need to train a classifier to detect a specific event, the occurrence of which can be precisely defined in advance on the training data. Therefore, we will mainly consider supervised rather than unsupervised techniques in the discussion in this chapter.

7.1.4 Geometric interpretation of patterns and pattern recognition

Some important aspects of pattern classification can be understood by viewing the problem in geometric terms. An input pattern vector can be considered to be a point in multi-dimensional Euclidean space, where the dimensionality is determined by the number of elements in the pattern vector. A pattern classifier can then be seen as a system that divides this input space into a given set of discrete regions, which correspond to different pattern categories. The surfaces that divide the points in this input space are known as decision surfaces.

It is possible to represent the decision surfaces for a classifier in terms of a set of decision functions $\{G_i(\mathbf{X})\}$. These functions are typically defined such that for a data vector **X** of pattern class i;

$$G_i(\mathbf{X}) > G_j(\mathbf{X})$$

for all j = 1,2,...,M pattern classes, where j is not equal to i. That is to say, for data vector **X** belonging to pattern class i, the decision function corresponding to class i has a value greater than all the other decision functions for all the other classes.

The simplest type of classifier is one with only two classes. In the case of such a classifier, the inputs patterns are discriminable, provided that input vectors for the two different classes do not occupy the same regions in the input space. In the one-dimensional case, the two-classes are discriminable if their respective probability distributions do not overlap. If they do, then it will not always be possible to classify the input correctly. Even when the pattern classes are theoretically discriminable, in practice it may be difficult to generate the required decision functions to discriminate them. If the required decision boundaries are very complex and their description requires many parameters, a large amount of training examples and a large number of processing may be required in order to determine the decision boundaries. As is the case with estimation problems in general, the more degrees of freedom characterising a decision boundary, the more effort is required to determine it.

7.1.5 Learning as functional approximation

One can also view pattern recognition as a problem in functional approximation (Poggio & Girosi, 1990). In this case we wish to approximate and estimate the function or functions that define the decision boundaries. The training data constitutes a set of data points that provides the basis for this estimation. In this case, one wishes to fit a function to divide the input data points into the appropriate decision regions.

Functional approximation using a look-up table

Let us suppose we have a set of input values (the input pattern vectors) and corresponding output values (the output pattern classes) from a function that we wish to approximate. Perhaps the simplest way to represent a function using a computer is to record each input pattern and its associated output value in a look-up table. Indeed, this approach is often adopted for many simple tasks, especially if is necessary to have fast access to the results: such a look-up table can be used to give an input to output mapping without any arithmetic computation.

Such a scheme works satisfactorily until a new input value is presented that does not exactly match any of the inputs present in the training data. This may be due to some kind of variability in the input measurement and corresponds to a location in the input space for which there was no training example. In this case, there are various courses of action that can be taken. For example, the look-up table could be searched to find the recorded input that best matches the new input, using a criterion of similarity such as Euclidean distance. This approach is precisely the one employed in classification using distance functions (Tou & Gonzalez, 1974).

Another approach is not to use a look-up table at all, but rather to approximate the input-output relation using a mathematical function. In this case, the parameters of some pre-specified function (which can be linear in some circumstances) are estimated to provide the "best fit" to the data-points. In this case, providing that the function is smooth and continuous, values between data samples are still defined. As long as there are sufficient data points to define the function (that is to say, provided that the data points are not too sparse) the interpolated values of the function between data-points will also be appropriately defined. Any input values can then be mapped to a corresponding output value, whether or not they coincide with the original data points.

It is the behaviour (due to interpolation) of a pattern classifier between the points defined by the training data that determines its generalization properties (Broomhead & Lowe, 1988). Such generalizations are only possible because the input vectors are not randomly related to their output class for most real-world problems, but are in fact structured. If they were random, and there would be no relationship between a pattern

and its corresponding class. In this case one would indeed need to record each input pattern in a look-up table in order to determine the output class, since useful interpolation would not be possible.

7.1.6 An example of a simple pattern recognition task

The following example is designed to clarify some of the issues that have just been discussed. A group of people are described in terms of two quantities for each individual; weight x_w and height x_h. In this case we can define a pattern vector $\mathbf{X}_p$ composed of the two measurements x_w, and x_h; that is

$$\mathbf{X}_p = [x_w, x_h]$$

Let us suppose that we had these measurements from four people; two men and two women. Their weight and height measurements are given in the table below:

Name	Weight	Height	Sex
John	150Kg	2.00m	male
Kevin	80Kg	1.80m	male
Jane	70Kg	1.75m	female
Maria	60Kg	1.70m	female

Then it is possible to write the pattern vector for each person as;

$$\mathbf{X}_{John} = [150.0, 2.00]$$
$$\mathbf{X}_{Kevin} = [80.0, 1.80]$$
$$\mathbf{X}_{Jane} = [70.0, 1.75]$$
$$\mathbf{X}_{Maria} = [60.0, 1.70]$$

Now consider the possibility of distinguishing the sex of a person from this group on the basis of these measurements. That is to say, given a pattern vector corresponding to one of these subject, is it possible to determine whether they are male or female. In

general, one may expect women to be shorter and lighter than men. (NB: The purpose of this example is not to suggest that weight and height characterize a person's sex, but rather to demonstrate a simple pattern recognition task).

If the pattern vectors are plotted on a graph where the axes correspond to weight and height respectively (see figure 7.1), it can be seen that the points corresponding to John and Kevin are distinct from those due to Jane and Maria. Therefore in this case, a line (one of many that will partition the space in this way) drawn across the graph as shown in the figure 7.1 can partition the pattern space into two regions; One corresponding to male and one corresponding to female.

If we are now presented with another pattern vector for an unknown speaker, this gives us a mechanism whereby we can numerically evaluate their sex. For example,

$\mathbf{X}_{unknown} = [40, 1.40]$

If this new point is plotted on the graph in figure 7.1, it can be see that lies in the region designated female. Therefore, on the basis of our part sets of measurements, we could deduce that it is plausible (although we cannot be absolutely sure) that the unknown speaker is female. This graph provides a means to distinguish between male and female subjects on the basis of their weight and height. If we so wished, it would be possible to characterise the decision line on the graph mathematically, using the equation for a straight line

$y = mx + C$

where in this case y would represent the parameter "weight", x would represent the parameter "height", and m and C are the gradient and intercept of the line respectively. The values of C and m define the decision function for the classification of the input patterns. That is

if $x_w - mx_h < C$ then **X** belongs to the class female; otherwise **X** belongs to the class

male.

Normally, one would not determine the decision boundary for a pattern classification task by hand, but employ an automatic technique, some of which are described in the next sections. Clearly in this simple example there were only a **very** limited number of training patterns (that is, only 4) and in practice one would require many more measurements to characterize a large population of subjects in order to be able to estimate a useful decision surface. Of course, it would not always be possible to determine the correct sex of a person on the basis of their weight and height, because there are tall heavy women and short light men. To avoid this source or error, more measurements could be added to the pattern vectors to represent other differences between men and women.

7.1.7 Basic structure of a pattern recognition system

A typical pattern recognition system involves three distinct operations: Signal measurement, signal pre-processing, and finally classification via the implementation of a decision function. A schematic diagram for a pattern classifier system is shown in figure 7.2.

Measurement

The first stage in a pattern processing system uses an appropriate transducer. It is the task of the transducer to measure the desired physical property and produce an output in the format required. The measurements that are made should be chosen to characterise adequately the phenomenon or object under investigation. If this is not the case, then a given pattern may appear similar to ones arising from different phenomena. It is not always obvious what the measurements should be or the format in which they should be presented to a pattern classifier. Unfortunately there is little theory available to assist with this task and the problem is generally left to the intuition of the designer (Tou & Gonzalez, 1974). A good solution in a particular cases often depends very much on the nature of the specific problem.

Pre-processing

After the initial raw measurements have been made, there is usually a pre-processing stage. Although the purposes of pre-processing are many-fold, it has two main objectives; data reduction and emphasis of features. It is advantageous only to present information to the input of a classifier if it is useful in the specific discrimination task. Unnecessary information serves no useful purpose, but may greatly increase the dimensionality of the input vectors. This increases the amount of computation that the classifier is required to perform, and in addition may make training the classifier more difficult.

It is desirable to do as much as possible of the overall transformation implemented by a complete pattern classifier in the initial stages, because this will result in a pattern recognition system that uses an adaptive pattern classifier (the part that is trained) stage of lower complexity than would otherwise have been the case. Consequently, such a system will be easier to train and once trained may be more efficient in its operation. In this case, the pre-processing may also be thought of in terms of implementing feature detectors that respond to important structural relationships in the signal. The pre-processing employed will affect the generalization capabilities of the following classifier, and is the subject of much research (Pao, 1989; Giles & Maxwell, 1987). One approach, for example, that uses a fixed input transformation to increase the dimensionality of the input vectors so that is only needs a simple linear classifier is the method of Radial basis functions (Broomhead & Lowe, 1988).

Determination of the decision function

Decision functions can be selected in a number of different ways. Perhaps the most obvious approach is classification based on distance functions. This technique employs a measure of similarity between the input pattern and a set of example patterns (the prototypes) to decide which class the input should be in. In the case of classification using likelihood functions, they can be implemented using a-priori knowledge concerning the nature of the pattern distributions. The latest artificial neural network

classifiers, which are (loosely) based on biological neural models, estimate the decision functions using iterative training procedures. Iteratively trainable classifier techniques have an advantage over statistical likelihood classifier schemes in that they do not make a-priori assumptions concerning the form of the probability distributions of the input patterns (Huang & Lippmann, 1987). The following sections explore some of these different approaches to pattern classification.

7.2 CLASSIFICATION USING DISTANCE FUNCTIONS

Classification using distance functions uses the similarity between patterns in terms of functions relating to their geometric proximity in the multi-dimensional vector space in which they are represented.

7.2.1 Template matching; nearest neighbour pattern classification

A simple approach using distance functions is that of template matching, which is also known as nearest neighbour classification (Nillson, 1965; Tou, 1969). During the training mode, an example of each class of pattern is recorded as a template. In recognition mode, an input pattern is compared against each template using a pre-defined distance similarity criterion. The class of the closest matching template (which is regarded as a prototype pattern for that class) is then assigned to the input data vector. One method that uses this approach employs classification based upon Euclidean distance functions. Expressed mathematically, consider the case where there are M pattern classes represented by the template patterns $\mathbf{Z}_1$ $\mathbf{Z}_2$ ...,$\mathbf{Z}_M$. The Euclidean distance between an input pattern $\mathbf{X}$ and the i_{th} template pattern is given by

$$D_i = [(\mathbf{X} - \mathbf{Z}_i)'.(\mathbf{X} - \mathbf{Z}_i)]^{1/2}$$

The class of the input pattern $\mathbf{X}$ is then assigned to class w_i such that $D_i < D_j$ for all i and j, except when i = j.

This technique is useful when the pattern classes have limited variability and form

clusters. Problems with this approach arise when it is difficult to select a good representative pattern for each class.

7.2.2 k-nearest neighbour pattern classification

Such an approach can easily be extended to the k-nearest-neighbour algorithms by using more than one example of each pattern class (Nillson, 1965; Tou, 1969). If, instead of using the one nearest pattern to determine the class of the input patterns, the class of the k-nearest patterns is calculated, and then the class with the majority is selected, this results in the k-nearest-neighbour classifier. A problem with this type of classifier is that it sometimes requires very large amount of memory to store the prototype patterns and requires a large amount of computation to perform the recognition.

7.2.3 Cluster seeking algorithms

The identification of characteristic prototype patterns and clusters of patterns form a central issue in the classification of patterns using distance functions. The reason for finding clusters is to represent the decision function using fewer parameters that would be required by using the individual vectors directly. Consequently various techniques have been devised to estimate a set of prototype patterns, and these are based on pattern clustering. Unfortunately these procedures have an intrinsic tendency to operate in an ad hoc fashion and the classification performance which they provide depends to some extent on the particular problem being addressed.

Although the measure of similarity most commonly used in pattern clustering techniques is Euclidean distance, it is also possible to use other metrics such as the Mahalanobis distance, which effectively warps the dimensions of the space depending on the statistical properties of the patterns (Tou & Gonzalez, 1974). The criterion for finding clusters may be heuristic or based on the optimization of some mathematical performance index.

The heuristic approaches are based on intuition and experience and consist of a set of

rules that exploit the chosen measure of similarity. It is usually necessary to set a threshold of acceptable similarity in these algorithms.

A typical performance index that is often used makes use of the overall sum of the squared errors between the samples of a cluster domain and their corresponding mean. This results in the k-means clustering algorithm (MacQueen, 1967). This algorithm involves the following operations:

Step 1. K initial cluster centres $\mathbf{Z}_1(1),\mathbf{Z}_2(1),...\mathbf{Z}_k(1)$, are selected arbitrarily.

Step 2. At the kth iteration, distribute the samples $\{\mathbf{X}\}$ amongst the K cluster domains using the relationship

$\mathbf{X}$ is a member of $S_j(k)$ if $|\mathbf{X} - \mathbf{Z}_j(k)| < |\mathbf{X} - \mathbf{Z}_i(k)|$

for all i = 1,2,...,K such that i is not equal to j, where $S_j(k)$ denotes the set of samples whose cluster centre is specified by $\mathbf{Z}_j(k)$ (any ties are resolved arbitrarily).

Step 3. Using results for step 2, new cluster centres $\mathbf{Z}_j(k+1)$ for j = 1,2,...,K are computed, such that the sum of the squared distances from all points in $S_j(k)$ to the new cluster centre is minimized. That is, a new cluster centre is computed so that the performance index

$$\mathbf{J}_j = \sum_{\mathbf{X} \text{ member } S_j(k)} |\mathbf{X} - \mathbf{Z}_j(k+1)|^2 \text{ , for } j = 1,2,..,K$$

is minimized. The value of $\mathbf{Z}_j(k+1)$ which minimizes $\mathbf{J}_j$ is the sample mean of $S_j(k)$, which means that the new cluster centre is given by

$$\mathbf{Z}_j(k+1) = 1/N_j \cdot \sum_{\mathbf{X} \text{ member } S_j(k)} \mathbf{X}, \text{ for } j = 1,2,..,K$$

where N_j is the number of samples in $S_j(k)$.

Step 4. If $\mathbf{Z}_j(k+1) = \mathbf{Z}_j(k)$, for j = 1,2,...,K, then the algorithm has converged and is therefore terminated; otherwise the operation loops back to step 2.

In practical applications, the value of K that gives good results will have to be found by trial and error. After the clusters have been found, they can be used as the basis for the prototype patterns in a distance classifier, such as the k-nearest-neighbour algorithm.

An example of an approach which is based on the use of in-built additional heuristic procedures is the Isodata algorithm (Ball & Hall, 1965).

7.2.4 Unsupervised pattern recognition

Cluster-seeking algorithms can also form the basis of unsupervised training schemes. This is because they self-organize so as to comply with some underlying principle, rather than by the minimization of an error resulting from a comparison with an explicitly defined target pattern class, which is the case in supervised training. In this case, different output classes can be thought of as being associated with different clusters. A fundamental problem with this approach is that, although classification of the input into some class may be possible, this may not be the class that is required for a particular application. Unsupervised pattern recognition is discussed again in the section of artificial neural networks (section 7.5.19).

7.3 CLASSIFICATION USING LIKELIHOOD FUNCTIONS

7.3.1 Introduction

Instead of adopting an approach to pattern recognition based on distance functions, one may consider the task as a statistical decision problem (Blackwell & Girshick, 1954). In this case the task is to find, in a statistical sense, the function that leads to the optimal decision. This is achieved by specifying a loss for each possible decision made

by the classifier. The task of the classifier is then, on the basis of the input data vector, to select the pattern class that results in the minimum overall estimated loss.

Three important strategies employing this approach are Bayes', Minimax and Neyman-Pearson (Tou & Gonzalez, 1974). The only difference between the three is in the threshold criterion used in the decision process. We will only consider the Bayes' classifier, because it is the most widely used.

7.3.2 Bayes' classifier

In a Bayesian classifier, the threshold criterion is defined such that the classifier has the task of minimizing the total expected loss (Reza, 1961; Van Trees, 1968; Helstrom, 1968; Tou & Gonzalez, 1974). Consider the case of a classifier with the following parameters:

W_i is pattern class i.
X is the data vector.

The probability of pattern class W_i given data vector **X** is given by the conditional probability $p(W_i|\mathbf{X})$.

The probability of the input data vector **X** given pattern class Wi is given by the conditional probability $p(\mathbf{X}|W_i)$.

The probability of the pattern class itself is given by p(W) and the probability of the data vector is given by p(**X**).

If the classifier decides that the input vector **X** came from patten class W_j whereas it actually came from pattern class W_i, then there is an associated loss given by L_{ij}. This loss matrix must be defined in advance. Often a loss of 1 is assigned to an incorrect classification and a loss of 0 is assigned to a correct classification (Tou & Gonzalez, 1974). If there are M classes from which the input vector **X** may have come, then the

total expected loss incurred in assigning the data vector $\mathbf{X}$ to class W_j is given by the sum

$$R_j(\mathbf{X}) = \sum_{i=1}^{i=M} L_{ij} p(W_i|\mathbf{X})$$

The term R_j is referred to as the conditional average risk or loss. We may use Bayes' theorem to change the form of the equation for the loss. Since

$$p(W_i|\mathbf{X}) = p(W_i)p(\mathbf{X}|W_i)/p(\mathbf{X})$$

This leads to

$$R_j(\mathbf{X}) = 1/p(\mathbf{X}) \sum_{i=1}^{i=M} L_{ij} p(\mathbf{X}|W_i)p(W_i)$$

In the selection of the most likely pattern class, we must choose the pattern class with the lowest associated loss. Since the $1/p(\mathbf{X})$ is a common factor in the losses, the loss equation can be redefined as

$$R_j(\mathbf{X}) = \sum_{i=1}^{i=M} L_{ij} p(\mathbf{X}|W_i)/p(W_i)$$

In the case of a two-class classifier ($M = 2$), the two loss equations are

$$R_1(\mathbf{X}) = L_{11}p(\mathbf{X}|W_1)p(W_1) + L_{21}p(\mathbf{X}|W_2)p(W_2)$$

$$R_2(\mathbf{X}) = L_{12}p(\mathbf{X}|W_1)p(W_1) + L_{22}p(\mathbf{X}|W_2)p(W_2)$$

Thus class 1 is assigned if $R_1(\mathbf{X}) < R_2(\mathbf{X})$, and class 2 otherwise. In the multi-class

cases, the procedure is illustrated in figure 7.3.

7.3.3 Bayes' classifier for Gaussian patterns

An assumption that is often made is that the probability distributions of the pattern classes follow a Gaussian distribution (Anderson & Bahadur, 1962; Cooper, 1967; Duda & Hart, 1973; Tou & Gonzales, 1974). In this case the pattern classes need only be represented in terms of their mean vectors and covariance matrices.
In this case the multivariate distribution is given by

$$p(\mathbf{X}|W_i) = 1/((2\pi)^{n/2} |\mathbf{C}_i|) \exp[-1/2(\mathbf{X} - \mathbf{M}_i)'\mathbf{C}_i(\mathbf{X}- \mathbf{M}_i)]$$

where $\mathbf{M}_i$ is the means vector and $\mathbf{C}_i$ is the covariance matrix. It is convenient to take logs of the terms in the loss equations; this does not affect the comparisons but makes the calculation simpler. In this case

$$d_i(\mathbf{X}) = \ln(p(W_i)) - 1/2\ln|\mathbf{C}_i| - 1/2[(\mathbf{X} - \mathbf{M}_i)'\mathbf{C}_i(\mathbf{X} - \mathbf{M}_i)]$$

where d_i represents a decision boundary. Again class 1 is assigned if $d_1(\mathbf{X}) < d_2(\mathbf{X})$, and class 2 otherwise. The values of the mean vectors and covariance matrices are estimated from a set of training data.

Often the assumption that the pattern classes are normally distributed is not valid. Under these circumstances, one may resort to a functional approximation to the probability distributions. However, it becomes more difficult to model the distributions as they become more complex. It then requires a large amount of training data in order to get a good estimate of the probability distributions. The Bayes' classifier was compared to the MLP in a voicing determination experiment that is reported in chapter 8.

7.4 BRIEF REVIEW OF ARTIFICIAL NEURAL NETWORKS

7.4.1 Introduction

Artificial neural networks (also known as connectionist models) are systems that were designed to mimic some of the organizational characteristics of biological networks of neurons in the nervous systems of animals. Many of these models arose out of a desire to explain observations made in the fields of psychology and neuro-biology (Lippmann, 1987; Anderson & Rosenfeld, 1988; Widrow & Lehr, 1990).

The type of problems that are best solved by the brain tend to be those that involve the satisfaction of a large number of weak constraints. These problems are difficult to solve using conventional algorithms running on digital computers despite being easily solved by the brain. An example of this is the recognition of noisy images. This task is relatively easy for a human to perform, but is difficult to achieve by computer. On the other hand, the types of processing that humans are poor at, such as unaided arithmetic operations, can be achieved relatively easily by computer.

With the current revival of interest in neural networks, there is also interest in building efficient hardware implementations. As well as using traditional VLSI technology (Mead, 1989), other technologies are being explored, including the use of optical implementations (Farhat et al., 1985).

7.4.2 Characteristics of biological neurons

One feature of biological neural networks is their massive parallel use of very many similar basic computing units (the neurons). For example, in the human brain it is estimated that there are around 10^{10} neurons. These computing units are slow compared to their electronic counterparts (for example, transistors). However, at a system level within a brain this speed limitation is overcome by the co-ordination of very many such units operating in parallel. It is only recently that computers are being built that also exploit parallelism in their operation.

A biological neuron functions as a complicated electro-chemical device. A neuron

receives input from other neurons at special structures called synapses, and transmits to other neurons via output lines known as axons. A neuron in the cortex can have receive up to 10^5 inputs and send output to a similar number of other neurons (Kandel & Schwartz, 1985). There are two basic kinds of input synapse; excitatory ones and inhibitory ones. If a neuron receives enough input via its different excitatory synapses, this causes the electric potential within the body of the neuron (the membrane potential) to rise above a threshold value, at which point there is a high probability that a pulse-like signal (known as the action potential) is generated and transmitted out from the body of the neuron along the axon to other neurons. Conversely, any inputs received on the inhibitory synapses reduce the likelihood of the generation of such an action potential.

Biological neurons do not simply operate in a simple binary fashion, even though the action potential is a two-state pulse. The output response from a neuron is coded in terms of frequency of output pulses it generates, and this frequency relates to the membrane potential of the neuron averaged over time (which in turn is a function of the inputs to the neuron). Many artificial neurons mimic this graded response by using a continuous output function, although some simpler models assume binary computing units.

7.4.3 Basic characteristics of artificial neural network models

Most neural network models distinguish two phases of operation of the systems: learning and retrieval. These two phases correspond to the training and recognition modes attributed to classical pattern recognizers. In the learning phase the connection strengths in the network are modified such that the network changes its function so that it constitutes a better model of the required transformation. In the retrieval phase, a stimulus is presented to the network and this changes the internal activity of the network and gives rise to an output pattern. During this process, the connection strengths are not altered. Because of the complexity of the network connections, it is often difficult to analyze the function of the network.

7.4.4 Comparison between traditional classifiers and artificial neural networks

Traditional classification techniques usually operate serially, although many can be implemented using parallel computation. There are many similarities between classical and neural network classifiers. For example, the computational structure used by a linear perceptron classifier is the same as a Bayes' classifier with Gaussian patterns, for which the covariance matrices for each class are identical (Lippmann, 1987). However, the training in the two approaches differs, which results in different operational performances. As opposed to the Bayes' classifier, the perceptron makes no assumption about the a priori probabilities of the input patters and simply operates to correct errors. Consequently its operation tends to be more robust (Huang & Lippmann, 1987). Of course, neither are appropriate when the patterns are not linearly separable (that is, cannot be discriminated using a hyper-plane decision function that uses a linear combination of the input vector elements).

7.4.5 Origins of artificial neural networks

The ideas behind the field of neural networks are not new. Anderson et al. (1988) have pointed out that the elementary principle of association was appreciated over a hundred years ago by the psychologist William James (James, 1890). He stated this principle as "When two brain processes are active together or in immediate succession, one of them, on reoccurring tends to propagate its excitement into the other". This is almost a direct statement of the Hebb learning rule, which is discussed in a later section. In addition, James gave a summing rule for brain activity which can be related to a model of a neuron in which its activity is due to its inputs weighted by their connection strength, the values of which were previously determined by past correlations. This scheme is very similar to current neural network models which employ Hebbian learning and compute the linear sum of their synaptic inputs.

7.4.6 Early models of the nervous system

An attempt to understand the process of neural computation was undertaken in a paper

by McCulloch and Pitts (1943). They made various assumptions about the operation of neurons. These McCulloch-Pitts neurons, as they became known, were binary devices (that is to say, their output could only be in one of two possible states). A neuron could receive input from either excitatory or inhibitory synapses. All the excitatory synaptic connections had an equal strength, and if the integrated activity over a time quantum was greater than a preset threshold, the neuron would fire, provided no inhibitory input was present. The time quantum for integration of the synaptic inputs was loosely based on the synaptic delay that is observed in biological neurons. The authors proved that, using a network of such neural elements, it was possible to implement any finite logical expression. These results had much influence in the fields of neuroscience as well as in computer science (von Neumann, 1958).

7.4.7 The Hebb learning rule

The first explicit statement of a physiological learning rule for neural networks was given by Hebb (1949); "When an axon of a cell A is near enough to excite a cell B and repeatedly or persistently takes part in firing it, some growth process or metabolic change takes place in one or both cells such that A's efficiency, as one of the cells firing B, is increased". Hebb was also one of the first to refer to the term connectionism in the context of complex neural models.

Because Hebb did not make a mathematical statement concerning his postulate, its definition is somewhat wider than may otherwise have been the case. Hebb also was well aware of the "distributed nature" of computation in the nervous system and that representations in a complex nervous system require the participation of many nerve cells.

7.4.8 Early computer simulations of neural networks

One of the earliest attempts to simulate the nervous system using a computer was carried out by Farley and Clark (1954). Slightly later Rochester et al. (1956) carried out work on the learning system proposed by Hebb (1949). One important result of this work was

that it showed that a working simulation required a level of model detail that can easily be overlooked in qualitative discussions. Computer simulations proved themselves to be valuable techniques to test theories in a way that was not possible merely by analysis. They found by experimentation that the Hebb rule needed a normalization component to prevent synaptic weights from growing without bounds. They also found a fatigue factor necessary, whereby the firing of a cell reduces the probability of an immediate subsequent future firing.

7.4.9 The perceptron

The first precisely specified neural network was the perceptron (Rosenblatt, 1958). A number of different variants of perceptron were initially described. A perceptron is illustrated in figure 7.4. A "simple" perceptron was a three layer device, that employed a "winner-take-all" operation at its output. The basic architecture of such a "simple" perceptron consisted of a sensory surface (which is the input to the system) known as the "retina", which then connected to a second layer, known as the "association layer". The connections between these layers were local and random, and units in the association layer were known as A-units. A given A-unit only received input from a local area of the retina. The A-units in the association layer were connected in turn to R-units in a response layer. To prevent more than one response unit from becoming active at a given time, there were a set of inhibitory connections from each R-unit that inhibited all A-units in the association layer to which a given R-unit was **not** connected.

Rosenblatt considered that computation in the brain should be regarded in terms of the association, discrimination and classification of stimuli, rather than the computation of logical functions. He felt that the latter was inappropriate in view of the randomness and noise present in the nervous system. The learning rules for the perceptron were mainly based on simple reinforcement. The simplest rule adopted a self-organizing principle, but "forced" learning (that is, supervised learning) was also mentioned.

7.4.10 The Pandemonium model

Another early parallel neural model was the Pandemonium model due to Selfridge (1958). This consisted of multiple independent sub-systems that simultaneously processed the input and responded appropriately when a feature specific to each sub-system was detected. Selfridge proposed the technique of "hill climbing" to adjust the connection strengths between units and the input, or units and the output. The idea behind this is as follows: The response of a given unit depends upon its connection strengths. If it was known to what input a given unit should to respond to, and to which ones it should not respond, the connections could be altered until the values that give the "optimum" performance were found (that is, the best with respect to some pre-defined criterion). One way to do this is to alter the connection strengths by a small amount in all directions and then choose the values that give the best improvement in performance. We could then repeat this process again and again, until a good solution had been found. This procedure is analogous to climbing a hill, where height of the ground relates to the performance of our system. Unfortunately if we adopt the policy of following the maximum local gradient, then whenever we reach the top of a hill where the gradient is zero, we stop moving. If we are interested in finding the highest point in the landscape, it can be seen that this approach can fail whenever there are small foot-hills around the main hill, because we can become trapped on a local hill. Similarly if we adopt this local search procedure, the optimization of the connection strength for the units can become trapped in local performance maxima. The essential point is that a local search does not generally find the global maxima (or minima) of a function, unless the function contains no local maxima.

7.4.11 Widrow and Hoff learning rule

The perceptron learning rule changed the synaptic strengths (weights) on the basis of whether or not a correct classification was made. One problem with such rules is that a large number of iterations are required before training is complete. A system that is related to the perceptron was proposed by Widrow & Hoff (1960). This was called an adaptive neuron (ADALINE) and it computed the sum of the inputs with their corresponding connections strengths, plus an additional bias term. If the sum was greater than zero the output was set high, whereas otherwise it was set low.

Training is performed using a supervised training scheme which provides a target output of +1 or -1 as appropriate. The learning rule is different from the perceptron learning rule in that this time during training, an input pattern is presented and an **error signal** is computed, which corresponds to the difference between the output from the summing stage and the desired target output. The connection strengths of the neuron are again adjusted so that the error is minimized. Now because the error is minimized, the learning process still occurs even when the neuron correctly classifies the input; this is not the case with the perceptron. It is intuitively evident that by providing more information concerning the way in which the system fails, the faster should its performance improve. This is one reason that this type of neuron learns faster than the perceptron. In addition, in the case of input classes that are not linearly separable, the Widrow-Hoff rule converges to produce the best mean square fit, whereas in the case of the perceptron convergence procedure, the decision boundary may not converge, but rather oscillate continuously. Mainly because of its rigorous mathematical definition, the ADALINE became an established technique in adaptive signal processing.

Learning in this type of neuron model actually operates by minimizing the square of the error. One can consider this error as a function of all the possible connection strengths for a given neuron. This leads to an error surface in weight space (as before with the pandemonium model). Again, we wish to find the "optimum" weights that give the overall global minimum error. If we use the hill-climbing approach and follow the path on minimum gradient, then we can only guarantee to find the global minimum only if there are no local minima. However, Widrow and Hoff showed that if we minimize the square error, the error surface (for a linear network) is a simple quadratic surface with only one global minimum. Consequently to find this point only requires that we follow the minimum gradient path which can be computed from the partial derivative of the error with respect to the weights. They also showed that this derivative is proportional to the error signal. This error correction routine is also known as the LMS (Least Mean Squares) algorithm.

A development of the ADALINE model was the MADALINE model which consisted of multiple ADALINE elements (Widrow, 1962). This consisted of a set of ADALINE

units connected to a fixed logic gate and this structure was only adaptive over the first layer. Another early neural model was Steinbuch's learning matrix, which was based on linear processing elements (Steinbuch & Piske, 1963).

7.4.12 Limitations of linear networks

An analysis of the capabilities of the perceptron was performed by Block (1962). Originally, Rosenblatt (1958) had found that the classification of the perceptron for random vectors was rather poor, but better with patterns that were correlated.

The basic perceptron can only divide the input space by means of a hyper-plane (see later section), by virtue of its linear threshold operation (a unit that performs a scalar product of its input vector with a weight vector. It then generates a binary output depending upon whether or not this is above or below a threshold value is known as a linear threshold unit). Therefore, it is unable to learn to classify input vectors that are not linearly separable in this way. Minsky & Papert (1969) pointed out that such a linear classifier is unable to classify (or rather distinguish) many patterns that human can classify. Another issue relates to whether or not a classification can be learned, even if it is theoretically possible for it to be performed. The perceptron convergence theory (Block, 1962) proved that any classification that is possible can be learned. The initial interest in artificial neural networks considerably waned, and this is partially attributed to the discussion of the limitations of perceptrons by Minsky & Papert (1969), who placed the discussion of their capabilities on a firm theoretical foundation. Rosenblatt considered a R-unit in perceptron as calculating a classification, whereas Minsky and Papert felt that it was computing a logical predicate. In their view, the A-units that received input from the R-units in the retina constituted local predicates that can be called Φ. These local predicates were considered to be points on a plane, to which they applied geometrical arguments. The question then naturally arose as to what logical functions could be realised by the perceptron. They discussed two important kinds of limitations on the local predicates. The first of these were order-limited, in which only a given maximum number of retinal points could be connected to the unit that computed the local predicate. The others were diameter-limited, in which a geometrically limited

region of the retina was connected to the local predicate. They showed that connectedness (whether or not lines in a figure are joined together) could not be computed using limited perceptrons, and neither could parity (whether the number of points present is odd or even).

7.4.13 Hopfield networks

The usual approach adopted in specifying a neural network is to propose a learning rule (often based on the Hebb rule) and then investigate its behaviour. Hopfield (1982) took a different stance in the derivation of a learning rule for a neural network. He started from the view that the function of the nervous system is to develop locally stable points in state-space he called attractors. Consequently, other points in state-space tend to be attracted towards there points. This model consisted of linear threshold units as the processing elements, with neurons connected to each other, but not to themselves. Hopfield also introduced the important concept of using a description of the network related to physical energy; that is

$$E = -1/2 \sum_{i \neq j} \sum T_{ij} V_i V_j$$

where V_i are the individual element activities, and $T_{ij} = T_{ji}$ is the symmetrical connection matrix. This system evolves as a function of time, due to the dynamics of the feedback, until an energy minimum is reached. These dynamics follow a simple rule whereby an element is chosen at random, and by consideration to the effect of its input, its state changes depending on whether or not its summed activity is above or below a threshold value. Hopfield showed that such a network can reliably store no more that 0.15N orthogonal patterns (ones that do not interact with others also stored), where N is the number of nodes.

Although this neural model used McCulloch-Pitts type neurons that were easy to analyze, it is not very realistic from a biological viewpoint. Hopfield (1984) extended this earlier basic neuron model by replacing the output threshold function with a sigmoid

non-linearity. This matches the graded output characteristics of real neurons (although they use pulse frequency modulation). This is a monotonic function which is something of a compromise between a threshold and linear function (see figure 7.5). In addition, a neuron contained a state variable that represented the weighted sum of the synaptic inputs. Hopfield again showed that the system finds a stable state by performing a minimization of energy.

7.4.14 Problems of training multi-layered networks

To overcome the computational limitations of single-layered linear networks, it is necessary to use multi-layer networks with non-linear processing nodes. Such nodes that are not directly connected to the input or output are known as hidden units. When we deal with single layered networks, we have access to both the input and the output of all the processing nodes in the system. In these cases, it is possible to use a learning rule such as those due to Hebb or Widrow and Hoff. However in the case of multi-layered networks, which contain hidden units, this is no longer directly possible. The problem of specifying how to change the connection strengths is often referred to as the "credit assignment problem".

7.4.15 The Neocognitron

There are several solutions to training such systems. In an approach proposed by Fukushima et al. (1983), which used what they described as neocognitrons (based on the earlier cognitron, Fukushima, 1975). Here it was assumed that the designer has an idea as to what the basic internal network structure should be. They adopted an approach to optical character recognition which exploited knowledge about the visual system in the form of simple and complex feature detectors, as described by Hubel and Wiesel (1962,1965). This enabled them to build a set of modules to perform initial processing of their images. They then trained higher levels of the system, using the input from the lower layer. Consequently they were able to build a system in which the input was first analyzed using a set of feature detectors, the outputs from which were fed into a final character recognition stage.

A different variant of the credit assignment problem was solved by Barto et al. (1983). They made use of what they called an adaptive critic element to monitor the performance of an adaptive search element. It was the task of the latter to generate the appropriate response to the input stimuli, whereas the former element has the function of predicting the expected reinforcement that should be applied to the adaptive search element.

7.4.16 Simulated annealing

A problem that faces all credit assignment tasks that operate by "hill climbing", is that they can fail when there are local error sub-minima, because if the algorithm encounters one, it cannot escape from it (Selfridge, 1958). To avoid this situation requires more than just local information concerning the nature of the error landscape. One technique that can overcome the problems associated with local minima is known as stimulated annealing (Kirkpatrick et al., 1983). The basic idea is that instead of following the path of maximum gradient **all** the time, it should only be followed **most** of the time. The technique involves defining a system parameter **temperature** which determines how random the search is, and this relates to the physical parameter of temperature. The probability $P(C_n)$ of finding the system in a given configuration C_n is given by the Boltzmann factor

$$C_n = \exp(-E(C_n)/KT)$$

In the case of the two configurations C_1 and C_2 with associated energies E_1 and E_2, the ratio of the probabilities of the two configurations is given by the relationship

$$P(C_1)/P(C_2) = \exp([-E(C_1) - E(C_2)]/KT)$$

At high temperatures (large T) the denominator term KT becomes very large compared to either $E(C_1)$ or E(C2), and consequently the ratio between the two probabilities approaches 1.0. That is, the system is just as likely to change between the two states as to stay in the same state, irrespective of the energies of the different states. However,

as the temperature is lowered, the probability of occupation of the high-energy state becomes small compared to the low-energy state. The strategy that Kirkpatrick et al. adopted was to start off with a high temperature, and gradually reduce it to zero. If this process is successful, the system should then end up in its lowest overall-energy state, having avoided local minima by virtue of the fact that the initial search was carried out over the entire error landscape. When this process is used to soften metals, it is known as annealing; when it is used to find a minimum in an optimization problem, it is term simulated annealing.

It was later shown (Geman & Geman, 1984) that simulated annealing converges to its overall-energy minimum if the temperature at the k^{th} step is kept above the value T(k), given by the relationship

$$T(k) >= c/[\log(1 + k)]$$

where c is a constant independent of k.

7.3.17 The Boltzmann machine

As previously pointed out, there are two distinct phases of operation of pattern classifiers (including neural networks). There is the learning phase, during which the connection strengths are estimated. Secondly, there is the operational phase, in which the connection strengths remain unaltered, but the system responds to the input stimulus. In the case of the Boltzmann machine (Ackley et al., 1985), the training phase and the operational phase both make use of simulated annealing to enable them to "relax" into their solutions. The term relax is used to convey the fact that the system reaches a minimum energy state due to the inherent behaviour of its dynamics. The Boltzmann machine uses units that are the same as those used by Hopfield; They can only be on or off (binary) and their state depends upon whether or not the weighted sum of input exceeds an internal threshold level. The energy of the system is also defined in a similar way to that in a Hopfield network, as a quadratic function. The individual units are stochastic, that is the state of a unit is a probability. Thus, if two states for a given

units are separated by energy given by ΔE, the unit is switched on with probability p given by

$$p = 1/[1 + \exp(-\Delta E/T)]$$

where T is the temperature parameter. The name Boltzmann machine arose for this type of network because the relative probabilities of two states (above) is given by the Boltzmann distribution.

Learning in a Boltzmann machine corresponds to the determination of the connection strengths between the units such that the system simulates the probabilities associated with its external environment (the training data). Ackley et al. (1985) showed that there is a simple procedure that can modify the weights so that this can be achieved. This involves letting the system run "free", with no constraints from the external environment and estimating the probability of all the states in the network. The visible surface units (that is, those that are directly connected to the environment) are then "clamped" (forced) to take desired values. Once again, the probabilities of the states of the units in the network are estimated. Weight changes are then made which are proportional to the difference between the un-clamped and clamped state probabilities. As might be expected of a stochastic procedure, simulations of Boltzmann machines on serial computers are very slow.

7.4.18 The multi-layer perceptron

One of the biggest recent break-throughs in the field of neural networks was the discovery of an algorithm to permit the training of multi-layer networks. This is the back-propagation algorithm, and it was discovered independently in four different places at around the same time (Le Cun, 1986; Parker 1985; Rumelhart et al., 1986; Werbos, 1984).

Back-propagation is a generalization of the Widrow-Hoff error correction rule. However, the Widrow-Hoff rule only applied to single layer networks in which there

was direct access to the units. In this case the error at each unit could be directly formulated from the desired overall output from the network. In the case of networks that contain hidden units, the error cannot be explicitly computed in this way.

The generalized delta-rule

The generalized delta rule provides the means to calculate the required weight changes in the case when hidden units are present. However, it first requires that the effective error at a hidden unit is estimated. This is done by passing the error from the output layer **backwards** through the weighted connections to the hidden units. A given hidden unit sums up all of its weighted back-error contributions. Given additional information concerning the strength of the input it receives, its weight changes can then be computed as they would be in the case of a unit in a single layer network. Therefore, back-propagation first involves a forward-pass of the activity of the network arising from the input through the network. The output error is then calculated, and is then back-propagated through the network so that the weights can be modified.

It is clear that the back-propagation of error is not a biologically plausible mechanism (because the error is propagated backwards through the same weights as the forward activation). However, it is currently the most effective technique for training multi-layer networks and has proved itself to be capable of training pattern classifiers that perform as well or better than many classical techniques.

Advantages of the multi-layer perceptron

The MLP has also shown itself to be a robust pattern recognition technique in many applications of speech pattern processing (Peeling & Moore, 1986; Boulard & Wellekens, 1986; Howard & Huckvale 1988a,c). For example, Huang & Lippmann (1987) found the MLP performed as well as or better than a Gaussian and K-nearest-neighbour classifiers using vowel formant data. Atlas et al., (1990) found that the MLP performed as well or better than classification trees.

Developments of the basic architectures of systems using the MLP have been widely investigated, often with beneficial effects (Lang et al., 1990). The MLP provides a convenient formalism for constructing systems that are trained in parts (Waibel et al., 1988) and then combined together to provide a network that is organized hierarchically (Howard & Huckvale, 1989). Also, the uniform structure of the MLP makes real-time implementations with special DSP systems relatively easy (Howard & Walliker, 1989).

Because the multi-layer perceptron is the technique used in this thesis for pattern recognition, it is discussed in more detail in the next section.

7.4.19 Networks that employ unsupervised training

So far we have considered neural network models which require an explicit teaching signal. These networks are of practical value in the solution of engineering problems, and indeed a supervised neural network classifier is used in this thesis (the MLP). From a biological point of view, unsupervised techniques are perhaps more plausible, as is argued by the authors of these algorithms.

Early mode-seeking (clustering) schemes were due to Stark, Okajaima & Whipple (1962). The spontaneous learning rule for the perceptron is also an example of unsupervised learning (Rosenblatt, 1962).

Another type of unsupervised neural model is due to Grossberg (1976,1980). The inspiration behind this model comes from developmental physiology in the organization of the cortex (the highest level in the brain). There is strong evidence to suggest that feature detectors in the visual cortex develop and modify their response depending upon the particular environment in which an animal is raised. In this way the cortex may adapt itself to make use of the most useful features in an environment, and ignore others. Grossberg's work on neural models start from the development of non-linear lateral inhibition as a means to normalize the dynamic range of input patterns. He also considers the problems of how to implement short term memory as a means to maintain patterns after their stimuli have been removed. He proposes that this could be achieved

by means of feedback that reinforces a given pattern of activity. Many neural models require the presence of a teacher to specify the target pattern class (for example in the perceptron, or in ADALINE). Grossberg considers it a key point that in biological neural networks, this cannot be done explicitly. Consequently a neural network must correct errors itself without outside help, and this has implications with regard to their structural organization. He make the suggestion that there could be reciprocal connections between two groups of neurons such that a response travelling "upward" can provoke learning by means of feedback traveling "downward".

Another class of unsupervised neural models are the self-organizing arrays due to Kohonen (1982). This type of network develops topographical representations of the input space. An important characteristic of neurons in sensory pathways of the brain is that their placement reflects some physical characteristic of the external stimulus that is detected. For example, in the peripheral parts of the auditory system, neurons are arranged according to the frequency of the stimulus to which they respond best (Kandel & Schwartz, 1985). A Kohonen net consists of a two-dimensional array of units. The scheme operates according to the principle that nearby units respond in a similar fashion. The learning rule adopted by Kohonen achieves this kind of organization in a straightforward way. Initially all the units respond randomly to the input stimulus. However, one unit will in general respond **most strongly** to the input, and this unit is located. Neighbouring units then have their weights changed so that they also respond more strongly to the input than they did before. Provided there is some kind of overall normalization of the weights so that the overall sum of the weights remains about constant, this rule usually leads to topographical ordering of the units. Kohonen's networks can perform what is known classically as vector quantization (Kohonen et al., 1984).

Work on unsupervised neural networks was also carried out by Rumelhart & Zipser (1985). They showed that a set of simple competitive mechanisms could give rise to feature detectors that capture important characteristics of the input stimuli. They also showed that these feature detectors could then be used as part of a multi-layer classifier system. By using the feature detectors as the input to a linear classifier, the overall

system could classify patterns that were not linearly discriminable, thus demonstrating the usefulness of competitive learning schemes in the training of multi-layer networks.

7.5 IMPORTANT ASPECTS OF THE MULTI-LAYER PERCEPTRON

7.5.1 Introduction

There now follows a more detailed discussion of neural network pattern classifiers and the multi-layer perceptron, because this is the pattern recognition technique used in this thesis. Then there is a mathematical analysis of the learning procedure. Finally, some practical issues concerning the multi-layer perceptron are examined. In particular, a technique developed during the work in this thesis, known as selective emphasis, is described. This provides a means to increase training speed by about ten times.

7.5.2 Computation using a linear network

The simplest structure for a network classifier is a one-layer system, in which each output node is connected directly to each input node via a weight. The value of each output is computed by summing the contributions due to each input value multiplied by its appropriate connection strength. Such a network computes the scalar product of the input vector $\mathbf{X}_1$ with its connection weight matrix $\mathbf{W}_1$. That is

$$\mathbf{Y}_1 = \mathbf{W}_1.\mathbf{X}_1$$

Where $\mathbf{Y}_1$ is the output vector for the network. When such a linear network is used to implement two-class pattern classification, the relationship used to calculate the output class is given by

if $\mathbf{W}_1.\mathbf{X}_1 \geq T$, $w_i = w_1$

else if $\mathbf{W}_1.\mathbf{X}_1 < T$, $w_i = w_2$

where T is a threshold term. It can be seen that this is the equation of a hyper-plane.

7.5.3 Effect of cascading linear networks

Such a network is only capable of linear transformations of the input vector. To achieve more complex operations, multi-layer networks must be used. However, it is easy to show that there is no computational advantage in cascading linear networks. Consider the effect of feeding the output of one linear network into the input of a second linear network defined by

$$\mathbf{Y}_1 = \mathbf{W}_1.\mathbf{X}_1$$

$$\mathbf{Y}_2 = \mathbf{W}_2.\mathbf{X}_2$$

In this case, setting $\mathbf{X}_2 = \mathbf{Y}_1$ gives

$$\mathbf{Y}_2 = \mathbf{W}_2.(\mathbf{W}_1.\mathbf{X}_1)$$

Therefore

$$\mathbf{Y}_2 = \mathbf{W}_3.\mathbf{X}_1$$

where $\mathbf{W}_3 = \mathbf{W}_2.\mathbf{W}_1$

Thus the computational effect of the two cascaded networks can be realised using a single network in which the connection matrix is represented by the product of the previous connection matrices. It follows that the computational function of a cascade of any number of linear networks can always be implemented using a single layer linear network with the appropriate weight matrix.

7.5.4 Limitations of linear networks

A classical example used to demonstrate the limitation of linear networks is the exclusive-or (XOR) problem (see figure 7.6). As shown earlier, a linear classifier can only partition the input space by means of a hyper-plane. Consequently, it cannot solve the XOR problem.

7.5.5 The effect of "hidden units" on the classification capabilities of a network

A signal transformation of arbitrary complexity may be achieved by means of a network of connected elemental processing units, provided they incorporate some kind of non-linearity (Rumelhart & McClelland, 1986). Figure 7.7 illustrates networks with no hidden layers, one hidden layer and two hidden layers.

A cascaded network which employs output non-linearities at each units cannot be considered equivalent to a single layer network with adjusted weights. Typical non-linearities that are employed include simple threshold functions and sigmoid functions. Under these circumstances, cascading layers of units increases the generality of the computation that can be carried out.

In the case of threshold units, a single layer network achieves a linear decision region partition. It has recently been shown that one hidden layer is theoretically sufficient to solve any problem, but there are no constraints on the required number of units (Cybenko, 1989; Funahashi, 1989). However such a solution may not necessarily be the most efficient and it may sometimes be better to use to use two hidden layers (Lippmann, 1987; Widrow & Lehr, 1990). The decision region required by any classifier can be implemented using a 3-layer feed-forward network (Cybenko, 1989; Moore & Poggio, 1988). Figure 7.8 illustrates some of the possible decision regions that can (and cannot) be implemented using an MLP with different numbers of layers.

The consideration of the decision boundary complexities in relation to the hidden units provides some insight into the necessary number of hidden units for an application. In the case where decision regions are disconnected or meshed, there should be more than one unit in the second layer (Lippmann, 1987).

Similar behaviour is obtained using sigmoid non-linearities instead of linear threshold units, although this complicates matters somewhat. However, the use of sigmoid non-linearities enables the use of the back-propagation algorithm for the training of the multi-layer network, which cannot be used for threshold units.

It is to be noted there are other techniques employed to make classifiers capable of implementing non-linear decision boundaries. Rather than use other layers of non-linear elements, it is possible to introduce non-linear terms in the input to a linear classifier. One approach involves using polynomial terms of the original input vector to generate a modified input vector (Specht, 1966; Barron, 1984).

7.5.6 Mathematical analysis of learning

There are essentially two types of learning rule to estimate the weights in supervised network classifiers. Firstly there are rules that employ error-correction, which alter the weights in order to correct the output response. Error correction rules tend to operate in an ad hoc fashion. An example of this class of rules is the perceptron convergence theorem. Secondly, there are gradient descent rules which alter the weights with the intention of minimizing the average mean-square error of the network, with respect to all the training data. We shall only consider the latter in detail, as this is the class of rule employed to train the multi-layer perceptron.

7.5.7 The delta rule

The delta rule is the name given by Rumelhart et al. (1986) to the least-mean square training rule. It is an extension of the training scheme for linear networks due to Widrow & Hoff (1962) which is based on gradient descent. We shall consider the latter first and then show the extensions necessary to derive the generalized delta rule, which can be used to train multi-layer networks. This derivation follows that in Rumelhart & McClelland (1986).

In gradient descent learning rules, the weights in the network are altered according to

the relationship

$$\mathbf{W}_{k+1} = \mathbf{W}_k + \mu.\text{gradient}_k$$

Where $\mathbf{W}_k$ represents the weights and gradient_k is the gradient in the error surface with respect to the weights, both at iteration k, and μ is a constant. We must now calculate the term $\mu.\text{gradient}_k$ so that we can use this relationship to train the network.

Let the output from the network be O_{pj} and the target patterns be T_{pj}, where p is the particular pattern being processed at that time and j is the particular output node. We can define an error for one pattern presentation as

$$E_p = 1/2 \sum_j (T_{pj} - O_{pj})^2$$

The total error over the training data is the error sum for all patterns in the training set given by

$$E = \sum_p E_p$$

Since we wish to perform a gradient descent, we must calculate the change in error with respect to the weight changes. Using the chain rule, we may write

$$\delta E_p/\delta W_{ji} = \delta E_p/\delta O_{pj} \; . \; \delta O_{pj}/\delta W_{ji}$$

The first term is given by

$$\delta E_p/\delta O_{pj} = -(T_{pj} - O_{pj}) = \sigma_{pj}$$

In the case of a linear network the outputs are directly available, so

$$O_{pj} = \sum_i W_{ji}.I_{pi}$$

Therefore

$$\delta O_{pj}/\delta W_{ji} = I_{pi}$$

where I_{pi} is the input to unit i for pattern p. Therefore these equations yield

$$\delta E_p/\delta W_{ji} = \sigma_{pj}.I_{pi}$$

and summing the error over all training patterns gives

$$\delta E/dW_{ji} = \sum_p \delta E_p/\delta W_{ji}$$

To achieve gradient descent, we must alter the weights in proportion to the quantity $\sigma_{pj}.I_{pi}$, which is calculated after each pattern presentation. It is worth mentioning that a true gradient descent is only followed if all the weight changes are made together after the presentation of all the patterns in the training set. However, if the learning constant is sufficiently small (a constant used to scale the weight changes), the weight changes can be made after each presentation without the procedure departing significantly from gradient descent.

In the case of a linear network, the error surface is a hyper-quadratic function, with only one minimum (Widrow & Hoff, 1960). Therefore, in this case gradient descent will find the optimum solution. The situation becomes more complicated when there are hidden units in the network, because there are then local minima of the error surface. In addition, we cannot directly compute the error derivatives for the hidden units (the credit assignment problem mentioned earlier). There now follows a derivation of the general delta rule for gradient descent in multi-layer feed-forward networks which solves this problem.

7.5.8 The generalized delta rule

The error at the output of a multi-layer network can be computed as in the case for a linear network. However the output from a unit is given by

$$O_{pj} = F_j(NET_{pj})$$

where

$$NET_{pj} = \sum_i W_{ji}.O_{pi}$$

and squashing function F_j is a non-linear activation function. F_j must be both differentiable and non-decreasing (the former will soon become apparent when we require to calculate its differential). It was explained earlier that such a non-linearity is necessary if any benefit is to be gained from using multiple layers. We can write the change of error with respect to the weight change using the chain rule as

$$\delta E_p/\delta W_{ji} = \delta E_p/\delta NET_{pj} \,.\, \delta NET_{pj}/\delta W_{ji}$$

The second term can be written as

$$\delta NET_{pj}/\delta W_{ji} = \delta/\delta W_{ji} \,.\, \sum_k W_{jk}O_{pk} = O_{pi}$$

For clarity of representation, we shall now define

$$\sigma_{pj} = -\delta E_p/\delta NET_{pj}$$

Thus

$$-\delta E_p/\delta W_{ji} = \sigma_{pj}O_{pj}$$

Therefore to achieve a gradient descent in error E we must make weight changes according to the relationship

$$\Delta W_{ji} = \Gamma \sigma_{pj} O_{pi}$$

Where Γ is a constant known as the learning rate. We must now calculate the error term σ_{pj}. There is a simple recursive calculation for these weight changes that may be performed by the back-propagation of errors through the network. Again using the chain rule

$$-\delta E_p/\delta NET_{pj} = -\delta E_p/\delta O_{pj} . \delta O_{pj}/\delta NET_{pj}$$

where

$$\delta O_{pj}/\delta NET_{pj} = F'_j(NET_{pj})$$

This is the derivative of the squashing function F_j evaluated at the input NET_{pi} to that unit. There are now two cases that follow:

Case 1. If the unit u_j is an output unit, then the definition of E_p simply gives

$$\delta E_p/\delta O_{pj} = -(T_{pj} - O_{pj})$$

Therefore

$$\sigma_{pj} = (T_{pj} - O_{pj})\, F'(NET_{pj})$$

Case 2. If the unit u_j is not an output unit, then we must again use the chain rule

$$\sum_k [\delta E_p/\delta NET_{pk} . \delta NET_{pk}/\delta O_{pj}] = \sum_k [\delta E_p/\delta NET_{pk} . \delta/\delta O_{pj}] \sum_i W_{ki} O_{pj}$$

$$= \sum_k \delta E_p/\delta NET_{pk} W_{kj} = \sum_k \sigma_{pk} . W_{kj}$$

This yields

$$\sigma_{pj} = F'[NET_{pj}] \sum_{k} \sigma_{pk} W_{kj}$$

Therefore, there are three main equations that describe the operation of the generalized delta rule.

$$\Delta W_{ji} = -n.\sigma_{pj} O_{pj}$$

where Δ represents the change. For an output unit

$$\sigma_{pj} = (T_{pj} - O_{pj}) F'(NET_{pj})$$

and for units that are not output units

$$\sigma_{pj} = F'(NET_{pj}) \sum_{k} \delta_{pk} W_{kj}$$

The application of the generalized delta rule involves a forward and backward pass through the network. In the forward pass, the outputs O_{pj} for each unit are calculated and stored. The overall output from the output layer is then compared with the target patterns T_{pj}. An error signal is then propagated backwards through the network.

7.5.9 Sigmoid squashing function

A suitable squashing function that may be used in the MLP is the sigmoid non-linearity. One definition used by Rumelhart et al. (1986) is given by

$$O_{pj} = 1/[1 + \exp(-NET_{pj})]$$

where we redefine NET_{pj} to include a threshold term T_j as follows;

$$NET_{pj} = \sum_i W_{ji}O_{pi} + T_j$$

The derivative of the sigmoid function is given by

$$\delta O_{pj}/\delta NET_{pj} = O_{pj}(1 - O_{pj})$$

When this is substituted into the generalized delta rule equations, it gives the following definitions for the error. In the case of an output unit

$$\sigma_{pj} = (T_{pj} - O_{pj})O_{pj}(1 - O_{pj})$$

In the case of a unit that is not an output unit

$$\sigma_{pj} = O_{pj}(1 - O_{pj}) \sum_k \sigma_{pk}W_{kj}$$

A flow chart showing the operations involved in training an MLP with back-propagation is shown in figure 7.9.

7.5.10 Starting condition for networks

The initial weights in the network should ideally be set as near as possible to the final positions that will be found by the training (Lippmann, 1987). Naturally, it is typically not possible to know what they should be. In this event, a safe starting point is to use small random weights. It is important that the initial set-up is not symmetrical, or it will not be able to escape from a symmetrical configuration. In addition, no weights should be given an initial zero value, because they cannot escape from this value since any weight change is always zero (Rumelhart et al., 1986).

7.5.11 Performance of the MLP

Although there is no convergence theorem for the MLP, this technique has been shown to be useful in many applications. The back-propagation algorithm has been tested widely, and found to give good results for many tasks (Rumelhart, Hinton & Williams, 1986).

As for any minimization technique that works by gradient descent, the back-propagation algorithm can run into problems when local minima in the error function are encountered. However, these problems can be reduced by some basic steps. If more hidden units are used that are really necessary, local minima will often result in acceptable performance. Secondly training runs from a number of different random weight starting points can lead to different solutions, and the best can then be selected. Lowering the gain scaling terms can also help avoid local minima. When the training data set is relatively small and the patterns are presented many times, the presentation sequence should be random to prevent cyclic adaption (without learning) from taking place (Ridgeway, 1962). Additional algorithms have been proposed to speed up the training (Parker, 1986).

7.5.12 Using "Momentum terms" during training

The basic technique of gradient descent can be improved by adding a momentum term to the weight changes. This effectively provides low-pass filtering on the weight changes which helps avoid local minima in the error function. Thus

$$\Delta W_{ji}(n+1) = \Gamma \sigma_{pj} O_{pj} + \alpha . \Delta W_{ji}(n)$$

where Γ is another learning rate constant, which has the effect of dampening oscillations of the training weights, and α is the momentum term (notice that $\Gamma = 1$ and $\alpha = 0$ results in normal training). Rumelhart et al. (1986) quote suitable ranges of $0.05 \leq \Gamma \leq 0.75$ and $0 \leq \alpha \leq 0.9$. However, the optimal values of these constants are typically dependent upon the problem.

The effect of Γ and α on the learning process can be understood by considering two features of the error surface.

The first is the case of a ravine. When the value of the learning rate constant Γ is small, the learning trajectory proceeds satisfactorily down the ravine. However, when Γ is large, the trajectory can oscillate from side to side which slow down learning. This can be offset by the inclusion of the momentum term α.

The second feature of the error surface to be considered is a plateau. This is typically encountered as the search approaches the global minimum. Under such circumstances, the gradient becomes very small which results in further training becoming very slow. In this case, a large value of the learning rate parameter Γ is desirable. However, it is clear that a "good" value for this case will be too big for the ravine case. Consequently, an adaptive scheme is desirable, which alters the parameters depending upon the nature of the local error surface.

7.5.13 Adaption of the learning rate and the momentum term

One such technique is due to Chan & Fallside (1988) and operates by dynamically adjusting the momentum term and learning rate. Their approach is to monitor the angle Θ_n between the current gradient and previous weight update and the angle ϕ_n between successive weight updates, the first of which give an indication of the nature of the error surface, and the second gives an indication of the effect of the smoothing produced by the momentum term. These are given by the equations:

$$\cos(\Theta^n) = \frac{\text{grad}E_n . \Delta W_{n-1}}{|\text{grad}E_n| . |\Delta W_{n-1}|}$$

$$\cos(\phi^n) = \frac{\Delta W_n . \Delta W_{n-1}}{|\Delta W_n| . |\Delta W_{n-1}|}$$

where $gradE_n$ is the current gradient. They argue that the learning rate Γ_n should be reduced at a ravine and increased at a plateau, which corresponds to $90° < \Theta_n < 360°$ and Θ_n -> 360° respectively. Therefore a suitable adaption is given by

$$\Gamma_n = \Gamma_{n-1}(1+0.5\cos\Theta_n)$$

To adapt the momentum terms, they choose the relationship

$$\alpha_n = \Gamma_n \cdot \zeta_n$$

where ζ_n is given by

$$\zeta_n = \zeta_0 \frac{|gradE_n|}{|\Delta W_{n-1}|}$$

and $0 \leq \zeta_0 \leq 1$.

This adaption avoids the weight update from being dominated by the momentum term.

7.5.14 The number of patterns used to estimate weight changes

Chan & Fallside used the adaption scheme in conjunction with an updating of the weights over the **entire** training data set whenever practical, or over representative sub-sets of the data under those circumstances wherever such a scheme was not practical. The advantage of using the latter procedure is that it is then possible to make MLP weight changes over a relatively small set of patterns, but ones which are a good reflection of the possible range of patterns in the data set. This is better than making the update after each pattern, because the latter is not guaranteed to give a good gradient descent, and the direction of the weight changes tends to fluctuate widely between successive updates, which makes it impossible to use adaptive learning rate and momentum term learning schemes. In general one would not wish to update the weights only once per pass of all the data for large data sets, since this would result in very slow

learning. This is because it is only possible to alter the weights by a relatively small amount per update, and since the time taken to determine each update would be relatively large in this case, to perform enough updates to find a suitable solution would take a long time.

7.5.15 Selective emphasis training of the MLP

Training times for the MLP can become long when the training data set is large and pattern vectors are of high dimensionality. A method to increase training speed was developed, and it is called selective emphasis training by the author (Howard, 1990,1991). It operates by changing the relative emphasis of different pattern vectors during training the MLP. This is achieved by scaling the weight changes that result from a given pattern by a factor that depends upon the estimated importance of that pattern. The importance of the pattern vector is estimated with regard to several considerations.

The importance of a pattern to the classifier can be specified a priori if it is possible to have an idea of how reliable that output class is. For example, one has more confidence that the centre of a voicing region should be classified as voicing present than at the edge of the voicing region (this principle is illustrated in the case of period excitation markers in figure 9.11 in chapter 9). Consequently, it may make sense to de-emphasise boundaries, since their precise location is often difficult to assess.

In the case of a pattern class that only occurs infrequently, their contribution to the weight changes may be small compared to the contribution from the other class which occurs more frequently. In such cases, it is sometimes useful to emphasize the weight changes arising from different pattern classes by different amounts. However, since the effect of this is to alter the occurrence probability of the pattern classes, such a scheme cannot be used alone because it can results in a large number of false alarms being generated.

It has been found valuable to concentrate the training on the patterns that are falsely

recognized, and not overwhelm the MLP with less important weight changes from the data that is dealt with acceptably. This can be achieved by making the emphasis dependent on the output from the MLP as well as the target pattern class. Thus a pattern that gives rise to an output above a preset threshold gives rise to weight changes which are scaled differently than if the output was below the same threshold. In this thesis three thresholds were employed, one for period-marker-present target pattern classes, one for uncertain pattern classes and another for period-marker-absent classes (this is described more in chapter 9 and illustrated in figure 9.12). It is possible to arrange the emphasis such that patterns that give rise to outputs which are close enough to the targets give rise to weight changes that are ignored (and need not be computed, thus speeding up program operation). This makes it possible to reduce the contribution of certain regions in the training data to zero if required.

The parameters used with the technique are critical, and for the work here a low threshold L=0.1 and a high threshold H=0.9 were used to indicate when the output from the recognition is close enough to the binary targets for their corresponding updates to be ignored. A flow chart for the selective emphasis scheme is given in figure 7.10.

7.5.16 Similar techniques to selective emphasis

An algorithms due to Mays (1963) exhibits similarities to the selected emphasis method that has just been described. He called this "modified relaxation adaption".

In the modified relaxation adaption rule, the weight changes are not generally made in proportion to the error, but rather they are only changed if the output falls within the dead zone. That is:

$\mathbf{W}_{k+1} = \mathbf{W}_k$ if $|\text{output}| \geq \tau$

$\mathbf{W}_{k+1} = \mathbf{W}_k + \mu.\mathbf{X}_k/[|\mathbf{X}_k|^2]$ if $|\text{output}| < \tau$

For a value of the dead zone parameter $0 < \tau < 1$ and learning parameter $0 < \mu$ 2,

Mays showed this rule converged for linearly separable patterns.

This rule is similar to the selective emphasis method (that was discovered independently of Mays earlier result) although the selective emphasis technique in its general form uses a 2x2 matrix to specify the four cases that can occur, and associates a weight with each case. The most important point in relation to the selective emphasis technique is that in both cases something different is done depending upon whether or not the output from the network is close to the target, or far from the target.

7.5.18 Relationship between capacity and required training examples

A characteristic of pattern classifiers that is associated with their capacity is their ability to generalise. That is, the ability of a classifier to respond appropriately to an input pattern vector that was not used to train the classifier. If this ability was not needed, then we may just as well use a look-up table, as mentioned in an earlier section.

To achieve good generalization, the training data set should contain many more patterns than the capacity of the network (Cover, 1964; Brown, 1964; Nillson, 1965) that is for a two class classifier

$$N_p >> N_w$$

Where N_p is the number of patterns and N_w is the number of weights in the network. This is because the training examples must constrain the operation of the network to behave as desired. This will not be possible if the number of degrees of freedom of the network is greater than the number of training patterns. Under such conditions, the initial weights can interfere with the generalization capabilities of the network (Baum & Haussler, 1988).

7.5.19 Generalization of the training data to testing data

Baum & Haussler (1989) derived a relationship between the number of weights W in

a network, the accuracy ε of the classification and the required number of training examples m. Their calculations are based on considerations of the capacity of the network in terms of the number of dichotomies (the dichotomization capacity of a pattern classifier is the number of possible classifications it can distinguish) it can implement (Cover, 1965).

Provided that the training examples are taken from the same distribution as the testing examples, then for fully-connected feed-forward nets using any learning algorithm, they arrived at the relationship that using any fewer than $\Omega(W/\varepsilon)$ training patterns would result in the failure to correctly classify, for at least some of the time, more than a $1-\varepsilon$ fraction of the future test examples, where Ω is some constant. If we ignore this constant and set $\Omega=1$ to get a rough estimate, then for an accuracy of 90% (that is ε = 0.1) this implies we need at least 10 times as many training examples as there are weights in the network. This figure agrees reasonable well with the rule-of-thumb adopted by Widrow (1987).

One approach to increase amount of training data available for pattern classifiers involves adding noise to the measurements in the input vectors (Elman &, Zipser 1987). Such an approach may help improve generalization capability if only a small training set is used. The pattern distribution generated by adding noise will in general be different from the true distribution and it is clearly preferable to use more real data instead.

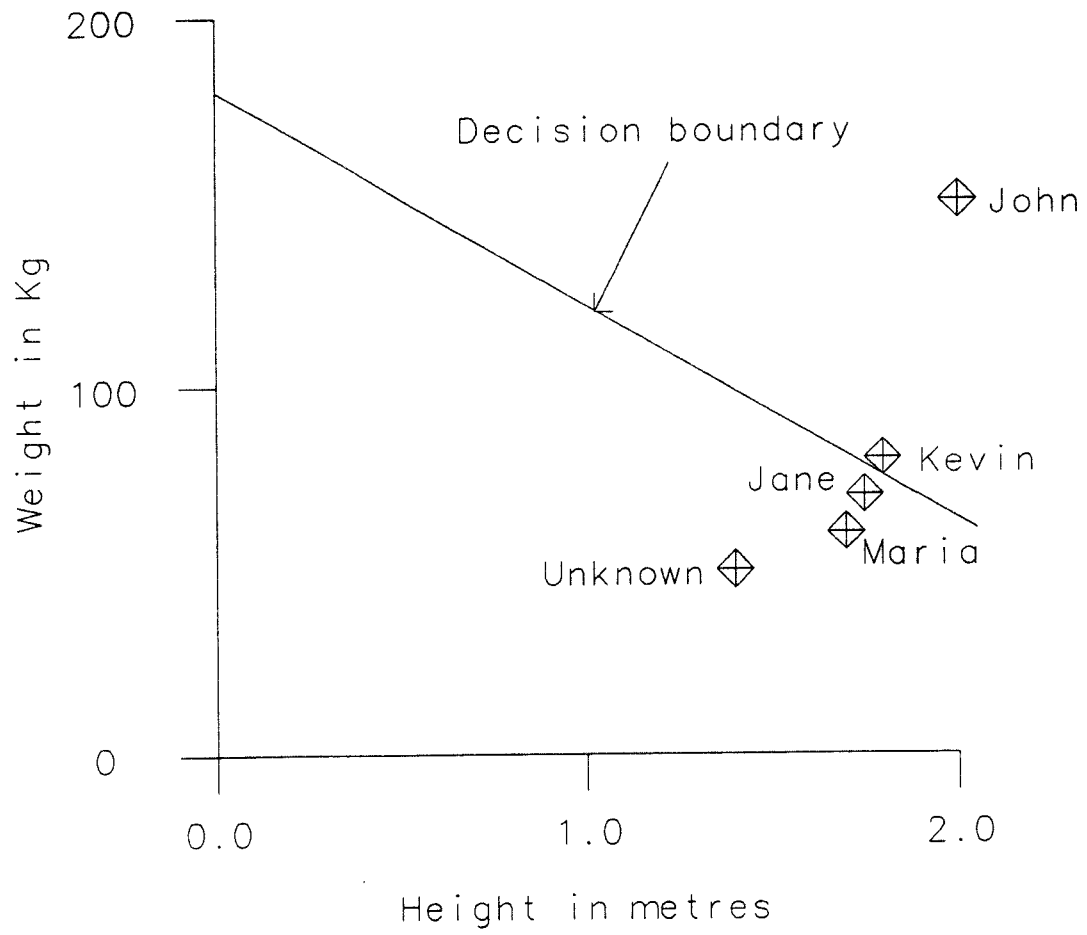


Figure 7.1 An example of a 2-dimensional patterns representing the height and weight of a group of subjects.

The points representing the two female and two male subject are shown, together with a possible decision boundary that separates the male and females. An unknown point is also shown, which falls in the female region.

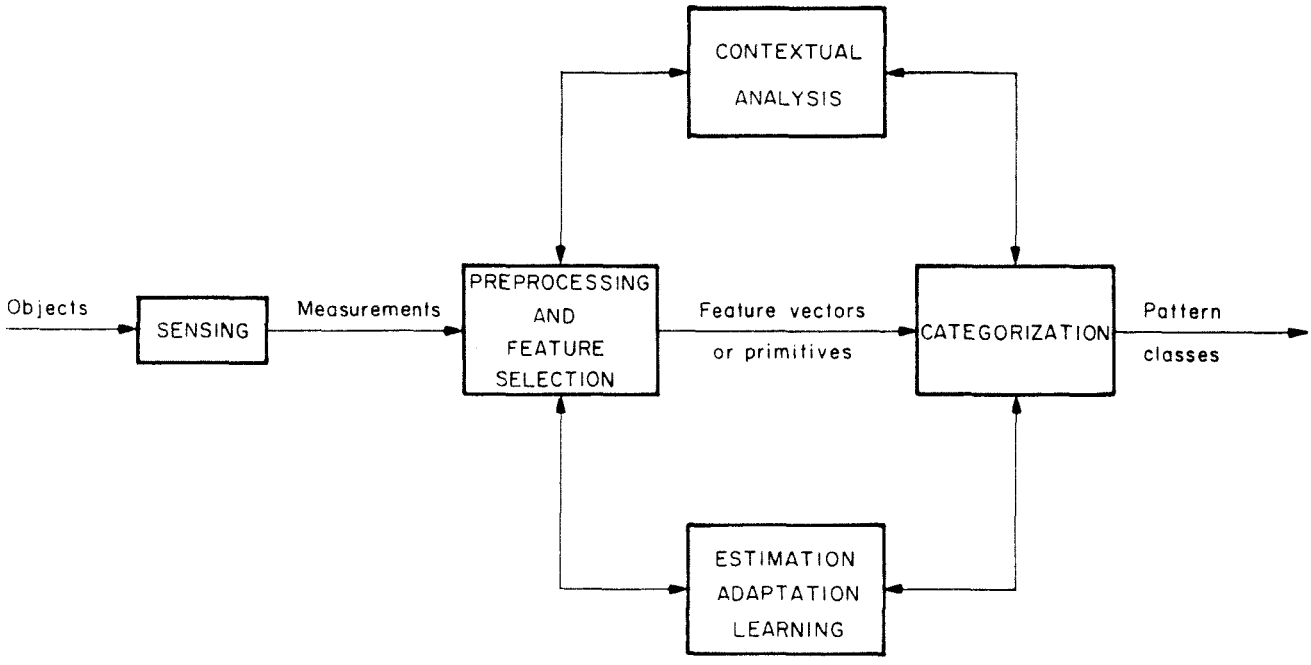


Figure 7.2 Schematic diagram of all the operational stages in a pattern recognition system.

The object (or input phenomenon) under investigation is first sensed and then the measurements are pre-processed. This stage can emphasise important aspects of the measurements. During training (learning) mode, the pre-processed data (together with the target pattern classes in a supervised system) is used to establish the patten classification decision boundaries (shown as the lower path). During recognition mode, the pre-processed measurements are classified using the pattern classifier. Sometimes, contextual information is also used to provide additional information for the classification, as illustrated in the top path).

(Taken form Tou & Gonzalez, 1974).

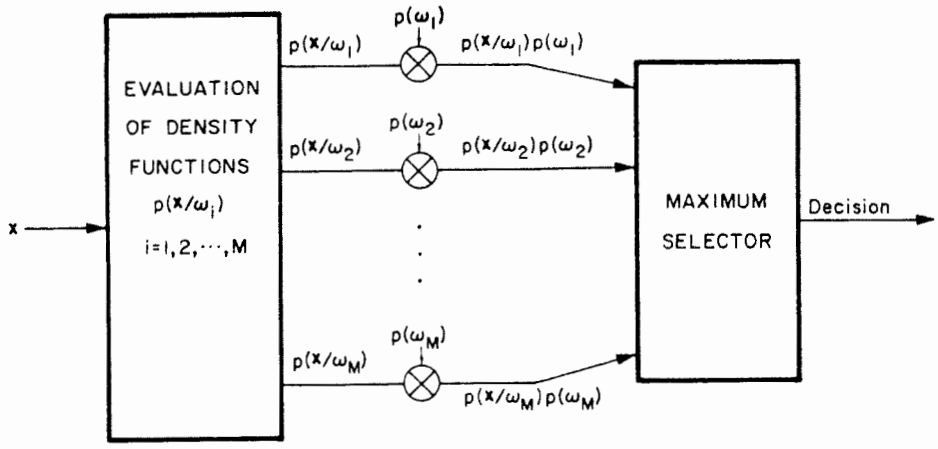


Figure 7.3 Schematic diagram for a multi-class Bayes' pattern classifier system. The implementation of the classifier involves calculating the probabilities of each possible class and then selecting the one with the higher probability.
(Taken form Tou & Gonzalez, 1974).

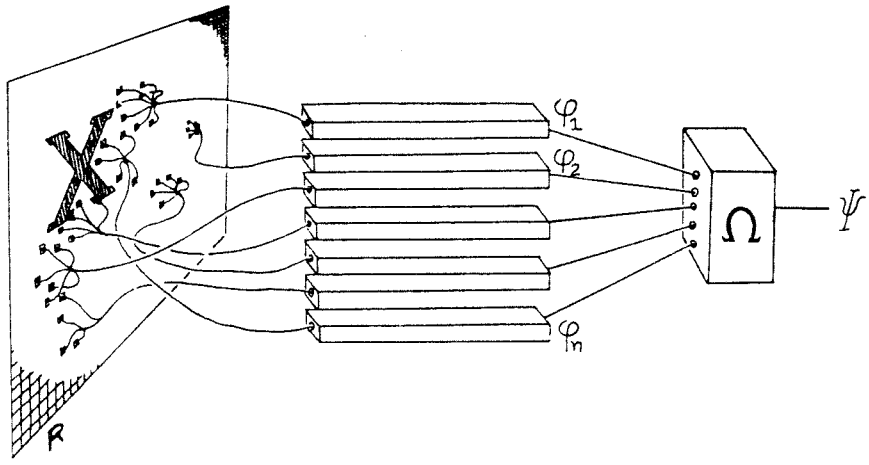
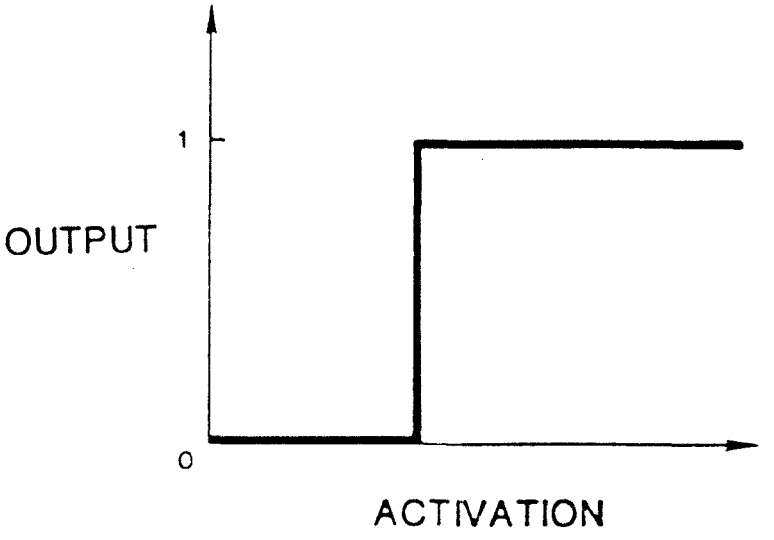


Figure 7.4 Schematic diagram for a perceptron.
The retina (R) is the source of the inputs. These are then weighted and then summed to give an overall output.
(After Rosenblatt, 1958).



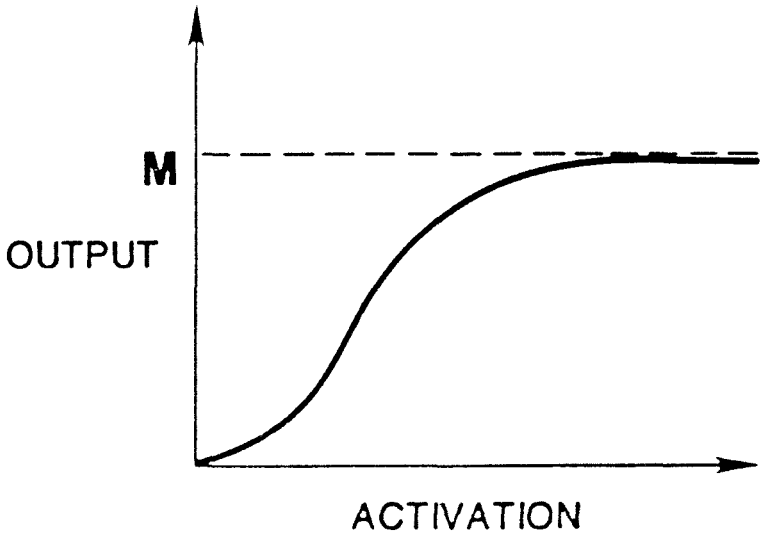


Figure 7.5 Comparison of threshold function with a sigmoid function. (Taken from Rumelhart & McClelland, 1986).

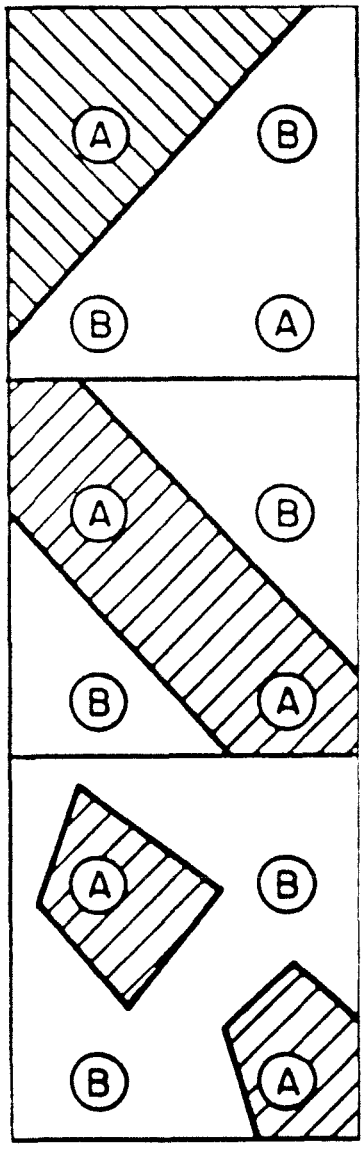
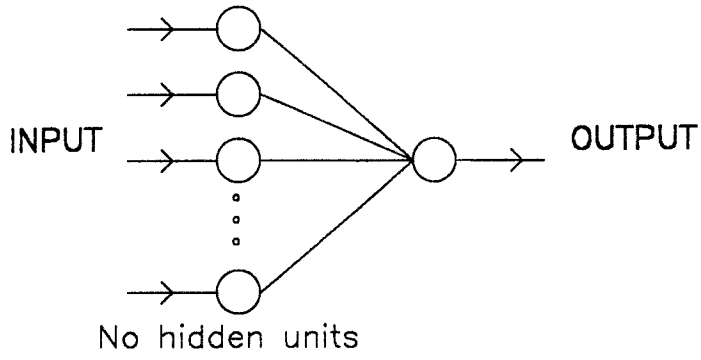
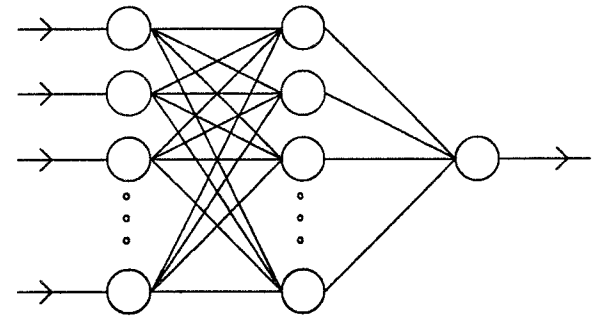


Figure 7.6 Diagram showing the pattern classifier decision boundaries required to solve the XOR problem.

Patterns of the two classes are represented as A and B respectively. Patterns of class A contains the points (1,0) and (0,1), whereas pattern class B contains the points (0,1) and (1,1). To compute the XOR function, it is necessary to discriminate the A and B patterns. They cannot be discriminated using a linear decision function, as shown in the top diagram, and a more complex boundary is required (shown in the lower two diagrams).

(After Lippmann, 1987).





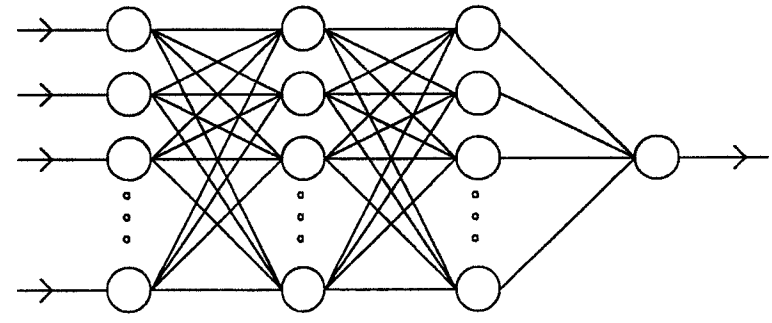
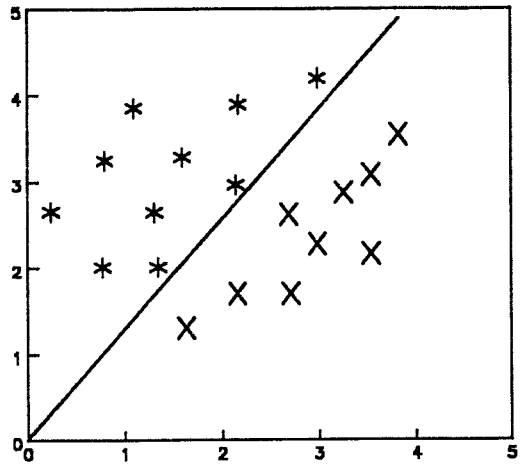
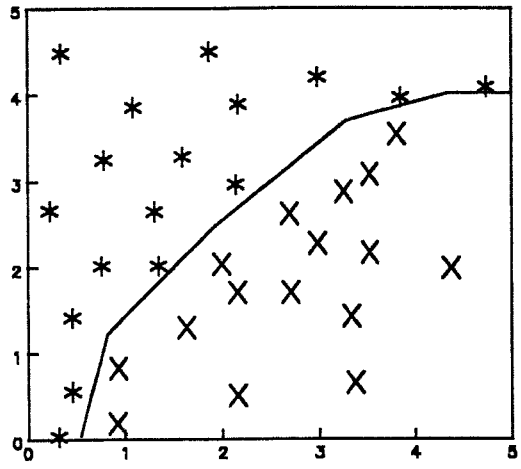


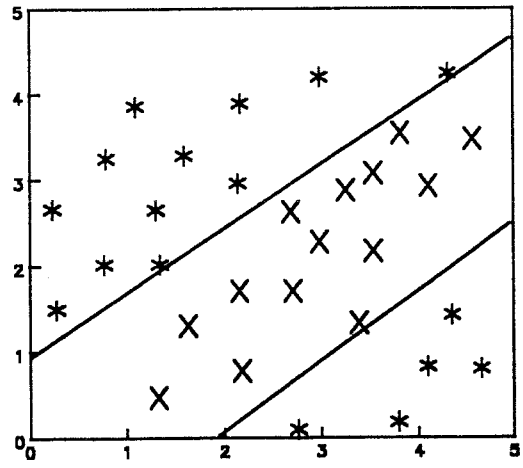
Figure 7.7 Schematic diagram of multi-layer perceptron with different number of layers. The figure shows a net with no hidden units, a net with one layer of hidden units, and a net with two layers of hidden units.



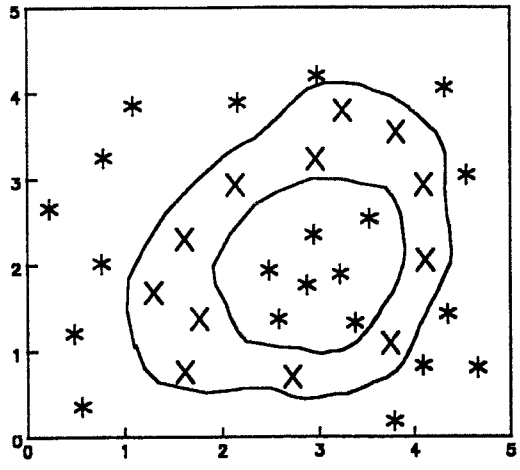
LINEAR
No hidden units



CONVEX OPEN
1 layer of hidden units



CLOSED
1 layer of hidden units



ARBITRARY
1 Or 2 layers of hidden units

Figure 7.8 Schematic diagram showing relationship between decision boundaries and layers in the multi-layer perceptron.

A network with no hidden units can only implement linear decision surfaces. One or two layers of hidden units can implement decision surfaces of arbitrary complexity.

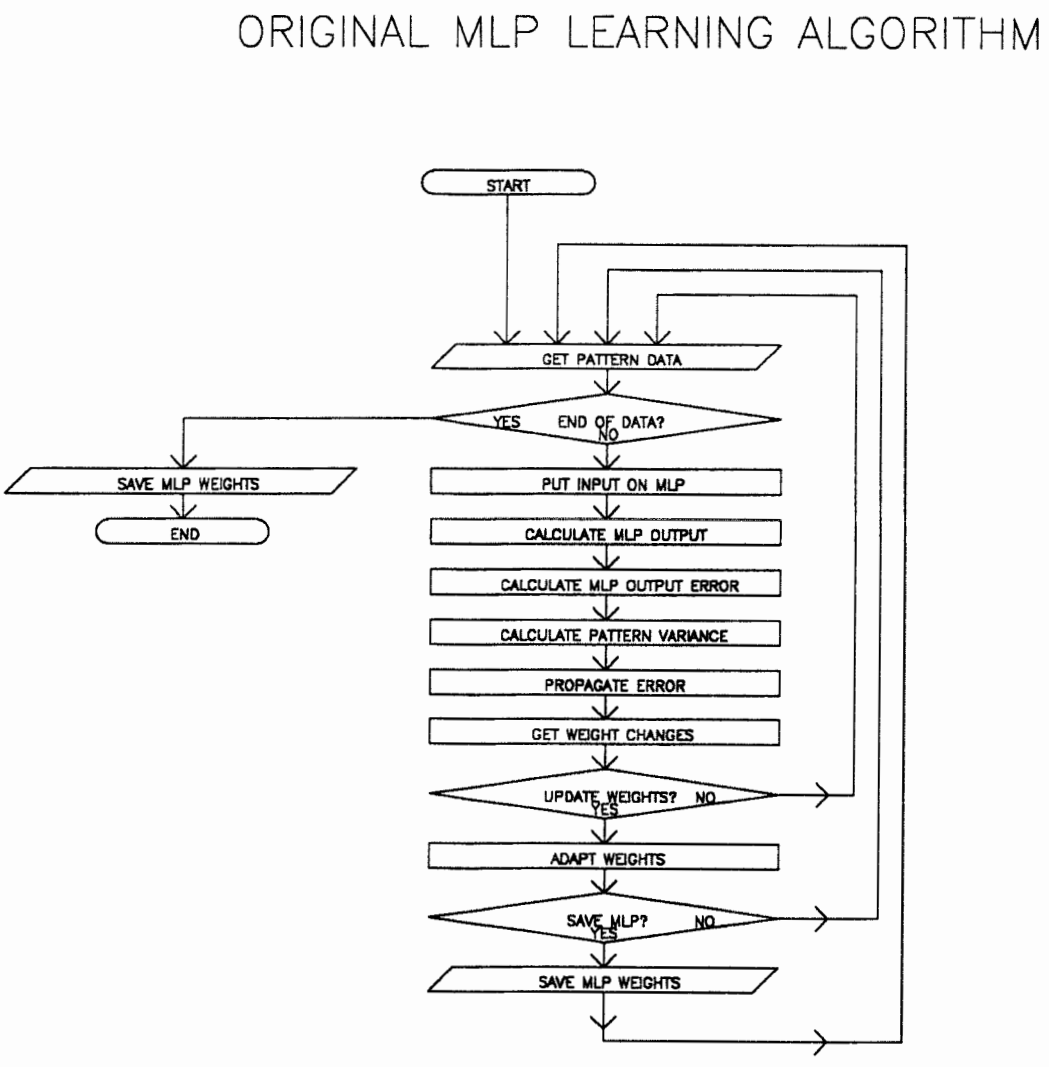


Figure 7.9 Flow chart for multi-layer perceptron learning algorithm using back-propagation.

The operations required to train an MLP using back-propagation are explained in the main text.

(Program by M A Huckvale).

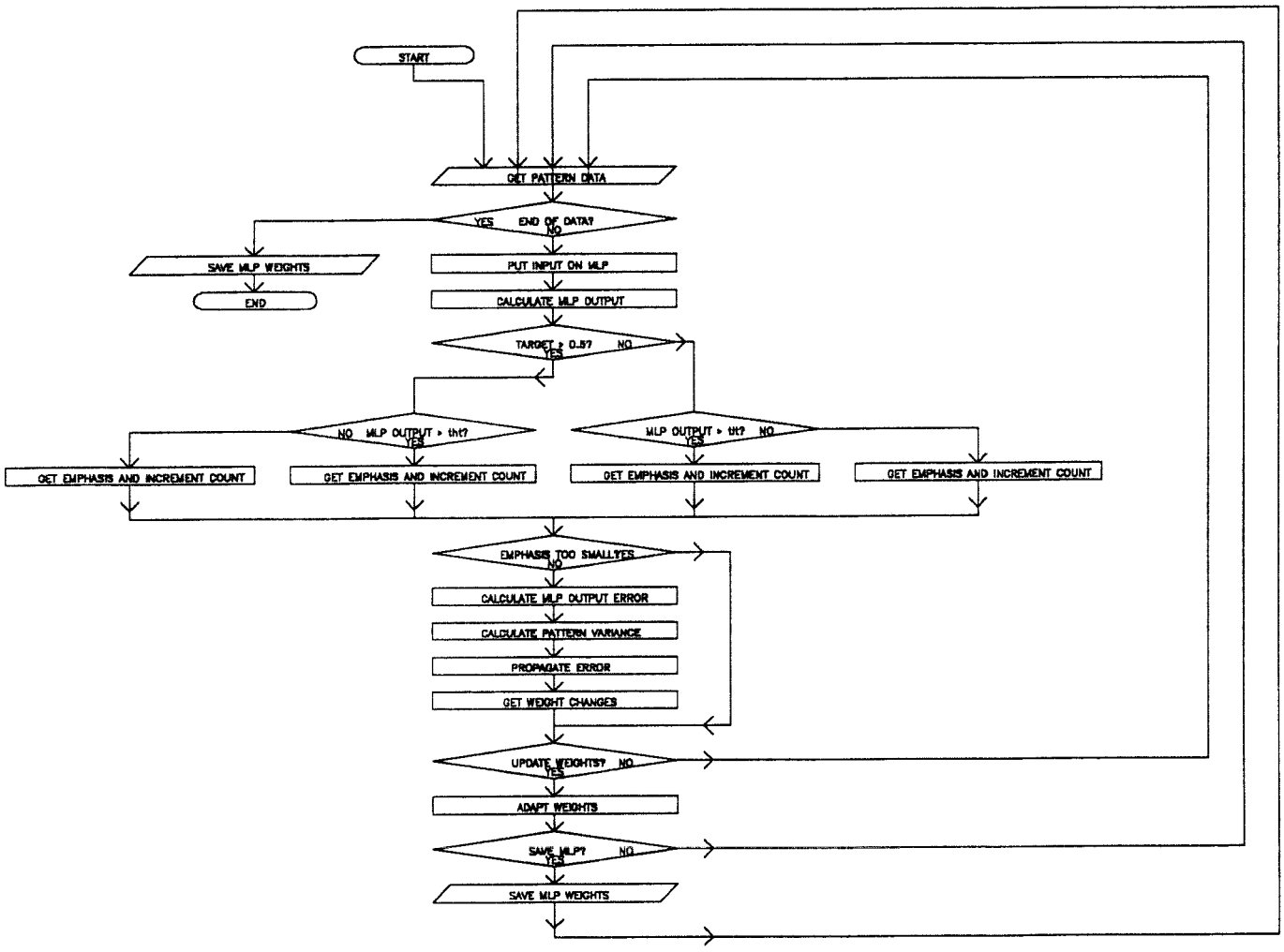


Figure 7.10 Flow chart for multi-layer perceptron learning algorithm using back-propagation with selective emphasis.

The essential difference with normal back-propagation training is that weight updates are made to depend upon the response to the input pattern and the target output class. (Program by I S Howard).

CHAPTER 8: BASIC CONCEPTS AND PRELIMINARY EXPERIMENTS IN SPEECH FUNDAMENTAL PERIOD ESTIMATION USING PATTERN CLASSIFICATION

8.1 BACKGROUND TO THE DEVELOPMENT OF THE MLP-Tx ALGORITHM

8.1.1 Introduction

This chapter provides the background reasoning behind the design and preliminary development of an algorithm that performs speech fundamental period estimation using pattern classification. This algorithm is called the MLP-Tx algorithm by the author, because it uses a multi-layer perceptron classifier (MLP) to estimate the speech excitation markers (Tx). It is shown that the estimation of the fundamental period of speech can be performed using a system with the same basic structure as one used earlier to perform the task of voicing determination, providing the system parameters in the latter are suitably adjusted. Some initial experimental results generated using a preliminary configuration of the MLP-Tx algorithm are given. Limitations of this initial configuration and of the first experiments are then discussed.

8.1.2 Initial work task at UCL

Preliminary research by the author was concerned with voicing determination algorithms that made use of the laryngograph, in order to provide a reference voicing system against which future speech-based systems could be compared. Some early schemes developed made use of envelope detection of the laryngograph waveform. However it was soon appreciated that such schemes can smear out the onset and offset of voicing in time. An example of this is shown in figure 8.1. In addition, gross larynx movements, which can easily be separately identified in the original laryngograph time waveform, become indistinguishable from true voicing regions after passing through the envelope detector. Simple high pass filtering can be useful but these problems can both be avoided if the presence of voicing is based upon the detection of individual cycles in the laryngograph waveform by using features that signify their shape, rather than using the gross laryngograph envelope. In this case, the onset of voicing can be

determined to within a fundamental period and the problem of time-smearing by the envelope detector does not arise. Similarly, because the laryngograph cycles are detected on an individual basis, spurious cycles can be identified by their isolation.

One feature that reliably occurs within each laryngograph cycle (in normal voice) is the point of maximum positive gradient, as discussed in previous chapters. At this point it became clear that a system that would provide a good estimate of the voicing regions (using the laryngograph waveform) was one that also performed fundamental period estimation. Various algorithms to perform this task were then implemented on the computer system, and the final result of these developments were the laryngograph-based reference period marker programs described in chapter 5. This pair of programs provided a reliable means to label data with fundamental period markers (Tx).

8.1.2 Use of the laryngograph to indicate voicing

The fundamental period markers were converted to labels to indicate gross voicing using a simple procedure. The onset of a voiced region was defined as the time of occurrence of the first marker in a sequence of markers. The end of a voiced region was defined as the time of occurrence of the last marker in the sequence, plus the last period value. Markers were only considered to occur within the same voiced region if they were closer than 20ms in time. This algorithm was also implemented on the computer system, and together with the fundamental period estimation algorithms, it provided a means to automatically label laryngograph data with annotations signifying the presence of larynx voicing. Provided speech and laryngograph were recorded simultaneously, this gave the means to label a speech pressure waveform with voicing annotations.

8.1.3 Speech voicing determination using pattern classification

After the establishment of a reliable system to estimate voicing from the laryngograph signal, attention was turned to the estimation of voicing using the acoustic speech signal. Various approaches have been employed in the past to tackle this problem. Many of these schemes involved the analysis of a single parameter of the speech signal, and often

operate in conjunction with fundamental frequency estimation algorithms; for example, the cepstrum algorithm (Noll, 1967), or autocorrelation (Sondhi, 1968). Such single feature schemes typically generate an estimate of the regularity of the input signal, and the speech is classified as voiced or unvoiced depending upon whether this value is above or below a preset threshold. One more sophisticated approach that appeared particularly encouraging was the statistical pattern recognition approach of Atal & Rabiner (1976), since their scheme provided the means to combine several features in an optimal and automatic way (by training the classifier). The features they employed were the speech energy, zero-crossing rate, autocorrelation coefficient at one unit sample delay, the first LPC predictor coefficient and the energy of the LPC prediction error. These were defined on a frame-by-frame basis and used to generate an input vector which was then fed to the input of a pattern classifier. The classifier was initially provided with labelled speech data and was trained to perform the desired voicing determination task.

A scheme using the then newly emerging pattern recognition technique, the multi-layer perceptron, was used by Peeling & Bridle (1986) to recognize several acoustic-phonetic qualities of the speech signal, including voicing. Their system employed input pre-processing using a 19-channel vocoder, and its performance was shown to be high.

8.1.4 Experiment using pattern classification to estimate voicing

The work by Peeling & Bridle (1986) prompted the author and a colleague (Mark Huckvale) to set about to develop and test voicing determination algorithms that employed 19-channel vocoder input pre-processing (a schematic diagram for which is shown in figure 8.2), and used either a Bayes' classifier for Gaussian pattern or a multi-layer perceptron to implement the pattern classifier. This work was reported by Howard & Huckvale (1987), but it is briefly described here because it forms a good basis for the introduction of the MLP-Tx algorithm.

Database for voicing determination experiments

To provide a set of training data for the classifiers and testing data so that the systems could be evaluated, five male speakers were recorded in an anechoic room using a high quality B&K 4134 condenser microphone, together with the output from a laryngograph. The microphone was maintained 15cm from the subjects lips and was located equally forward and to the side to avoid wind noise from the subject's breath. Each speaker was require to read "The Rainbow Passage" twice (Mermelstein, 1977; It also appears in appendix A3); once to provide the training data and once to provide the testing data. This text was chosen because it was phonetically balanced. Both channels of the data were then low-pass filtered at 5kHz using eighth-order Butterworth filters and acquired onto a Masscomp 5500 computer via a 12-bit A/D converter running at a 10kHz sampling rate.

The voicing regions on all the data were then automatically labelled using the laryngograph-based techniques described earlier. Next, the speech data was analysed using a 19-channel vocoder, which generated 19-element output frames at 10ms intervals. Either one or three input frames were used as the input to the pattern classifiers. The additional adjacent frames included in the input vector provided context (Boulard & Wellekens, 1989) for the recognition task. The Bayes' classifier was trained by estimating the mean vectors and covariance matrices for the voiced and unvoiced pattern vectors. Various configurations of the MLP were investigated. The MLPs were trained using back propagation with weights updated after each pattern presentation. Passes over the training data were made until the networks showed no sign of further learning.

The initial results from this voicing determination experiment were most encouraging. The results for the Bayes' classifier and the MLP classifiers are given as the receiver operating characteristics for the respective detectors, and are shown in figures 8.3 and 8.4 (Howard & Huckvale, 1987). The MLP was shown to give better performance than the Bayes' classifier for this task.

8.2 INITIAL MLP-Tx EXPERIMENTS

8.2.1 Similarities between voicing determination and fundamental period estimation

The problem of detecting the points of excitation in the speech signal is somewhat akin to voicing determination, except the event to be detected (the excitation marker) is essentially impulsive rather than a steady-state region. Because the precise location of the excitation marker is essential to achieve an accurate fundamental period estimate, a much higher resolution is needed than in the case of voicing determination. Formulating the problem of period estimation in this way has the advantage over short-term analyses that the excitation points can be detected on a period-by-period basis. In addition, it is the actual excitation point in the speech waveform that is estimated using this approach, as opposed to an arbitrary repetitive point which is the function of many time-domain period estimators (see chapters 3 and 4).

It was desirable to use previously written software for this task, to reduce the programming workload. The system used for voicing determination was consequently modified to perform speech fundamental period estimation. The MLP-based voicing determination scheme involved the classification of the input speech signal into voiced or unvoiced frames, each lasting 10ms. By reducing the frame duration, and modifying the pre-processing, the system could be used to classify frames into those that contained an excitation marker, and those that did not.

8.2.2 Initial system structure

The two questions that then arose were what would constitute a suitable set of input pre-processing filters, and what should the frame rate be. The original vocoder analyzer was clearly an unsuitable pre-processor for such a task, because it was specifically designed to lose information regarding the excitation present in the input speech. That is, any temporal fluctuations in the output from a channel after the full-wave rectifier stage are smoothed out using a 50Hz cutoff low-pass filter. In addition, the output frame rate of 10ms was much too low to be of any value in fundamental period estimation. Such a frame rate would give rise to a frequency quantization error of 100% at 100Hz.

Wideband spectrogram

In the task of deciding upon the characteristics of an appropriate set of pre-processing filters, consideration was then given to the appearance of a wideband (that is, with an analysis bandwidth of around 300Hz) spectrogram for (male) voiced speech. Such a spectrogram shows a vertical striation whenever there is a well defined excitation point in the input speech (see figure 8.5), and it is widely appreciated that such a spectrogram can be used to give a crude estimate of the fundamental period (Borden & Harris, 1980). By using input pre-processing to the fundamental period estimation algorithm that retained temporal information in the same way, the problem can be viewed as the detection of the vertical striations (this considers the problem to simply be the image analysis of a wideband spectrogram. There are, of course, other and probably better ways to view the problem, and other potential pre-processing schemes are discussed in chapter 9).

MLP-Tx wideband filterbank

The filterbank that was initially used as a pre-processor for the fundamental period estimation task constituted an approximation to a wideband filterbank that was implemented using the 19-channel vocoder program. Filter band-widths of 300Hz are typically used in a wideband spectrogram, although in the wideband filterbank the bandwidths of the higher filters were increased slightly from 300Hz to reduce the number of filters required to cover the desired frequency range and also to mimic the behaviour of auditory filters (Moore & Glasberg, 1989). There was an imperative need to keep the number of channels to a minimum, because the computational load was proportional to the number of channels. Consequently much coarser steps between the filter centre frequencies were used than in a genuine wideband spectrogram (which may typically employ in the order of 100). The frequency range of interest for voiced speech covers the frequency range of about 40Hz to 3kHz. The filter channels were selected to cover this range, with intersection at their -3dB points to prevent any signal frequencies from being poorly represented in their outputs. The filterbank that was consequently arrived at comprised nine fourth order IIR band-pass Butterworth filters

with -3dB points of 40-300Hz, 300-600Hz, 600-900Hz, 900-1200Hz, 1200-1600Hz, 1600-2000Hz, 2000-2400Hz, 2400-2800Hz and 2800-3300Hz.

Selection of frame rate

The outputs from the filters were half-wave rectified, smoothed by means of a second order low-pass Butterworth filter with a cut-off frequency of 1kHz and then down-sampled to 2kHz. Half-wave rectification was employed as opposed to the full-wave rectification originally employed in the vocoder, because the filter outputs would often be sinusoidal (or more generally symmetrical), and therefore full wave rectification would double their periodicities (Hess, 1984; This issue was also discussed in chapter 4). The smoothing was carried out to prevent aliasing of the signal in the subsequent downsampling to 2kHz, which was performed to achieve data-reduction. This sampling rate was a compromise between a usable time-resolution and the amount of computation required by the subsequent pattern classifier stage. It is to be noted than several established fundamental frequency estimation algorithms also perform downsampling to 2kHz before their basic extraction stages, again to reduce the computational load (for example SIFT, Markel, 1972). The sampling resolution issue is looked at in more detail in chapter 9. However, it must be emphasised here that the immediate objective of the work was to produce an aid for the profoundly deaf. A maximum useful auditory input of 500Hz is well catered for by this sampling frequency.

Figure 8.6 shows a close up of part of a small piece of speech and associated signals. Trace A is the speech pressure waveform, trace B the laryngograph (Lx) waveform. The output from the filterbank is shown as a grey-level display in item C. It can be seen that temporal variation concerning the excitation is retained. Trace D shows the reference period markers generated from B.

Pattern vector generation

Since the pattern classifier only receives information about the input speech from within its input window (because it has no memory of past inputs), in order for it to perform

its task of detecting the excitation points, it is necessary for the window to span a width greater than just a single 0.5ms frame. The latter would give little, if an, evidence of the filterbank time response due to an excitation, which manifests itself over several milliseconds. Initially a symmetrical input window of 10.5ms was used to generate the input to the pattern classifier, because it spans about one period of speech at the lowest fundamental frequency likely to be encountered for many speakers (that is, 100Hz), although some male speakers reach lower frequencies than this and creaky voice may typically be at 30Hz. Thus the initial input vector was comprised of the current frame of wideband filterbank data together with 10 frames back in time and 10 forward in time (that is, 21 frames in all), with the current frame in the centre of the input window (there was computational advantage to be gained by keeping the input window as small as possible. Issues concerning the input window size are discussed further in chapter 9).

8.2.3 Preliminary attempts at fundamental period estimation

The initial multi-layer perceptron configuration chosen was selected with regard to the configurations that gave reasonable results for voicing determination (Howard & Huckvale, 1987). These experiments indicated that at least one layer of hidden units would be useful and an initial arbitrary guess at 8 was made. Hence the very first trial of the MLP-Tx system made use of an MLP with 189 inputs, 8 hidden units and 1 output. The training (and in this case testing) data consisted of about 2 seconds of close microphone anechoic speech and laryngograph signal for 1 male speaker that had been acquired using a 12 bit A/D converter at a 10kHz sampling rate.

Labelling training and testing data with excitation markers

The output pattern classes for the data were labelled automatically using the reference period marker algorithm (described in chapter 5) operating on the output of the laryngograph. In this case, an output class was defined every 0.5ms. Whenever a period epoch marker was present within a frame, the target pattern for that frame was set to a value of 1. Otherwise the target patterns were set to a value of 0.

The training was carried out using back propagation with an update after each pattern presentation using learning rate $\Gamma = 0.05$ and momentum term $\alpha = 0.9$. The recognition output obtained after 32 training cycles appears in figure 8.7. This was the very first sign that the task could be performed, and encouraged further investigations. After running a set of similar experiments, it was found that better results on the small piece of data could be achieved using a MLP network with a larger input window of 41 frames. In addition two layers of hidden units gave output values from the MLP that more consistently reached their period-marker-present training target values of 1 than could be achieved using only one hidden layer. Finally a network was chosen with 369 inputs, two hidden layers each of ten units and 1 output. Adjacent layers were fully interconnected. The schematic diagram for this configuration of the MLP-Tx algorithm is shown in figure 8.8. The overall system block diagram is shown in figure 8.9.

8.3 FIRST MLP-Tx EXPERIMENTS ON A LARGE DATABASE

8.3.1 Data for the MLP-Tx experiment

To reliably evaluate the system, a much larger set of data was required to train the algorithm, and an additional different set of data was required to test it. From the results given in chapter 7 concerning the amount of training data needed to give good generalizations of a single output classifier, we require $N_p >> N_w$, where N_p is the number of training patterns and N_w is the number of weights in the classifier. In this case $N_w = 3690$, so we require $N_p >> 3690$. The data used to train and test the MLP-Tx algorithm was the same as that for the earlier voicing estimation task. Since the test data was independent from the training data, any limitation due to insufficient training data does not affect the legitimacy of the test results. The "Rainbow passage" training data contained approximately 3×10^5 0.5ms frames of data, and therefore a similar number of different pattern vectors. However, there were many more patterns corresponding to the period-marker-absent case than for the period-marker-present case. Altogether, there were about 8000 patterns corresponding to the period-marker-present case. This number is still twice the number of weights in the classifier. Consequently, even assuming the number of period-marker-present patterns were the most important,

the training data still does not violate the requirement that the number of patterns should be greater than the number of weights in the classifier.

Adding noise to the speech signal

To give a more realistic task for the MLP-Tx algorithm, all copies of the original Rainbow passage data were contaminated with additive canteen noise at levels of 0dB and 20dB SNR, providing two sets (one at 20dB SNR and one 0dB SNR) of data for training and similarly two sets of data for testing. The noise signal was recorded in the UCL refectory at lunchtime, and included impulsive noise and background conversations. The SNR was specified as follows: The power in adjacent 500ms frames of each signal was calculated, and the frame in each signal containing the maximum power was identified. The noise signal was then scaled so that the ratio of the signal and noise signals corresponded to the desired SNR, and the two signals were then added together to give a speech signal with the appropriate SNR. The spectrum for the canteen noise appears in figure 8.10. This plot was generated by averaging 203 frames, each of which was calculated using an FFT with an 80ms window on the noise signal. It can be seen that a lot of noise power lies in the frequency region below 1kHz, were most of the power in voiced speech resides.

All sets of noise-contaminated speech data were pre-processed using the wideband filterbank and the output period markers were generated automatically from the laryngograph signal.

8.3.2 Training the networks

Two separate networks with the same structure were trained for operation in the two different noise conditions. The training of these MLP networks was performed using 10 passes over the input data with learning parameters $\alpha = 0.9$ and $\Gamma = 0.05$. The weight changes were made after each pattern presentation, and all in all about 3 million pattern presentations were made. The MLPs were then trained using the Pattern Processing System described in appendix A.1, which was written in 'C' and ran under

Unix on a Masscomp MC5600 series computer. The training took several weeks. The error signal generated during training was used to gauge the completion of training (this was the normalized mean-square error between the targets and the MLP output averaged over all the training data).

The network trained on the 20dB SNR training speech was used to generate output for inputs from the 20dB SNR test speech, and the network trained on the 0dB SNR speech was used to generate output for inputs from the 0dB SNR test speech. The location of the period markers were determined from the MLP outputs by simply locating the local maximum peaks in the output signal that exceeded a threshold set at 0.5 (the midway point between the 0.0 and 1.0 values that can be generated by the MLP). In addition, a minimum period criterion was set, which avoided the generation of spurious pulses close to the main marker by setting the minimum detectable period to 2ms (which corresponds to a maximum fundamental frequency of 500Hz).

8.3.3 Qualitative evaluation of results

The discussion of performance presented in this chapter is intended to give an initial indication of the operation of the algorithm. The results from these preliminary experiments are first given in terms of visual examination of the output from the MLP network and of frequency contours generated from its period-by-period excitation speech estimates. Two quantitative comparisons were then made. More rigorous quantitative comparisons were discussed in chapter 6 and results using these techniques are given in chapter 9, in which a much wider set of environmental conditions and speakers are investigated.

Trace C in figure 8.11 shows the output of the MLP-Tx algorithm operating on a sample of test speech at a 20dB SNR which is shown in trace A. Trace D shows the period markers from the reference algorithm using the laryngograph waveform, shown in trace B. It can be seen that there is good correspondence between this reference and the output from the MLP-Tx algorithm. Trace E shows the period markers that can be obtained from the MLP-Tx output by detecting its peaks. Trace F shows the output of

the peak-picker algorithm (discussed in chapter 4), which is shown because it is the fundamental period estimation algorithm used in the EPI signal processing hearing aid that the MLP-Tx algorithm will replace. It can be seen that extra period markers are inserted in those regions where the speech signal exhibits pronounced secondary peaks in a period.

Trace C in figure 8.12 shows the frequency contour generated from the output of the MLP-Tx algorithm operating on the 20dB SNR speech shown in trace A. For the purpose of comparison, the frequency contour for the reference laryngograph algorithm and the peak-picker algorithm are shown in traces B and D respectively. Both the MLP-Tx and peak-picker frequency contours correspond to the reference quite well. The effects of the reduced sampling rate of 2kHz can be seen in the coarser quantization of the MLP-Tx contour than those due to the reference or the peak-picker.

Figure 8.13 shows the output from the MLP-Tx algorithm operating on the 0dB SNR speech signal. The first window shows the noisy speech pressure waveform. The second window is its 300 Hz bandwidth spectrogram. The third window shows the corresponding laryngograph waveform. The MLP-Tx output is given in the lowest window. It can be seen that performance is relatively unaffected by the additive canteen noise.

In figure 8.14 a section of figure 8.13 is shown with the time scale expanded to give a better close-up view of the output from the MLP-Tx algorithm.

Trace C in figure 8.15 shows the frequency contour generated from the output of the MLP-Tx algorithm operating on the 0dB SNR speech shown in trace A. The frequency contour for the reference laryngograph algorithm and the peak-picker algorithm are shown in traces B and D respectively. The MLP-Tx frequency contour again follows the reference quite well, whereas the peak-picker shows poor performance with the noisy speech, not only in the period marker estimates, but also in the voiced/unvoiced discrimination.

8.3.4 Quantitative evaluation of results

Two quantitative comparisons were carried out on the output fundamental periods generated by the MLP-Tx algorithms and the peak-picker. The methods used for these comparisons were fully described in chapter 6. They were also reported by Howard & Howard (1986).

The first comparison involved calculating the receiver operating characteristic, or ROC (Levine & Schefner, 1981), for both the MLP-Tx algorithm and for the peak-picker. This is a plot of the number of false alarms (incorrectly generated period markers) against the number of hits (correctly generated period markers) generated by a given detector as its detection criterion is swept between lax (that is hits are never missed, but false alarms are detected) and harsh (that is, false alarms are never generated, but some hits are not detected). In the case of the MLP-Tx algorithm this corresponds to changing the pulse detection threshold between 0 and 1. Figure 8.16 shows the ROCs for the MLP-Tx algorithm (marker A) and for the peak-picker (marked B) respectively, both operating on the 20dB SNR speech. The higher curve for the MLP-Tx algorithm indicates that it is a better detector of the period marker than the peak-picker (these ROCs were only calculated for a small portion of the test data, because of a limitation in the earlier analysis programs. However, the form of the ROCs obtained for different portions of the data all showed the same trends as indicated in figure 8.16).

The second comparison involves calculating the "jitter" in the placement of the period markers by the test algorithm relative to those generated by the reference laryngograph algorithm. The results are presented as histograms of the jitter for all the periods that have corresponding test and reference markers. Figure 8.17 shows the jitter histograms for all the Rainbow test data at both 20dB and 0dB SNRs. Plots A) and C) show the results for the peak picker and MLP-Tx algorithm respectively, operating on the 20dB SNR speech. Similarly, plots B) and D) show the results in the case of 0dB SNR speech. It can be seen that the distribution due to the MLP-Tx algorithm is narrower than for the peak-picker, and it is no wider in the 0dB SNR case than in the 20dB SNR case. The distribution due to the peak-picker is wider than that for the MLP-Tx

algorithm in both cases, and shows degradation between the 20dB SNR and the 0dB SNR conditions.

The results from the ROC indicate that on the test database used, the MLP-Tx algorithm performs firstly as a better detector (one average) of the excitation marker events, although this measure does not take their exact location into account. The jitter histograms indicate that it was a more accurate (on average) detector of the excitation points in the speech than the peak-picker. Notice that this result was achieved even though there was a limit on the time resolution from the MLP-Tx algorithm of 0.5ms, whereas the peak-picker operated to a 0.1ms resolution determined by the 10kHz sampling rate of the speech waveform.

8.2.5 Conclusions on preliminary results

The results obtained were found to be encouraging. Although these were only preliminary experiments, they indicated that useful results could be obtained using this new approach to speech fundamental period estimation. It was after this initial set of results had been collected that the MLP-Tx algorithm was considered to be of potential value in a fundamental frequency extracting hearing aid. However more experiments were needed in order to prove the generality of the approach, because there were several limitations to the preliminary experiments. Some of the main limitations were as follows:

1] The training and testing data was only composed from speech from men. It was clearly necessary to include speech from women in any future experiments.

2] The same speakers were used for testing and for training. To avoid any possible advantage to the algorithm by enabling it to adapt to the training speaker, different training and testing speakers should be used in future experiments.

3] The filterbank used was a first attempt at a pre-processing system, and this stage certainly needed greater analysis and investigation.

4] The structure used for the MLP network was found by trial and error on a limited amount of data and different MLP structures needed to be quantitatively investigated.

5] The output frame duration of 0.5ms precludes the use of the algorithm in many applications, simply because the quantization error associated with this frame duration was too large. It must be noted, however, that this degree of temporal resolution was **suitable** for use in signal processing hearing aids, because the users of such aids typically have reduced frequency discriminative abilities (see chapters 1 and 3) and a severely reduced frequency range (typically only up to 1kHz or less).

All of these limitations as well as other issues are discussed more fully in chapter 9. In addition, the MLP-Tx algorithm needed to be compared against other established algorithms. Chapter 6 provided a discussion of techniques for making quantitative comparisons among fundamental frequency and fundamental period estimation algorithms. Some of these techniques are used to evaluate the different configurations of the MLP-Tx algorithm which arose from this present phase of the work, and other algorithms (chapter 9).

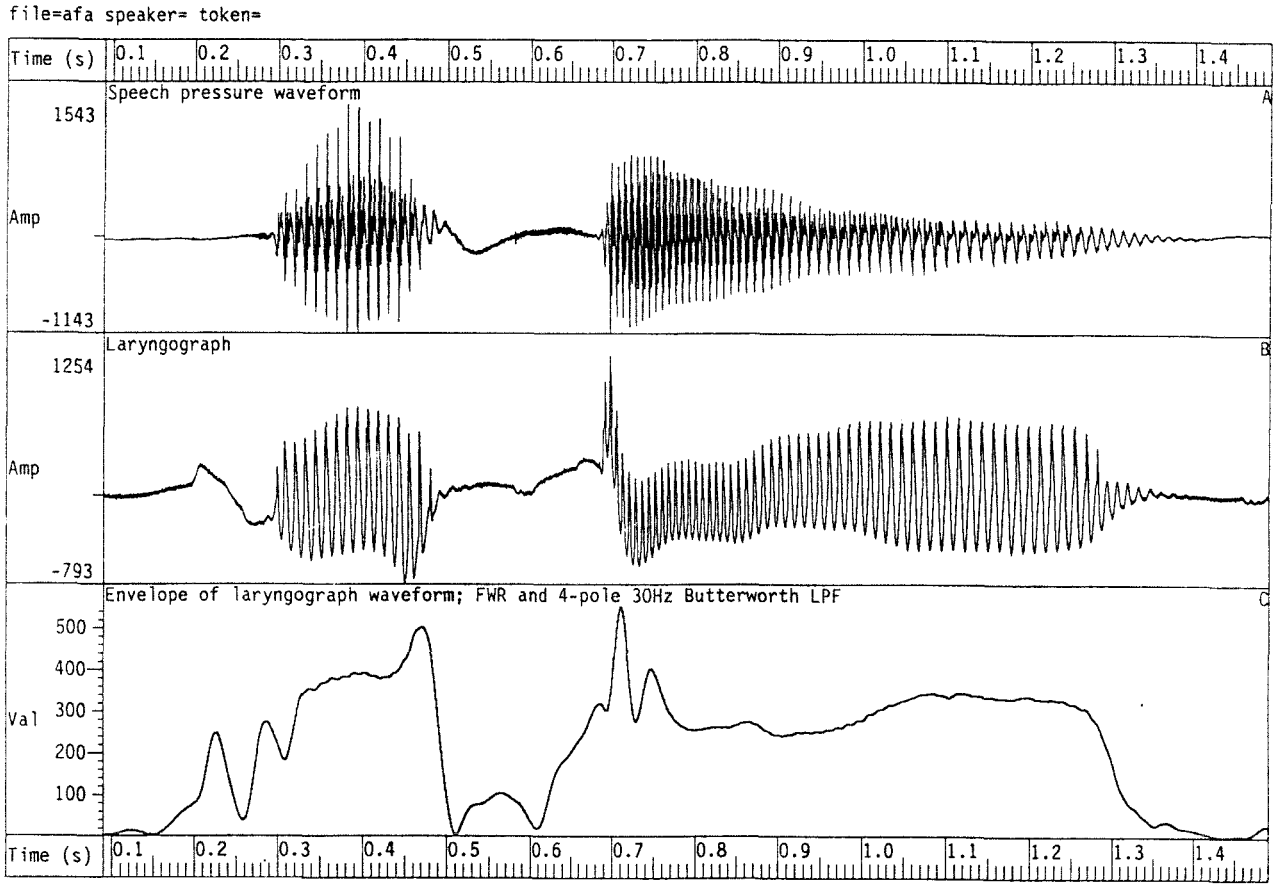


Figure 8.1 Voicing determination by envelope detection of the laryngograph waveform. It can be seen that the onset and offset of voicing can become smeared using this approach. This is particularly evident between the two voiced regions, where the envelope detector output slowly drops down and rises again. The speech is the utterance /afa/ from a male subject.

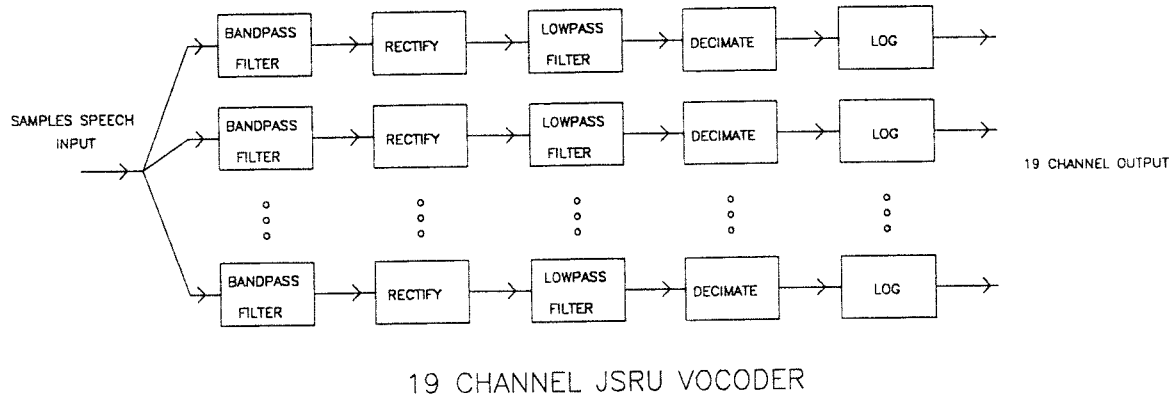


Figure 8.2 Schematic diagram for the JSRU 19-channel vocoder.

A modification of this system was used as the pre-processing filterbank in the early configurations of the MLP-Tx algorithm.

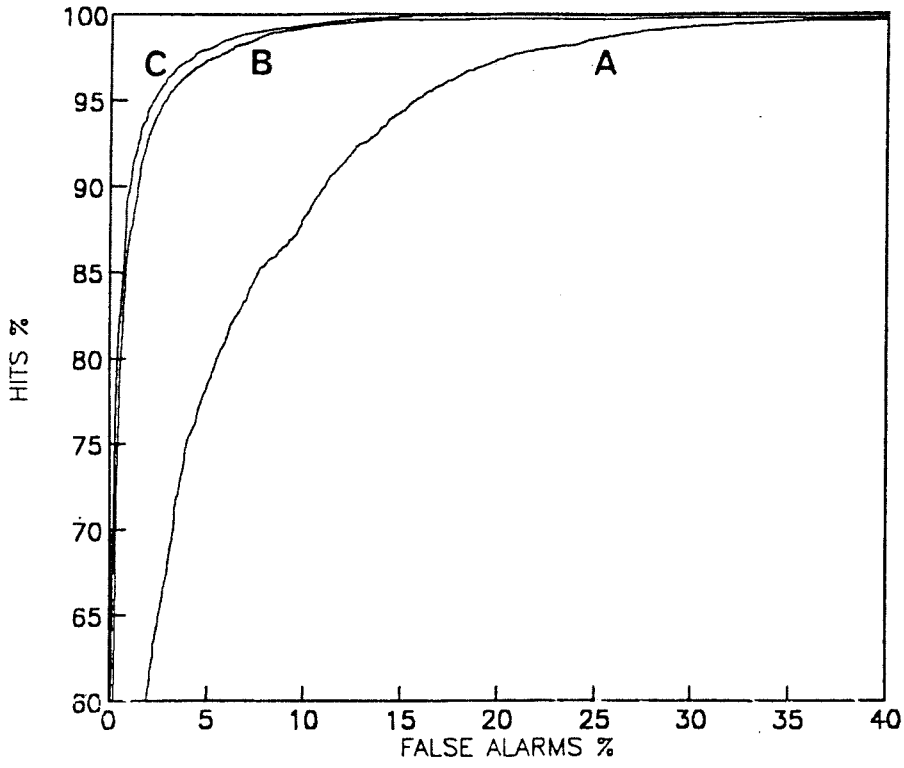


Figure 8.3 Receiver operating characteristics for voicing estimation algorithms operating on anechoic speech.

Curve A is the ROC for a Bayes' classifier for Gaussian patterns with one vocoder frame in the input vector. Curve B is the ROC for a MLP with no hidden units and one vocoder frame in the input vector. Curve C is the ROC for the MLP using three adjacent vocoder frames in the input vector, both with and without hidden units. In the last case, hidden units did not improve the performance.

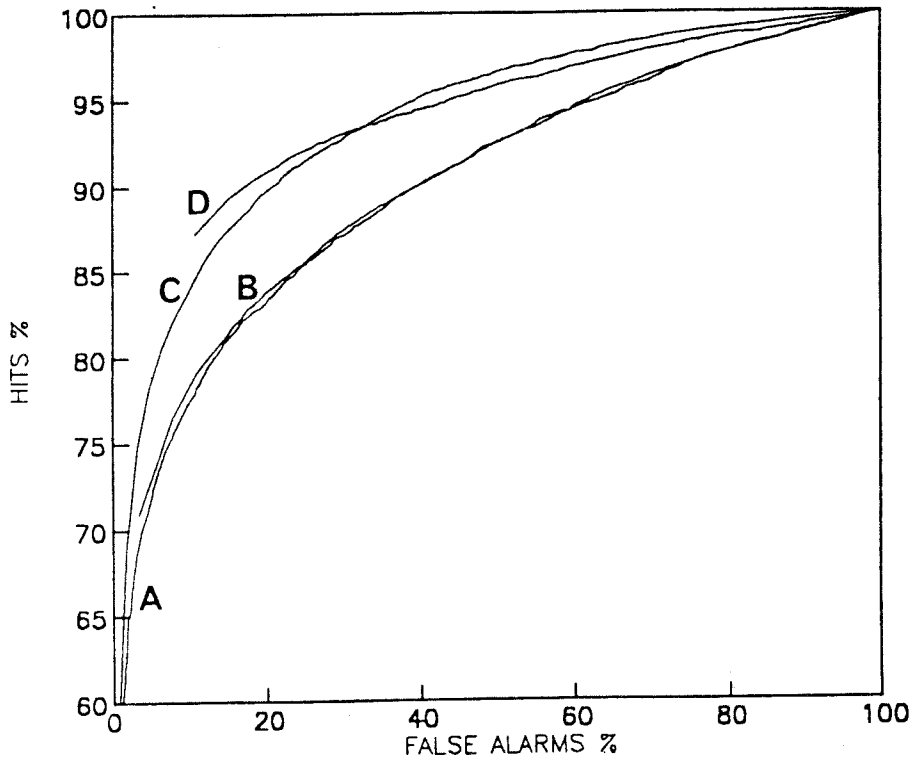


Figure 8.4 Receiver operating characteristics for voicing estimation algorithms operating on speech with additive white noise at 0dB SNR.

Curve A is the ROC for a Bayes' classifier for Gaussian patterns with one vocoder frame in the input vector. Curve B is the ROC for a MLP with one layer of hidden units and one vocoder frame in the input vector. Curve C is the ROC for a MLP in which the input vectors employed a frame of smoothed vocoder input over a 100ms averaging window in addition to a normal vocoder frame. Curve D is the ROC for the MLP using adjacent input frames, and using hidden units. It can be seen that the MLP with adjacent vocoder frames and hidden units gave the best overall performance.

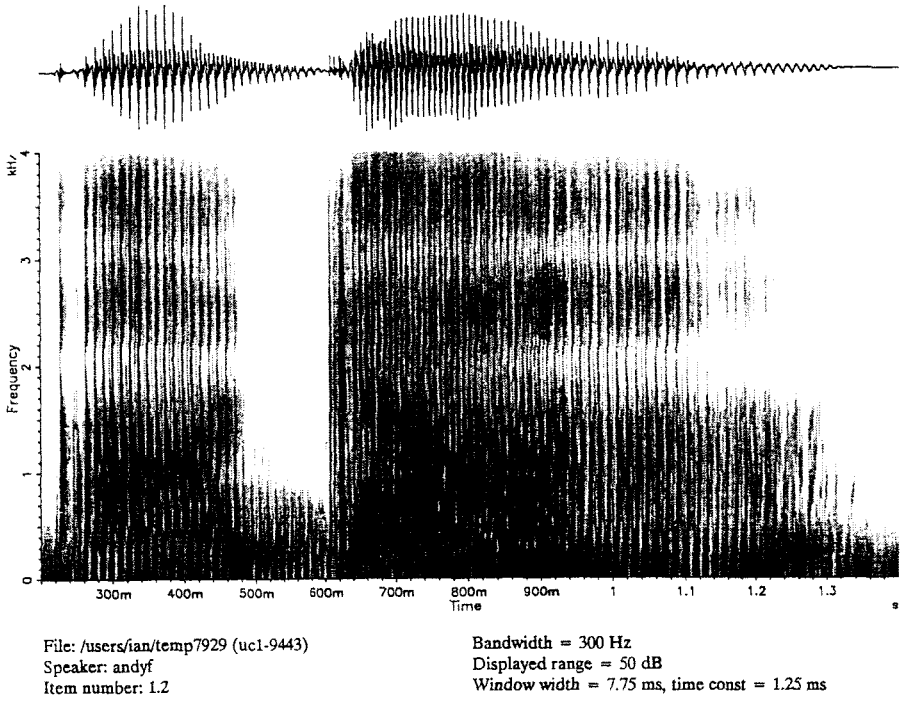


Figure 8.5 Wideband spectrogram (300Hz bandwidth) for a short section of male speech.

This illustrates the vertical striations that correspond to the periodic acoustic excitations resulting from the snapping together of the vocal folds. The utterance is /aga/ from a male subject.

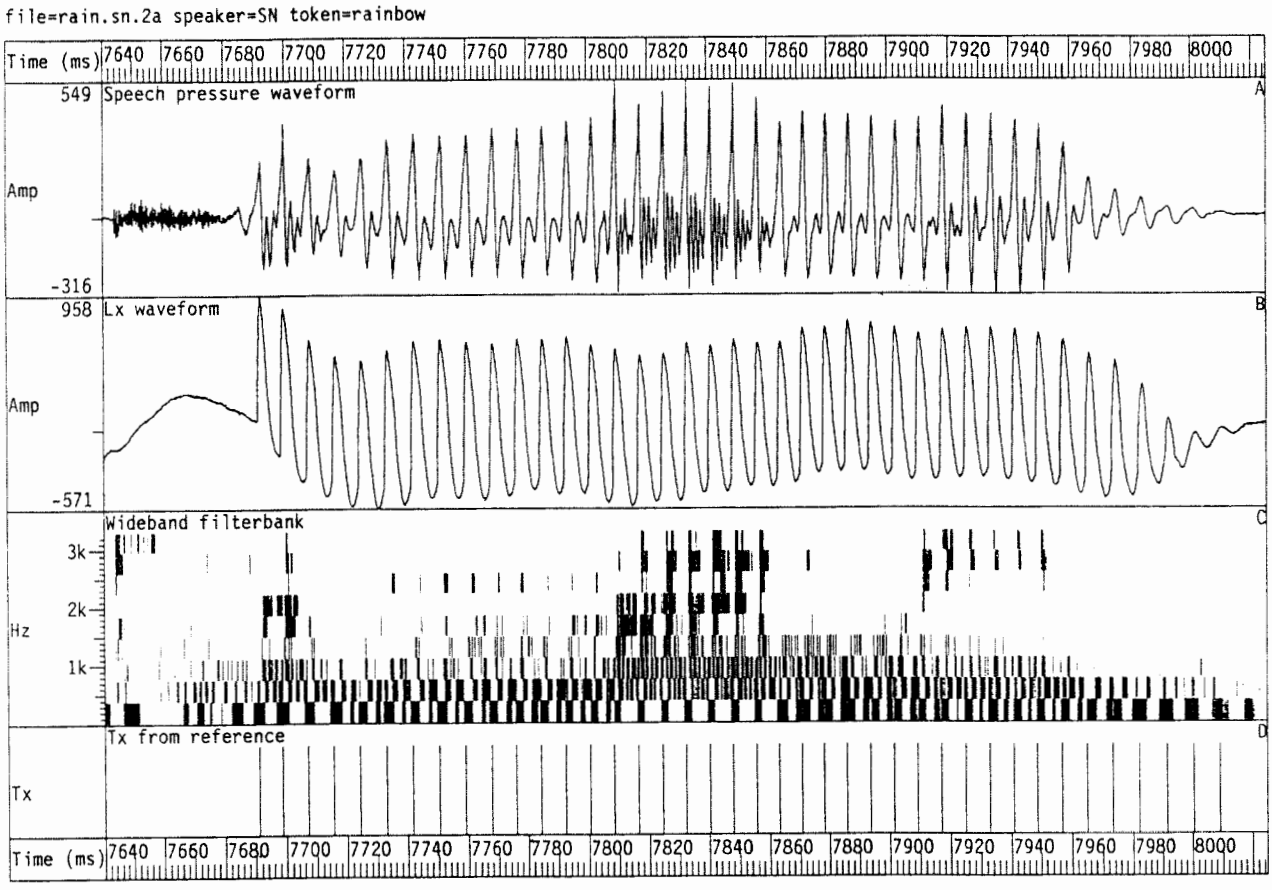


Figure 8.6 Input and output signals for initial wideband filterbank MLP-Tx algorithm. Plot showing speech pressure waveform (trace A) and corresponding laryngograph signal (trace B). The output from the 9-channel wideband filterbank is shown in trace C. The fundamental period estimates obtained from the laryngograph are shown in trace D. The utterance is "..too many.." from a male subject.

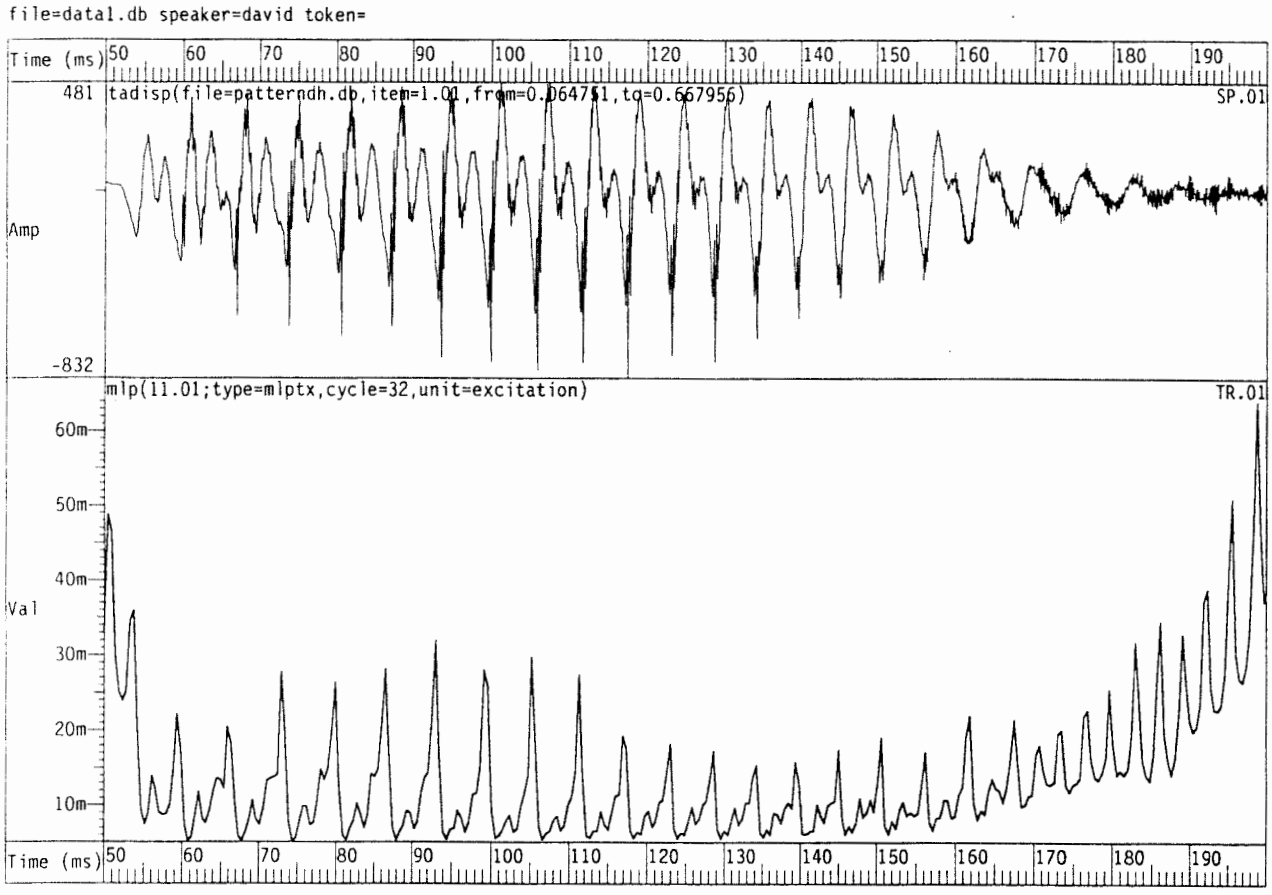


Figure 8.7 Very first output generated by the MLP-Tx algorithm.

The top trace shows the input speech pressure waveform and the lower traces shows the MLP-Tx output.

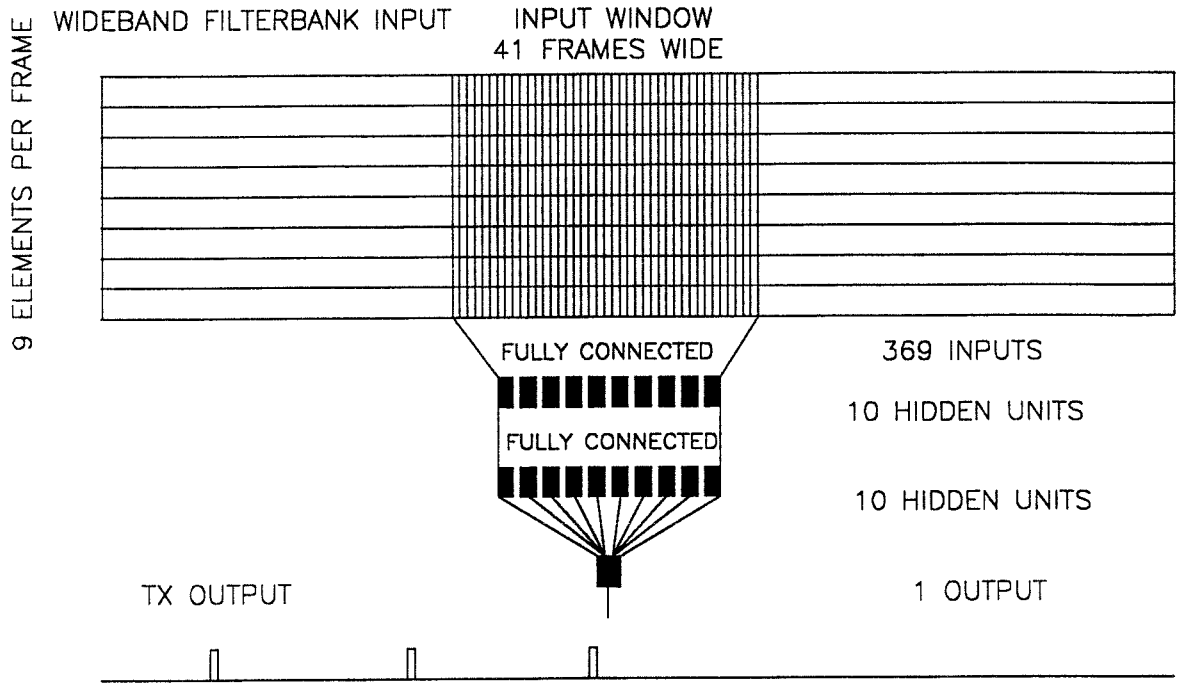


Figure 8.8 Schematic diagram of the MLP-Tx algorithm used for the first experiments employing a moderately sized database.

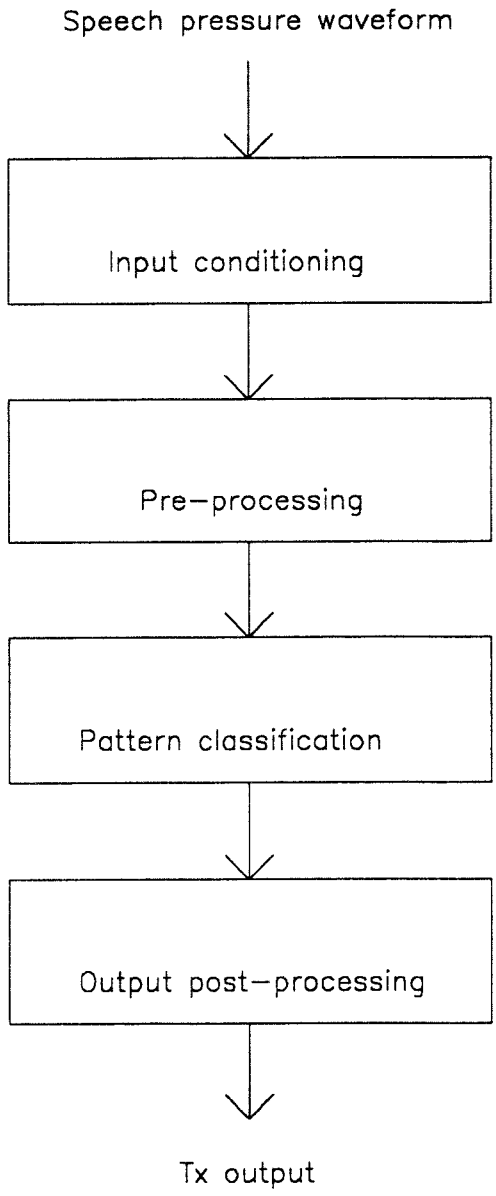


Figure 8.9 Overall system schematic diagram for the various stages in the MLP-Tx fundamental period estimation system.

The first stage consists of input condition, which encompasses the microphone, anti-aliasing filtering and digitization of the input speech pressure waveform. The next stage pre-processes the raw input signal, to give an input suitable to be used as the input to a pattern classifier. Next the pattern classification transformation is applied, which results in a raw estimate of the vocal fold closures. Finally, the output is cleaned up, and converted into the overall output format by a post-processing stage.

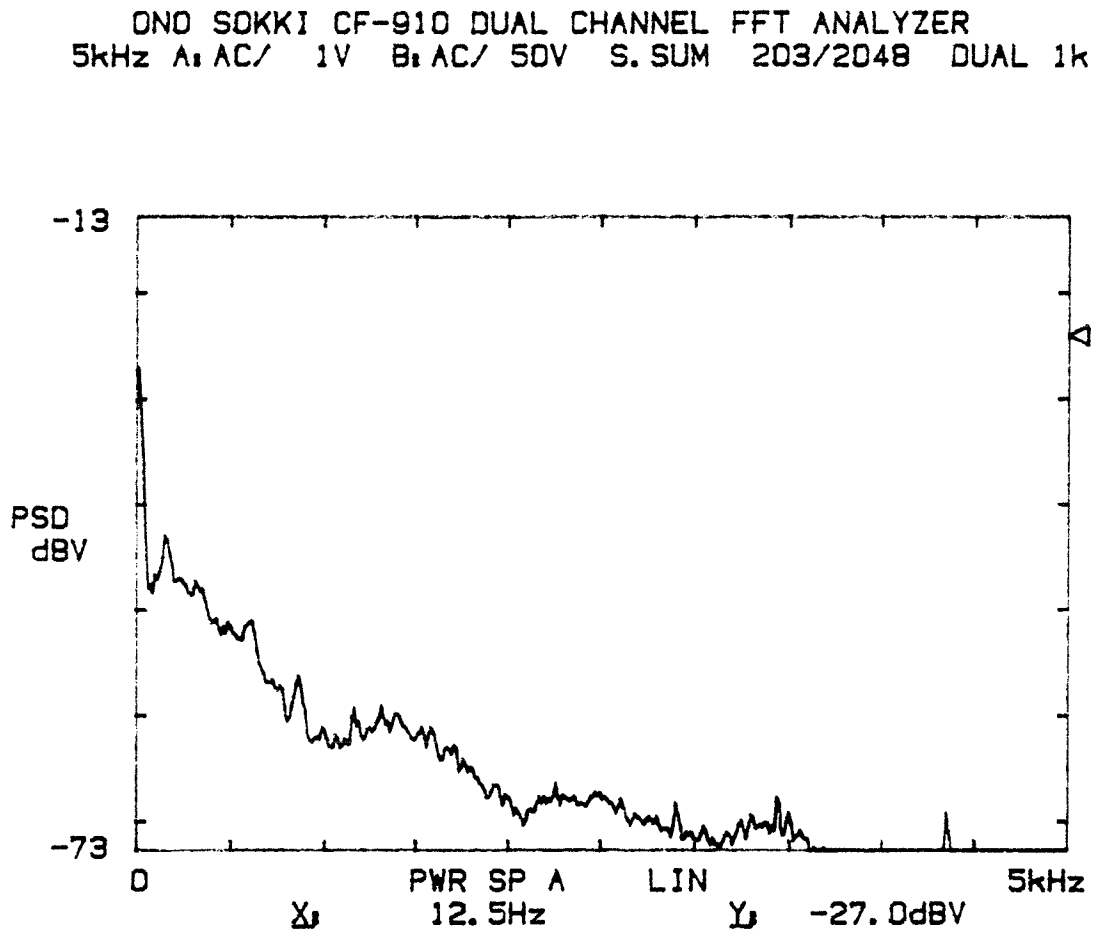


Figure 8.10 Power spectrum for the canteen noise.

The input was filtered at filtered at 5kHz. An FFT window size of 80ms was used and 203 frames were averaged to generate the spectrum.

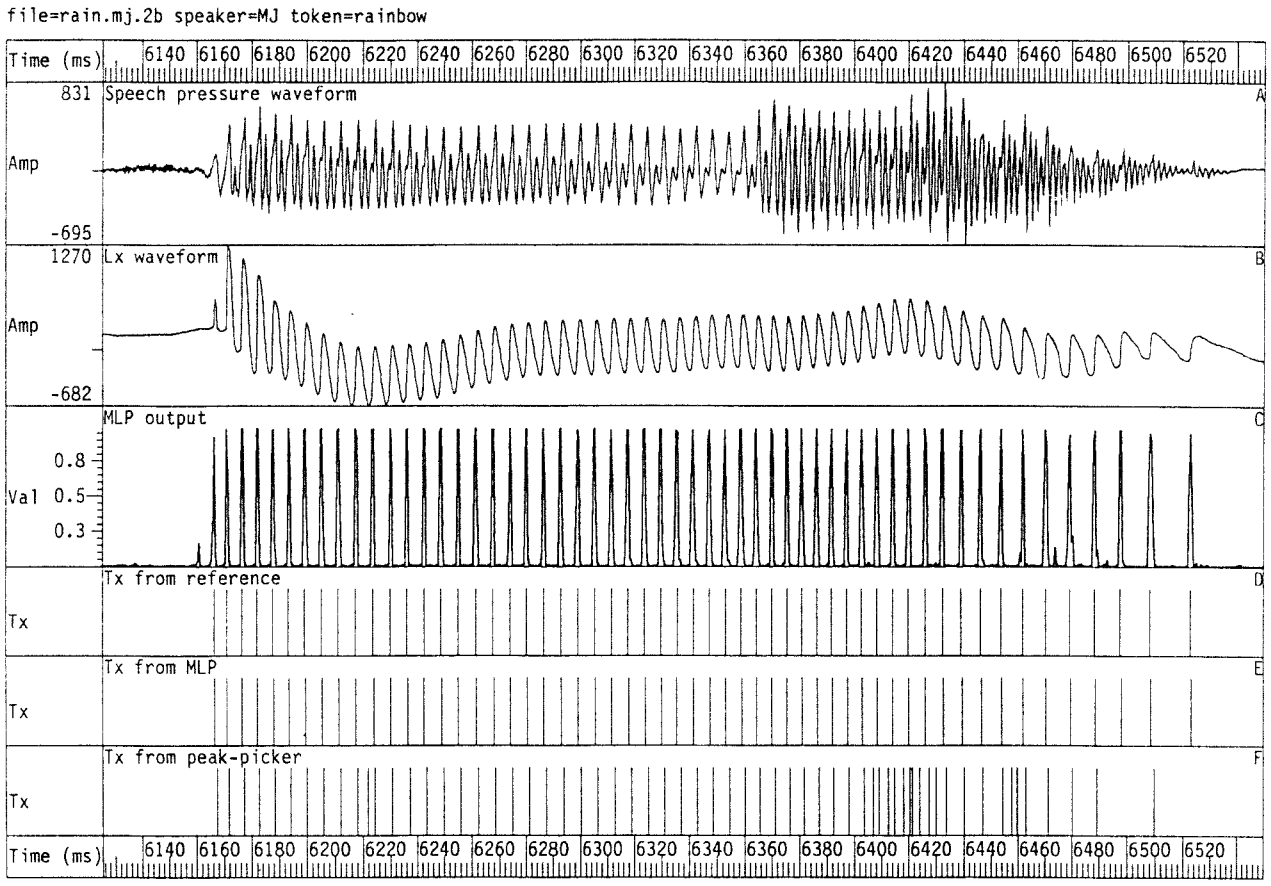


Figure 8.11 Plot showing the output from the wideband MLP-Tx algorithm on 20dB SNR speech.

Trace A shows the 20dB SNR speech pressure waveform and its corresponding laryngograph waveform is shown in trace B. Trace C shows the MLP output. The period markers from the reference, MLP-Tx algorithm and peak-picker are shown in traces D, E and F respectively. It can be seen that there is good agreement in the location of the output from the MLP-Tx algorithm and the reference period markers, with no false full-size pulses being generated or missing pulses. The utterance shown is "people look" from a male subject.

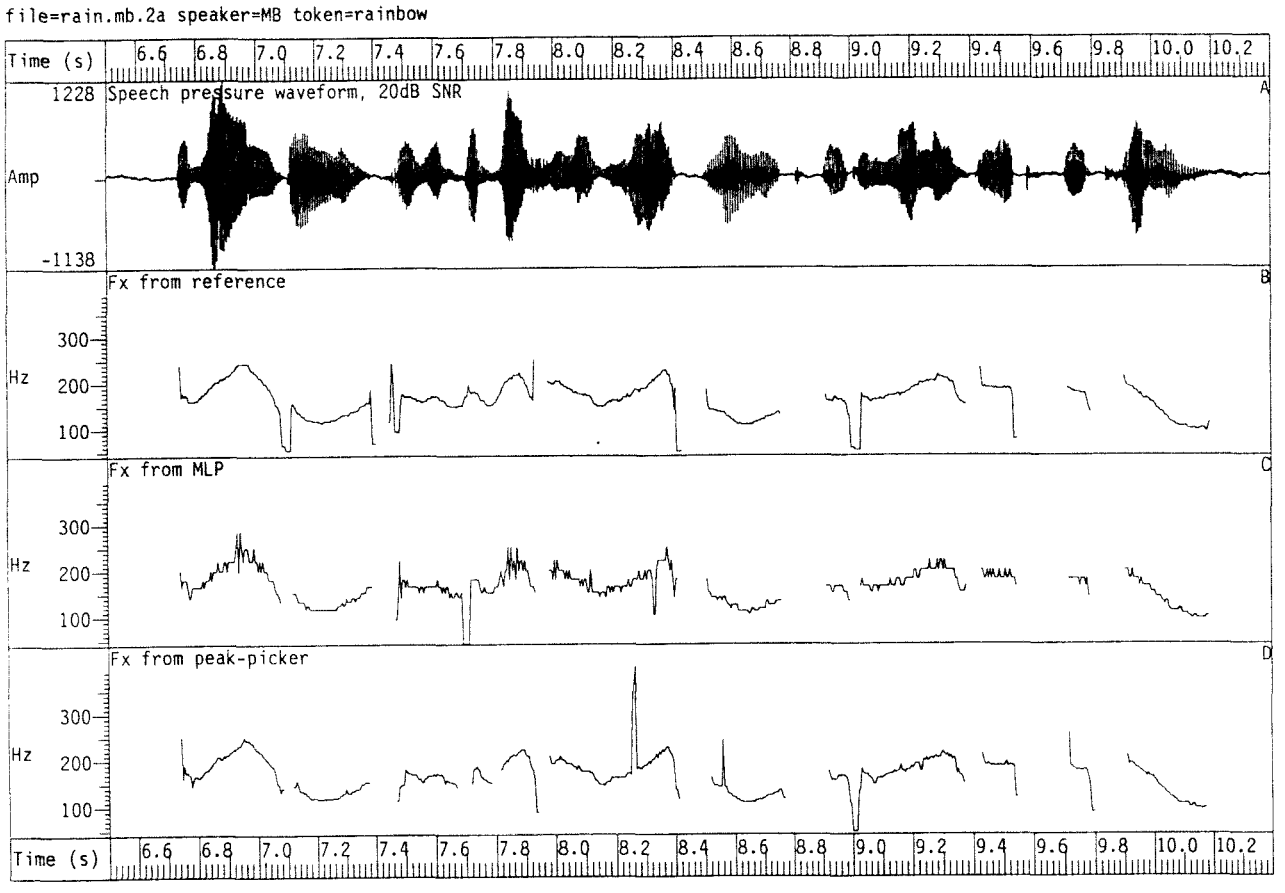


Figure 8.12 Plot showing the frequency contour from the MLP-Tx algorithm on 20dB SNR speech.

Trace A shows the speech pressure waveform at a 20dB SNR. The reference frequency contour is shown in trace B, the frequency contour from the MLP-Tx algorithm is shown in trace C and the frequency contour from the peak-picker is shown in trace D. The quantization error due to the 2kHz output frame rate is clearly visible in the output due to the MLP-Tx algorithm. The utterance shown is "The rainbow is a division of white light into many beautiful colours" from a male subject.

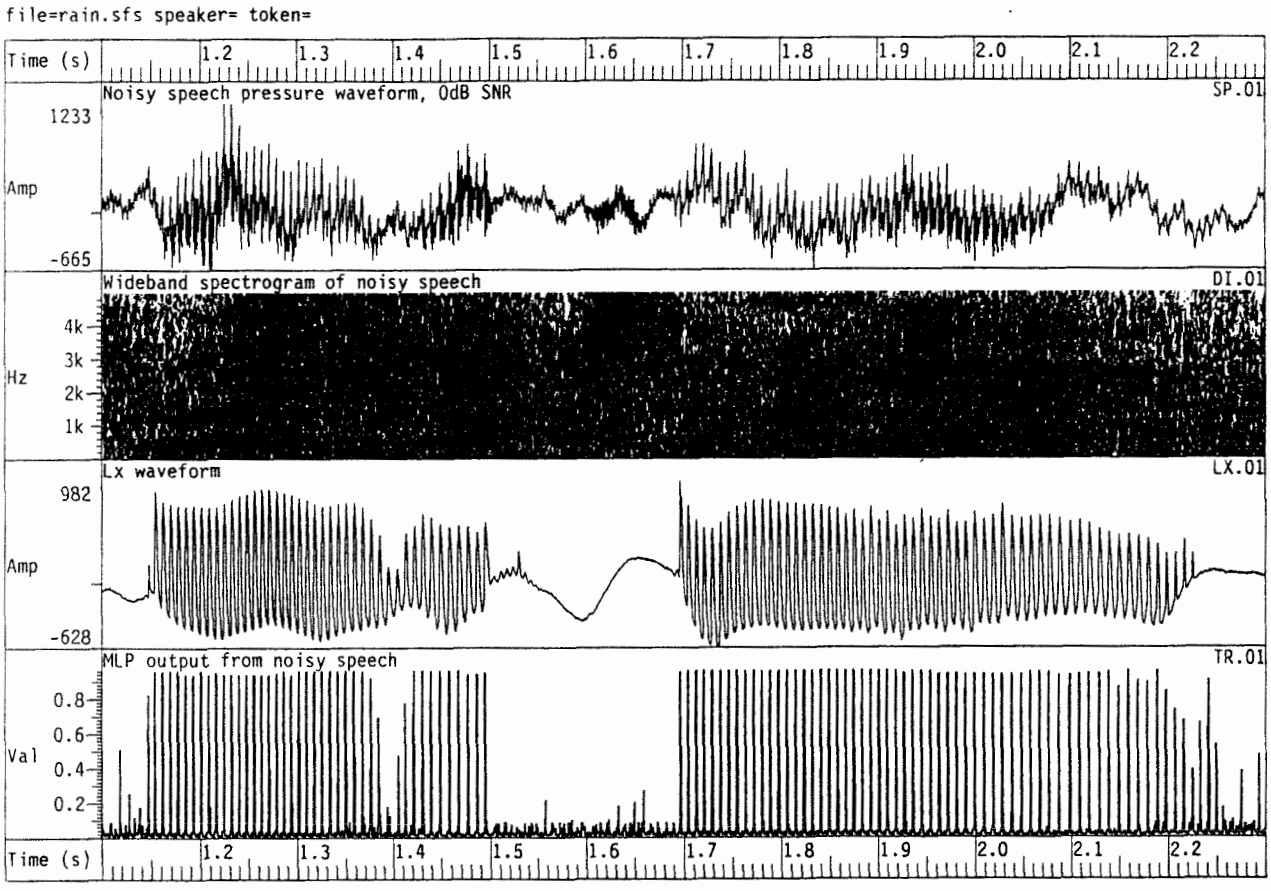


Figure 8.13 Output from the MLP-Tx algorithm operating on speech with added canteen noise at a 0dB SNR.

The MLP-Tx output is shown in the bottom trace. The second trace shows a wideband spectrogram of the input speech, with the laryngograph waveform shown below it. The utterance shown is from a male subject.

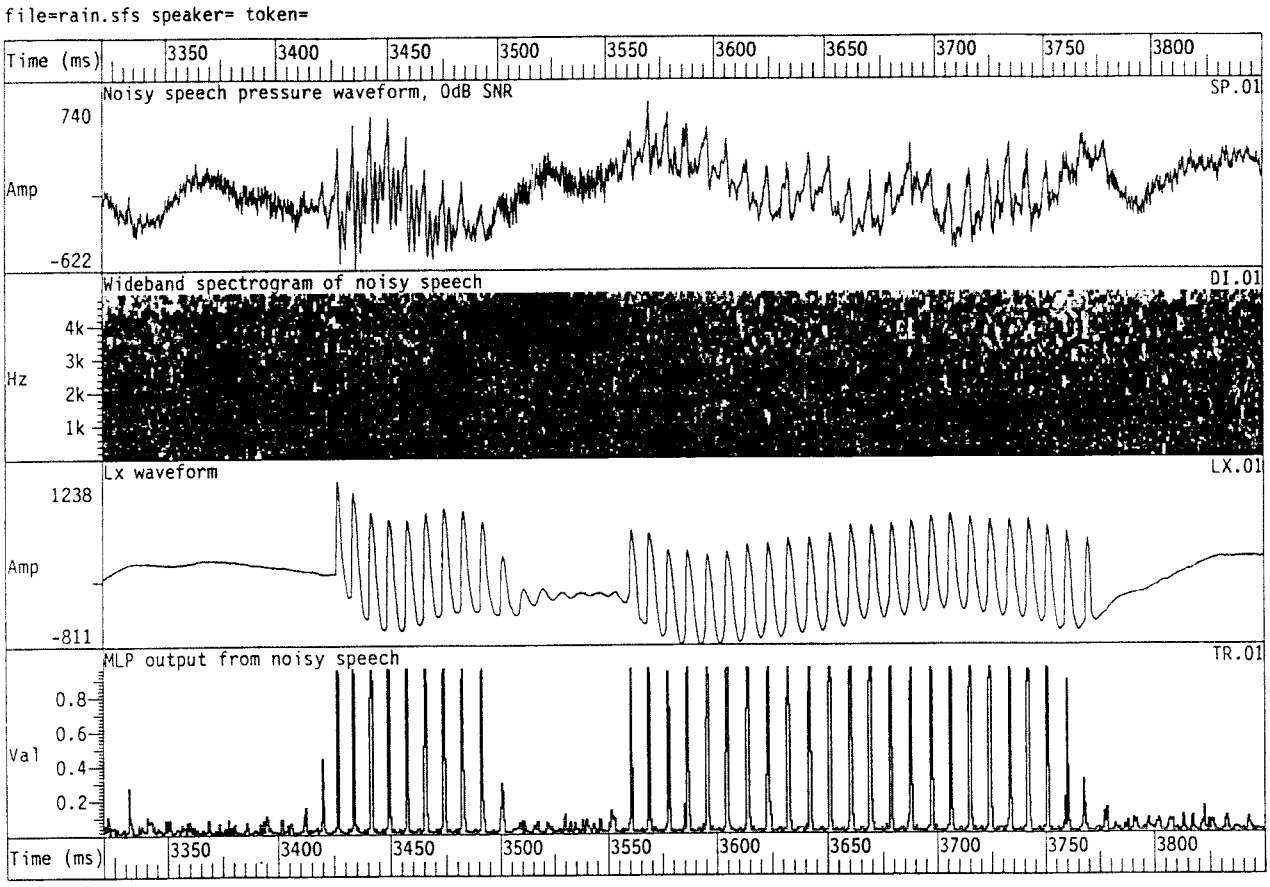


Figure 8.14 Same as in figure 8.13, but with an expanded time-scale.

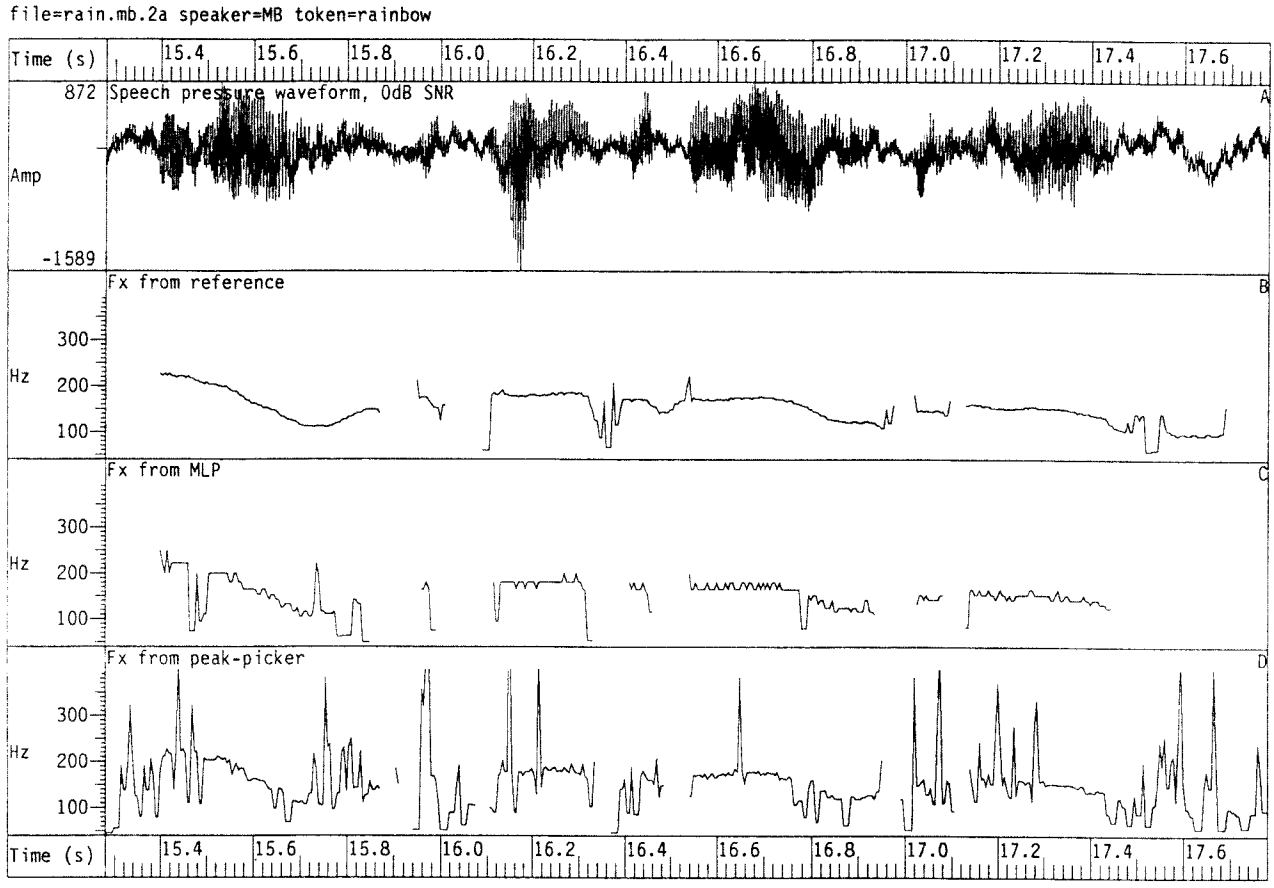


Figure 8.15 Frequency contour from the MLP-Tx algorithm operating in the presence of "canteen noise" at a 0dB SNR.

The noisy speech pressure waveform is shown in trace A. The reference frequency contour is shown in trace B, the MLP-Tx frequency contour is shown in trace C and the frequency contour from a peak-picker is shown in trace D. It can be seen that the performance of the peak-picker is more affected by the noise than is the MLP-Tx algorithm. The utterance shown is "..two ends apparently beyond the horizon" from a male subject.

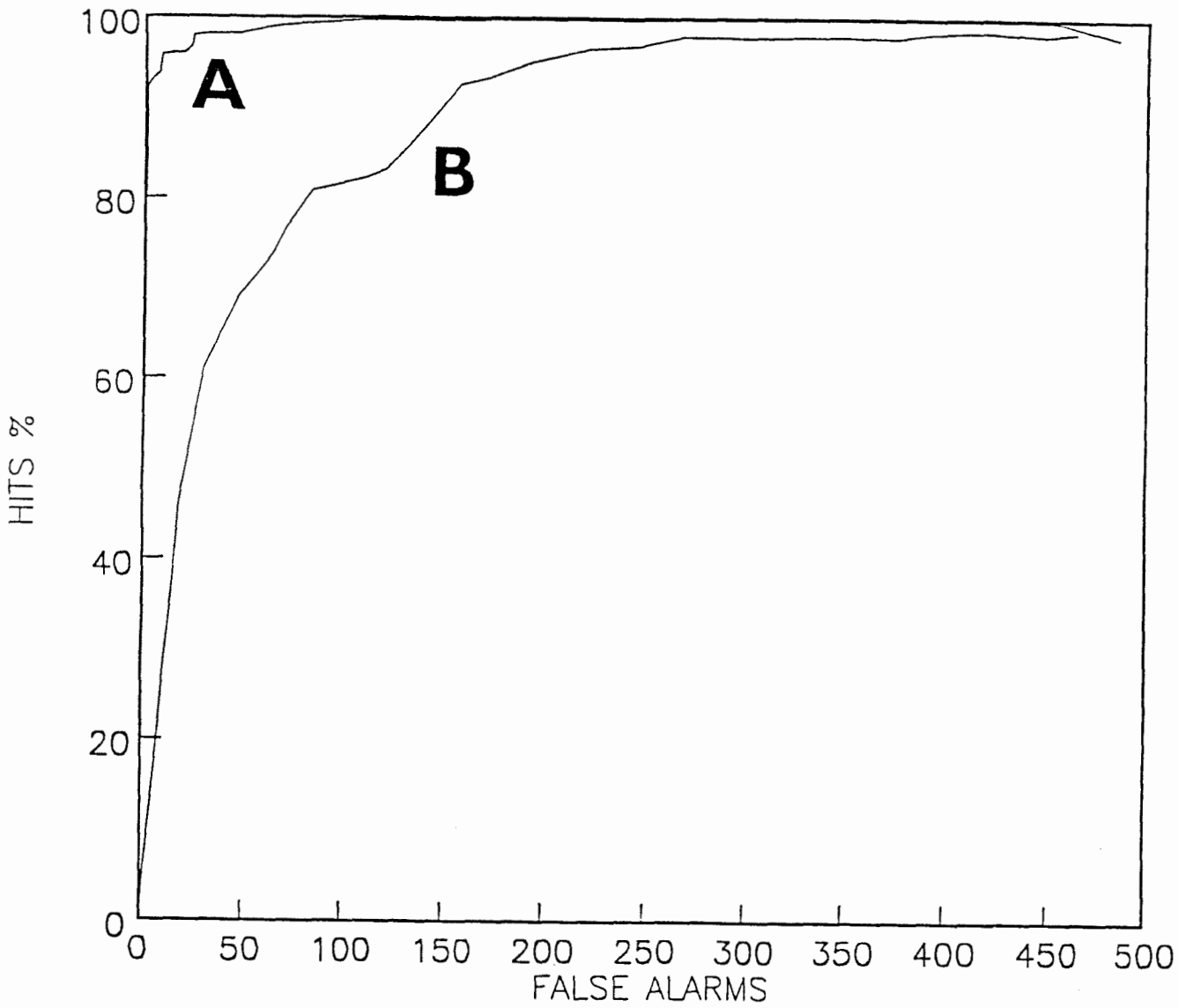
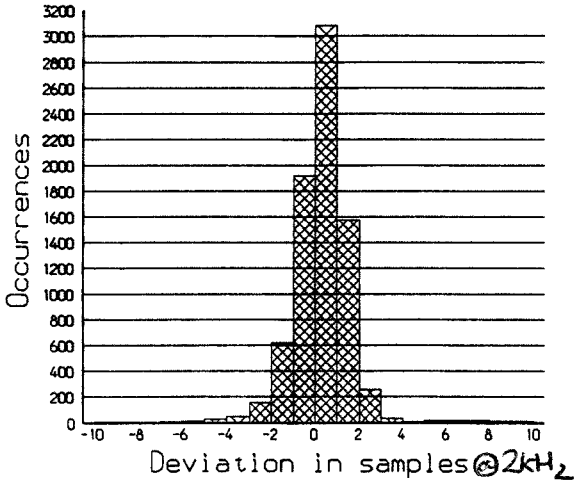
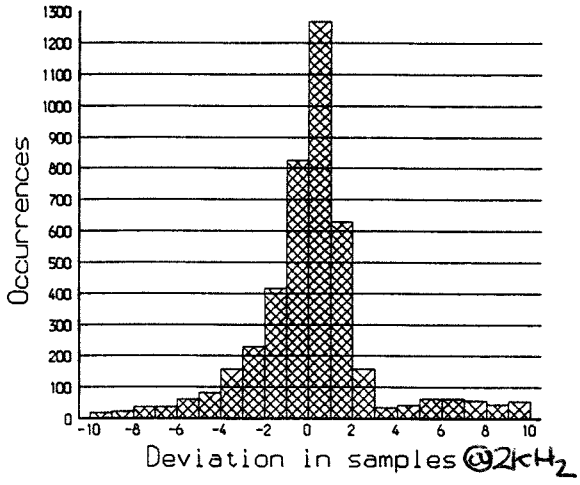


Figure 8.16 Receiver operating characteristic for the MLP-Tx algorithm and the peak-picker.

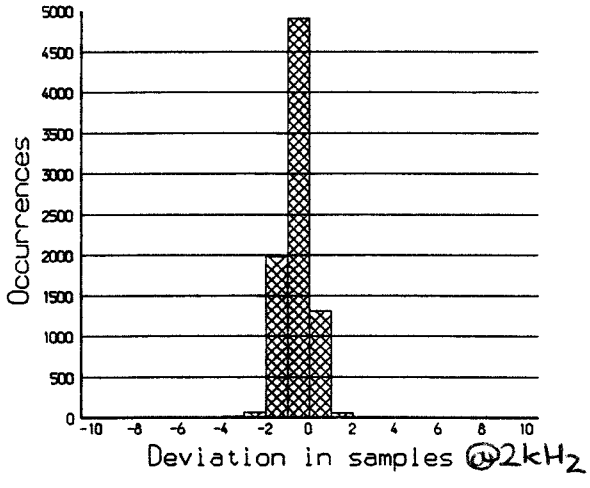
The ROC for the MLP-Tx algorithm is curve A, and that for the peak-picker is curve B, both operating on speech at a 20dB SNR. These curves indicate that the MLP-Tx algorithm performs as a better detector on the test data than does the peak-picker.



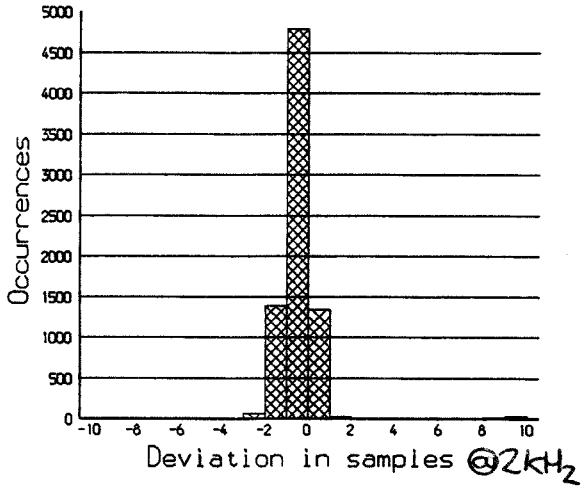
A) Peak-picker, 20dB SNR.



B) Peak-picker, 0dB SNR.



C) MLP-TX, 20dB SNR.



D) MLP-TX, 0dB SNR.

Figure 8.17 Jitter histograms for the MLP-Tx algorithm and the peak-picker. Ideal performance would be represented by a single bar at zero deviation. Graphs A and B are for the peak-picker, and graphs C and D are for the MLP-Tx algorithm, in 20dB and 0dB SNR conditions respectively. Notice that the MLP-Tx algorithm is less affected by noise than the peak-picker.

CHAPTER 9: MORE DETAILED DISCUSSION OF ISSUES IN SPEECH FUNDAMENTAL PERIOD ESTIMATION USING PATTERN CLASSIFICATION

9.1 LIMITATIONS OF PRELIMINARY EXPERIMENT

9.1.1 Preliminary experiment

The initial results showed that the MLP-Tx system performed creditably on the noisy speech used in the experiment and better than the single alternative time-domain system. However more experiments were needed in order to prove the generality of the approach, because there were several limitations to the preliminary experiments. Some of the main limitations were as follows.

9.1.2 Limitations in the testing data

In the preliminary experiment, the training and testing data were only composed of adult male speech. It was clearly necessary to test the algorithm on speech from women. In addition, the same speakers were used for testing and for training. To avoid any possible advantage because the MLP-Tx algorithm adapted to the training speakers, different training and testing speakers should be used in future experiments. The speech in the original experiments was recorded anechoically and was therefore free from the effects that reverberation introduces. In addition, the speech was recorded at a constant fixed distance of 15cm from the speaker's lips. In real use (for example, in the EPI hearing aid) the MLP-Tx algorithm would have to function in reverberant conditions at a range of distances from the speaker.

9.1.3 Lack of optimization of MLP-Tx parameters

The MLP-Tx algorithm had not been rigorously investigated especially with respect to the pre-processing employed. The filterbank used was a first attempt at a pre-processing system, and needed greater analysis and investigation.

9.1.4 Limited output period marker time resolution

The output frame duration of 0.5ms precluded the use of the algorithm in many applications, simply because the quantization error associated with this frame duration is too large. It must be noted that this **was suitable** for use in signal processing hearing aids, because such patients have poor frequency difference limens.

9.1.5 Organization of this chapter

This section first describes the issues concerning the requirements of a new database. This involves the specification of the speech material and its recording conditions, as well as the important issue of the alignment of the speech and the laryngograph signals. Next the pre-processing stage in the MLP-Tx algorithm is considered in more detail and several schemes are described. The training of the MLP is then considered, and two schemes to reduce training times are used. Finally quantitative frequency contour comparisons, described in chapter 6, are used to compare different configurations of the MLP-Tx algorithm against two established techniques.

9.2 DATABASE CONSIDERATIONS

9.2.1 Required range of speech and speakers.

In order to both train and then rigorously evaluate the MLP-Tx fundamental period estimation algorithm, it was necessary to have a set of speech data that were representative of the kind of data that will be typical in the real use of such an algorithm. Consequently a wide range of samples were required from both men and women.

9.2.2 Choice of reading passages

It is important to take into account the amount of time and effort involved in obtaining the database, since there was not unlimited time or resources available for this task. By

using reading passages, the recording sessions could be kept relatively short and undemanding on the speakers. Continuous interactive discourse would have been a more realistic situation, but with this approach it would be more difficult to get a good coverage of the phonetic range of the speaker. To achieve a good coverage of different speech sounds, speakers were asked to read two phonetically balanced passages, the Rainbow passage (Mermelstein, 1977) and Arthur the Rat (Abercomie, 1964). The passages were divided into paragraphs that lasted about 15 seconds each. This was done so that there would be natural break-points in the recordings that would aid loading them into separate data files on the computer system.

9.2.3 Selection of recording environments

In addition to having a database that was representative of variations in speech between different speakers, it was also important to represent a wide range of environmental conditions and for the recordings to contain natural reverberation and background noise. To achieve this aim, the speech was recorded in five typical rooms, which were chosen to provide a good range of reverberant conditions. There was also to be some background noise present in these recordings. The distance between the speaker and microphone was also varied between about 30cm and 200cm, since these reflect the typical range of operation for the SiVo hearing aid during use.

9.2.4 Time delay between speech and laryngograph signals

The reason for recording the laryngograph signal is so that the speech pressure waveform can be labelled with excitation period markers, for the purpose of training and testing the MLP-Tx algorithm.

With regard to the training data for MLP-Tx algorithm, it is essential that these markers always correspond to the same point in a speech cycle, because this algorithm explicitly detects these points. That is to say, it is important that there is a constant delay (which is ideally zero, but any small offset, less than about 1ms, is acceptable) between the excitation period marker and the excitation point in each speech cycle. If this was not

the case, the training data would not uniquely define the relationship between the excitation point in the speech pressure waveform and the peak differential in each laryngograph cycle. Consequently, the training data would have been contradictory and the MLP-Tx algorithm would not train properly.

For the testing data, a less precise alignment can often be tolerated, because it is the distance between excitations that is generally important, rather than their absolute locations. This is particularly true if the output from the MLP-Tx algorithm is converted to a fixed rate sampled frequency contour, since this loses all the information concerning the exact period marker locations.

Because the laryngograph records the activity of the vocal folds by direct electrical measurement and the speech signal was recorded after it had propagated though the air over a distance between 30cm and 200cm, there was always a significant time delay between the speech signal recorded at the microphone and the corresponding laryngograph signal.

It was always necessary to compensate for this delay, because it changed with recording distance, so that the delay corresponding to different speakers, each of which was typically recorded at a different distance, was different.

9.2.5 Effect of head movements

For a sampling rate of 8kHz, taking the speed of sound as $340ms^{-1}$, a time delay between the laryngograph signal and the speech signal equal to a single sample period corresponds to a distance of:

340/8000 m = 4.25cm.

It is quite possible that the speaker could move by a distance of 4.25cm away or towards the recording microphone in the course of reading the passages. Under these circumstances, there would be a detectable delay that was a function of time. Although

it is possible to use dynamic time warping to compensate for cases in which the time-delay is a function of time (as described in chapter 6), it is very much easier to compensate for the delay (and the result are more reliable) if it constant.

9.2.6 Original experiment

In the preliminary MLP-Tx experiment, using a frame rate of 2kHz, one frame corresponded to a distance of

340/2000 = 17cm.

It is clear that the movement of the head is less critical at this frame rate. Indeed, problems relating to the alignment of the speech never arose in the preliminary experiments because the recordings were carried out with the speaker in a chair with a head rest and using a fixed 15cm microphone distance from the speaker's lips. In this case, the 17cm was also sufficient to take account of the differences in the lengths of the different speaker's vocal tracts.

9.2.7 Recording speech and laryngograph data with a fixed time delay between the two signals

To insure that there was a constant delay (or at least constant enough for changes to be insignificant compared to a sampling period at the 8kHz rate) between the speech pressure waveform and the laryngograph signal for the training data, the speech and laryngograph signals were recorded with a microphone that was attached to a fishing rod that was fitted into a helmet that the speaker wore. In this way, head movements had no effect on the time delay between the speech and laryngograph signal, because the microphone was always a constant distance from the speaker.

9.2.8 Selection of number of speakers

Finding willing subjects for the purpose of the generation of a database was a difficult

and time consuming operation. Altogether around 80 speakers were recorded, which provided a large pool of testing and training data, and also permitted several poor recordings to be discarded (sometimes it was difficult to get good laryngograph signals, and this was only fully evident after the data had been acquired onto the computer system for a full analysis).

9.2.9 Training, preliminary testing and final testing data sets.

Three separate data sets were needed to both train and test the MLP-Tx algorithm. Firstly, it was necessary to use different speakers for the training and testing of the MLP-Tx algorithm. Secondly, because the algorithm was optimised in the course of its development by evaluation on test data, it was necessary to ensure that there was a final previously unseen set of testing data (no used for optimization) against which the optimised MLP-Tx algorithm could be compared against established techniques. This avoided any bias towards the MLP-Tx algorithm it may have had due to adaption of its performance to the preliminary testing data. A list of the speakers, their ages and the recording conditions appear in appendix A.7. Frequency distributions for the training and final test data appear in appendix A.8.

9.2.10 Training data

The training data set was composed of 4 men and 4 women speakers, each of whom read the Rainbow Passage and the Arthur the Rat passage. The reason that only 8 speakers, with a large amount of data per speaker were used is because it was very important to be able to guarantee that the speech and laryngograph signals were accurately time aligned, and this could only be checked using the consistency of alignments between different files for the same speaker recorded at the same distance. If many separate speakers had been used, with only a small amount of data per speaker, it would not have been possible to check alignment in this way.

9.2.11 Preliminary testing data set

The preliminary evaluation testing data set consisted of 10 men and 10 women. For each speaker, both the Rainbow Passage and the Arthur the Rat passages were recorded, but only one paragraph was actually used. Treating the men and women as two separate groups, the paragraphs were selected such that the first paragraph were used for the first speaker, the second paragraph for the second speaker, etc, until all the speakers were accounted for. This achieved coverage of most of the material in the passages, without any paragraphs being repeated.

9.2.12 Final testing data set

The final testing data set consisted of 20 men and 20 women speakers. Only one paragraph per speaker was used, and these were rotated to achieve maximum coverage of the passages.

9.3 INPUT SIGNAL CONDITIONING AND RECORDING OF THE DATABASE

9.3.1 Recording the test databases

The microphone used for the testing recordings was a B&K 4134 condenser omni-directional pressure microphone (standard type) fitted in a B&K sound pressure meter. The microphone was mounted on a tripod and could be raised and lowered to between 100-200cm from the ground to give a range of different microphone heights. The output was calibrated to 94dBA at 1kHz using a B&K calibrator. The output from a laryngograph was also recorded on the other channel. Recordings were made using a Sony DAT recorder, with 16-bit resolution at a 48kHz sampling rate. The levels for the recordings were left fixed after initial setting up from the calibrator, thus giving an absolute calibration level.

Training data was recorded using a small high quality Knowles BL-1785 piezo-electric microphone at a constant distance from the speaker, for reasons previously discussed. The frequency response of the microphone was flat within ±1dB over the range 40Hz-1kHz, and to within ±3dB over the range of 1kHz to 8kHz.

The output signal from the given microphone was fed via a pre-amplifier into a 3rd order Bessel high-pass filter with a 50Hz cut-off frequency, to remove low-frequency noise. This was necessary because there was significant noise power present below 50Hz which would otherwise lead to problems with dynamic range at the A/Ds. To achieve minimum phase distortion, a Bessel filter was used instead of a Butterworth filter. The input conditioning is illustrated in figure 9.1.

9.3.2 Choice of sampling rate for digital acquisition

The data was acquired directly from a DAT recorder onto the MASSCOMP computer using a 12bit A/D converters operating at 8kHz in conjunction with 4-pole Butterworth low-pass anti-aliasing filtering at 3.5kHz. A sampling frequency of 8kHz constituted the lowest practical rate at which the intelligibility of the speech could be preserved. In addition it is about the lowest acceptable time resolution of the period markers that are of general use. More importantly, it is also the sampling frequency adopted by telephone companies. As a consequence of this, A/D and D/A converters that operate at this frequency are easily available and significantly cheaper than those that operate at (for example) 10kHz. For practical implementations, such as in the EPI hearing aid, an 8kHz sampling rate is an practical and economically sensible choice. For example, one such device that performs the required function is the 16-bit sigma delta linear Codec chip AD28MSP02, available from Analog Devices. The data was acquired in sections of about 15 seconds length, which was possible because pauses had been left between groups of sentences in the passages. The level was set up to make use of the full 12bit range of the A/D converters. The passages were placed into a set of SFS files (Huckvale, 1988) to facilitate further signal processing and manipulation.

9.3.3 Automatic alignment of the speech and laryngograph signals

An automatic and highly reliable method was devised to align the speech and laryngograph signals in the training data that did not require any distance measurement to be made between the speaker and recording microphone. The first stage involved in this alignment was the estimation of the period markers derived from the laryngograph

signal. These markers were aligned to correspond to the speech pressure waveform using a two stage process.

9.3.4 Initial bootstrap alignment

In the first phase of the alignment procedure, the peaks of one speech file, corresponding to one particular distance, were found using the peak-picker algorithm (Howard & Fourcin, 1983). A linear alignment program then calculated the cross-correlation of coincidences of all the period markers from the reference and peak-picker algorithms for a range of positive and negative offsets. The correlation peak was found automatically and its location corresponded to the time-shift between the peak differential in the laryngograph cycles and the peaks in the speech. Notice that this is the same procedure described in chapter 6 to align reference and test period marker with a constant time-shift between then. This was then used to time-align the reference period markers. This provided one file of appropriately aligned training data for the MLP-Tx algorithm, and a direct speech MLP-Tx algorithm (description given later) was initially trained upon this. Period markers generated by the partially trained MLP-Tx algorithm were used to align the speech and the laryngograph on ALL the training speech data. This ensures that all the training data is aligned self-consistently to within 1 sample at the 8kHz sampling rate. This procedure has been found to be very successful.

9.3.5 Checking speech polarity

A vital issue concerning the alignment of the speech and the laryngograph signals is that of speech polarity. It was very important that the speech polarity is self-consistent for all the recordings, because otherwise the alignment could not be performed. Observation of the speech pressure waveform alone is not sufficient to guarantee speech polarity. In addition, it is also not possible to use the quality of the frequency estimates from the MLP-Tx algorithm to reliably estimate speech polarity, although it does often exhibit a preference for speech of the same polarity for which it was trained. However particularly in the initial boot-strap alignment stage, when the MLP-Tx algorithm was not fully trained, it was not always easy to determine polarity on the basis of the

frequency contours. This is illustrated in figure 9.2. One manifestation of speech inversion is the location of the period markers. This is illustrated in figure 9.3. There is a significant shift in marker location depending upon speech polarity. This phenomenon can be used to give a very clear indication of speech polarity if the cross-correlation alignment procedure is applied to MLP-Tx period markers found using both polarities. The cross-correlation for the correct polarity showed a much more distinct correlation peak than for the incorrect polarity. This is illustrated in figure 9.4. Notice that the difference shown in this figure is considerable, even though the correponding frequency contours showed little difference. The correctly time aligned speech, laryngograph and MLP-Tx output signals are shown in figure 9.5.

9.4 USING DIFFERENT PRE-PROCESSING SCHEMES

9.4.1 The task of the pre-processing stage

The input vector to a pattern classifier should be chosen such that it contains the information necessary to permit the desired discrimination to be carried out. In addition, the data should be represented in such a way that aspects of the signal of importance in discrimination are emphasised as much as possible, whilst at the same time information that is not required for the discrimination should be suppressed. Three different pre-processing strategies were investigated. The speech was either used directly after a linear scaling, after processing by a wideband filterbank (similar to before) or after processing by an auditory filterbank. Figure 9.6 illustrates the different pre-processing schemes used.

One problem with the original design for MLP-Tx was that the location of vocal fold closures in time were too imprecise for many applications. There is naturally a compromise between the amount of processing required and the time resolution. Adopting a brute force approach, increasing the resolution by a factor N results in an input vector with N times as many elements, for a given time window width. In addition, it increases the number of frames in a given unit of time of the input data by a factor N. Consequently there is an increase of computation in the classifier by a factor

of N^2. The computation is also proportional to the number of output channels generated by the pre-processing scheme. Using direct speech input only generates a single output channel. In this case, using the full sampling rate of 8kHz without decimation is not too computationally expensive, because there is only one input channel. Using a high input sampling rate posses much more of a problem in computational terms when the filterbanks are used for pre-processing, because they give rise to multiple output channels. For this reason, the full 8kHz sampling rate was only investigated on the direct waveform pre-processing configuration.

The input pattern vectors were generated from a contiguous number of frames from the input data. The two parameters that specify the vector generation are the number of frames in the window and the offset location from the beginning of the window at which the output target occurs.

9.4.2 Symmetrical input window

The window was chosen with regard to the minimum useful T_0 value that would be encountered in the speech signal, and with regard to the computational load. It is a reasonable assumption to make that for the window to operate satisfactorily it will have to give evidence of at least one period at a time within it, and preferably more. Assuming good operation for 100Hz and higher frequencies, this set a minimum symmetrical window size of 20ms.

9.4.3 Asymmetric input window

If the offset of the observation window is asymmetric, it can incorporate the effect of another excitation for lower fundamental periods than a symmetrical window would permit.

Possible advantages of using an asymmetrical window are that one may be able to use a smaller window overall than otherwise needed, and secondly it may be possible to reduce the time delay from the system if a short look-ahead in time can be used with

a rather longer look-back in time. The window parameters are illustrated in figure 9.7. Tests on the evaluation data showed that a 20ms asymmetric window produced fewer chirp (fewer false markers) errors than the equivalent symmetrical window. However, the system was a poorer voicing detector (see appendix A.8).

In all the final results reported in this chapter, a symmetrical input window of 20.5ms was used.

9.4.5 Direct operation on the sampled speech pressure waveform

To process the input speech samples directly, the speech was first multiplied by a small number (0.001) to scale the values of the speech samples to within the ±1.0 range. This is necessary because the MLP system used trained best when the range of inputs was of the same order as the output range of the sigmoid non-linearity.

9.4.6 Filterbank to approximate wide band spectrogram

The filterbank comprised six second order IIR band-pass Butterworth filters with -3dB points of 50-300Hz, 300-600Hz, 600-900Hz, 900-1200Hz, 1200-2000Hz, 2000-3000Hz. The outputs were half-wave rectified, low-pass filtered at 1kHz, down-sampled to 2kHz and then linearly scaled to the range of ±1.0. This system was a cut-down version of the original filterbank designed to permitted its real-time operation on a portable DSP system, and its design is described in chapter 11 (Howard & Walliker, 1989; Walliker & Howard, 1990). An example of the output waveforms from this filterbank is shown in figure 9.8.

9.4.7 Pre-processing using an 'Auditory filterbank'

It is well known that the auditory system performs filtering of the input sounds incident on the ears. It was considered prudent to investigate an input filterbank using filters with some of the properties of those in the auditory system, because one can be sure that they do not discard important information relating to the speech excitation. This can be

understood because with such an overlap, the overall power in the input signal is maintained in the output from the filters. For the purposes of this work, a simplified auditory model was used that consisted of a bank of gamma-tone filters, with a 1 ERB (equivalent rectangular bandwidth) spacing between their centre frequencies. This is the minimum filter density that maintains the information present in the input signal. This resulted in 12 filter channels to cover the required frequency range of 50Hz to 1kHz. The output from the filterbank channels were then half-wave rectified, low-pass filtered at 1kHz, down-sampled to 2kHz and then linearly scaled to a ±1.0 range. This filterbank is described by Holdsworth, Nimmo-smith, Patterson & Rice (1988). An example of the output waveforms from this filterbank is shown in figure 9.9. A comparison of the output from the auditory filterbank and the wideband filterbank is illustrated as a grey-level display in figure 9.10.

9.6 TRAINING THE MLP CLASSIFIER

9.6.1 Long training times

The original MLP-Tx algorithm took a long time to train, because the MLP algorithm needed many iterations over the data-set, each of which required a lot of processing. Two techniques were used to speed up the training.

9.6.2 Adaption of the learning rate and the momentum term

One such technique is due to Chan & Fallside (1987) and it operates by dynamically adjusting the momentum term and learning rate parameters.

9.6.3 The number of patterns used to estimate weight changes

In their work, Chan & Fallside used the adaption scheme in conjunction with an updating of the weights over the **entire** training data set whenever practical, or over representative sub-sets (batches) of the data for those circumstances wherever such a scheme was not practical. The advantage of using the latter procedure is that it is

possible to make MLP weight changes over a relatively small set of patterns, but ones which give a good reflection of the possible range of patterns in the data set. This is better than making the update after each pattern, because the latter is not guaranteed to give a good gradient descent, and the direction of the weight changes tends to fluctuate widely between successive updates, which makes it impossible to use adaptive learning rate and momentum term learning schemes. In practice one would not wish to update the weights only once per pass of all the data, since this would result in very slow learning. This is because it is only possible to alter the weights by a relatively small amount per update, and since the time taken to determine each update would be relatively large in this case, to perform enough updates to find a suitable solution would take a long time.

9.6.4 Sorting the pattern vectors

To implement the batch learning, the data pattern vectors used to train the MLPs were sorted into representative groups such that each group contained at least one pattern corresponding to the presence of a period marker.

9.7 SELECTIVE EMPHASIS TRAINING OF THE MLP

Another method to speed up training was to use selective emphasis of the training data. This method works by changing the relative emphasis of different pattern vectors, depending upon various factors during training, and was developed during the course of this work. It operates by scaling the weight changes that result from a given pattern by a factor that depends upon the estimated importance of that pattern. The importance of the pattern vector is estimated with regard to several considerations.

9.7.1 Emphasise incorrectly recognized patterns

It has been found valuable to concentrate the training on the patterns that are falsely recognized, and not overwhelm the MLP with less important weight changes from the data that is dealt with acceptably. Using this scheme, the network is only trained on

those patterns it has difficulty with. This can be achieved by making the emphasis dependent on the output from the MLP as well as the target pattern class. Thus a pattern that results in an output above a preset threshold is made to give rise to weight changes which are scaled differently than if the output was below the same threshold. In practice three thresholds were employed, one for high output target pattern classes, one for low output target patterns and another for uncertain output target pattern classes. It is possible to arrange the emphasis such that patterns that give rise to outputs which are close enough to the targets are ignored (thus speeding up program operation). This makes it possible to reduce the contribution of certain regions in the training data to zero.

9.7.2 De-emphasis of the importance of boundaries

In addition to emphasizing patterns that are wrongly recognized, it was found beneficial to take less notice of patterns if their precise labelling was not important or even uncertain. This was the case in the close vicinity of a period marker. The input patterns adjacent to the one corresponding to the excitation marker will be similar. Consequently, it is difficult to train the MLP to generate one class (1.0 in this case) at this pattern, and the other class (0.0 in this case) immediately around it. However, providing the MLP can be trained to generate an output that has a monotonic rising and falling transitions around the period marker frame, the fact that adjacent output frames from the MLP are non-zero will not be important.

The use of an uncertain region around the period marker was found to be very important and beneficial. The different zones are shown in figure 9.11. It can be seen that zone0 corresponds to an unvoiced signal, zone1 corresponds to the pre-period marker uncertain zone, zone2 corresponds to a period marker, zone3 corresponds to the post-period marker uncertain region and zone4 corresponds to the region in between period markers within a voiced segment of speech. The presence of a period marker (zone2) used a high MLP target of 1.0, whereas the absence of a period marker (all other zones) used a low MLP a target of 0.0. Thresholds of 0.1 for the low zone, 0.85 for the uncertain zone and 0.9 for high zones were employed. Notice that normally the class of the

uncertain zones would be set low (0.1). Using a high threshold here (which means that these regions are ignored, **unless** the output from the MLP is above a value of 0.85, in which case it may compete with the **true** period marker location as the local maximum. In this case, the uncertain region is used, and trained to be of low class. These thresholds are illustrated in figure 9.12.

9.7.3 Faster training with selective emphasis

The selective emphasis training substantially speeded up the training of MLP networks. Using one pattern vector presentation per weight update, there is a speed-up in passing through the training data in excess of ten times. With larger numbers of pattern presentations per weight updates (such as 1000), there is a speed-up in the passage through the data by about three times. This results from the fact that for those patterns that are correctly recognised, no back-propagation of error or adaption of the weights needs to be carried out.

The number of excitation markers in the training data used in this work far exceed the number of weights in the MLP networks, since the largest network used has about 1620 weights and there were well over 50000 period markers in the women training data set.

9.8 TRAINING DIFFERENT CONFIGURATIONS OF THE MLP-Tx ALGORITHM

9.8.1 Training different MLP-Tx configurations

The best configurations of the MLP-Tx algorithm were selected from observation of the performance of the algorithms on the evaluation data (appendix A.9). Three different experiments were carried out, using each of the pre-processing schemes described. In each case, the algorithms were only trained and tested on speech from women (results on men and women were carried out on the evaluation data set, and appear in appendix A.9). In each case, the MLP networks were trained on all the women training data. Three passes through the data were made, all using selective emphasis. The first pass was made using a weight update after each pattern presentation, with no adaption of the

learning parameters. The second pass was made using batch learning with 100 patterns contributing to weight updates, and employing learning parameter adaption. The final pass was the same, but updating weights after 1000 pattern presentations. This strategy has been found effective because the initial small updates result in fast training, whereas the later larger updates provide a final improvement in the quality of the training. The MLP networks used were as follows: For the direct speech experiment, the network had 161 inputs, 10 hidden units and 1 output unit. For the wideband filterbank, the network had 246 inputs, two layers of hidden units each containing 6 hidden units, and 1 output. This was the largest network that could be run in real-time on the TMS320C25. For the auditory filterbank, the input had 533 inputs, two layer of hidden units each containing 6 hidden units, and 1 output. All of these network configurations were arrived at by consideration to the performance of the MLP-Tx algorithms on the preliminary training data. Quantitative frequency contour comparisons illustrating the effect of MLP configuration parameters are given in appendix A.9.

9.8.2 Effect of different updates (patterns per group used for batch learning)

As discussed in chapter 7, changing the number of pattern vectors used to estimate the weight updates before any weight changes are made affects the training of the MLP. Essentially, updating per pattern presentation results in the faster initial training, but the training is of a higher quality if a larger update is employed. If normal training is employed (that is, standard back-propagation), a considerable amount of computation time is spent in adapting the weights. Consequently, the training takes longer to pass though the training data if the update is small than if it is large, although the overall training is faster. Quantitative frequency contour comparisons illustrating the effects of altering the update parameters are given in appendix A.9.

9.9 POST-PROCESSING TECHNIQUES

9.9.1 Task of the post-processor

After the output from the MLP in the MLP-Tx algorithm has been generated, it is

necessary to locate the period markers. The task of the post-processor is to take the sampled output waveform from the MLP network, and determine from it discrete events that correspond to the period excitation markers. In the preliminary work, this was done simple by means of a comparator circuit, with forward and backward inhibition. A more sophisticated schemes was also investigated. This involved the use of another MLP-Tx algorithm that was trained as before, but this time using the input from another MLP-Tx algorithm that worked in the previous fashion by processing the input speech pressure waveform.

9.9.2 Threshold with local inhibition

The simplest preprocessing scheme is one which simply assigns a frame to a high state if the value is greater than a predetermined threshold value. Local inhibition can be used to reduce the generation of spurious pulses around the main one. A flow diagram to explain the operation of this algorithm is shown in figure 9.13.

9.9.3 Secondary network continuity classifier

There is clearly some constraint on the temporal patterning of occurrences of fundamental period epoch marker locations. However, it is not always possible or desirable to make decisions over a long time-scale. For example, the constraints between adjacent fundamental period values are not always that strong, as is in the case of creaky voice. In the second instance, there may be a delay in processing that is unacceptable, unless only past information is used.

A method used to take advantage of the temporal patterning of the period markers employed another MLP network. Instead of using an MLP with input from the sampled speech pressure waveform, the output from a previous MLP-Tx algorithm is used as the input. This is illustrated in figure 9.14. This scheme was investigated using the input from the wideband filterbank, and there was a reduction in the chirp errors. Results for this system are included in appendix A.9. The output of the secondary and primary networks is examined in chapter 10.

An extension of this technique would be to use several different inputs from different MLP-Tx algorithms trained to detect different qualities of speech, or trained to operate on different speakers or environments. In this case, the secondary classifier must perform a data fusion task in order to combine the evidence from the primary extractors together to give an overall period marker estimate.

9.9.4 Generating frequency contours from the MLP-Tx algorithms

The raw MLP output were then processed to generate period markers. The task of the post-processor is to take the sampled output waveform from the MLP network that is generated as a function of time, and determine from it discrete events that correspond to the period excitation markers. This was done as before by means of a comparator circuit, with forward and backward inhibition. The period marker outputs from the respective MLP-Tx algorithms were generated on the women testing data, and the outputs converted to frequency contours sampled at 100Hz by taking the reciprocal of the resulting period values. This format was required to permit comparison between the MLP-Tx algorithm and the established techniques. For the comparisons presented here, frequency contour comparisons were used, because this is a format that all algorithms could generate. Only comparisons on female speech are given. In all cases, the laryngograph based algorithms (described previously in chapter 8) were used to provide the reference frequency contours.

9.10 COMPARING BEST MLP-Tx CONFIGURATIONS AGAINST ESTABLISHED TECHNIQUES

9.10.1 Standard fundamental frequency analysis techniques for comparison

Comparisons of the best configurations of three pre-processing configurations of the MLP-Tx algorithm and two established algorithms and were made against the reference laryngograph analysis system. That is, the best reduced filterbank MLP-Tx algorithm that could run on the TMS320C25 was selected, the best auditory filterbank MLP-Tx algorithm was selected and finally the best direct speech MLP-Tx algorithm was

selected. The established techniques chosen for the purpose of comparison were cepstral analysis and a peak-picker. The cepstrum algorithm was selected because it is often regarded as a standard (chapter 4), whereas the peak-picker was chosen because it is the algorithm the MLP-Tx algorithm is intended to replace. These were both described in chapter 4.

9.10.2 Discussion of results

The results given are the average of the 20 women speakers. Figure 9.15 shows the gross error for the six different algorithms. It can be seen that the cepstral analysis gives the fewest number of gross errors, and the MLP-Tx algorithm using direct speech operation gives the next least. The MLP-Tx algorithm using the auditory filterbank is better than with the wideband filterbank. The peak-picker gave the worst performance.

Figure 9.16 shows the chirp errors for the six different algorithms. The cepstrum gives the lowest chirp errors, and the direct speech MLP-Tx algorithm gives second best performance. The auditory filterbank MLP-Tx algorithm is again better than the wideband filterbank MLP-Tx algorithm.

Figure 9.17 shows the drop errors for the six different
algorithms. The lowest error rate is due to the cepstral algorithm. The direct MLP-Tx algorithm is again second best. The auditory filterbank MLP-Tx algorithm is again better than the wideband filterbank MLP-Tx algorithm. The peak-picker algorithm gave the most drop errors.

Figure 9.18 shows the standard deviation of the fine frequency differences for the six different algorithms. The cepstrum algorithm gave best performance, closely followed by the direct speech MLP-Tx algorithm. The two filterbank MLP-Tx algorithms gave the worst performance. This is probably because their period estimates were determined to a precision of 0.5ms, whereas all the other MLP-Tx algorithm used the 0.125ms resolution of the input speech.

Figure 9.19 shows the voiced-to-unvoiced errors for the six different algorithms. These results show that the wideband filterbank MLP-Tx algorithm made the fewest errors, and the direct MLP-Tx and the peak-picker were about the same, with the auditory filterbank MLP-Tx not far behind. The cepstral algorithm gave the most errors.

Figure 9.20 shows the unvoiced-to-voiced errors for the six different algorithms. The direct MLP-Tx algorithm gave the best results, with the wideband filterbank MLP-Tx algorithm next. The peak-picker and auditory filterbank MLP-Tx were about the same, with the cepstral algorithm giving the most errors and performing badly.

9.10.3 Conclusions

The MLP-Tx algorithm has been shown to be an effective means of speech fundamental period estimation. Pre-processing that employed direct operation of the speech pressure waveform generally gave better results than using either a wideband filterbank or an auditory filterbank. This result supports the statement by Widrow & Lehr (1990) that it is often better to let the network classifier develop its own pre-processing rather than to try and devise it a priori.

The performance of the direct speech MLP-Tx algorithm exceeded that of the simple time-domain algorithms used for comparisons in all the tests. It also compared favorably to the established techniques of cepstral analysis in terms of accuracy and gave better voicing determination performance, although its performance in terms of gross errors was not quite as good. However, care must be take to compare like-with-like, since cepstral analysis inherently averages frequency values over the analysis window (which was 50% wider that of the MLP-Tx algorithm), whereas the MLP-Tx algorithm locates the excitation points on a period-by-period basis.

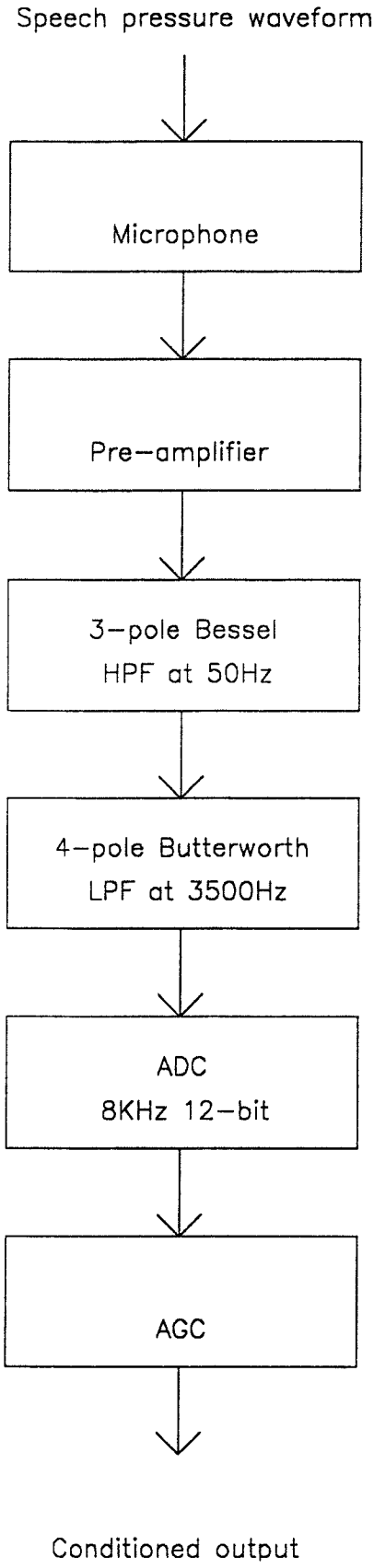


Figure 9.1 The input conditioning strategy used for the MLP-Tx algorithm.

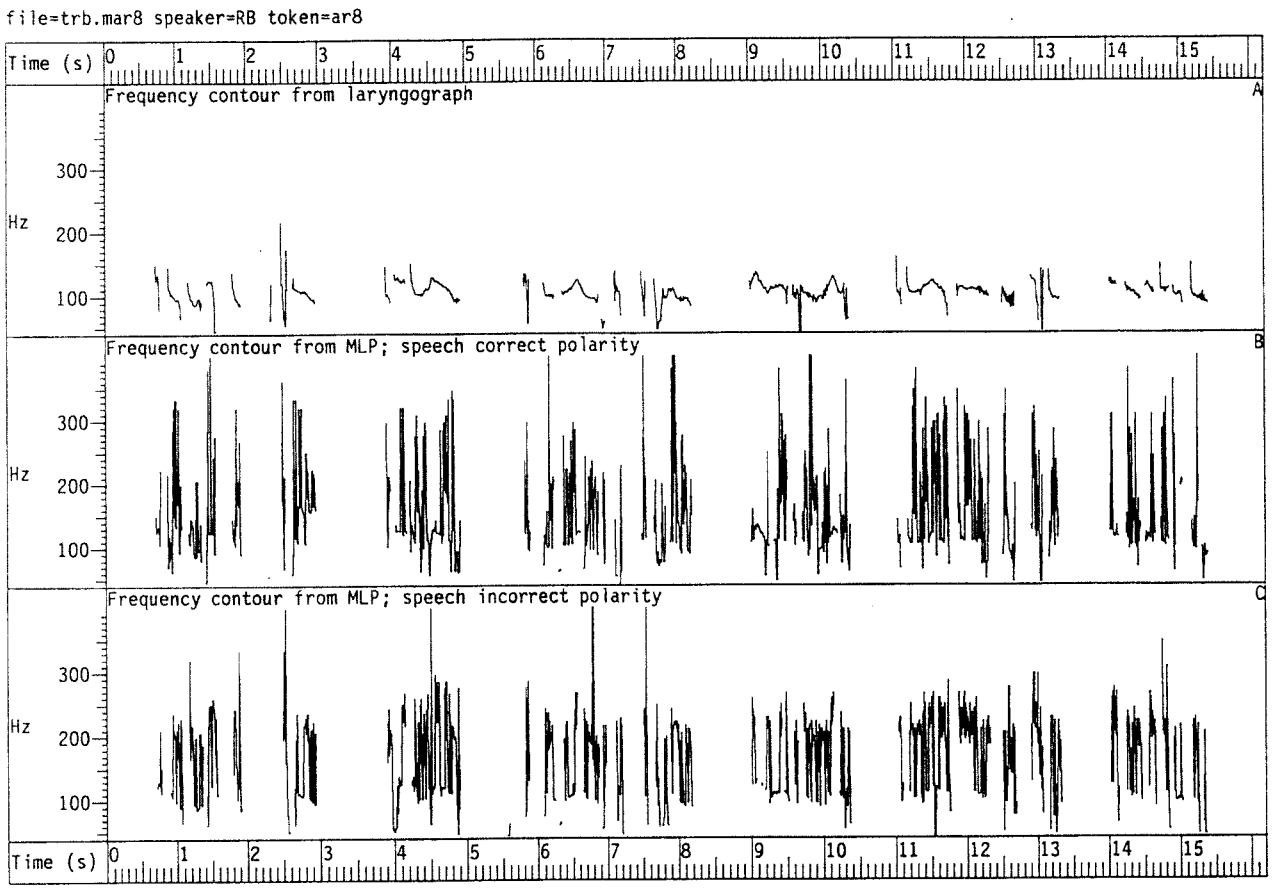


Figure 9.2 Plot illustrating the effect of speech inversion on frequency contours. These contours were obtained from a partially trained MLP-Tx algorithm, and was used as a means of speech polarity determination and time-alignment. The reference laryngograph contour is shown in trace A. Traces B and C show the frequency contours from the MLP-Tx algorithm with correct and incorrect polarity speech respectively. In this example, the performance is poor in both cases, and the correct polarity give barely any observable improvement in contour form. It would not be possible to determine polarity on the basis of these contours. The utterance is for paragraph ar8 from the Arthur the rat passage, from a male subject.

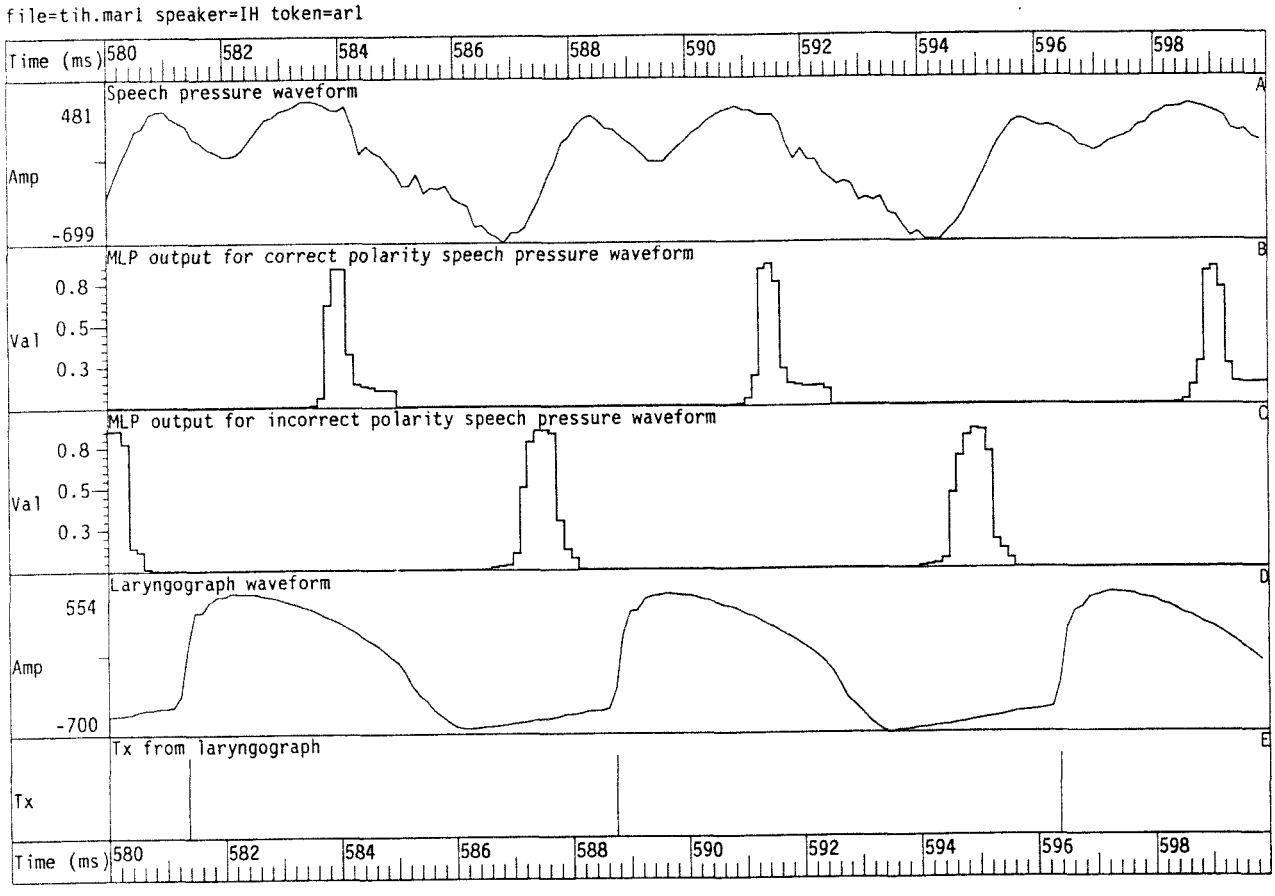


Figure 9.3 Plot showing the different occurrence times of MLP-Tx period marker estimates, depending upon the speech polarity.

Trace A shows the speech pressure waveform. Traces B and C show an output from a partially trained MLP operating on correct and incorrect polarity speech respectively. It can be seen that the two sets of locations differ by up to half a period in this case. Notice that this order of delay can also arise because of the time-delay between the speech and laryngograph signals and is therefore the effects of inversion and delay could easily be confused.Trace D shows the unaligned laryngograph waveform (that is, it is shown before alignment with the speech pressure waveform using the period marker cross-correlation procedure). Trace E shows the period markers obtained from the laryngograph signal. The utterance is the sound /w/ from a male subject.

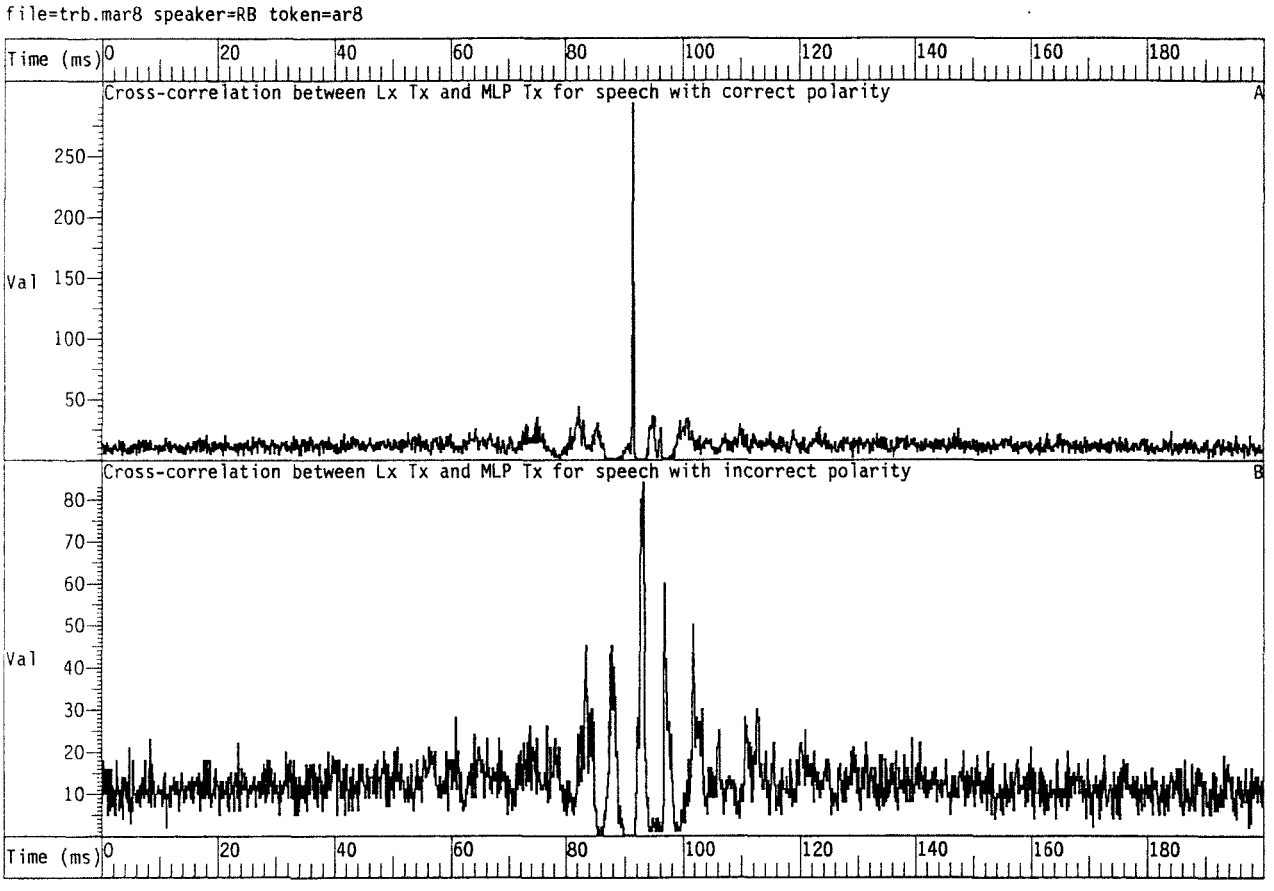


Figure 9.4 Use of the cross-correlation between the reference and MLP-Tx period markers as a means of polarity determination.

This plot is for the same speech passage as shown in figure 9.2. In the top of the figure, trace A shows the cross-correlation for the correct speech polarity, whereas trace B shows the cross-correlation for the incorrect polarity. It can be seen that this measure gives a clear indication of speech polarity, and provides a reliable method of its estimation whereas observation of the frequency contours did not. It simultaneously provides the time delay between the speech pressure waveform and the laryngograph signal that is need to align them.

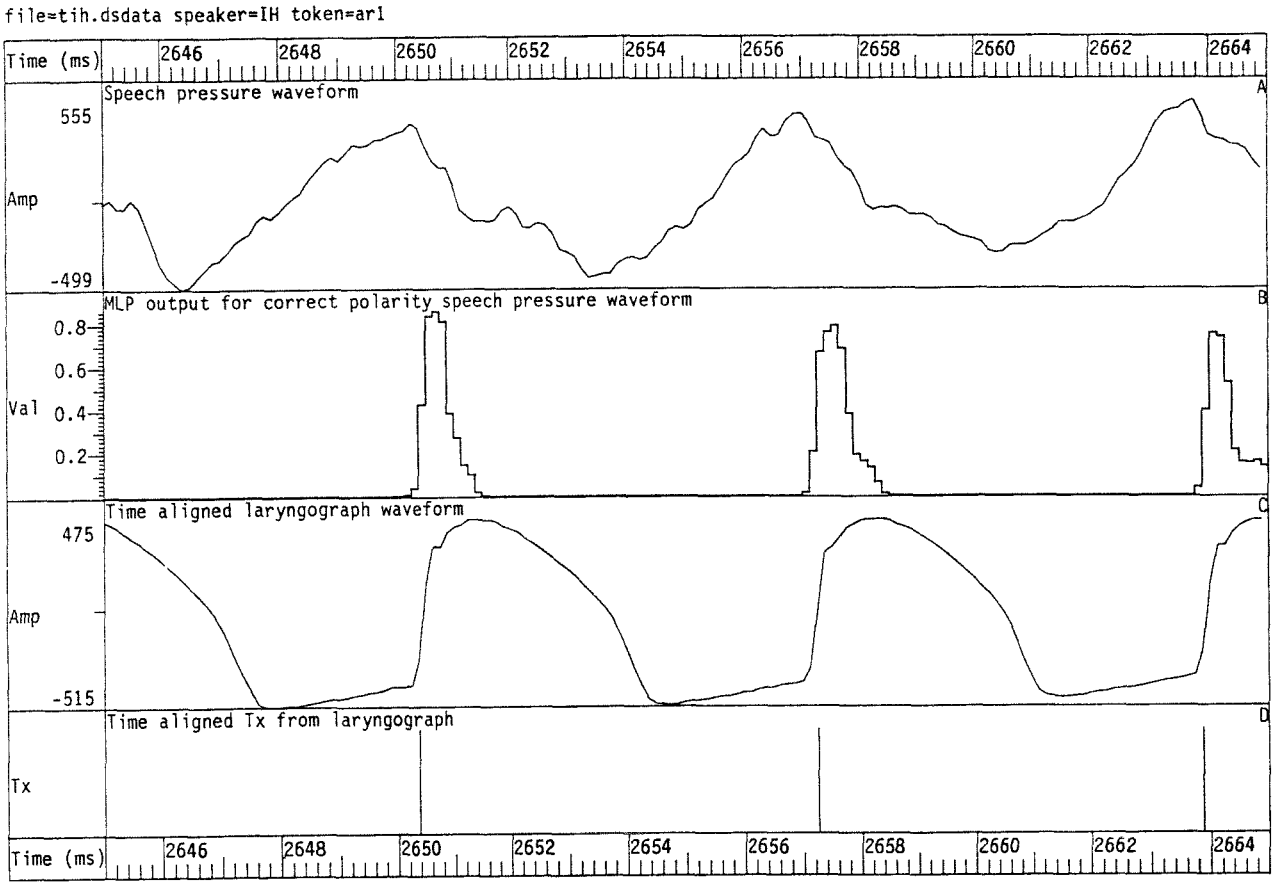


Figure 9.5 Alignment achieved using cross-correlation between the MLP-Tx and laryngograph period markers.

Trace A shows the speech pressure waveform, and trace B shows the output from a partially trained MLP-Tx algorithm operating on it. The aligned laryngograph waveform and associated period markers are shown in traces C and D respectively. The utterance is the sound /v/ from a male subject.

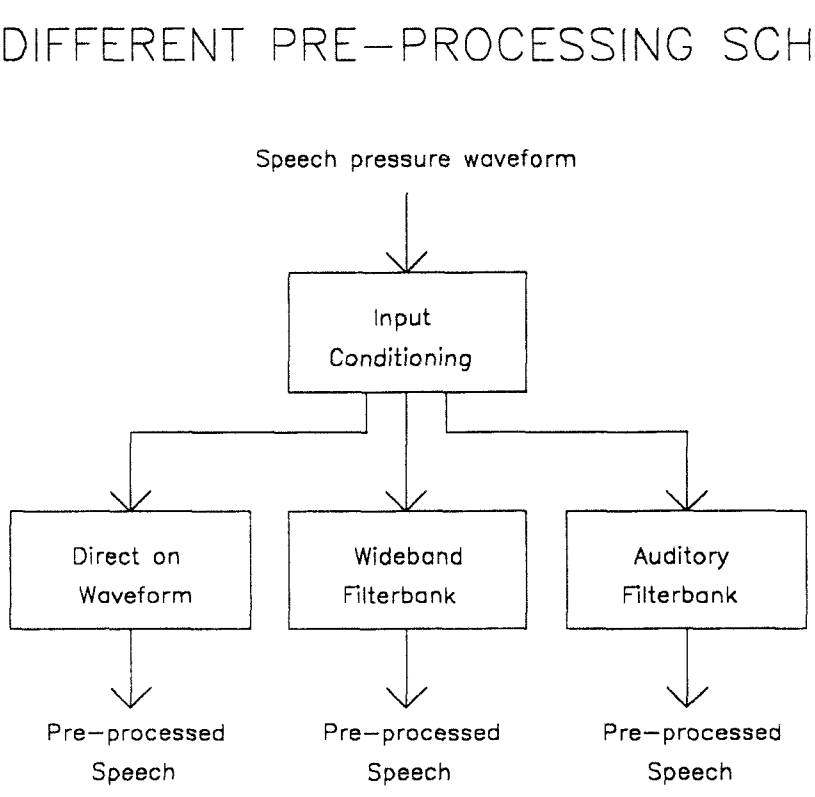


Figure 9.6 Different pre-processing schemes used for the MLP-Tx algorithm.
The three different approaches involved using the pattern classifier directly on the samples time-waveform, on the output from a wide-band filterbank and finally a filterbank with filters that share some of the characteristics of auditory filters.

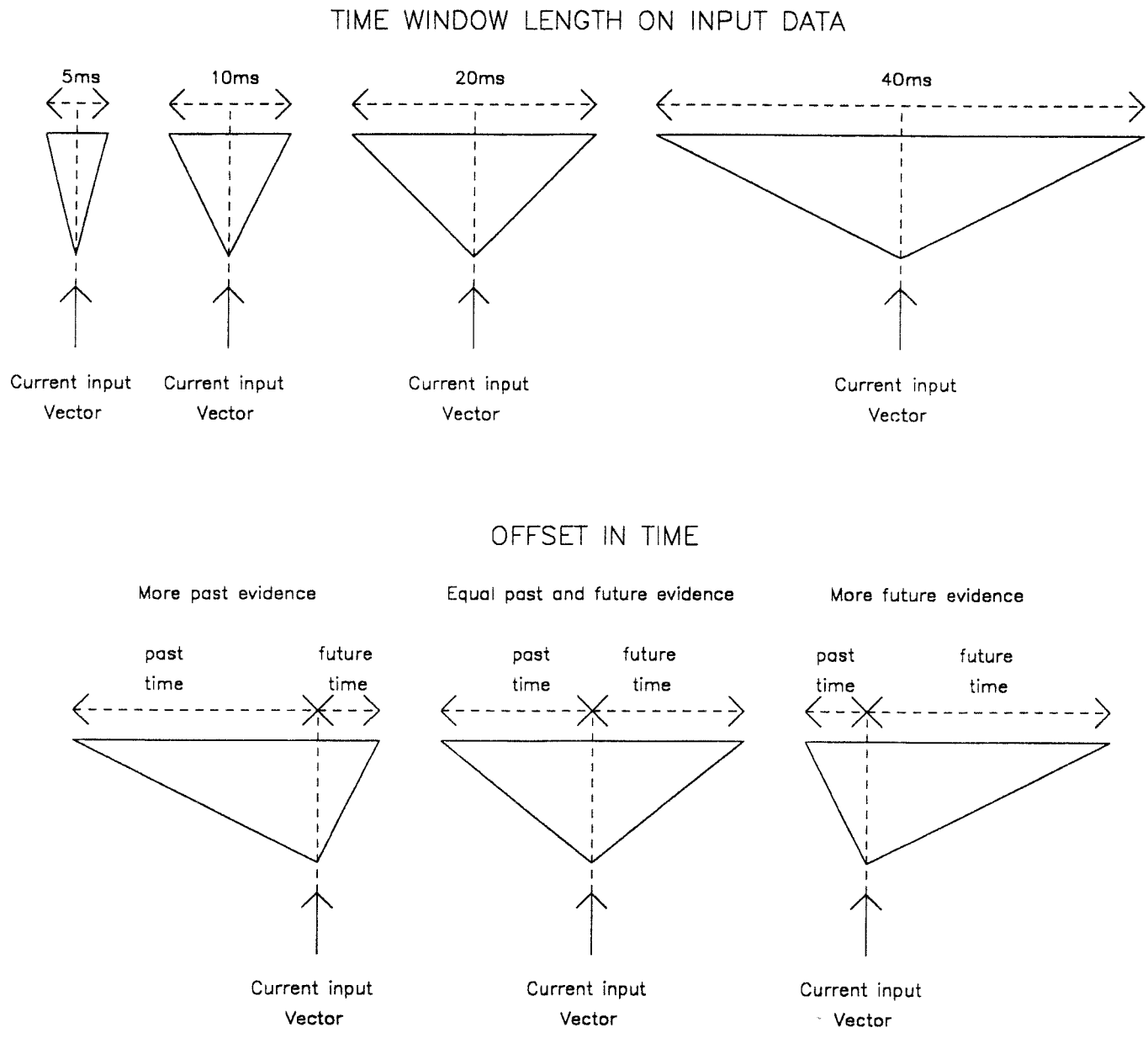


Figure 9.7 Different window configurations tested with the MLP-Tx algorithm. Window length can be varied. In addition, asymmetrical windows can be used, employing either more past or future evidence.

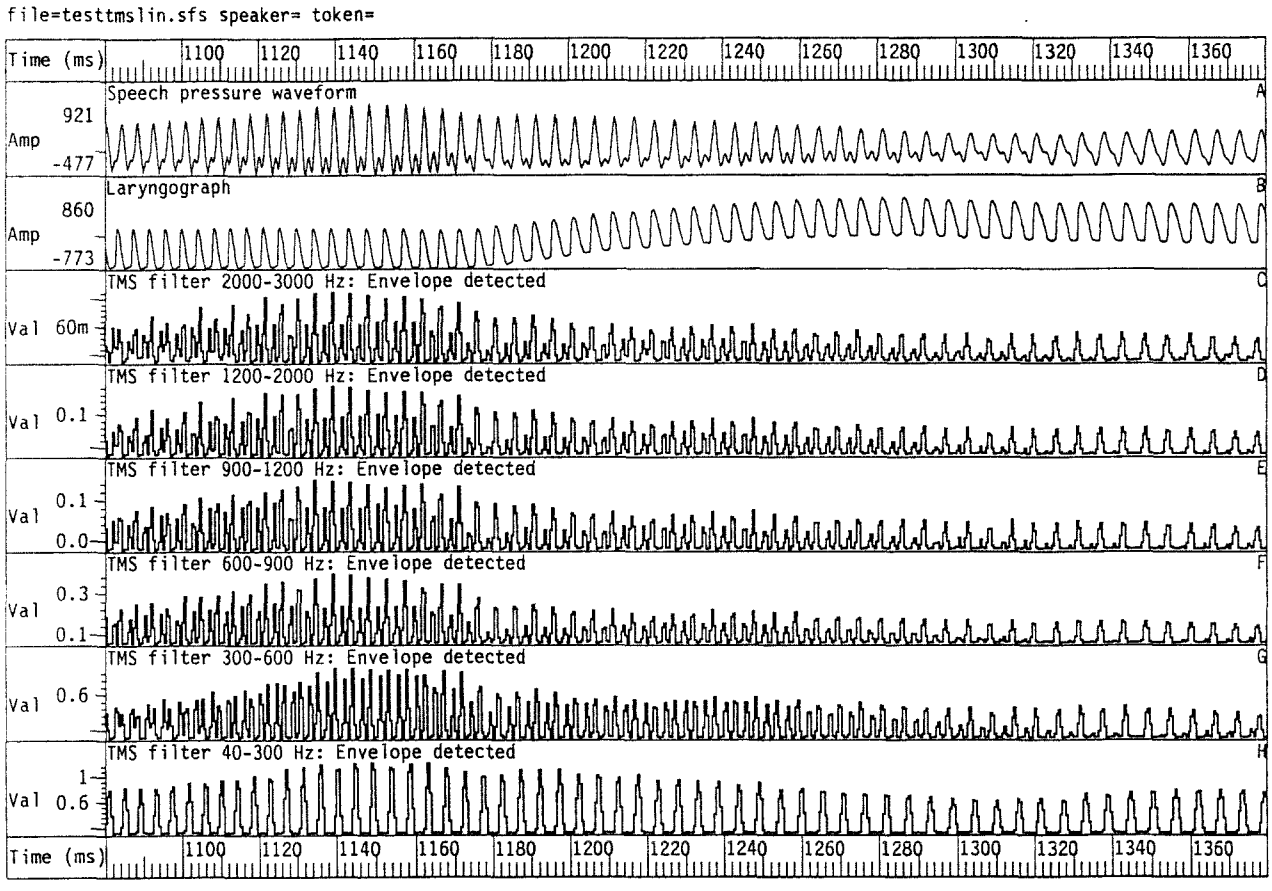


Figure 9.8 Output waveforms from wideband filterbank.

Plot showing speech, laryngograph waveform and corresponding output waveform from the six channels of the wide-band filterbank. The outputs from the envelope detectors are linearly scaled in this example. The periodicity in the speech pressure waveform is evident in output channels.

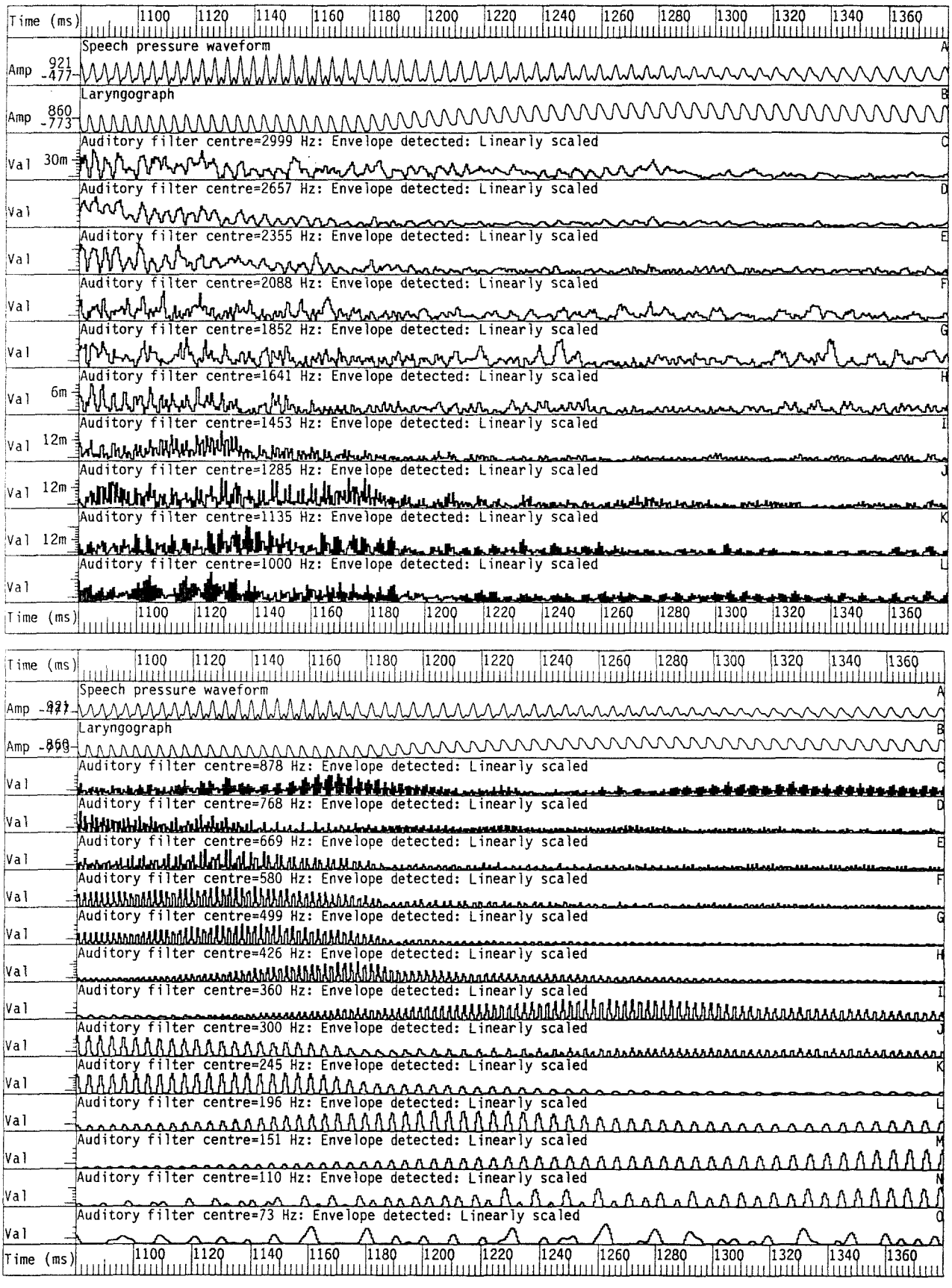


Figure 9.9 Output waveforms from auditory filterbank.

Plot showing speech, laryngograph waveform and corresponding output waveform from the 23 channels of the auditory filterbank. The outputs from the envelope detectors are linearly scaled in this example. The periodicity in the speech pressure waveform is evident in output channels.

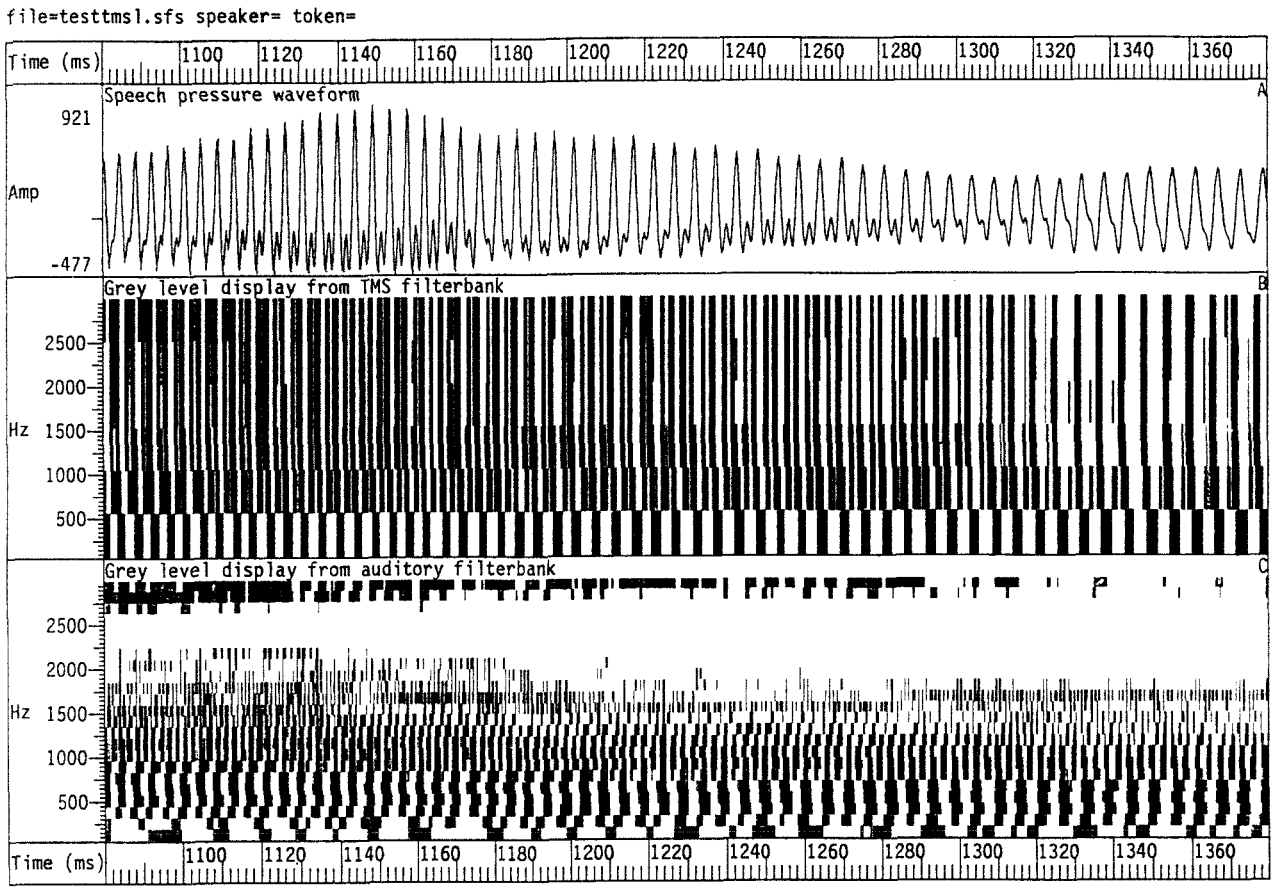


Figure 9.10 Diagram showing a piece of input speech and the corresponding wideband and auditory filterbank outputs.
Both are displayed on a grey-level scale.

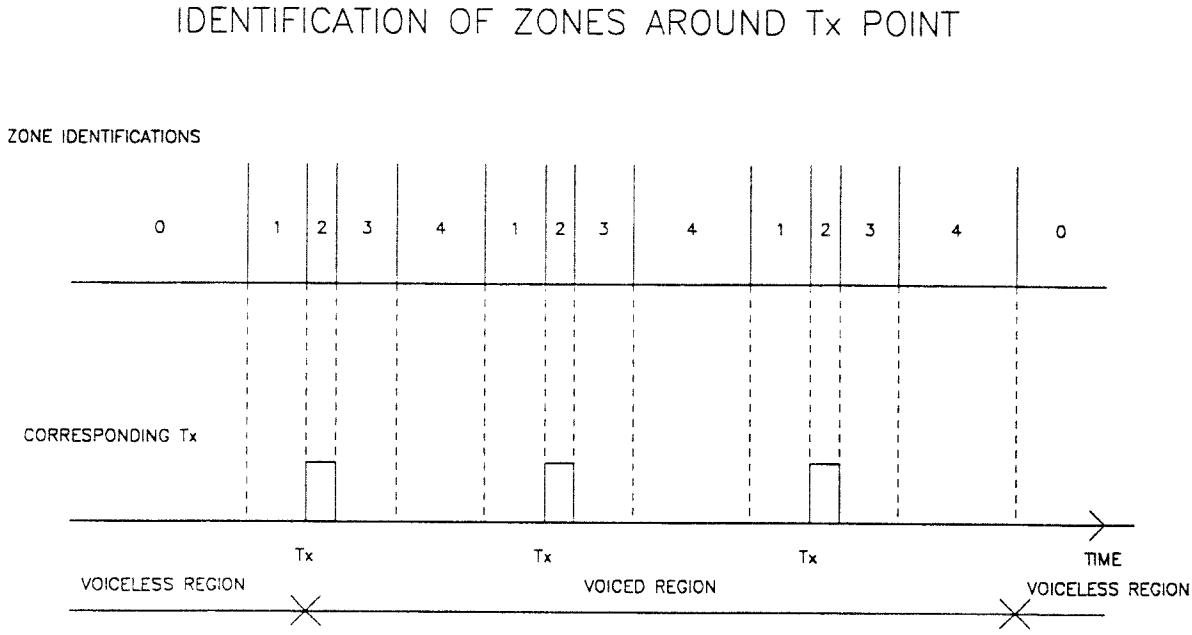


Figure 9.11 Definition of the regions around period excitation time markers.

The labelling of different zones of the training data makes it possible to treat the zones differently during training. In particular, the importance of patterns that occurs in zones 1 and 3, before and after a period excitation marker, can be de-emphasised.

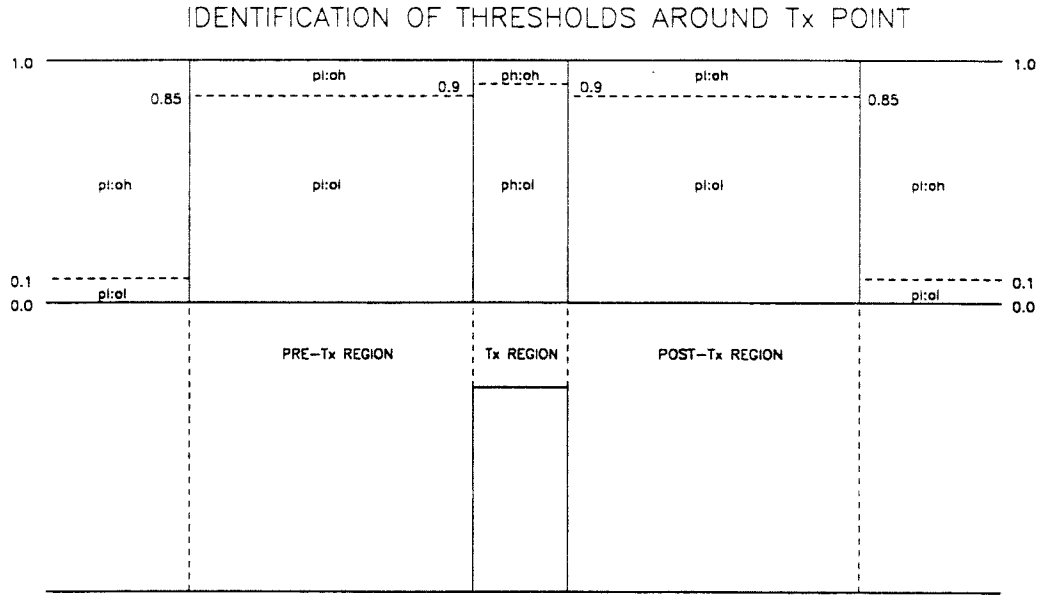


Figure 9.12 Identification of thresholds around a period excitation marker.

These thresholds are used to determine whether or not an output is close enough to its target. This operation is used during selective emphasis training of the MLP. For the period excitation marker zone, a threshold of 0.9 is employed. For the zones adjacent to the period excitation marker zone, a threshold of 0.85 is employed. Elsewhere, a threshold of 0.1 is used.

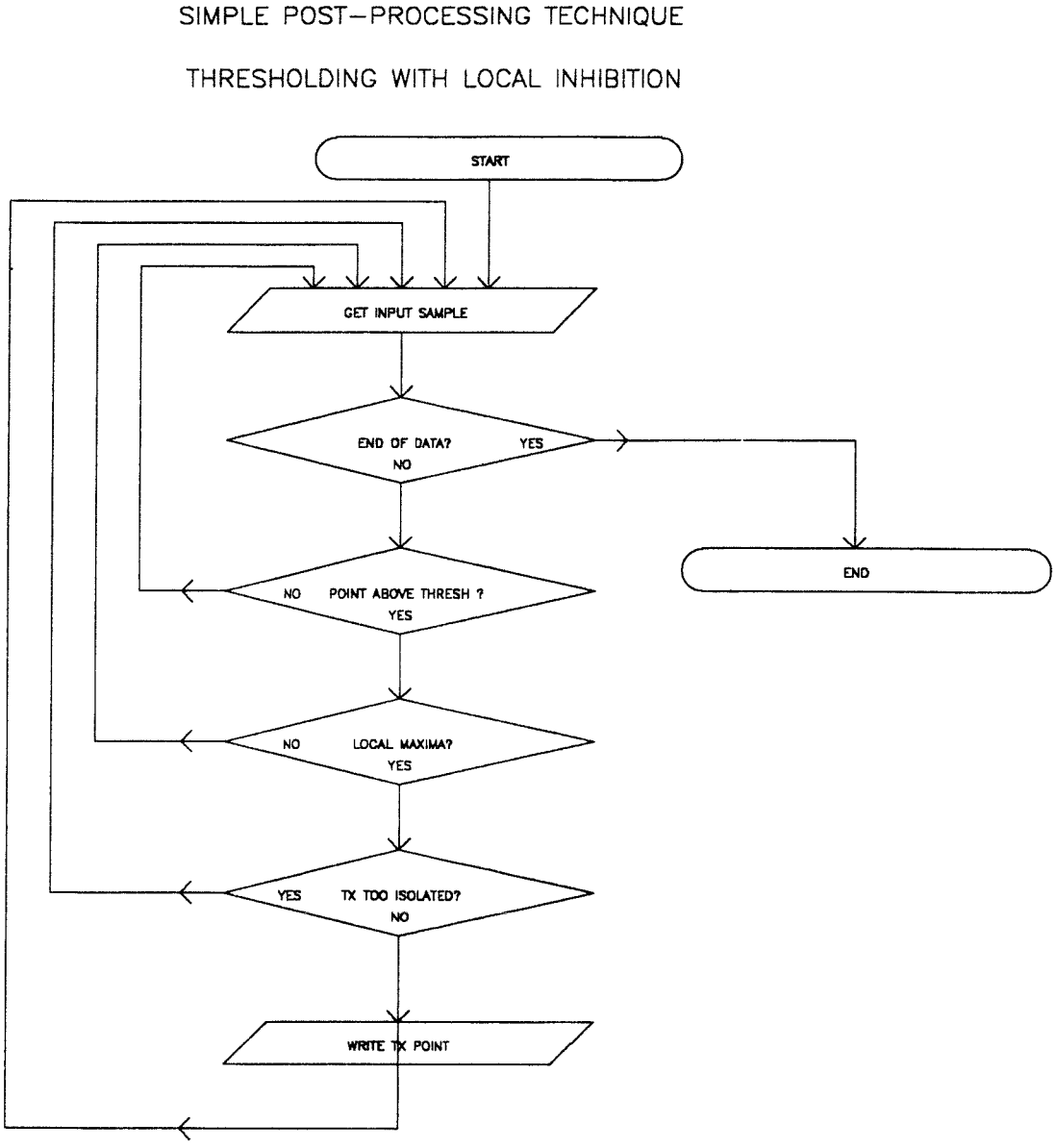


Figure 9.13 Flow diagram for the operation of simple threshold with local inhibition post-processing algorithm.

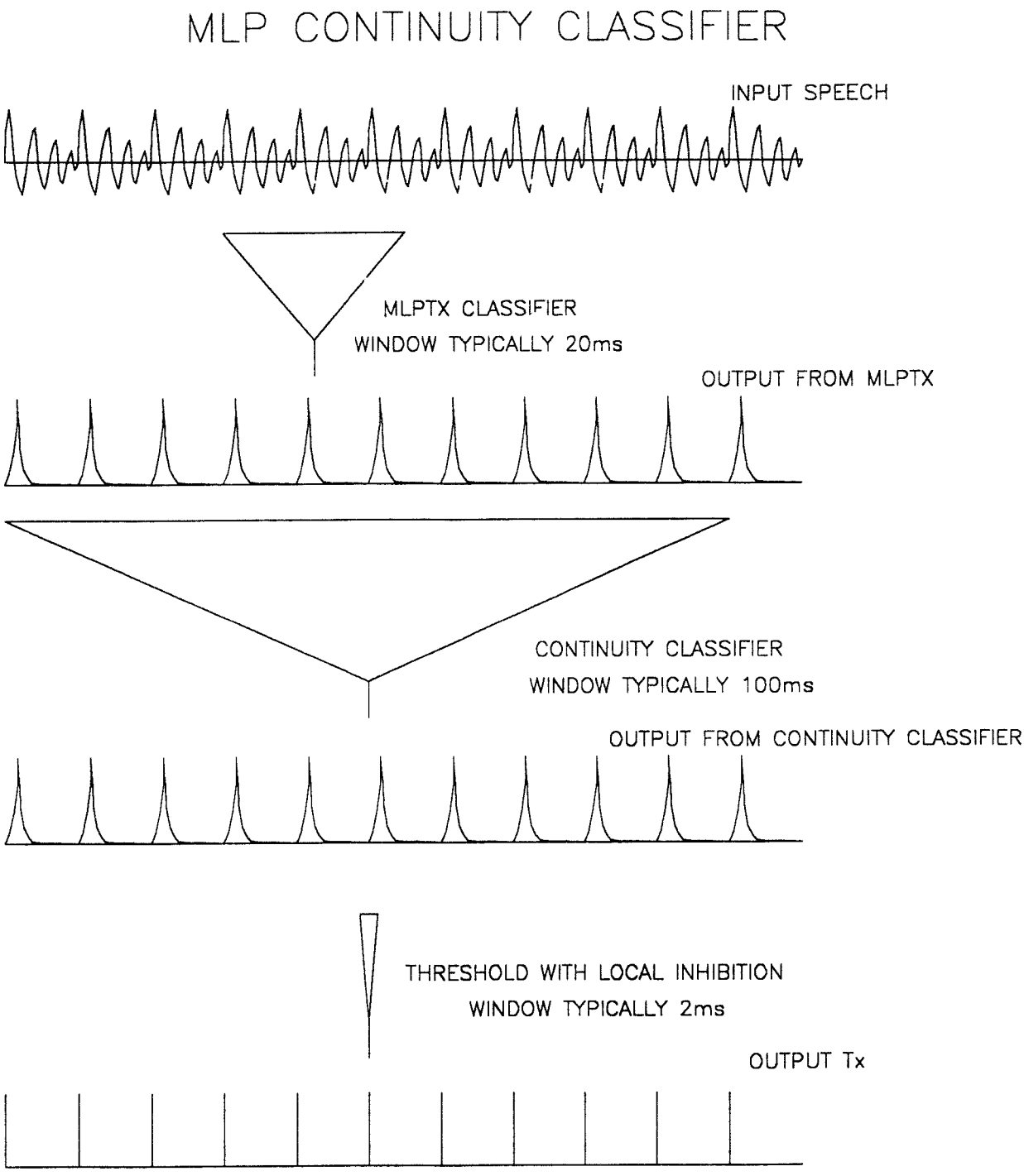


Figure 9.14 Schematic diagram illustrating principle of using a secondary pattern classifier for post-processing.

The first MLP-Tx algorithm generates an output in the normal way. This is then used as the input to a secondary MLP-Tx algorithm, which is trained as before to estimate the period markers.

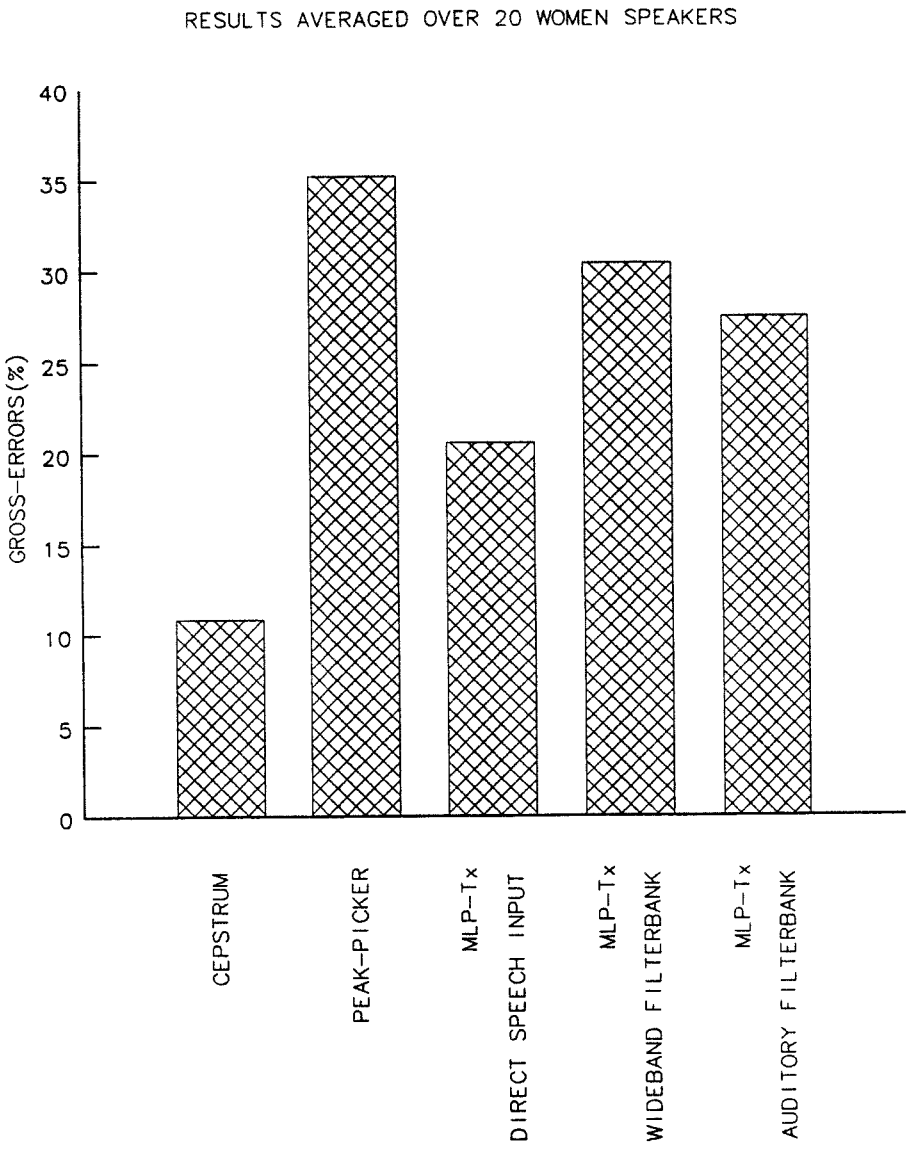


Figure 9.15 Bar-graph showing the gross errors generated by the six algorithms on final test data.

The comparisons were all made against an interactive reference algorithm that makes use of the output from a laryngograph. The results shown are the average performance over 20 different women speakers, with about 15 seconds of speech per speaker.

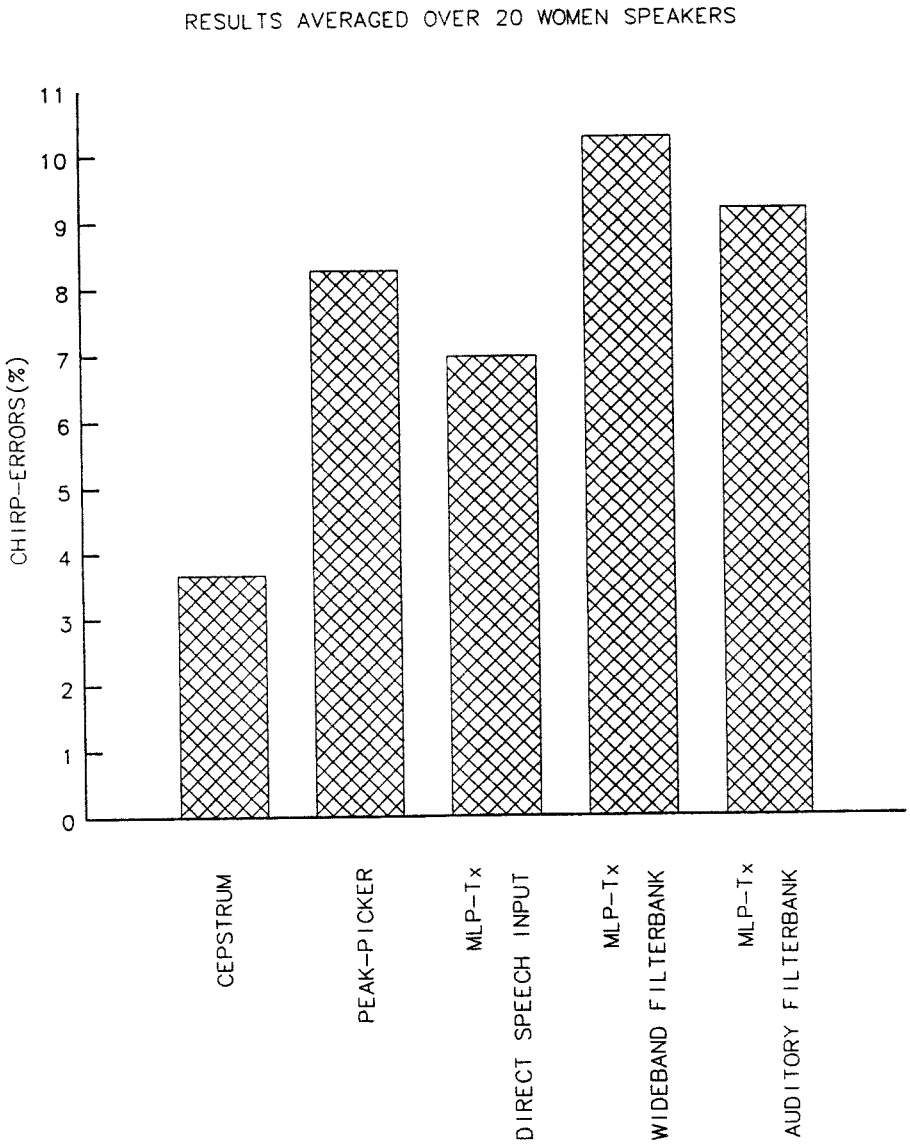


Figure 9.16 Bar-graph showing the chirp errors generated by the six algorithms on final test data.

The comparisons were all made against an interactive reference algorithm that makes use of the output from a laryngograph. The results shown are the average performance over 20 different women speakers, with about 15 seconds of speech per speaker.

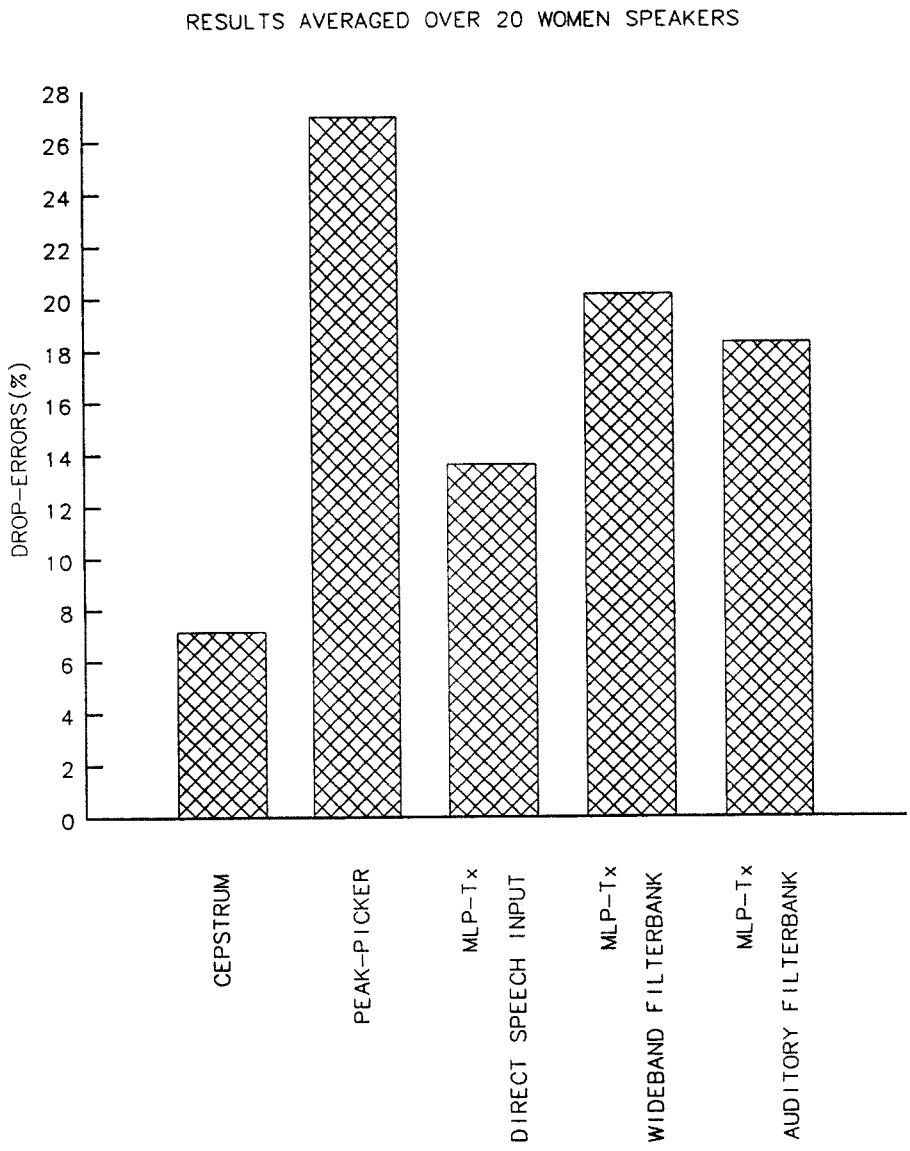


Figure 9.17 Bar-graph showing the drop errors generated by the six algorithms on final test data.

The comparisons were all made against an interactive reference algorithm that makes use of the output from a laryngograph. The results shown are the average performance over 20 different women speakers, with about 15 seconds of speech per speaker.

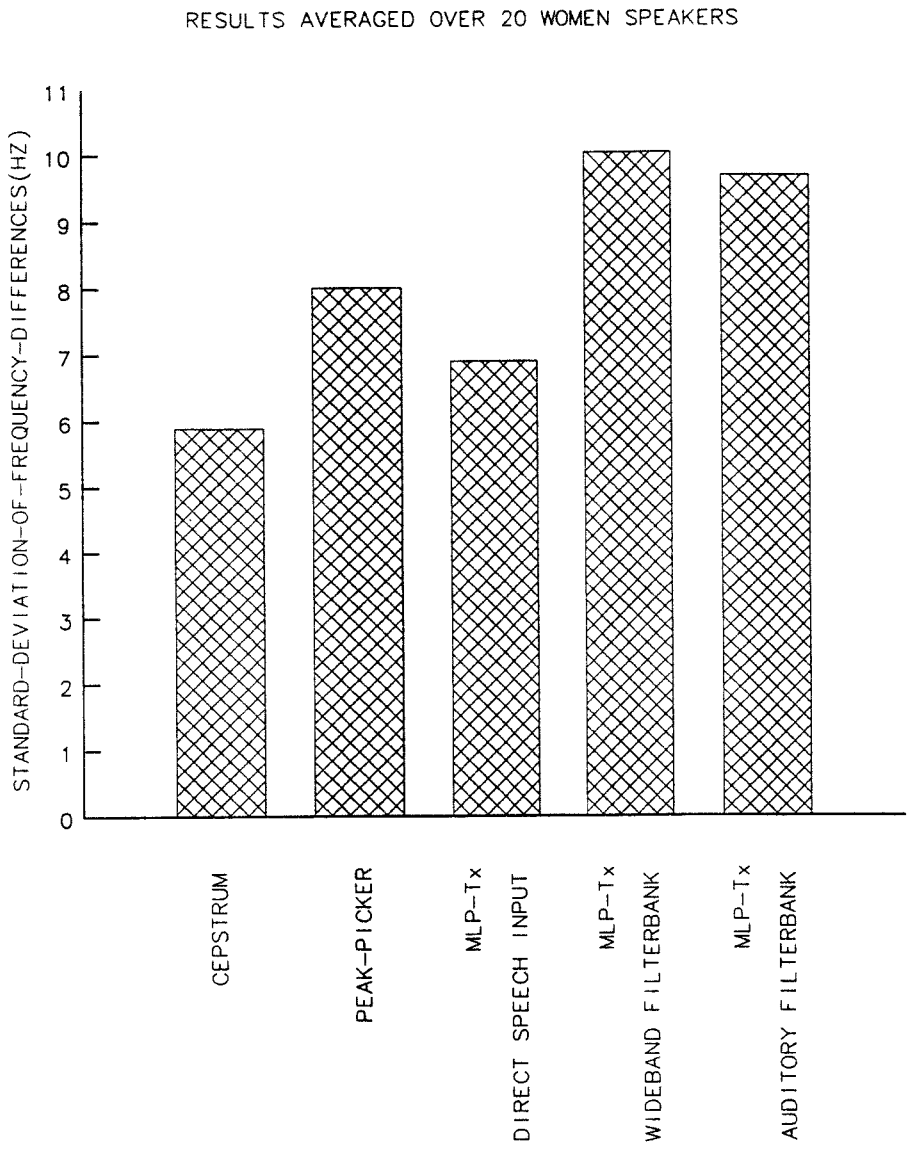


Figure 9.18 Bar-graph showing the standard deviation of fine frequency differences generated by the six algorithms on final test data.

The comparisons were all made against an interactive reference algorithm that makes use of the output from a laryngograph. The results shown are the average performance over 20 different women speakers, with about 15 seconds of speech per speaker.

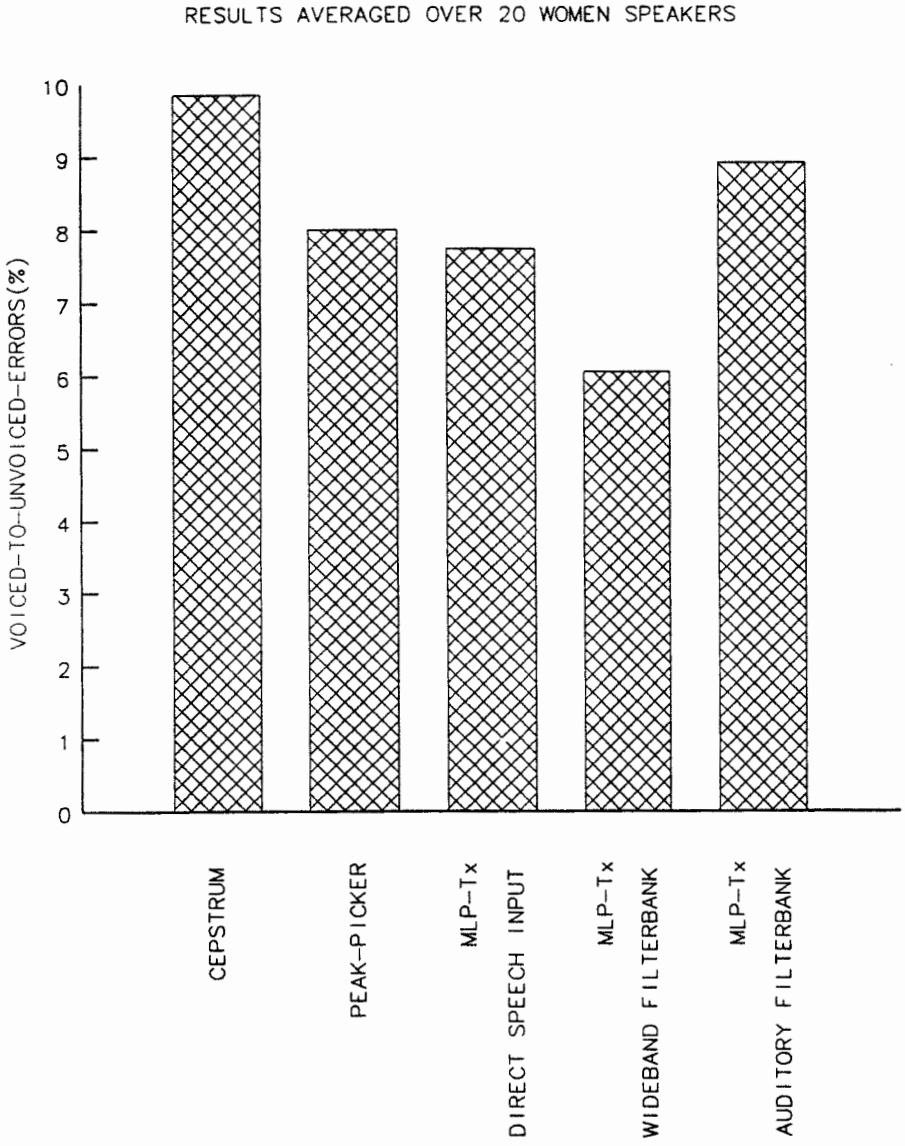


Figure 9.19 Bar-graph showing the voiced-to-unvoiced errors generated by the six algorithms on final test data.

The comparisons were all made against an interactive reference algorithm that makes use of the output from a laryngograph. The results shown are the average performance over 20 different women speakers, with about 15 seconds of speech per speaker.

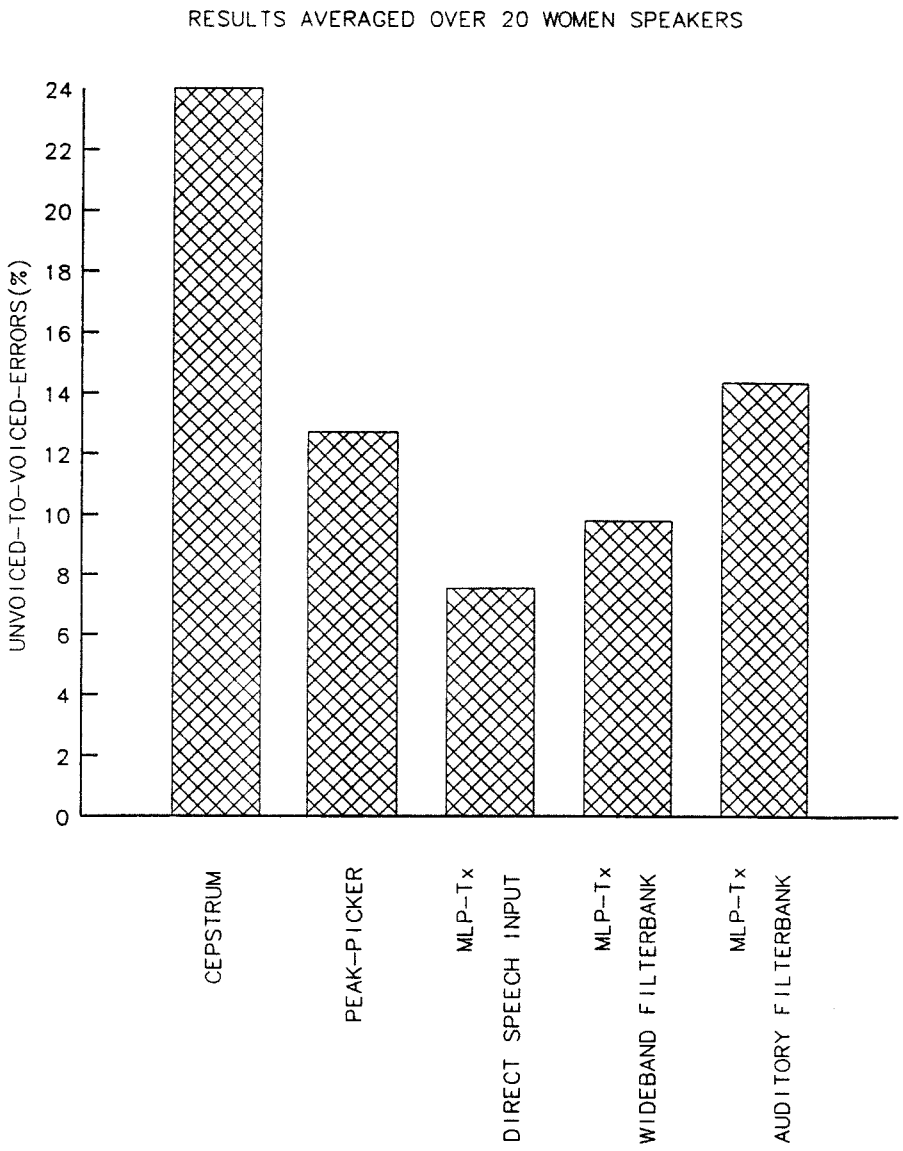


Figure 9.20 Bar-graph showing the unvoiced-to-voiced errors generated by the six algorithms on final test data.

The comparisons were all made against an interactive reference algorithm that makes use of the output from a laryngograph. The results shown are the average performance over 20 different women speakers, with about 15 seconds of speech per speaker. These results show much poorer performance of the cepstrum algorithm than of the MLP-Tx algorithms (and the direct speech version in particular).

CHAPTER 10: EXAMINATION OF MLP-Tx NETWORK FUNCTION

10.1 EXAMINING MLP OUTPUTS

10.1.1 Introduction

This chapter contains a series of observations concerning the operation of the MLP-Tx algorithm. The results given in this section are mainly for the direct speech MLP-Tx algorithm, rather than for the filterbank versions, because the former generates an output at a higher sampling rate. One consequence of this is that it makes visible certain phenomena (such as the occasional generation of double pulses at nasal transitions) that does not show up as clearly at the lower 2kHz output frame rate (although the effect is still present).

Firstly, the normal output from the MLP-Tx algorithm is examined. Conditions that lead to error in the estimation of fundamental period are then discussed and some corresponding MLP-Tx outputs for these conditions are shown. Investigation of error conditions is valuable because it provides the basis for future developments and improvements.

The patterns of weights are displayed both as time waveforms and as rectangles (in which the size and colour represents the weight). The power spectra of some of the weight time-functions are given.

Finally, the activity occurring at the hidden units is examined for normal operation and two error conditions.

10.1.2 Analysis of correct output from the MLP-Tx algorithm

Figure 10.1 shows a typical output from the direct speech MLP-Tx algorithm in trace D and the output from the reduced wideband filterbank MLP-Tx algorithm in trace E. Figure 10.2 shows the same on an expanded time-scale. It can be seen that the leading

edge peak frames of the pulses align with the (delay compensated) period markers derived from the laryngograph signal. Notice that the peak is well defined and the pulses typically take several frames (at 8kHz) to reach their peak value, and several frames to die away again. The pulse width usually corresponds to the number of de-emphasised samples around the period marker that was used during the selective emphasis training procedure. The pulse width is wider for the filterbank MLP-Tx.

Figure 10.3 shows the output from the direct MLP-Tx algorithm of irregular speech. It can be seen that the excitation points are individually detected by the algorithm.

10.1.3 Analysis of failures of the MLP-Tx algorithm

Figure 10.4 shows a case when the response of the direct speech MLP-Tx algorithm is poor. The speech is for the token "named" from a female speaker. This example was selected from the evaluation test data because it illustrates the main sources of error that currently occur with the MLP-Tx algorithm (although is must be noted that this output is **not** typical of the operation of the MLP-Tx algorithm).

It can be seen that the centre region of the utterance gives rise to double-pulse output from the MLP. In addition, at the transition between the vowel and the final nasal region (occurring at about 1885ms on the time-scale shown) there is a brief reduction of the height of the output pulses from the MLP. From observation of all the evaluation data for women speakers, it has been found that the double-pulsing phenomenon is the main source of chirp errors generated by the algorithm on women speakers, whilst the abrupt reduction in pulse height at nasal transitions is the main source of drop error in the algorithm on both men and women speakers.

Double-pulse generation at transitions

A close-up of the initial nasal-to-vowel region is shown in figure 10.5, which makes visible the occurrence of the MLP output pulses with respect to the (delay compensated) laryngograph period marker. It can be seen that during a nasal, there is a tendency for

the MLP output to be generated with a time-lag of the order of a millisecond behind the reference period marker. As the transition proceeds into the vowel, another pulse appears that has the correct timing relationship with respect to the reference marker. In this situation, it is likely there is as much evidence of the nasal as the oral vowel that follows. Finally, after the vowel is fully established, the second delayed pulse disappears, leaving only the appropriately located markers.

False pulses in cycle for male speech

There are additional chirp errors in the case of speech from men, which can arise at secondary peaks in the speech cycles. This is illustrated in figure 10.6 and a close-up of the phenomenon is given in figure 10.7. This constitutes the main source of error in the MLP-Tx algorithm when it has been trained on male + female, or just female speech.

Reduction in pulse height at transitions

Figure 10.8 shows a close-up of a transition between a vowel and a nasal that exhibits the reduction in height of the MLP output at the transition. It is believed that this occurs **because** of the miss-alignment in timing around nasal-vowel and vowel-nasal transitions. Since the MLP-Tx algorithm tries to generate output pulses in the **wrong** place, during the training phase such outputs are **suppressed**, resulting in a reluctance for the algorithm to respond to at all in such circumstances.

Secondary MLP post-processing

The effect of the form of the output from the MLP-Tx algorithm using the secondary-window post-processing technique (discussed in chapter 9) is illustrated in figure 10.9. It can be seen that the post-processing has the effect of suppressing unwanted secondary pulses. In addition it reduces pulse-width and gives a more uniform pulse height of the legitimate pulses. Figure 10.10 shows the MLP output waveforms over an expanded time-scale.

10.2.1 Weight patterns represented as Hinton diagrams

To give some indication of how the trained MLPs in the MLP-Tx algorithm are organised, the weights are presented in two different formats. Firstly, the weights are shown graphically in terms of images in which the values of the weights are represented such that weight magnitude is represented by the size of a square. Positive weights have black solid squares, and negative weights have unfilled (white) squares. Secondly, the weights are shown as time-waveforms.

For the original MLP-Tx algorithm (trained on five male speakers), the patterns for the weights in layer 1-2 are given in the figures 10.11 and 10.12. The weights for layer 2-3 and layers 3-4 are given in figure 10.13.

Observation of the lower layer weights in figures 10.11 and 10.12 shows that the magnitude of the positive excitatory weights is greatest at the centre of the window and just ahead in time of the centre. This suggests that most of the activity in the detection process takes place around the centre of the window, where the weights are of the largest magnitude. This is not surprising, since the central region in the window is the zone that always lines up with the point in the speech data that contains the excitation point whenever a pulse to signify an excitation marker pulse is generated.

10.2.2 Weight patterns represented as time-waveforms

Figure 10.14 shows the weights in two direct speech MLP-Tx algorithms, each using 161 inputs and no hidden units. The weights represented in trace A were generated by training on the four women and three men whereas the weights shown in trace B were generated by training only on women. These traces correspond to the time function that is correlated against the speech input to generate the output period markers. Both of these waveforms show a large positive to negative transition around the centre of the window, although the negative peak is one sample later for the network trained on men

and women..

10.2.3 Power spectra of the weight time-waveforms

Normally one would characterise linear filters in terms of their frequency response (both in phase and magnitude). The power spectrum for the weight time-function, shown trace A in figure 10.15, is given in figure 10.16. It can be seen that the magnitude frequency response associated with the weighs is complicated and non-monotonic, although there is an attenuation of 20dB at the lowest frequencies. Figure 10.17 shows the first layer weight time-functions for the case of a direct speech MLP-Tx algorithm using a MLP with 161 input and one layer of 10 hidden units (again trained on the four women and three men). Figure 10.18 shows the first layer weight time-functions for a direct speech MLP-Tx algorithm using a MLP with 161 input and one layer of 5 hidden units, again trained on the four women and three men. The corresponding power spectra for these weight time-waveforms are shown in figures 10.19 to 10.23. It can be seen that the different first layer weights have considerably different time-functions and power spectra. In all cases, the attenuation due to the weights is not that great, compared to what might be expected from, for example, a normal low-pass filter. That is, the typical peak response to minimum response is only about 20-30dB. In all cases, the power spectra are complicated functions.

10.2.4 Internal activations

To examine the action of the internal nodes in an MLP during its operation to detect the period markers, the MLP-Tx algorithm was run in recognition mode and the activities at the internal nodes were recorded.

Normal operation

The node activity for the original wideband filterbank MLP-Tx algorithm (trained on five male speakers) is shown in figure 10.24. The speech pressure waveform (20dB SNR) in shown trace A, the reference period markers in trace B, the output from the

MLP-Tx algorithm in trace C and the internal activations in layer 2 for all the nodes in that layer in traces E to M. Certain observations can be made from these traces. Firstly it appears that there are some traces with no activity, for example F and H. To be conclusive about this would require observations of these node activities over all possible input conditions. Another feature of the traces concerns the timing relationships between the traces. It can be seen that some traces go high just before the overall output, such as G. Some traces go high just after the output, such as D and E. Some traces go low as the output goes high, such as L. Some traces go high just before and after the output goes high, trace J. In addition, certain traces appear to be time-shifted versions of others.

Some of the traces in layer 3 (shown figure 10.25) exhibit more similarity with each other than do those in layer 2, possibly suggesting that there were more nodes in this layer that necessary.

Internal activation during double pulse error condition

Figure 10.26 shows the internal activations from the nodes in a direct-speech MLP-Tx algorithm with 161 inputs and 5 hidden units (trained on the four women and three men) in the training data. Figure 10.27 shows the same example with an expanded time-scale. These nodes correspond to the weight functions shown in figure 10.18 and power spectra shown in figures 11.19 to 11.23. These internal node activity traces are given for an example where double pulses are generated at the transition between a nasal and a vowel. The different nodes in the hidden layer generate quite different outputs. During the transition, nodes 0,1 and 2 generate almost the same shaped pulses. In fact the output from node 2 (trace F) generates an output that corresponds well to the time when firm vocal fold closure is made. However, nodes 3 and 4 change and generate more regular and narrower pulses as the transition progresses from the nasal into the more temporally complex vowel.

The more slowly varying traces from nodes 0, 1 and 2 have corresponding power spectra (figure 11.19, 11.20 & 11.21) that give greater emphasis to the lower frequencies.

Conversely, the more rapidly varying traces from nodes 3 and 4 have associated power spectra that attenuate low frequencies.

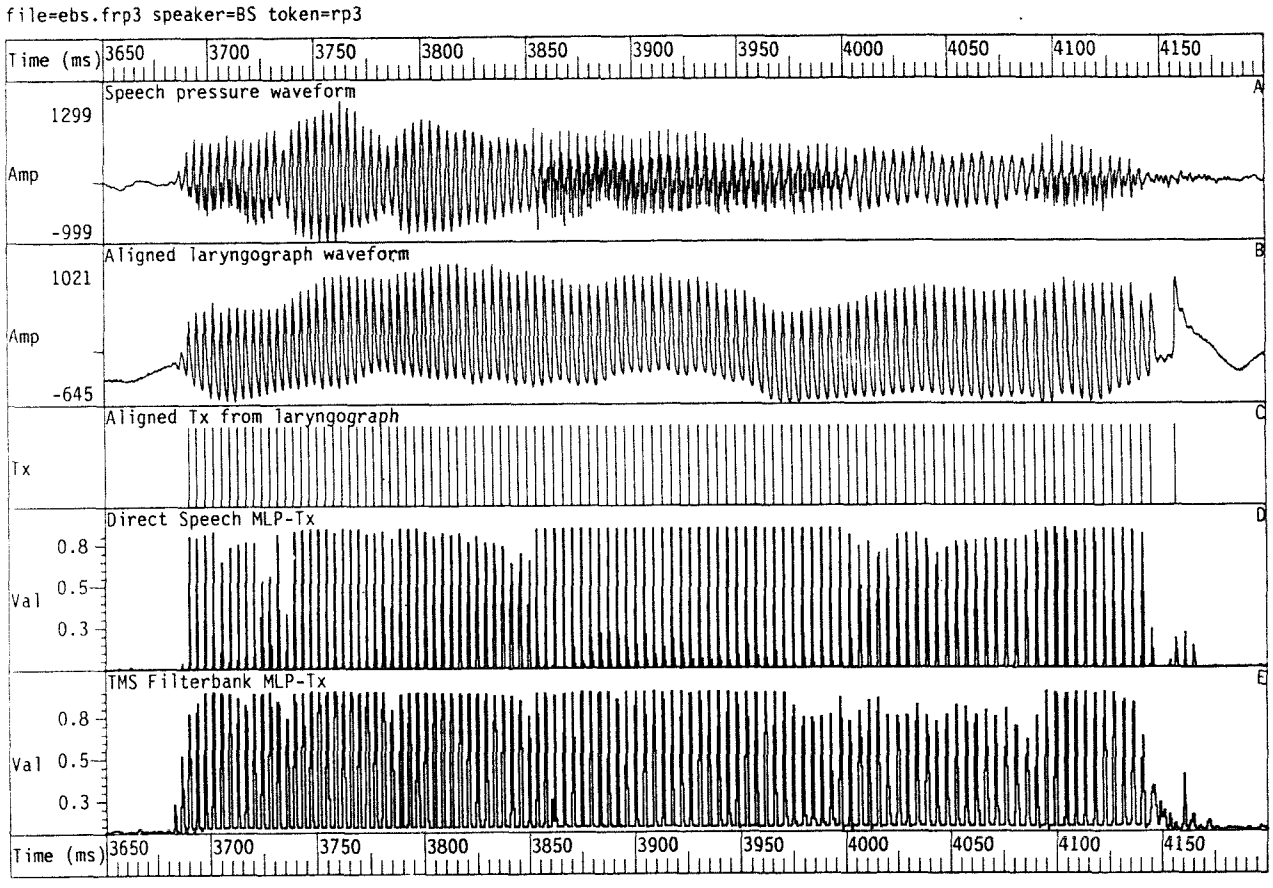


Figure 10.1 Diagram showing normal response of direct speech MLP-Tx algorithm. Trace A shows the speech pressure waveform, trace B shows the corresponding laryngograph waveform and trace C shows the marker derived from it. The output from the direct speech MLP-Tx and the reduced filterbank MLP-Tx algorithms are shown in traces D and E respectively. It can be seen that there is good agreement with the period markers and the MLP outputs. The speech is the utterance "when an man..." from a women.

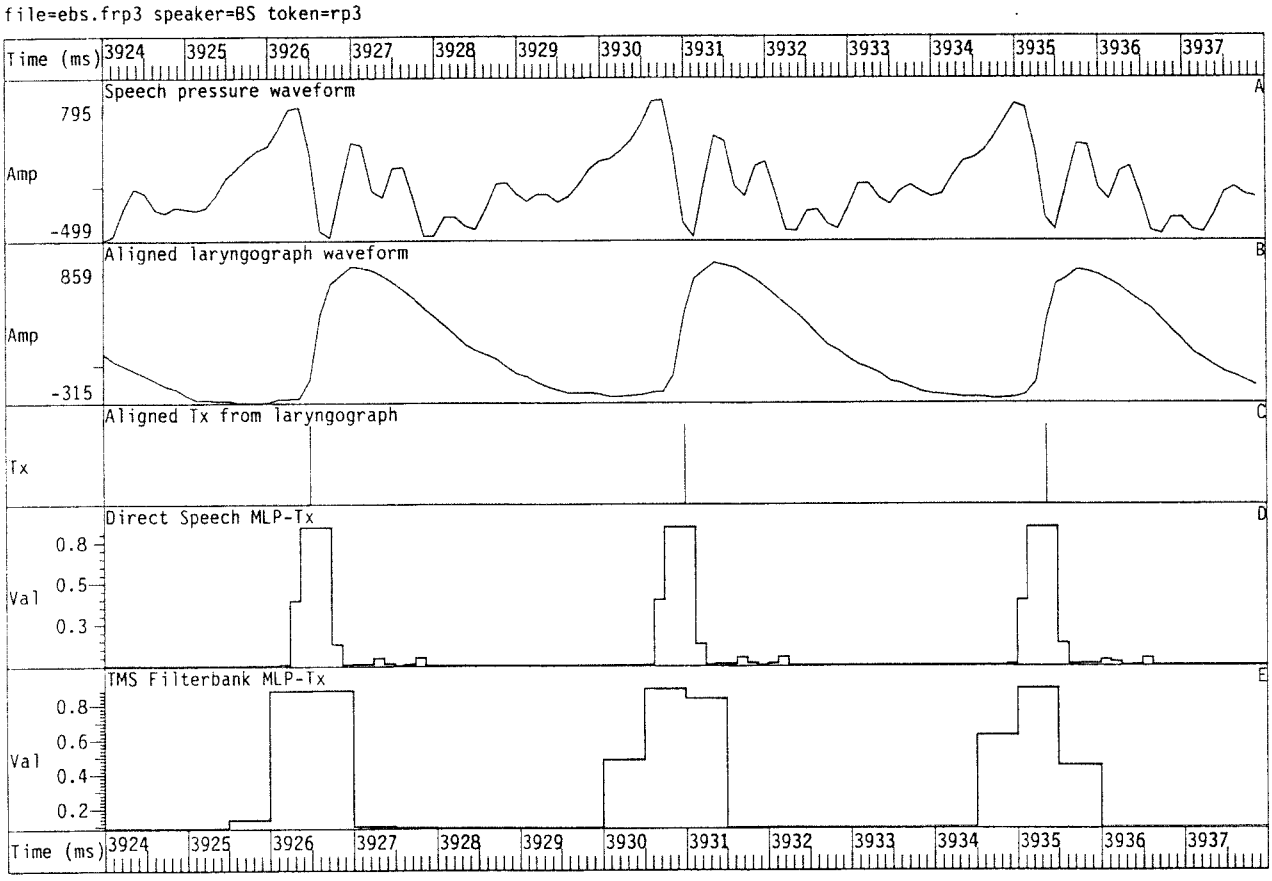


Figure 10.2 Diagram showing same as figure 10.1, but with an expanded time-scale. It can be seen that each MLP-Tx output pulse exhibits only one maximum and that its time location coincides with the (aligned) period marker derived from the laryngograph. The direct speech MLP-Tx algorithm operating at an 8kHz frame-rate gives better time resolution than the reduced wideband filterbank MLP-Tx algorithm. The speech is for the vowel in "man".

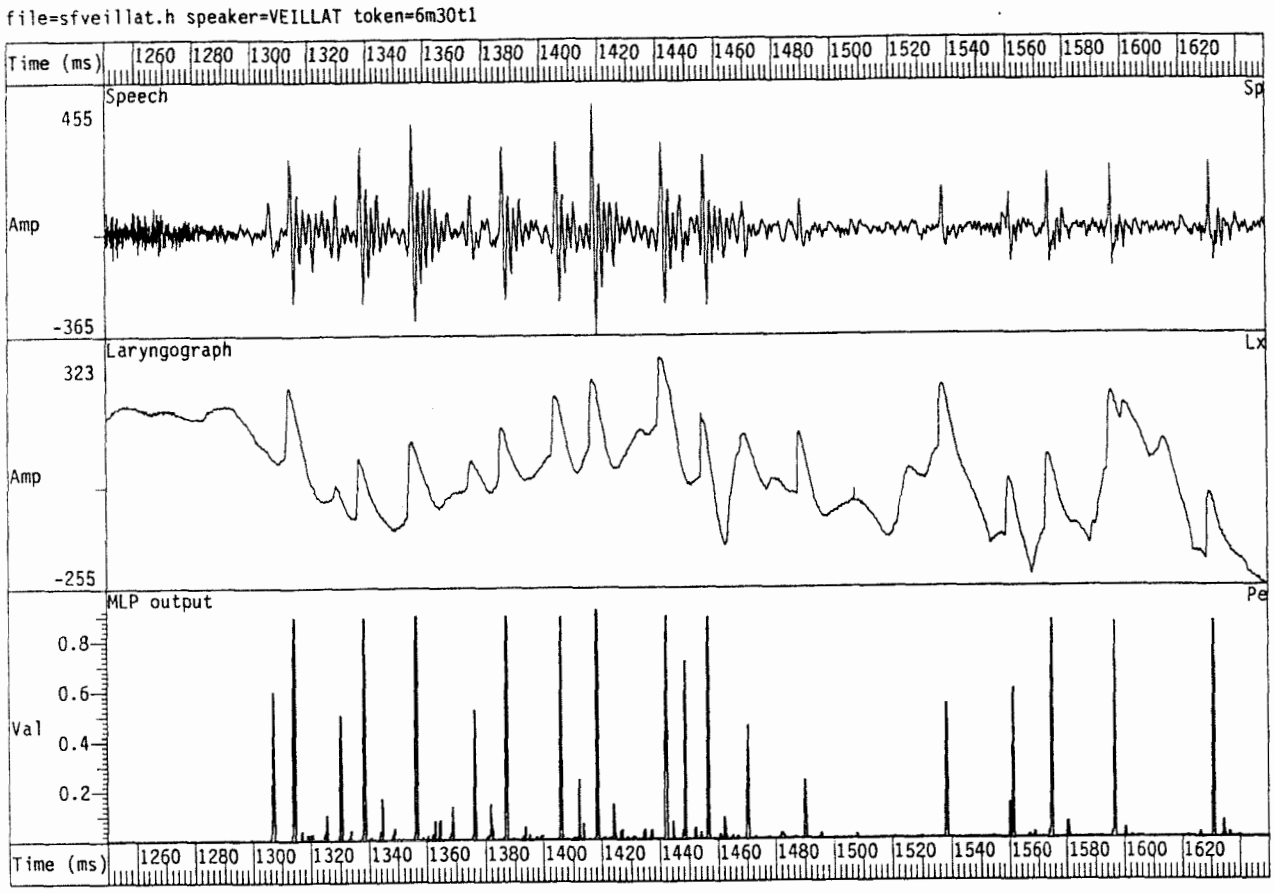


Figure 10.3 Diagram showing response of direct speech MLP-Tx algorithm to irregular speech.

The output from the MLP-Tx algorithm shows good agreement with the laryngograph output. The speech is from a women.

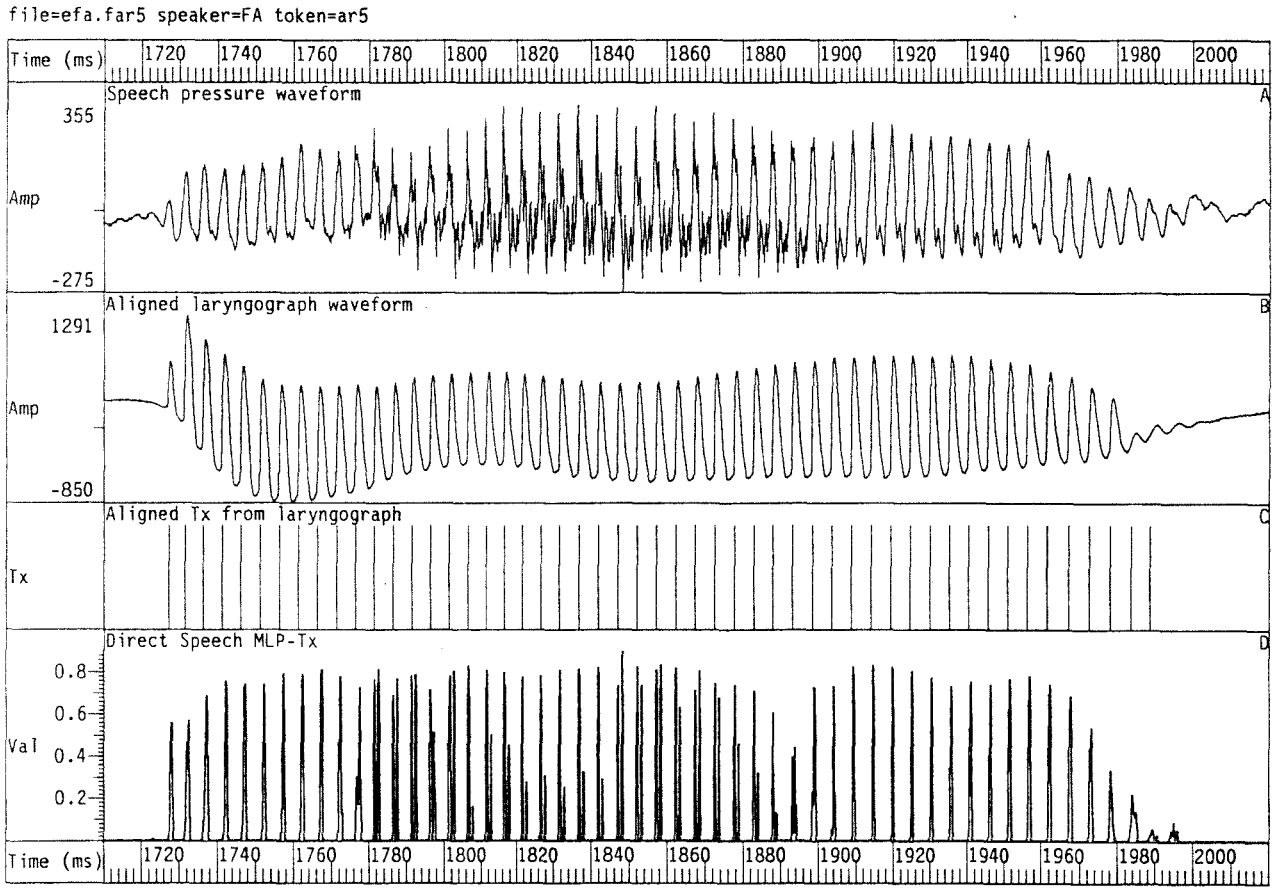


Figure 10.4 Diagram showing strongest erroneous double-pulse response of direct speech MLP-Tx algorithm at nasal-vowel transitions.

The utterance is "named" from a women speaker. Trace A shows the speech pressure waveform, trace B shows the corresponding laryngograph waveform and trace C shows the marker derived from it. The output from the MLP-Tx algorithm is shown in trace D. It can be seen that there double pulses are most strongly generated by the MLP at the transitions with the initial and final nasal regions, although the effect persists throughout the vowel.

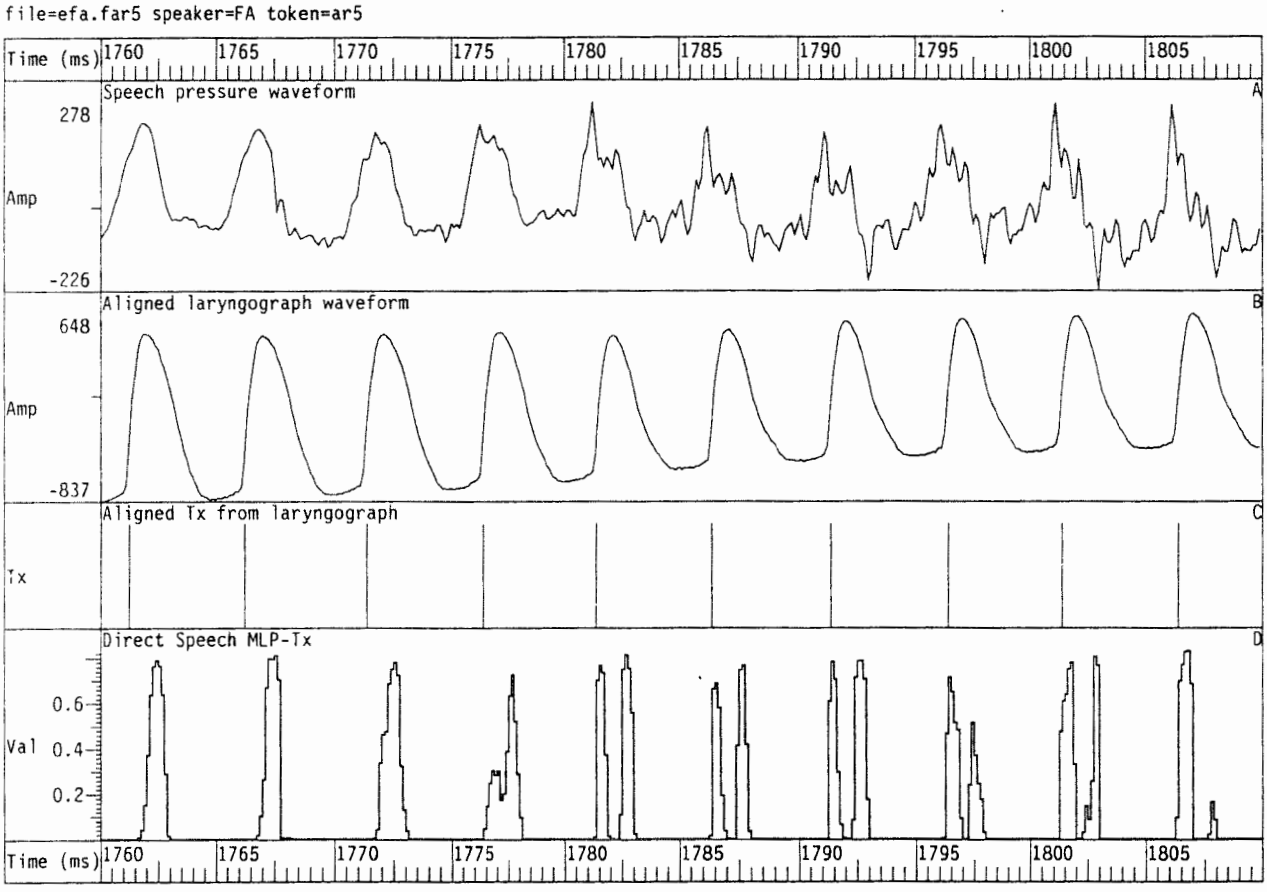


Figure 10.5 Diagram showing same as figure 10.4, but with an expanded time-scale over initial nasal .."na.." transition region.

It can be seen that during the nasal, the MLP-Tx output pulses exhibit only one maxima but their time locations are ahead of the (aligned) period markers derived from the laryngograph. During the transition, two pulses are generated, the first of which does align with the reference from the laryngograph. Proceeding into the vowel, the secondary delayed pulse dies away.

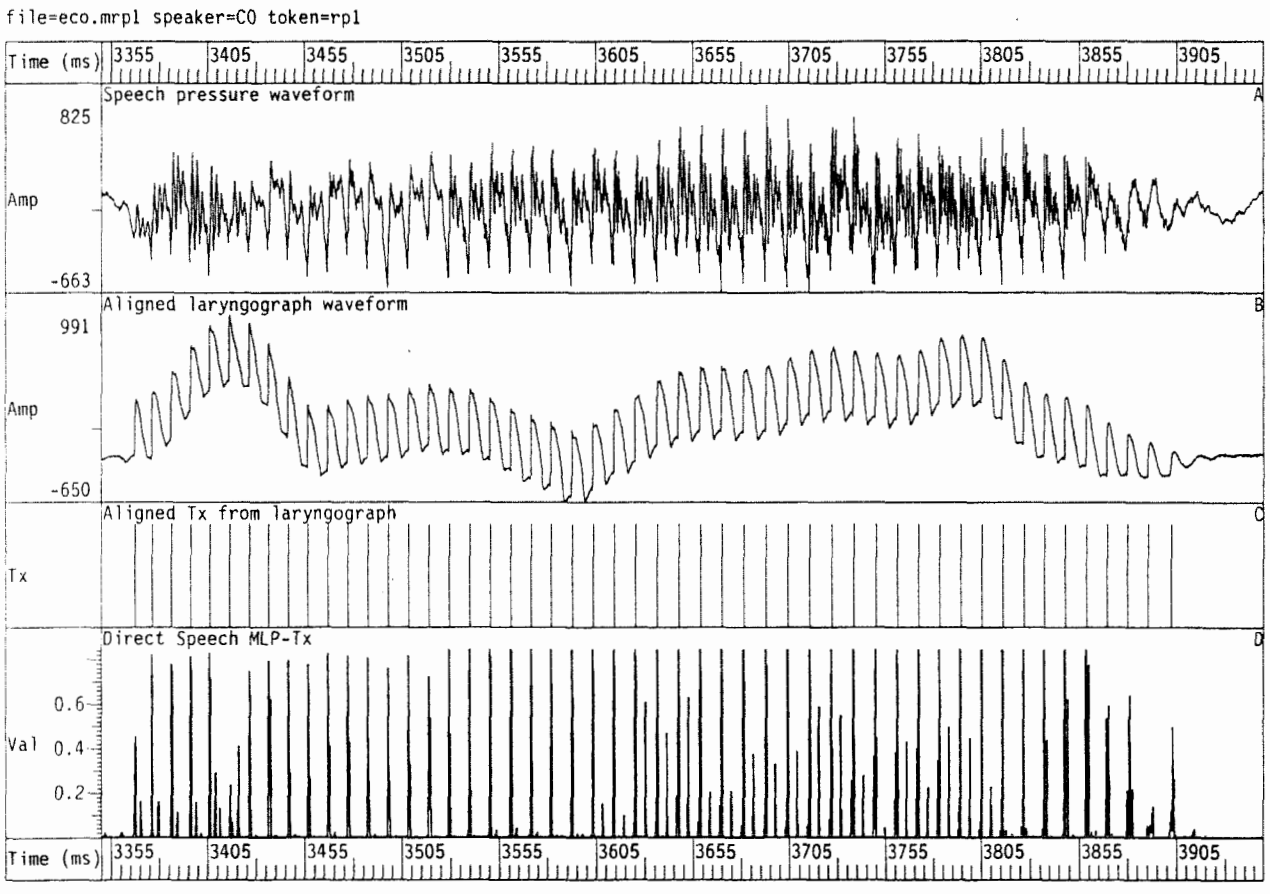


Figure 10.6 Diagram showing erroneous generation of unwanted period marker pulse from the direct speech MLP-Tx algorithm on male speech.

Trace A shows the speech pressure waveform, trace B shows the corresponding laryngograph waveform and trace C shows the marker derived from it. The output from the MLP-Tx algorithm is shown in trace D. It can be seen that erroneous output pulses are sometimes generated between the legitimate ones. The speech is the utterance "...in the air...".

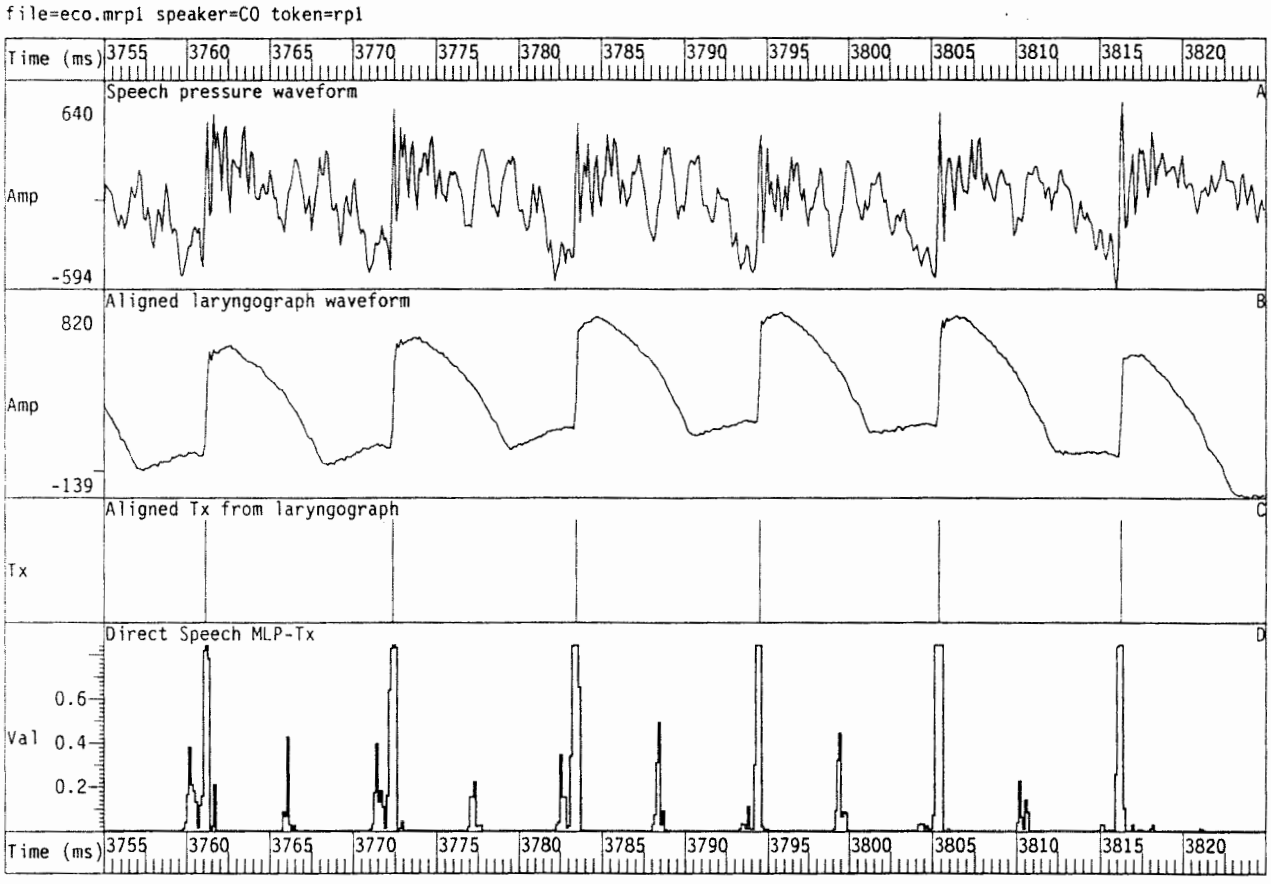


Figure 10.7 Same as figure 10.6 with expanded time-scale.

The erroneous pulses correspond to the secondary peaks in the speech cycles. The speech is the utterance "a" from "...in the air...".

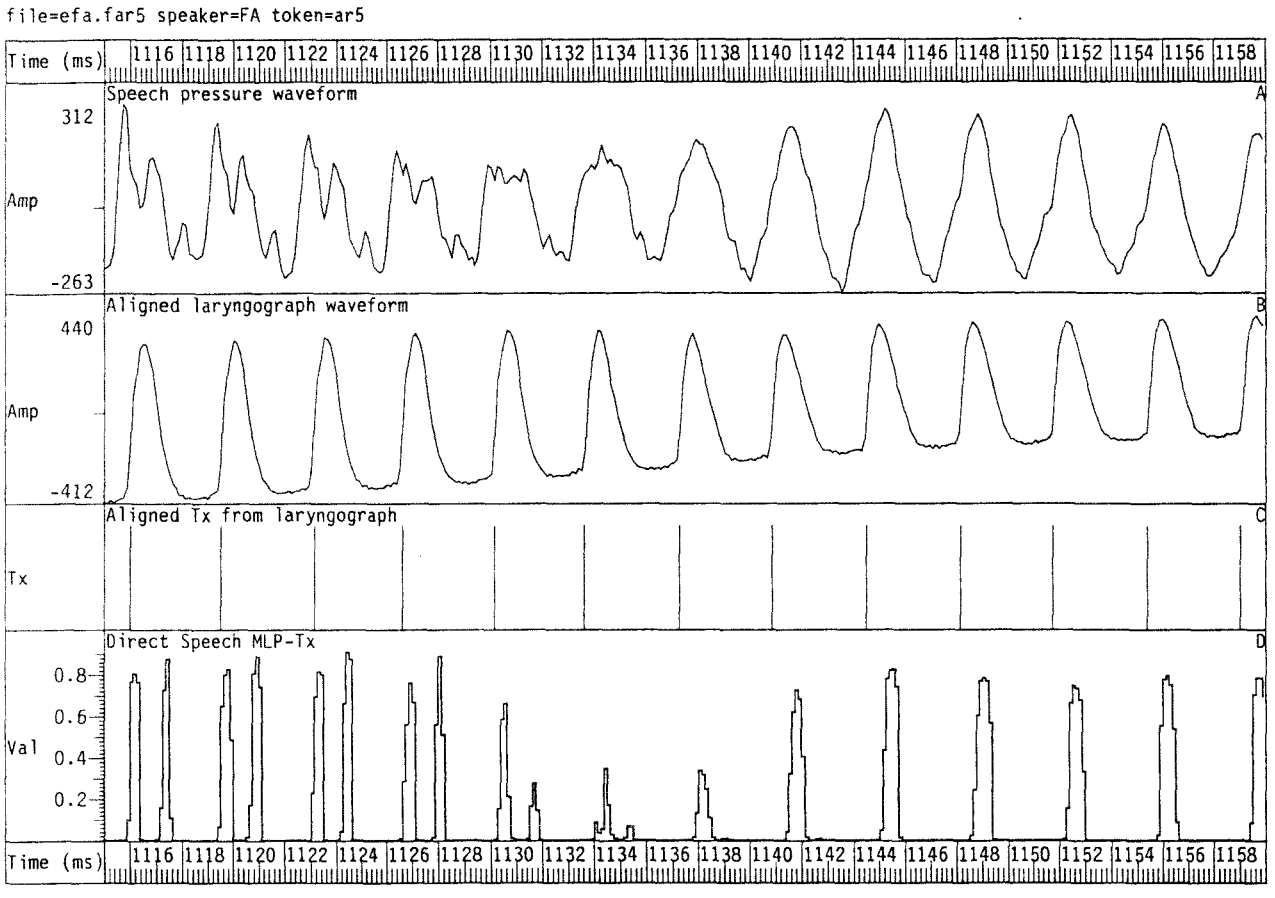


Figure 10.8 Erroneous reduction on MLP-Tx pulse height at nasal-vowel transition. This reduction in MLP-Tx output pulse height that sometimes occurs under such conditions. Trace A shows the speech pressure waveform, trace B shows the corresponding laryngograph waveform and trace C shows the marker derived from it. The output from the MLP-Tx algorithm is shown in trace D. The utterance is the section "..in.." from "..kindly.." from a female subject.

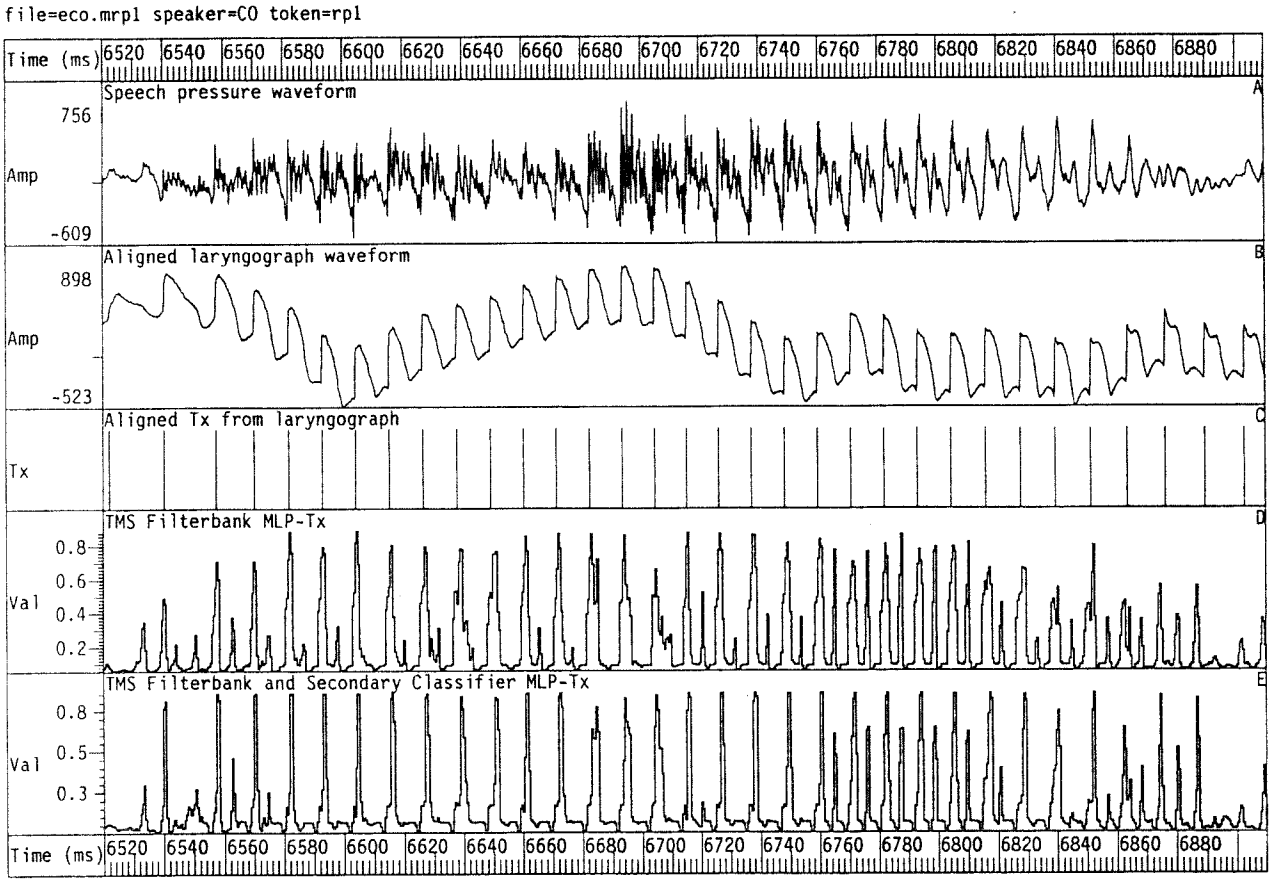


Figure 10.9 Effect of using a secondary MLP-Tx algorithm trained on the output from a primary (that is, normal) MLP-Tx algorithm.

Trace A shows the speech pressure waveform, trace B shows the corresponding laryngograph waveform and trace C shows the marker derived from it. The output from the primary reduced filterbank and secondary MLP-Tx algorithms are shown in traces D and E respectively. It can be seen that there is a tendency to suppress secondary pulses, reduce the pulse-widths and produce pulses with more uniform heights. The speech is the utterance "...rainbow..." from a male speaker.

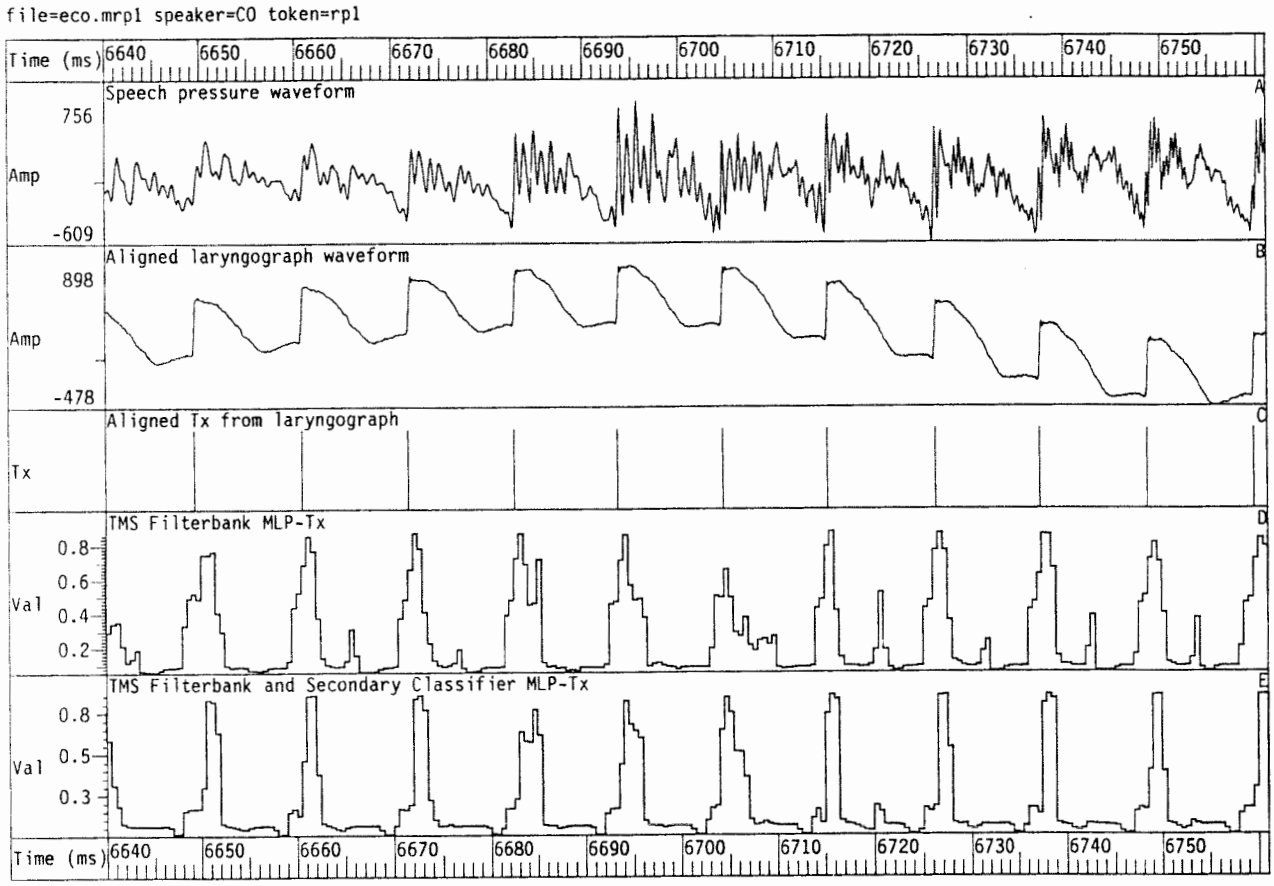


Figure 10.10 Same as figure 10.9 with expanded time-scale.
The speech is the utterance "ai" from "..rainbow..".

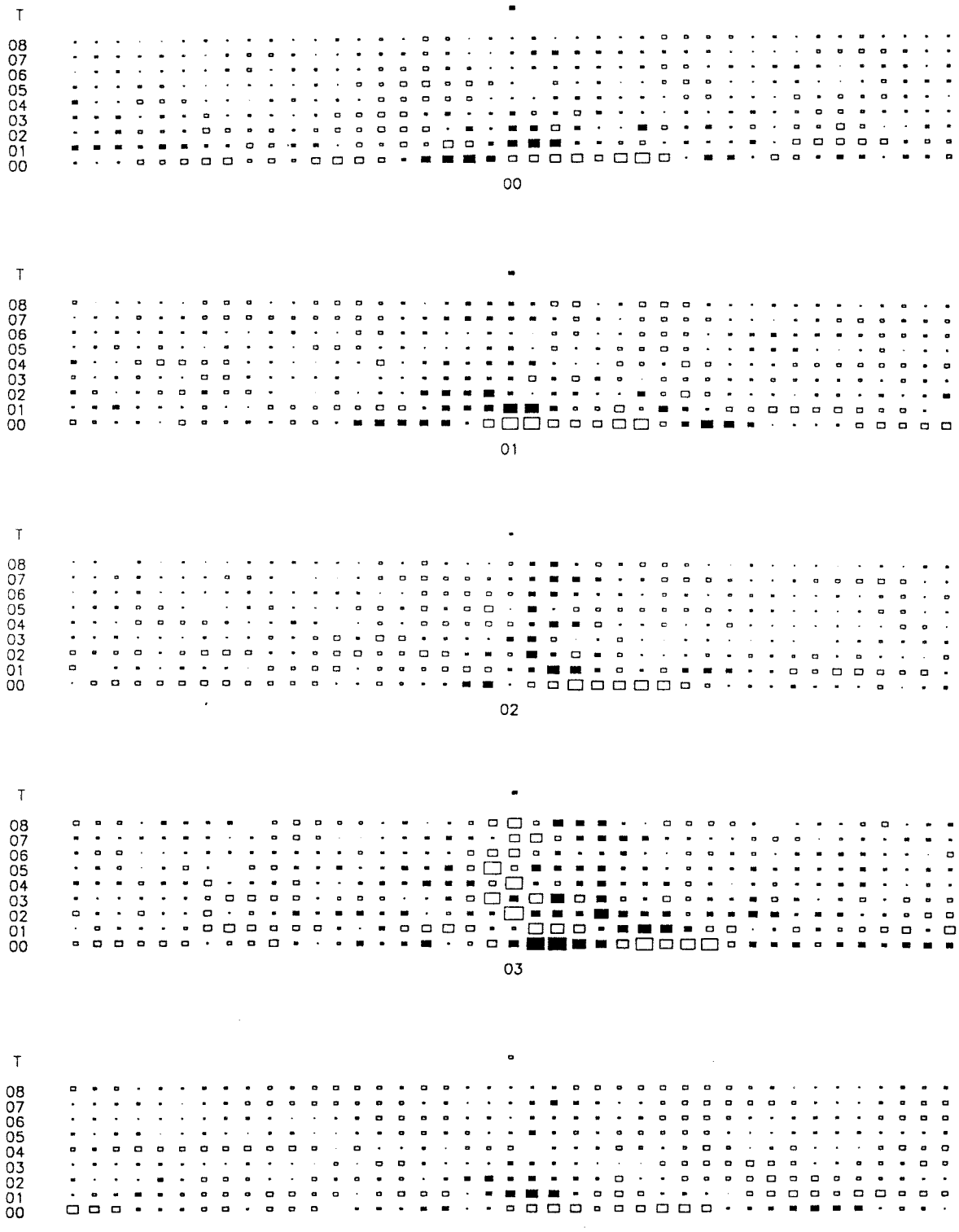


Figure 10.11 Weight patterns for the original wideband (9-channel) filterbank MLP-Tx algorithm.

The size of the square represents the magnitude of a weight, and positive and negative weights are denoted by black and white squares respectively. This network has 369 inputs, two layers of 10 hidden units and one output. The vertical scale represents the 9-input frequency channels (0-8), whereas the horizontal scale represents the 41 time-frames. The window is symmetrical with the current frame represented in the centre. The threshold weight is shown above channel 08. Nodes 0-4 are shown in this diagram for layers 1-2.

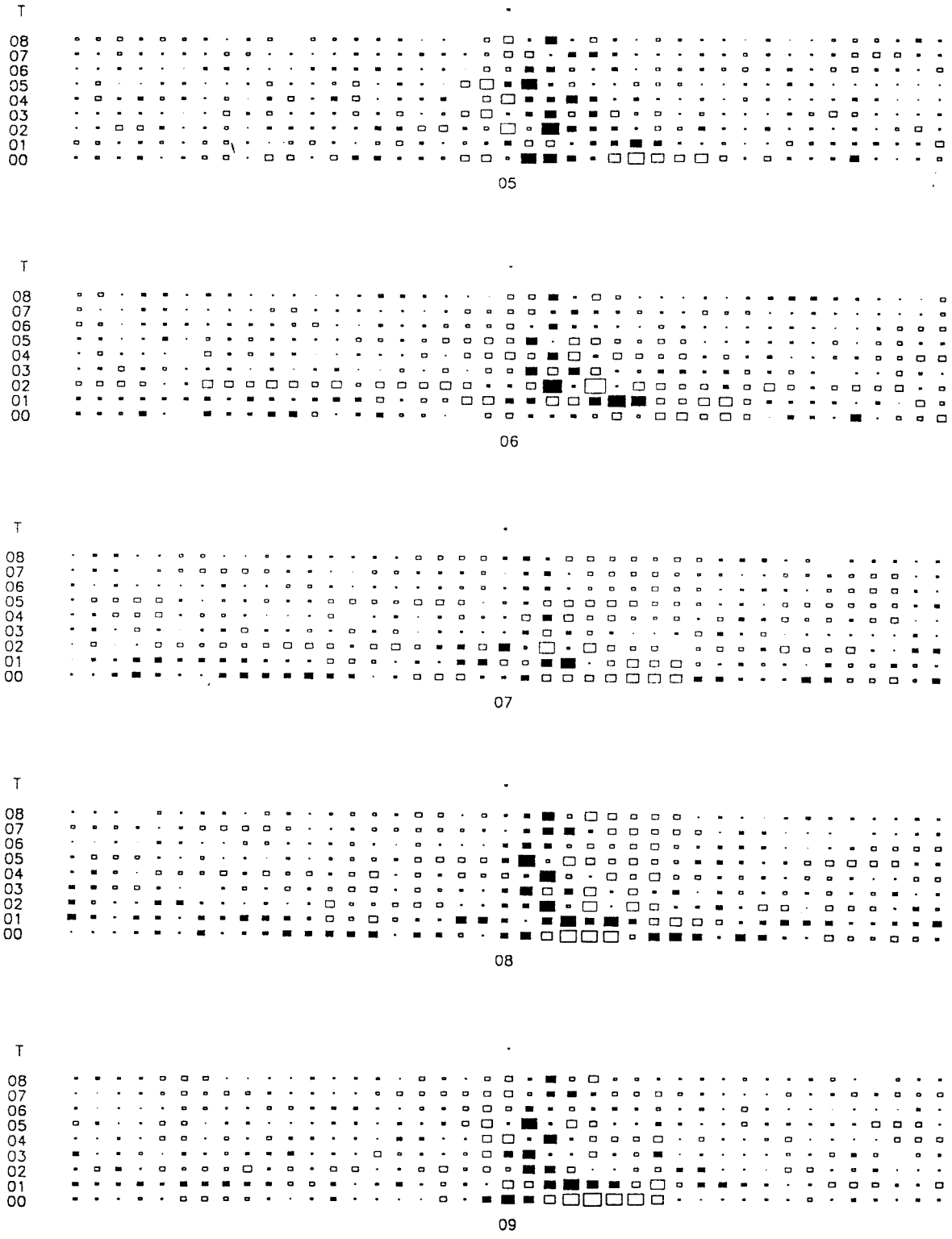


Figure 10.12 Weight diagrams as in figure 10.11, but this time for nodes 5-9 in layers 1-2.

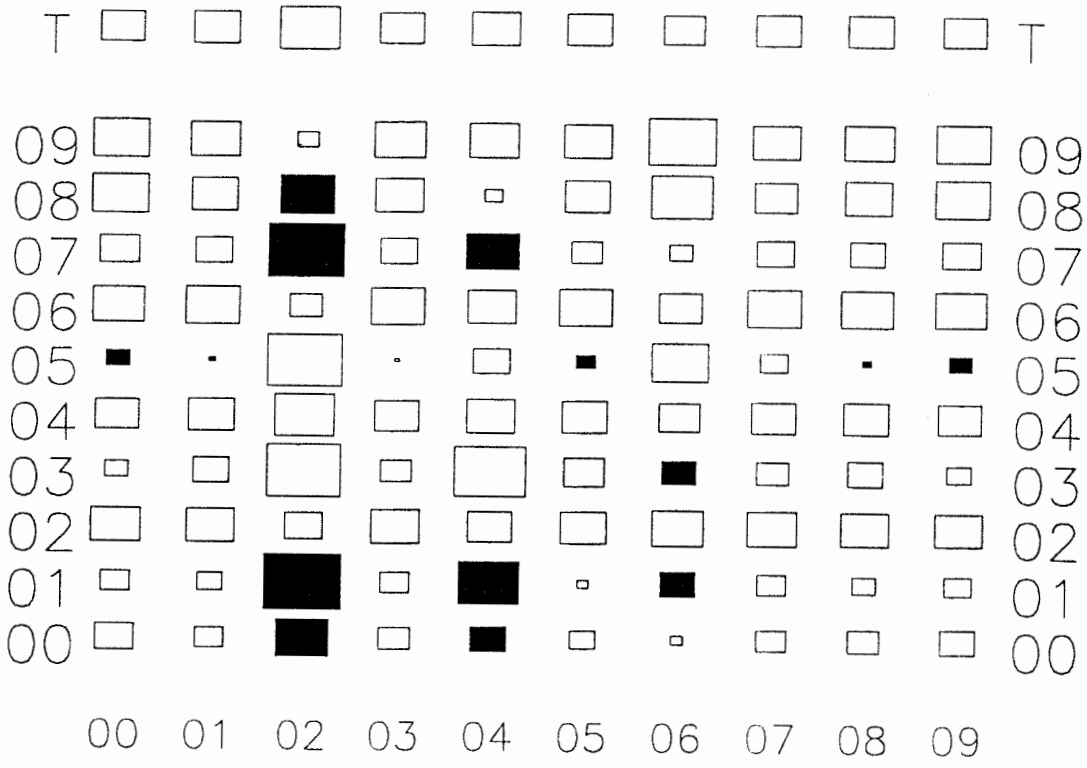


Figure 10.13 Weight diagrams as in figure 10.11, but this time for nodes 0-9 in layers 2-3.

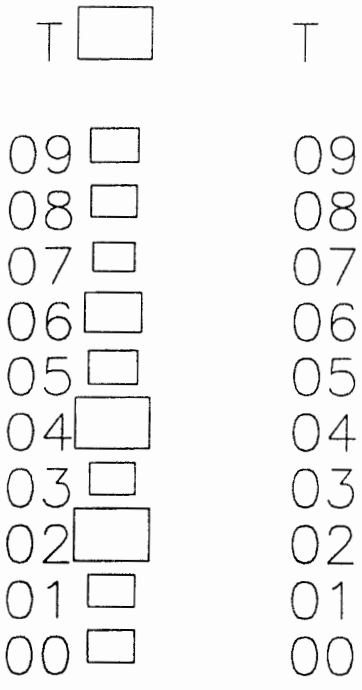


Figure 10.14 Weight diagrams as in figure 10.11, but this time for output node in layers 3-4.

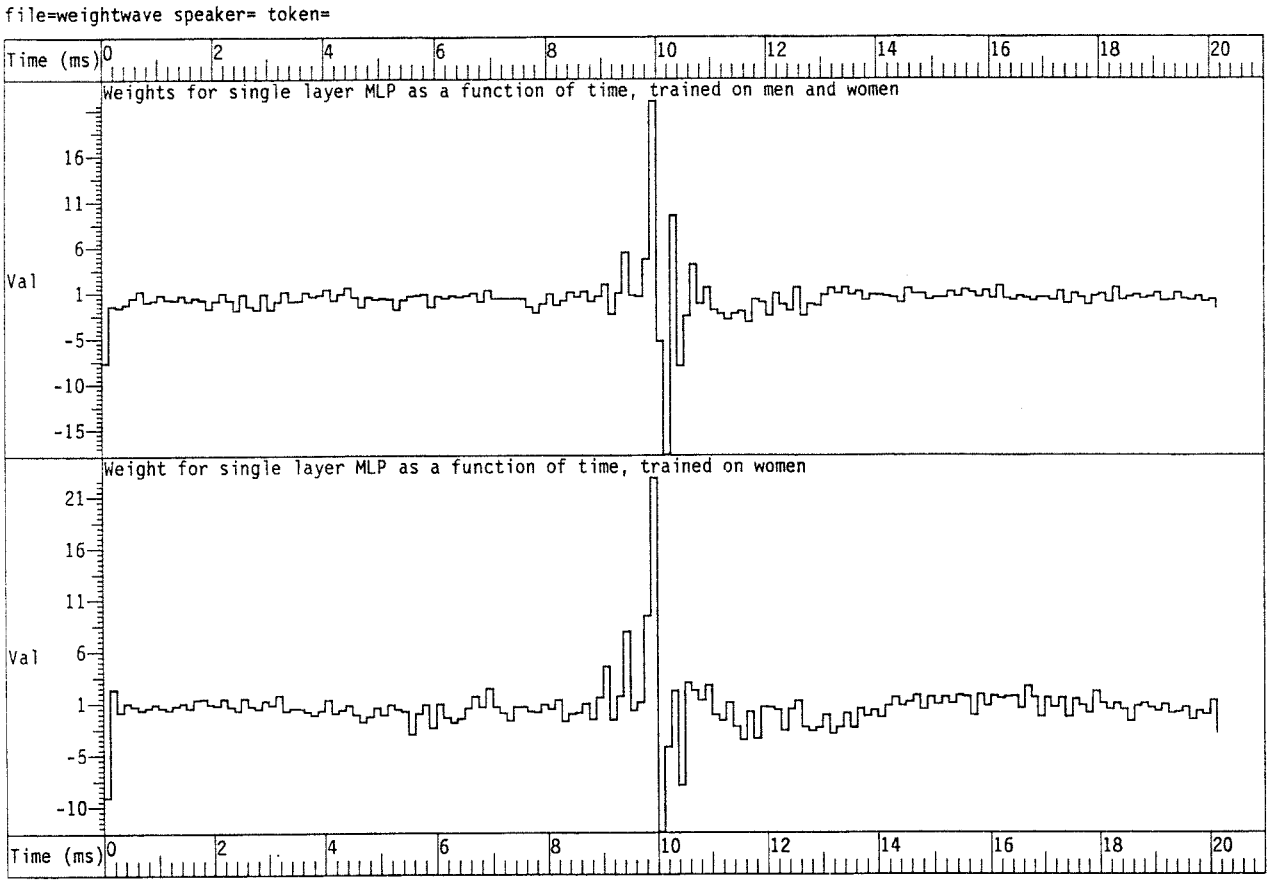


Figure 10.15 Weights in direct speech MLP-Tx algorithms, with no hidden units, represented as time-waveforms.

The network had 161 inputs and 1 output. The weights are represented as the time-waveform that is correlated with the input speech and then passed via the sigmoid non-linear function to generate the output. Trace A shows the weight waveform for a network trained on both men and women. Trace B shows a network trained only on women.

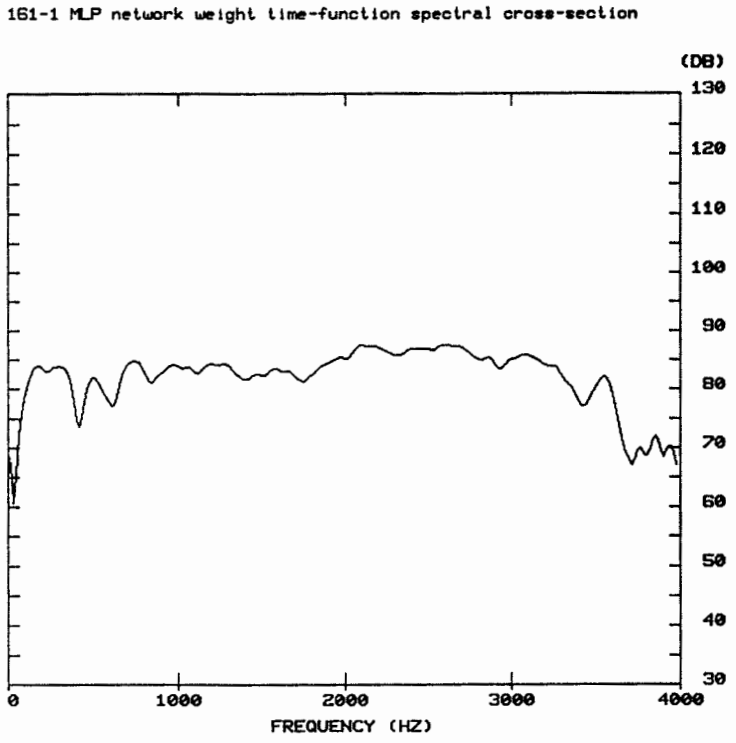


Figure 10.16 Power spectrum corresponding to the weight time-waveform shown in trace A in figure 10.15.

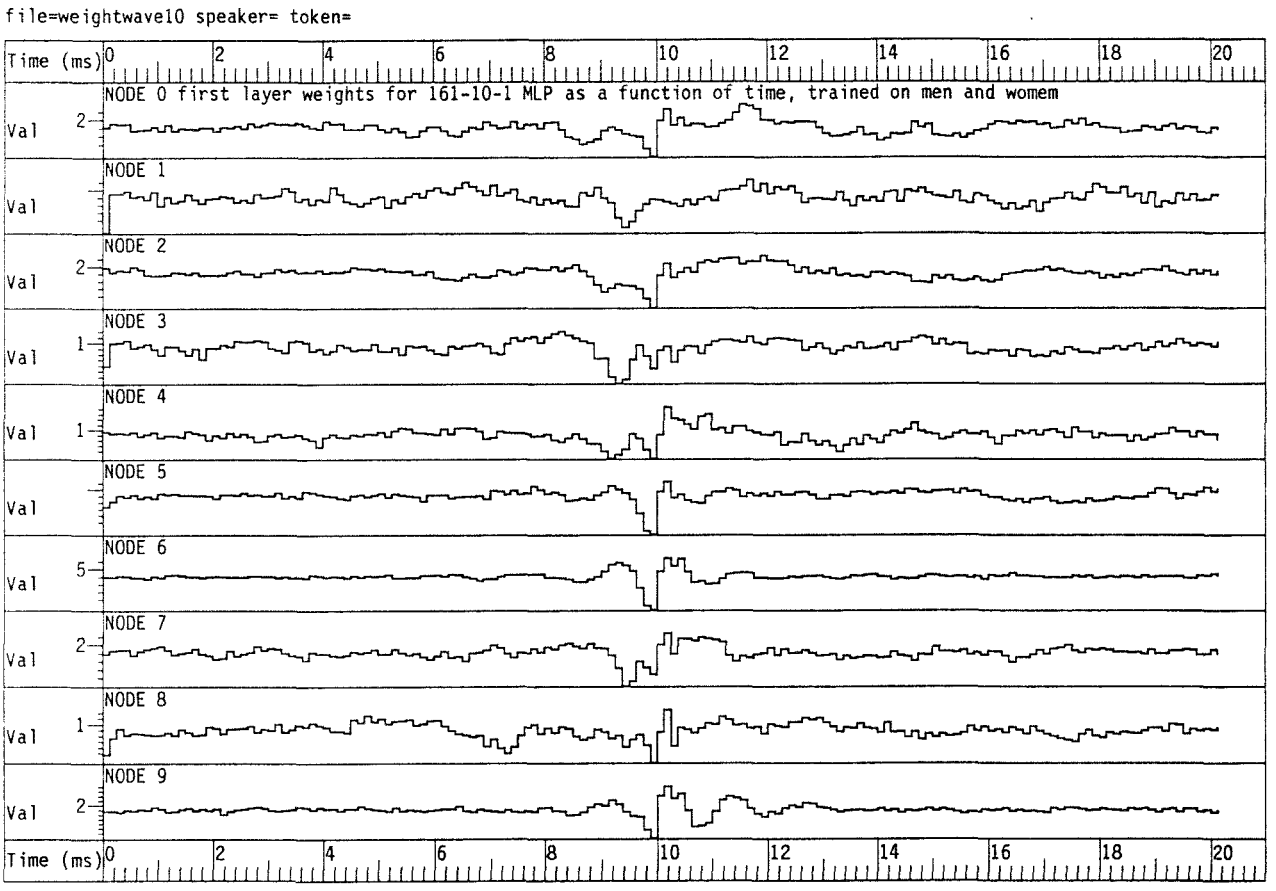


Figure 10.17 First layer weights for MLPs with 10 hidden units in direct speech MLP-Tx, represented as time-waveforms.

The network had 161 inputs, one layer of 10 hidden units and 1 output. The weights are represented as a time-waveform. The network was trained on men and women.

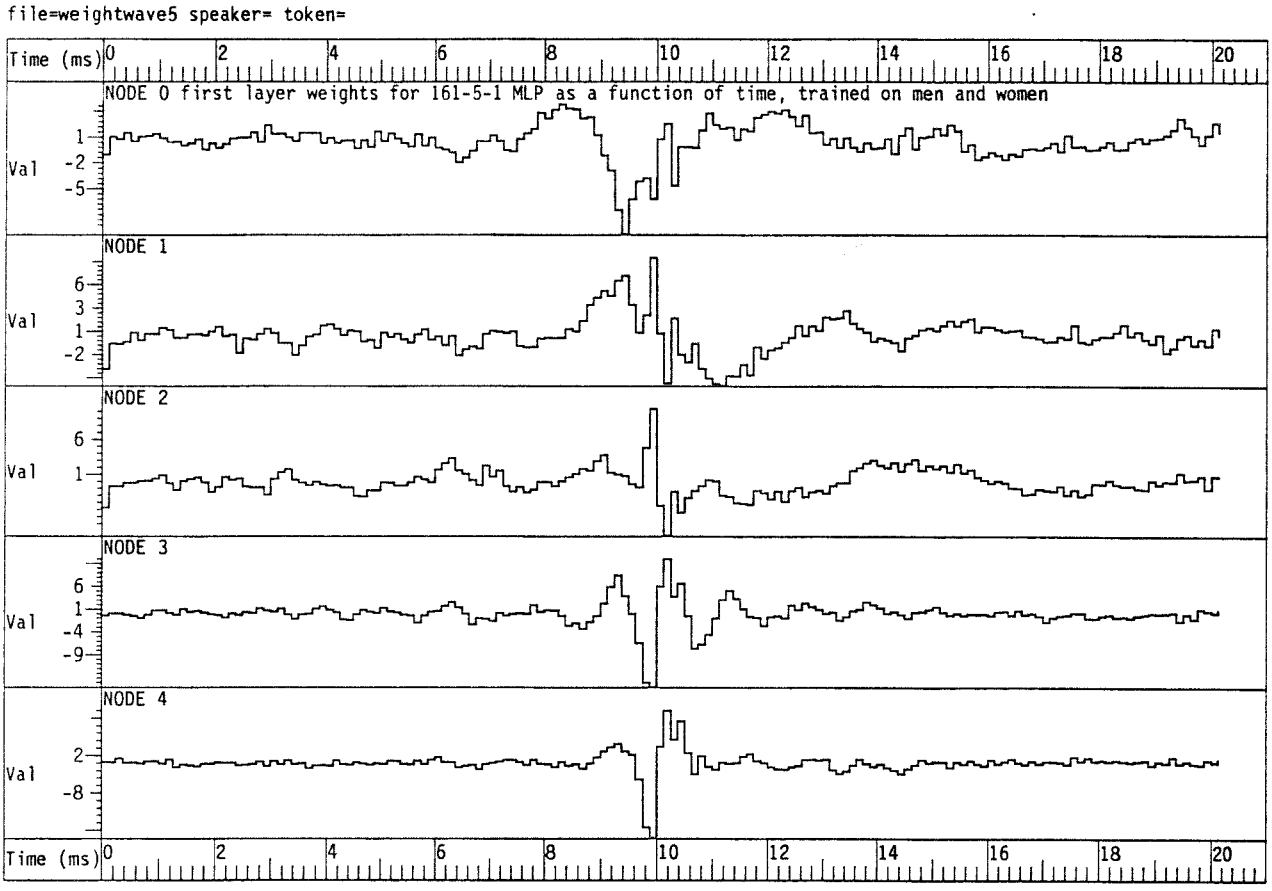


Figure 10.18 First layer weights for MLPs with 5 hidden units in direct speech MLP-Tx, represented as time-waveforms.

The network had 161 inputs, one layer of 5 hidden units and 1 output. The weights are represented as a time-waveform. The network was trained on men and women.

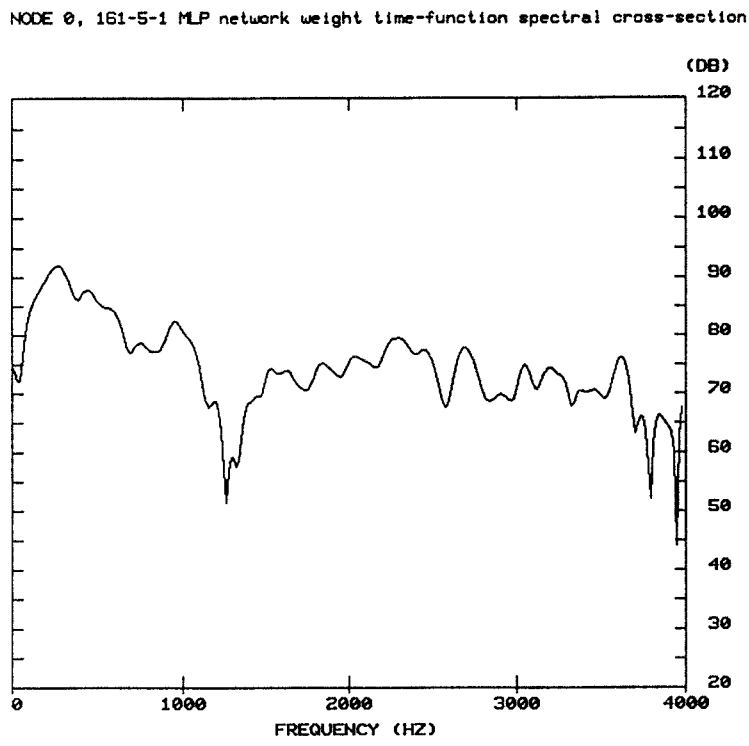


Figure 10.19 The power spectrum corresponding to the weight time-waveform for hidden node 0 shown in trace A in figure 10.18.

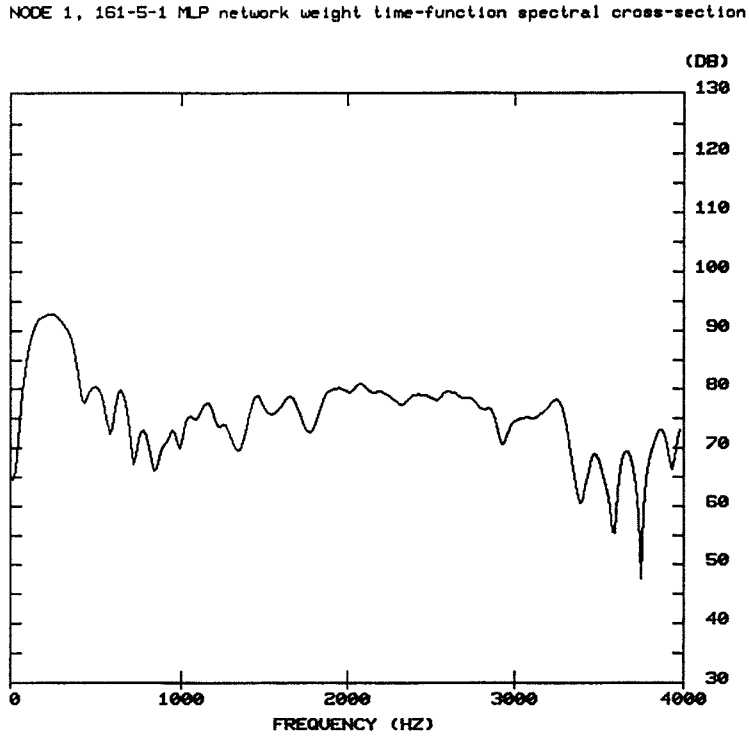


Figure 10.20 The power spectrum corresponding to the weight time-waveform for hidden node 1 shown in trace B in figure 10.18.

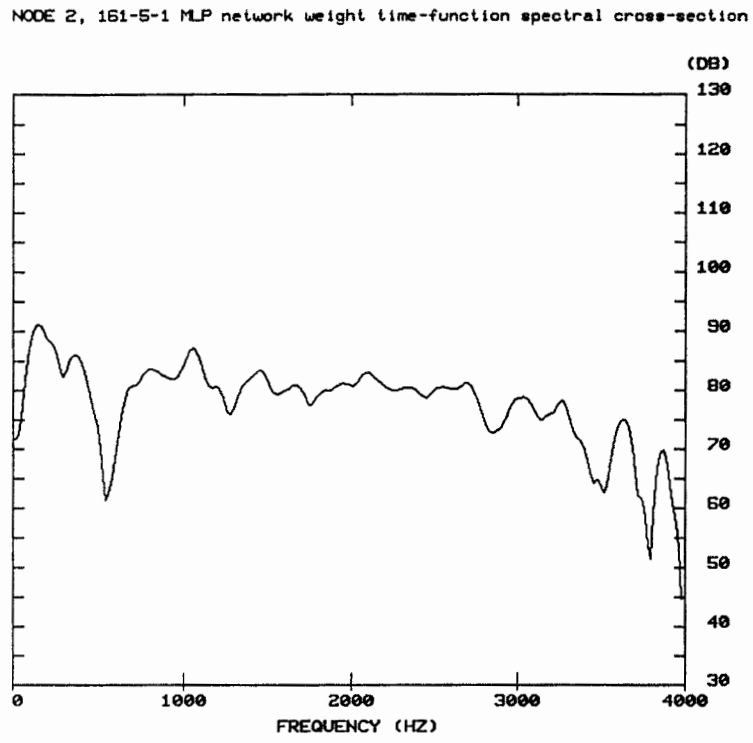


Figure 10.21 The power spectrum corresponding to the weight time-waveform for hidden node 2 shown in trace C in figure 10.18.

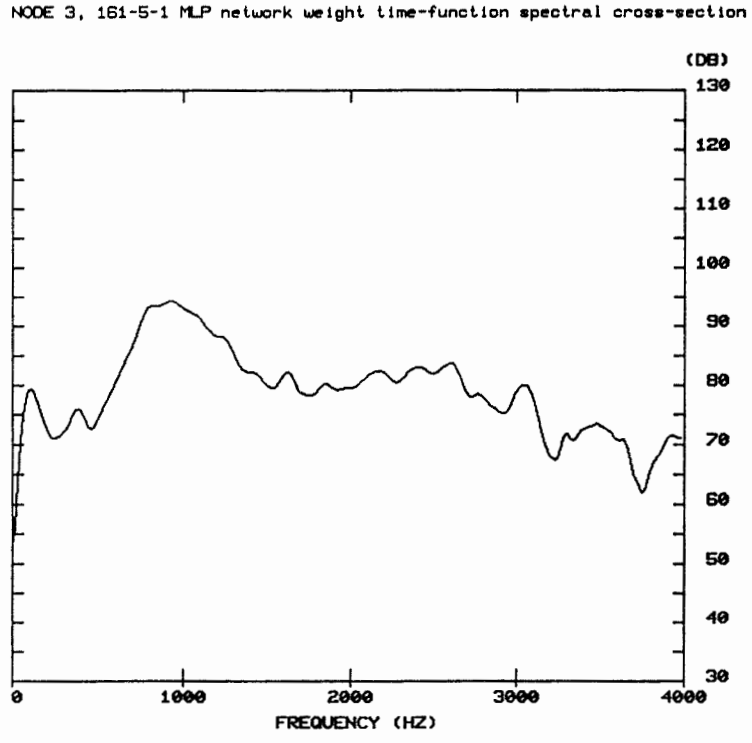


Figure 10.22 The power spectrum corresponding to the weight time-waveform for hidden node 3 shown in trace D in figure 10.18.

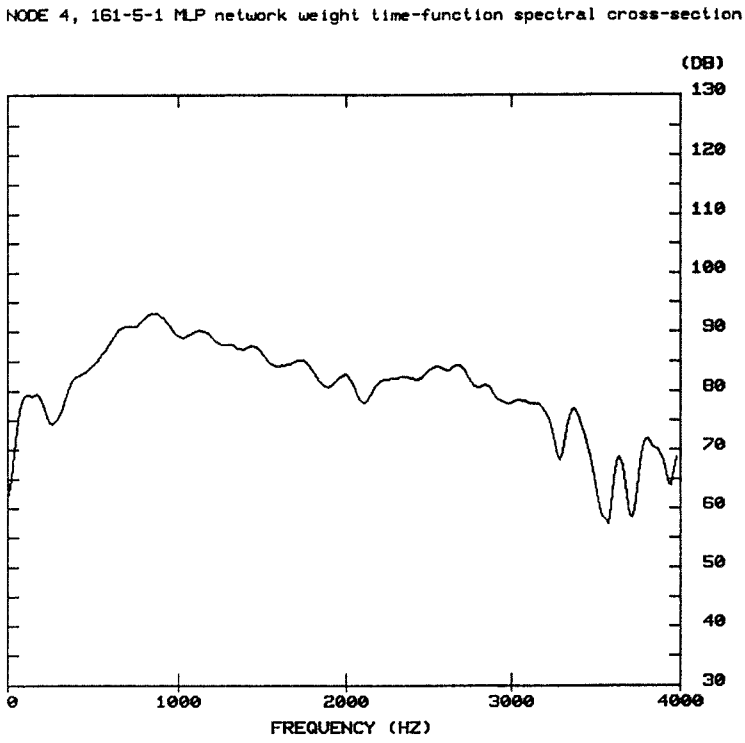


Figure 10.23 The power spectrum corresponding to the weight time-waveform for hidden node 4 shown in trace E in figure 10.18.

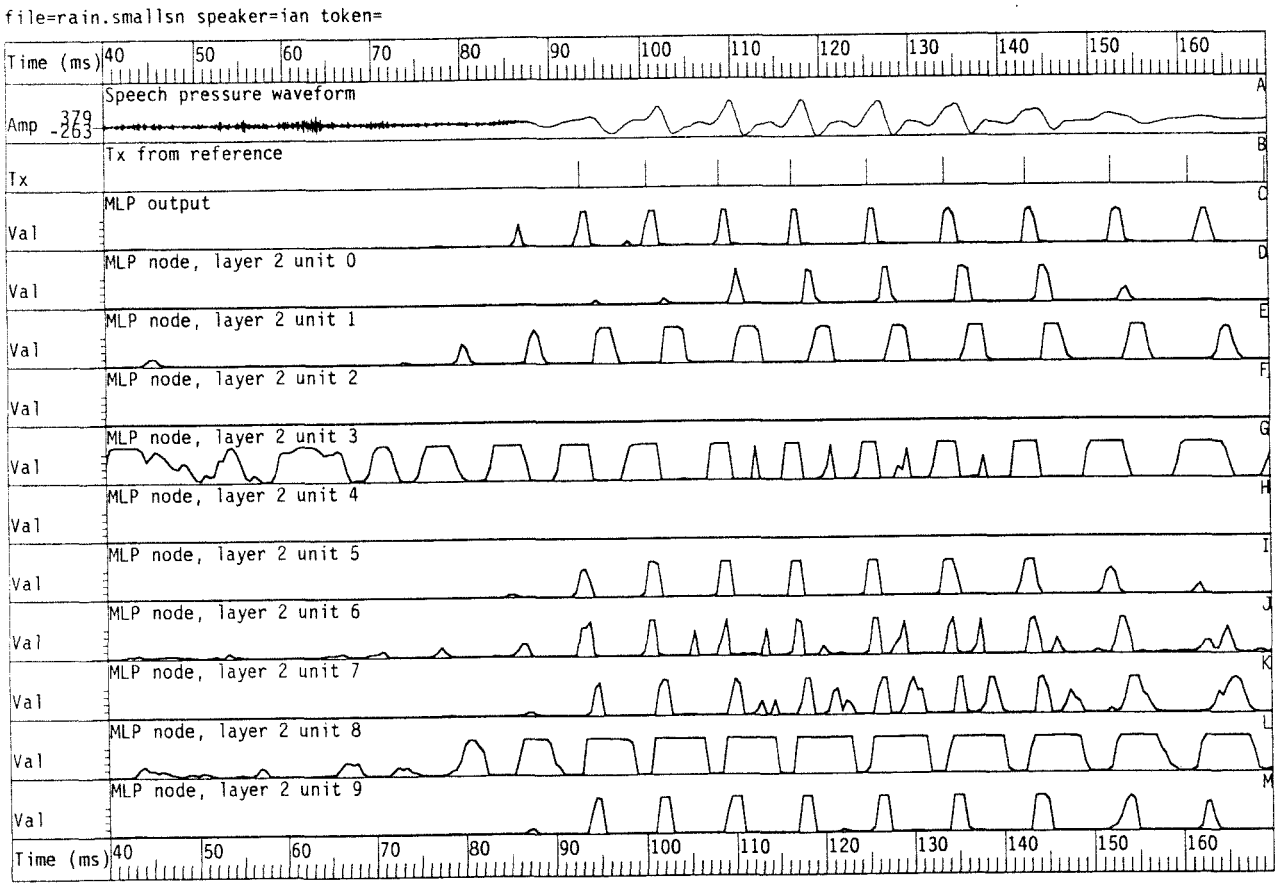


Figure 10.24 The output from the first layer nodes in the original wideband filterbank (9-channel) MLP-Tx algorithm.

This system used an MLP with the configuration 369-10-10-1 configuration. The speech is the first part of the utterance "seem" from a male speaker, and is shown in trace A. Trace B shows the laryngograph waveform and the period marker derived from it are shown in trace C. Traces D to M show the MLP node outputs.

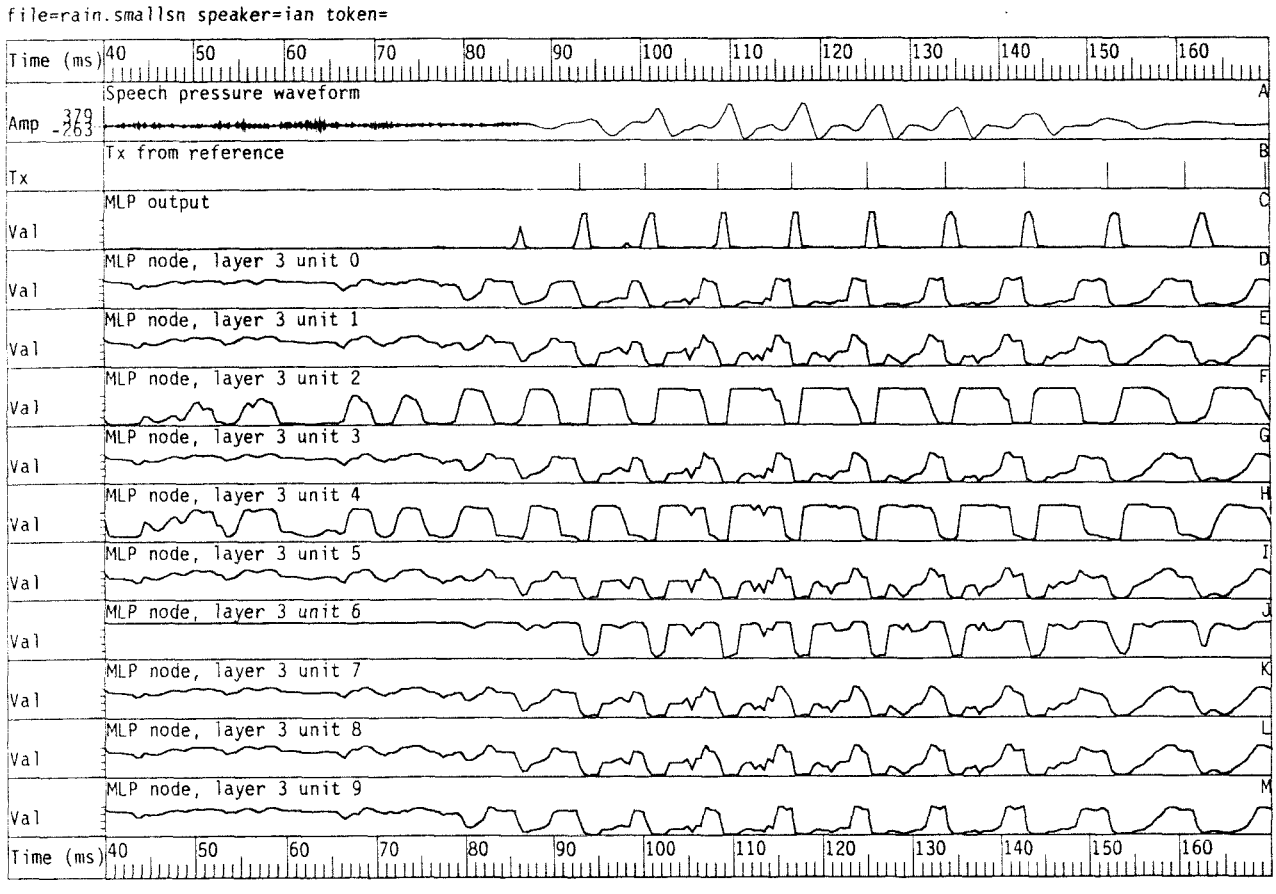


Figure 10.25 The output from the second layer nodes in the original wideband filterbank (9-channel) MLP-Tx algorithm.

This system used an MLP with the configuration 369-10-10-1 configuration. The speech is the first part of the utterance "seem", and is shown in trace A. Trace B shows the laryngograph waveform and the period marker derived from it are shown in trace C. Traces D to M show the MLP node outputs.

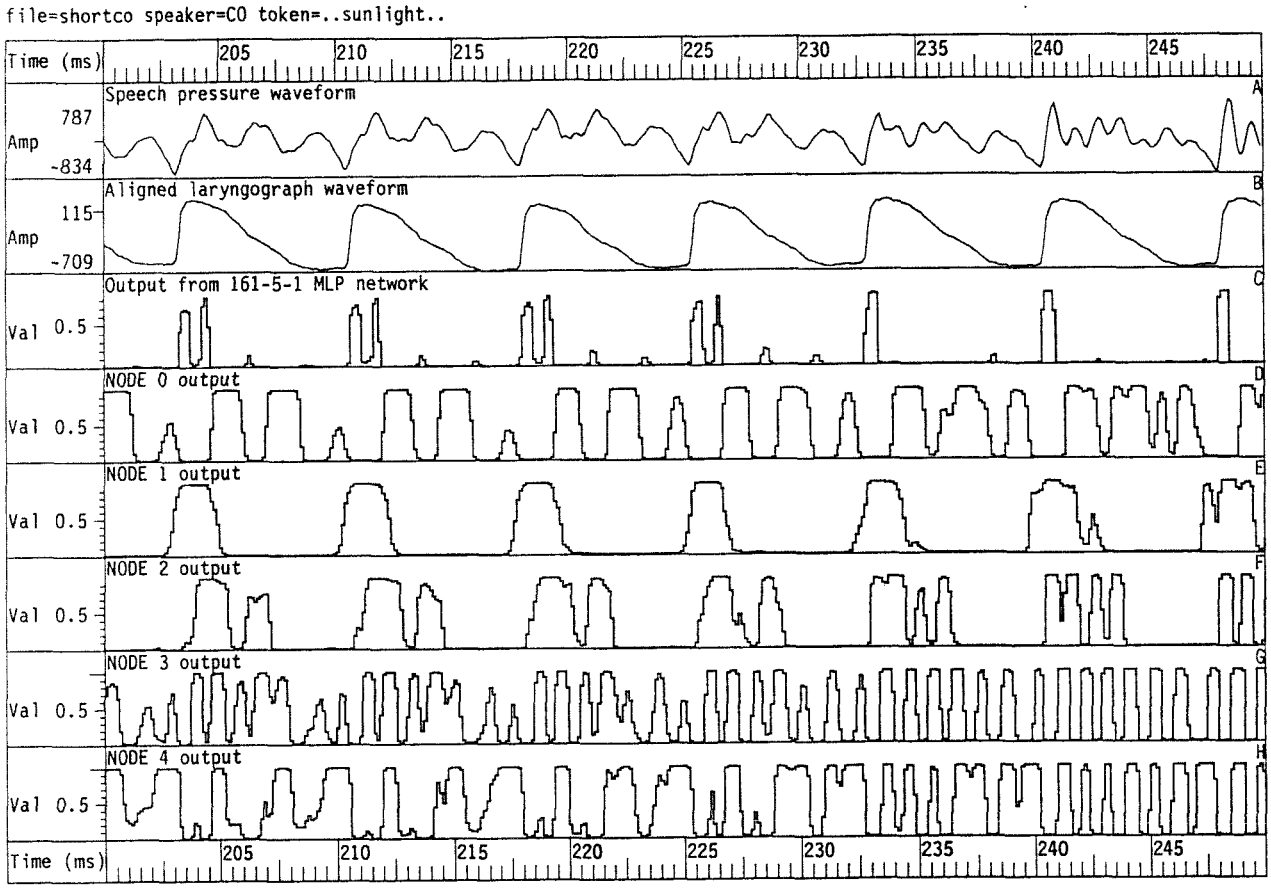


Figure 10.26 Output from first layer units in direct speech MLP-Tx algorithm, showing double-pulse error condition.

The network has 161 inputs, one layer of 5 hidden units and 1 output units, and was trained on men and women. Speech is shown in trace A, laryngograph waveform is shown in trace B and the period markers derived from is are shown in trace C. Trace C shows the overall MLP-Tx output and traces D to H show the outputs from the 5 first layer nodes. The speech was from a male speaker and was for the transition "na".

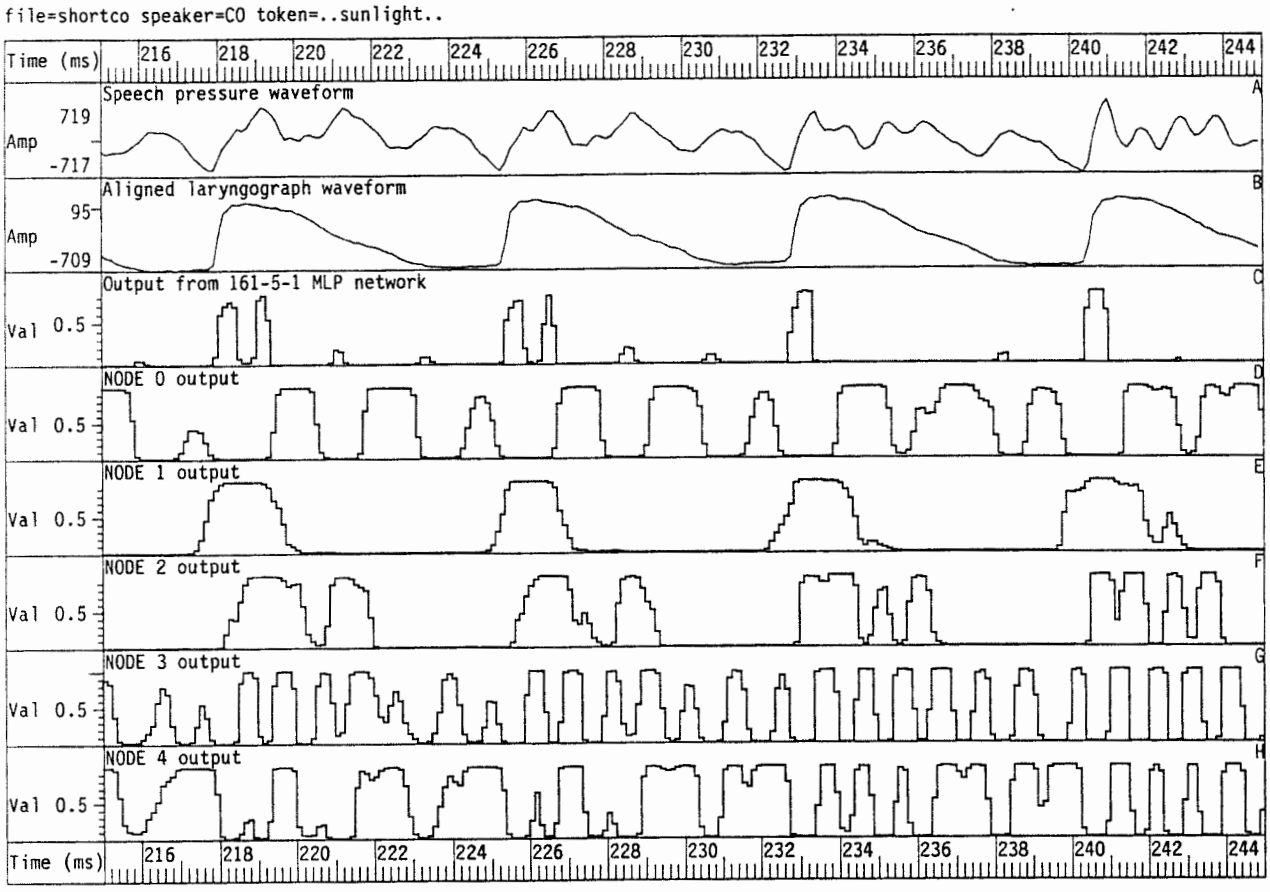


Figure 10.27 Same as in figure 10.26, but with an expanded time-scale.

CHAPTER 11: REAL-TIME IMPLEMENTATION OF THE MLP-Tx ALGORITHM

11.1 COMPUTATIONAL LOAD CONSIDERATIONS

11.1.1 Introduction

This chapter provides a discussion of the issues and problems involved in implementing the MLP-Tx algorithm in real-time on a TMS320C25 signal processor. Many of the issues in this chapter represent collaborative discussion and work with John Walliker. In particular, the real-time implementation of the MLP-Tx algorithm on the TMS320C25 was written by John Walliker, with the weights for the MLP provided by the author.

Firstly, the limitations of the TMS320C25 are described. Next, the processing load needed to implement the different stages in the MLP-Tx algorithm were estimated and this resulted in a system that could run on the TMS320C25. Results of a simulation are given which investigate the effect of quantization of the weights in the MLP, and the use of look-up tables to implement the sigmoid non-linearities. Finally, perceptual results for patients and normal listeners using a real-time MLP-Tx algorithm are presented (the perceptual tests were carried out by Dr. A Faulkner).

Due to its uniform structure, it is relatively easy to implement a MLP in a real-time device. Clearly processing requirements limit the size of the network and filterbank that can be used in a real-time system. To this end, the largest network and filterbank that could be run in real-time on a TMS320C25 signal processor was estimated and then investigated.

11.1.2 Limitations of the TMS320C25

For the MLP-Tx algorithm to be useful in signal-processing hearing aids, it must be implemented to run in real-time. Such a task requires a large amount of computation and consequently a fast digital signal processor was needed.

To achieve real-time operation of an algorithm, it is necessary that the processing load per unit time of input speech is no more than can be achieved per unit time by the available processor, and that the output from the algorithm becomes available at a short-time after the signal was presented. The latter issue presents no problem, since the delay from the MLP-Tx algorithm can be made small (that is, about 10ms).

11.1.3 Desirability of integer arithmetic and a look-up table

In a practical portable implementation of the MLP-Tx algorithm, it is advantageous if the arithmetic can be carried out by means of integer multiplications. This is because integer processors are generally more readily available and use less power than floating point ones and this is important when battery operation is required (which it is in a hearing aid). Additionally, the use of a look-up table to calculate the sigmoid non-linearity is also highly desirable, if not essential, because full calculation of the sigmoid is computationally expensive.

The operations involved in the original wideband filterbank MLP-Tx algorithm included the calculation of sigmoid non-linearities and logarithmic compression. To implement this by direct calculation on a integer DSP would use a large number of processor cycles. A practical solution was to use look-up tables to implement these functions.

The TMS320C25 was selected (by John Walliker) because it was as fast as other integer processor available ,it also had a relatively low power consumption, and at that time documentation and support for it were more readily available.

11.1.4 Limit on computation

This section shows how the size of a MLP-Tx algorithm that could run on a TMS320C25 in real-time was calculated.

The TMS320C25 run at a 40MHz clock-frequency is capable of 10 MIPs (million operations per second). For portable applications there are three considerations that

reduce this figure to 8 MIPs. Firstly, it is desirable to run the processor on a reduced voltage, to minimise battery power consumption. Secondly, the fastest available low power EPROMS had an access time that limited the maximum clock speed to 32MHz. Thirdly, 32MHz is a readily available crystal frequency.

11.1.5 Processor cycles for filters

The issue of the number of processing cycles necessary to implement the various operations in the MLP-Tx algorithms is now addressed. The overheads for the filtering constitute a large proportion of the overall number of cycles required to implement a second order filter. The DSP applications manual for the TMS320C25 (Texas Instruments, 1986) gives an example of a 4th order filter that requires 28 processor cycles per input sample. A filter lower in order by one will require one less multiply-add and associated overheads. Therefore, a second order filter will need at least 14 processor cycles per input sample (half the previous figure). Additional operations are then needed to implement half-wave rectification and logarithmic output scaling.

11.1.6 Processing load for previous filterbank system

The initial system consisted of 9 bandpass filters of 4th order running at an input frequency of 10kHz, followed by rectification and smoothing using 2nd order filter. Downsampling to 2kHz was then carried out followed by a log operation. This required at least 9*(28+14+2) = 396 processor cycles every 0.1ms for the filters and 9 for the log look-up. This corresponds to about 4 MIPs, which is 50% of the processor capacity.

11.1.7 Processor cycles for MLP

A multiply and add can be carried out within one processor cycle. There are set-up overheads associated with the calculation of a unit of about 10 processor cycles. The non-linear look-up table required about 10 cycles per node to implement. The delay-line shifts required no extra processor cycles.

The MLP network used 369 inputs and had two layers of hidden nodes each containing 10 units. This required 369*10+10*10+10 = 3800 multiply-add operations every 0.5ms and 20*10 + 28*10 = 400 operations per 0.5ms to implement the look-up operations and unit initializations. This corresponds to 8.4 MIPs.

11.1.8 Reduced computation filterbank

Consequently, in its original form, the MLP-Tx algorithm was estimated to use approximately 12.4 MIPs whereas only 8 MIPs were available. To reduce computation, the input sampling rate was reduced from the original 10kHz to 8kHz. This also has the beneficial effect that the latter is the telecommunications standard frequency, and consequently components at this frequency are readily available and good value for money.

The number of filter channels was reduced to 6. This was achieved by replacing the original five highest channels with two channels that covered the same 1kHz-3kHz range, because it was believed that this part of the frequency range was less important than the lower regions in which the fundamental frequency lies. In addition, the band-pass filters were reduced from 4th order to 2nd order. The exact cut-off frequencies for the filters were also chosen with regard to the stability of the filters (some initial cut-off frequencies resulted in unstable filters). Finally, the filterbank comprised six second order IIR Butterworth filters with -3dB points of 40-300Hz, 300-600Hz, 600-900Hz, 900-1200Hz, 1200-2000Hz, and 2000-3000Hz.

To implement this new filterbank required about 6*(14+14+2) = 180 processor cycles every 0.125ms. This corresponds to 1.4 MIPs.

11.1.9 Reduced computation MLP

Reducing the filterbank to 6 channels automatically reduced the size of the corresponding MLP network. However, it was necessary to additionally reduce the number of hidden units in the first layer to 6. The number of units in the second hidden

layer were also reduced to 6. Therefore the MLP network used 246 inputs and had two layers of hidden nodes each containing 6 units. This required 246*6+6*6+6 = 1518 multiply-add operations every 0.5ms and 20*10 + 28*10 = 400 operations per 0.5ms to implement the look-up operations and unit initializations. This corresponds to 3.8 MIPs. A schematic diagram illustrating the configuration for the reduced computation MLP-Tx algorithm is shown in figure 11.1.

This gives an overall processing load of 1.4 + 3.8 = 5.2 MIPs which can be run on the TMS320C25 in real-time. In addition, it leaves some spare capacity that is needed by other operations that must be implemented in the hearing aid, such as the formatting of the output signal, etc.

11.2 SIMULATION OF HARDWARE IMPLEMENTATION

11.2.1 Introduction

To investigate the effects of quantization and the use of look-up tables for the non-linear functions, a set of simulation experiments was carried out. This involved performing the calculations in the MLP to the specified number of levels of quantization. The effect of different size look-up tables was also investigated.

11.2.2 Investigation into the effects of quantization

The effect of quantization were assessed on a version of the real-time MLP-Tx algorithm that was trained (using a floating point representation of the weights) on the four female training speakers (as described in chapter 9), and it was then tested on the ten female speakers in the evaluation data set. Frequency contour comparisons were then used to asses the performance.

The effect of quantization of the weights is to introduce inaccuracy in their representation. To investigate the effect of quantization, all the numbers (weights and results) used in the MLP-Tx algorithm were quantized. This was achieved as follows:

First the sign of the number was recorded and the number set to its positive magnitude value. It was then scaled such that the range of numbers used for weights would lie in the interval 0 to 1. This value was then multiplied by half the required quantization levels, and the number was then rounded to the nearest integer. The number was then scaled back to its original magnitude, but now with quantization uncertainty introduced. Its sign was then restored.

11.2.3 Qualitative evaluation of the effect of quantization

The outputs from the MLP-Tx algorithm for different levels of quantization are shown in figure 11.2. Trace A shows the speech pressure waveform and B the corresponding laryngograph waveform. Trace C shows the output with no quantization. Trace D shows the output with 512 levels of quantization, which is visually the same as the un-quantized case. Traces E to J show the output for 256, 128, 64, 32, 16 and 8 levels of quantization respectively. It can be seen that fewer than 64 levels of quantization affects the height of the output waveform. The output pulses remain well-defined even with 16-levels of quantization, although unwanted pulses outside the voicing region become emphasised.

11.2.4 Quantitative assessment of the effect of quantization

The quantitative performance of the algorithm for different levels of quantization is shown in figures 11.3 to 11.8.

It can be seen that the performance does not change much until the quantization uses less than about 128 levels, and operation using 8-bit resolution is certainly satisfactory. It is interesting to notice that the different comparison metrics show different amounts of degradation (which demonstrates the value of using all the metrics). For example, as the quantization levels go between infinite (unquantized) to 16 levels, the voiced-to-unvoiced errors reduce but the unvoiced-to-voiced errors increase. Similarly, the frequency contour drop errors reduce and the chirp errors increase - and consequently the overall number of **gross** errors stays almost constant up to 64 levels of quantization.

The metrics that measure the hit rate and the fine accuracy of the period estimates show less degradation than some of the others, and the hit rate actually increases with decreasing quantization levels, until the 8-level point is reached. For 8 levels of quantization, most of the metrics indicate degraded performance.

11.2.5 Investigation into the effects of using a look-up table

To investigate the effect of using a look-up table, the sigmoid non-linearity in the MLP was replaced with an appropriate look-up table and a level clipping function. That is, if the input was greater than the upper clipping limit, the output was set to 1.0. If it was lower than the lower clipping limit, it was set to 0.0. If the input lay in between the two clipping limits, the output was determined by the look-up table. The clipping limits were determined so that numbers of greater magnitude would give no change in the quantized output from the look-up table. In this way, the look-up table would be used efficiently to model the more linear region of the sigmoid, and not simply act as a clipper itself.

The look-up table to perform the function of the sigmoid non-linearity was generated as follows. Firstly it was assumed that the output would be quantized into 512 levels, since this appeared to be a generous value in view of the quantization results, and includes a substantial safety margin. Consider that there were N entries in the look-up table. This range was offset to -N/2 to +N/2 by adding N/2 to the input value. This provided a means to deal with negative inputs to the sigmoid function. A scaling factor was then used to map numbers in the range between the two clipping levels to the -N/2 to +N/2 range of the look-up table. The look-up table output values were then generated by using the real sigmoid over the range that could be dealt with by the look-up table.

11.2.6 Qualitative evaluation of the effect of a look-up table

The outputs from the MLP-Tx algorithm for different sized look-up tables are shown in figure 11.9. Trace A shows the speech pressure waveform and B the corresponding laryngograph waveform. Trace C shows the output with no look-up table. Trace D

shows the output with a look-up table with 128 entries, which appears visually the same as without any look-up table. Traces E to J show the output for look-up tables of sizes 128, 64, 32, 16, 8, 4 and 3 respectively. It can be seen that fewer than 32 entries has a significant effect on the height output waveform, as well as the shape of the output pulses. No useful output is generated in this example for tables using only 4 and 3 entries.

11.2.7 Quantitative assessment of the effect of a look-up table

The performance of the algorithm in quantitative terms for different levels of quantization is shown in figure 11.10 to 11.15. These evaluations was carried out using the same MLP-Tx algorithm and test data used for the quantization experiments.

It can be seen that the performance does not change much until a look-up table with fewer than about 32 entries is used, and operation using a look-up table with an 8-bit input resolution is completely satisfactory. The different comparison metrics again show different amounts of degradation as the number of entries in the look-up table are decreased.

There was virtually no degradation in any metric until fewer than 32 locations were used. As the number of locations decreased, the voiced-to-unvoiced errors reduced but the unvoiced-to-voiced errors increased. Similarly, the frequency contour drop errors reduce and the chirp errors increase. The fine accuracy of the period estimates showed less degradation than some of the others. With look-up tables with only 4 locations, most of the metrics indicate degraded performance. It is interesting to notice that a 3 entry look-up table gave much better quantitative results than the 4 entry look-up table, and was in fact almost as good as 32 entries.

11.3 PERCEPTUAL EVALUATIONS OF THE REAL-TIME MLP-Tx ALGORITHM IN THE EPI HEARING AID

11.3.1 Introduction

A real-time implementation of the reduced filterbank and network MLP-Tx algorithm was written by John Walliker, with weights provided for the MLP by the author. This system was then used in perceptual evaluations with normal listeners and patients with a profound hearing loss.

MLP-Tx algorithm used for perceptual tests

The reduced computation wideband filterbank MLP-Tx algorithm described above was trained on two female speakers in pink noise at a signal to noise ratio of 3dB, with respect to 500ms frames in each signal and noise that contained the maximum power. Details of the training are the same as described in chapter 9, except a final training pass over the data was made in which the emphasis of the period marker patterns were increased to ten, as opposed to the usual value of one. This has the effect of changing the operating point of the MLP-Tx detector so that it generates more hits, but also more false alarms. In the perceptual tests, it was felt by the tester that it was more important to get a large number of hits, because the occasional false alarms were less important than misses.

11.3.2 Perceptual assessment task

An intervocalic consonant test (Rosen et al., 1979) was used to provide the material for the perceptual evaluations. This test consists of the presentation of a vowel-consonant-vowel (VCV) sound, which must then be identified by the subject. This is done for the twelve British English consonants [m b p v f n d t z s g k]. Because half of the consonants are voiced and the other half are unvoiced, it is possible to compute the errors in voicing that were made by the subject by analyzing his/her responses to the test. The test was presented in the form of a video recording of a female speaker of standard southern British English.

The fundamental frequency (periods) was estimated using the MLP-Tx algorithm and also using the peak-picker, so that a comparison could be made.

11.3.3 Comparing amplified speech presentation against using fundamental period estimate presentation from the MLP-Tx algorithm

Figures 11.16 and 11.17 both show the overall correct and voicing information reception in the consonant identification task for two patients (S1 and S11) using the SiVo hearing aid. Each symbol represents at least 48 trials. Lines are drawn between the means of the plotted points and the vertical bars show the range of the points.

The tests conditions were:
L+Sp - lipreading with speech information from the patient's amplifying hearing aid.
L+(Sx)A - lipreading with fundamental frequency information from the MLP-Tx algorithm,. presented as a frequency controlled sinusoid, and speech amplitude information presented as an amplitude modulation of the sinusoid. The amplitude envelope was extracted by half-wave rectification and low-pass filtering at 20Hz and 24dB/octave.
Lipreading alone.

In subjects 1 and 3 (S1 and S3), data was also collected for lipreading with fundamental frequency information from the MLP-Tx algorithm without amplitude information.

In quiet conditions, there is little difference in voicing information performance using either speech or the MLP-Tx algorithm. However, as the signal-to-noise ratio increases, the performance using the speech degrades significantly, whereas the performance using the MLP-Tx algorithm is much less affected. At a 5dB SNR, using the speech signal, no useful voicing information is transferred, whereas the MLP-Tx algorithm still provides a significant amount of information relating to voicing.

These results demonstrate the value of using the MLP-Tx algorithm as opposed to merely presenting an amplified version of the speech to the patients. The MLP-Tx algorithm gave better performance in noise for the discrimination of voicing contrasts than the patients could achieve using the whole speech signal.

11.3.4 Comparing fundamental period estimate presentation from the peak-picker or from the MLP-Tx algorithm

Figures 11.18 and 11.19 both show the overall correct and voicing information reception in the consonant identification task for two normal subjects and a cochlear implant patient using the UCH/RNID single channel cochlear implant. Each symbol represents at least 48 trials. Lines are drawn between the means of the plotted points and the vertical bars show the range of the points.

The tests conditions were:
L+Fx(mlp) - lipreading with fundamental frequency (Fx) information from the MLP-Tx algorithm.
L+Fx(pp) - lipreading with fundamental frequency (Fx) information from the peak-picker algorithm.
L+Sp - lipreading with speech information from the patient's amplifying hearing aid.
Lipreading alone.

The MLP-Tx algorithm was able to extract information concerning the voiced-unvoiced contrast in a 5dB SNR, and its performance under such conditions were no worse than in the noise-free situation. The peak-picker, however, showed a performance degradation with increasing SNR, and at the 5dB SNR point it provided no useful voicing information.

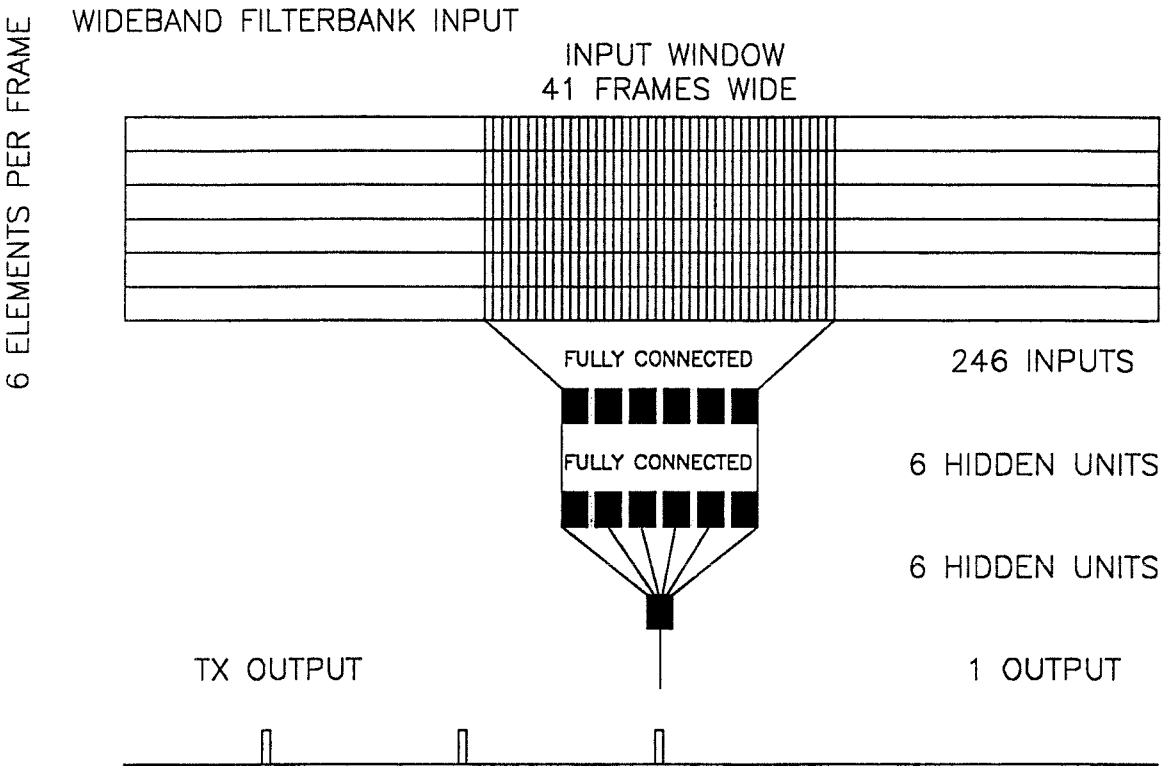


Figure 11.1 Schematic diagram for the MLP-Tx algorithm of reduced computational complexity.

The reduction in computation was necessary to permit real-time operation on a TMS320C25 digital signal processor.

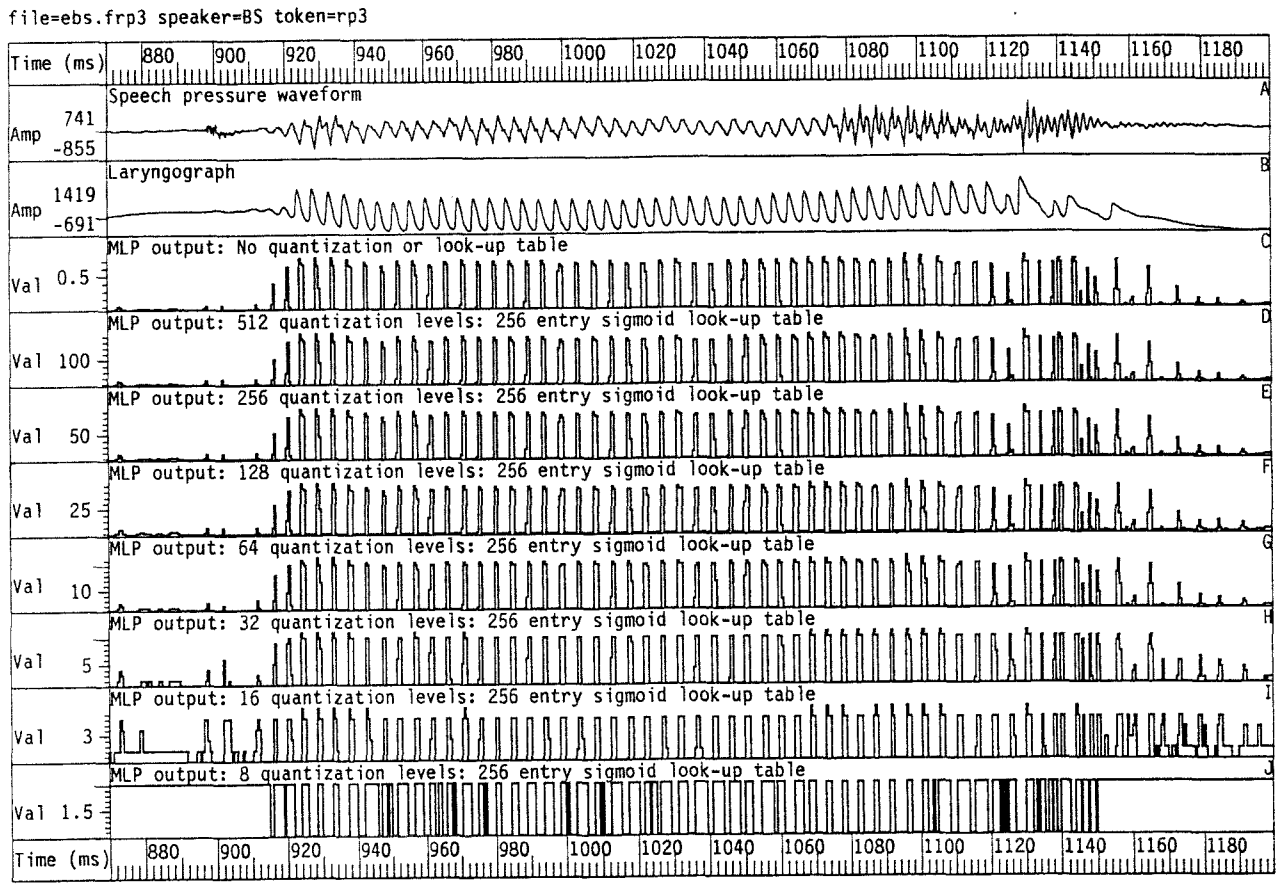


Figure 11.2 Diagram illustration the effect of quantization.

The outputs are from the reduced computational complexity MLP-Tx algorithm. Trace A shows the speech pressure waveform from a female speaker. Trace B shows the laryngograph waveform. Traces C to J show the MLP-Tx algorithm output for varying degrees of quantization of the weights. In all cases, a 256 entry look-up table was used.

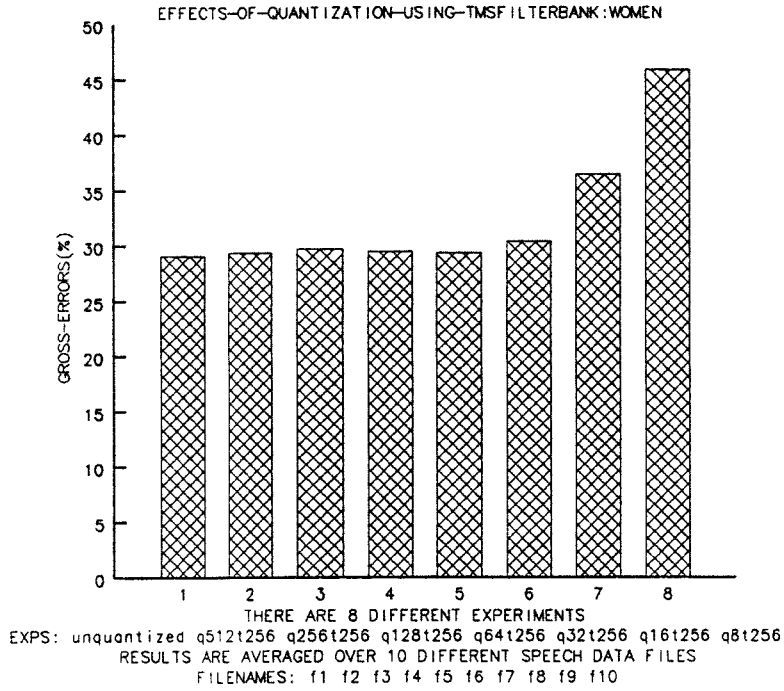


Figure 11.3 Bar-graph showing the gross errors generated for different levels of quantization on women evaluation data set.

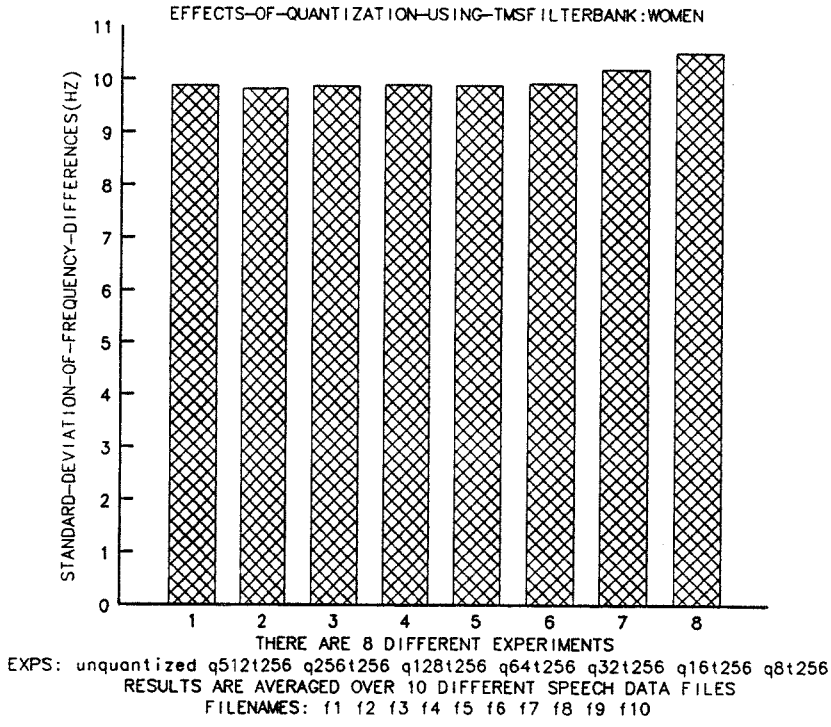


Figure 11.4 Bar-graph showing the standard deviation of fine frequency differences errors generated for different levels of quantization on the women evaluation data set.

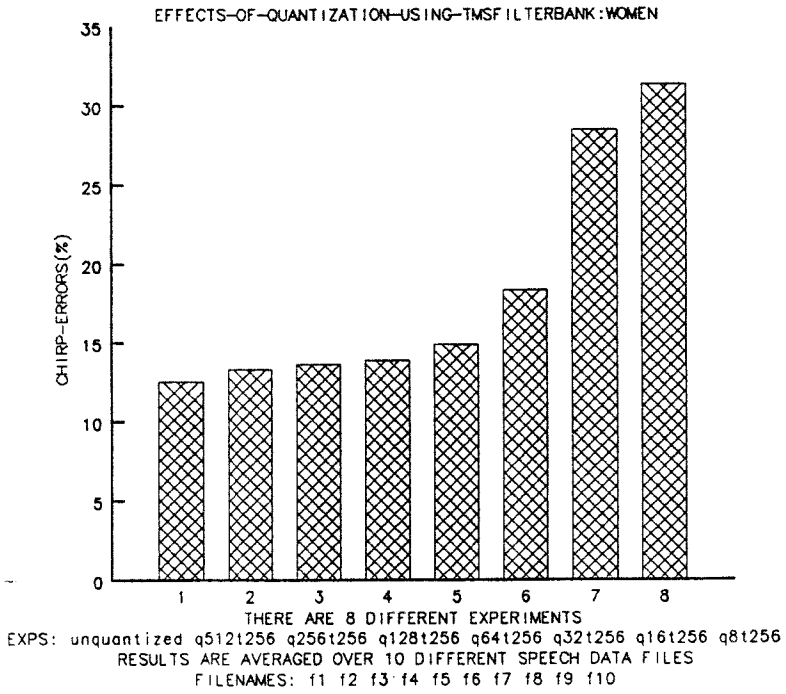


Figure 11.5 Bar-graph showing the chirp errors generated for different levels of quantization on women evaluation data set.

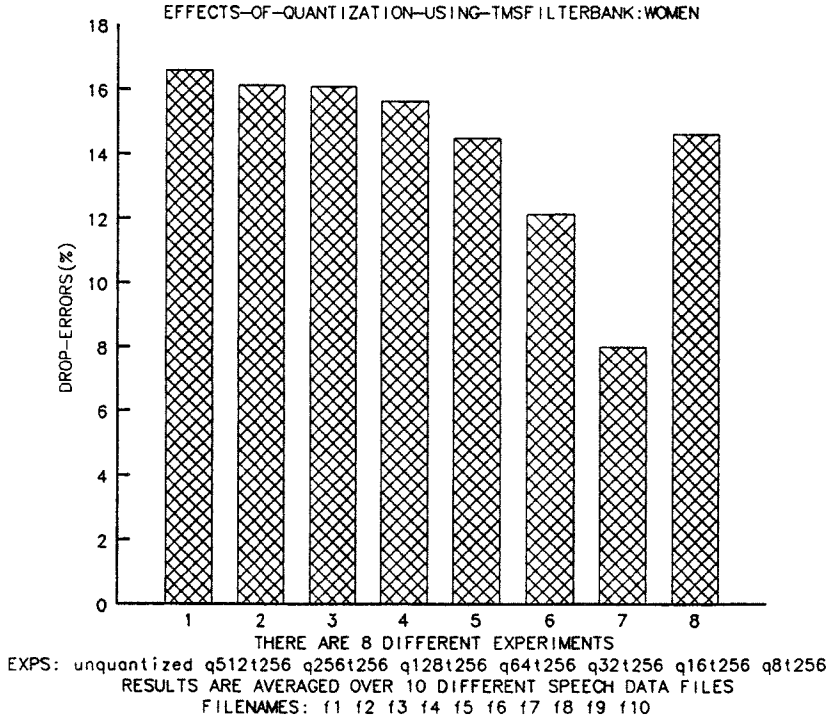


Figure 11.6 Bar-graph showing the drop errors generated for different levels of quantization on women evaluation data set.

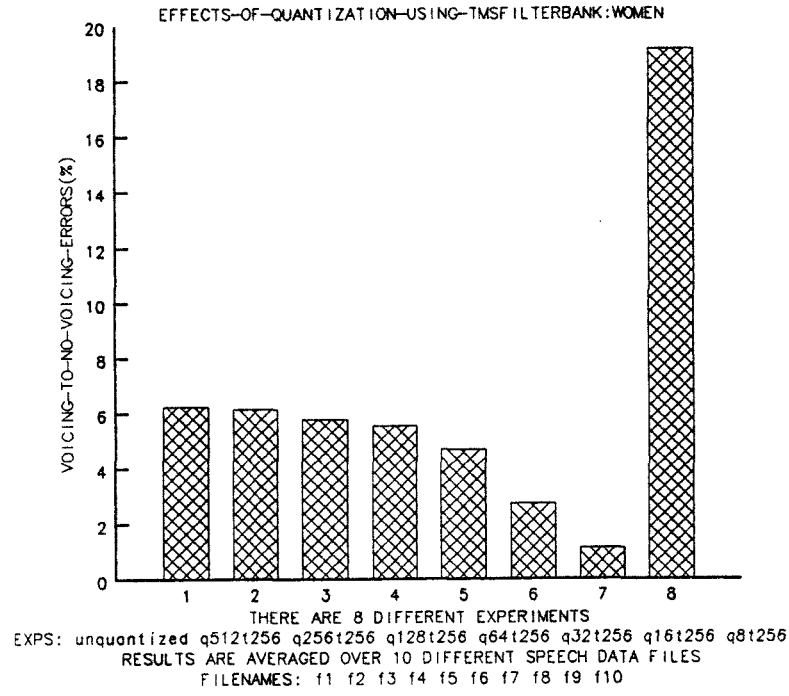


Figure 11.7 Bar-graph showing the voiced-to-unvoiced errors generated for different levels of quantization on the women evaluation data set.

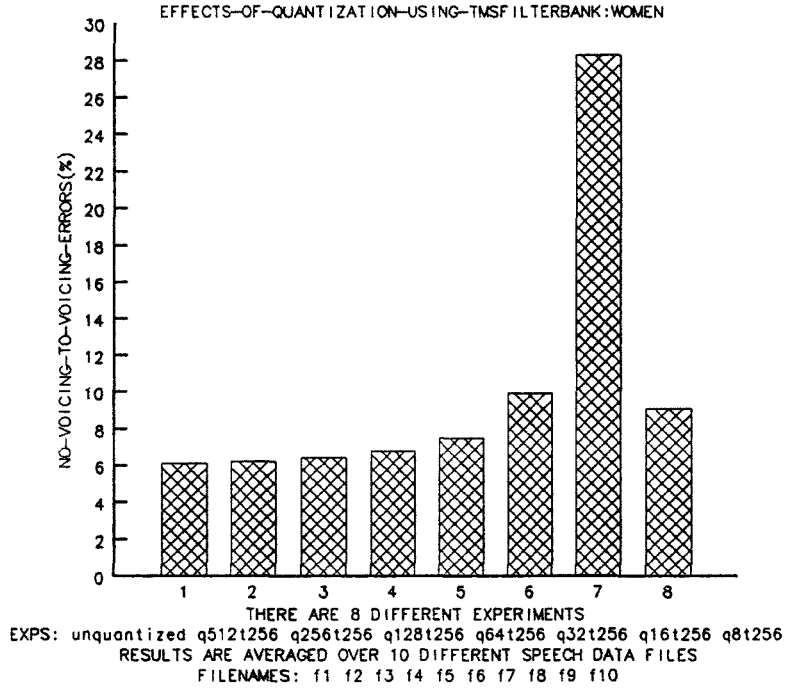


Figure 11.8 Bar-graph showing the unvoiced-to-voiced errors generated for different levels of quantization on the women evaluation data set.

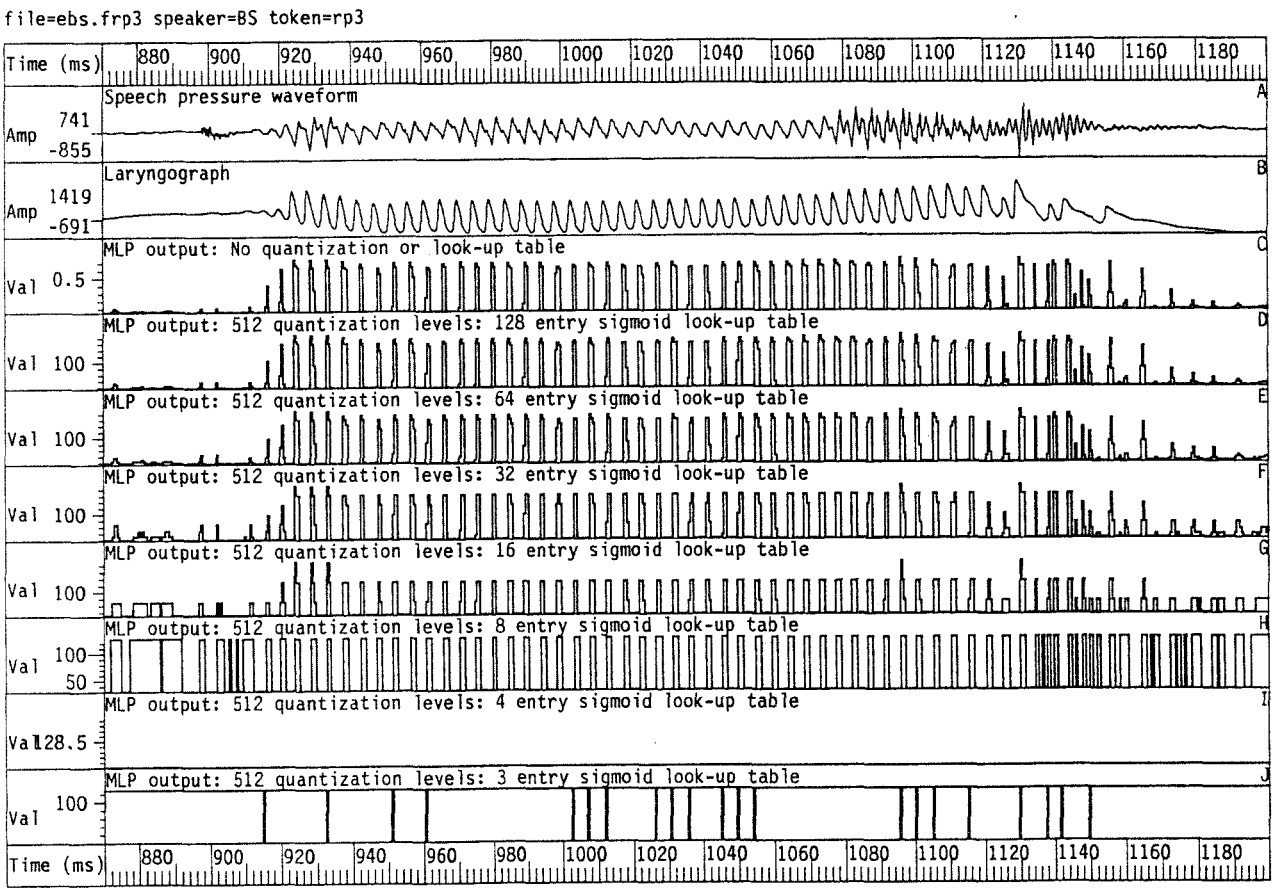


Figure 11.9 Illustration of the effect of look-up table size on the output from the reduced computational complexity MLP-Tx algorithm.

Trace A shows the speech pressure waveform from a female speaker. Trace B shows the laryngograph waveform. Traces C to J show the MLP-Tx algorithm output for varying degrees of quantization of the weights. In all cases, 512 quantization levels were used.

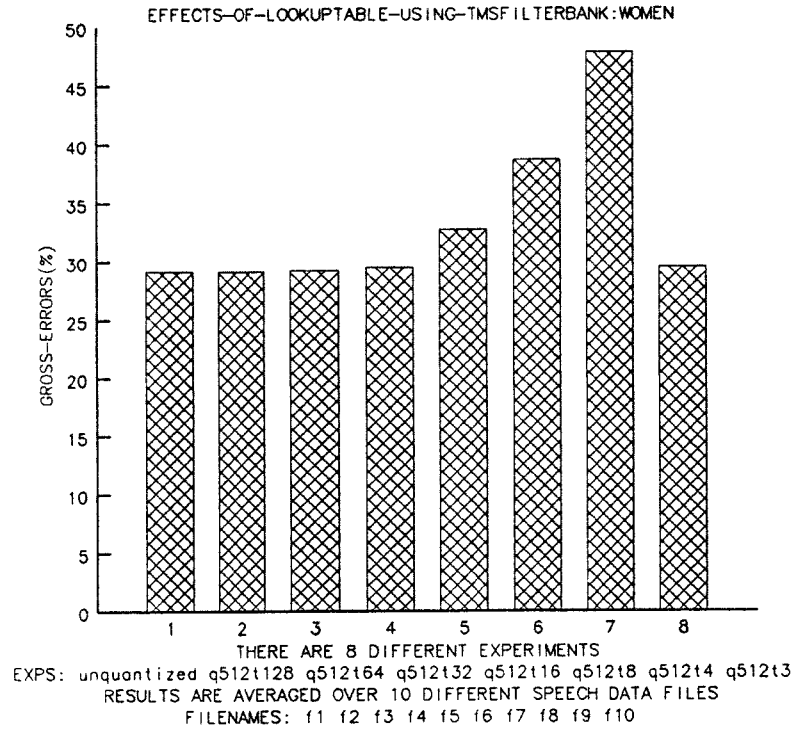


Figure 11.10 Bar-graph showing the gross errors generated for different look-up table sizes.

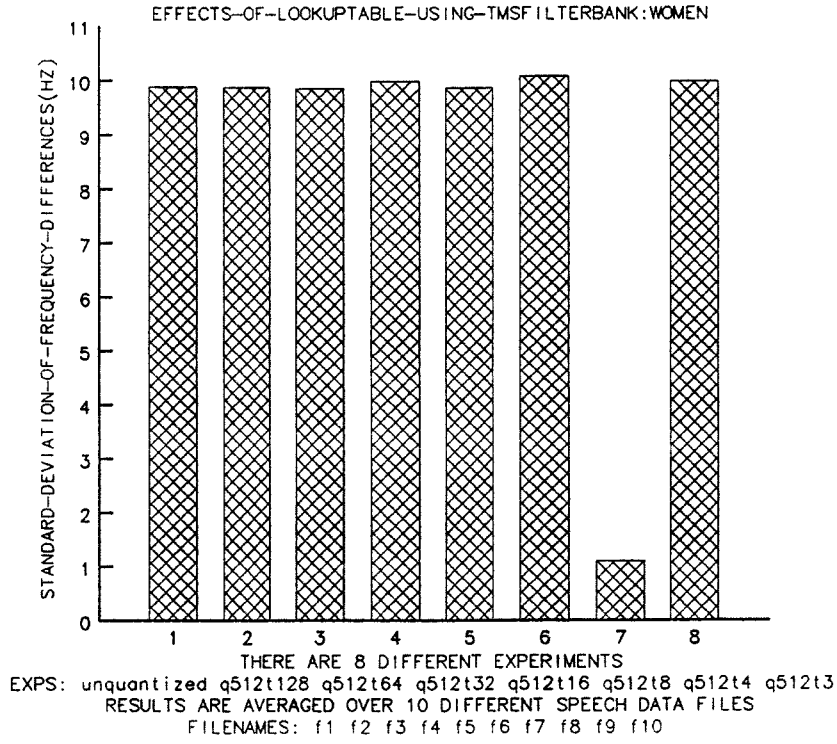


Figure 11.11 Bar-graph showing the standard deviation of fine frequency differences errors generated for different look-up table sizes.

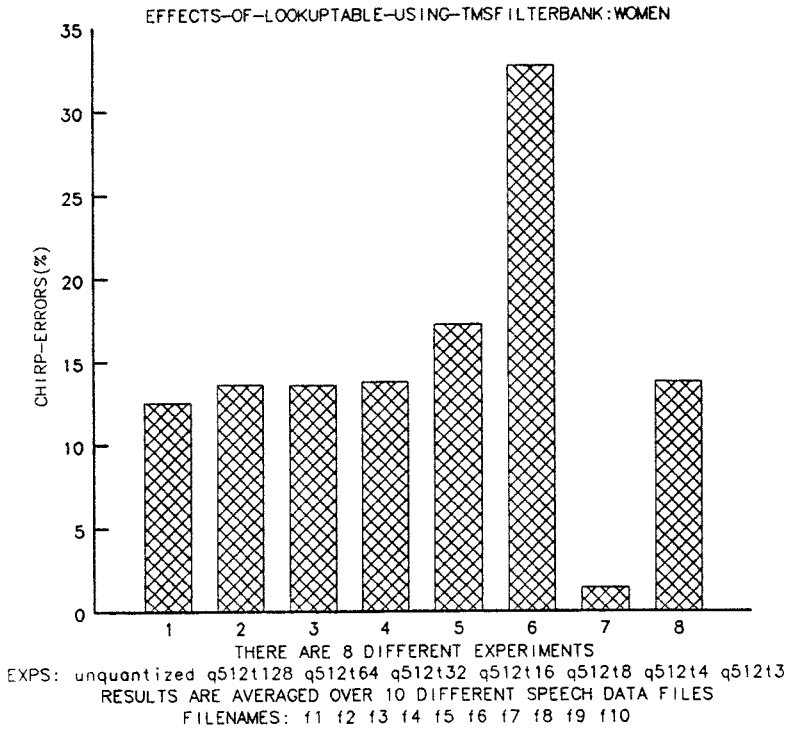


Figure 11.12 Bar-graph showing the chirp errors generated for different look-up table sizes.

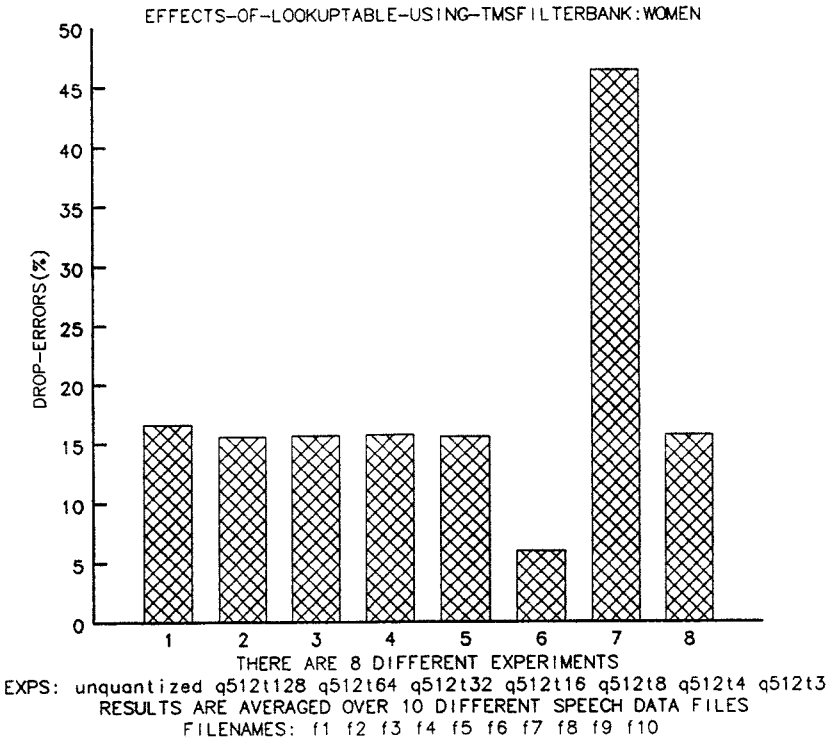


Figure 11.13 Bar-graph showing the drop errors generated for different look-up table sizes.

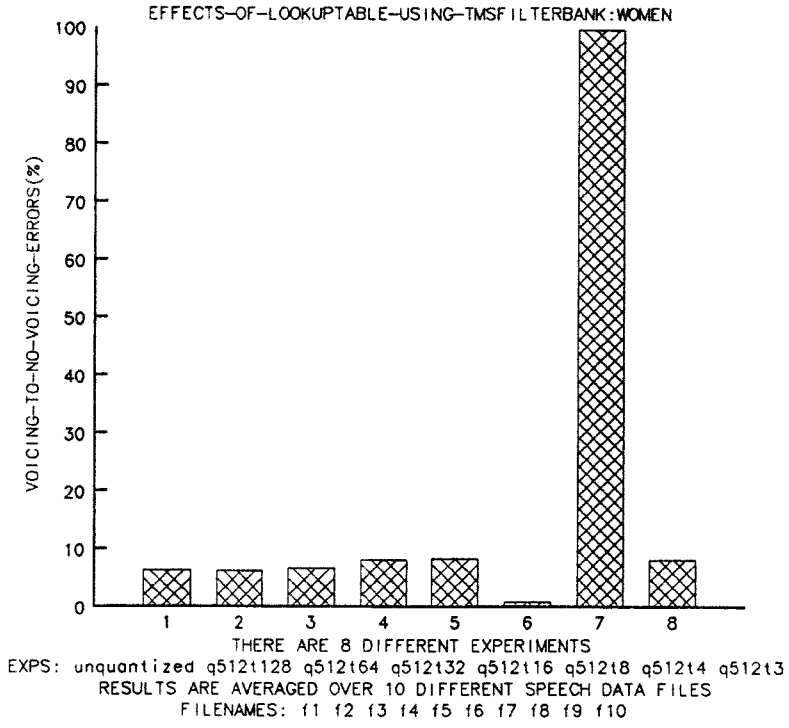


Figure 11.14 Bar-graph showing the voiced-to-unvoiced errors generated for different look-up table sizes.

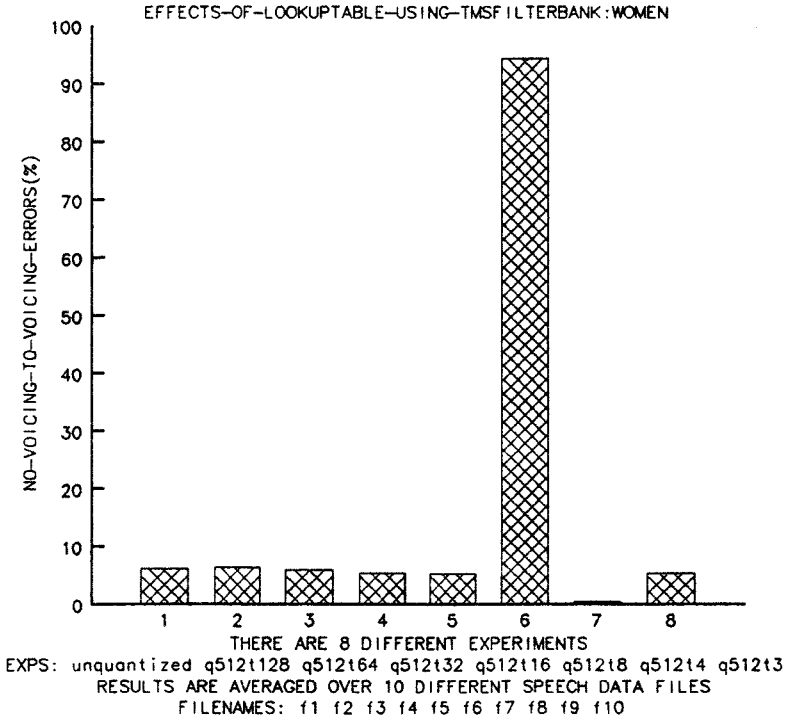
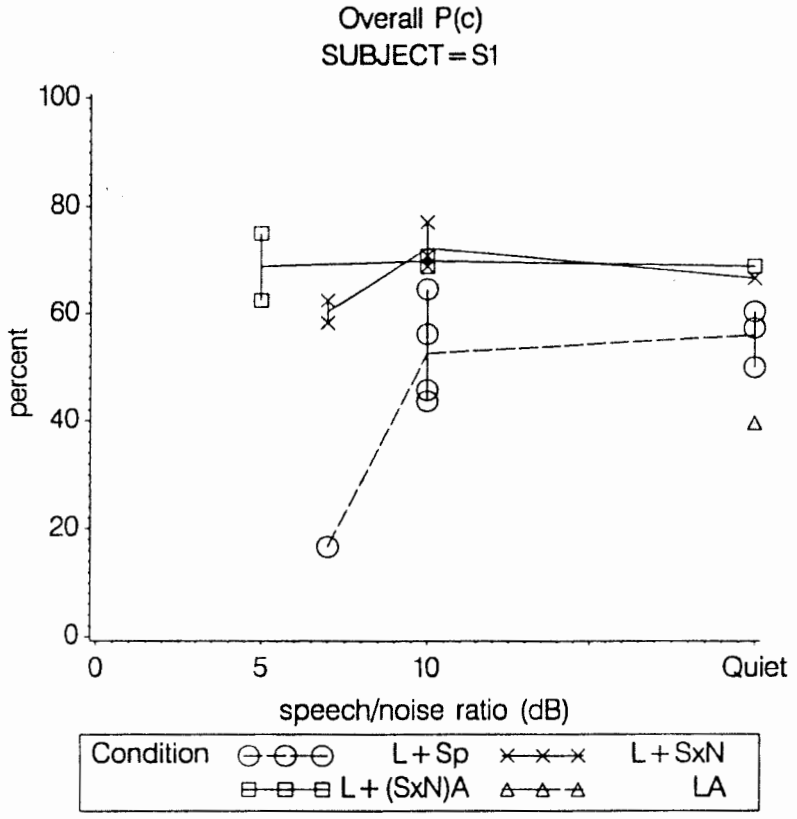


Figure 11.15 Bar-graph showing the unvoiced-to-voiced errors generated for different look-up table sizes.



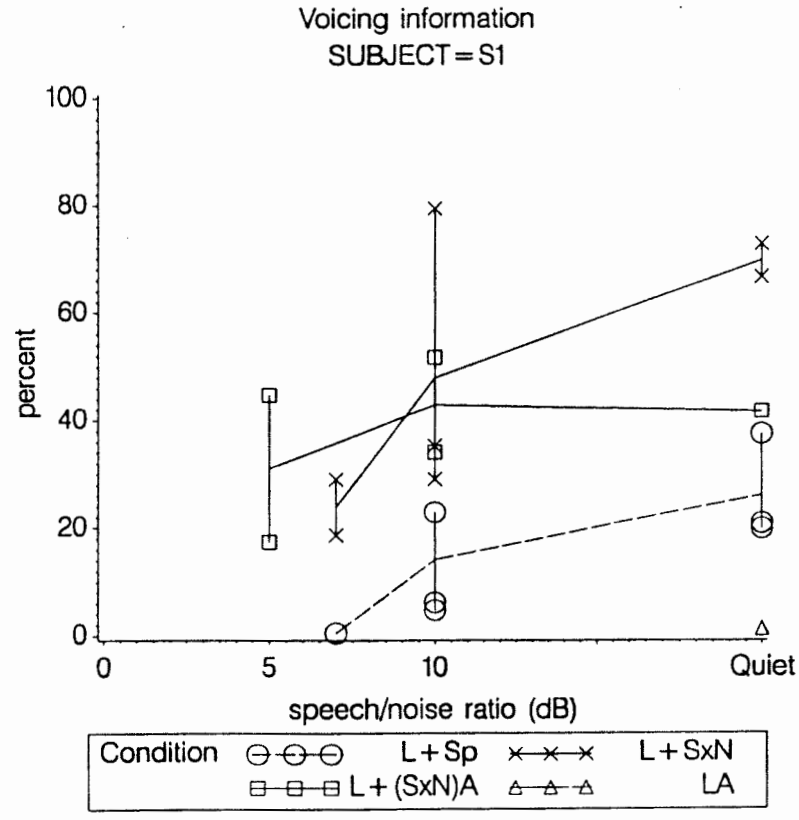
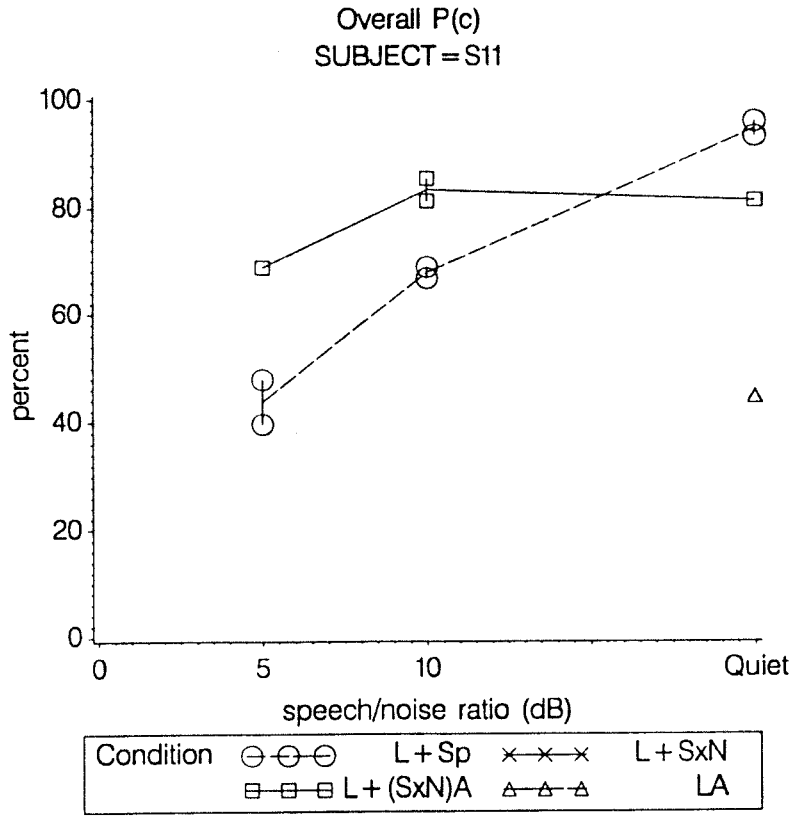


Figure 11.16 Overall correct and voicing information reception in audio-visual consonant identification using the MLP-Tx algorithm and direct speech presentation for subject S1.

(After Andrew Faulkner).



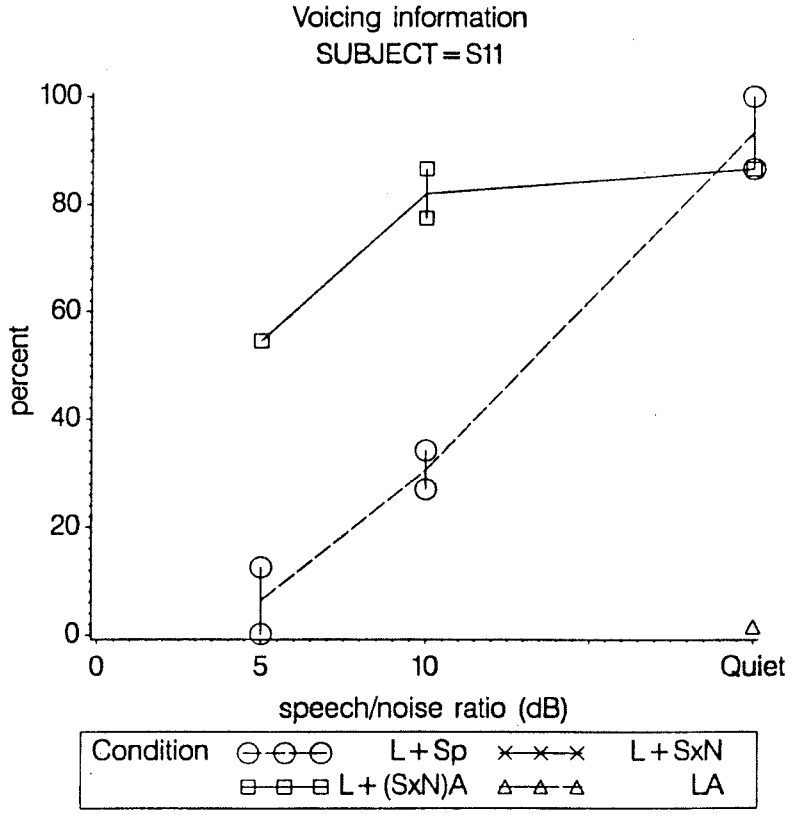
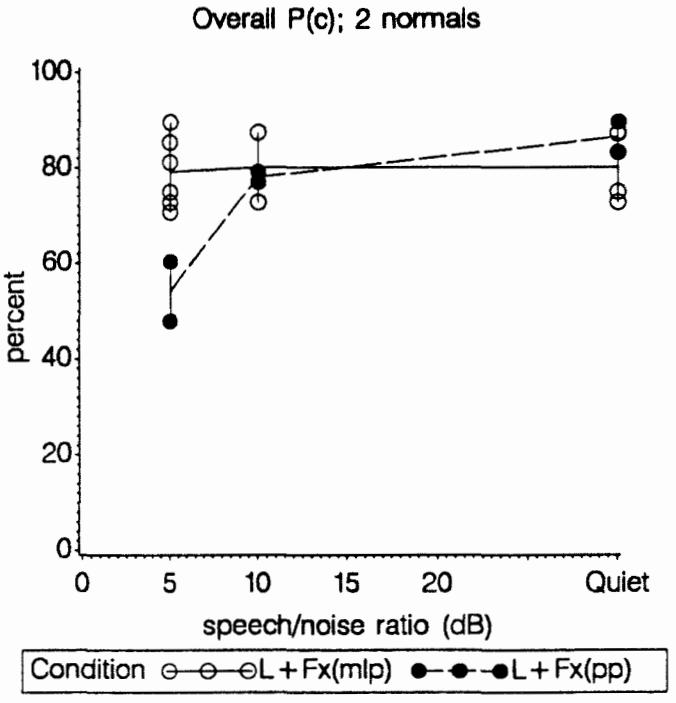


Figure 11.17 Overall correct and voicing information reception in audio-visual consonant identification using the MLP-Tx algorithm and direct speech presentation for subject S11.

(After Andrew Faulkner).



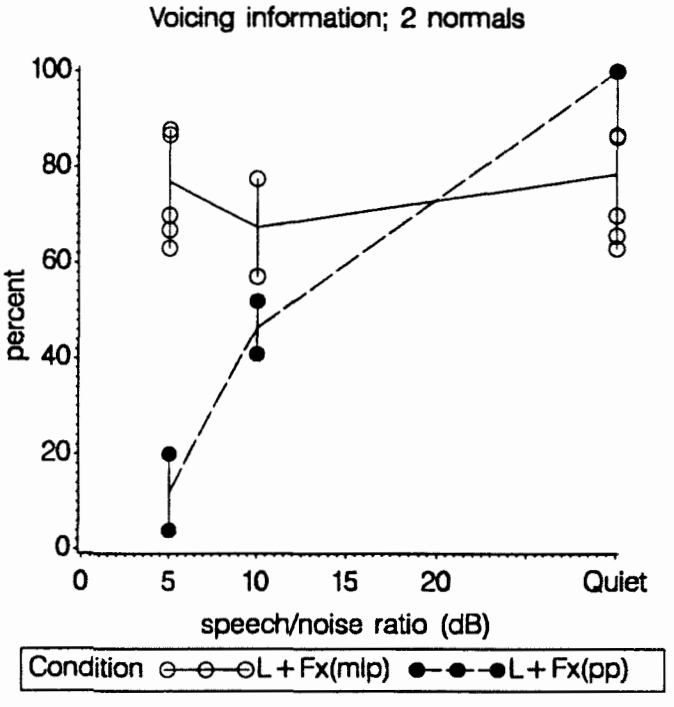
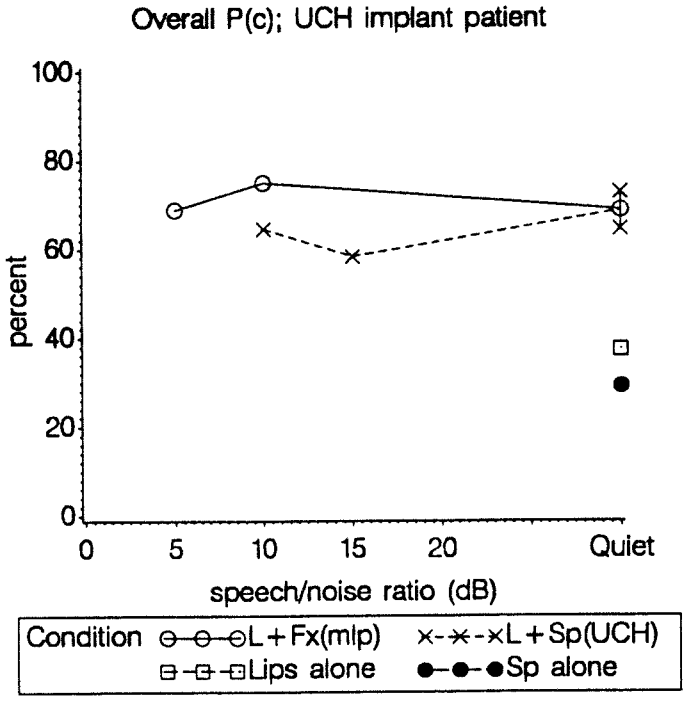


Figure 11.18 Overall correct and voicing information reception in audio-visual consonant identification using the MLP-Tx algorithm and the peak-picker algorithm for two normal subjects.

(After Andrew Faulkner).



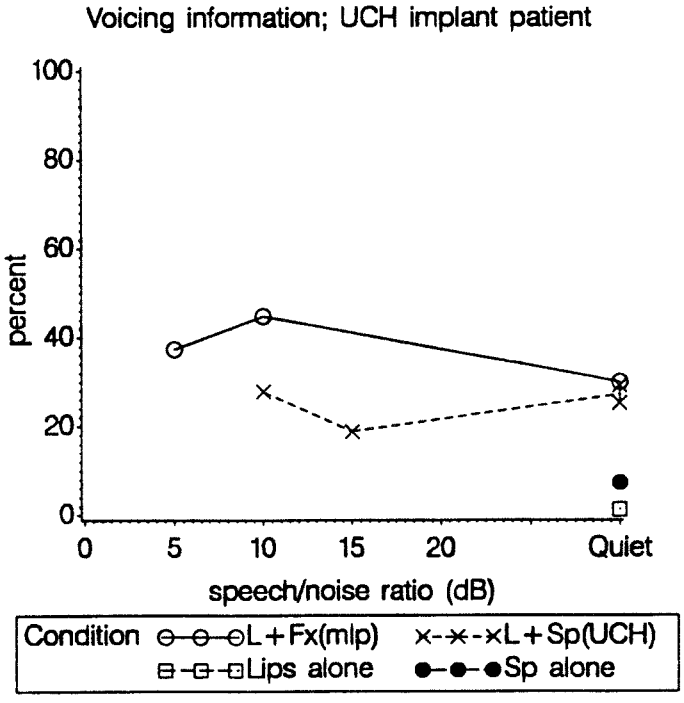


Figure 11.19 Overall correct and voicing information reception in audio-visual consonant identification using the MLP-Tx algorithm and the peak-picker algorithm for a UCH/RNID cochlear implant patient.
(After Andrew Faulkner).

CHAPTER 12: CONCLUSIONS

This chapter provides a summary and conclusions of the most important experiments and achievements of the work described in this thesis.

12.1 SPEAKER DEPENDENT INITIAL EXPERIMENTS

12.1.1 Preliminary experiment

The first experiment carried out on several speakers used a wideband-filterbank pre-processing. Performance (on five male speakers used to train the algorithm) was shown to be better than a that for the peak-picker in the presence of noise. A limitation of this experiment was that the same speakers were used for training and testing of the algorithm, and the results were only on male speech.

12.2 SPEAKER INDEPENDENT EXPERIMENTS USING REVERBERANT SPEECH

12.2.1 New database

A new database was recorded, so that the performance on female speakers could be gauged, and to permit speaker independent comparisons to be made. In addition, the recording conditions were selected to be representative of real conditions likely to be encountered by a fundamental period estimation algorithm operating in a signal processing hearing aid. That, background noise and reverberation.

Three separate sets of data were recorded. Firstly, there was a training data set. An important requirement was that the speech and laryngograph signals had to be recorded with a constant time-delay, and this was achieved by fixibg the microphone to a rod attached to a helmet worn by the subjects. The speech and laryngograph signals for the training data were then time-aligned.

There were also two separate testing data sets, each of which contained different

speakers. Two data sets were used so that primary evaluations of algorithm parameters could be made on one set, and then final unbiased comparisons with other technique could be made on the other set.

12.2.2 Three types of pre-processing

Whereas the preliminary experiments used filterbank pre-processing, later experiments were carried out using direct speech input to the MLP, a reduced sized wideband filterbank and an auditory filterbank. The direct speech MLP-Tx algorithm operated at the full frame-rate of the input speech (8kHz), whereas the filterbanks operated at a reduced rate (2kHz).

12.2.3 Selective emphasis training

The original training using back-propagation was slow. A technique was devised whereby the contribution of a pattern to the MLP weight updates was made on the basis of the response of the MLP to the particular pattern. If a patterns generated an output within a preset range of the target value, it was ignored. This procedure resulted in approximately between 3-10 times faster training.

Results on the evaluation data set for the direct speech, reduced wideband filterbank and auditory filterbank MLP-Tx algorithms were generated for the male and female data. The output waveforms from the MLP-Tx algorithm was examined for normal and erroneous conditions.

Final results for the best configurations of each type of MLP-Tx algorithm were then generated on the final test data. The results for the peak-picker and cepstral analysis were also generated for the purpose of comparison.

12.2.4 Frequency contour comparisons

Fundamental frequency contour comparisons were then made. It was found that the

direct speech operation gave fewer gross errors than the two filterbanks. Its period estimate resolution was also higher, because it operates at a higher frame-rate In addition, the auditory filterbank gave better performance than the reduced wideband filterbank. It was found that the MLP-Tx algorithm performed better than cepstrum and the peak-picker in terms of voicing determination. Its performance in terms of gross errors was worse than the cepstrum in terms of gross errors, but it must be borne in mind that the cepstrum algorithm included a gross error correction routine, whereas the MLP-Tx algorithm only employed simple post-processing. The cepstrum performed much worse than the MLP-Tx algorithm in terms of voicing determination.

12.3 REAL TIME IMPLEMENTATION AND PERCEPTUAL RESULTS

12.3.1 Real-time implementation

To run the MLP-Tx algorithm in real-time, the computational load of the original wideband filterbank version was reduced. A real-time implementation was carried out in conjunction with John Walliker, and this was then used by Andrew Faulkner as a source of fundamental period information in perceptual tests using mormal and profoundly deaf subject in a consonant resognition task.

12.3.2 Perceptual results for normal subjects and profoundly deaf patients

The perceptual results showed that the real-time MLP-Tx algorithm performed better in the presence of noise than the peak-picker, and gave useful output at 5dB SNR, whereas the peak-picker gave no useful output at this noise level. The tests also showd the value of using pattern element extraction since better results were obtained with the profoundlt deaf patients using the MLP-Tx output than using the whole speech signal.

The general conclusion is that a new approach to fundamental period estimation has been designed and developed sufficiently that it is of practical value in signal processing hearing aids, and in noisy conditions it has been shown to give better results in such an application than the peak-picker.

REFERENCES

ABERCOMIE, (1964), English Phonetic Texts, London, Faber & Faber, (The story of Arthur the Rat, after H SWEET, (1895) A Primer of Spoken English. Oxford: Clarendon Press).

E ABBERTON, (1974), Listener identification of speakers from larynx frequency, Proc. 8th Int. Congr. On Acoustics, London: Chapman & Hall.

E ABBERTON, (1976), A laryngographic study of voice quality, PhD thesis, University of London.

E R M ABBERTON & A J FOURCIN, (1973), A visual display for teaching intonation and rhythm, E L T Documents, 73/5, pp2-6.

E R M ABBERTON, A J FOURCIN, S R ROSEN, J R WALLIKER, D M HOWARD, B C J MOORE, E E DOUEK, & S FRAMPTON, (1985), Speech perceptual and productive rehabilitation in electro-cochlear stimulation, In R A Schindler & M Merzenich (eds), Cochlear Implants, New York: Raven Press, pp527-537.

E R M ABBERTON, D M HOWARD & A J FOURCIN, (1989), laryngographic assessment of normal voice: A tutorial, Clinical Linguistics & Phonetics, Vol. 3, No. 3, pp281-296.

N ABRAMSON, Information theory and coding, McGraw-Hill.

D H ACKLEY, G E HINTON & T J SEJNOWSKI, (1985), A learning algorithm for Boltzmann machines, Cognitive Sciences 9: pp147-169.

J ALLEN, (1985), A perspective on man-machine communication by speech, Proc. IEEE, Vol 73, No. 11.

T V ANANTHAPADMANABHA & B YEGNANARAYANA, (1975), Epoch extraction of voiced speech, IEEE Trans. ASSP-23, pp562-569.

T V ANANTHAPADMANABHA & B YEGNANARAYANA, (1979), Epoch extraction from linear prediction residual for identification of closed glottis interval, IEEE Trans. ASSP-27, pp309-319.

S AMARI, (1990), Mathematical foundations of neurocomputing, Proceedings of the IEEE, Volume 78, No. 9, September 1990.

F ANDERSON, (1960), An experimental pitch indicator for training deaf scholars, J. Acoust. Soc. Amer., 32, pp1065-1074.

T W ANDERSON & R R BAHADUR, (1962), Classification into two multivariate normal distributions with different covariance matrices, Ann. Math. Stat., Vol. 33, pp420-431.

J A ANDERSON & E ROSENFELD (Eds), (1988), Neurocomputing: Foundations of research, Bradford Books.

B S ATAL, (1972), Automatic speaker recognition based on pitch contours, J. Acoust. Soc. Amer., 52, 6, pp1687-1697.

B S ATAL, (1974), Effectiveness of linear prediction characteristics of the speech wave for automatic speaker identification and verification, JASA, 55, pp.1304-1312.

B S ATAL & S L HANAUER, (1971), Speech analysis and synthesis by linear prediction of the speech wave, J. Acoust. Soc. Amer., 50, pp637-655.

B S ATAL & L R RABINER, (1976), A pattern recognition approach to voiced-unvoiced-silence classification with applications to speech recognition, IEEE Trans. ASSP-24, No.3, pp.201-212.

L ATLAS, R COLE, Y MUTHUSAMY, A LIPPMANN, J CONNOR, D PARK, M EL-SHARKAWI & R J MARKS, (1990), A performance comparison of trained multi-layer perceptrons and trained classification trees, Proceedings of the IEEE, Volume 78, No. 9, September 1990.

G H BALL & D J HALL, (1965), Isodata, an iterative method of multivariate analysis and pattern classification, Proc. of the IFIPS congress.

A G BARTO, R S SUTTON & C W ANDERSON, (1983), Neuron-like adaptive elements that can solve difficult learning control problems, IEEE Transactions on Systems, Man and Cybernetics SMC-13: pp834-846.

E B BAUM & D HAUSSLER, (1989), Neural computation 1, pp151-160, MIT.

A R BARRON, (1984), Adaptive learning networks: Development and application in the United States of algorithms related to gmdh, in Self-Organizing methods in modelling, S J FARLOW, Ed., New York: Marcel Dekker Inc., 1984, pp25-65.

S P BARONIN, (1974), Determination of the fundamental tone of the voice, Electrosvyaz 5, pp50-56.

H L BARNEY, (1958), Transmission and reconstruction of artificial speech, United states patent No. 2,819,341. Issued Jan. 7, 1958; filed sept. 30, 1954.

D BLACKWELL & M A GIRSHICK, (1954, Theory of games and statistical decision, John Wiley & Sons, New York.

H D BLOCK, (1962), The perceptron: a model for brain functioning, Reviews of Modern Physics 34: pp123-135.

E de BOER & P KUYPER, (1968), Triggered correlation, IEEE Trans. on Biological and Medical Engineering, BME 15-3.

G J BORDEN & K S HARRIS, (1980), Speech science primer, Baltimore, Williams and Wilkins.

H BOURLARD & C J WELLEKENS, (1987), Multi-layer perceptrons and automatic speech recognition, IEEE First annual international conference on neural networks, San Diego, California.

H BOURLARD & C J WELLEKENS, (1989), Speech dynamics and recurrent neural networks, IEEE ICASSP, Vol. 1, Glasgow, pp33-36.

D S BROOMHEAD & D LOWE, (1988), Radial basis functions, multi-variable function interpolation and adaptive networks, RSRE memorandum 4148, SRU, RSRE, Malvern, UK.

R J BROWN, (1964), Adaptive multiple-output threshold systems and their storage capacities, Thesis, Tech. Report 6771-1, Stanford Electron. Labs., Stanford, CA June 1964.

J C CATFORD, (1964), Phonation types: The classification of some laryngeal components of speech production, In honour of Daniel Jones; eds. D ABERCROMBIE et al., (Longmans, London), pp26-37.

R M CHAMBERLAIN & J S BRIDLE, (1983), ZIP: a dynamic programming algorithm for time-aligning two indefinitely long utterances, IEEE ICASSP 816.

L W CHAN & F FALLSIDE, (1987), An adaptive training algorithm for back propagation networks, Cambridge University Engineering Department, CUED F-INFENG\TR2.

D G CHILDERS & J N LARAR, (1984), Electro-glottography for laryngeal function assessment and speech analysis, IEEE Trans. Bio. Eng. Vol. BME-31 No. 12, pp807-817.

M CHONG & F FALLSIDE, (1988), Implementation of neural networks for speech recognition on a transputer array, Cambridge University Engineering Department, CUED F-IN-SENG\TR8.

R H COLTON & E G CONTURE, (1990), Problem and pitfalls of electro-glottography, Journal of voice, Vol.4 No. 1, Raven Press Ltd, New York, pp10-24.

J W COOLEY & J W TURKEY, (1965), An algorithm for the machine calculation of complex Fourier series, Math. Computation 19, pp287-301.

P W COOPER, (1967), Some topics in non-supervised detection for multivariate normal distributions, in Computer and Information Sciences - 2 (J T TOU ed.), Academic Press, New York.

L COOPER & M COOPER, (1983), Introduction to dynamic programming, Pergamon press.

T M COVER, (1964), Classification and generalization capabilities of linear threshold units, Rome Air Develop. Centre Tech. Report RADC-TDR-64-32.

G CYBENKO, (1989), Approximation by superposition of a sigmoidal function, Mathematics of Control, Signals and Systems, Vol. 2, 1989.

P G M DAWE & M A DEUTSCH, (1955), An audio frequency meter for graphing frequency variations in the human voice, Electron. Eng., 27, 2-6.

P DELLATRE, A M LIEBERMAN, F S COOPER & L JU GERSTMAN, (1952), An experimental study of the acoustic determinants of vowel color: Observations on one- and two-formant vowels synthesized from spectrographic patterns, Word. 8, pp195-210.

M E DEMPSEY, R P SISKIND, T D HANLEY & M D STEER, (1953), A fundamental frequency recorder for complex sounds, (Purdue University, Lafayette, IN., DCC-AD-

45504).

P B DENES & E N PINSON, (1973), The speech chain, New York, Anchor books.

U DIBBERN, (1972), Fundamental frequency measurements from human speech, Plenarvortraege und kurtzeferate der Gemeinschaftstagung Stuttgart, 1972, (VDE-Verlag, Berlin), pp345-348.

L O DOLANSKY, (1954), Instantaneous pitch period indicator, J. Acoust. Soc. Amer., 26, A pp953.

L O DOLANSKY, (1955), An instantaneous pitch period indicator, J. Acoust. Soc. Amer., 27, A pp67-72.

J R DUBNO & D D DIRKS, (1989), Auditory filter characteristics and consonant recognition for hearing impaired listeners, J. Acoust. Soc. Amer., 85, pp1666-1675.

J J DUBNOWSKI, R W SCHAFER, & L R RABINER, (1976), Real time digital hardware pitch detector, IEEE Trans. ASSP-24 2-8.

R DUDA & P HART, (1973), Pattern classification and scene analysis, John Wiley & Sons, New York.

H DUDLEY, (1939), The Vocoder, Bell Sys. Tech. J., Vol. 45, pp1493-1509.

H DUIFHUIS, L F WILLEMS & R J SLUYTER, (1978), Measuring pitch in speech, IPO Annual progress report 13, pp24-30.

H DUIFHUIS, L F WILLEMS & R J SLUYTER, (1979), Pitch in speech: A hearing theory approach, ASA*50, pp245-248.

H DUIFHUIS, L F WILLEMS & R J SLUYTER, (1982), Measurement of pitch in

speech; an implementation of Goldstein's theory of pitch perception, J. Acoust. Soc. Amer., 71, pp1568-1580.

J L ELMAN & D ZIPSER, (1987), Learning the hidden structure of speech, ICS report 8701, University of California at San Diego.

G FANT, (1960), Acoustic theory of speech production, with calculations based on X-ray studies of Russian articulations, (Mouton & Company, 's-Gravenhage).

G FANT, (1970), Acoustic theory of speech production, Mouton, The Hague.

N H FARHAT, D PSALTIS, A PRATA & E PAEK, (1985), Optical implementation of the Hopfield model, Applied Optics 24: pp1469-1475.

B FARLEY & W A CLARK, (1954), Simulations of self-organizing systems by digital computer, IRE Trans. of Info. Theory, 4:pp76-84.

A FAULKNER, V BALL & A J FOURCIN, (1990), Compound speech pattern information as an aid to lipreading, Speech, Hearing and Language: Work in progress, University College London, Department of Phonetics and Linguistics, 4, pp.63-80.

C B H FELDMAN & A C NORWINE, (1958), Derivation of vocoder pitch signals, United States patent No. 2,859,405. Issued Nov. 4, 1958, filed Feb. 17, 1956.

M FILIP, (1967), Some aspects of high-accuracy analogue fundamental frequency recording, ICPhS-6, pp319-322.

M FILIP, (1969), Envelope periodicity detection J. Acoust. Soc. Amer., 45, pp719-732.

J L FLANAGAN, (1972), Speech analysis, synthesis and perception, New York, Springer-Verlag.

J L FLANAGAN & M G SASLOW, (1958), Pitch discrimination for synthetic vowels, J. Acoust. Soc. Amer., 30, pp435-442.

A J FOURCIN, (1974), Laryngographic examination of vocal fold vibration, In: Ventilatory and phonatory control systems, Ed B. Wyke, London, OUP pp315-333.

A J FOURCIN, (1979), Auditory patterning and vocal fold vibration, in B LINDBLOM & S OHMAN (eds.), Frontiers of speech communication research, London: Academic press, pp167-176.

A J FOURCIN & E ABBERTON, (1971), First applications of a new laryngograph, Med. and Biol. Illust., Vol. 21, pp172-182.

A J FOURCIN, E DOUEK, B C J MOORE, S R ROSEN, J R WALLIKER, D M HOWARD, E R M ABBERTON, & S FRAMPTON, (1983), Speech perception with promontory stimulation, An. New York Acad. Sci., 405, pp280-294.

D H FRIEDMAN, (1977), Pseudo-maximum likelihood pitch estimation, IEEE Trans. ASSP-25, pp213-221.

D H FRIEDMAN, (1979), Multichannel zero-crossing-interval pitch estimation, ICASSP-79, pp764-767.

K FUKUSHIMA, (1975), Cognitron: A self-organizing multi-layered neural network, Biological Cybernetics, 20, pp121-136.

K FUKUSHIMA, S MIYAKE & T ITO, (1983), Neocognitron: a neural model for a mechanism of visual pattern recognition, IEEE Transactions on Systems, Man and Cybernetics SMC-13: pp826-834.

K FUNAHASHI, (1989), On the approximate realization of continuous mappings by neural networks, Neural Networks, Vol. 2 pp183-192.

D GABOR, (1947), New possibilities in speech transmission, J. IEE (London), 94, 111.

S GEMAN & D GEMAN, (1984), Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images, IEEE Transactions on pattern Analysis and Machine Intelligence PAMI-6: pp721-471.

G J GIBSON & C F N COWAN, (1990), On the decision regions of Multi-layer Perceptrons, Proceedings of the IEEE, Volume 78, No. 9, September 1990.

C L GILES & T MAXWELL, (1987), Learning, invariance and generalization in higher-order neural networks, Applied Optics, Vol. 26, pp4972-4978.

J S GILL, (1962), Estimation of larynx pulse timing during speech, ICA-4, Paper G33.

B R GLASSBERG & B C J MOORE, (1990), Derivation of auditory filter shapes from notched-noise data, Hearing Research, 47, pp103-138.

B GOLD, (1962), Computer program for pitch extraction, J. Acoust. Soc. Amer., 34, pp916-921.

B GOLD, (1977), Digital speech networks, Proc. IEEE 65, pp1636-1658.

B GOLD & L R RABINER, (1969), Parallel processing techniques for estimating pitch periods of speech in the time-domain, J. Acoust. Soc. Amer., 46, pp442-448.

B GOLD & C M RADER, (1967), Systems for compressing the bandwidth of speech, IEEE Trans. Audio and Electroacoustics, Vol. AU-15, No. 3, pp131-135.

J L GOLDSTEIN, (1973), An optimal processor theory for the central formation of the pitch of complex tones, J. Acoust. Soc. Amer., 35, pp1358-1366.

S GROSSBERG, (1976), Adaptive pattern classification and universal recordings: I.

Parallel development and coding of neural feature detectors,Biological Cybernetics 23: pp121-134.

S GROSSBERG, (1980), How does a brain build a cognitive code, Psychological Review 87: pp1-51.

O O GRUENZ & L O SCHOTT, (1949), Extraction and portrayal of pitch of speech sounds, J. Acout. Soc. Amer., 21, pp487-495.

R W HAMMING, (1980), Coding and information theory, Prentice-Hall.

D O HEBB, (1949), The organizational of Behaviour, New York: Wiley.

C W HELSTROM, (1968), Statistical Theory of Signal Detection, Pergamon Press, New York.

W HESS, (1983), Pitch determination of speech signals, Springer-Verlag, Berlin.

W HESS & H INDEFRY, (1984), Accurate pitch determination of speech signals by means of a laryngograph, Proc. ICASSP-84, 1-4.

W HESS & H INDEFRY, (1987), Accurate pitch time-domain determination of speech signals by means of a laryngograph, Speech Communication 6, pp55-68.

J HOLDSWORTH, I NIMMO-SMITH, R PATTERSON & P RICE, (1988), Implementing a GammaTone Filterbank, Annex C of the SVOS Final Report, MRC APU, Cambridge.

H HOLLEIN, (1963), Fundamental frequency indicator, Am. Speech Hear. Assoc. (Paper M10; 39th Meet., ASHA).

H HOLLEIN, (1972), Three major vocal registers; a proposal, ICPhS-7, pp320-331.

J N HOLMES, (1973), The influence of glottal waveform on the naturalness of speech from a parallel formant synthesizer, IEEE Trans. Audio Electroacoustics, Volume AU-21, pp.298-305.

J N HOLMES, (1976), Formant excitation before and after glottal closure, ICASSP-76, pp39-42.

J N HOLMES, (1980), The JSRU 19-channel vocoder, IEE Proc., vol 127, part F, No. 1.

J N HOLMES, (1988), Speech synthesis and recognition, Van Nostrand Reinhold (UK).

J J HOPFIELD, (1982), Neural networks and physical systems with emergent collective computational abilities, Proceedings of the National Academy of Sciences 79: pp2554-2558.

J J HOPFIELD, (1984), Neurons with graded response have collective computational properties like those of two state neurons, Proceedings of the National Academy of Sciences 81: pp3088-3092.

D M HOWARD, (1986), Digital peak-picking fundamental frequency estimation, Speech hearing and language; Work in progress, 2, London: UCL.

D M HOWARD & A J FOURCIN, (1983), Instantaneous voice period measurement for cochlear stimulation, Elect. Letters, Vol. 19, pp776-778.

D M HOWARD, J A MAIDMENT, D A J SMITH & I S HOWARD, (1986), IEE Conf. Pub., 258, pp172-177.

D M HOWARD & G LINDSEY, (1988), Conditioned variability in voicing offsets, IEEE Trans. on ASSP, Vol. 36, No. 3.

I S HOWARD, (1990), Experiments with modular and two-level training of multi-layer perceptrons for isolated word speech recognition, Speech, Hearing and Language, Work in Progress, 1990, Vol, 4, University College London, Dept. of Phonetics and Linguistics, pp151-178.

I S HOWARD, (1991), Speech fundamental period estimation using pattern classification, Speech, Hearing and Language, Work in Progress, 1991, Vol, 5, University College London, Dept. of Phonetics and Linguistics, pp91-152.

I S HOWARD & D M HOWARD, (1986), Quantitative comparisons between time-domain speech fundamental frequency estimation algorithms, Proc. IOA, Vol 8, pp323-330.

I S HOWARD & M A HUCKVALE, (1987), The application of adaptive constraint satisfaction networks to acoustic phonetic attribute determination,Proc. Euro. Confr. Sp. Tech.

I S HOWARD & M A HUCKVALE, (1988a), Acoustic phonetic attribute determination using multi-layer perceptrons, IEE colloquium digest 11.

I S HOWARD & M A HUCKVALE, (1988b), Speech fundamental period estimation using a trainable pattern classifier, FASE88.

I S HOWARD & M A HUCKVALE, (1988c), Acoustic-phonetic feature determination for ASR, FASE88.

I S HOWARD & M A HUCKVALE, (1989), Two level recognition of isolated words using neural nets, First IEE conference on Artificial Neural Networks, London.

I S HOWARD & J R WALLIKER, (1989), The implementation of a portable real-time multi-layer perceptron speech fundamental period estimator, Proc. Eurospeech, Paris.

Y HORII, (1979), Jitter and shimmer as physical correlates of roughness in sustained phonation reexamination, J. Acoust. Soc. Amer., 66 (A), S65 (Paper EE12; 98th Meet. ASA).

W Y HUANG & R LIPPMANN, (1987), Comparisons between neural net and conventional classifiers, ICNN, San Diego, CA, 21-24 June 1987.

D H HUBEL & T N WIESEL, (1962), Receptive fields, binoclear interactions and functional architectures in cat's visual cortex, J. Physiol., London, Vol. 160, pp106-154.

D H HUBEL & T N WIESEL, (1965), Receptive fields and functional architectures in two nonstriate visual area (18 & 19) of the cat, J. Neurophysiol., Vol. 23, pp229-289.

M A HUCKVALE, (1988), Speech filing system, Part 1, SFS for users, Version 1.1, October 1988; Part 2, SFS for programmers, November 1988, Phonetics & Linguistics, University College London.

M J HUNT & C E HARVENBERG, (1986), Generation of controlled speech stimuli by pitch-synchronous LPC analysis of natural utterances, Proc. Int. Cong. Acoust., Vol. 1, Paper A4-2, Toronto.

M J HUNT, D A ZWIERZYNSKI & R C CARR, (1989), Issues in high quality LPC analysis and Synthesis, Proc. Eurospeech, Paris.

W JAMES, (1890), Psychology (Briefer course), New York; Holt, Chapter XVI, "Association", pp253-279.

E R KANDEL & J H SCHWARTZ, (1985), Principles of neural science, Elsevier, New York.

S A KIRKPATRICK, C D GELATT, Jr., & M P VECCHI, (1983), Optimization by simulated annealing, Science 220: pp671-680.

T KOHONEN, (1982), Self-organized formation of topologically correct feature maps, Biological Cybernetics 43: pp59-69.

T KOHONEN, (1984), Self-organization and associative memory, Berlin: Springer.

P LADEFORED, (1975), A course in phonetics, London, University of Chicago press.

K J LANG, A H WAIBEL & G E HINTON, (1990), A time-delay neural network architecture for isolated word recognition, Neural Networks, Volume 3, pp33-43, 1990.

W LAWRENCE, (1953), The synthesis of speech from signals which have a low information rate, Communication Theory, Ed. W Jackson, Butterworths, London, England, pp.460-469.

Y LE CUN, (1986), Learning processes in an asymmetric threshold network, Disordered Systems and Biological Organization, E Bienenstock, F Fogelman Souli, & G Weisbuch (Eds.), Berlin: Springer.

F L E LECLUSE, (1977), Electro-glottography (in Dutch), Dissertation, University Rotterdam.

M LEVINE & J SHEFNER, (1981), Fundamentals of sensation and perception, Addison-Wesley, Reading, MA.

P LIEBERMAN, (1961), Perturbation in vocal pitch, J. Acoust. Soc. Amer., 33, pp597.

P LIEBERMAN, (1963), Some acoustic measures of the fundamental periodicity of normal and pathological larynges, J. Acoust. Soc. Amer., 35, pp344-353.

C E LIEDTKE, (1971), Rechnergesteuerte Sprachzeugung, Heinrch-Hertz-Institut, Berlin; Technischer Bericht Nr. 137).

R LIPPMANN, (1987), An introduction to computing with neural nets, IEEE ASSP Magazine April 1987.

J MAKHOUL, (1975), Linear prediction; a tutorial review, Proc. IEEE 63, pp561-580.

J D MARKEL, (1972), The SIFT algorithm for fundamental frequency estimation, IEEE Trans. AU-20, pp367-377.

J D MARKEL & A H GREY, (1976), Linear prediction of speech, Communications and cybernetics, Vol. 12, Berlin, Springer.

Ph MARTIN, (1981), Detection de F_0 par intercorrelation avec une function peigne, Jounnees d'Etude sur la Parole 12, pp221-232.

Ph MARTIN, (1982), Comparison of pitch detection by cepstrum and spectral comb analysis, ICASSP-82, pp180-183.

V S MARTYNOV, (1958), Pitch determination by amplitude selection, Referenced by Baronin, 1974.

C H MAYS, (1963), Adaptive threshold logic, PhD Thesis, Technical report 1557-1, Stanford Electron. Labs, Stanford, CA, April 1963.

J MACQUEEN, (1967), Some methods for calculation and analysis of multivariate data, Proc. 5th. Berkeley Symposium on Probability and Statistics, University of California Press, Berkeley.

W S McCULLOCH & W PITTS, (1943), A logical calculus of the ideas immanent in nervous activity, Bulletin of Mathematical Biophysics 5: pp115-133.

C A McGONEGAL, L R RABINER & A E ROSENBERG, (1975), A semi-automatic pitch detector, IEEE Trans. ASSP-23, 6, pp570-574.

C A McGONEGAL, L R RABINER & A E ROSENBERG, (1977), A subjective evaluation of pitch detection methods using LPC synthesised speech, IEEE Trans. ASSP-25, pp221-229.

M McGRATH & Q SUMMERFIELD, (1985), Intermodal timing relations and audio-visual speech recognition by normal-hearing adults, JASA, 77, pp678-685.

N P McKINNEY, (1965), Laryngeal frequency analysis for linguistic research, University of Michigan, Comm. Sci. Lab. Report 14, Sept. 1965, Confr. No. 1224 (22), ER O49-122.

C MEAD, (1989), Analog VLSI and Neural Systems, Reading, MA: Addison-Wesley.

P MERMELSTEIN, (1977), On detecting nasals in continuous speech, JASA, Vol. 61 pp581.

R L MILLER, (1953), Determination of pitch frequency of complex waves, United States Patent No. 2,627,541; issued Feb. 3, 1953. Filed June 20, 1951.

N J MILLER, (1974), Pitch detection by data reduction, IEEE Symp. Speech Recognition, Paper T9, pp122-128.

N J MILLER, (1975), Pitch detection by data reduction, IEEE Trans. ASSP-23, pp72-79.

M MINSKY & S PAPERT, (1969), Perceptrons, Cambridge, MA: MIT Press.

M MOERNER, F FRANSSON & G FANT, (1964), Voice register terminology and standard pitch, STL-QPSR (4).

B C J MOORE & B R GLASBERG, (1983), Suggested formulae for calculating auditory-filter bandwidths and excitation patterns, J. Acoust. Soc. Amer., 74, pp750-753.

B C J MOORE, R W PETERS & B R GLASBERG, (1990), Auditory filter shapes at low centre frequencies, J. Acoust. Soc. Amer.

M A MOORER, (1974), The optimum comb method of pitch period analysis of continuous digitized speech, IEEE Trans. ASSP-22, pp330-338.

B MOORE & T POGGIO, (1988), Representation properties of multi-layer feed-forward networks, in Abstracts of the first annual INNS Meetings, p502, New York, Pergamon Press.

L P NGUYEN & S IMAI, (1977), Vocal pitch detection using generalized distance function associated with a voiced-unvoiced decision logic, Bull. P.M.E. (T.I.T) 39, pp 11-21.

H NEY, (1982), A time warping approach to fundamental period estimation, IEEE Trans. SMC-12 pp383-388.

N J NILSSON, (1965), Learning machines, McGraw-Hill.

A M NOLL, (1964), Short time spectrum and cepstrum techniques for vocal pitch detection, J. Acoust. Soc. Amer., 36, pp296-302.

A M NOLL, (1967), Cepstrum Pitch determination, J. Acoust. Soc. Amer., 41, pp293-309.

A M NOLL, (1970), Pitch determination of human speech by the harmonic product spectrum, the harmonic sum spectrum, and a maximum likelihood estimate. In Symposium on computer processing in communication; ed. by the Microwave Institute (Univ. Brooklyn Press, New York) Vol. 19, pp779-797.

J von NEUMANN, (1958), The computer and the brain, New Haven: Yale University Press, pp66-82.

H NYQUIST, (1928), Certain topics in telegraph transmission theory, Trans. AIEE, Vol. 47, pp617-644.

J D O'CONNOR, (1973), Phonetics, Harmondsworth, Pelican books.

J D O'CONNOR & G F ARNOLD, (1961), Intonation of colloquial english, London, Longmans.

A V OPPENHEIM & R W SCHAFER, (1975), Digital signal processing, Prentice-hall.

Y H PAO, (1989), Functional link nets: Removing hidden layers, AI Expert, pp60-68.

D PARKER, (1985), Learning logic, Tech. report TR-87, Centre for computational research in Economics and Management Science, MIT, Cambridge, MA.

D B PARKER, (1986), A comparison of algorithms for neuron-like cells, in J S DENKER (Ed.), AIP Conference Proceedings 151, Neural Networks for Computing, Snowbird Utah, AIP.

R D PATTERSON & B C J MOORE, (1986), Auditory filters and excitation patterns as representations of frequency resolution, in B C J MOORE (Ed.), Frequency Selectivity in Hearing, Academic Press.

D J B PEARCE & L C WHITAKER, (1986), Reference formant analysis, Int. confr. on Speech Input/Output; Techniques and applications, London.

S M PEELING & J S BRIDLE, (1986), Experiments with a learning network for a simple phonetic recognition task, Proc. I.O.A. Confr., Windemere.

S M PEELING, R K MOORE & M J TOMLINSON, (1986), The multi-layer perceptron as a tool for speech pattern processing research, Proc. IOA, vol.8 part 7, pp307-314, Windermere.

J B PECKHAM, (1979), A device for tracking the fundamental frequency of speech and its application in the assessment of strain in pilots and air traffic controllers. Technical report 79056; Farnborough: RAE.

E PETERSON, (1952), Analyzer for determining fundamental frequency of a complex wave, United Patent No. 2,593,695. Issued April 22, 1952; filed May 10 1948.

G E PETERSON & H L BARNEY, (1952), Control methods used in the study of vowels, J. Acoust. Soc. Amer., 24, pp175.

G E PETERSON & G G PETERSON, (1968), Fundamental frequency detector utilizing plural filters and gates, United States Patent No. 3,364,425. Issued Jan. 16, 1968.

A A PIROGOV, (1963), Synthetic telephony: In Russian, Svyazizdat, Moscow).

T POGGIO & F GIROSI, (1990), Networks for approximation and learning, Proceedings of the IEEE, Volume 78, No. 9, September 1990.

L R RABINER, (1977), One the use of autocorrelation analysis for pitch detection, IEEE Trans. ASSP-25, 23-33.

L R RABINER, M H CHENG, A E ROSENBERG & C A McGONEGAL, (1976), A comparative study of several pitch detection algorithms, IEEE Trans. ASSP-24, 5, pp399-413.

L R RABINER & M R SAMBUR, (1977), Application of an LPC distance measure to the voiced-unvoiced-silence detection problem, IEEE Trans. ASSP-25 pp338-343.

L R RABINER & R W SCHAFER, (1978), Digital processing of speech signals, Prentice-Hall.

C M RADER, (1964), Vector pitch detection, J. Acoust. Soc. Amer., 36 (C), pp1463.

D R REDDY, (1966), An approach to computer speech recognition by direct analysis of the speech wave, (Stanford University, Berkeley, CA; Tech. Rept. CS-49).

D R REDDY, (1967), Pitch determination of speech sounds, Comm. ACM 10, pp343-348.

F M REZA, (1961), An introduction to information theory, McGraw-Hill Book Co., New York.

W C RIDGWWAY III, An adaptive logic system with generalizing properties, PhD Thesis, Tech. Report 1554-1, Stanford Electron, Labs., Stanford, CA.

R R RIESZ, (1952), Apparatus for determining pitch frequency in a complex wave, United States Patent No. 2,593,698. Issued April 22, 1952; filed May 10, 1948.

R R RIESZ & L O SCHOTT, (1946), Visible speech cathode-ray converter, J. Acoust. Soc. Amer., 18, pp50-61.

A RISBERG, (1961), Statistical studies of fundamental frequency range and rate of change, STL-QPSR, (4), p7-8.

N ROCHESTER, J H HOLLAND, L H HAILBT & W L DUDA, (1956), Tests on a cell assembly theory of the action of the brain, using a large digital computer, IRE Transactions on Information Theory IT-2: pp80-93.

S ROSEN & A J FOURCIN, (1983), When less is more, Work in progress, Departments of Phonetic and Linguistics, University College, London.

S ROSEN, A J FOURCIN, J R WALLIKER, E E DOUEK & B C J MOORE, (1982), External electrical stimulation of the cochlea in the totally deaf, Uses of computers in aiding the Disables, J RAVIV (Ed.), North-Holland Publishing Company, IFIP-IMIA.

S ROSEN, B C J MOORE & A J FOURCIN, (1979), Lipreading with fundamental frequency information, Proc. Inst. Acoust. Autumn Confr. Windermere, Paper 1A2: pp5-8.

S ROSEN, J R WALLIKER, A FOURCIN & V BALL, (1987), "A microprocessor based acoustic hearing aid for the profoundly impaired listener", J. Rehab. Res. Dev., Vol. 24, pp. 239-260.

F ROSENBLATT, (1958), The perceptron: a probabilistic model for information storage and organization in the brain, Psychological review 65, pp386-408.

A E ROSENBERG & M R SAMBUR, (1975), New techniques for automatic speaker verification, IEEE Trans. ASSP-23, 2, pp169-176.

F ROSENBLATT, (1958), The perceptron: a probabilistic model for information storage and organization in the brain, Psychological Review 65: pp386-408.

M J ROSS, L H SHAFFER, A COHEN, R FREUDBERG & H J MANLEY, (1974), Average magnitude difference function pitch extractor, IEEE Trans. ASSP-22, pp353-361.

D E RUMELHART, G E HINTON & R J WILLIAMS, (1986), Learning internal representations by error propagation, Parallel Distributed Processing: Explorations in the Microstructure of Cognition, Volume 1, D E Rumelhart & J L McClelland (Eds.), Cambridge, MA: MIT Press, pp318-362.

D E RUMELHART, G E HINTON & R J WILLIAMS, (1986), Learning representations by back-propagating errors, Nature 323: pp533-536.

D E RUMELHART & D ZIPSER, (1985), Feature discovery by competitive learning, Cognitive Science 9:1.

F J SANCHEZ GONZALES, (1977a), Application of dissimilarity and a periodicity function to fundamental frequency measure of speech and voiced/unvoiced decision, ICA-9, paper I1 17, pp523.

F J SANCHEZ GONZALES, (1977b), Dissimilarity and aperiodicity functions. Temporal processing of quasi-periodic signals, ICA-9, paper 13, pp859.

M A SAPOZHKOV, (1963), The speech signal in cybernetics and communication. In Russian, Svyazizdat, Moscow.

K SCHAEFER-VINCENT, (1983), Pitch period detection and chaining: method and evaluation, Phonetica, 40: pp177-202.

J F SCHOUTEN, (1938), The perception of subjective tones. Proceedings Kon. Acad. Wetenseh. (Neth.) 41, pp1086-1094.

M R SCHROEDER, (1966), Vocoders; Analysis and synthesis of speech, Proc. IEEE, Vol. 54, pp720-734, May 1966.

M R SCHROEDER, (1968), Period histograms and product spectrum: new methods for fundamental frequency measurement, J. Acoust. Soc. Amer., 43, pp829-834.

M R SCHROEDER, (1970), Parameter estimation in speech: A lesson in unorthodoxy, Proc. IEEE 58, pp707-712.

M R SCHROEDER & B S ATAL, (1987), Code-excited linear prediction (CELP); High quality speech at very low bit rates, Proc. of ICASSP-87, pp1649-1652.

M R SCHROEDER & E E Jr. DAVID, A vocoder for transmitting 10kc/s speech over a 3.5kc/s channel, Acustica 10, pp35-43.

H J SCHULTZ-COLSON, (1975), Grundtoneanalyse - ein Beitrag zur objecktiven

Beurteilung der Sprechstimme, HNO 23, pp218-225.

O G SELFRIDGE, (1958), Pandemonium: a paradigm for learning, Mechanization of Thought Processes: Proceedings of a Symposium Held at the National Physical Laboratory, November 1958, London, HMSO, pp513-526.

P M SELLICK, R PATUZZI & B M JOHNSTONE, (1982), Measurement of basilar membrane motion in guinea pig using the Moessbauer technique, J. Acoust. Soc. Amer., 72, pp131-141.

C E SHANNON, (1968), A mathematical theory of communication, Bell System Tech. J., Vol. 27, pp623-656.

M J SHAILER, B C J MOORE, B R GLASBERG, M WATSON & S HARRIS, (1990), Auditory filter shapes at 8 and 10kHz, J. Acoust. Soc. Amer.

C P SMITH, (1954), Device for extracting the excitation function from speech signals, United States Patent No. 2,691,137, Issued Oct. 5 1954; filed June 27 1952.

C P SMITH, (1957), Speech data reduction: Voice communication by means of binary signals at rates under 1000 bits/sec (ACFRC, Bedford MA, DDC-AD-117290).

M M SONDHI, (1968), New methods of pitch extraction, IEEE Trans. AU-16, pp262-266.

D F SPECHT, Generation of polynomial discriminant functions for pattern recognition, Phd Thesis, Tech. Rept. 67645, Stanford Electron. Labs., Stanford, CA.

L STARK, M OKAJIMA & G H WIPPLE, (1963), Computer pattern recognition techniques: Electrocardiographic diagnosis, Commun. Ass. Comput. Mach., Vol. 5, pp 527-532.

K STEINBUCH & V A W PISKE, Learning matrices and their applications, IEEE Trans. Electron. Computing, Vol. EC-12. pp846-862.

H W STRUBE, (1974), Determination of the instant of glottal closure from the speech wave, J. Acoust. Soc. Amer., 56, pp1625-1629.

J SUNDBERG, (1979), Maximum speech of pitch changes in singers and untrained subjects, J Phonetics, 7, pp71-79.

E TERHARDT, (1972a), Tonhoerenwahrnehmung und harmonisches Empfinden, In Akustic and Schwingungstechnik, Stuttgart, VDE-Verlag: Berlin).

E TERHARDT, (1972b), Zur Tonhoerenwahrnehmung von Klaengen. Psychoakustische Grundlagen; II. Ein funktionsschema. Acustica ,26, pp173-199.

E TERHARDT, (1974), Pitch consonance and harmony, J. Acoust. Soc. Amer., 55, pp1061-1069.

E TERHARDT, (1979), Calculating virtual pitch, Hearing Res. 1, pp155-182.

E TERHARDT, G STOLL & M SEEWANN, (1982a), Pitch of complex signals according to virtual-pitch theory; tests, examples and predictions, J. Acoust. Soc. Amer., 71, pp671-678.

E TERHARDT, G STOLL & M SEEWANN, (1982b), Algorithm for extraction of pitch and pitch salience from complex tonal signals, J. Acoust. Soc. Amer., 71, pp679-688.

TEXAS INSTRUMENTS, (1986), Digital Signal Processing Applications with the TMS320 Family; Theory, Algorithms, and Implementations, Texas Instruments.

J T TOU, (1969), Engineering principles of pattern recognition, in Advances in Information Systems Science, Vol. 1, (J T TOU ed.), Plenum Press, New York.

J T TOU & R C GONZALEZ, (1974), Pattern recognition principles, Addison-Wesley.

W H TUCKER & R H T BATES, (1978), A pitch estimation algorithm for speech and music, Electron. Lett. 13, pp357-358.

J VAN DEN BERG, J T ZANTEMA & P DOORNENBAL, (1957), On the air resistance and the Bernoulli effect of the human larynx, J. Acoust. Soc. Amer., 29, pp626-631.

H L VAN TREES, (1968), Detection, estimation and modulation theory - Part 1, John Wiley & Sons, New York.

A WAIBEL, (1989), Neural computation, 1, No. 1, pp39-46.

A WAIBEL, T HANAZAWA, G HINTON, K SHIKANO & K LANG, IEEE Trans. ASSP-37, pp328.

J R WALLIKER, E E DOUEK, S FRAMPTON, E A ABBERTON, A J FOURCIN, D M HOWARD, S NEVARD, S ROSEN & B C J MOORE, (1985), Physical and surgical aspects of external single channel electrical stimulation of the totally deaf, Schindler R.A. & Merzenich, M.M. (Eds), Cochlear Implants, Raven Press, New York.

J R WALLIKER & I S HOWARD, (1990), Real-time portable multi-layer perceptron voice fundamental-period extractor for hearing aids and cochlear implants, Speech Communication 9, Elsevier Science Publishers B.V (North Holland), pp63-71.

J R WALLIKER, S ROSEN & A FOURCIN, (1986), Speech pattern prostheses for the profoundly and totally deaf, IEE Confr. Pub. No. 258, Int. Conf. Speech Input/Output, London, England, pp194-199.

D L WEBER, (1977), Growth of masking and the auditory filter, J. Acoust. Soc. Amer., 62, pp424-429.

J C WELLS & G COLSON, (1971), Practical phonetics, London, Pitman.

P WERBOS, (1974), Beyond regression; new tools for prediction and analysis in the behavioral sciences, PhD Thesis, Harvard University, Cambridge, MA.

B WIDROW & M E HOFF, (1960), Adapting switching circuits, 1960 IRE WESCON Convention Record, New York: IRE, pp96-104.

B WIDROW & M A LEHR, (1990), 30 Years of Adaptive Neural Networks: Perceptron, Madaline and Back-propagation, Proceedings of the IEEE, Volume 78, No. 9, September 1990.

B WIDROW, (1962), Generalization and information storage in networks of ADALINE neurons, in Self-Organizing Systems 1962, M YOVITZ , G JACOBI & G GOLDSTEIN, Eds. Washington DC: Spartan Books, pp435-461.

B WIDROW, (1987), Adaline and Madaline - 1963 Plenary Speech, Volume 1: Proc. IEEE, 1st Int. Confr. on Neural Networks, San Diego, CA, pp143-158.

B WIDROW & M E HOFF, (1960), Adapting switching circuits, 1960 IRE WESCON Convention Record, New York: IRE, pp96-104.

B WIDROW & M A LEHR, (1990), 30 Years of Adaptive Neural Networks: Perceptron, Madaline and Back-propagation, Proceedings of the IEEE, Volume 78, No. 9, September 1990.

F L WIGHTMAN, (1973), The pattern transformation model of pitch, J. Acoust. Soc. Amer., 54, pp407-416.

J D WISE, J D CAPRIO & T D PARKS, (1976), Maximum likelihood pitch estimation, IEEE Trans. ASSP-24, pp418-423.

R WILLIS, (1829), Trans. Camb. Soc., Vol. 3, Pt. 1, pp231.

L A YAGGI, (1962), Full duplex digital vocoder. Scientific report No. 1 (Texas Instruments, Dallas, TX; Report No. sp14-A62; DCC-AD -282986).

L A YAGGI, (1963), Full duplex digital vocoder. Final report No. 2 (Texas Instruments, Dallas, TX; Report No. sp16-A63).

E ZWICKER, W HESS & E TERHARDT, (1967), Erkennung gesprochener Zahlworte mit Funktionsmodell und Rechenanlage, Kybernetik 3, pp267-272.

E ZWICKER & R FELDKELLER, (1967), Das Ohr als Nachrichtenempfaenger (Hirzel, Stuttgart).

APPENDIX A.1: PATTERN PROCESSING SYSTEM

The pattern processing system provides a convenient means of performing pattern processing operations on speech items stored in sfs format (Huckvale, 1988). It was felt important to make the system as flexible as possible with regard to data vector generation and the type of pattern classification algorithms that could be used.

The requirement for the generation of pattern vectors are:

1) It must be possible to generate input vectors using different input coefficient items at the same time, each resulting perhaps from different types of preprocessing routines. Thus it should be possible to build a pattern vector with elements composed from frames from coefficient items with different frame rates.

2) It must be possible to specify the number of frames from each coefficient item separately,as well as the offset of the window. This will allow different amounts of context on the different input items to be implemented.

It was felt important to ensure that the system would lend itself to the use of a variety of pattern recognition techniques.

System description

The system is composed of three main sections:

1) Data formatting and sorting programs that generate data vectors, and their target patterns, of appropriate format for use with the pattern recognition programs.

2) Pattern recognition programs that operate on the data vectors either in learn mode, recognition mode or test mode.

3) Conversion programs that reformat the results of processing back into a suitable

format for use.

The input items required by the pattern processing system are data in the form of coefficient items and in the case of learn mode, target annotations.

The input data to the pattern processing system consists of items of coefficient data. More than one coefficient item may be used in the generation of the input vector, each with different frame durations, provided that the sampling rates are all integer multiples of the shortest frame duration. Each item requires specification of how many of its frames are used at once in the data vector. This constitutes the width of the observation window on each item. In addition, the offset of the window must be specified.

The use of multiple input items with potentially different frame rates is valuable because it permits the construction of pattern vectors with elements that change on different time scales. Thus one may use a fast frame rate item with another slower item where the latter provides a wider context to the classifier than would otherwise be possible.

The target pattern must be supplied in the form of a track item that has the same sampling rate as the highest sampling rate coefficient item. The target track item can be generated by the program antr, which maps the target annotations to the desired feature track.

In order to label the pattern vectors with information needed by the sorting program, there should also be a suitable annotation item in the sfs file. This can, of course, be the same as the annotation used to generate the target track item.

Preparation of data vectors for the pattern classifier is then carried out using the two programs pform and psort.

The program pform is a formatter program, the input of which consists of two files.

The first is a sfs data file containing the input data coefficient items, target track item

and label annotations.

The second is a text parameter file containing information indicating which coefficient, annotation and track items should be used as input. This file has the name <featurename.par>.

The output from the program consists of three files:

An identification file <featurename.info>.

A binary data file composed from the input coefficient data items <featurename.vec>.

An ascii labels file <featurename.LAB> that contains information on how to generate the pattern vectors from the <featurename.vec> file as well as the target pattern class for the respective vectors and information useful for sorting.

The ascii labels file <featurename.LAB> is a list of structures that specify how to generate a data vector from the binary <featurename.vec> file. Each structure has five entries:

1) annotation labels.
The first entry consists of a concatenation of the specified annotations that were valid at that point in time.

2) Target value.
The second entry is the target value for the pattern vector, as specified by the track item. In the case of a binary classifier, this value is either 0 or 1.

3) Offset.
The third entry is an offset (in bytes) which points into the binary data file and specifies the data value in the pattern vector.

4) Size.

The fourth entry is the number of bytes forward of the offset that are also in the pattern vector.

5) numsets.

The fifth and final entry is a number which specifies how many of these structures are needed to specify the pattern vector. If the number is greater than one, the next (numsets-1) structures point to data that is also in the same pattern vector, and their numsets values is always 0.

Control of the formatter is achieved by means of the parameter text file <featurename.par>, the format of which is given below. This text file specifies which input coefficients are used as data inputs, and how the data vector is constructed from the data items. In addition this file specifies the track item used to specify the target values for the pattern vectors. The annotation item to be used in sorting is also specified here.

The parameter text file < featurename.par> has the following format:

number_of_input_coefficient_items <number>

input_coefficient_match <match>

elements_in_frame <width>

number_of_frames_in_window <frames>

window_frames_offset <offset>

output_track_target_match <match>

number_of_label_annotations <value>

label_annotation_match <match>

These lines have the following meaning:

line 1 number_of_input_coefficient_items <number>

This specifies the number of coefficient items that will be read from the sfs datafile. If <number> is greater than one, then lines 2,3,4 and 5 must be repeated once for each

coefficient item.

line 2 input_coefficient_match <match>

This string <match> specifies which coefficient item will be read from the sfs file. It follows the sfs matching convention.

line 3 elements_in_frame <width>

The value of <width> is the number of elements in one frame of the coefficient data.

line 4 number_of_frames_in_window <frames>

The value of <frames> specifies how many contiguous frames of the coefficient data are used in the pattern vector.

line 5 window_frames_offset <offset>

The value of <offset> determines how far, in frames, the output is from the start of the window.

line 6 output_track_target_match <match>

The string <match> specifies the track item that will be read in and used as the target pattern value for the pattern vectors.

line 7 number_of_label_annotations <number>

This specifies the number of annotation items that will be read from the sfs data file. If <number> is greater than 1, then line 8 must be repeated for each annotation item.

line 8 label_annotation_match <match>

The string <match> specifies the annotation item that will be read in which may then be useful for sorting by the psort program.

The psort program has the option either to write or append to the files. In case where one has a large number of files for use with the pattern processing system, one would use the write mode for the first file and the append mode for the remainder.

The order of presentation of groups of data vectors is important for certain iterative trainable pattern classifier, such as the multi-layer-perceptron. It is therefore necessary to have the ability to sort the data vectors in order to generate groups that are as representative of the entire data set as possible. This is achieved by means of the program psort.

The program psort is a sorting or copying program.
The input to the psort program consists of two files:

An ascii labels file <featurename.LAB> generated by pform.
A text identification file <featurename.info> generated by pform.

The output form the psort program consists of two files:

A binary labels file <featurename.lab>

The text file <featurename.info>, which was the file generated by the program pform, is appended with information about the sorting process.

The next stage is to run a pattern classification algorithm on the data. At present only a multi-layer perceptron algorithm has been implemented, but it is intended to have a selection of standard algorithms as options in the near future.

The program mlpw is a general purpose mlp classifier that can operate in three modes: Learn mode, recognition mode or test mode.

The program expects details of the mlp network to be contained in a configuration file <mlpmodel.mlp>. This file contains the specification of initial network configuration and the weights that are determined during learn mode. The pattern vectors are read from the binary files <featurename.vec> using the specifications in <featurename.lab>.

The configuration file has the following format:

configuration
<layer0>-<layer1>-..-<layerN>
history
<cycle> <normalized error> <date>
layer 1
<connection> <weight> <delta>
layer 2
<connection> <weight> <delta>
layer 3
<connection> <weight> <delta>

Initial setup is achieved by specifying the required configuration and then the interconnections for each layer. If full interconnect is required, this is specified as <full>. For example for a 19 input, no hidden layer and 1 output mlp with full interconnect, the initial configuration file would be:

configuration
19-1
layer 1
full

In learn mode the mlp model in the configuration files is updated using the data in the binary files <featurename.lab> and <featurename.vec>. There are currently two adaptation methods available: method 0 (standard adaption) which uses fixed α and Γ values; method 1 (Lai-Wan adaption) which used changing α and Γ values, which are chosen as a function of the direction of adaption in parameter space.

In test mode, the mlp specified model is used to generate outputs for each vector which are then compared with the supplied target label values. Performance statistics are printed on the basis of the location of a threshold that results i equal miss-rate and false-alarm rate. The receiver operating characteristic (ROC) may be generated by selecting the <-s> option.

In recognition mode, the mlp model in the configuration file is used to generate output values for the input vectors. The input labels and recognised outputs are written to the files <featurename.mlp.i> and <featurename.mlp.r> respectively.

APPENDIX A.2: COMPUTER ANALYSIS PROCEDURES

ACQUISITION AND LABELLING OF THE DATA

This section contains a complete list of execution of programs commands used to prepare the data for either training or testing purposes.

Stage 1: Acquisition of the speech and laryngograph data

The speech and laryngograph signal for the subjects was previously recorded onto DAT tape. The output of each channel of the DAT recorder was fed into a 8th order Butterworth low-pass filter with a 3.5kHz cutoff frequency. The two signal channels were then connected to the A/D converts on the Masscomp 5600 computer. Acquisition was carried out using the inwd program, which acquires speech and laryngograph into a SFS file that has previously had its header set-up using the hed command. For example, the hed command line for one occasions was:

```
hed tmd.far1
```

The inwd program was then executed using the command line:

```
inwd -f8000.0 -l -m8 tmd.far1
```

This results in speech and laryngograph being saved into the file tmd.far1. The summary of the file is as follows:

```
1. SPEECH   (1.01)153600     frames from inwd/SP(freq=8000,linked)
2.     LX   (2.01)153600     frames from inwd/LX(freq=8000,linked)
```

Stage 2: Generation of reference period markers

The reference period markers are generated by analysis of the laryngograph in the SFS

file by means of two programs: The first of these is lxtx, which generates a bandpass filtered version of the laryngograph, and a differentiated version of this bandpass filtered laryngograph. In addition, it generated a first approximation to the period marker locations, by estimating the noise threshold in the laryngograph signal, and using it to set the reference value of a comparator that operates on the differentiated laryngograph signal.

In order to estimate the background noise in the laryngograph signal, it is necessary to annotate the speech and laryngograph signal with regions that are only laryngograph noise. This is done using the standard interactive Es program, using the command line:

Es -lsil tmd.far1

After running this, there is an annotation item saved into the SFS file that can be used by the lxtx program to identify laryngograph noise. Thus the lxtx program is run by using the command line:

lxtx -s -i5.01 tmd.far1

The next stage is to run the interactive program lxia to permit manual cleaning-up of the period marker estimates. In addition, this program allows the placement of annotations to denote that the laryngograph period markers estimate is unreliable in certain regions, and should not be used for training of for testing. Thus, the lxia program is run using the command line:

lxia -S1.01 -L16.01 -D16.02 -T3.01 tmd.far1

This then results in a hand-checked estimate of the period marker placements as well as annotations to indicate regions of the signal that must be rejected. The summary of the SFS file is now:

1. SPEECH (1.01)153600 frames from inwd/SP(freq=8000,linked)

2. LX (2.01)153600 frames from inwd/LX(freq=8000,linked)

3. ANNOT (5.01) 5 frames from Es/AN(type=sil)

4. TRACK (16.01)153600 frames from lxtx(2.01;firproclx;delay=0)

5. TRACK (16.02)153600 frames from lxtx(2.01;diffLX;delay=0)

6. TX (3.01) 2114 frames from lxtx(2.01;tx=maxdiff;thresh=55.8742;delay=0)

7. ANNOT (5.02) 67 frames from lxia(type=r+/r-)

8. TX (3.02) 2225 frames from lxia(16.02,3.01)

Stage 3: Alignment of the reference period markers to the speech.

Because the time-delay between the laryngograph signal and the speech is different for different recordings made at different distances (and different vocal tract lengths) it is necessary to align all the reference period markers and its corresponding speech signal so that it is the same for all recordings. This is achieved by running a partially (or fully trained) MLP-Tx algorithm operating at the full 8kHz sampling rate on the speech in a given file. The corresponding period markers for this are then generated using the trtx program, and the delay between this and the reference period markers is computed. This is then used to time-shift the reference period markers to align with the speech in the desired fashion. The alignment and shifting of the reference period markers is carried out using the program align.

Thus, the three steps involved in aligning the reference period markers to the speech are:

1) Run the preprocessor for the MLP-Tx algorithm, that is

preproc -i1.01 agc.par tmd.far1

2) Run an 8kHz MLP-Tx algorithm on this pre-processed speech, that is

mlptr -w161 -O80 -i16.03 -s1.0 -heavb103 eavb103 tms.far1

3) Generate the period markers from the MLP-Tx algorithm

trtx -t0.4 -i16.04 -f500 tmd.far1

4) linearly align the reference period markers to the MLP-Tx tx:

align -i3.03 -i3.02 tmd.far1

The summary of the SFS file is now:

1. SPEECH (1.01)153600 frames from inwd/SP(freq=8000,linked)
2. LX (2.01)153600 frames from inwd/LX(freq=8000,linked)
3. ANNOT (5.01) 5 frames from Es/AN(type=sil)
4. TRACK (16.01)153600 frames from lxtx(2.01;firproclx;delay=0)
5. TRACK (16.02)153600 frames from lxtx(2.01;diffLX;delay=0)
6. TX (3.01) 2114 frames from lxtx(2.01;tx=maxdiff;thresh=55.8742;delay=0)
7. ANNOT (5.02) 67 frames from lxia(type=r+/r-)
8. TX (3.02) 2225 frames from lxia(16.02,3.01)
9. TRACK (16.03)153600 frames from preproc(1.01;parameterfile=agc.par;scale=1;output)
10. TRACK (16.03)153600 frames from mlptr(16.03;mlp=eavb103,window=161,offset=80,output=0,sc=1,eavb103)
11. TX (3.03) 1825 frames from trtx(16.04;threshold=0.4;maxpulse;maxfx=500)
12. TRACK (16.05) 1600 frames from align(type=txaligned;3.03,3.02;offset=0.1)
13. TX (3.04) 2225 frames from align(type=txdata;alignedby=3.03,from=3.02;offset=0.004375)

stage 4: Removing unnecessary items in the SFS file

The data is now prepared for either training or testing purposes, and all the items that

are no longer required are then removed. This is achieved by running the SFS remove utility program:

remove -i 16.01 -i16.02 -i 16.03 -i16.04 tmd.far1

The SFS file summary thus gives:

1. SPEECH (1.01)153600 frames from inwd/SP(freq=8000,linked)
2. LX (2.01)153600 frames from inwd/LX(freq=8000,linked)
3. ANNOT (5.01) 5 frames from Es/AN(type=sil)
4. TRACK -(16.02)153600 frames from lxtx(2.01;diffLX;delay=0)
5. TX (3.01) 2114 frames from lxtx(2.01;tx=maxdiff;thresh=55.8742;delay=0)
6. ANNOT (5.02) 67 frames from lxia(type=r+/r-)
7. TX (3.02) 2225 frames from lxia(16.02,3.01)
8. TRACK -(16.03)153600 frames from preproc(1.01;parameterfile=agc.par;scale=1;output)
9. TRACK -(16.03)153600 frames from mlptr(16.03;mlp=eavb103,window=161,offset=80,output=0,sc=1,eavb103)
10. TX (3.03) 1825 frames from trtx(16.04;threshold=0.4;maxpulse;maxfx=500)
11. TRACK (16.05) 1600 frames from align(type=txaligned;3.03,3.02;offset=0.1)
12. TX (3.04) 2225 frames from align(type=txdata;alignedby=3.03,from=3.02;offset=0.004375)

Further processing of the data then differs depending on whether the data will be used for training or testing purposes. The processing is also different depending on the different pre-processing and post-processing operations that are carried out.

PREPARING DATA FOR TRAINING THE MLP DIRECTLY ON THE SPEECH PRESSURE WAVEFORM

The data for training the MLP directly on the speech pressure waveform was generated by simply scaling the amplitude of the input speech and then generating sorted pattern vector files of the appropriate format. These would then be used by the mlpwe program.

stage 1

The first processing step was to scale the speech to an input range of -1.0 to +1.0. This was achieved using the preproc program by executing the command line:

preproc -i1.01 agc.par tmd.far1

The parameters for this program are contained in the parameter file agc.par. This file contains the following information:

pre_processing_parameters
input_scaling_factor 1.0
automatic_gain_control
no_iir_filter
no_iir_filter
no_rectifier
no_iir_filter
decimation_factor 1
linear_output
output_scaling _factor 0.001
no_automatic_gain_control
end_of_parameters

stage 2

The target labels for the speech data are specified using the program txtar. This program write out an output track that specifies the identity of the regions around a period marker. The widths of the zones are specified in the parameter file txtar.sp This

parameter file contains the following information:

txtar_parameters
tx_pulse_width 1
pre_uncertain_width 3
post_uncertain_width 4
voice_width 160
end_of_parameters

The txtar program is executed using the command line:

txtar -f8000.0 -i3.04 txtar.sp tmd.far1

Notice that for operation directly on the speech, an output sampling rate of 8kHz is specified.

Stage 3

The input speech has now been suitably scaled and the output target labels have been specified. The next stage is to generate the pattern vector files used by the mlpwe program.

The initial formatting process in carried out using the program pform. The operation of this program is specified by the generic parameter file md2far1.par. It contains he following information:

annotation_target_mode
input_track_data
elements_in_frame 1
input_track_match preproc(*;*agc.par*)
number_of_frames_in_window 161
window_frames_offset 80

```
number_of_track_outputs 1
output_track_target_match txtar(*;pars=txtar.sp;rc=regionid;*)
number_of_label_annotation_items 2
label_annotation_match lxia(type=r+/r-)
label_annotation_match txtar(*;pars=txtar.sp;rc=ann;*)
end_of_parameters
```

The command line to execute the formatting program is:

```
pform -s1.0 md2far1
```

This generates a ascii labels file md2far1.LAB and an input vectors file md2far1.vec. The order of occurrence of the vectors is the same as they occurred in the input SFS file. In order to sort them into more representative groups, the program psort must be used. Sorting in performed on the basis of the annotation labels specified in the pform parameter file; for the work here, the sorting annotations were simple period marker annotations. This is executed using the command line:

```
psort -s md2far1
```

This results in a sorted labels file md2far1.lab, which is used together with the vectors file md2far1.vec, by the mlpwe program.

PREPARING DATA FOR TRAINING THE MLP USING WIDEBAND FILTERBANK

The data for training the MLP on the output of a wideband filterbank was generated by running a filterbank program and then generating sorted pattern vector files of the appropriate format. These would then be used by the mlpwe program.

stage 1

The first processing step was to filter the speech into the required number of bands and

then to scale the outputs to an input range of -1.0 to +1.0. For each channel in the filterbank analysis, this was achieved using the preproc program by executing the command line (with the appropriate parameter file for a given channels):

preproc -i1.01 c1tms.par tmd.far1

The parameters for this program are contained in the parameter file c1tms.par. This file contains the following information:

pre_processing_parameters
input_scaling_factor 1.0
no_automatic_gain_control
iir_filter_order 2
denominator_coeff_a0 1.0
denominator_coeff_a1 -1.944336E+0
denominator_coeff_a2 9.459229E-01
numerator_coeff_b1 9.725647E-01
numerator_coeff_b0 -1.945129E+00
numerator_coeff_b2 9.725647E-01
iir_filter_order 2
denominator_coeff_a0 1.0
denominator_coeff_a1 -1.166943E+0
denominator_coeff_a2 7.166748E-01
numerator_coeff_b0 1.181030E-02
numerator_coeff_b1 2.362061E-02
numerator_coeff_b2 1.1811030E-02
half_wave_rectifier
iir_filter_order 2
denominator_coeff_a0 1.0
denominator_coeff_a1 -9.428711E-01
denominator_coeff_a2 3.333740E-01
numerator_coeff_b0 9.762573E-02

numerator_coeff_b1 1.952515E-01
numerator_coeff_b2 9.762573E-02
decimation_factor 4
linear_output
output_scaling _factor 0.025
no_automatic_gain_control
end_of_parameters

stage 2

The target labels for the speech data are specified using the program txtar. This program write out an output track that specifies the identity of the regions around a period marker. The widths of the zones are specified in the parameter file txtar.fb This parameter file contains the following information:

txtar_parameters
tx_pulse_width 1
pre_uncertain_width 1
post_uncertain_width 1
voice_width 40
end_of_parameters

The txtar program is executed using the command line:

txtar -f2000.0 -i3.04 txtar.fb tmd.far1

Notice that for operation on the output of the filterbank, an output sampling rate of 2kHz is specified.

Stage 3

The input speech has now been suitably filtered and scaled, and the output target labels

have been specified. The next stage is to generate the pattern vector files used by the mlpwe program.

The initial formatting process in carried out using the program pform. The operation of this program is specified by the generic parameter file md2far1.par. It contains he following information:

```
annotation_target_mode
input_track_data
elements_in_frame 6
input_track_match preproc(*;*c1tms.par*)
input_track_match preproc(*;*c1tms.par*)
input_track_match preproc(*;*c1tms.par*)
input_track_match preproc(*;*c1tms.par*)
input_track_match preproc(*;*c1tms.par*)
input_track_match preproc(*;*c1tms.par*)
number_of_frames_in_window 41
window_frames_offset 20
number_of_track_outputs 1
output_track_target_match txtar(*;pars=txtar.fb;rc=regionid;*)
number_of_label_annotation_items 2
label_annotation_match lxia(type=r+/r-)
label_annotation_match txtar(*;pars=txtar.fb;rc=ann;*)
end_of_parameters
```

The command line to execute the formatting program is:

```
pform -s1.0 md2far1
```

This generates a ascii labels file md2far1.LAB and an input vectors file md2far1.vec. The order of occurrence of the vectors is the same as they occurred in the input SFS file. In order to sort them into more representative groups, the program psort must be

used. Sorting in performed on the basis of the annotation labels specified in the pform parameter file; for the work here, the sorting annotations were simple period marker annotations. This is executed using the command line:

psort -s md2far1

This results in a sorted labels file md2far1.lab, which is used together with the vectors file md2far1.vec, by the mlpwe program.

PREPARING DATA FOR TRAINING THE MLP USING AN AUDITORY FILTERBANK

The data for training the MLP on the output of a auditory filterbank was generated by running a filterbank program and then generating sorted pattern vector files of the appropriate format. These would then be used by the mlpwe program.

stage 1

The first processing step was to filter the speech into the required number of bands and then to scale the outputs to an input range of -1.0 to +1.0. For each channel in the filterbank analysis, this was achieved using the coch program to filter the speech and then the preproc program to half-wave rectify, smooth and decimate the outputs. The initial filtering was performed using the command line:

coch -l50 -h3500 -i1.01 tmd.far1

The half-wave rectification was then performed using the command line:

preproc -i1.01 env0.par tmd.far1

The parameters for this program are contained in the parameter file env0.par. This file contains the following information:

pre_processing_parameters
input_scaling_factor 1.0
no_automatic_gain_control
no_iir_filter
no_iir_filter
half_wave_rectifier
iir_filter_order 2
denominator_coeff_a0 1.0
denominator_coeff_a1 -9.428711E-01
denominator_coeff_a2 3.333740E-01
numerator_coeff_b0 9.762573E-02
numerator_coeff_b1 1.952515E-01
numerator_coeff_b2 9.762573E-02
decimation_factor 4
linear_output
output_scaling _factor 0.025
no_automatic_gain_control
end_of_parameters

stage 2

The target labels for the speech data are specified using the program txtar. This program write out an output track that specifies the identity of the regions around a period marker. The widths of the zones are specified in the parameter file txtar.fb This parameter file contains the following information:

txtar_parameters
tx_pulse_width 1
pre_uncertain_width 1
post_uncertain_width 1
voice_width 40
end_of_parameters

The txtar program is executed using the command line:

txtar -f2000.0 -i3.04 txtar.fb tmd.far1

Notice that for operation on the output of the filterbank, an output sampling rate of 2kHz is specified.

Stage 3

The input speech has now been suitably filtered and scaled, and the output target labels have been specified. The next stage is to generate the pattern vector files used by the mlpwe program.

The initial formatting process in carried out using the program pform. The operation of this program is specified by the generic parameter file md2far1.par. It contains he following information:

```
annotation_target_mode
input_track_data
elements_in_frame 6
input_track_match preproc(*;*env0.par*)
input_track_match preproc(*;*env1.par*)
input_track_match preproc(*;*env2.par*)
input_track_match preproc(*;*env3.par*)
input_track_match preproc(*;*env4.par*)
input_track_match preproc(*;*env5.par*)
number_of_frames_in_window 41
window_frames_offset 20
number_of_track_outputs 1
output_track_target_match txtar(*;pars=txtar.fb;rc=regionid;*)
number_of_label_annotation_items 2
label_annotation_match lxia(type=r+/r-)
```

label_annotation_match txtar(*;pars=txtar.fb;rc=ann;*)
end_of_parameters

The command line to execute the formatting program is:

pform -s1.0 md2far1

This generates a ascii labels file md2far1.LAB and an input vectors file md2far1.vec. The order of occurrence of the vectors is the same as they occurred in the input SFS file. In order to sort them into more representative groups, the program psort must be used. Sorting in performed on the basis of the annotation labels specified in the pform parameter file; for the work here, the sorting annotations were simple period marker annotations. This is executed using the command line:

psort -s md2far1

This results in a sorted labels file md2far1.lab, which is used together with the vectors file md2far1.vec, by the mlpwe program.

TRAINING THE MLP ON THE PATTERN VECTOR FILES

A MLP is trained on a pattern vector file using the program mlpwe. The selective emphasis training is selected using the -S flag on the mlpwe program, and it requires the specification of the different zone thresholds and emphasis values. These are specified in the generic parameter file spfvf.mp. The file for normal selective emphasis training contains the following information:

```
mlpwe_parameters
number_of_zones     5
target_zone0        0.0
hce_zone0           1.0
lce_zone0           0.0
```

thresh_zone0	0.1
target_zone1	0.0
hce_zone1	1.0
lce_zone1	0.0
thresh_zone1	0.85
target_zone2	1.0
hce_zone2	0.0
lce_zone2	1.0
thresh_zone2	0.9
target_zone3	0.0
hce_zone3	1.0
lce_zone3	0.0
thresh_zone3	0.85
target_zone4	0.0
hce_zone4	1.0
lce_zone4	0.0
thresh_zone4	0.1
end_of_parameters	

A typical training run consists of training at the update of 1 for the first part of the training, then increasing the update to 100 and finally to 1000. About 60 cycles over the vector file or files is typically carried out, and this takes about 3 days for the system operating directly on the speech pressure waveform with a suitable MLP configuration; any more cycles than this takes too long to run. Using the pattern vector file md2ar1 and the MLP sp4fv, this can be achieved using the command lines:

```
mlpwe -L1 -S -e -u1 -a0 -c60 md2ar1 sp4fv
mlpwe -L1 -S -e -u100 -a2 -c60 md2ar1 sp4fv
mlpwe -L1 -S -e -u1000 -a2 -c60 md2ar1 sp4fv
```

TESTING THE MLP ON THE PATTERN VECTOR FILES

It is possible to test the performance of an MLP on a pattern vector file using the mlpwe program. For example, to test the MLP spjd11 on the vectors file jd1ar8, the command line would be:

```
mlpwe -S -L1 -t jd1ar1 spjd11
```

The resulting output from the program is as follows:

```
Specialmode testing
Patterns aligned along 1 channels
MLP model       :    spjd11
Labels file     :    jd1ar8
Cycles          :    9
Time            :    Thu Dec 20 23:44:46 1990

zone0   tar=  0   thresh=   0.1    percenthits=   99.91   hits=54917/54965
zone1   tar=  0   thresh=   0.85   percenthits=   82.68   hits=2081/2517
zone2   tar=  1   thresh=   0.9    percenthits=   49.94   hits=419/839
zone3   tar=  0   thresh=   0.85   percenthits=   64.52   hits=2160/3348
zone4   tar=  0   thresh=   0.1    percenthits=   93.46   hits=57260/61268

zone0   tar=  0   thresh=   0.5    percenthits=   99.99   hits=54961/54965
zone1   tar=  0   thresh=   0.5    percenthits=   65.28   hits=1643/2517
zone2   tar=  1   thresh=   0.5    percenthits=   76.40   hits=641/839
zone3   tar=  0   thresh=   0.5    percenthits=   40.68   hits=1362/3348
zone4   tar=  0   thresh=   0.5    percenthits=   98.60   hits=60409/61268
```

The output gives the zone identification, and the target for the zone. Then, a threshold is indicated, and the percentage patterns that are above of below the threshold (depending on the pattern target class are given, as well as the absolute hits and the total patterns for that zone. The analysis is carried out for the training thresholds and also for the 'usable' threshold of 0.5, which is typically what would be used in real-

operation. Consequently the latter gives a good estimate of real usable performance.

GENERATING RECOGNITION OUTPUT FROM THE MLP

To generate a recognition output of a trained MLP, the most general technique is to use the mlpwe program operating on an unsorted pattern vector file. For example, using the MLP spjd11 and the pattern labels file jd1ar1, this generates the generic output ascii file jd1ar1.mlp.r, in which the output order reflect the order of the labels in the input pattern vector file. Because of this, to maintain the time-order of the input speech, the psort program must be run with the -C option, so that no pattern class sorting is carried out. Thus, having formatted the patterns as described before using the pform program, the psort program is run as follows:

```
psort -C jd1ar1
```

The mlpwe program is then run using the command line:

```
mlpwe -S -L1 -r jd1ar1 spjd11
```

The output file jd1ar1.mlp.r can then be loaded into a SFS file using the trload program. In the case of operation at a 2kHz sampling rate with a 10ms delay, loading into the SFS file jd1.far1 will be accomplished using the command line:

```
trload -f2000 -hspjd11 -O0.01  jd1.far1
```

POST-PROCESSING OF RECOGNITION TRACKS

A recognition track in a SFS file can be used as the input to another MLP, by using the formatting, training and recognition techniques previously described. This is carried out by specifying direct operation on the track. The remaining procedure is the same as that followed for operation directly on the speech pressure waveform.

A recognition track can also be converted to a excitation epoch marker (Tx) using the program trtx. This performs a simple threshold analysis of the input track, with a specified inhibition window around any detected period markers to reduce spurious detections.

A typical command line for the trtx program is as follows:

trtx -i16.05 -t0.5 -f500 jd1.far1

This programs writes out a Tx item into the specified SFS file. These period marker values can be converted to a frequency contour using the program fx. The input items o this program are the Tx items to be converted, and the frame rate of the output Fx contour. An example command line for this program is:

fx -f100 -i3.04 jd1.far1

FREQUENCY CONTOUR (Fx) COMPARISONS

Comparisons between a test frequency contour and a reference frequency contour can be carried out using the program fxcomp. The two inputs must be in a SFS file the form of frequency contours.

For the SFS file ejt.frp2 with a reference Fx contour item 4.01 and test Fx item 4.02, a typical command line for the fxcomp program is:

fxcomp -r4.01 -t4.02 -i5.02 -p50 -f50 ejt.frp2

The options -p50 and -f50 results in the program searching forward and backwards by 50 frames to find the best fit between the reference an test Fx contours. This is in general necessary, because of different time delays between different algorithms. The rejection annotation item 5.02 is selected and this results in the analysis being carried out for all the annotation labels, and in additions, for each label separately. The only

valuable results are given for the annotation label r-, which indicates a valid reference period markers in that region. The output from the algorithm is written to the file ejt.frp2.fxs, and this file contains the following information:

++

Filename ejt.frp2

reference item = 01

reference history = fx(3.04;m40,M800,f100)

test item = 01

test history = fx(3.03;m40,M800,f100)

Best offset = 0

ANNOTATION LABEL [0] = all

****

Reference data

Voiced frames = 714/1679 Percent = 42.5253

Unvoiced frames = 965/1679 Percent = 57.4747

Test data

Voiced frames = 725/1679 Percent = 43.1805

Unvoiced frames = 954/1679 Percent = 56.8195

Voicing to no-voicing errors = 75/714 Percent = 10.5042

No-voicing to voicing errors = 86/965 Percent = 8.91192

Gross errors = 231/639 Percent = 36.1502

F0 doubling errors = 37/639 Percent = 5.7903

Fine errors = 408/639 Percent = 63.8498

Fine mean = -0.112745

Fine std = 5.56442

ANNOTATION LABEL [1] = lost

****

No results for this annotation

ANNOTATION LABEL [2] = r-

****

Reference data

Voiced frames = 679/1469 Percent = 46.2219

Unvoiced frames = 790/1469 Percent = 53.7781

Test data

Voiced frames = 695/1469 Percent = 47.3111

Unvoiced frames =774/1469 Percent = 52.6889

Voicing to no-voicing errors = 59/679 Percent = 8.68925

No-voicing to voicing errors = 75/790 Percent = 9.49367

Gross errors = 224/620 Percent = 36.129

F0 doubling errors = 35/620 Percent = 5.64516

Fine errors = 396/620 Percent = 63.871

Fine mean = -0.0555556

Fine std = 5.57156

ANNOTATION LABEL [3] = r+

****

Reference data

Voiced frames = 35/210 Percent = 16.6667

Unvoiced frames = 175/210 Percent = 88.3333

Test data

Voiced frames = 30/210 Percent = 14.2857

Unvoiced frames = 180/210 Percent = 85.7143

Voicing to no-voicing errors = 16/35 Percent = 45.7143

No-voicing to voicing errors = 11/175 Percent = 6.28571

Gross errors = 7/19 Percent = 36.8421

F0 doubling errors = 2/19 Percent = 10.5263

Fine errors = 12/19 Percent = 63.1579

Fine mean = -2

Fine std = 4.96655

TIME OF EXCITATION MARKER (Tx) COMPARISONS

Comparisons between a test Tx item and a reference Tx item can be computed using the program dpalign. The name is indicative of the dynamic programming procedure used

by the program to align the test and reference period markers.

For the SFS file ejt.frp2 with the reference Tx item 3.04 and the test Tx item 3.03, the appropriate command line to run the analysis is:

dpalign -p1.0 -r3.04 -t3.03 -i5.02 ejt.far2

The program writes its analysis to a file ejt.far2.txs. The contents of this file is shown below. The rejection annotation item 5.02 is selected and this results in the analysis being carried out for all the annotation labels, and in additions, for each label separately. The only valuable results are given for the annotation label r-, which indicates a valid reference period markers in that region. In addition to this statistical output, the dpalign program writes two items to the SFS file. The first of these is difference in absolute time of occurrence between the test and reference Tx items. The second is the differences in period values between the test and reference items.

++

Filename ejt.frp2

Reference data

Ref item = 04

Ref hist= align(type=txdata;alignedby=3.03,from=3.02;offset=0.005375)

Test data

Test item = 03

Test hist = trtx(16.04;threshold=0.4;maxpulse;maxfx=500)

ANNOTATION LABEL [0] = all

****

Reference Tx frames = 1256

Test Tx frames = 1233

Tx hits = 981/1256 Percent = 78.1051

Tx misses = 275/1256 Percent = 21.8949

Tx total false alarms = 252/1233 Percent = 20.438

Tx voiced false alarms = 190/1233 Percent = 15.4096

Tx unvoiced false alarms = 62/1233 Percent = 5.02839

Absolute mean jitter in samples = 6.82263

Absolute sd jitter in samples = 12.9232

Relative mean jitter in samples = 0.0254842

Relative sd jitter in samples = 10.0666

ANNOTATION LABEL [1] = lost

****

No results for this annotation

ANNOTATION LABEL [2] = r-

****

Reference Tx frames = 1189

Test Tx frames = 1184

Tx hits = 951/1189 Percent = 79.9832

Tx misses = 238/1189 Percent = 20.0168

Tx total false alarms = 233/1184 Percent = 19.6791

Tx voiced false alarms = 179/1184 Percent = 15.1182

Tx unvoiced false alarms = 54/1184 Percent = 4.56081

Absolute mean jitter in samples = 6.6

Absolute sd jitter in samples = 12.9823

Relative mean jitter in samples = 0.0120219

Relative sd jitter in samples = 10.1251

ANNOTATION LABEL [3] = r+

****

Reference Tx frames = 66

Test Tx frames = 49

Tx hits = 30/66 Percent = 45.4545

Tx misses = 36/66 Percent = 20.0168

Tx total false alarms = 19/49 Percent = 38.7755

Tx voiced false alarms = 11/49 Percent = 22.449

Tx unvoiced false alarms = 8/49 Percent = 16.3265

Absolute mean jitter in samples = 654

Absolute sd jitter in samples = 2.7389e+07

Relative mean jitter in samples = 14

Relative sd jitter in samples = 1815

UTILITY PROGRAM TO MANAGE RUNNING OF PROGRAMS

One important issue that arises when many stages of processing are carried out, all of which involve a large amount of processing, is what to do in the event of a computer crash. Clearly, one wishes to re-start programs in such a way that one can continue from the point reached just before the computer crash. A program to facilitate this was written as is called pms (program management system). The program is run using the command line

pms commandfile

This program operates by reading a list of command lines placed in the file <commandfile>.run by the user. Each time a command line is executed, it is then written into the file <commandfile>.done. In the event of a computer crash (or any other circumstances which would kill all processes), the command lines can be re-started by re-typing the command line

pms filename

The pms program then reads in the <commandfile>.done file and the <commandfile>.run files, and continues by executing the command lines from just after the previously executed line.

APPENDIX A.3: TEXT FOR THE RAINBOW PASSAGE

NB: The individual paragraphs are labelled rp1-rp3.

Today's date is the .....

My name is .....

THE RAINBOW PASSAGE

(rp1)

When the sunlight strikes raindrops in the air, they act like a prism and form a rainbow. The rainbow is the division of white light into many beautiful colours.

PAUSE FOR 2 SECONDS

(rp2)

These take the shape of a long round arch with its path high above and its two ends apparently beyond the horizon. There is, according to legend, a boiling pot of gold at one end.

PAUSE FOR 2 SECONDS

(rp3)

People look, but no-one ever finds it. When a man looks for something beyond his search, his friends say that he is looking for the pot of gold at the end of the rainbow.

APPENDIX A.4: TEXT FOR ARTHUR THE RAT

NB: The individual paragraphs are labelled ar1-ar8.

THE STORY OF ARTHUR THE RAT

(ar1)

There was once a young rat named Arthur, who would never take the trouble to make up his mind. Whenever his friends asked him if he would like to go out with them, he would only answer, "I don't know." He wouldn't say "Yes" and he wouldn't say "No" either. He could never learn to make a choice.

PAUSE FOR 2 SECONDS

(ar2)

His aunt Helen said to him, "No-one will ever care for you if you carry on like this. You have no more mind than a blade of grass." Arthur looked wise, but said nothing.

PAUSE FOR 2 SECONDS

(ar3)

One rainy day the rats heard a great noise in the loft where they lived. The pine rafters were all rotten, and at last one of the joists had given way and fallen to the ground. The walls shook and the rats' hair stood on end with fear and horror.

PAUSE FOR 2 SECONDS

(ar4)

"This won't do," said the old rat who was chief. "I'll send out scouts to search for a new home." Three hours later the seven scouts came back and said, "We've found a stone house which is just what we wanted. There's room and good food for us all.

PAUSE FOR 2 SECONDS

(ar5)

There's a kindly horse named Nelly, a cow a calf and a garden with an Elm tree." Just then the old rat caught sight of Arthur. "Are you coming with us ?" he asked.

PAUSE FOR 2 SECONDS

(ar6)

"I don't know," Arthur sighed, "the roof may not come down just yet." "Well, said the old rat angrily, "we can't wait all day for you to make up your mind. Right about face! March!" And they went off.

PAUSE FOR 2 SECONDS

(ar7)

Arthur stood and watched the other rats hurry away. The idea of an immediate decision was too much for him. "I'll go back to my hole for a bit," he said to himself, "just to make up my mind."

PAUSE FOR 2 SECONDS

(ar8)

That night there was a great crash that shook the earth, and down came the whole roof. Next day some men rode up and looked at the ruins. One of them moved a board, and under it they saw a young rat lying on his side, quite dead, half in and half out of his hole.

PAUSE FOR 2 SECONDS, AND SPEAK IN CREAKY VOICE

(C+ar8)

That night there was a great crash that shook the earth, and down came the whole roof. Next day some men rode up and looked at the ruins. One of them moved a board, and under it they saw a young rat lying on his side, quite dead, half in and half out of his

hole.

APPENDIX A.5: SUBJECT QUESTIONNAIRE FOR RECORDINGS
RECORD OF DETAILS FOR SPEAKER AND ACOUSTIC ENVIRONMENT FOR MLP-Tx DATABASE

Date .....

SPEAKER DETAILS

Your Name .....
Your age .....
Male or Female .....
Your Weight .....
Your Height .....

SPEAKER ACCENT DETAILS

Please list any regions of the UK or other countries in which you have lived for more than a year and indicate roughly how long you spent there

REGION	TIME
.....	.....
.....	.....
.....	.....
.....	.....

Please give a general description of your parents' accents (if known), for example North of England, West Country, S.E.England, etc.

Father .....
Mother .....

GENERAL HEALTH

Are you under the care of a doctor at the moment for anything other than a purely physical injury (such as a broken leg)? YES/NO

Are you taking any prescribed drugs at the moment? YES/NO

Do you suffer from any respiratory disease or asthma? YES/NO

Do you suffer from hay fever? YES/NO

Are you allergic to the house dust mite? YES/NO

Are you a diabetic? YES/NO

Do you have heart disease? YES/NO

Are any of your front teeth (top or bottom) missing? YES/NO

Have you ever had a general anaesthetic? YES/NO

As far as you can tell, do you think you have normal hearing? YES/NO

LEISURE ACTIVITIES

Are you, or have you ever been a regular smoker? YES/NO

If YES: Have you been a regular smoker within the last year? YES/NO

When smoking regularly, approximately how much would you smoke in a typical week? .....

Have you drunk undiluted spirits within the last year? YES/NO

If YES: Do you drink undiluted spirits

A) Rarely (for example, at christmas) .....

B) Occasionally (a few times a year) .....

C) Regularly (once a week or more) .....

Have you drunk any diluted spirits within the last year? YES/NO
(for example, gin & tonic, rum & coke, etc).

If YES: Do you drink diluted spirits

A) Rarely (for example, at christmas) .....

B) Occasionally (a few times a year) .....

C) Regularly (once a week or more) .....

Do you take part in any activity which involves shouting or cheering (for example a football supporter)? YES/NO

If YES, please say what the activity is, give some idea of how often you take part in it and say how recently you last did so?

.....

.....

.....

Do you take part in any activity in which you "project" your voice, for example, singing (solo or in a choir or group), acting teaching or lecturing? YES/NO

If YES, please say what the activity is, give some idea of how often you take part in it and say how recently you last did so?

.....

.....

.....

Do you attend discos or pop concerts where the levels of the music is sufficient for you to have to shout to your companions? YES/NO

If YES: How long ago did you attend such an event? .....

How often would you expect to do so during a typical month? .....

Do you take part in any other activity which involves shouting or using your voice in ways other than taking part in normal conversation? YES/NO

If YES, please say what the activity is, give some idea of how often you take part in it and say how recently you last did so?

.....

.....

.....

Do you use volatile chemicals at works or as part of you leisure activity? YES/NO

If YES, please say what the activity is, give some idea of how often you take part in it and say how recently you last did so?

.....

.....

.....

FINALLY

Are you aware of anything not covered in this questionnaire which might affect your speaking voice or make it unusual in any way? YES/NO

If YES, please specify.

.....

.....

.....

THANK YOU FOR YOUR COOPERATION

INFORMATION TO BE FILLED IN BY RECORDER OPERATOR

Building name and location.

.....

.....

.....

Room use for recordings. .....

Room height. .....

Room width. .....

Room length. .....

Does the room have carpets? YES/NO

Does the room have curtains? YES/NO

Is the room particularly reverberant or damped?
VERY-REVERBERANT/VERY-DAMPED

Height of microphone from ground. .....

Distance of microphone from two nearest walls. ....

Is speaker sitting or standing? SITTING/STANDING

Distance of microphone to speaker. .....

Distance of speaker from two nearest walls. ....

Background noise level	VERY LOW
	LOW
	MEDIUM
	HIGH
	VERY HIGH
	DBA reading .....
Background noise description	.....
Microphone used	.....
Calibration level	.....

APPENDIX A.6: LIST OF ROOMS USED FOR RECORDINGS

AR - Anechoic chamber, Gordon Square, Phonetics, UCL.
Very low-noise and very low reverberation.

HD - DR, Medium sized dinning room, in a semi-detached house. Furnished with a carpet and curtains.

RRB - Recording room B, Wolfson House, Phonetics, UCL.
Medium-sized low-noise recording room.

RRC - Recording room C, Wolfson House, Phonetics, UCL.
Small low-noise recording room.

SSL - Speech science laboratory, Wolfson House, Phonetics, UCL.
Large room with low-level car traffic background noise.

SSCR - Speech science common room, Wolfson House, Phonetics, UCL.
Large room with little background noise.

CIR - Cochlear implants room, Guy's Hospital, London.
Long narrow room with air-conditioning background noise.

PD - Room 409, Psychology, UCL, Bedford Way, London.
Small room with loud car traffic background noise.

GR - Green Room, Monarch House, Ace Editing, Acton, London.
Large room with background office noise.

APPENDIX A.7: LIST OF FILENAMES AND THEIR CORRESPONDING SPEAKERS

This list provides details of the speech and laryngograph data acquired into files onto the computer system. For each test datafile, the speech material used, the speaker's name, the speaker's age, the recording distance and recording location are provided. (Note: not all information was avaliable for all subjects).

The filename was constructed to represent the identity of the speaker (the characters before the first point), to explicitly identify male and female speakers (either a "f" or an "m" after the point), and to explicitly identify the speech passage (ar1-ar8 and rp1-rp3. The latter correspond to the paragraph labels given in appendices 3 & 4 for Arthur the Rat and the Rainbow passage respectively.

TRAINING DATA: ALL RECORDED USING FIXED DISTANCE HEAD-MOUNTED MICROPHONE

FILEMANE	DATA	NAME	SEX	RECORDING ROOM
ea2.far1	ar1	Evelyn Abberton	F	UCL, RRB
ea2.far2	ar2	Evelyn Abberton	F	UCL, RRB
ea2.far3	ar3	Evelyn Abberton	F	UCL, RRB
ea2.far4	ar4	Evelyn Abberton	F	UCL, RRB
ea2.far5	ar5	Evelyn Abberton	F	UCL, RRB
ea2.far6	ar6	Evelyn Abberton	F	UCL, RRB
ea2.far7	ar7	Evelyn Abberton	F	UCL, RRB
ea2.far8	ar8	Evelyn Abberton	F	UCL, RRB
ea2.frp1	rp1	Evelyn Abberton	F	UCL, RRB
ea2.frp2	rp2	Evelyn Abberton	F	UCL, RRB
ea2.frp3	rp3	Evelyn Abberton	F	UCL, RRB
ea3.frp1	rp1	Evelyn Abberton	F	UCL, AR
ea3.frp2	rp2	Evelyn Abberton	F	UCL, AR
ea3.frp3	rp3	Evelyn Abberton	F	UCL, AR
vb2.far1	ar1	Ginny Ball	F	UCL, RRB
vb2.far2	ar2	Ginny Ball	F	UCL, RRB
vb2.far3	ar3	Ginny Ball	F	UCL, RRB
vb2.far4	ar4	Ginny Ball	F	UCL, RRB
vb2.far5	ar5	Ginny Ball	F	UCL, RRB

vb2.far6	ar6	Ginny Ball	F	UCL, RRB
vb2.far7	ar7	Ginny Ball	F	UCL, RRB
vb2.far8	ar8	Ginny Ball	F	UCL, RRB
vb2.frp1	rp1	Ginny Ball	F	UCL, RRB
vb2.frp2	rp2	Ginny Ball	F	UCL, RRB
vb2.frp3	rp3	Ginny Ball	F	UCL, RRB
vb3.frp1	rp1	Ginny Ball	F	UCL, AR
vb3.frp2	rp2	Ginny Ball	F	UCL, AR
vb3.frp3	rp3	Ginny Ball	F	UCL, AR
tmd.far1	ar1	Maria Dahl	F	UCL, SSCR
tmd.far2	ar2	Maria Dahl	F	UCL, SSCR
tmd.far3	ar3	Maria Dahl	F	UCL, SSCR
tmd.far4	ar4	Maria Dahl	F	UCL, SSCR
tmd.far5	ar5	Maria Dahl	F	UCL, SSCR
tmd.far6	ar6	Maria Dahl	F	UCL, SSCR
tmd.far7	ar7	Maria Dahl	F	UCL, SSCR
tmd.far8	ar8	Maria Dahl	F	UCL, SSCR
tmd.frp1	rp1	Maria Dahl	F	UCL, SSCR
tmd.frp2	rp2	Maria Dahl	F	UCL, SSCR
tmd.frp3	rp3	Maria Dahl	F	UCL, SSCR
tsh.far1	ar1	Sylvia Howard	F	HD, DR
tsh.far2	ar2	Sylvia Howard	F	HD, DR
tsh.far3	ar3	Sylvia Howard	F	HD, DR
tsh.far4	ar4	Sylvia Howard	F	HD, DR
tsh.far5	ar5	Sylvia Howard	F	HD, DR
tsh.far6	ar6	Sylvia Howard	F	HD, DR
tsh.far7	ar7	Sylvia Howard	F	HD, DR
tsh.far8	ar8	Sylvia Howard	F	HD, DR
tsh.frp1	rp1	Sylvia Howard	F	HD, DR
tsh.frp2	rp2	Sylvia Howard	F	HD, DR
tsh.frp3	rp3	Sylvia Howard	F	HD, DR
tih.mar1	ar1	Ian Howard	M	HD, DR
tih.mar2	ar2	Ian Howard	M	HD, DR
tih.mar3	ar3	Ian Howard	M	HD, DR
tih.mar4	ar4	Ian Howard	M	HD, DR
tih.mar5	ar5	Ian Howard	M	HD, DR
tih.mar6	ar6	Ian Howard	M	HD, DR

tih.mar7	ar7	Ian Howard	M	HD, DR
tih.mar8	ar8	Ian Howard	M	HD, DR
tih.mrp1	rp1	Ian Howard	M	HD, DR
tih.mrp2	rp2	Ian Howard	M	HD, DR
tih.mrp3	rp3	Ian Howard	M	HD, DR
rb2.mar1	ar1	Richard Baker	M	UCL, RRC
rb2.mar2	ar2	Richard Baker	M	UCL, RRC
rb2.mar3	ar3	Richard Baker	M	UCL, RRC
rb2.mar4	ar4	Richard Baker	M	UCL, RRC
rb2.mar5	ar5	Richard Baker	M	UCL, RRC
rb2.mar6	ar6	Richard Baker	M	UCL, RRC
rb2.mar7	ar7	Richard Baker	M	UCL, RRC
rb2.mar8	ar8	Richard Baker	M	UCL, RRC
rb2.mrp1	rp1	Richard Baker	M	UCL, RRC
rb2.mrp2	rp2	Richard Baker	M	UCL, RRC
rb2.mrp3	rp3	Richard Baker	M	UCL, RRC
jd2.mar1	ar1	Julian Daley	M	UCL, RRC
jd2.mar2	ar2	Julian Daley	M	UCL, RRC
jd2.mar3	ar3	Julian Daley	M	UCL, RRC
jd2.mar4	ar4	Julian Daley	M	UCL, RRC
jd2.mar5	ar5	Julian Daley	M	UCL, RRC
jd2.mar6	ar6	Julian Daley	M	UCL, RRC
jd2.mar7	ar7	Julian Daley	M	UCL, RRC
jd2.mar8	ar8	Julian Daley	M	UCL, RRC
jd2.mrp1	rp1	Julian Daley	M	UCL, RRC
jd2.mrp2	rp2	Julian Daley	M	UCL, RRC
jd2.mrp3	rp3	Julian Daley	M	UCL, RRC

EVALUATION TEST DATA

FILENAME	DATA	NAME	SEX	AGE	DIST	ROOM
eco.mrp1	rp1	Con Onisiphone	M	35	0.6m	GH, CIR
ehd.frp1	rp1	Hilary Dodson	F		0.8m	GH, CIR
ero.mrp2	rp2	Roger	M		0.8m	GH, CIR
ego.mrp3	rp3	Godfrey	M		0.6m	GH, CIR
emh.mar1	ar1	Marcus Hampshire	M	27	1.0m	MH, GR
ear.mar2	ar2	Andrew Rendell	M	27	1.3m	MH, GR
ekj.mar3	ar3	Ken Joyce	M	29	3.0m	MH, GR

ejt.frp2	rp2	Joanne Thomson	F	20	2.5m	MH, GR
epf.mar4	ar4	Perry Foran	M	30	2.5m	MH, GR
enn.mar5	ar5	Nitan Negandhi	M	26	3.0m	MH, GR
ean.mar6	ar6	Alfred Ng	M	29	2.0m	UCL, SSL
eaf.mar7	ar7	Andrew Faulkner	M	38	1.5m	UCL, SSL
ebs.frp3	rp3	Beatrice Sayers	F	23	0.7m	UCL, SSL
eag.far1	ar1	Ann Guyon	F	37	1.3m	UCL, RRB
erc.far3	ar3	Rosie Casson	F	20	0.8m	UCL, RRB
ejd.far4	ar4	Judy Davies	F	32	0.4m	UCL, RRB
eCfa.far7	ar7	Freda Ali	F	22	0.9m	UCL, RRA
efa.far5	ar5	Freda Ali	F	22	0.9m	UCL, RRA
esr.far6	ar6	Suzanne Riley	F	21	1.2m	UCL, RRA
ejr.far6	ar7	Janet Rowland	F	23	1.3m	UCL, RRA

FINAL TESTING DATA

FILENAME	DATA	NAME	SEX	AGE	DIST	ROOM
fgt.mrp1	rp1	Graham Taylor	M	43	1.2m	GH, CIR
fmg.frp1	rp1	Marilyn George	F	26	0.6m	GH, CIR
fsa.frp2	rp2	Sarah Allen	F	27	0.9m	GH, CIR
fcb.mrp3	rp3	Chris Banaham	M	30	1.0m	MH, GR
fxp.frp3	rp3	Xanthe Parkin	F	28		MH, GR
fjv.mar4	ar4	John Vigar	M	50	1.5m	MH, GR
fsa.mar6	ar6	Simon Ashcroft	M	27	3.0m	MH, GR
fjr.far1	ar1	Judy Reddaway	F	21	1.5m	MH, GR
fre.mar3	ar3	Richard Elerson	M	23	2.0m	MH, GR
fjd.mar1	ar1	James Duncan	M	28	1.0m	MH, GR
fjh.mar8	ar8	James Howard	M	21	1.0	HD, DR
fdqh.mar7	ar7	David Howard	M	18	2.0m	HD, DR
fpe.far3	ar3	Patricia Evans	F	55	1.5m	HD, DR
frb.french	fp	Remi Brun	M	25	1.1m	UCL, SSL
fdmh.mrp2	rp2	David Howard	M	33		UCL, SSL
fmj.mrp3	rp3	Mike Johnston	M	34		UCL, SSL
fsn.mrp1	rp1	Steven Nevard	M	40		UCL, SSL
fss.mar1	ar1	Steve Sakin	M	23		UCL, SSL
fgl.mar2	ar2	Geoff Linsey	M	30	0.5m	UCL, SSL
fjh.far4	ar4	Jill House	F	45	1.1m	UCL, SSL
fje.frp2	rp2	Jane 'Espinasse	F	45	0.7m	UCL, SSL

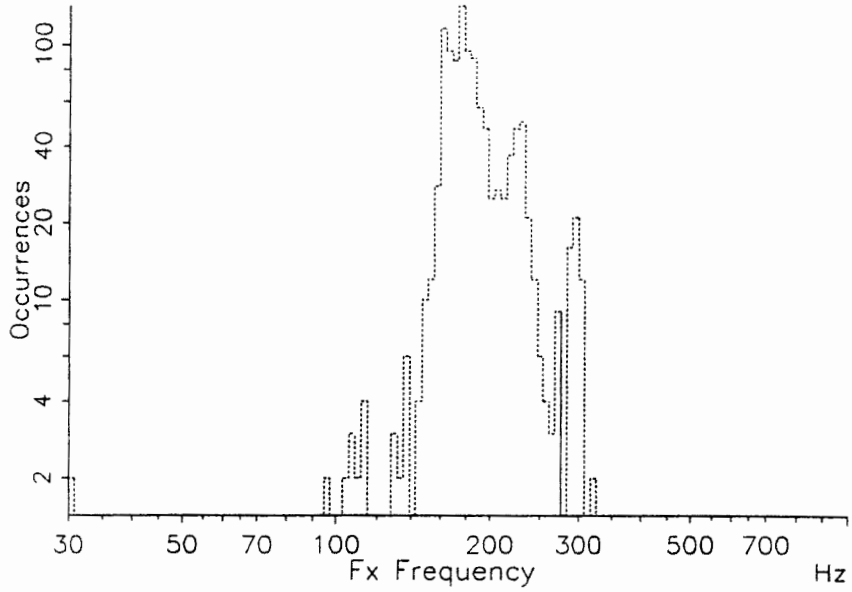
fba.frp3	rp3	Bridget Allen	F	24	2.0m	UCL, SSL
far.far1	ar1	Alison Rumbold	F	22	1.4m	UCL, RRB
fjor.far3	ar3	Joe Robson	F	19	1.3m	UCL, RRB
fbh.far2	ar2	Barbera House	F	40	1.2m	UCL, RRB
fmag.far4	ar4	Martha Gibson	F	44	0.6m	UCL, RRB
fmag.far6	ar6	Martime Grice	F	30	0.7m	UCL, SSL
feg.far8	ar8	Emma Gash	F	20	0.9m	UCL, RRB
fjc.far5	ar5	Jean Chambers	F	40	0.4m	UCL, RRB
fcis.far6	ar6	Cindy Strictland	F	34	0.5m	UCL, RRB
faCjf.mar8	ar8	Adrian Fourcin	M	62		UCL, RRB
fajf.mrp1	rp1	Adrian Fourcin	M	62		UCL, RRB
fdb.frp3	rp3	Donna Blakemore	F	21	0.9m	UCL, RRA
fed.far1	ar1	Emma Daniels	F	21	0.9m	UCL, RRA
fdv.frp2	rp2	Deborah Vickers	F	21	1.3m	UCL, RRA
fab.far2	ar2	Ashleigh Bullen	F	22	0.7m	UCL, RRA
fvh.far7	ar7	Valarie Hazan	F	32	0.4m	UCL, RRB
fjs.mrp3	rp3	John Scoyles	M	31	1.5m	UCL, PD
fky.mar2	ar2	Keith Young	M	24	0.5m	UCL, PD
ftt.mrp2	rp2	Thiery Towllelan	M	24	1.5m	UCL, PD
fshs.mar1	ar1	Shah Shahidi	M	25	0.5m	UCL, PD
fme.far4	ar4	Maeve Ennis	F	48	1.0m	UCL, PD

APPENDIX A.8: FREQUENCY DISTRIBUTION HISTOGRAMS FOR TRAINING AND FINAL TESTING DATA

fdv.frp2 Speaker: dv Duration: 13.4 s

Mean: 184.4 Hz Sdev: 26 % Skewness: −2.5, Kurtosis: 17.9

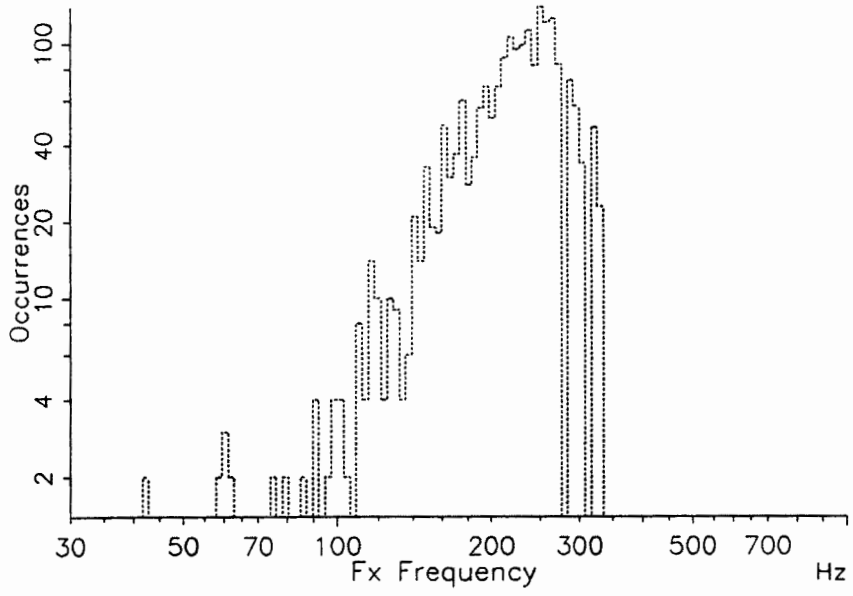
1141/1167 values Bin width: 2.8% Order: 1 Deviation: 2.8%



fjw.china Speaker: jw Duration: 20.0 s

Mean: 216.5 Hz Sdev: 30 % Skewness: −1.5, Kurtosis: 4.3

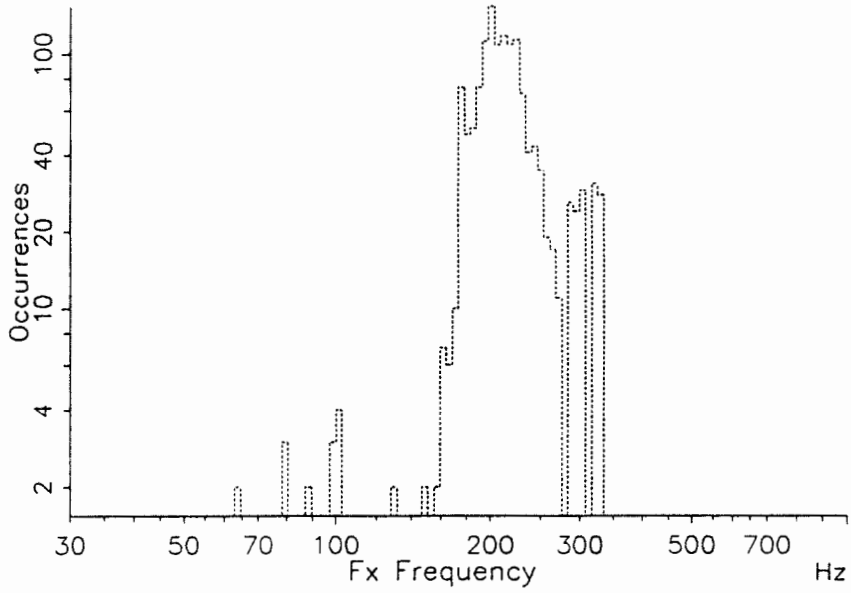
1980/2028 values Bin width: 2.8% Order: 1 Deviation: 2.8%



fjor.far3 Speaker: jor Duration: 15.0 s

Mean: 212.8 Hz Sdev: 24 % Skewness: −1.5, Kurtosis: 15.1

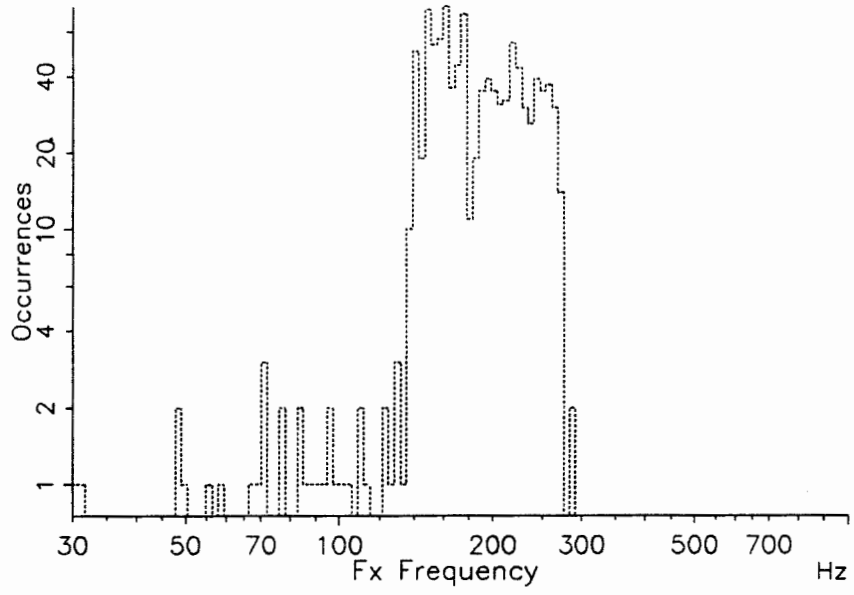
1389/1429 values Bin width: 2.8% Order: 1 Deviation: 2.8%



fcis.far6 Speaker: cis Duration: 13.4 s

Mean: 182.8 Hz Sdev: 29 % Skewness: −1.6, Kurtosis: 7.7

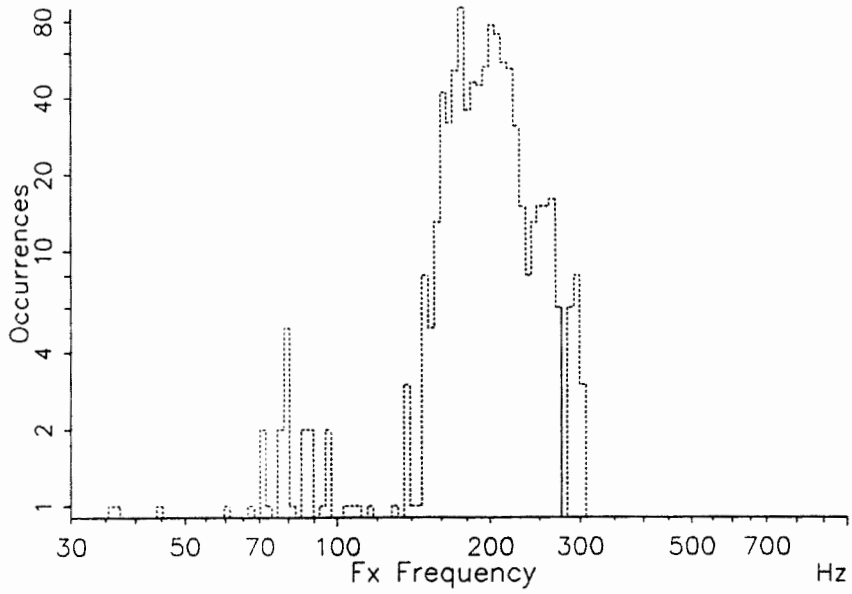
1033/1061 values Bin width: 2.8% Order: 1 Deviation: 2.8%



fxp.frp3 Speaker: xp Duration: 10.8 s

Mean: 190.4 Hz Sdev: 25 % Skewness: −2.6, Kurtosis: 13.5

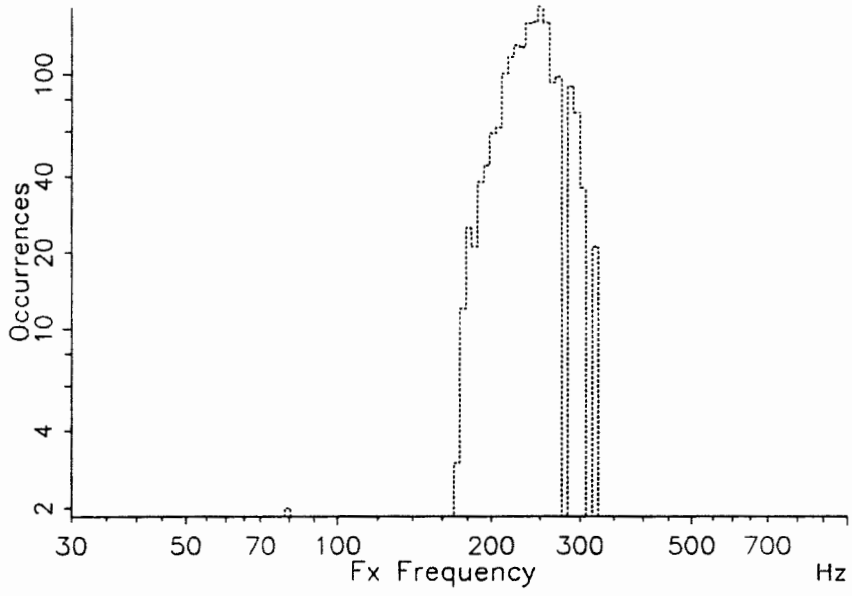
845/871 values Bin width: 2.8% Order: 1 Deviation: 2.8%



far.far1 Speaker: ar Duration: 19.6 s

Mean: 235.9 Hz Sdev: 20 % Skewness: −3.9, Kurtosis: 33.3

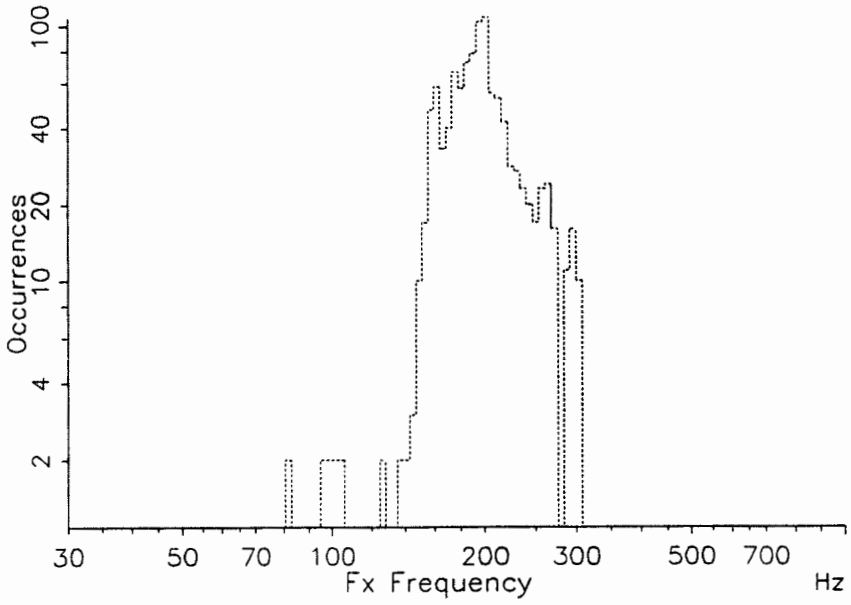
1833/1879 values Bin width: 2.8% Order: 1 Deviation: 2.8%



fmg.frp1 Speaker: mg Duration: 12.0 s

Mean: 194.1 Hz Sdev: 23 % Skewness: −1.9, Kurtosis: 13.5

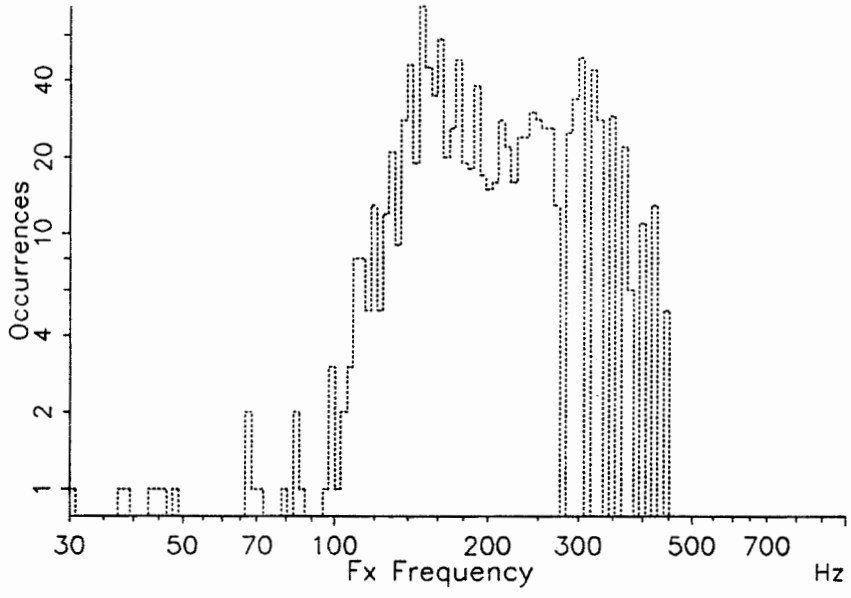
1084/1112 values Bin width: 2.8% Order: 1 Deviation: 2.8%



fjh.far4 Speaker: jh Duration: 17.4 s

Mean: 200.6 Hz Sdev: 45 % Skewness: −0.3, Kurtosis: 1.1

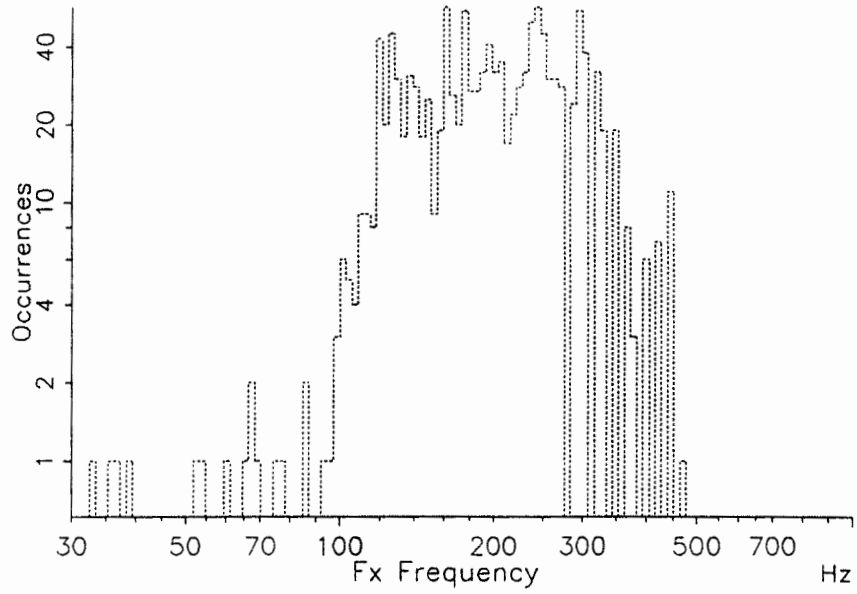
1135/1171 values Bin width: 2.8% Order: 1 Deviation: 2.8%



fmag.far4 Speaker: mag Duration: 16.2 s

Mean: 196.5 Hz Sdev: 43 % Skewness: −0.4, Kurtosis: 1.0

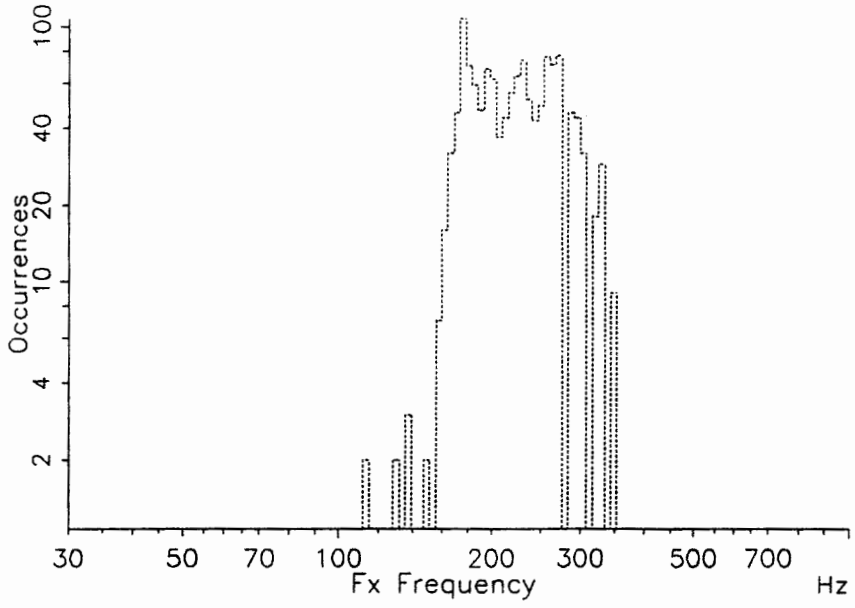
1261/1303 values Bin width: 2.8% Order: 1 Deviation: 2.8%



fmag.far6 Speaker: mag Duration: 19.4 s

Mean: 220.8 Hz Sdev: 25 % Skewness: −0.9, Kurtosis: 7.7

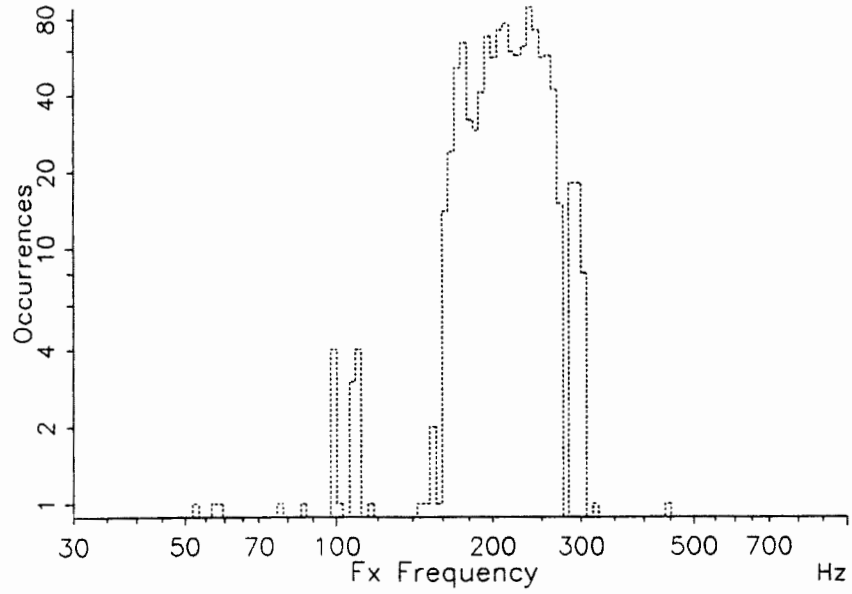
1353/1381 values Bin width: 2.8% Order: 1 Deviation: 2.8%



fbh.far2 Speaker: bh Duration: 12.8 s

Mean: 212.5 Hz Sdev: 21 % Skewness: −1.6, Kurtosis: 8.6

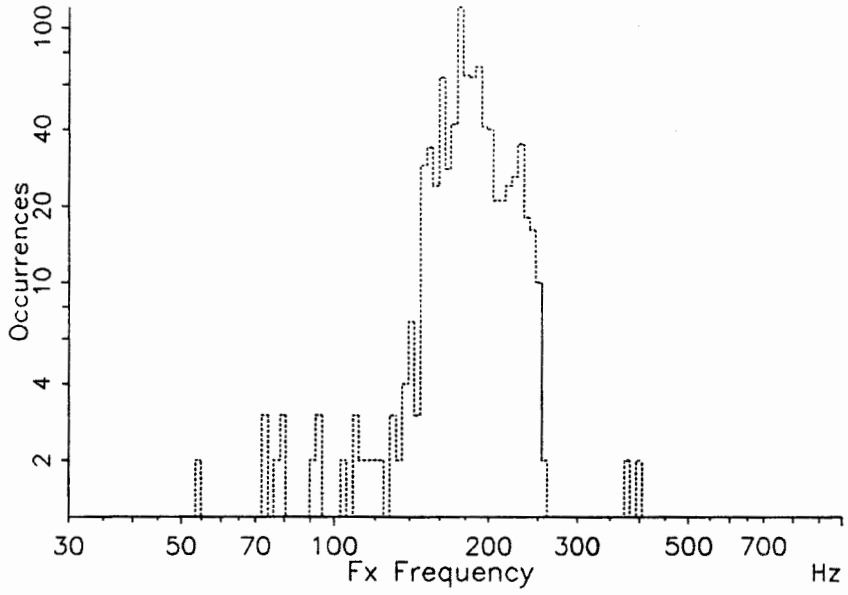
1104/1129 values Bin width: 2.8% Order: 1 Deviation: 2.8%



fvh.far7 Speaker: vh Duration: 11.2 s

Mean: 179.1 Hz Sdev: 25 % Skewness: −2.1, Kurtosis: 12.6

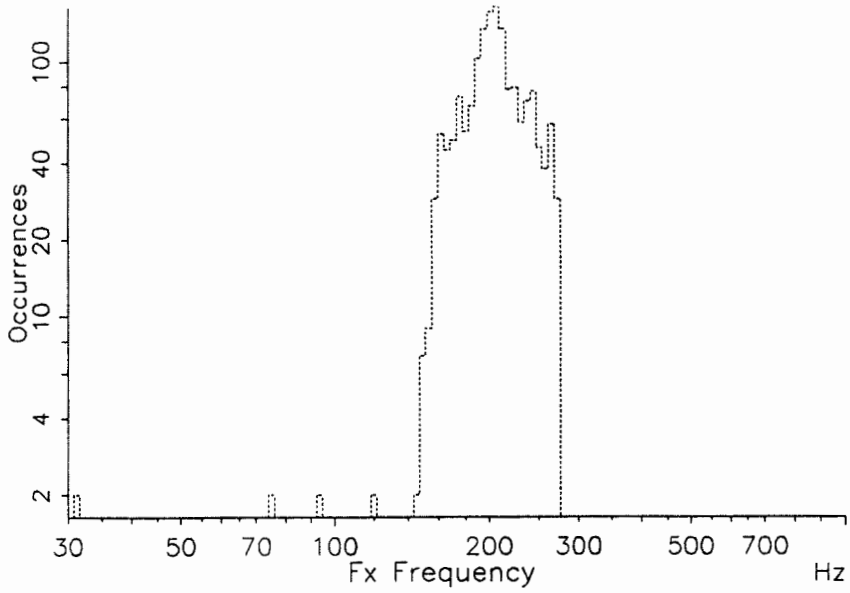
856/886 values Bin width: 2.8% Order: 1 Deviation: 2.8%



fpe.far3 Speaker: pe Duration: 16.6 s

Mean: 202.1 Hz Sdev: 21 % Skewness: −3.5, Kurtosis: 28.4

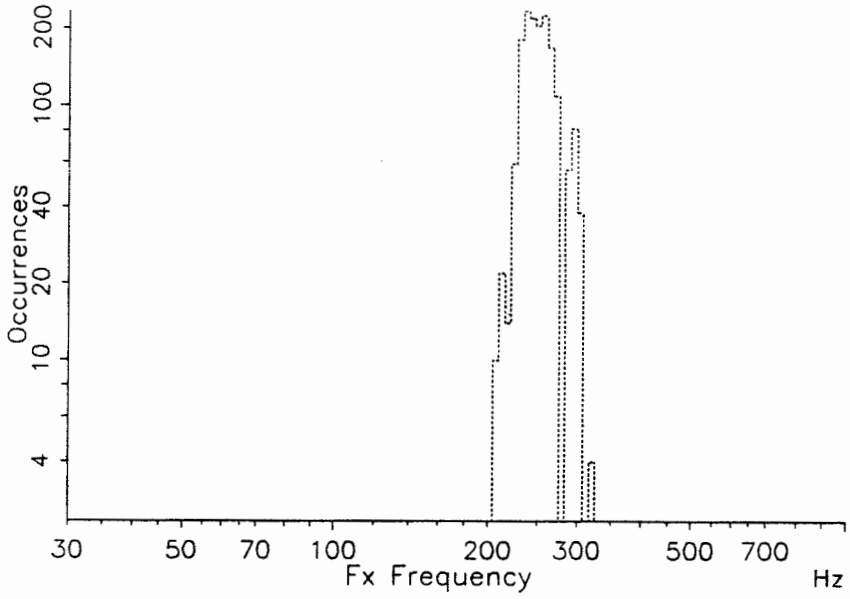
1630/1663 values Bin width: 2.8% Order: 1 Deviation: 2.8%



feg.far8 Speaker: eg Duration: 16.6 s

Mean: 249.3 Hz Sdev: 14 % Skewness: −5.3, Kurtosis: 67.0

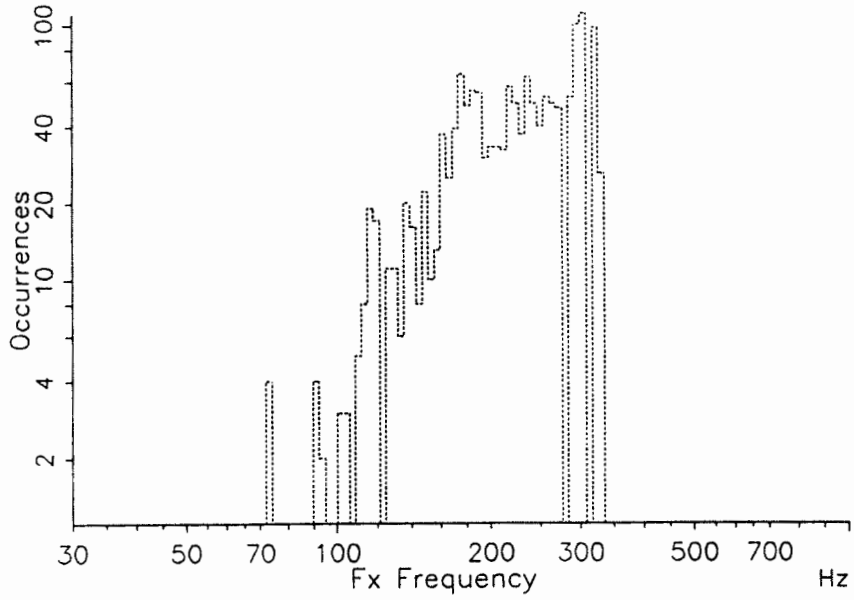
1646/1685 values Bin width: 2.8% Order: 1 Deviation: 2.8%



fme.far4 Speaker: me Duration: 16.2 s

Mean: 217.6 Hz Sdev: 35 % Skewness: −0.9, Kurtosis: 1.3

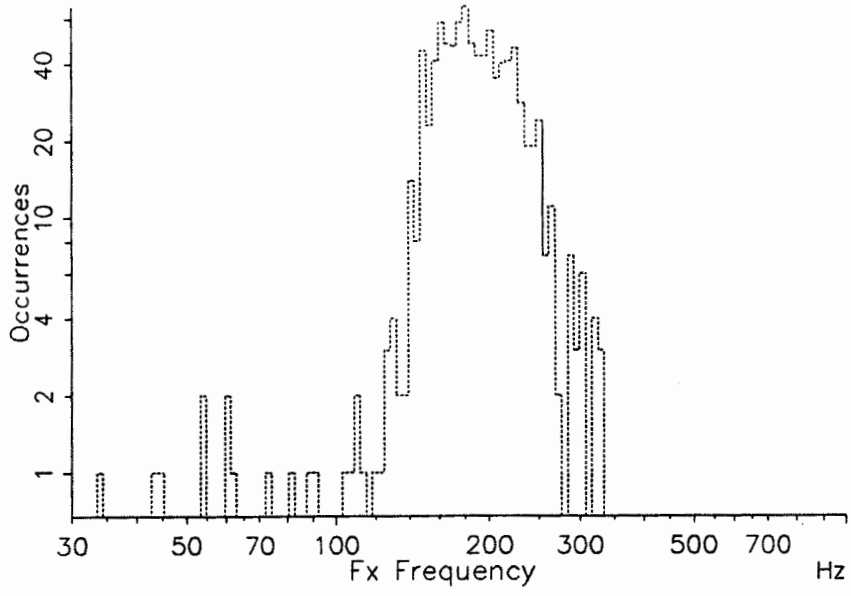
1469/1506 values Bin width: 2.8% Order: 1 Deviation: 2.8%



fje.frp2 Speaker: je Duration: 10.0 s

Mean: 186.1 Hz Sdev: 25 % Skewness: −1.8, Kurtosis: 11.1

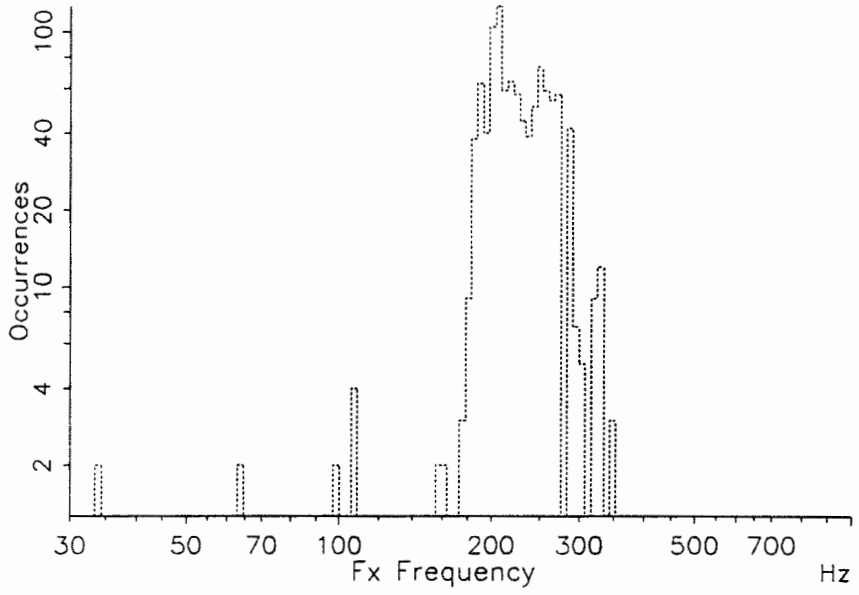
922/952 values Bin width: 2.8% Order: 1 Deviation: 2.8%



fdb.frp3 Speaker: db Duration: 12.0 s

Mean: 220.8 Hz Sdev: 26 % Skewness: −3.6, Kurtosis: 23.0

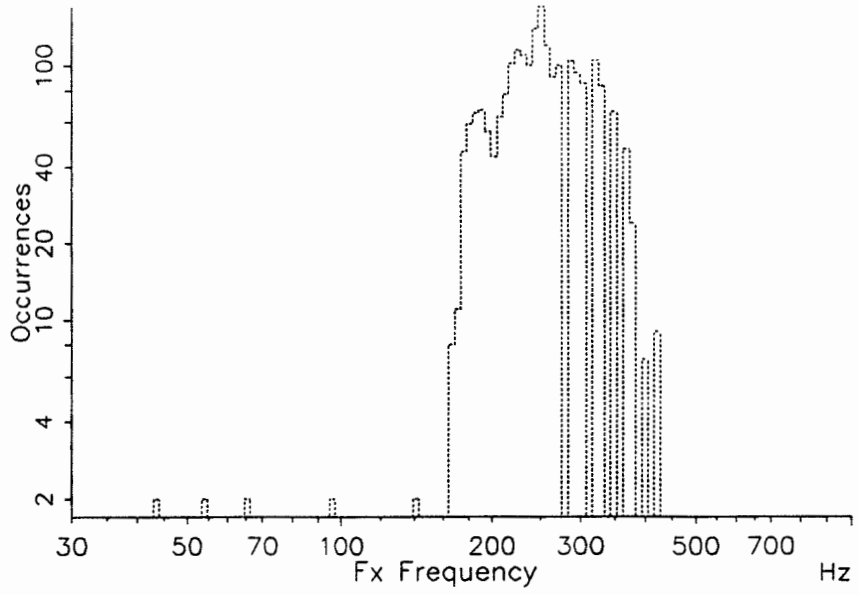
1049/1074 values Bin width: 2.8% Order: 1 Deviation: 2.8%



fed.far1 Speaker: ed Duration: 19.8 s

Mean: 246.1 Hz Sdev: 28 % Skewness: −1.9, Kurtosis: 12.6

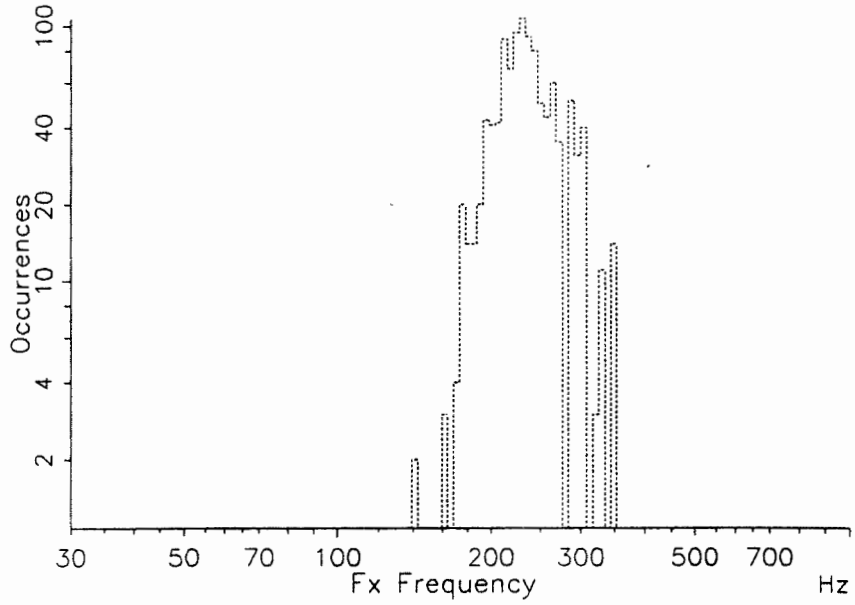
2200/2241 values Bin width: 2.8% Order: 1 Deviation: 2.8%



fba.frp3 Speaker: ba Duration: 10.4 s

Mean: 229.7 Hz Sdev: 23 % Skewness: −2.7, Kurtosis: 16.3

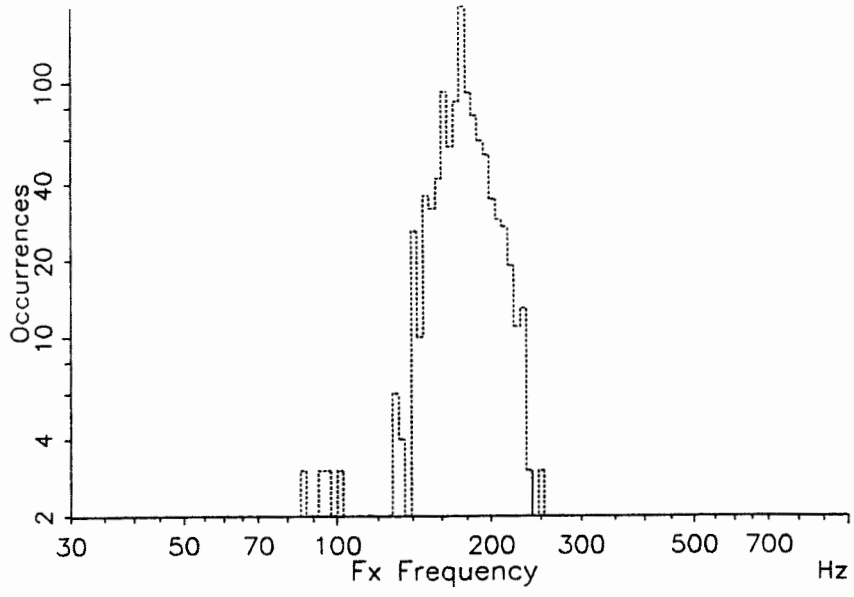
1093/1117 values Bin width: 2.8% Order: 1 Deviation: 2.8%



fsa.frp2 Speaker: sa Duration: 11.8 s

Mean: 171.4 Hz Sdev: 22 % Skewness: −3.4, Kurtosis: 19.5

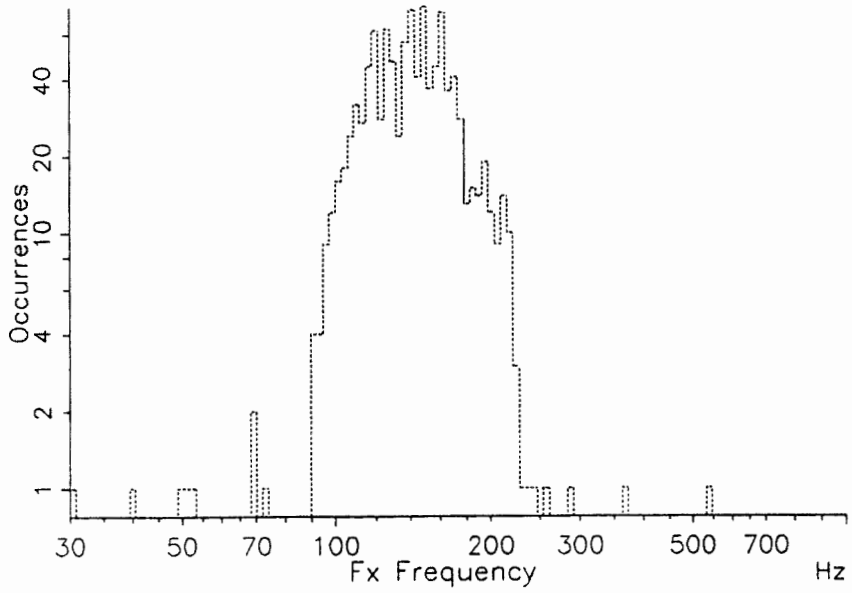
1048/1076 values Bin width: 2.8% Order: 1 Deviation: 2.8%



fjv.mar4 Speaker: jv Duration: 15.4 s

Mean: 141.0 Hz Sdev: 24 % Skewness: −0.4, Kurtosis: 5.5

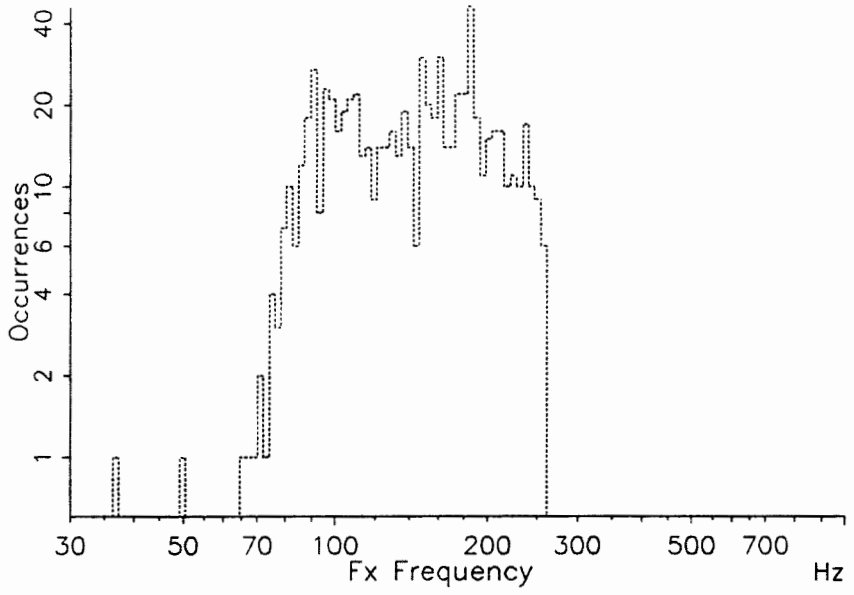
1048/1085 values Bin width: 2.8% Order: 1 Deviation: 2.8%



fky.mar2 Speaker: ky Duration: 10.4 s

Mean: 140.5 Hz Sdev: 39 % Skewness: −0.2, Kurtosis: −0.7

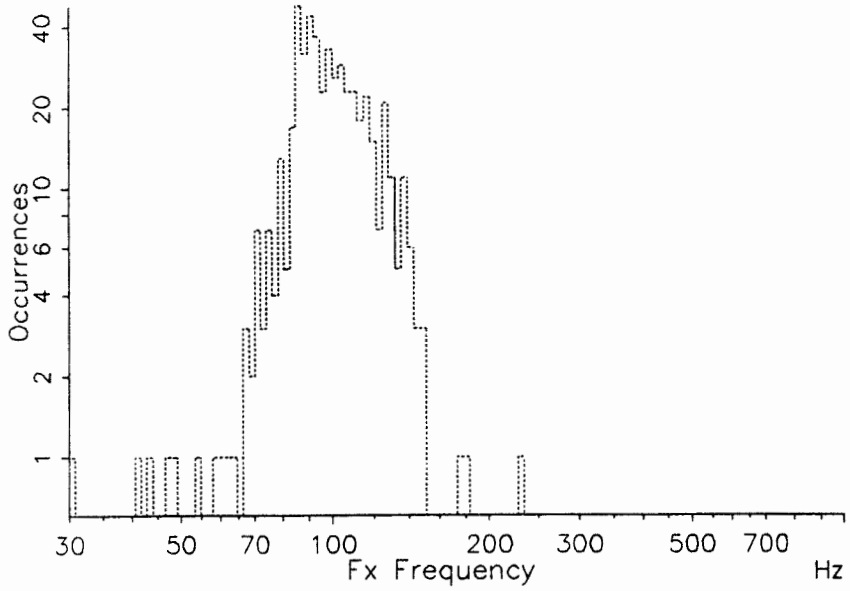
722/745 values Bin width: 2.8% Order: 1 Deviation: 2.8%



fgt.mrp1 Speaker: gt Duration: 9.8 s

Mean: 98.7 Hz Sdev: 22 % Skewness: −0.6, Kurtosis: 4.3

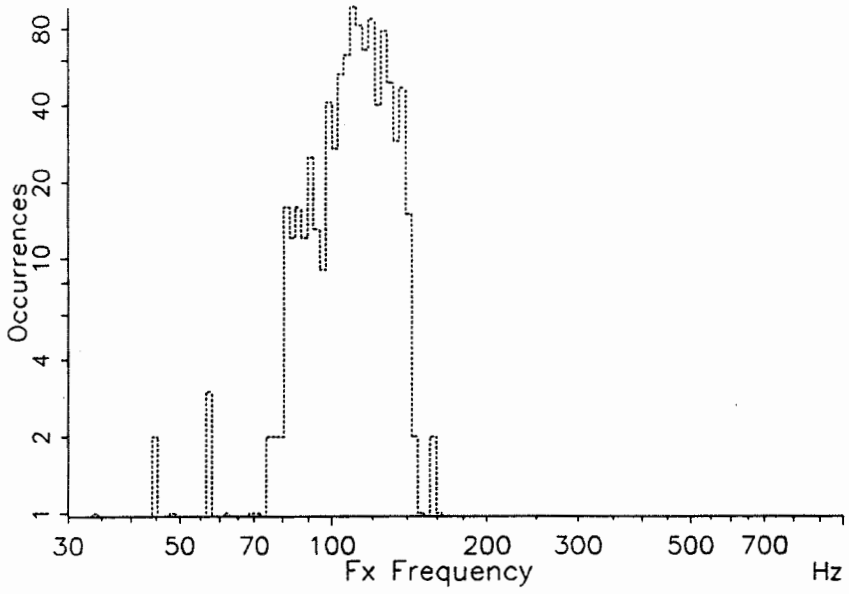
514/534 values Bin width: 2.8% Order: 1 Deviation: 2.8%



ftt.mrp2 Speaker: tt Duration: 14.8 s

Mean: 112.1 Hz Sdev: 17 % Skewness: −1.8, Kurtosis: 7.3

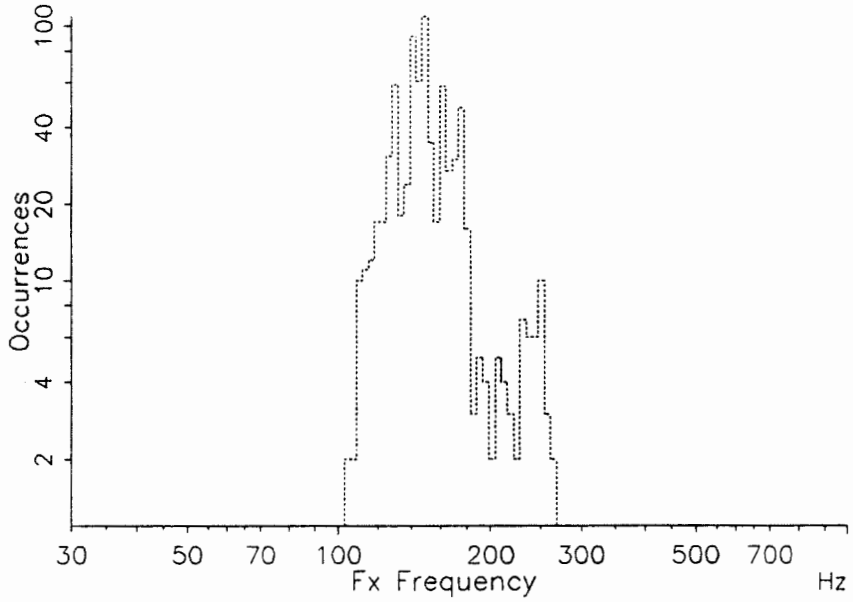
899/925 values Bin width: 2.8% Order: 1 Deviation: 2.8%



fjs.mrp3 Speaker: js Duration: 13.0 s

Mean: 150.1 Hz Sdev: 20 % Skewness: 0.0, Kurtosis: 4.9

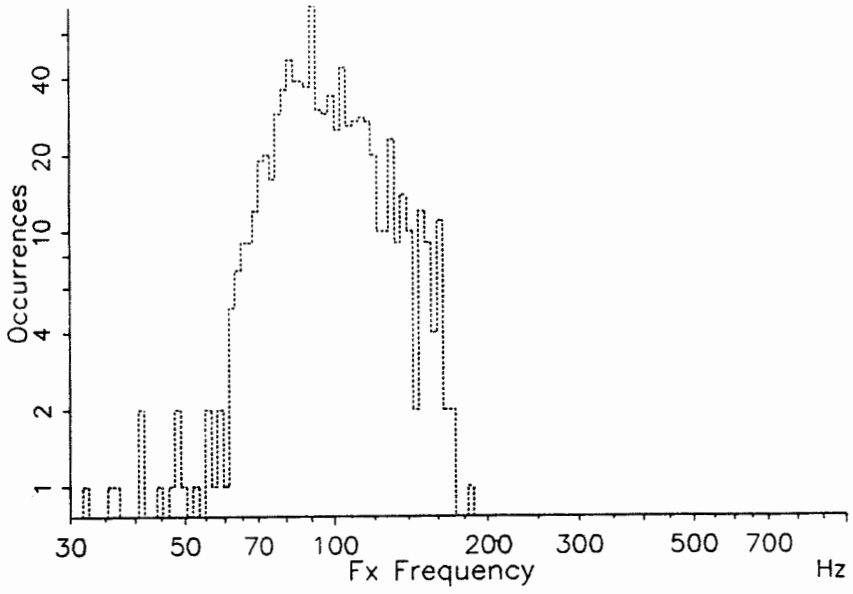
766/795 values Bin width: 2.8% Order: 1 Deviation: 2.8%



fss.mar1 Speaker: ss Duration: 16.4 s

Mean: 95.0 Hz Sdev: 27 % Skewness: −0.1, Kurtosis: 0.9

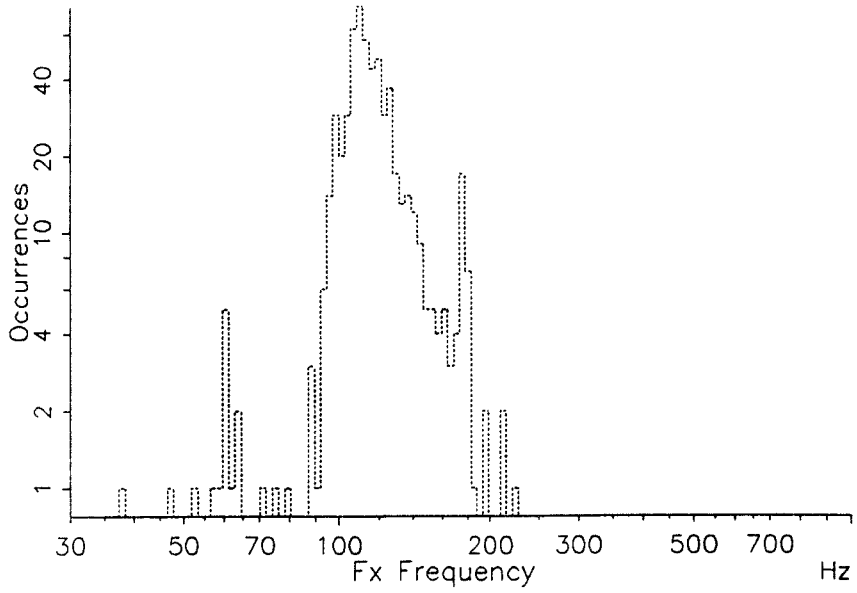
826/858 values Bin width: 2.8% Order: 1 Deviation: 2.8%



fsn.mrp1 Speaker: sn Duration: 11.0 s

Mean: 116.6 Hz Sdev: 21 % Skewness: −0.6, Kurtosis: 4.8

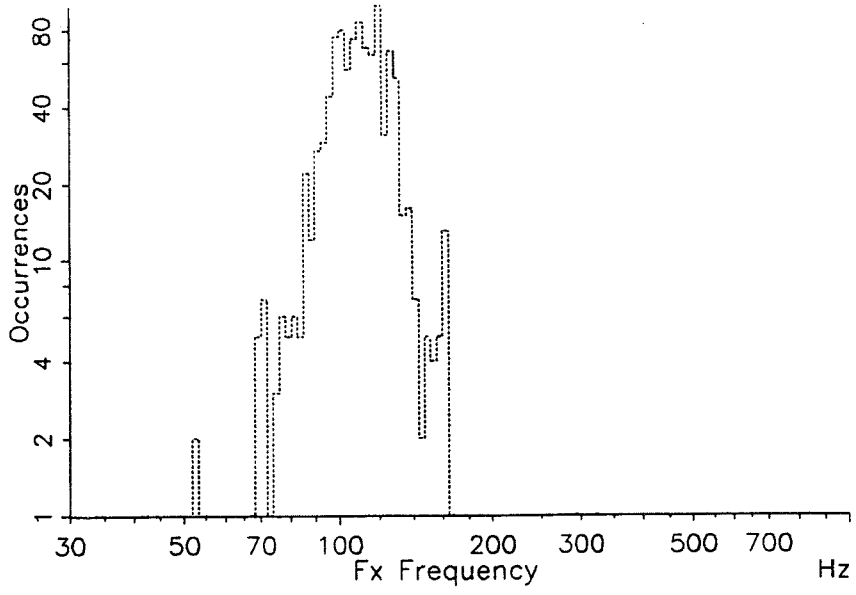
594/615 values Bin width: 2.8% Order: 1 Deviation: 2.8%



fshs.mar1 Speaker: shs Duration: 19.6 s

Mean: 109.0 Hz Sdev: 20 % Skewness: −1.2, Kurtosis: 10.4

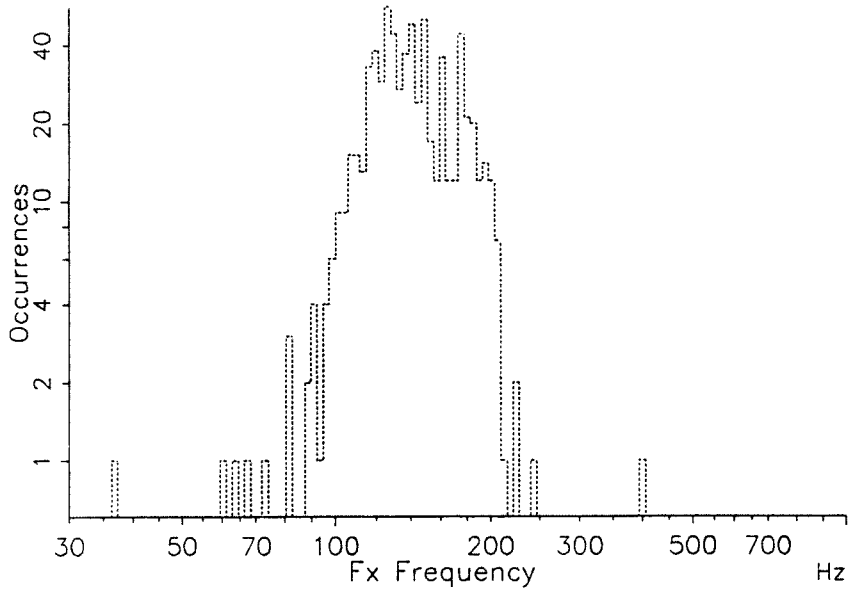
1004/1038 values Bin width: 2.8% Order: 1 Deviation: 2.8%



fmj.mrp3 Speaker: mj Duration: 12.6 s

Mean: 140.3 Hz Sdev: 23 % Skewness: −0.4, Kurtosis: 3.3

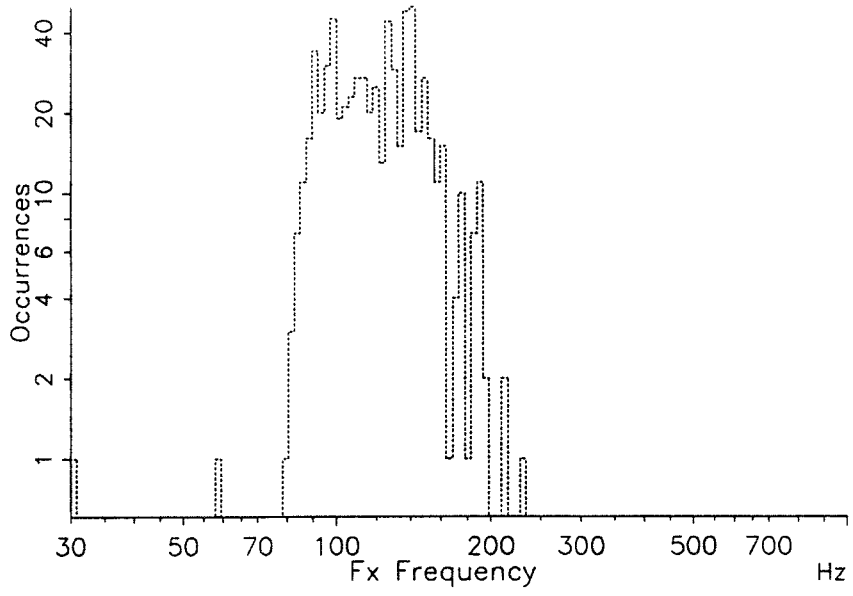
696/716 values Bin width: 2.8% Order: 1 Deviation: 2.8%



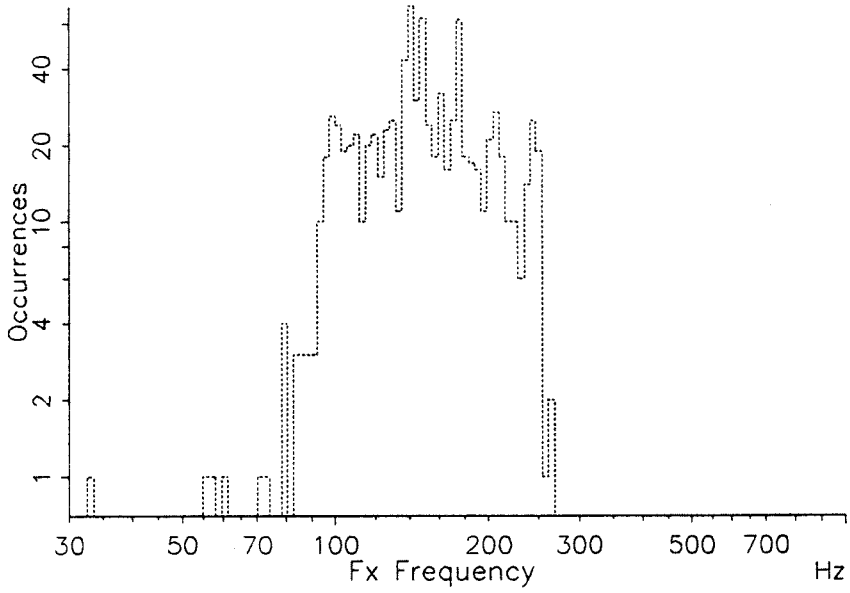
fgl.mar2 Speaker: gl Duration: 12.0 s

Mean: 120.1 Hz Sdev: 24 % Skewness: −0.2, Kurtosis: 1.6

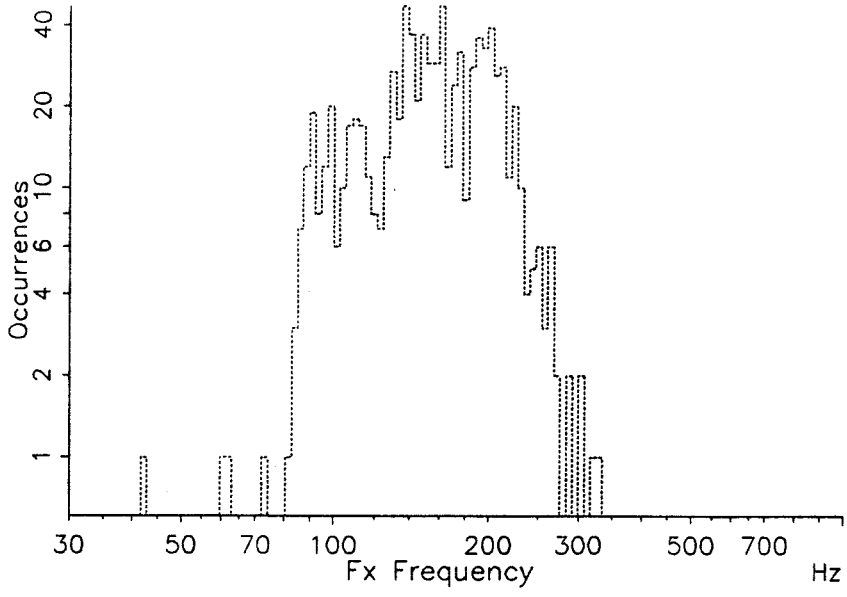
655/677 values Bin width: 2.8% Order: 1 Deviation: 2.8%



fdmh.mrp2 Speaker: dmh Duration: 13.6 s
Mean: 148.5 Hz Sdev: 32 % Skewness: −0.3, Kurtosis: 0.5
885/914 values Bin width: 2.8% Order: 1 Deviation: 2.8%



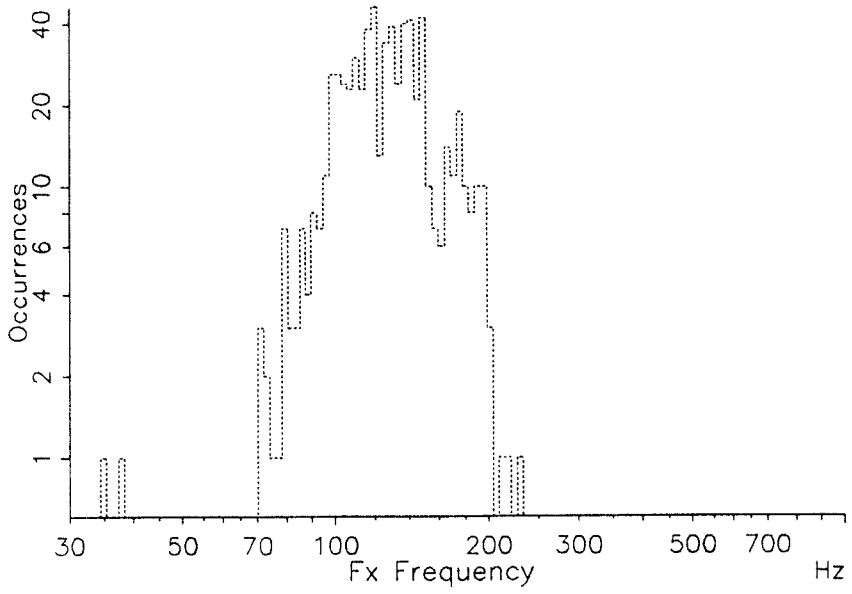
fdqh.mar7 Speaker: dqh Duration: 13.2 s
Mean: 153.0 Hz Sdev: 33 % Skewness: −0.4, Kurtosis: 0.0
825/851 values Bin width: 2.8% Order: 1 Deviation: 2.8%



fajf.mrp1 Speaker: ajf Duration: 11.4 s

Mean: 126.4 Hz Sdev: 25 % Skewness: −0.5, Kurtosis: 2.4

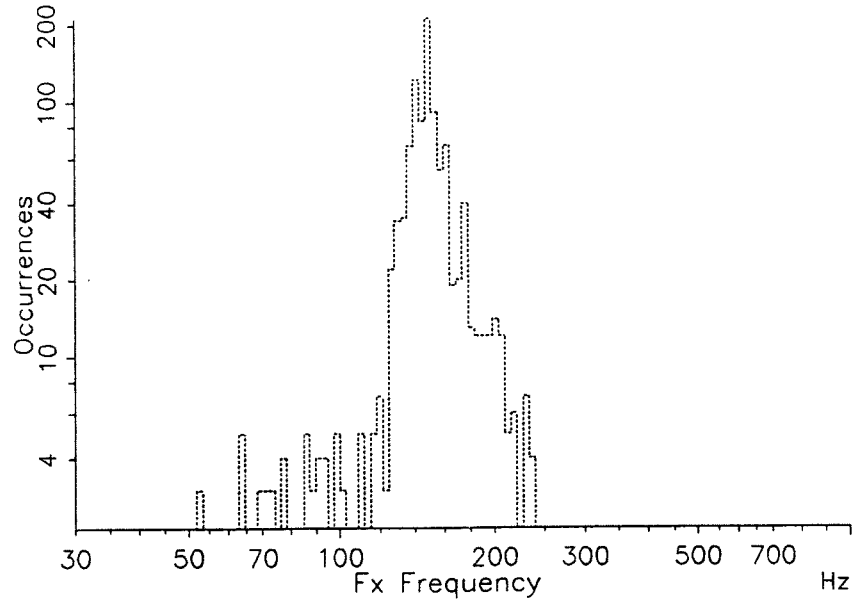
660/680 values Bin width: 2.8% Order: 1 Deviation: 2.8%



fjh.mar8 Speaker: jh Duration: 17.0 s

Mean: 145.6 Hz Sdev: 26 % Skewness: −2.2, Kurtosis: 9.0

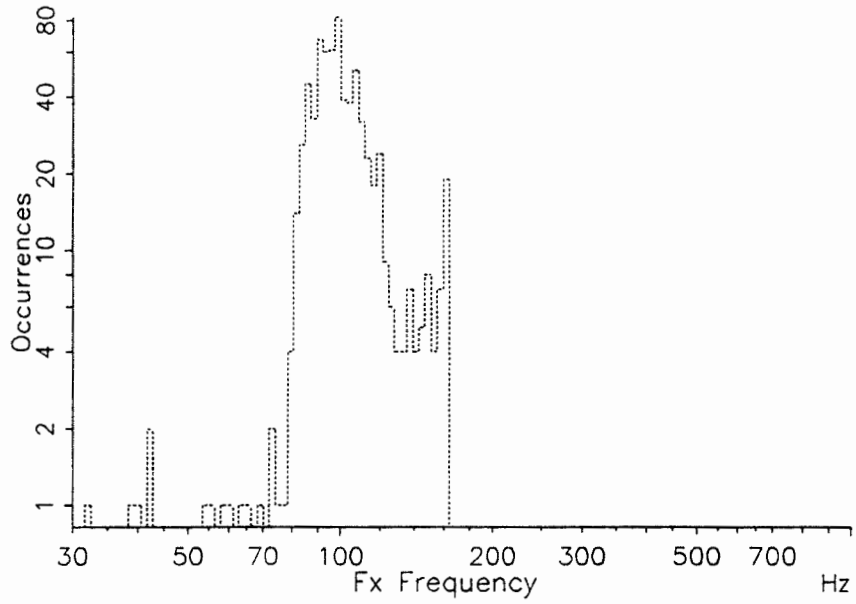
1063/1094 values Bin width: 2.8% Order: 1 Deviation: 2.8%



fre.mar3 Speaker: re Duration: 17.4 s

Mean: 100.7 Hz Sdev: 20 % Skewness: −0.4, Kurtosis: 5.5

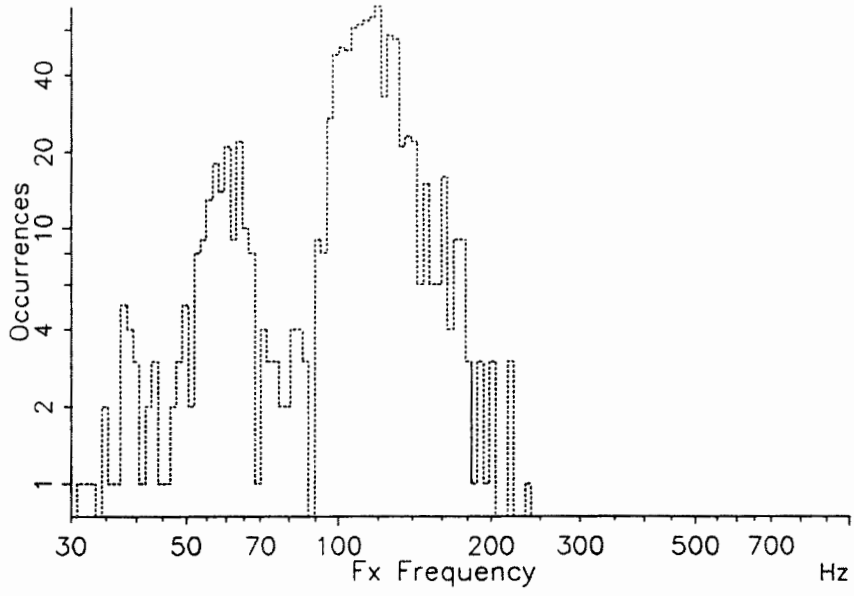
710/743 values Bin width: 2.8% Order: 1 Deviation: 2.8%



fCajf.mar8 Speaker: ajf Duration: 20.0 s

Mean: 103.3 Hz Sdev: 39 % Skewness: −1.1, Kurtosis: 1.2

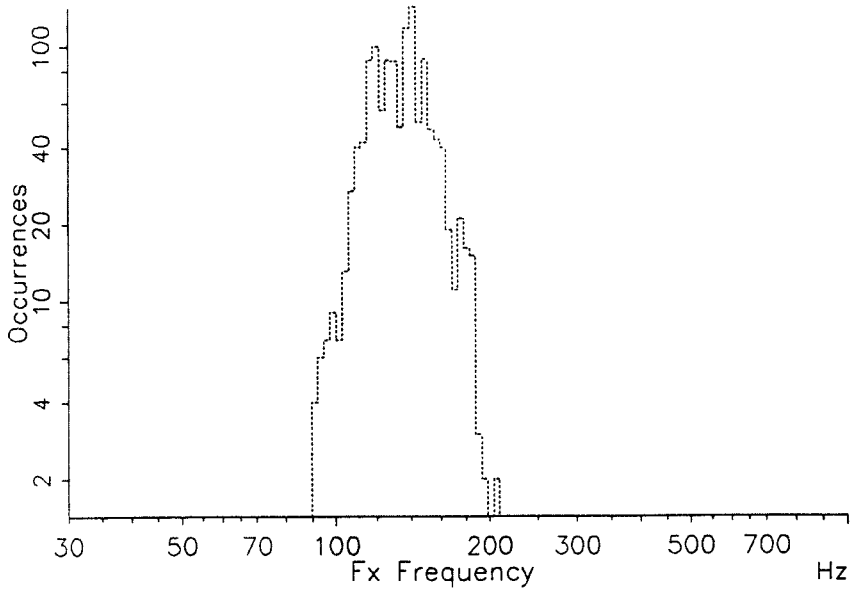
1017/1045 values Bin width: 2.8% Order: 1 Deviation: 2.8%



frb.french Speaker: rb Duration: 20.0 s

Mean: 133.6 Hz Sdev: 17 % Skewness: −0.9, Kurtosis: 6.3

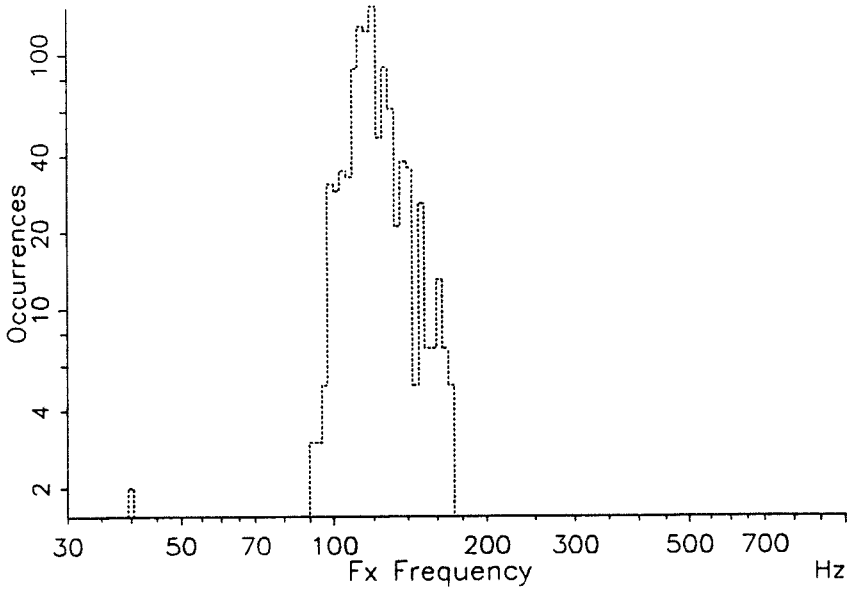
1247/1303 values Bin width: 2.8% Order: 1 Deviation: 2.8%



fjd.mar1 Speaker: jd Duration: 18.4 s

Mean: 120.0 Hz Sdev: 14 % Skewness: −1.4, Kurtosis: 16.0

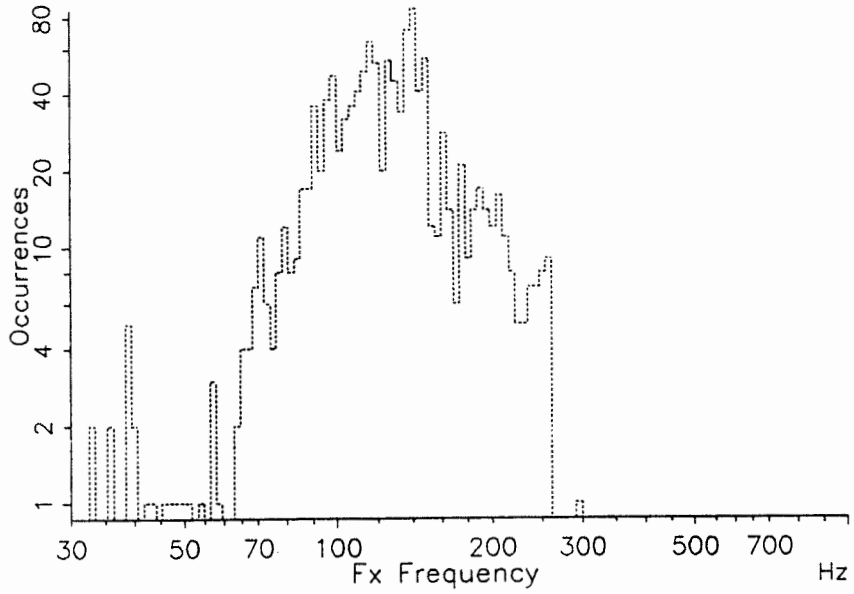
1000/1041 values Bin width: 2.8% Order: 1 Deviation: 2.8%



fsa.mar6 Speaker: sa Duration: 19.4 s

Mean: 124.1 Hz Sdev: 36 % Skewness: −0.5, Kurtosis: 1.7

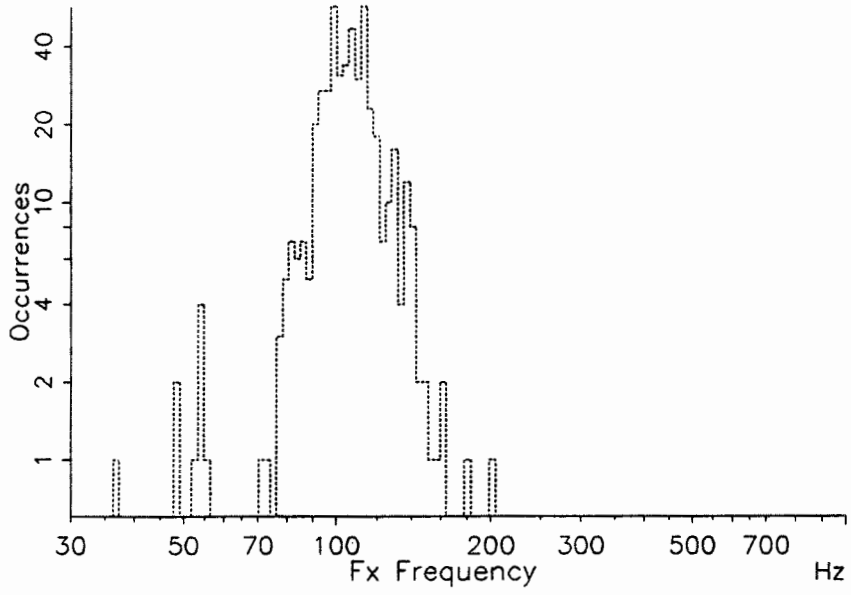
1213/1240 values Bin width: 2.8% Order: 1 Deviation: 2.8%



fcb.mrp3 Speaker: cb Duration: 9.8 s

Mean: 105.2 Hz Sdev: 19 % Skewness: −1.2, Kurtosis: 6.5

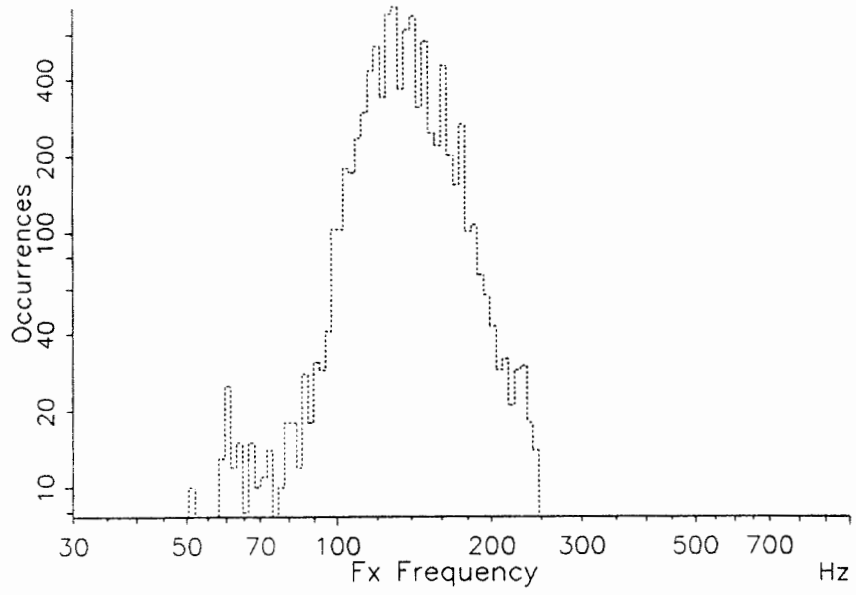
482/502 values Bin width: 2.8% Order: 1 Deviation: 2.8%



11 files, Speaker: IH Duration: 147.2 s

Mean: 133.7 Hz Sdev: 25 % Skewness: −1.1, Kurtosis: 6.3

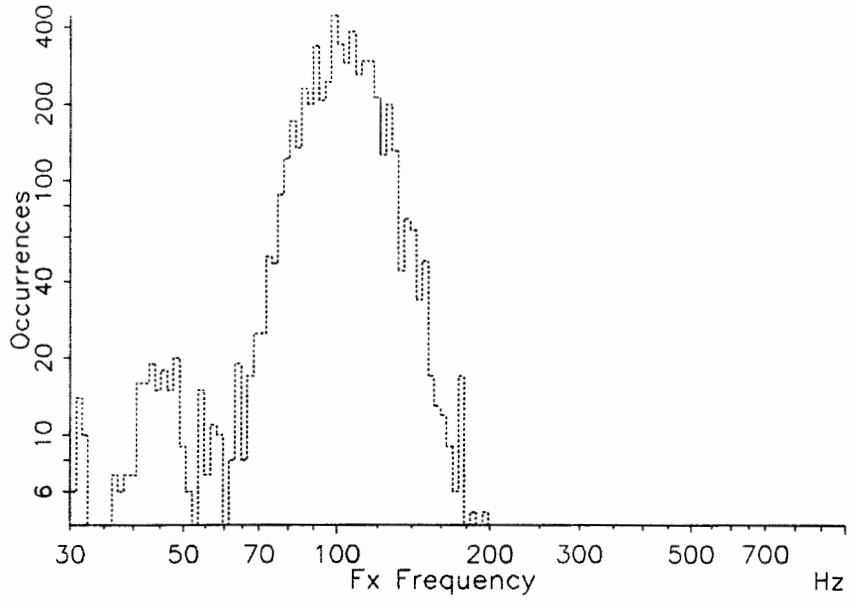
8988/9296 values Bin width: 2.8% Order: 1 Deviation: 2.8%



12 files, Speaker: DM Duration: 155.8 s

Mean: 98.8 Hz Sdev: 28 % Skewness: −1.5, Kurtosis: 4.3

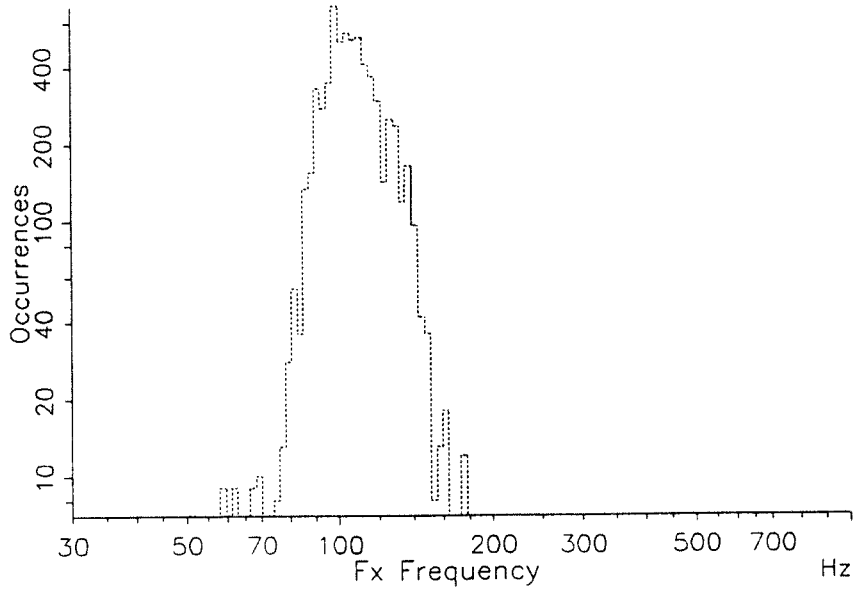
5517/5940 values Bin width: 2.8% Order: 1 Deviation: 2.8%



10 files, Speaker: RB Duration: 140.2 s

Mean: 106.3 Hz Sdev: 19 % Skewness: −1.4, Kurtosis: 8.0

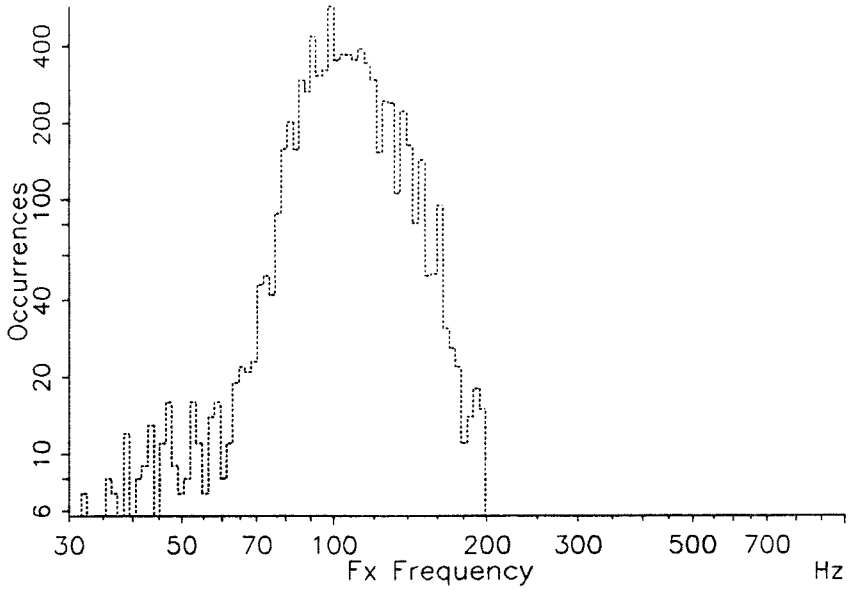
6536/6842 values Bin width: 2.8% Order: 1 Deviation: 2.8%



12 files, Speaker: JD Duration: 168.8 s

Mean: 103.7 Hz Sdev: 28 % Skewness: −0.8, Kurtosis: 3.6

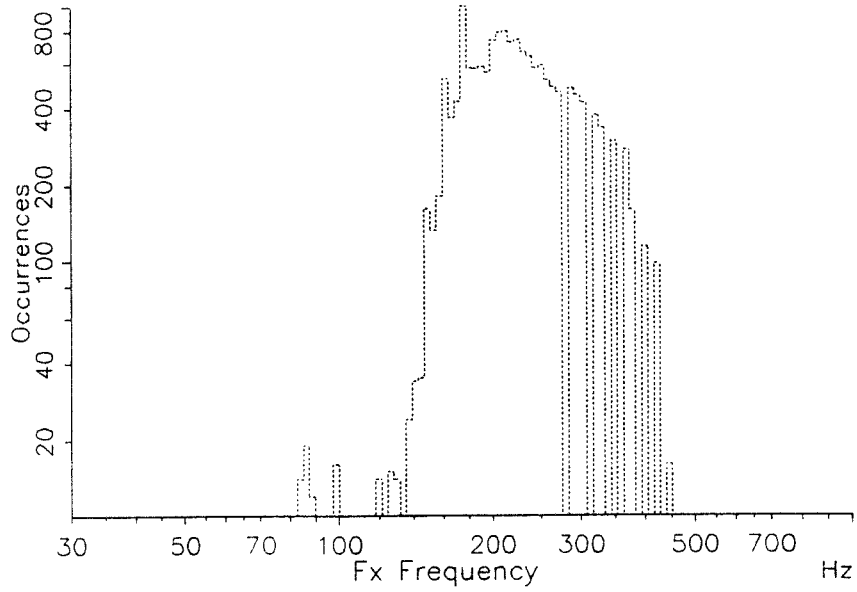
7423/7795 values Bin width: 2.8% Order: 1 Deviation: 2.8%



11 files, Speaker: SJH Duration: 156.4 s

Mean: 221.1 Hz Sdev: 31 % Skewness: −0.6, Kurtosis: 4.2

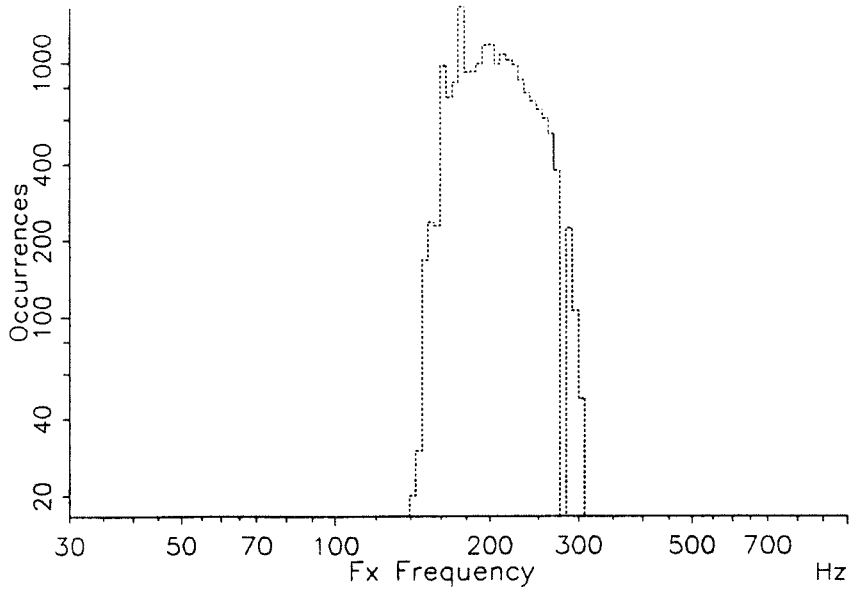
16177/16508 values Bin width: 2.8% Order: 1 Deviation: 2.8%



14 files, Speaker: VB Duration: 207.8 s

Mean: 201.1 Hz Sdev: 21 % Skewness: −1.4, Kurtosis: 13.2

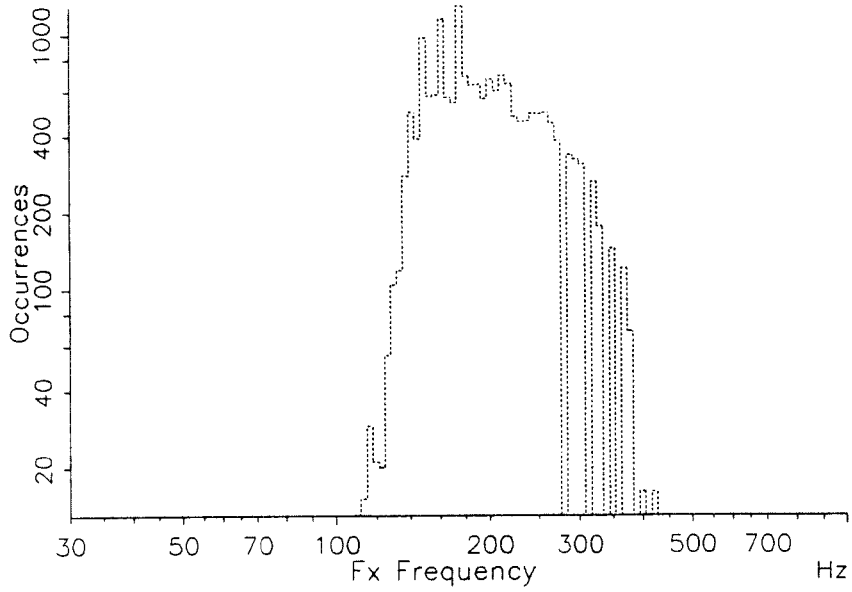
19192/19635 values Bin width: 2.8% Order: 1 Deviation: 2.8%



14 files, Speaker: EA Duration: 195.4 s

Mean: 195.9 Hz Sdev: 30 % Skewness: −0.2, Kurtosis: 2.6

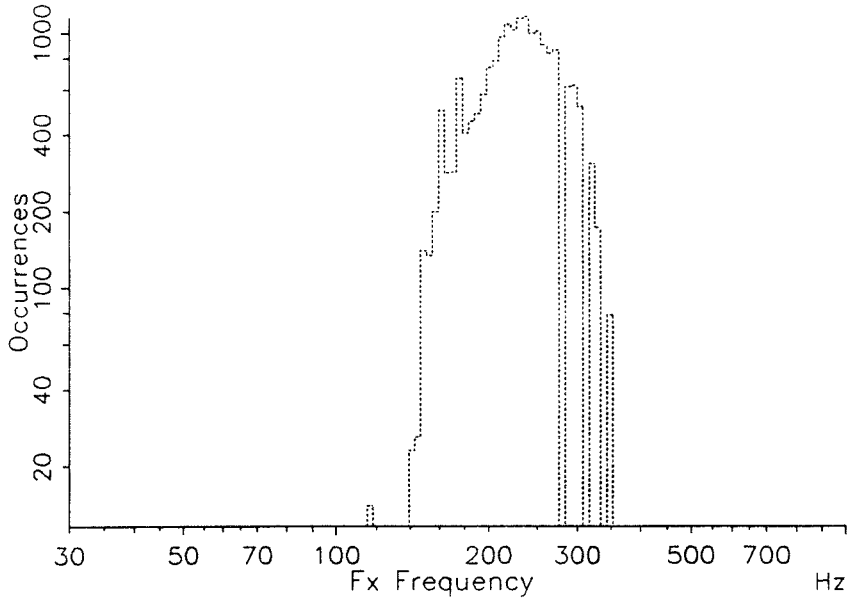
17895/18283 values Bin width: 2.8% Order: 1 Deviation: 2.8%



11 files, Speaker: MD Duration: 168.4 s

Mean: 224.0 Hz Sdev: 24 % Skewness: −1.4, Kurtosis: 9.5

18144/18457 values Bin width: 2.8% Order: 1 Deviation: 2.8%



APPENDIX A.9: RESULTS FROM PRELIMINARY TESTING DATA

The results of the investigation of MLP parameters for the direct speech, wideband filterbank and auditory filterbank experiments are given here in the appendix.

KEY TO EXPERIMENTS

Adaption refers to the use of the adaption of the learning rate and momentum terms for the training of the MLP.

Selective emphasis refers to the use of selective emphasis training of the MLP.

Update refers to the number of patterns that were used to estimate the weight updates during training.

TITLE- Comparing training parameters using TMS filterbank.

PURPOSE - Show effect of training parameters.

MLP network structure 246-6-6-1

Average results given for 10 women from evaluation data set.

All trained for 1 pass over all female training data, except last experiment.

ue0a0 - trained selective emphasis, update=1, no adaption.

ue0a2 - trained selective emphasis, update=1, adaption.

ue2a0 - trained selective emphasis, update=100, no adaption.

ue2a2 - trained selective emphasis, update=100, adaption.

ue3a2 - trained selective emphasis, update=1000, adaption.

u1a0n - no selective emphasis, update=1, no adaption

u1a0nl - ignore uncertain zones, update=1, no adaption

u1u100u1000 trained selective emphasis for three passes: pass 1, update=1, no adaption; pass 2, update=100, adaption; pass 3, update=1000, adaption.

TITLE- Comparing training data using sp.

PURPOSE - Show effect of MLP structure and using male, female, and male + female training data with direct speech input to the MLP.

Average results given separately for 10 men and 10 women from evaluation data set.

All input windows are symmetrical.

All MLPs trained selective emphasis for three passes: pass 1, update=1, no adaption; pass 2, update=100, adaption; pass 3, update=1000, adaption. An overall pass corresponds to the three stages above.

sp664f32 - MLP 161-10-1, trained on women, 2 overall passes.

sp663m23 - MLP 161-10-1, trained on men, 2 overall passes.

exp1p3 - MLP 161-10-1, trained on men and women.

exp2p3 - MLP 161-5-1, trained on men and women.

exp4p3 - MLP 161-20-1, trained on men and women.

exp5p3 - MLP 161-1, trained on men and women.

exp6p3 - MLP 161-10-10-1, trained on men and women.

exp7p3 - MLP 321-10-1, trained on men and women.

exp8p3 - MLP 321-10-1, trained on men, 2 overall passes.

TITLE- Comparing training data using fb.

PURPOSE - Show effect of MLP structure and using male + female training data with reduced wideband filterbank input to the MLP.

Average results given separately for 10 men and 10 women from evaluation data set.

Offset refers to the relationship between the current frame and the start of the window. If the offset is not half the window-1, then the window is asymmetric.

All MLPs trained selective emphasis for three passes over male and female training data: pass 1, update=1, no adaption; pass 2, update=100, adaption; pass 3, update=1000, adaption.

fbexp10p3 - MLP 246-1, offset=20 frames.

fbexp11p3 - MLP 246-3-1, offset=20 frames.

fbexp12p3 - MLP 246-6-1, offset=20 frames.

fbexp13p3 - MLP 246-12-1, offset=20 frames.

fbexp14p3 - MLP 246-6-6-1, offset=20 frames.

TITLE- Comparing auditory filterbank parameters.

PURPOSE - Show effect of MLP structure and using male and male + female training data with simple auditory filterbank input to the MLP.

Average results given separately for 10 men and 10 women from evaluation data set.

All auditory filterbanks use 1ERB filter spacings.

All input windows symmetrical and 20.5 ms.

All MLPs trained selective emphasis for three passes over training data: pass 1, update=1, no adaption; pass 2, update=100, adaption; pass 3, update=1000, adaption.

afb1k66f - MLP 533-6-1, trained on women, 50-1kHz filter range.

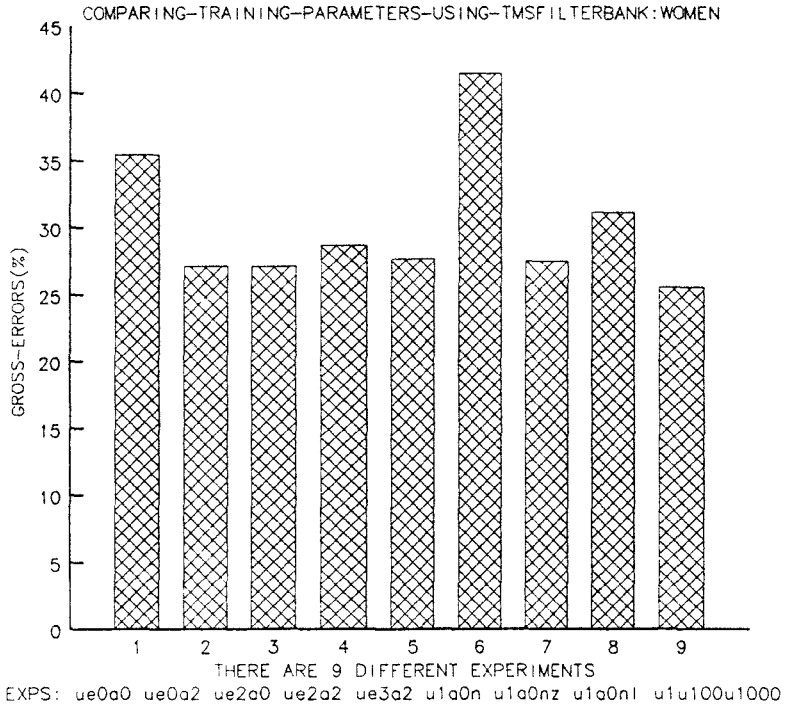
afb3k66f - MLP 943-6-1, trained on women, 50-3kHz filter range.

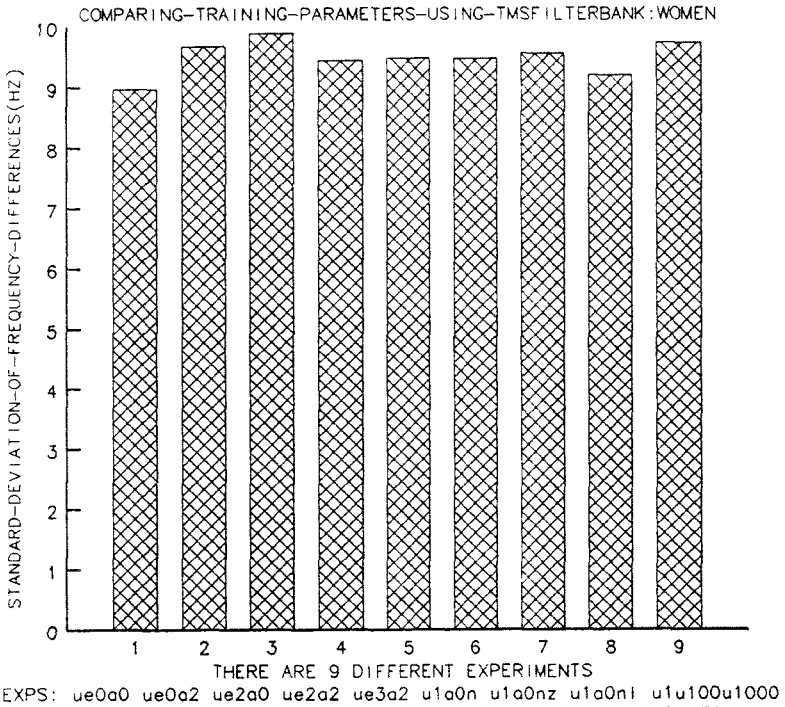
afbexp20 - MLP 943-1, trained on women, 50-3kHz filter range.

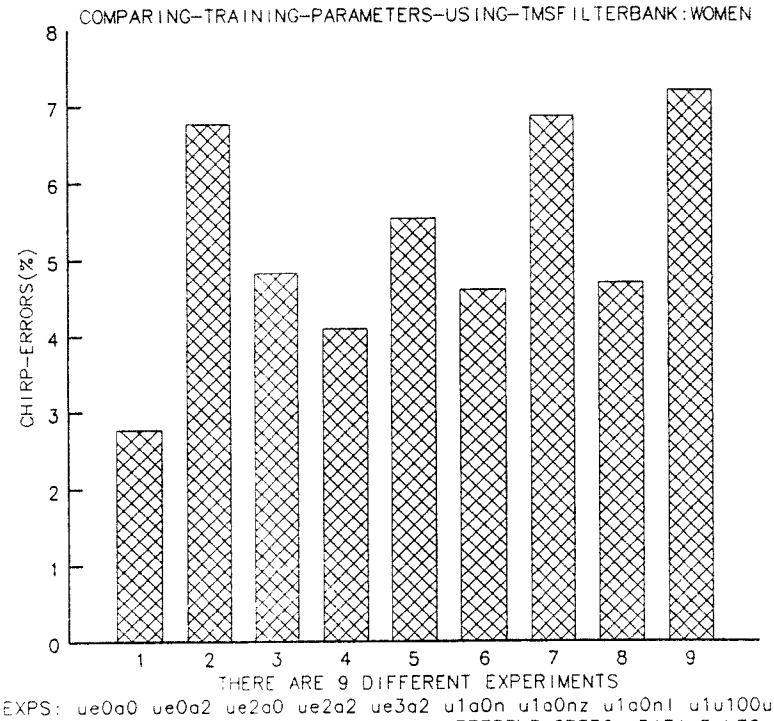
afbexp21 - MLP 943-3-1, trained on women, 50-3kHz filter range.

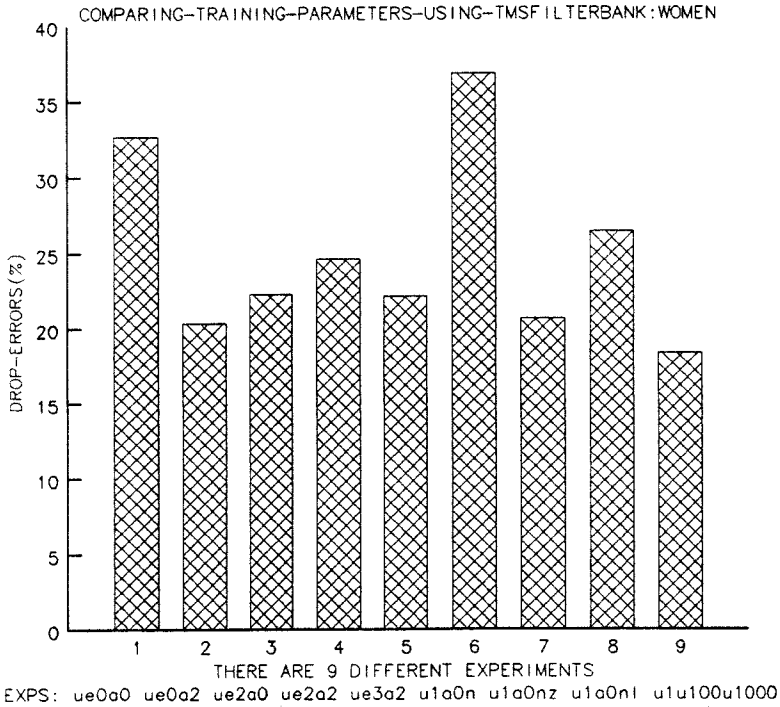
afbexp22 - MLP 943-12-1, trained on women, 50-3kHz filter range.

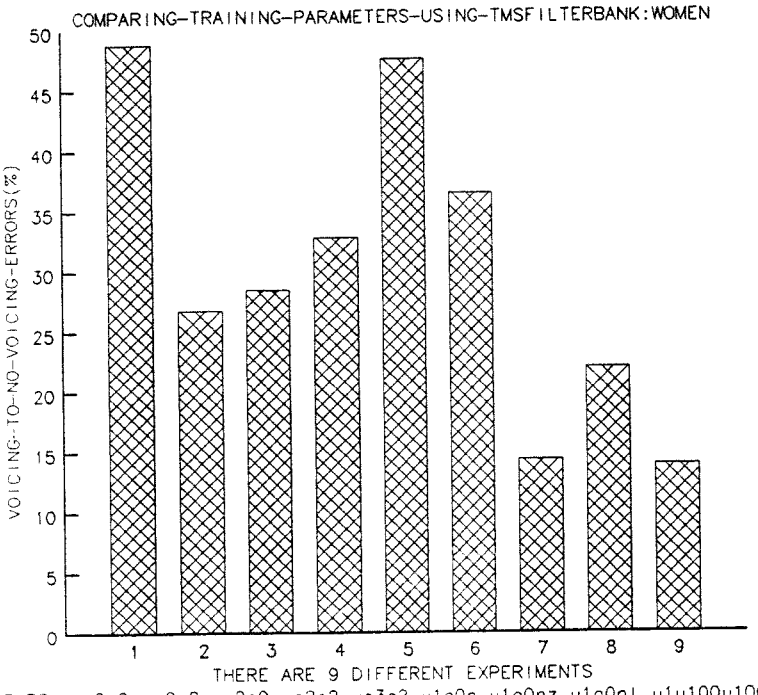
afbexp23 - MLP 943-6-6-1, trained on women, 50-3kHz filter range.

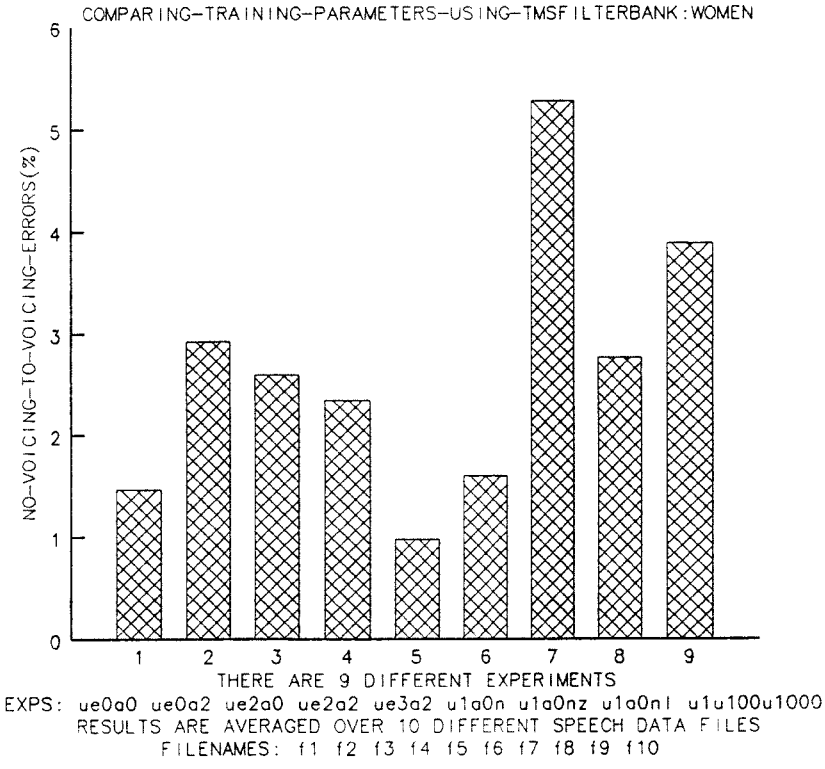

COMPARING-TRAINING-PARAMETERS-USING-TMSFILTERBANK:WOMEN
GROSS-ERRORS(%)
0
5
10
15
20
25
30
35
40
45
1
2
3
4
5
6
7
8
9
THERE ARE 9 DIFFERENT EXPERIMENTS
EXPS: ue0a0 ue0a2 ue2a0 ue2a2 ue3a2 u1a0n u1a0nz u1a0nl u1u100u1000
RESULTS ARE AVERAGED OVER 10 DIFFERENT SPEECH DATA FILES
FILENAMES: f1 f2 f3 f4 f5 f6 f7 f8 f9 f10

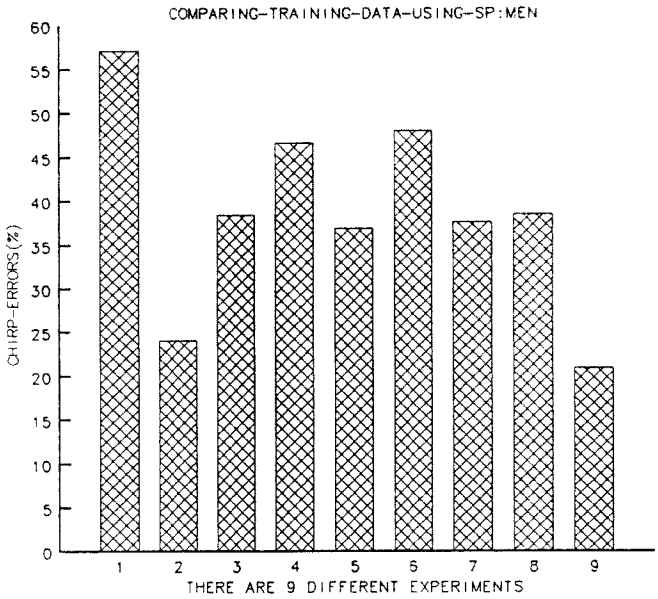

COMPARING-TRAINING-PARAMETERS-USING-TMSFILTERBANK:WOMEN
STANDARD-DEVIATION-OF-FREQUENCY-DIFFERENCES(HZ)
0
1
2
3
4
5
6
7
8
9
10
1
2
3
4
5
6
7
8
9
THERE ARE 9 DIFFERENT EXPERIMENTS
EXPS: ue0a0 ue0a2 ue2a0 ue2a2 ue3a2 u1a0n u1a0nz u1a0nl u1u100u1000
RESULTS ARE AVERAGED OVER 10 DIFFERENT SPEECH DATA FILES
FILENAMES: f1 f2 f3 f4 f5 f6 f7 f8 f9 f10


COMPARING-TRAINING-PARAMETERS-USING-TMSFILTERBANK:WOMEN
CHIRP-ERRORS(%)
0
1
2
3
4
5
6
7
8
1
2
3
4
5
6
7
8
9
THERE ARE 9 DIFFERENT EXPERIMENTS
EXPS: ue0a0 ue0a2 ue2a0 ue2a2 ue3a2 u1a0n u1a0nz u1a0nl u1u100u1000
RESULTS ARE AVERAGED OVER 10 DIFFERENT SPEECH DATA FILES
FILENAMES: f1 f2 f3 f4 f5 f6 f7 f8 f9 f10

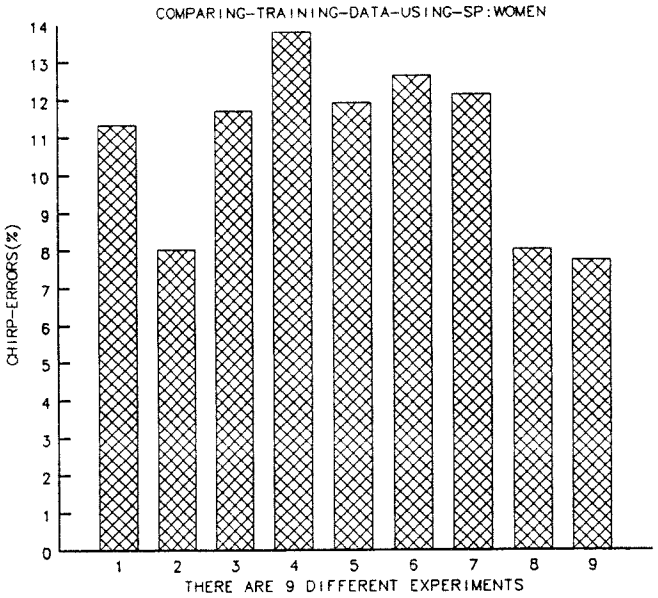

COMPARING-TRAINING-PARAMETERS-USING-TMSFILTERBANK:WOMEN
DROP-ERRORS(%)
0
5
10
15
20
25
30
35
40
1
2
3
4
5
6
7
8
9
THERE ARE 9 DIFFERENT EXPERIMENTS
EXPS: ue0a0 ue0a2 ue2a0 ue2a2 ue3a2 u1a0n u1a0nz u1a0nl u1u100u1000
RESULTS ARE AVERAGED OVER 10 DIFFERENT SPEECH DATA FILES
FILENAMES: f1 f2 f3 f4 f5 f6 f7 f8 f9 f10


COMPARING-TRAINING-PARAMETERS-USING-TMSFILTERBANK:WOMEN
VOICING-TO-NO-VOICING-ERRORS(%)
0
5
10
15
20
25
30
35
40
45
50
1
2
3
4
5
6
7
8
9
THERE ARE 9 DIFFERENT EXPERIMENTS
EXPS: ue0a0 ue0a2 ue2a0 ue2a2 ue3a2 u1a0n u1a0nz u1a0nl u1u100u1000
RESULTS ARE AVERAGED OVER 10 DIFFERENT SPEECH DATA FILES
FILENAMES: f1 f2 f3 f4 f5 f6 f7 f8 f9 f10

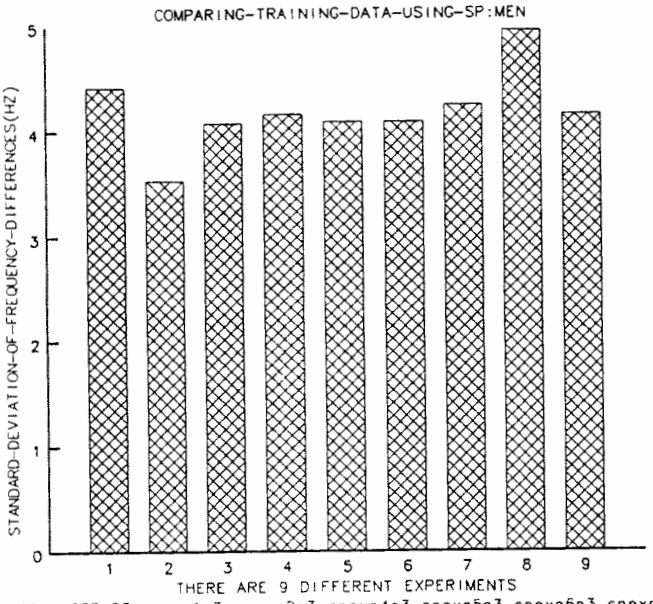

COMPARING-TRAINING-PARAMETERS-USING-TMSFILTERBANK:WOMEN
NO-VOICING-TO-VOICING-ERRORS(%)
0
1
2
3
4
5
6
1
2
3
4
5
6
7
8
9
THERE ARE 9 DIFFERENT EXPERIMENTS
EXPS: ue0a0 ue0a2 ue2a0 ue2a2 ue3a2 u1a0n u1a0nz u1a0nl u1u100u1000
RESULTS ARE AVERAGED OVER 10 DIFFERENT SPEECH DATA FILES
FILENAMES: f1 f2 f3 f4 f5 f6 f7 f8 f9 f10



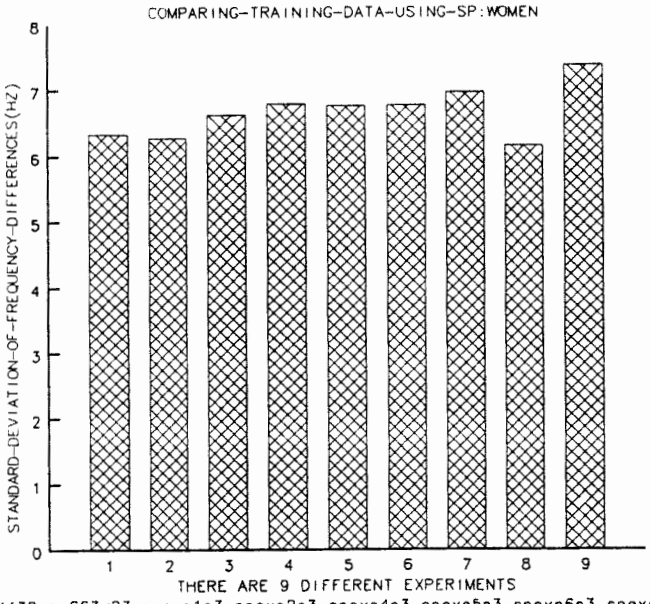
EXPS: sp664f32 sp663m23 spexp1p3 spexp2p3 spexp4p3 spexp5p3 spexp6p3 spexp7p3 spexp8p3
RESULTS ARE AVERAGED OVER 10 DIFFERENT SPEECH DATA FILES
FILENAMES: m1 m2 m3 m4 m5 m6 m7 m8 m9 m10



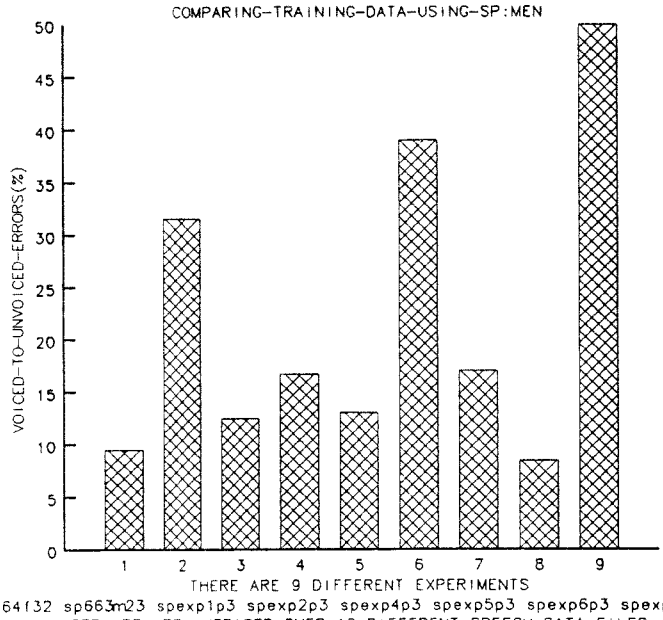
EXPS: sp664f32 sp663m23 spexp1p3 spexp2p3 spexp4p3 spexp5p3 spexp6p3 spexp7p3 spexp8p3
RESULTS ARE AVERAGED OVER 10 DIFFERENT SPEECH DATA FILES
FILENAMES: f1 f2 f3 f4 f5 f6 f7 f8 f9 f10



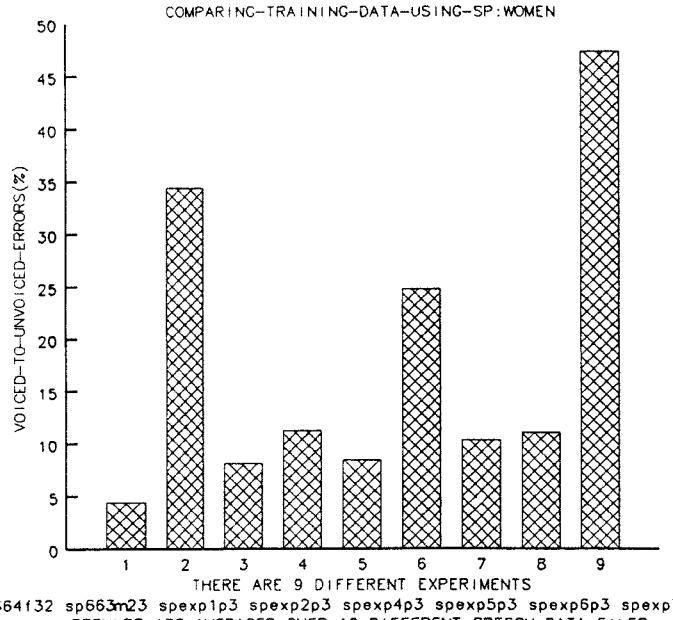
EXPS: sp664f32 sp663m23 spexp1p3 spexp2p3 spexp4p3 spexp5p3 spexp6p3 spexp7p3 spexp8p3
RESULTS ARE AVERAGED OVER 10 DIFFERENT SPEECH DATA FILES
FILENAMES: m1 m2 m3 m4 m5 m6 m7 m8 m9 m10



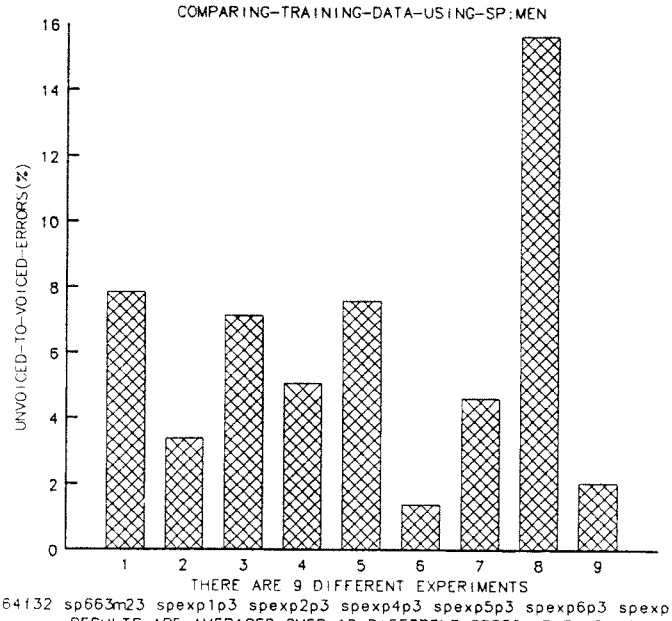
EXPS: sp664f32 sp663m23 spexp1p3 spexp2p3 spexp4p3 spexp5p3 spexp6p3 spexp7p3 spexp8p3
RESULTS ARE AVERAGED OVER 10 DIFFERENT SPEECH DATA FILES
FILENAMES: f1 f2 f3 f4 f5 f6 f7 f8 f9 f10



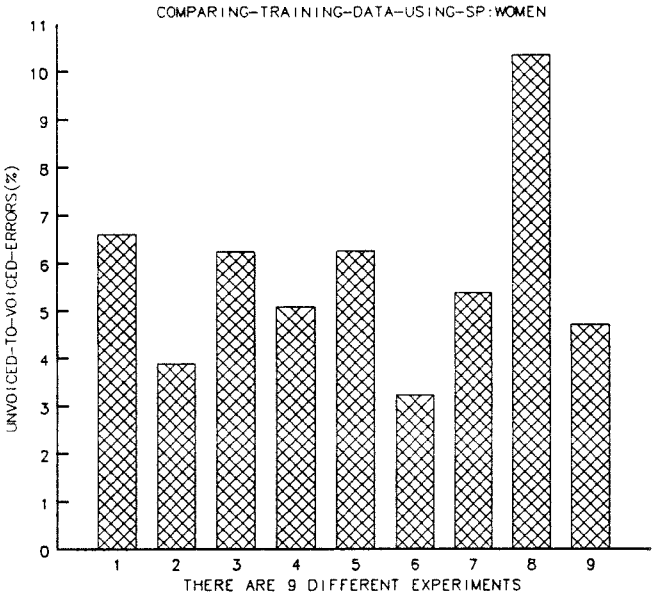
EXPS: sp664f32 sp663m23 spexp1p3 spexp2p3 spexp4p3 spexp5p3 spexp6p3 spexp7p3 spexp8p3
RESULTS ARE AVERAGED OVER 10 DIFFERENT SPEECH DATA FILES
FILENAMES: m1 m2 m3 m4 m5 m6 m7 m8 m9 m10



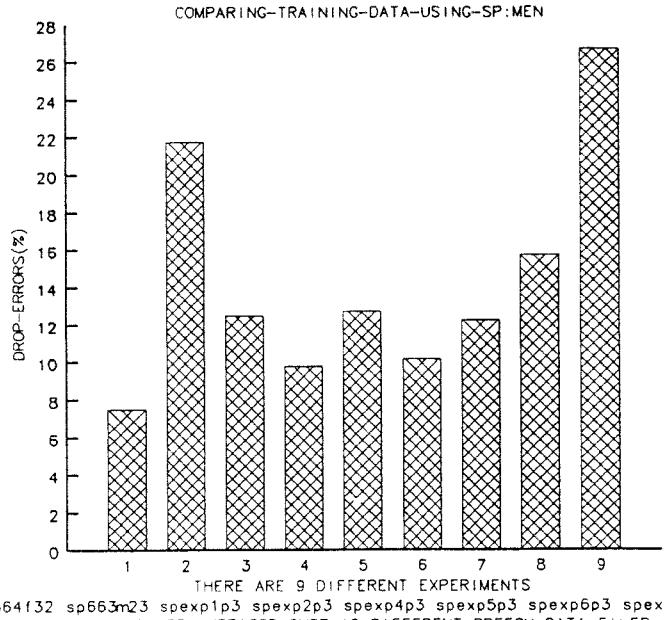
EXPS: sp664f32 sp663m23 spexp1p3 spexp2p3 spexp4p3 spexp5p3 spexp6p3 spexp7p3 spexp8p3
RESULTS ARE AVERAGED OVER 10 DIFFERENT SPEECH DATA FILES
FILENAMES: f1 f2 f3 f4 f5 f6 f7 f8 f9 f10



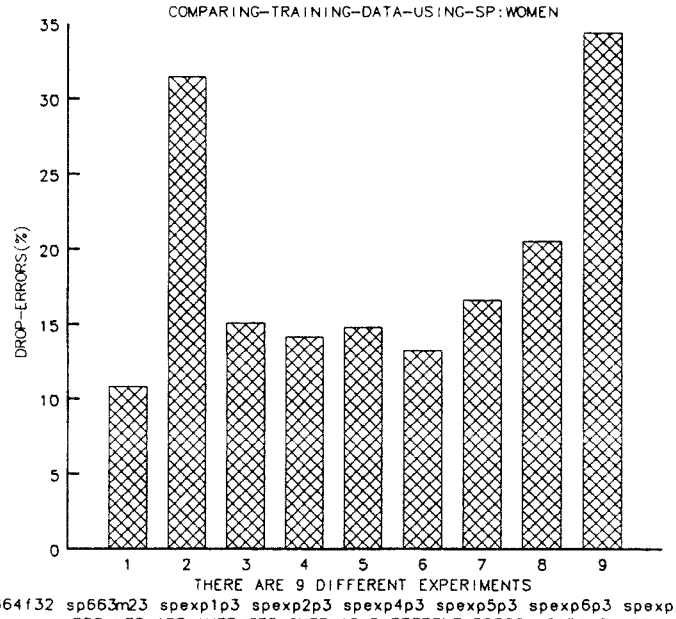
EXPS: sp664f32 sp663m23 spexp1p3 spexp2p3 spexp4p3 spexp5p3 spexp6p3 spexp7p3 spexp8p3
RESULTS ARE AVERAGED OVER 10 DIFFERENT SPEECH DATA FILES
FILENAMES: m1 m2 m3 m4 m5 m6 m7 m8 m9 m10



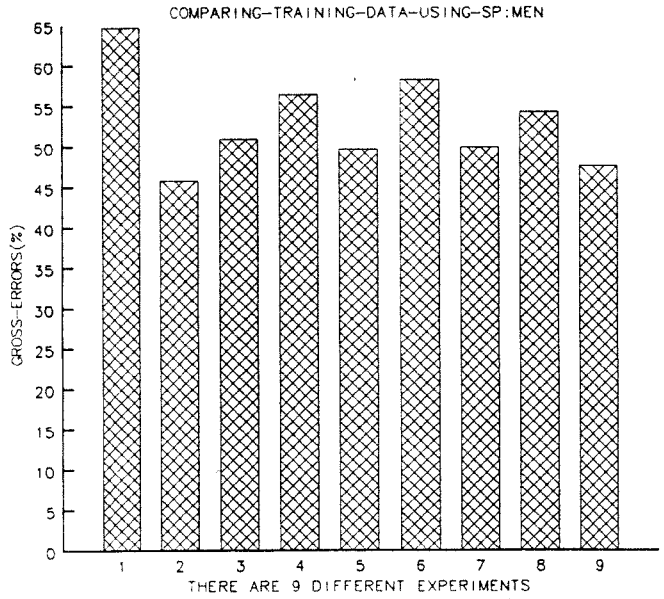
EXPS: sp664f32 sp663m23 spexp1p3 spexp2p3 spexp4p3 spexp5p3 spexp6p3 spexp7p3 spexp8p3
RESULTS ARE AVERAGED OVER 10 DIFFERENT SPEECH DATA FILES
FILENAMES: f1 f2 f3 f4 f5 f6 f7 f8 f9 f10



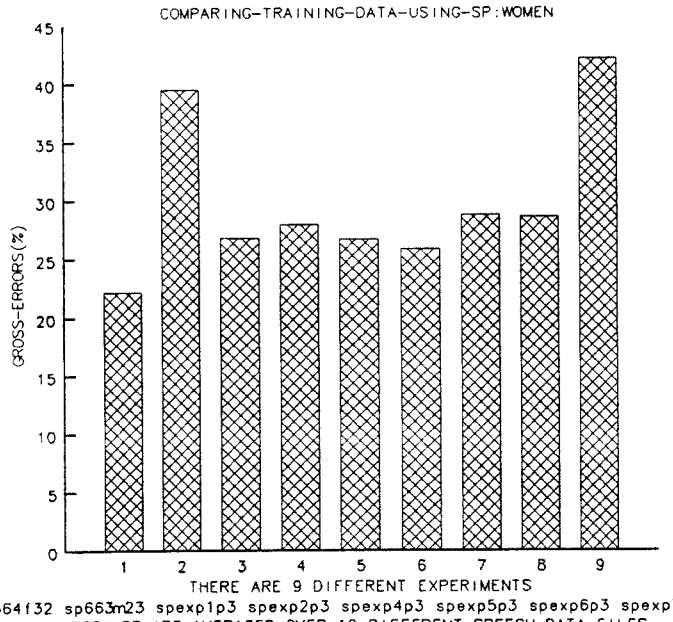
EXPS: sp664f32 sp663m23 spexp1p3 spexp2p3 spexp4p3 spexp5p3 spexp6p3 spexp7p3 spexp8p3
RESULTS ARE AVERAGED OVER 10 DIFFERENT SPEECH DATA FILES
FILENAMES: m1 m2 m3 m4 m5 m6 m7 m8 m9 m10



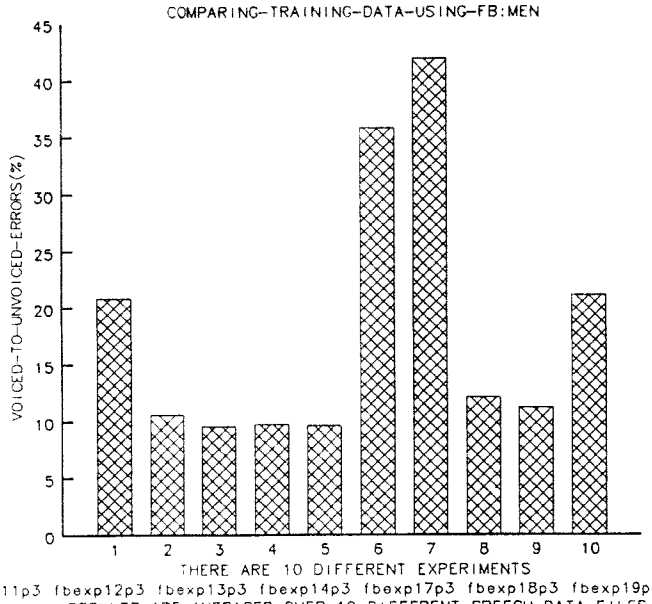
EXPS: sp664f32 sp663m23 spexp1p3 spexp2p3 spexp4p3 spexp5p3 spexp6p3 spexp7p3 spexp8p3
RESULTS ARE AVERAGED OVER 10 DIFFERENT SPEECH DATA FILES
FILENAMES: f1 f2 f3 f4 f5 f6 f7 f8 f9 f10



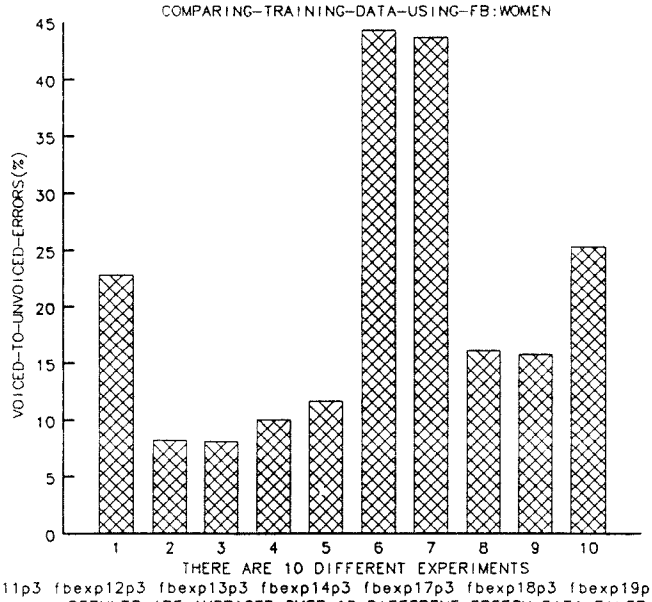
EXPS: sp664f32 sp663m23 spexp1p3 spexp2p3 spexp4p3 spexp5p3 spexp6p3 spexp7p3 spexp8p3
RESULTS ARE AVERAGED OVER 10 DIFFERENT SPEECH DATA FILES
FILENAMES: m1 m2 m3 m4 m5 m6 m7 m8 m9 m10



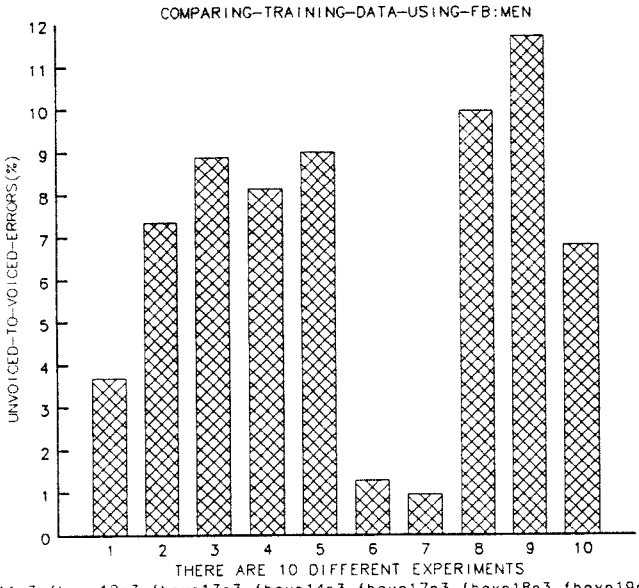
EXPS: sp664f32 sp663m23 spexp1p3 spexp2p3 spexp4p3 spexp5p3 spexp6p3 spexp7p3 spexp8p3
RESULTS ARE AVERAGED OVER 10 DIFFERENT SPEECH DATA FILES
FILENAMES: f1 f2 f3 f4 f5 f6 f7 f8 f9 f10



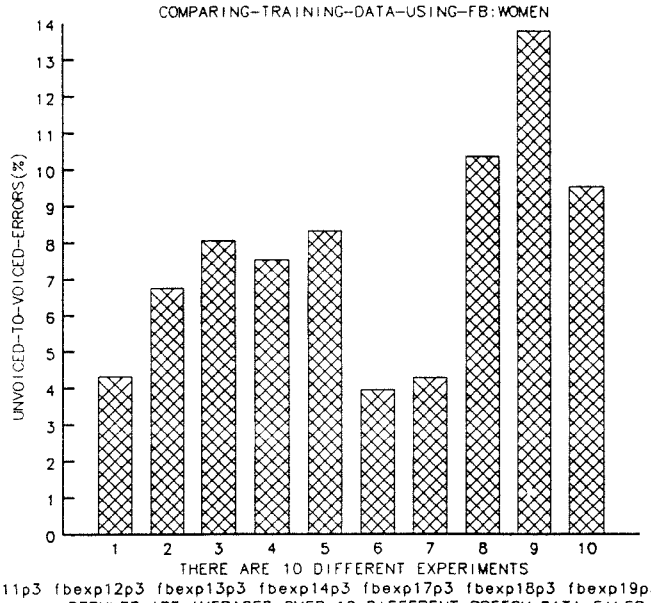
EXPS: fbexp10p3 fbexp11p3 fbexp12p3 fbexp13p3 fbexp14p3 fbexp17p3 fbexp18p3 fbexp19p3 exp14p3+con3 exp14p3+con
RESULTS ARE AVERAGED OVER 10 DIFFERENT SPEECH DATA FILES
FILENAMES: m1 m2 m3 m4 m5 m6 m7 m8 m9 m10



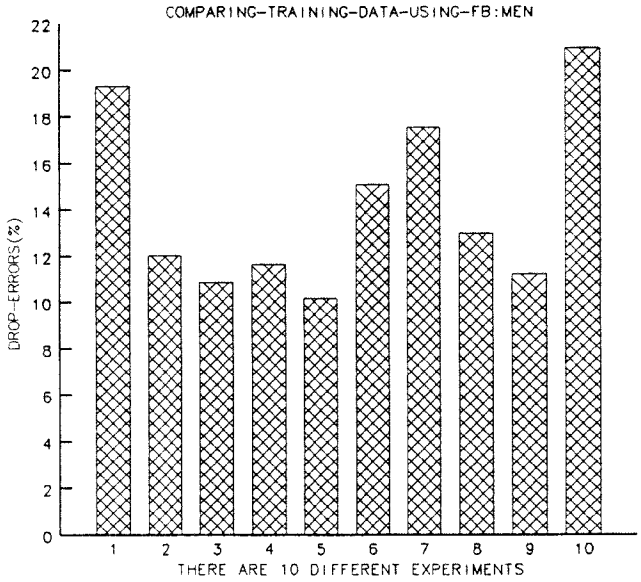
EXPS: fbexp10p3 fbexp11p3 fbexp12p3 fbexp13p3 fbexp14p3 fbexp17p3 fbexp18p3 fbexp19p3 exp14p3+con3 exp14p3+con
RESULTS ARE AVERAGED OVER 10 DIFFERENT SPEECH DATA FILES
FILENAMES: f1 f2 f3 f4 f5 f6 f7 f8 f9 f10



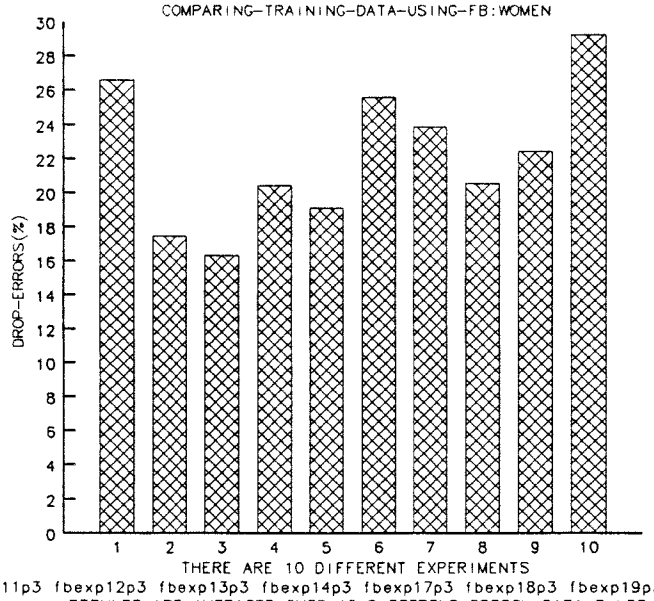
EXPS: fbexp10p3 fbexp11p3 fbexp12p3 fbexp13p3 fbexp14p3 fbexp17p3 fbexp18p3 fbexp19p3 exp14p3+con3 exp14p3+con
RESULTS ARE AVERAGED OVER 10 DIFFERENT SPEECH DATA FILES
FILENAMES: m1 m2 m3 m4 m5 m6 m7 m8 m9 m10



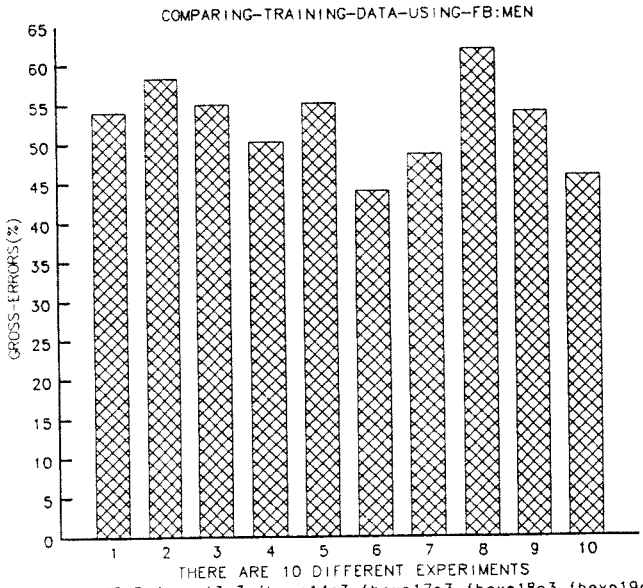
EXPS: fbexp10p3 fbexp11p3 fbexp12p3 fbexp13p3 fbexp14p3 fbexp17p3 fbexp18p3 fbexp19p3 exp14p3+con3 exp14p3+con
RESULTS ARE AVERAGED OVER 10 DIFFERENT SPEECH DATA FILES
FILENAMES: f1 f2 f3 f4 f5 f6 f7 f8 f9 f10



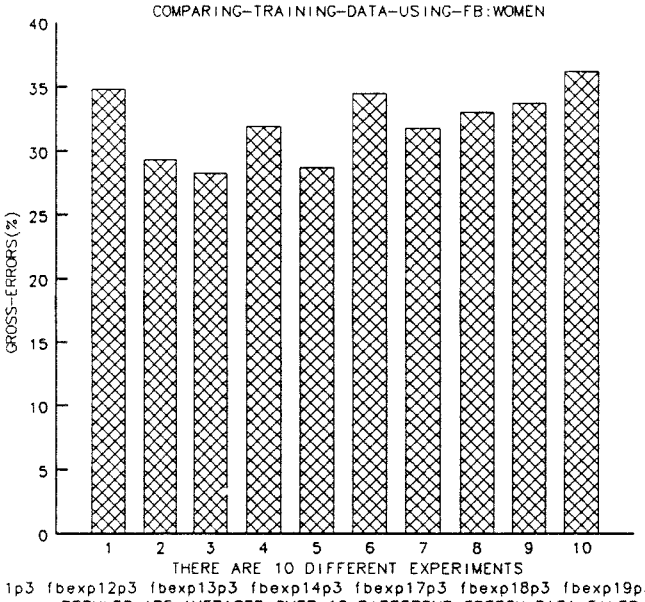
EXPS: fbexp10p3 fbexp11p3 fbexp12p3 fbexp13p3 fbexp14p3 fbexp17p3 fbexp18p3 fbexp19p3 exp14p3+con3 exp14p3+con
RESULTS ARE AVERAGED OVER 10 DIFFERENT SPEECH DATA FILES
FILENAMES: m1 m2 m3 m4 m5 m6 m7 m8 m9 m10



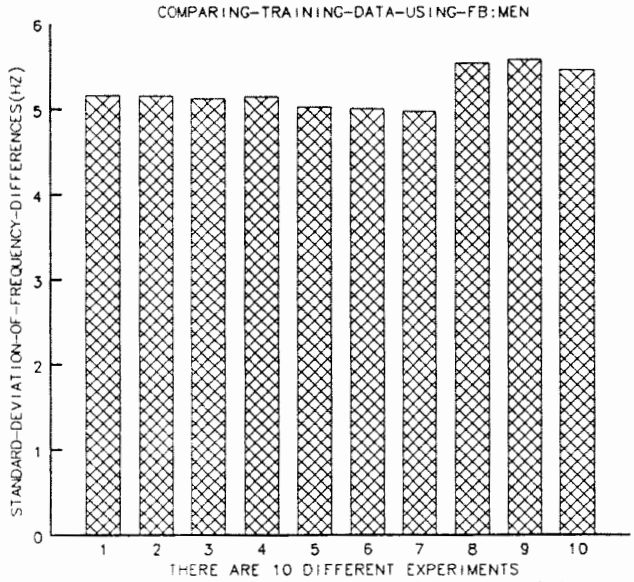
EXPS: fbexp10p3 fbexp11p3 fbexp12p3 fbexp13p3 fbexp14p3 fbexp17p3 fbexp18p3 fbexp19p3 exp14p3+con3 exp14p3+con
RESULTS ARE AVERAGED OVER 10 DIFFERENT SPEECH DATA FILES
FILENAMES: f1 f2 f3 f4 f5 f6 f7 f8 f9 f10



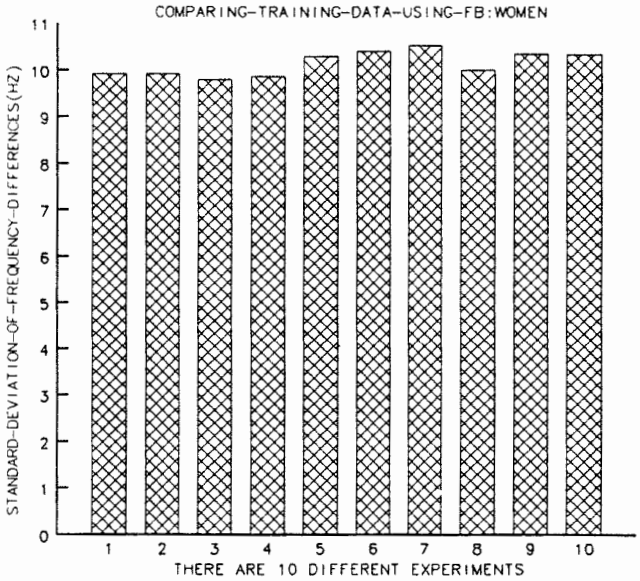
EXPS: fbexp10p3 fbexp11p3 fbexp12p3 fbexp13p3 fbexp14p3 fbexp17p3 fbexp18p3 fbexp19p3 exp14p3+con3 exp14p3+con
RESULTS ARE AVERAGED OVER 10 DIFFERENT SPEECH DATA FILES
FILENAMES: m1 m2 m3 m4 m5 m6 m7 m8 m9 m10



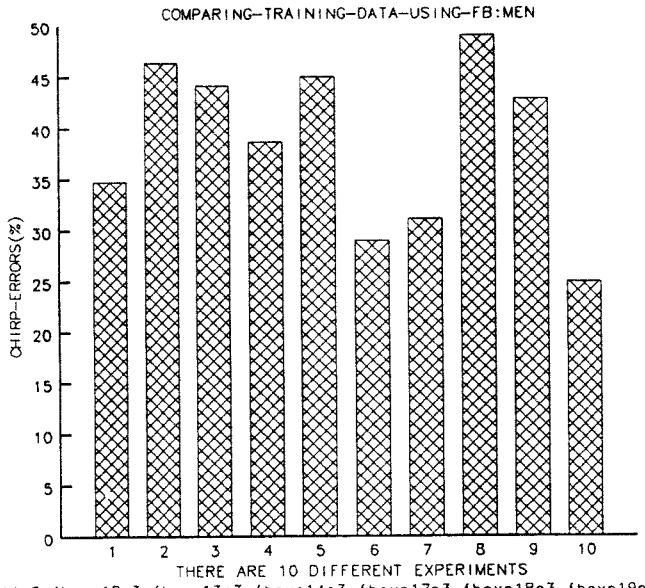
EXPS: fbexp10p3 fbexp11p3 fbexp12p3 fbexp13p3 fbexp14p3 fbexp17p3 fbexp18p3 fbexp19p3 exp14p3+con3 exp14p3+con
RESULTS ARE AVERAGED OVER 10 DIFFERENT SPEECH DATA FILES
FILENAMES: f1 f2 f3 f4 f5 f6 f7 f8 f9 f10



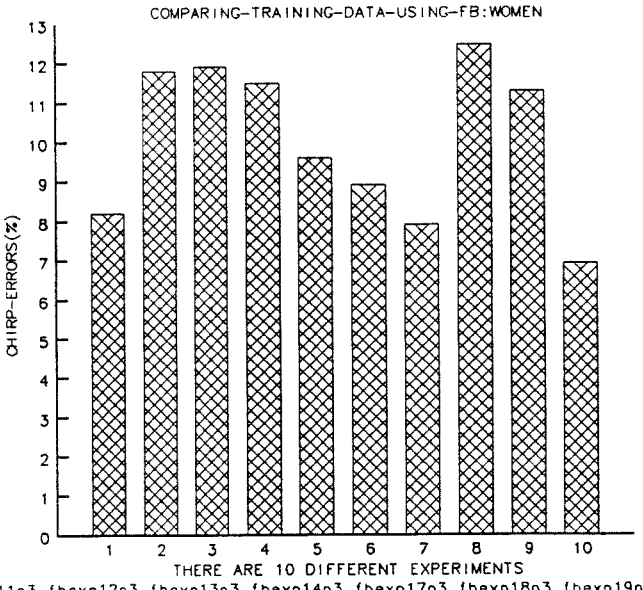
EXPS: fbexp10p3 fbexp11p3 fbexp12p3 fbexp13p3 fbexp14p3 fbexp17p3 fbexp18p3 fbexp19p3 exp14p3+con3 exp14p3+con
RESULTS ARE AVERAGED OVER 10 DIFFERENT SPEECH DATA FILES
FILENAMES: m1 m2 m3 m4 m5 m6 m7 m8 m9 m10



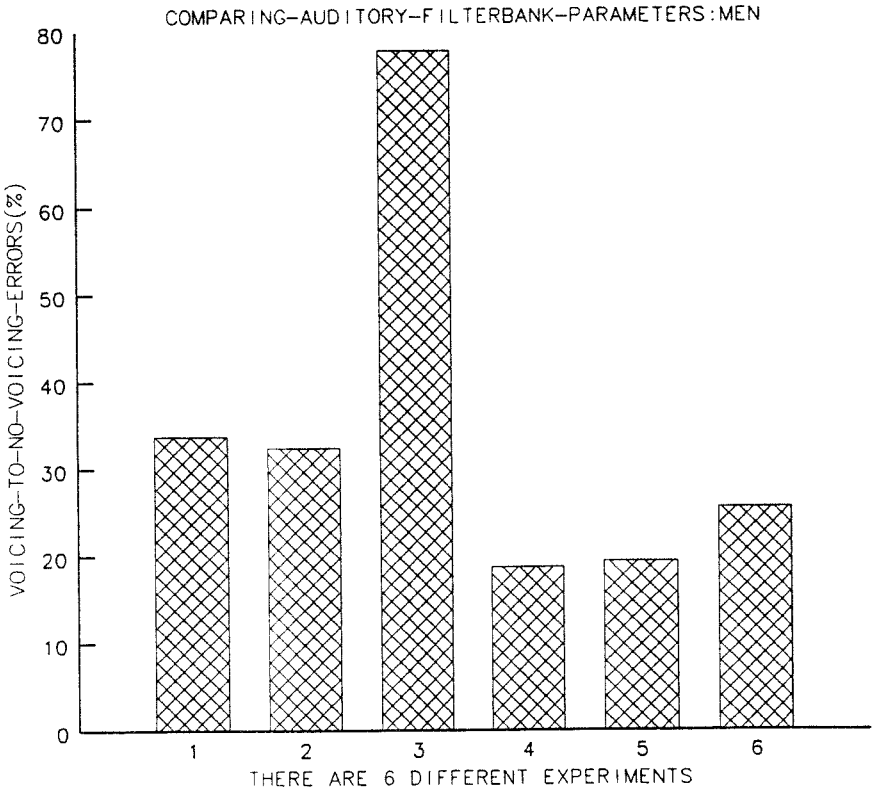
EXPS: fbexp10p3 fbexp11p3 fbexp12p3 fbexp13p3 fbexp14p3 fbexp17p3 fbexp18p3 fbexp19p3 exp14p3+con3 exp14p3+con
RESULTS ARE AVERAGED OVER 10 DIFFERENT SPEECH DATA FILES
FILENAMES: f1 f2 f3 f4 f5 f6 f7 f8 f9 f10

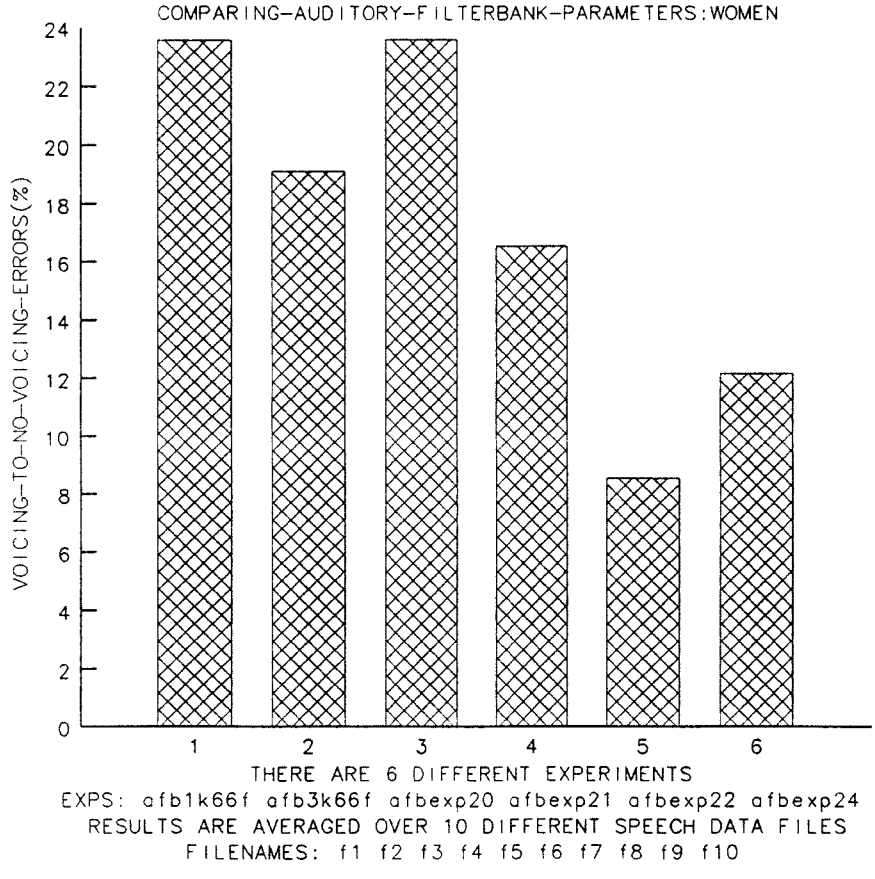


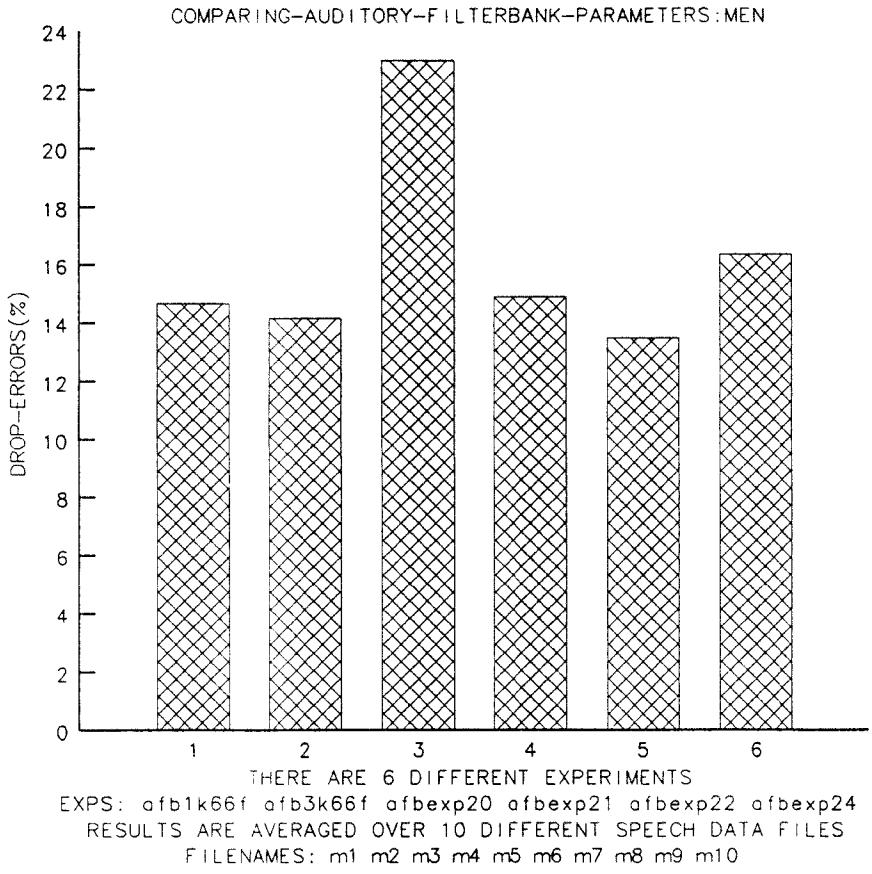
EXPS: fbexp10p3 fbexp11p3 fbexp12p3 fbexp13p3 fbexp14p3 fbexp17p3 fbexp18p3 fbexp19p3 exp14p3+con3 exp14p3+con
RESULTS ARE AVERAGED OVER 10 DIFFERENT SPEECH DATA FILES
FILENAMES: m1 m2 m3 m4 m5 m6 m7 m8 m9 m10

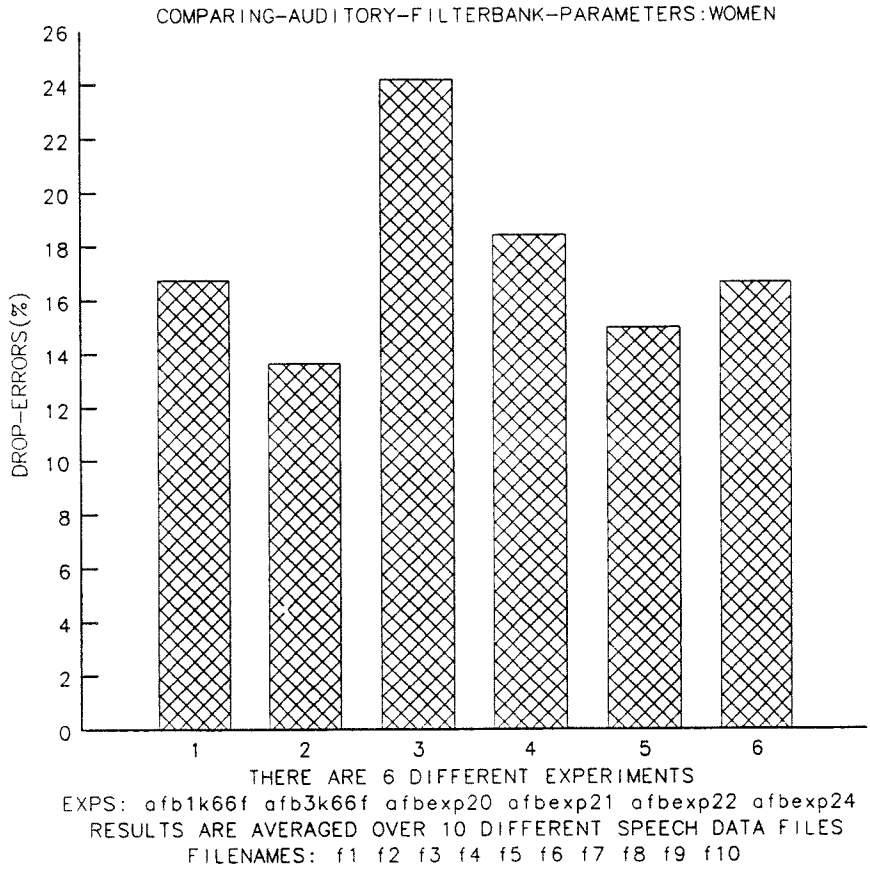


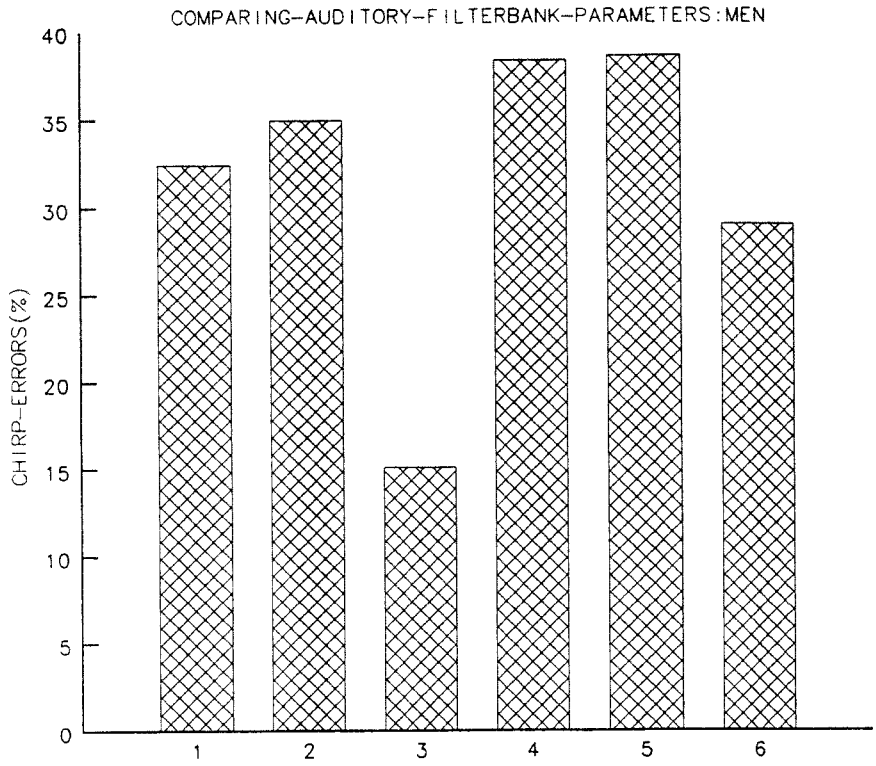
EXPS: fbexp10p3 fbexp11p3 fbexp12p3 fbexp13p3 fbexp14p3 fbexp17p3 fbexp18p3 fbexp19p3 exp14p3+con3 exp14p3+con
RESULTS ARE AVERAGED OVER 10 DIFFERENT SPEECH DATA FILES
FILENAMES: f1 f2 f3 f4 f5 f6 f7 f8 f9 f10

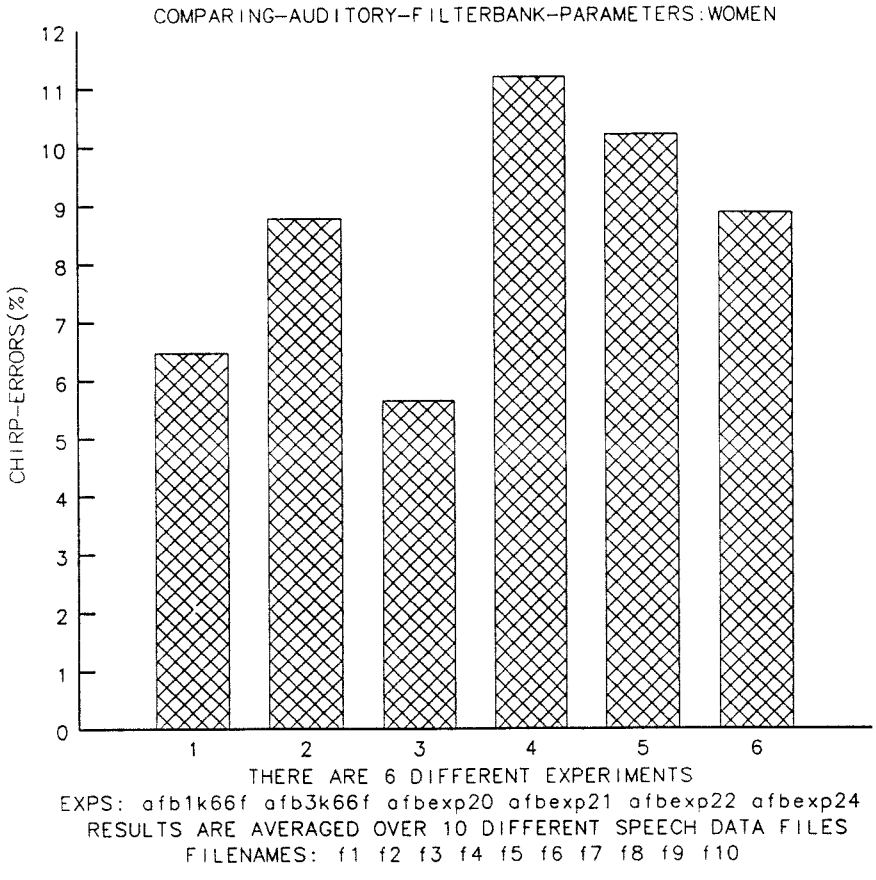

COMPARING-AUDITORY-FILTERBANK-PARAMETERS:MEN
VOICING-TO-NO-VOICING-ERRORS(%)
0
10
20
30
40
50
60
70
80
1
2
3
4
5
6
THERE ARE 6 DIFFERENT EXPERIMENTS
EXPS: afb1k66f afb3k66f afbexp20 afbexp21 afbexp22 afbexp24
RESULTS ARE AVERAGED OVER 10 DIFFERENT SPEECH DATA FILES
FILENAMES: m1 m2 m3 m4 m5 m6 m7 m8 m9 m10

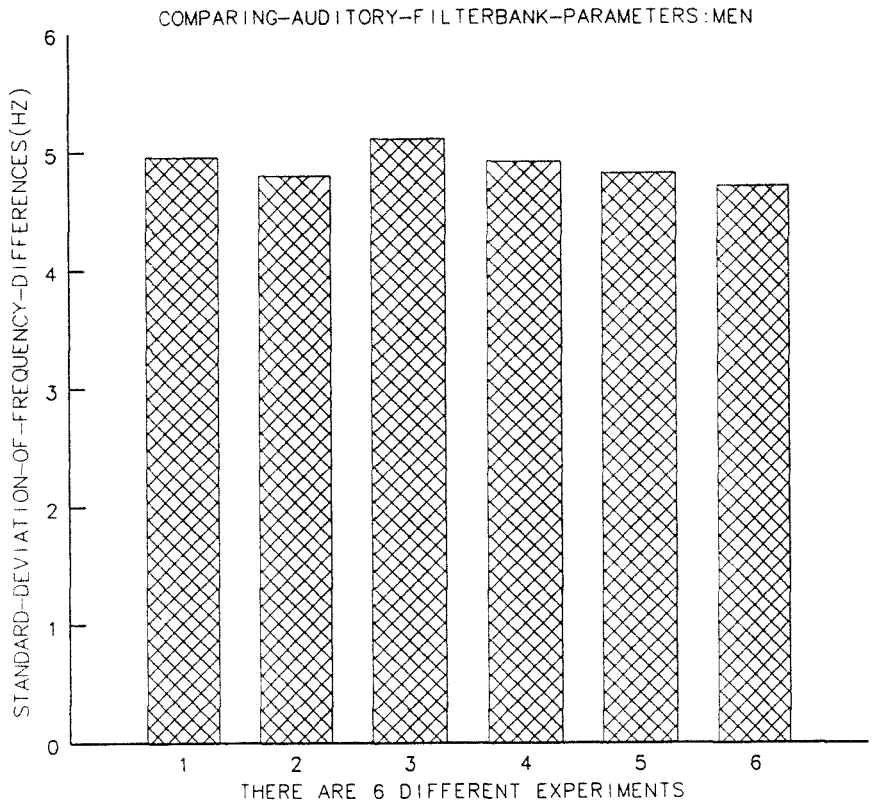

COMPARING-AUDITORY-FILTERBANK-PARAMETERS:WOMEN
VOICING-TO-NO-VOICING-ERRORS(%)
0
2
4
6
8
10
12
14
16
18
20
22
24
1
2
3
4
5
6
THERE ARE 6 DIFFERENT EXPERIMENTS
EXPS: afb1k66f afb3k66f afbexp20 afbexp21 afbexp22 afbexp24
RESULTS ARE AVERAGED OVER 10 DIFFERENT SPEECH DATA FILES
FILENAMES: f1 f2 f3 f4 f5 f6 f7 f8 f9 f10

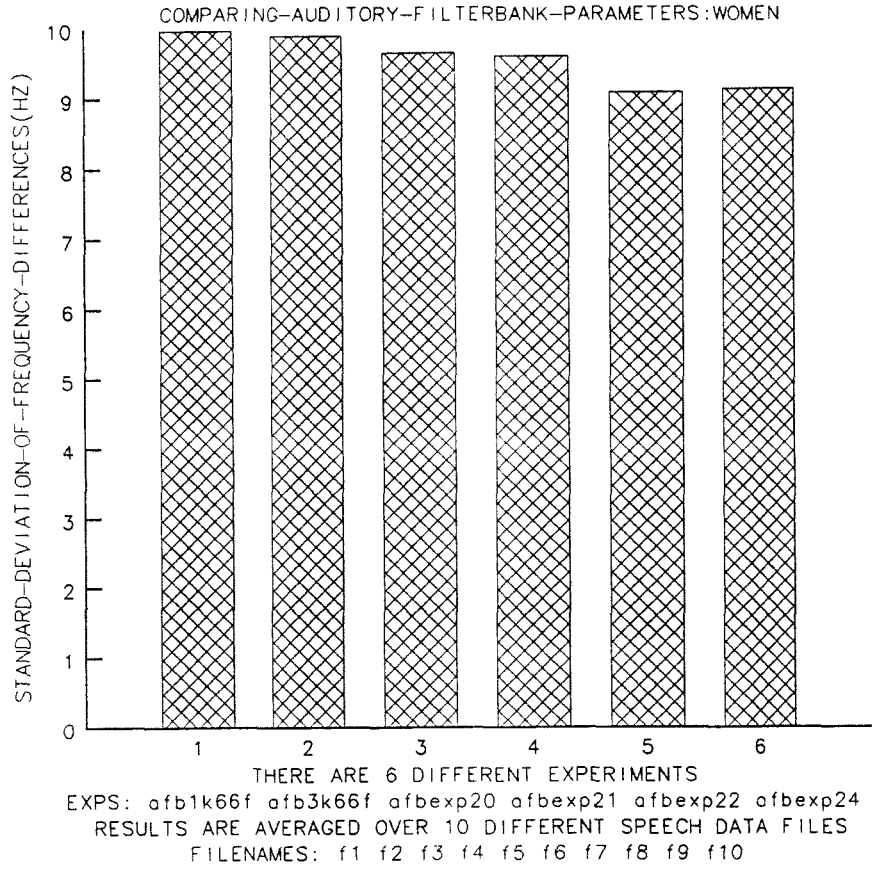

COMPARING-AUDITORY-FILTERBANK-PARAMETERS:MEN
DROP-ERRORS(%)
0
2
4
6
8
10
12
14
16
18
20
22
24
1
2
3
4
5
6
THERE ARE 6 DIFFERENT EXPERIMENTS
EXPS: afb1k66f afb3k66f afbexp20 afbexp21 afbexp22 afbexp24
RESULTS ARE AVERAGED OVER 10 DIFFERENT SPEECH DATA FILES
FILENAMES: m1 m2 m3 m4 m5 m6 m7 m8 m9 m10

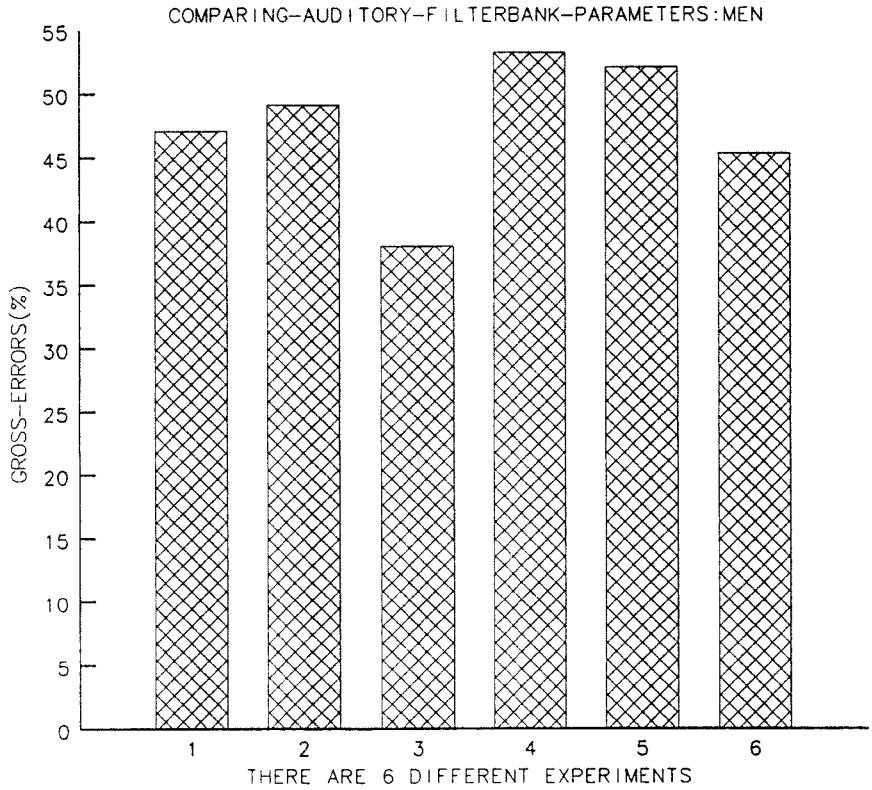

COMPARING-AUDITORY-FILTERBANK-PARAMETERS:WOMEN
DROP-ERRORS(%)
0
2
4
6
8
10
12
14
16
18
20
22
24
26
1
2
3
4
5
6
THERE ARE 6 DIFFERENT EXPERIMENTS
EXPS: afb1k66f afb3k66f afbexp20 afbexp21 afbexp22 afbexp24
RESULTS ARE AVERAGED OVER 10 DIFFERENT SPEECH DATA FILES
FILENAMES: f1 f2 f3 f4 f5 f6 f7 f8 f9 f10

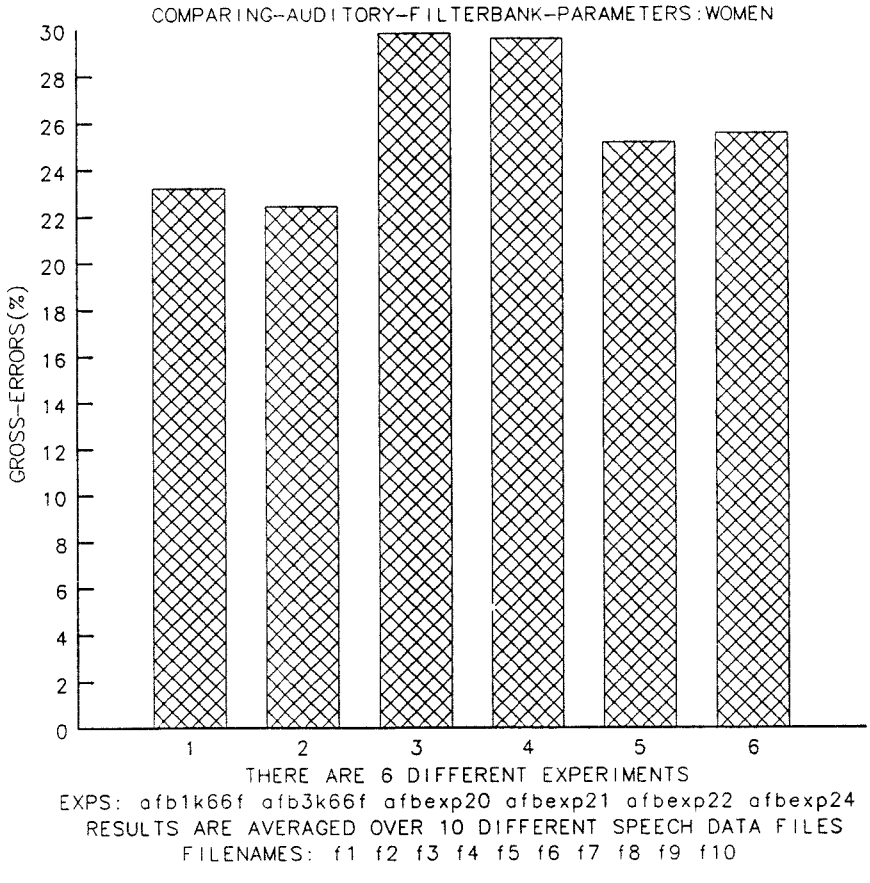

COMPARING-AUDITORY-FILTERBANK-PARAMETERS:MEN
CHIRP-ERRORS(%)
0
5
10
15
20
25
30
35
40
1
2
3
4
5
6
THERE ARE 6 DIFFERENT EXPERIMENTS
EXPS: afb1k66f afb3k66f afbexp20 afbexp21 afbexp22 afbexp24
RESULTS ARE AVERAGED OVER 10 DIFFERENT SPEECH DATA FILES
FILENAMES: m1 m2 m3 m4 m5 m6 m7 m8 m9 m10

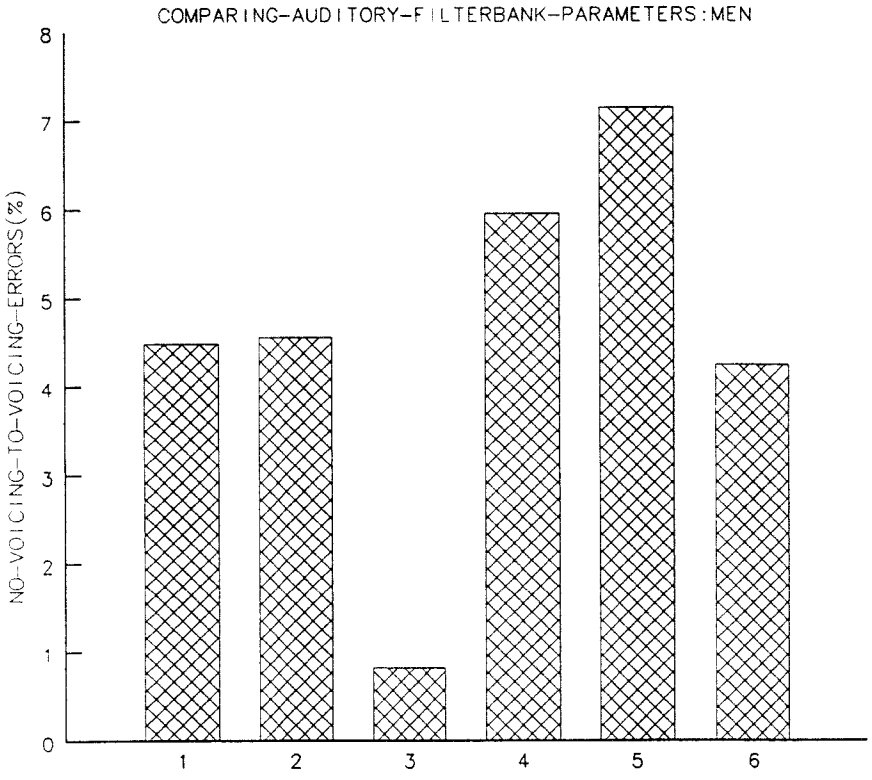

COMPARING-AUDITORY-FILTERBANK-PARAMETERS:WOMEN
CHIRP-ERRORS(%)
0
1
2
3
4
5
6
7
8
9
10
11
12
1
2
3
4
5
6
THERE ARE 6 DIFFERENT EXPERIMENTS
EXPS: afb1k66f afb3k66f afbexp20 afbexp21 afbexp22 afbexp24
RESULTS ARE AVERAGED OVER 10 DIFFERENT SPEECH DATA FILES
FILENAMES: f1 f2 f3 f4 f5 f6 f7 f8 f9 f10

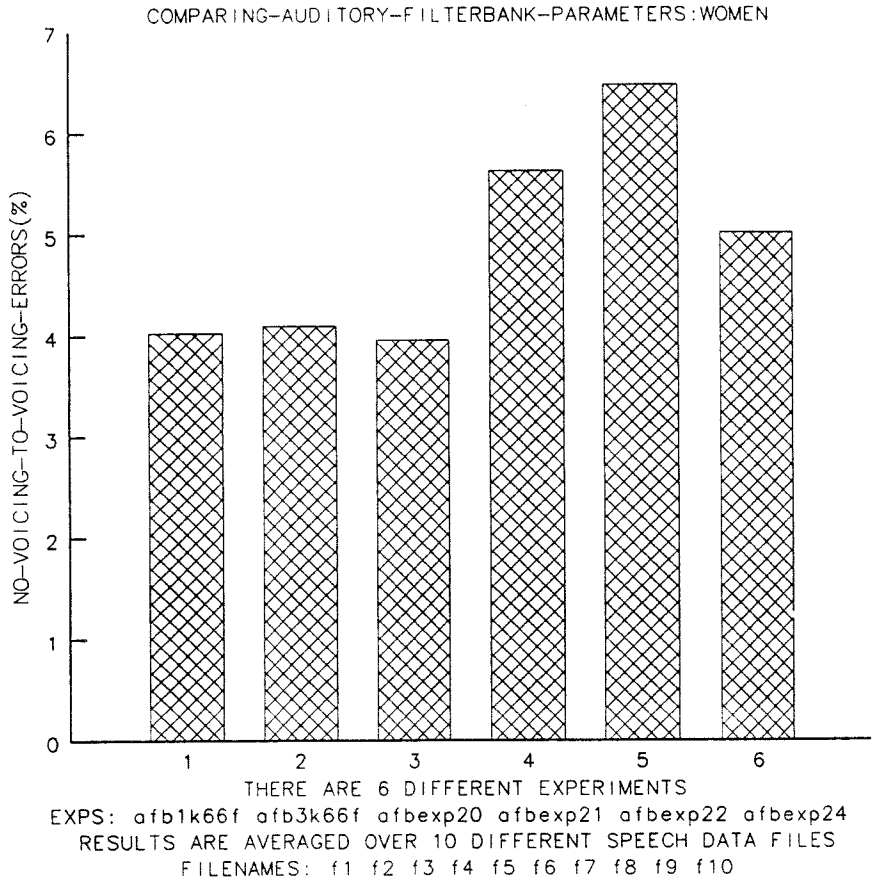

COMPARING-AUDITORY-FILTERBANK-PARAMETERS:MEN
STANDARD-DEVIATION-OF-FREQUENCY-DIFFERENCES(HZ)
0
1
2
3
4
5
6
1
2
3
4
5
6
THERE ARE 6 DIFFERENT EXPERIMENTS
EXPS: afb1k66f afb3k66f afbexp20 afbexp21 afbexp22 afbexp24
RESULTS ARE AVERAGED OVER 10 DIFFERENT SPEECH DATA FILES
FILENAMES: m1 m2 m3 m4 m5 m6 m7 m8 m9 m10


COMPARING-AUDITORY-FILTERBANK-PARAMETERS:WOMEN
STANDARD-DEVIATION-OF-FREQUENCY-DIFFERENCES(HZ)
0
1
2
3
4
5
6
7
8
9
10
1
2
3
4
5
6
THERE ARE 6 DIFFERENT EXPERIMENTS
EXPS: afb1k66f afb3k66f afbexp20 afbexp21 afbexp22 afbexp24
RESULTS ARE AVERAGED OVER 10 DIFFERENT SPEECH DATA FILES
FILENAMES: f1 f2 f3 f4 f5 f6 f7 f8 f9 f10


COMPARING-AUDITORY-FILTERBANK-PARAMETERS:MEN
GROSS-ERRORS(%)
55
50
45
40
35
30
25
20
15
10
5
0
1
2
3
4
5
6
THERE ARE 6 DIFFERENT EXPERIMENTS
EXPS: afb1k66f afb3k66f afbexp20 afbexp21 afbexp22 afbexp24
RESULTS ARE AVERAGED OVER 10 DIFFERENT SPEECH DATA FILES
FILENAMES: m1 m2 m3 m4 m5 m6 m7 m8 m9 m10


COMPARING-AUDITORY-FILTERBANK-PARAMETERS:WOMEN
GROSS-ERRORS(%)
30
28
26
24
22
20
18
16
14
12
10
8
6
4
2
0
1
2
3
4
5
6
THERE ARE 6 DIFFERENT EXPERIMENTS
EXPS: afb1k66f afb3k66f afbexp20 afbexp21 afbexp22 afbexp24
RESULTS ARE AVERAGED OVER 10 DIFFERENT SPEECH DATA FILES
FILENAMES: f1 f2 f3 f4 f5 f6 f7 f8 f9 f10


COMPARING-AUDITORY-FILTERBANK-PARAMETERS:MEN
NO-VOICING-TO-VOICING-ERRORS(%)
0
1
2
3
4
5
6
7
8
1
2
3
4
5
6
THERE ARE 6 DIFFERENT EXPERIMENTS
EXPS: afb1k66f afb3k66f afbexp20 afbexp21 afbexp22 afbexp24
RESULTS ARE AVERAGED OVER 10 DIFFERENT SPEECH DATA FILES
FILENAMES: m1 m2 m3 m4 m5 m6 m7 m8 m9 m10


COMPARING-AUDITORY-FILTERBANK-PARAMETERS:WOMEN
NO-VOICING-TO-VOICING-ERRORS(%)
0
1
2
3
4
5
6
7
1
2
3
4
5
6
THERE ARE 6 DIFFERENT EXPERIMENTS
EXPS: afb1k66f afb3k66f afbexp20 afbexp21 afbexp22 afbexp24
RESULTS ARE AVERAGED OVER 10 DIFFERENT SPEECH DATA FILES
FILENAMES: f1 f2 f3 f4 f5 f6 f7 f8 f9 f10