NON-UNIFORM TIME-SCALE MODIFICATION OF SPEECH

by

SAMUEL HOLTZMAN DANTUS

Submitted to the Department of Electrical Engineering and Computer Science, on August 8, 1980, in partial fulfillment of the requirements for the degrees of Master of Science and Electrical Engineer.

ABSTRACT

A system that performs non-uniform time-scale modification (TSM) of speech signals, based on the Discrete Short-Time Fourier Transform, is developed. This system builds on Portnoff's [1978] design, but has two major improvements: first, it allows a time-varying scale factor (as opposed to the constant scale factor required by Portnoff's system) and, second, the system requires less than 2 percent of the storage required by its predecessor.

The TSM system is used to perform feature-dependent time-scale modification of speech, in which the scale factor varies in response to changes in the local structure of the signal. Feature-dependent TSM is then compared to uniform TSM. Finally, to illustrate other potential application areas of the system, applications to long speech passages, to a compression/expansion communications scheme, and to music signals are presented.

Thesis Supervisor: Jae S. Lim

Title: Assistant Professor of Electrical Engineering and Computer Science

ACKNOWLEDGMENTS

I would like to express my sincere gratitude to Professor Jae S. Lim for his supervision of this thesis. His insightful comments, penetrating questions and uncompromising standards contributed greatly to the success of this work. I would also like to thank Dr. Michael R. Portnoff, my original thesis supervisor who provided the initial motivation of this thesis.

Grateful acknowledgment is also due to the present and past members of the Digital Signal Processing Group, who provided an excellent research environment. Special thanks go to Tom Bordley and Webster Dove who contributed many valuable comments and a wealth of useful computer programs. I would also like to express my thanks to Margaret Brandeau who carefully edited this thesis.

I am deeply indebted to my parents. Their support, both personal and financial, has been invaluable during my years at MIT.

Finally, I want to thank Ms. Joanne Klys who typed this thesis quickly and accurately.

TABLE OF CONTENTS

LIST OF FIGURES AND TABLES

Chapter 5

TABLE NO.

Chapter 2

Chapter 3

Chapter 4

Chapter 5

CHAPTER 1

INTRODUCTION

1.1 Problem Motivation

The ability to control the speed at which recorded speech is played back has a number of potential applications. On one hand, people can understand everyday speech at a rate up to three times greater than that at which it is physiologically possible to produce it. Thus, a system that increases the speed of a speech signal, without appreciably degrading its perceptual characteristics, can be used to increase the information rate of the auditory channel. On the other hand, speech recordings containing particularly complex segments, such as those in a technical or foreign language, may be made more understandable and easier to listen to by processing them with a system capable of slowing them down.

Time compression (speed increase) and time expansion (speed decrease) can also be used together as a coding/decoding scheme for speech communications. Several speech signals can be compressed in time and subsequently time-multiplexed into a channel that would otherwise be capable of carrying a single signal. The corresponding receiver would demultiplex and expand its input by the inverse of the coding rate.

For the blind, whose access to printed information is often limited to recorded speech versions, a playback speed control is likely to be particularly useful. Without such control, the rate

at which a person can listen to the speech is completely paced by the recording. A time compression system would allow the listener to increase the speed of the speech in order to scan the recording, and then to adjust it to a more comfortable level for normal listening.

Speech recognition systems could also use a time compression and expansion system to adjust the length of their input to a pre-determined value. This preprocessing step might simplify later pattern recognition algorithms by normalizing the length of the utterances to be recognized.

Processing a speech signal to obtain another that differs from the original only by its apparent rate of articulation is referred to as TIME-SCALE MODIFICATION (TSM) of the signal. The TSM scale factor is defined to be the ratio of the length of the input signal to the length of the output signal. Thus a TSM scale factor less than unity corresponds to expansion, and a scale factor greater than unity corresponds to compression.

1.2 Historical Background

From the standpoint of signal processing, TSM of speech signals is a complex problem. A simple change in the speed at which the tape travels in front of the playback head of a tape recorder or, similarly, in the output sampling rate of a digitally coded speech signal, leads to very significant degradation of the speech structure, even when the change is relatively small. This degradation is caused not only by a change in the pitch of the voiced

portions of the speech but, more importantly, by a change in the location on the frequency spectrum of the vocal tract resonances (formants) of the speech.

The problem of time-scale modification of speech signals has received considerable attention in the past. Most algorithms designed for this purpose have been based on the Fairbanks method [Fairbanks, et al., 1954, 1959; Lee, 1972].

The Fairbanks method performs the time-scale modification by automatically splicing the signal in time. It periodically repeats and discards sections of the speech which are chosen to have a length between that of a pitch period and that of a phoneme (both estimated a priori). This technique introduces significant perceptual degradation of the speech because the end of a section almost never continues smoothly into the beginning of the next.

The Fairbanks TSM technique has been refined in several ways. By introducing a pitch detector, the sections of speech that are repeated and discarded during voiced segments correspond more closely to the actual pitch periods than with a priori estimation [Scott and Gerber, 1972; Huggins, 1974]. Although the performance of the pitch-synchronous implementation is an improvement over Fairbanks' pitch-independent system, pitch detection errors introduce objectionable distortion, particularly when the speech is corrupted by noise. Even with no pitch errors, section boundary discontinuities will not be completely eliminated. A pseudo-pitch-synchronous implementation has been tried by Neuburg [1977]. He uses average pitch period section lengths and a smoothing algorithm at the

section boundaries. This system is more robust than the pitch-synchronous one and it produces an output of higher quality.

An alternative to the Fairbanks approach is to use classical vocoder techniques to construct an analysis/synthesis system to obtain a representation of speech as a set of time-varying parameters. Time-scale modification can then be implemented by time-scaling the variations in the parameter values. However, since most vocoders are designed for bandwidth compression, the quality of their output is in general too low for use in a TSM system. One notable exception is the Phase Vocoder, introduced by Flanagan and Golden [1966]. This vocoding scheme is an efficient implementation of a Discrete Short-Time Fourier Transform analysis/synthesis system [Flanagan and Golden, 1966; Schafer and Rabiner, 1973(a); Portnoff, 1976, 1977, 1978].

Portnoff [1978] has developed a time-scale modification (TSM) system based on the Phase Vocoder. Portnoff's method changes the rate of recorded speech with significantly less degradation than TSM systems that operate in the time domain. It can compress speech intelligibly to about a third of its original length, and can expand the speech by about six times without significant degradation. In addition, since it does not require a pitch detector, the system behaves well when the speech is corrupted by additive white noise.

1.3 The Scope of This Thesis

The TSM system designed by Portnoff [1978] transforms the apparent articulation rate of speech signals with very little degradation. In addition, Portnoff develops the theory of time-scale modification (TSM) of speech signals in detail, and presents a computationally efficient TSM algorithm.

Nevertheless, it is desirable to further improve Portnoff's algorithm to make it applicable to a wider class of practical problems. Specifically, Portnoff's system assumes a constant modification rate, but this may not be a desirable characteristic of a TSM system. As was pointed out in Section 1.1, many applications of time-scale modification make it imperative for the TSM rate to be allowed to vary at runtime. Such is the case, for example, in a system that lets the listener scan a section of a speech recording rapidly, and then lets him readjust the listening rate to a more comfortable level.

Besides increasing the practical applicability of the TSM system, a variable modification rate will allow us to experiment with feature-dependent time-scale modification. When TSM is performed at a uniform rate, degradation sometimes becomes more severe during transition periods in the speech signal (such as stop consonants and intervowel glides). This happens because one needs to assume, for uniform TSM, that speech is always locally stationary (i.e., quasi-stationary), an assumption often violated during speech transitions. As Toong [1974] showed in the case of compression, allowing the modification rate to become milder (closer to

unity) during the transition periods reduces the overall degradation of the speech signal.

A second reason for reformulating Portnoff's system is that while Portnoff's algorithm is computationally efficient, it uses very large amounts of storage which make it difficult or even impossible to implement in a small or medium sized computer.

The first part of this thesis develops an alternative algorithm to the one that Portnoff used. This new algorithm requires less than one-fiftieth of the main memory storage needed by its predecessor, and has no peripheral memory storage needs (except for the input and output signals themselves) which Portnoff used extensively. In addition, the proposed algorithm eliminates certain computational steps of Portnoff's algorithm that were experimentally shown to be unnecessary. Moreover, the TSM system developed here allows a variable modification rate.

The usefulness of the non-uniform TSM system described in the first part of this thesis is evaluated in the second part. First, a feature-dependent TSM system is developed, in which the variable TSM rate is adjusted to accommodate the local structure of the speech. It is found that, in general, feature-dependent TSM is not necessarily an improvement over feature-independent (uniform) TSM. Second, an assortment of other applications of non-uniform TSM are informally evaluated.

In summary, this thesis deals with the design, implementation and evaluation of an efficient non-uniform time-scale modification system based on Portnoff's [1978] design. The resulting TSM

system is used to examine the advantages of feature-dependent TSM. Finally, to give an indication of the overall usefulness of TSM, the feasibility of several desirable applications of the TSM system is informally evaluated.

1.4 Thesis Overview

The remainder of this thesis is divided into two parts.

Part I, which consists of Chapters 2, 3 and 4, presents the reformulation of Portnoff's system. First, in Chapter 2, mathematical models of both the input speech signal and of the desired output signal are described. Of particular importance is the introduction of the concept of time-unwrapped phase. This concept is used later as an essential part of the time-scale modification algorithm.

The Short-Time Fourier Transform (STFT) is defined in Chapter 3. The classical discrete-time, continuous-frequency transform, with infinite summation limits, is introduced. It is then modified to a discrete-time, discrete-frequency form with finite summation limits, known as the Discrete Short-Time Fourier Transform (DSTFT). Algorithms to compute the DSTFT (Analysis) and its inverse (Synthesis) are described. While these algorithms are based on Portnoff's design, they are shown to be more efficient in terms both of the number of computations required and of the amount of storage needed.

Chapter 4 describes techniques used for time-scale modification of speech. The uniform-rate modification scheme developed by Portnoff [1978] is presented, and then extended to allow the rate to be varied at runtime.

Part II, which consists of Chapters 5, 6 and 7, presents and evaluates several possible uses of the TSM system developed in Part I.

Chapter 5 describes feature-dependent time-scale modification of speech. Three speech segmentation algorithms are introduced that can be used to control the TSM rate in response to speech features. These algorithms are based on a set of statistical measures of the signal designed to estimate its local level of quasi-stationarity. The resulting feature-dependent TSM system is then evaluated.

Chapter 6 presents the results of several potential applications of the non-uniform TSM system, to give an indication of its usefulness. Two specific application examples that have been considered are the compression/expansion communications scheme suggested in Section 1.1, and application of the TSM system to music signals.

Finally, Chapter 7 concludes the thesis with a summary of the major results obtained, followed by a set of suggestions for further research.

PART I

A TSM SYSTEM

CHAPTER 2

A MODEL OF TIME-SCALE MODIFIED SPEECH

In order to design a time-scale modification system, we must fully understand what such a system is to accomplish. In Section 1.1, TSM was defined as the process which takes a speech signal and generates a speech-like signal which is perceptually identical to the original, except for a change in its apparent rate of articulation. This chapter examines this definition in detail. In Section 2.1, a parametric model of normal speech is developed and the concept of a time-unwrapped phase, which is used extensively in later parts of the thesis, is introduced. Based on our analysis of normal speech, Section 2.2 describes the desired time-scale modified speech signal. Finally, having understood what the input and the desired output of a TSM system are, Section 2.3 outlines the major processing steps involved in time-scale modification.

2.1 A Model of Normal Speech

The standard engineering model of speech production is shown in figure 2.1. According to this model, speech signals are generated as the convolution of a time-varying linear system (modeling the vocal tract) with an excitation signal which can be either a quasi-periodic train of pulses (representing vocal fold excitation) or white noise (representing whisper, fricative or any other noise-like excitation).

The Standard Engineering Model of Speech Production

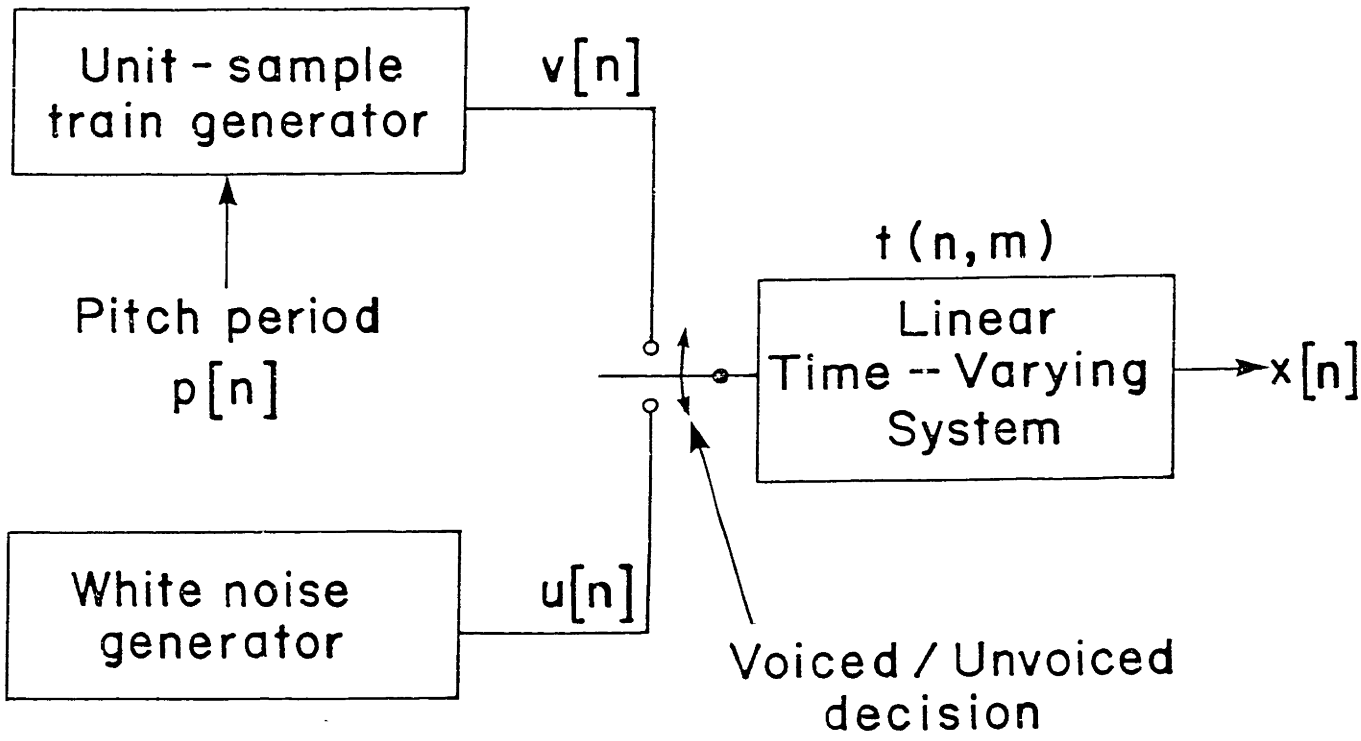


Figure 2.1

When the speech excitation can be represented as a train of pulses, we refer to the speech as Voiced. When it can be represented by noise, the speech is referred to as Unvoiced. In this chapter, the analysis of the speech signal is quite different in these two cases. Consequently, each case will be treated in a separate subsection.

2.1.1 Normal Voiced Speech

Let v[n] denote a quasi-periodic train of unit samples. By quasi-periodic we mean that the number of zero-valued samples lying between any two consecutive unit-valued samples of v[n] is approximately constant in the vicinity of any fixed point $n = n_0$. We will consider v[n] to be the excitation of the linear time-varying system during voiced speech segments.

By representing the local behavior of v[n] as a sum of harmonically related complex exponentials, Portnoff [1978] has shown that, during voiced segments, the speech signal x[n] can itself be represented as a linear combination of harmonically related exponentials.

To describe the behavior of v[n], let us define the following parameters:

Let p[n] denote the number of samples that separate the two consecutive unit samples surrounding the point n or, if n corresponds to a unit sample, the number of samples between the previous unit sample and n (figure 2.2). The quantity p[n] is commonly referred to as the local pitch period of v[n] and, consequently, of x[n].

The Voiced Excitation Sequence v[n]

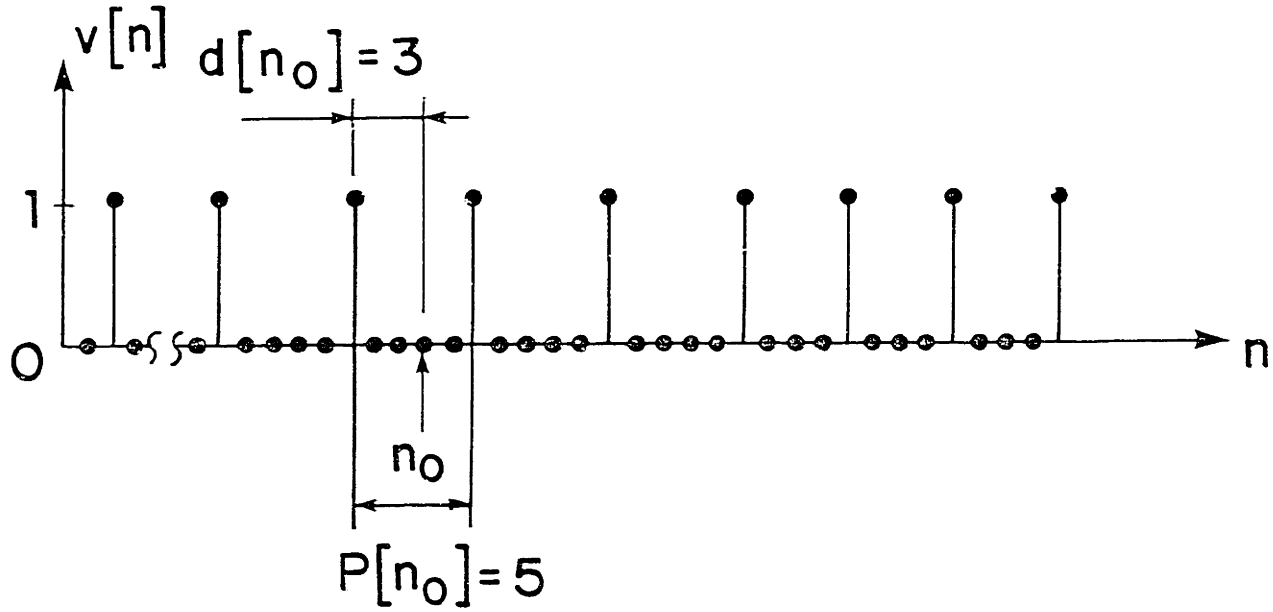


Figure 2.2

Let d[n] denote the number of samples that separate the point n from the latest unit sample in the sequence v[n]. Clearly, d[n] can only take values between zero and p[n] - 1, inclusively (figure 2.2).

In the vicinity of $n = n_0$, v[n] can be described locally as follows:

$$v[n_0+m] = \sum_{r=-\infty}^{+\infty} \delta[m-d[n_0]-rp[n_0]] \tag{2.1}$$

where m is assumed to take only small values and $\delta[n]$ is the unit sample function of n.

The periodic impulse sequence given by equation (2.1) can be expressed as a sum of harmonically related exponentials. In this form, equation (2.1) becomes:

$$v[n_0+m] = \frac{1}{p[n_0]} \sum_{k=0}^{p[n_0]-1} e^{j2\pi k(m+d[n_0]/p[n_0]}$$

$$= \frac{1}{p[n_0]} \sum_{k=0}^{p[n_0]-1} e^{jk(\phi[n_0]+m\Omega[n_0])} \tag{2.2}$$

where $$\Omega[n_0] = 2\pi/p[n_0] \tag{2.3}$$

$$\phi[n_0] = \Omega[n_0]d[n_0] + 2\pi I[n_0] \tag{2.4}$$

and $I[n_0]$ is the integer number of unit samples of v[n] that precede the point $n = n_0$.

The quantity $\Omega[n_0]$ is referred to as the local pitch frequency of v[n], and thus of the speech signal x[n], at $n = n_0$. The quantity $\phi[n_0]$ is the time-unwrapped phase value of the fundamental of v[n] at sample n.

The concept of time-unwrapped phase is novel and, therefore, requires some clarification. The main results concerning time-unwrapped phase are presented in equations (2.5) - (2.10).

Consider a signal $v_0[n]$ defined as the fundamental of v[n]. In general, the signal $v_0[n]$ will be quasi-sinusoidal in the sense that, in the vicinity of any point $n = n_0$, $v_0[n]$ will look like a sinusoid of frequency $\Omega[n_0]$ but, over time, this local frequency will not be constant. For small values of m, we can write $v_0[n]$ as:

$$v_0[n_0+m] = V \cdot \sin(\Omega[n_0]m+\phi[n_0]) \tag{2.5}$$

where V is the amplitude of the sinusoid.

The argument of the sine function in equation (2.5) is referred to as the time-unwrapped phase, $\phi[n_0+m]$, of $v_0[n_0+m]$.

$$\phi[n_0+m] = \Omega[n_0]m + \phi[n_0] \tag{2.6}$$

The name of $\phi[n_0+m]$ is motivated by the fact that if m is an integer multiple of the local pitch period $p[n_0]$, then $\phi[n_0+m]$ will be equal to $\phi[n_0]$ plus the same multiple of 2π.

In other words, every time v[n] completes a period, its time-unwrapped phase increases by 2π. To see this, replace m by $kp[n_0]$ in equation (2.6), for some integer k:

$$\begin{aligned}\phi[n_0+kp[n_0]] &= \Omega[n_0]kp[n_0] + \phi[n_0] \\ &= (2\pi/p[n_0])kp[n_0] + \phi[n_0] \\ &= 2\pi k + \phi[n_0]\end{aligned} \tag{2.7}$$

The term $2\pi I[n_0]$ appears in equation (2.4) to account for this fact.

An interesting property of the time-unwrapped phase is that its growth cannot be arbitrary. In fact, it is bounded by a constant.

This property can be derived by evaluating equation (2.6) for m = 1. Setting m = 1 is consistent with the definition of equation (2.6) since it assumes that m is small, and m = 1 is the smallest possible value of m. Equation (2.6) then becomes:

$$\phi[n_0+1] = \Omega[n_0] + \phi[n_0] \tag{2.8}$$

From the definition of p[n] it is evident that its value must be greater than or equal to unity. Therefore, equation (2.3) implies that the range of $\Omega[n]$ is:

$$0 < \Omega[n] \leq 2\pi \quad , \quad \text{for all integers } n \tag{2.9}$$

Combining equations (2.8) and (2.9) we can put a bound on the growth of $\phi[n]$:

$$0 < \phi[n+1] - \phi[n] \leq 2\pi \tag{2.10}$$

Equation (2.10) shows the motivation for referring to $\phi[n]$ as an unwrapped phase rather than as a principal value phase.

Two important issues can be raised regarding time-unwrapped phase. First, the constraint imposed by equation (2.10) applies to the difference between consecutive values of the sequence $\phi[n]$, and not to $\phi[n]$ itself, which can take arbitrary real values. Second, we can compare the concept of time-unwrapped phase with the more familiar one of frequency-unwrapped phase. In the latter case, the underlying assumption is that the phase curve is continuous along the frequency axis. No bound, such as the one imposed by equation (2.10), exists on the growth of the frequency-unwrapped phase. This causes any algorithm for the estimation of the unwrapped phase curve from its principal value curve to be rather cumbersome [Tribolet, 1977; Quatrieri, 1979].

To return to our derivation of a model of voiced speech, consider the local voiced excitation sequence $v[n_0+m]$ described by equations (2.2) - (2.4). The sequence can be expressed in terms of its time-unwrapped phase by direct replacement of equation (2.6) in equation (2.2):

$$v[n_0+m] = \frac{1}{p[n_0]} \sum_{k=0}^{p[n_0]-1} e^{jk\phi[n_0+m]} \tag{2.11}$$

Equation (2.11) describes the local behavior of the voiced excitation v[n]. To obtain a description of the global behavior of v[n], we can use equation (2.8) to redefine $\phi[n]$ recursively. Then, we can replace n_0 by n in equation (2.11) and, setting m equal to zero, we obtain:

$$v[n] = \frac{1}{p[n]} \sum_{k=0}^{p[n]-1} e^{jk\phi[n]} \tag{2.12}$$

where:

$$\phi[n] = \begin{cases} \phi[n+1] - \Omega[n] & , \quad n < 0 \\ 2\pi d[0]/p[0] & , \quad n = 0 \\ \phi[n-1] + \Omega[n-1] & , \quad n > 0 \end{cases} \tag{2.13}$$

During voiced segments, the speech signal is the convolution of the voiced excitation v[n] with the time-varying linear system that represents the behavior of the vocal tract. Let the doubly indexed sequence t[n,m] be the time-varying unit sample response of this linear system. Specifically, the sequence t[n,m] corresponds to the response of the system at time n to a unit sample which occurred m samples earlier.

The signal x[n], during voiced segments, can therefore be written as the time-varying convolution of v[n] with t[n,m]

along the index m:

$$x[n] = \sum_{\ell=-\infty}^{+\infty} t[n,\ell]v[n-\ell]$$

$$= \sum_{\ell=-\infty}^{+\infty} \left\{ t[n,\ell] \cdot \frac{1}{p[n-\ell]} \sum_{k=0}^{p[n-\ell]-1} e^{jk\phi[n-\ell]} \right\} \qquad (2.14)$$

It is commonly accepted in speech analysis that changes in the pitch period, p[n], occur slowly enough to assume that the sequence p[n] is constant for the duration in m of the sequence t[n,m]. By making this assumption, we can replace p[n-ℓ] by p[n] to simplify equation (2.14):

$$x[n] = \frac{1}{p[n]} \sum_{\ell=-\infty}^{+\infty} \left\{ t[n,\ell] \cdot \sum_{k=0}^{p[n]-1} e^{jk\phi[n-\ell]} \right\} \qquad (2.15)$$

Interchanging the order of summation and regrouping terms:

$$x[n] = \frac{1}{p[n]} \sum_{k=0}^{p[n]-1} \sum_{\ell=-\infty}^{+\infty} t[n,\ell]e^{jk\phi[n-\ell]} \qquad (2.16)$$

Our assumption that p[n] is constant for the duration in m of t[n,m] implies the same for Ω[n]. Therefore we can replace φ[n] in equation (2.16) by its local representation given in equation (2.6):

$$x[n] = \frac{1}{p[n]} \sum_{k=0}^{p[n]-1} \sum_{\ell=-\infty}^{+\infty} t[n,\ell] e^{jk(\phi[n]-\ell\Omega[n])}$$

$$= \frac{1}{p[n]} \sum_{k=0}^{p[n]-1} e^{jk\phi[n]} \left\{ \sum_{\ell=-\infty}^{+\infty} t[n,\ell] e^{-jk\ell\Omega[n]} \right\} \qquad (2.17)$$

The term in brackets in equation (2.17) can be interpreted as the second partial Fourier transform of the sequence $t[n,m]$. If we fix $n = n_0$, the Fourier transform of the one-dimensional sequence $t[n_0,m]$ is:

$$T[n_0,\omega] = \sum_{\ell=-\infty}^{+\infty} t[n_0,\ell] e^{-j\omega\ell} \qquad (2.18)$$

Equation (2.17) can then be rewritten in terms of $T[n,\omega]$:

$$x[n] = \frac{1}{p[n]} \sum_{k=0}^{p[n]-1} T[n,k\Omega[n]] e^{jk\phi[n]}$$

$$= \sum_{k=0}^{p[n]-1} \left(\frac{T[n,k\Omega[n]]}{p[n]} \right) e^{jk\phi[n]} \qquad (2.19)$$

Another assumption commonly made in the study of speech signals is that the sequence $t[n,m]$ varies much more slowly in n than in m. This is because the variation of $t[n,m]$ along its index m corresponds to the instantaneous unit sample response of the vocal tract (at a given time n), while the variation of $t[n,m]$

along its index n reflects the much slower changes in time of the acoustic characteristics of the vocal tract.

We can therefore consider the term in parentheses in equation (2.19) as a slowly varying lumped parameter $c_k[n]$. The desired model of normal voiced speech can then be obtained from equation (2.19):

$$x[n] = \sum_{k=0}^{p[n]-1} c_k[n] e^{jk\phi[n]} \tag{2.20}$$

$$\text{where} \quad c_k[n] = \frac{T[n, k\Omega[n]]}{p[n]} \tag{2.21}$$

The parameters $c_k[n]$ represent the slowly varying characteristics of the vocal tract and of the pitch period p[n]. They are referred to as the "complex harmonic amplitudes" of the speech and, seen as sequences in n, they contain non-negligible frequency components only up to the range of tens of Hertz [Portnoff, 1978]. Thus, their bandwidths are very small compared to the lowest frequency components of normal speech, a fact that will be very useful in Chapter 4, where the TSM system is presented.

2.1.2 Normal Unvoiced Speech

Let u[n] denote a zero-mean stationary white noise process. The sequence u[n] will be taken as the excitation of the linear time-varying system t[n,m] during the unvoiced speech segments. As is customary in speech processing applications, we shall limit

the description of u[n] to its second order moment characteristics. Thus, u[n] will be assumed to be sufficiently specified by its mean and autocorrelation sequence. The unvoiced speech sequence x[n] is generated as the convolution of u[n] with a linear system. Consequently, x[n] can also be specified by its mean and autocorrelation sequence.

We have assumed that u[n] is a zero-mean random process. Since t[n,m] is a linear system, the mean of x[n] is also zero.

Now, let $R_u[n]$ denote the autocorrelation sequence of u[n]. The fact that u[n] is a white noise process implies that, for some positive real value of its variance σ_u^2, the sequence $R_u[n]$ is:

$$R_u[n] = E\left[u[m]\ u^*[n+m]\right]$$

$$= \sigma_u^2 \cdot \delta[n] \qquad (2.22)$$

where "*" denotes complex conjugation.

The time-varying autocorrelation function, $R_x[n,m]$, of the speech signal x[n] can be obtained as the autocorrelation of the convolution of u[n] with t[n,m] along the index m:

$$R_x[n,m] = E\left[x[n]\ x^*[n+m]\right]$$

$$= E\left[\left\{\sum_{p=-\infty}^{+\infty} t[n,p]u[n-p]\right\}\left\{\sum_{q=-\infty}^{+\infty} t[n+m,q]u[n+m-q]\right\}^*\right]$$

$$= \sum_{p=-\infty}^{+\infty}\sum_{q=-\infty}^{+\infty} t[n,p]t^*[n+m,q]E\left[u[n-p]u^*[n+m-q]\right]$$

$$= \sum_{p=-\infty}^{+\infty}\sum_{q=-\infty}^{+\infty} t[n,p]t^*[n+m,q]R_u[p+m-q]$$

$$= \sum_{p=-\infty}^{+\infty}\sum_{q=-\infty}^{+\infty} t[n,p]t^*[n+m,q]\sigma_u^2\delta[p+m-q]$$

Therefore:

$$R_x[n,m] = \sigma_u^2 \sum_{p=-\infty}^{+\infty} t[n,p]t^*[n+m,p+m] \qquad (2.23)$$

Since we have assumed that $t[n,m]$ varies much more rapidly in m than in n, we can replace the term $t[n+m,p+m]$ in equation (2.23) by $t[n,p+m]$:

$$R_x[n,m] = \sigma_u^2 \sum_{p+-\infty}^{+\infty} t[n,p]t^*[n,p+m] \qquad (2.24)$$

Equation (2.24) expresses the time-varying autocorrelation sequence of x[n], $R_x[n,m]$, in terms of t[n,m] and the variance σ_u^2 of its unvoiced excitation. This equation, together with the fact that x[n] has zero mean, constitutes the desired second order model of unvoiced speech.

To understand the process of time-scale modification of unvoiced speech, however, we must carry our analysis somewhat further. We may recognize the summation in equation (2.24) as the deterministic autocorrelation of the sequence t[n,m] along the index m, with the index n being held constant. It can be easily shown [Bloomfield, 1976] that this summation has a Fourier transform, $T_n[\omega]$, which equals the square of the Fourier transform $T[n,\omega]$ of t[n,m] along m, with n held fixed, as defined by equation (2.18). That is:

$$T_n[\omega] = |T[n,\omega]|^2 \tag{2.25}$$

$$\text{where} \qquad T[n,\omega] = \sum_{m=-\infty}^{+\infty} t[n,m]e^{-j\omega m} \tag{2.26}$$

Equation (2.24) can then be expressed in terms of $T[n,\omega]$ as follows:

$$R_x[n,m] = \sigma_u^2 \mathcal{F}^{-1}\{T_n[\omega]\}$$

$$= \sigma_u^2 \frac{1}{2\pi} \int_{-\pi}^{\pi} T_n[\omega] e^{j\omega m} d\omega$$

$$= \sigma_u^2 \frac{1}{2\pi} \int_{-\pi}^{\pi} |T[n,\omega]|^2 e^{j\omega m} d\omega \qquad (2.27)$$

Let us now define the time-varying power spectrum $S_x[n,\omega]$ of the unvoiced speech sequence $x[n]$ as:

$$S_x[n,\omega] = \sigma_u^2 |T[n,\omega]|^2 \qquad (2.28)$$

The functions $R_x[n,\omega]$ and $S_x[n,\omega]$ thus constitute a Fourier transform pair, for n held constant:

$$R_x[n,m] = \frac{1}{2\pi} \int_{-\pi}^{\pi} S_x[n,\omega] e^{j\omega m} d\omega \qquad (2.29)$$

$$S_x[n,\omega] = \sum_{m=-\infty}^{+\infty} R_x[n,m] e^{-j\omega m} \qquad (2.30)$$

Since we have assumed that $t[n,m]$ has a slow variation in n with respect to its length in m, we shall refer to the unvoiced speech signal $x[n]$ as a quasi-stationary random process.

The desired model of normal unvoiced speech can now be formed by grouping together equations (2.24), (2.29) and (2.30)

with the added knowledge that the mean of x[n] is zero.

2.2 A Model of the Desired Time-Scale Modified Speech

As before, we will deal separately with voiced and unvoiced speech.

2.2.1 Time-Scale Modified Voiced Speech

From our discussion in Section 2.1, we know that the voiced speech signal x[n] can be represented as a set of time-varying parameters. This parametric model is in the form of a sum of harmonically related complex exponentials, as expressed by equation (2.20), which we repeat here for convenience:

$$x[n] = \sum_{k=0}^{p[n]-1} c_k[n] e^{jk\phi[n]} \tag{2.31}$$

The parameters p[n], $c_k[n]$ and $\phi[n]$ have meaningful interpretations in terms of speech features. The sequence p[n] is the local pitch period, as described in figure 2.2. The set of sequences $c_k[n]$ are a lumped parameter description of the local behavior of the time-varying linear system which represents the vocal tract, normalized by 1/p[n]. This can be seen in equation (2.21). Finally, the term $\phi[n]$ is the time-unwrapped phase of the fundamental of v[n], and is defined by equation (2.13).

We wish to determine the corresponding parametric description of a sequence $y[n] = x^{\beta}[n]$ which is perceptually identical to x[n] except for the fact that it appears to have been articulated

β times faster than the original.

To obtain this parametric description, it is useful to keep in mind that the sequence $x[n]$ is obtained by sampling a continuous-time signal $x(t)$ such that:

$$x[n] = x(nT) \tag{2.32}$$

where T is a constant sampling interval.

We shall distinguish between discrete-time and continuous-time signals by placing the arguments of discrete-time signals in brackets and the arguments of continuous-time signals in parentheses, as in equation (2.32).

The sequences $p[n]$, $c_k[n]$ and $\phi[n]$ can similarly be interpreted as sampled versions of continuous-time signals. It is therefore meaningful to define the sequences $x[\beta n]$, $p[\beta n]$, $c_k[\beta n]$ and $\phi[\beta n]$, for any real number β as follows:

$$x[\beta n] = x(\beta nT) \tag{2.33}$$

$$p[\beta n] = p(\beta nT) \tag{2.34}$$

$$c_k[\beta n] = c_k(\beta nT) \tag{2.35}$$

$$\phi[\beta n] = \phi(\beta nT) \tag{2.36}$$

We have implicitly assumed that the sampling interval T is small enough to guarantee that x[n] completely specifies the bandlimited continuous-time speech signal x(t), without any frequency domain aliasing. Therefore, it is possible to recover x(t) from x[n], at least in principle. If we were to actually recover the continuous-time signal from the discrete-time signal, a simple scaling of the time dimension of x(t) by β would yield $x(\beta t)$. The sequences $x[\beta n]$, $p[\beta n]$, $c_k[\beta n]$ and $\phi[\beta n]$ could then be generated as sampled versions of their continuous-time counterparts.

In practice, however, the sequences defined in equations (2.33) - (2.36) can be obtained without the need to return to continuous-time signals. In particular, when the number β is rational, Schafer and Rabiner [1973(b)] have shown that the procedure to obtain a sequence (such as $x[\beta n]$) from another that represents the same continuous-time signal with a different sampling rate (in this case x[n]), consists of a simple finite impulse response (FIR) filtering operation. Therefore, when β is rational, the sequences $x[\beta n]$, $p[\beta n]$, $c_n[\beta n]$ and $\phi[\beta n]$ can be obtained respectively from x[n], p[n], $c_k[n]$ and $\phi[n]$ by FIR filtering. Since any real number can be approximated by a rational one with arbitrary precision, we will assume for the remainder of this thesis that β is rational.

We will now proceed to show that the time-scale modified sequence $x^{\beta}[n]$ is equal to the sequence $x[\beta n]$ with its time-unwrapped phase divided by β.

In terms of the parametric representation of voiced speech given by equation (2.31), the sequence $x[\beta n]$ can be written as:

$$x[\beta n] = \sum_{k=0}^{p[\beta n]-1} c_k[\beta n] e^{jk\phi[\beta n]} \tag{2.37}$$

Although it may be intuitively appealing to say that $x[\beta n]$ is the desired time-scale modified sequence $x^{\beta}[n]$, a closer look at the structure of $x[\beta n]$ indicates that this is not the case.

For the time-scale modified sequence $x^{\beta}[n]$ to have the same pitch structure in time as the original sequence $x[n]$, its time-dependent pitch frequency $\Omega^{\beta}[n]$ must be equal to $\Omega[n]$ with its time dimension scaled by β:

$$\Omega^{\beta}[n] = \Omega[\beta n] \tag{2.38}$$

In other words, the spectral location of the pitch frequency $\Omega^{\beta}[n]$ must be the same as the spectral location of the original pitch frequency $\Omega[n]$, with its argument n replaced by βn.

Let $\tilde{\Omega}[\beta n]$ denote the pitch frequency of the sequence $x[\beta n]$. We will now show that $\tilde{\Omega}[\beta n]$ is not equal to $\Omega[\beta n]$.

To do this, we refer back to the continuous-time signals $x(t)$ and $x(\beta t)$. An approximate result can be derived with discrete-time signals [Portnoff, 1978]. Using continuous-time signals, however, we can obtain an exact result which can be directly interpreted in terms of $x[n]$ and $x[\beta n]$.

Equation (2.8) shows that $\Omega[n]$ is equal to the first forward difference of $\phi[n]$. In continuous time, this equation becomes:

$$\frac{d}{dt}\ \phi(t) = \Omega(t) \tag{2.39}$$

The functions $\phi(t)$ and $\Omega(t)$ are referred to, respectively, as the time-unwrapped phase and the instantaneous frequency of $x(t)$. For $x(\beta t)$ we can obtain a similar result:

$$\frac{d}{dt}\ \phi(\beta t) = \frac{d}{d(\beta t)}\ \phi(\beta t) \cdot \frac{d}{dt}\ \beta t$$

$$= \frac{d}{d\tau}\ \phi(\tau)\Big|_{\tau\ =\ \beta t} \cdot \beta$$

$$= \beta\Omega(\beta t) \tag{2.40}$$

By definition, the derivative of $\phi(\beta t)$ is the instantaneous frequency $\tilde{\Omega}(\beta t)$ of $x(\beta t)$, so:

$$\tilde{\Omega}(\beta t) = \beta\Omega(\beta t) \neq \Omega(\beta t) \tag{2.41}$$

This result can be directly interpreted in terms of discrete-time signals:

$$\tilde{\Omega}[\beta n] = \tilde{\Omega}(\beta nT)$$

$$= \beta \cdot \Omega(\beta nT)$$

$$= \beta \cdot \Omega[\beta n] \qquad (2.42)$$

Equation (2.42) shows that $\tilde{\Omega}[\beta n]$ does not equal $\Omega[\beta n]$. This result should not be surprising. The sequence $x[\beta n]$ is a sampled version of $x(\beta t)$ which, in turn, is exactly the signal that we would have obtained by speeding up the sampling rate of the digital-to-analog conversion of $x[n]$ to $x(t)$ by a factor of β (or, analogously, by speeding up the playback speed of an analog tape containing $x(t)$). As discussed in section 1.2, this method will not produce the desired time-scale modified signal.

To correct the pitch of $x[\beta n]$ we must divide its phase, $\phi[\beta n]$, by β. If we do this, equation (2.40) becomes:

$$\frac{d}{dt}\{\phi(\beta t)/\beta\} = \beta\Omega(\beta t)/\beta$$

$$= \Omega(\beta t) \qquad (2.43)$$

We have therefore transformed the sequence $x[\beta n]$ into the desired sequence $x^{\beta}[n]$. The expression for the time-scale modified voiced speech is thus:

$$x^{\beta}[n] = \sum_{k=0}^{p[\beta n]-1} c_k[\beta n] e^{jk\phi[\beta n]/\beta} \qquad (2.44)$$

2.2.2 Time-Scale Modified Unvoiced Speech

When x[n] is unvoiced, we have modeled it as the output of a time-varying linear system with impulse response t[n,m] which is excited by a white noise process u[n]. Thus, x[n] is a quasi-stationary random process with zero mean and time-varying second order statistics given by equations (2.24), (2.29) and (2.30). The time-scale modified sequence $x^{\beta}[n]$ is, in this case, another quasi-stationary random process with zero mean and time-varying second order statistics that are those of x[n] with their time dimension scaled by β. In terms of its second order model, the sequence $x^{\beta}[n]$, for the case of unvoiced speech, is given by the relations:

$$R_x^{\beta}[n,m] = R_x[\beta n,m] \tag{2.45}$$

or, equivalently:

$$S_x^{\beta}[n,\omega] = S_x[\beta n,\omega] \tag{2.46}$$

and:

$$E\left[x^{\beta}[n]\right] = 0 \tag{2.47}$$

2.3 Outline of a Time-Scale Modification System

In Section 2.1 we derived parametric models of normal voiced and unvoiced speech. In the case of voiced speech, it was shown that the speech signal x[n] can be interpreted as a sum of harmonically related exponentials, as given by equation (2.20):

$$x[n] = \sum_{k=0}^{p[n]-1} c_k[n] e^{jk\phi[n]} \tag{2.48}$$

where the time-varying parameters p[n], $c_k[n]$ and $\phi[n]$, respectively correspond to the pitch period, the normalized vocal tract characteristics, and the time-unwrapped phase of the speech signal.

In the case of unvoiced speech, our model of x[n] was a second order quasi-stationary stochastic characterization. Equations (2.24), (2.29) and (2.30) define the time-varying autocorrelation and power spectrum of the unvoiced speech signal:

$$R_x[n,m] = \sigma_u^2 \sum_{p=-\infty}^{+\infty} t[n,p]\, t^*[n,p+m] \tag{2.49}$$

$$R_x[n,m] = \frac{1}{2\pi} \int_{-\pi}^{\pi} S_x[n,\omega] e^{j\omega m} d\omega \tag{2.50}$$

$$S_x[n,\omega] = \sum_{m=-\infty}^{+\infty} R_x[n,m] e^{-j\omega m} \tag{2.51}$$

where the time-varying parameter $t[n,m]$ is the vocal tract impulse response, and the constant σ_u^2 is the variance of the white noise excitation.

In Section 2.2, we obtained the corresponding models of time scale modified voiced and unvoiced speech. We found that, for voiced speech segments:

$$x^{\beta}[n] = \sum_{k=0}^{p[\beta n]-1} c_k[\beta n] e^{jk\phi[\beta n]/\beta}$$

$$= x[\beta n] \text{ with its phase divided by } \beta. \qquad (2.52)$$

In the case of unvoiced speech, the time-scale modified sequence $x^{\beta}[n]$ is given by equations (2.45)-(2.47). If we assume that the unvoiced excitation $u[n]$ is a Gaussian white noise process (as opposed to a general white noise process), and if we can estimate $S_x[n,\omega]$ with enough resolution, then for the purpose of time-scale modification, the unvoiced portions of the speech may be treated as if they were voiced [Portnoff, 1978]. This means that we must divide the phase of $x[\beta n]$ during unvoiced speech segments (as we would normally do if they were voiced), to obtain the desired second order statistics.

This result is a very important one. Since voiced and unvoiced speech are to be treated in the same manner, no need exists for the TSM system to make voiced/unvoiced decisions. This

increases the robustness of the modification algorithm, particularly when the speech is corrupted by noise.

For the remainder of this thesis, the speech signal x[n] will be assumed to consist solely of voiced segments. Therefore, we shall assume that x[n] has the structure derived in Section 2.1.1 and, consequently, that $x^{\beta}[n]$ has the structure given in Section 2.2.1.

We can now see that a TSM system must perform two distinct operations on x[n]. First, it must compute x[βn] from x[n] and, second, it must estimate and modify the phase of x[βn] to yield the desired sequence $x^{\beta}[n]$.

The first step in the TSM process, which derives x[βn] from x[n] by scaling its time dimension by β, will be referred to as LINEAR TIME-SCALING. This will distinguish it from the overall TSM process which is non-linear due to the phase correction required to obtain the correct pitch contour of $x^{\beta}[n]$.

In order to perform the linear time-scaling and phase modification operations, the values of the time-varying parameters which form our model of voiced speech (equation 2.20) must be estimated. Then, the linear time-scaling and phase modification steps will yield $x^{\beta}[n]$ in parametric form. Finally, $x^{\beta}[n]$ has to be derived from its parametric description.

Thus, a complete TSM process can be divided into four steps which will be labeled, respectively: Analysis, Linear Time-Scaling, Phase Modification and Synthesis, as shown in Table I.

Table I

The TSM Process

1. Analysis

   Estimation of the time-varying parametric structure of the signal $x[n]$.

2. Linear Time-Scaling

   Computation of $x[\beta n]$ from $x[n]$.

3. Phase Modification

   Estimation and scaling of the time-unwrapped phase of $x[\beta n]$. This step will yield the parametric description of the desired signal $x^{\beta}[n]$.

4. Synthesis

   Generation of $x^{\beta}[n]$ from its parametric description.

CHAPTER 3

THE SHORT-TIME FOURIER TRANSFORM

The time-scale modification system described in this thesis is based on a mathematical technique referred to as the Short-Time Fourier Transform (STFT). In its classical definition, the STFT involves a continuous frequency variable, and a discrete time variable which is summed between infinite limits. In order to make the technique computationally viable, the DISCRETE Short-Time Fourier Transform (DSTFT) is introduced, in which the frequency variable is discrete and the summation limits are finite.

In this chapter, both the STFT and the DSTFT are described. Efficient algorithms for the computation of the DSTFT and its inverse are presented.

Most of this chapter is a reformulation of work previously done by Portnoff [1978]. The derivation presented here, however, improves on Portnoff's results in two ways. First, it eliminates several computational steps from Portnoff's algorithm that were experimentally shown to be unnecessary. Second, it introduces an alternative inverse DSTFT algorithm which requires less than one-twentieth of the storage needed by its predecessor. This reduction in the size of the memory requirements of the algorithm is important because, in its previous formulation, the inverse DSTFT operation had storage needs that made it difficult, or even impossible, to implement in a small to medium sized computer.

In the first section of this chapter, the Short-Time Fourier Transform (STFT) is presented. Its implementable counterpart, the Discrete Short-Time Fourier Transform (DSTFT), is introduced in the second section. Finally, in the third section, efficient algorithms for the computation of the DSTFT (referred to as Analysis) and its inverse (referred to as Synthesis) are described.

3.1 The Short-Time Fourier Transform

The Short-Time Fourier Transform (STFT) is defined as follows [Weinstein, 1966]:

Let $x[n]$ denote a real valued discrete-time signal defined for all integers n.

Let $h[n]$ denote a real valued discrete-time window defined for all integers n, with $h[0] = 1$. Usually, but not necessarily, $h[n]$ will be zero for $-h_1 < n < h_2$, for some positive integers h_1 and h_2.

The STFT of $x[n]$ windowed by $h[n]$, and its inverse, are given by the relations:

$$\mathcal{F}_s: \quad x[n] \xrightarrow{\mathcal{F}_s} X_s[n,\omega] = \sum_{m=-\infty}^{+\infty} x[m]h[n-m]e^{-j\omega n} \tag{3.1}$$

$$\mathcal{F}_s^{-1}: \quad X_s[n,\omega] \xrightarrow{\mathcal{F}_s^{-1}} x[n] = \frac{1}{2\pi} \int_{-\pi}^{\pi} X_s[n,\omega]e^{j\omega n}d\omega \tag{3.2}$$

For convenience, let us define the two-dimensional sequence $x[n,m]$:

$$x[n,m] = x[m]h[n-m] \tag{3.3}$$

The sequence $x[n,m]$ is best thought of as a succession of frames along m, indexed by n. If we fix $n = n_0$ and let m vary from $-\infty$ to $+\infty$, $x[n_0,m]$ will be the original signal multiplied by the window $h[n]$, which is positioned over $x[n]$ so that $h[0]$ lies directly over $x[n_0]$.

Equation (3.1) then becomes:

$$X_s[n,\omega] = \sum_{m=-\infty}^{+\infty} x[n,m]e^{-j\omega m} \tag{3.4}$$

Equation (3.4) corresponds exactly to the standard one-dimensional Fourier transform of the sequence x[n,m] along the variable m, with n held fixed. $X_s[n,\omega]$ can therefore be interpreted as a sequence, indexed by n, of local spectra of x[n]. This interpretation is shown graphically in figure 3.1.

Let us now show, for completeness, that the inverse STFT defined in equation (3.2) in fact yields x[n] from $X_s[n,\omega]$.

Fixing n in $X_s[n,\omega]$, and inverse Fourier transforming the resulting function of ω, we obtain:

$$\frac{1}{2\pi} \int_{-\pi}^{\pi} X_s[n,\omega]e^{j\omega m}d\omega = x[n,m]$$

$$= x[m]h[n-m] \tag{3.5}$$

Setting m = n in equation (3.5), and recalling that h[0] = 1 by definition, we have:

$$\frac{1}{2\pi}\int_{-\pi}^{\pi} X_s[n,\omega]e^{j\omega n}d\omega = x[n]h[n-n]$$

$$= x[n]h[0]$$

$$= x[n] \qquad (3.6)$$

which is the desired result.

Because $X_s[n,\omega]$ is a redundant representation of $x[n]$, other inverse STFT relations, which are equivalent to equation (3.2), can be derived.

For example, replacing n by n_0 and m by n in equation (3.5), $x[n]$ can be recovered from $X_s[n,\omega]$ by the alternative formula:

$$x[n] = \frac{1}{2\pi h[n_0-n]}\int_{-\pi}^{\pi} X_s[n_0,\omega]e^{j\omega n}d\omega \qquad (3.7)$$

for any n and n_0 such that $h[n_0-n] \neq 0$.

Alternatively, if $H(\omega)$ is the Fourier transform of $h[n]$, it can be shown [Parsons, 1976; Allen, 1977] that:

$$x[n] = \frac{1}{2\pi H[\omega]\Big|_{\omega=0}}\int_{-\pi}^{\pi}\sum_{r=-\infty}^{+\infty} X_s[r,\omega]e^{j\omega n}d\omega \qquad (3.8)$$

A general equation that contains all the above inverse STFT relations as special cases was introduced by Portnoff [1978]:

$$x[n] = \frac{1}{2\pi} \int_{-\pi}^{\pi} \sum_{r=-\infty}^{+\infty} f[n,n-r] X_s[r,\omega] e^{j\omega n} d\omega \qquad (3.9)$$

Equations (3.2), (3.7) and (3.8) are special cases of equation (3.9), respectively, for:

$$f[n,m] = \delta[n] \quad , \qquad (3.10)$$

$$f[n,m] = \delta[n-m-n_0]/h[-m] \qquad (3.11)$$

and $$f[n,m] = 1 \qquad (3.12)$$

where $\delta[n]$ is the unit sample function.

Portnoff showed that equation (3.9) will be a valid inverse STFT formula if the sequences f[n,m] and h[n] satisfy the relation:

$$\sum_{m=-\infty}^{+\infty} f[n,-m]h[n] = 1 \quad , \quad \text{for all integers } n. \qquad (3.13)$$

The two-dimensional sequence f[n,m] is referred to as the SYNTHESIS FILTER. In general, it is a time-varying sequence. In this thesis, however, only time-invariant synthesis filters will be considered.

We define the time-invariant synthesis filter f[n] by the relation:

$$f[n] = f[0,n] \tag{3.14}$$

In terms of f[n], equation (3.9) becomes:

$$x[n] = \frac{1}{2\pi} \int_{-\pi}^{\pi} \sum_{r=-\infty}^{+\infty} f[n-r] X_s[n,\omega] e^{j\omega n} d\omega \tag{3.15}$$

Equation (3.15) is the inverse STFT relation that will be used in this thesis.

As shown in figure 3.1, the STFT of x[n] can be interpreted as a time sequence of local spectra of x[n]. Alternatively, the sequence $X_s[n,\omega]$ can be viewed as the output of a bank of filters excited by x[n]. Specifically, consider equation (3.1). This equation can be interpreted as a convolution sum, along the index n, of the sequence h[n] and the demodulated speech sequence $x[n]e^{-j\omega n}$:

$$X_s[n,\omega] = \sum_{m=-\infty}^{+\infty} x[m]h[n-m]e^{-j\omega m}$$

$$= \sum_{m=-\infty}^{+\infty} (x[m]e^{-j\omega m}) h[n-m]$$

$$= (x[n]e^{-j\omega n}) *_n h[n] \tag{3.16}$$

The sequence $X_s[n,\omega]$ can therefore be interpreted, for a fixed ω, as the output of a linear time-invariant filter with

Sliding Window Interpretation of the STFT

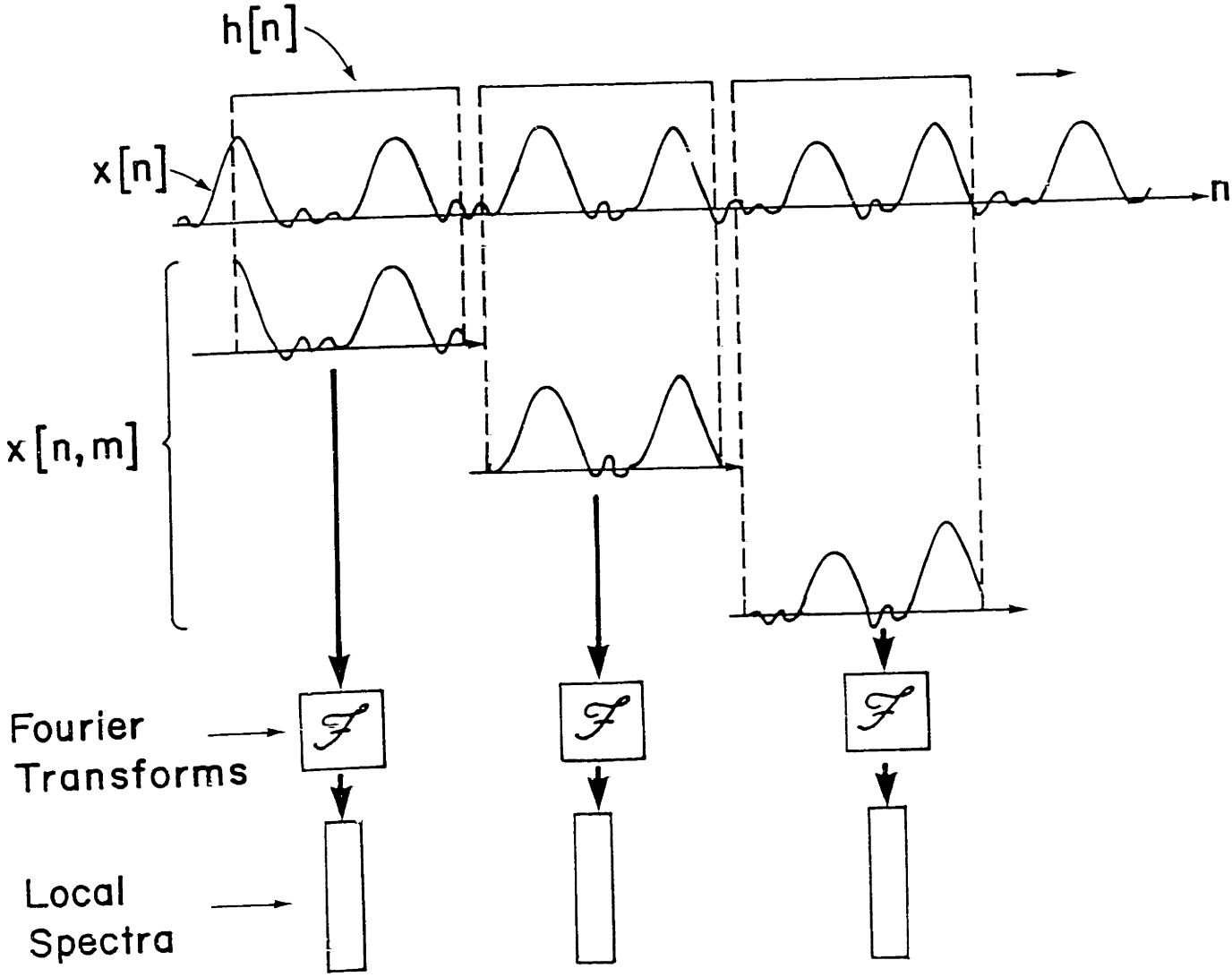


Figure 3.1

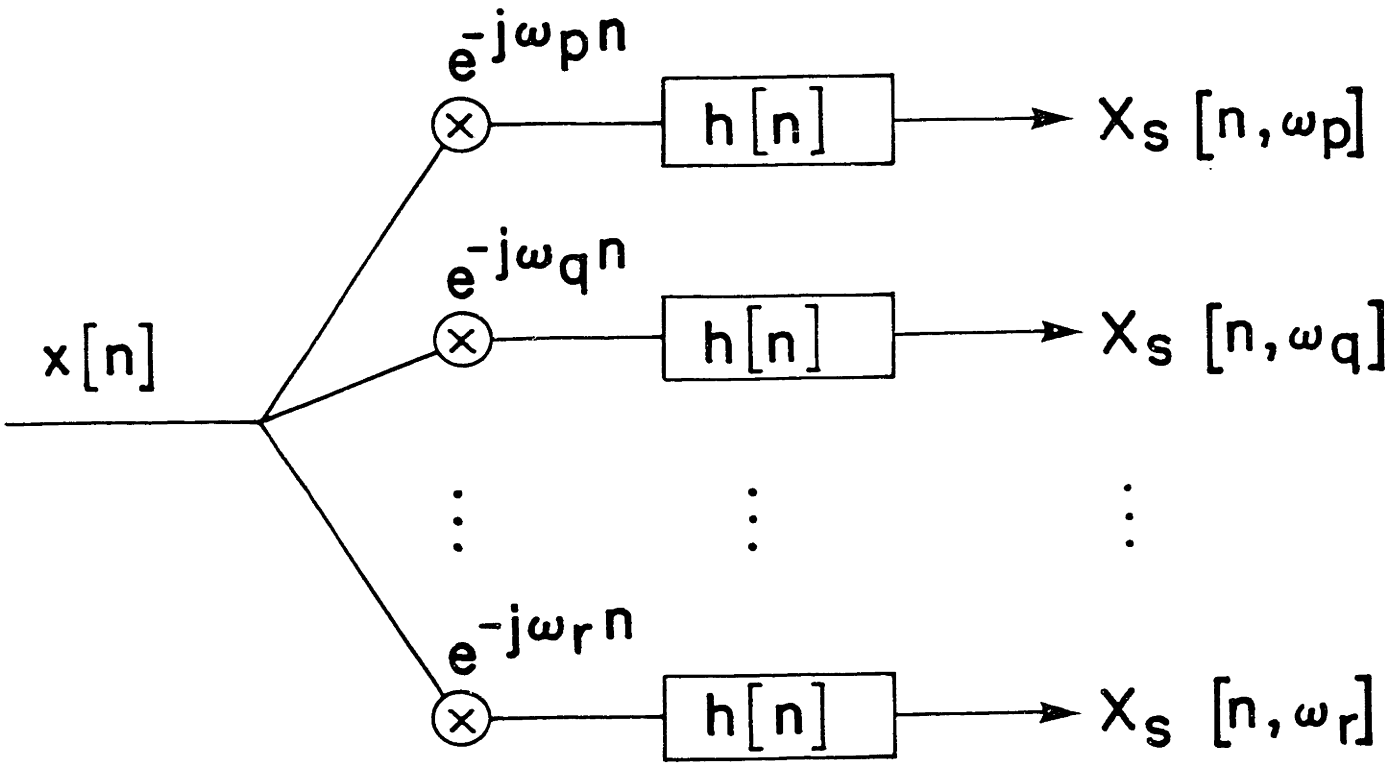
impulse response h[n], excited by the demodulated (frequency shifted) signal $x[n]e^{-j\omega n}$. Figure 3.2 shows schematically how $X_s[n,\omega]$ is obtained with a filter bank. Note that, although only a finite number of values of ω are shown in the figure, ω is a CONTINUOUS variable which takes values along all of $\mathbb{R}$.

In light of the filter bank interpretation of the STFT relations, it is natural to think of the filter h[n] as having low-pass spectral behavior. When this is the case, $X_s[n,\omega]$ will be the output of a bank of bandpass filters. As we shall see in the next section, if h[n] is a good low-pass filter, large computational savings can be obtained by decimating (downsampling in time) the function $X_s[n,\omega]$.

The alternative interpretations of the STFT of x[n] -- as a sequence of local spectra and as the output of a bank of bandpass filters -- impose conflicting constraints on the shape of h[n]. On one hand, the window h[n] must be small in order for each STFT sepctral frame (i.e. $X_s[n,\omega]$ seen along the variable ω, for a fixed value of the time index n) to correpond to the local behavior of x[n]. On the other hand, for h[n] to have good low-pass behavior, it must be relatively long. Experimentally, a good tradeoff between the two constraints is achieved by letting h[n] be a raised cosine (Hamming) window centered at h[0]. For the remainder of this thesis, then, h[n] will be assumed to be a Hamming window.

The relations for the STFT and its inverse constitute an analysis/synthesis pair. As mentioned earlier, we will refer to the sequence f[n] as the SYNTHESIS FILTER. Analogously, the

Filter Bank Interpretation of the STFT



NOTE: ω is a continuous variable; the indices p, q and r are used for illustrative purposes only.

Figure 3.2

window h[n] will be referred to as the ANALYSIS FILTER.

The STFT relations expressed in equations (3.1) and (3.15) are useful theoretically, but are of limited practical use in signal processing for two reasons. First, the frequency variable ω is continuous and, second, the summation limits in the analysis equation (3.1) are infinite. In the next section, these limitations will be removed from the STFT. The resulting transform pair is known as the Discrete Short-Time Fourier Transform (DSTFT).

3.2 The Discrete Short-Time Fourier Transform

The change from a continuous frequency variable in the STFT to a discrete frequency index can be accomplished by replacing ω in equations (3.1) and (3.15) with $k\Omega_0$. The number k is an integer which ranges from zero to some maximum value M-1, and $\Omega_0 = 2\pi/M$ is the frequency sampling interval. The index k does not vary from $-\infty$ to $+\infty$ because complex exponentials of the form $e^{-j2\pi kn/M}$ are periodic in k with period M and, therefore, a discrete Fourier transform representation of a signal involves only one such period [Oppenheim and Schafer, 1975].

The resulting Discrete Short-Time Fourier Transform (DSTFT) relations are:

$$X_s[n,k] = \sum_{m=-\infty}^{+\infty} x[m]h[n-m]e^{-j\Omega_0 km} \tag{3.17}$$

$$x[n] = \frac{1}{M} \sum_{k=0}^{M-1} \sum_{m=-\infty}^{+\infty} f[n-m]X_s[n,k]e^{j\Omega_0 kn} \tag{3.18}$$

$$\text{where } \Omega_0 = \frac{2\pi}{M} \tag{3.19}$$

and k is an integer between 0 and M-1.

Equation (3.18) is a direct counterpart of equation (3.15) with ω replaced by $k\Omega_0$. Since the new frequency variable is discrete, the integral along ω in equation (3.15) has been replaced by a summation, and the normalization constant $1/2\pi$ has been replaced by $1/M$.

The constraint on the pair of functions h[n] and f[n] for equation (2.17) to be a valid inverse DSTFT relation is similar to the one in the continuous frequency case, given in equation (3.13). Portnoff [1978] has shown that x[n] will be recovered from $X_s[n,k]$ by equation (3.18) if h[n], f[n] and M satisfy the relation:

$$\sum_{m=-\infty}^{+\infty} f[n-m]h[m-n+qM] = \delta[q] \tag{3.20}$$

where $\delta[n]$ is the unit sample function.

A functional diagram of equations (3.17) and (3.18) in their filter bank interpretation is presented in figure 3.3. Note that, in contrast to figure 3.2, the discrete frequency indices in figure 3.3 correspond to the actual structure of the DSTFT.

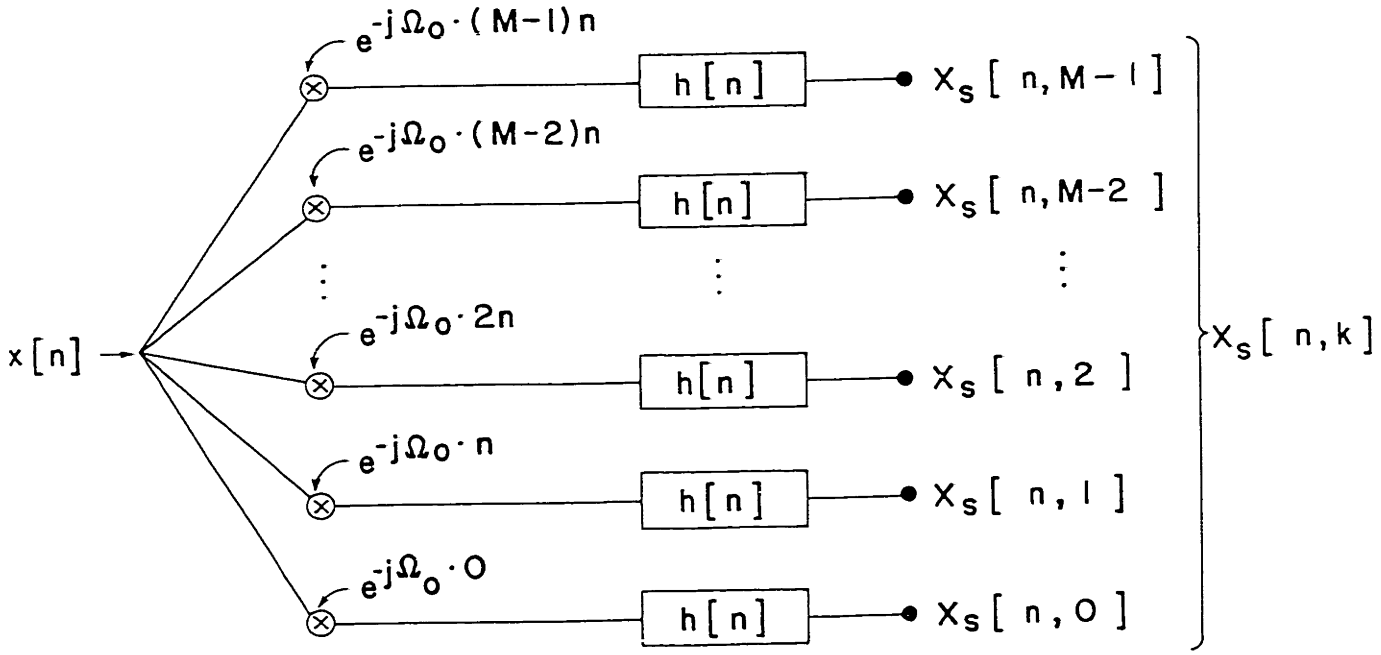
The output of each analysis filter in figure 3.3 consists of a demodulated and low-pass filtered version of the input signal $x[n]$. The sequence $X_s[n,k]$, viewed only as a function of n with k held fixed, has a spectrum that corresponds to a band-pass section of the spectrum of $x[n]$ centered at $\omega_0 = k\Omega_0$ but shifted so that ω_0 is mapped to zero. This result is shown graphically in figure 3.4.

This fact can be easily derived by noting that the Fourier transform of the signal $x[n]e^{-jk\Omega_0 n}$ is $X(\omega+k_0\Omega_0)$. Low-pass filtering $x[n]e^{-jk\Omega_0 n}$ with the filter $h[n]$ causes the frequencies outside the filter passband to be severely attenuated, yielding the desired result.

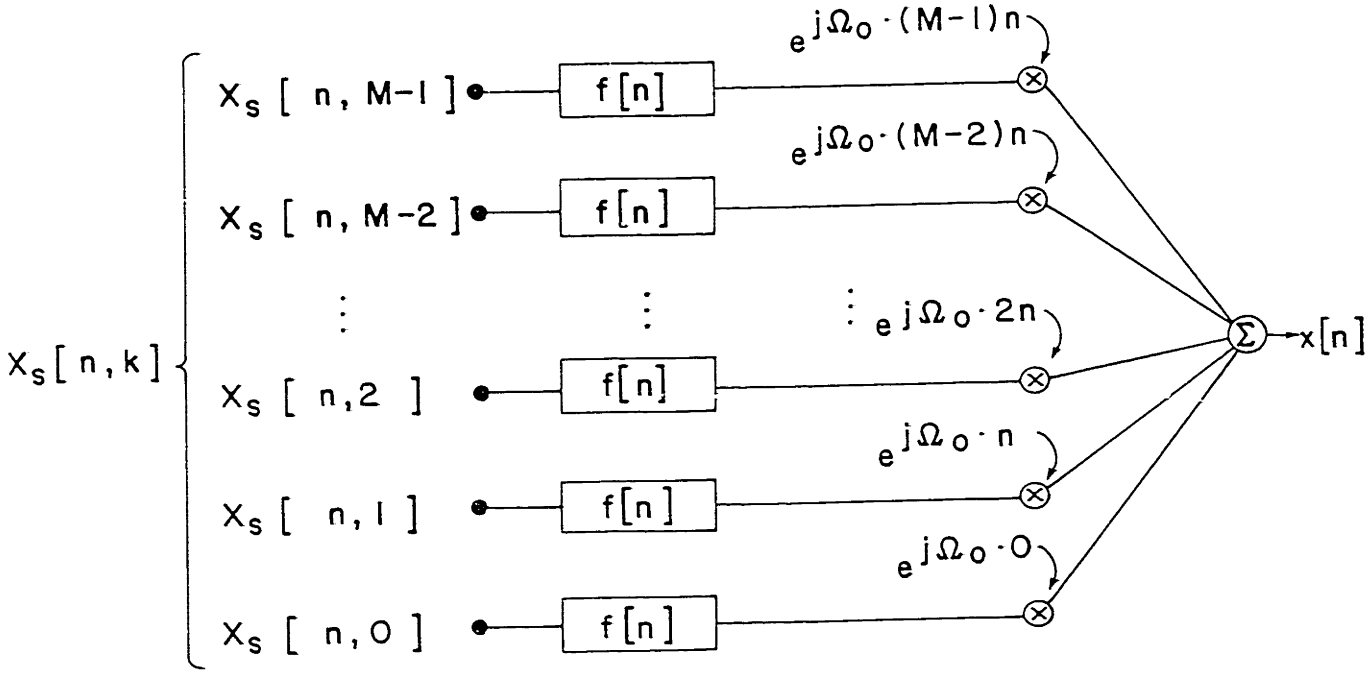
The band limitation of $X_s[n,k]$ as a sequence in n, holding k fixed, can be exploited to drastically reduce the amount of computation needed to evaluate it. Since its spectrum does not span the whole frequency range from $-\pi$ to π, $X_s[n,k]$ can be down-sampled (decimated) in n without any loss of information, as long as the resulting sampling rate exceeds the Nyquist rate associated with the low-pass filter $h[n]$.

Keeping only every R^{th} sample of $X_s[n,k]$, we obtain the decimated transform:

The DSTFT Seen as a Filter Bank



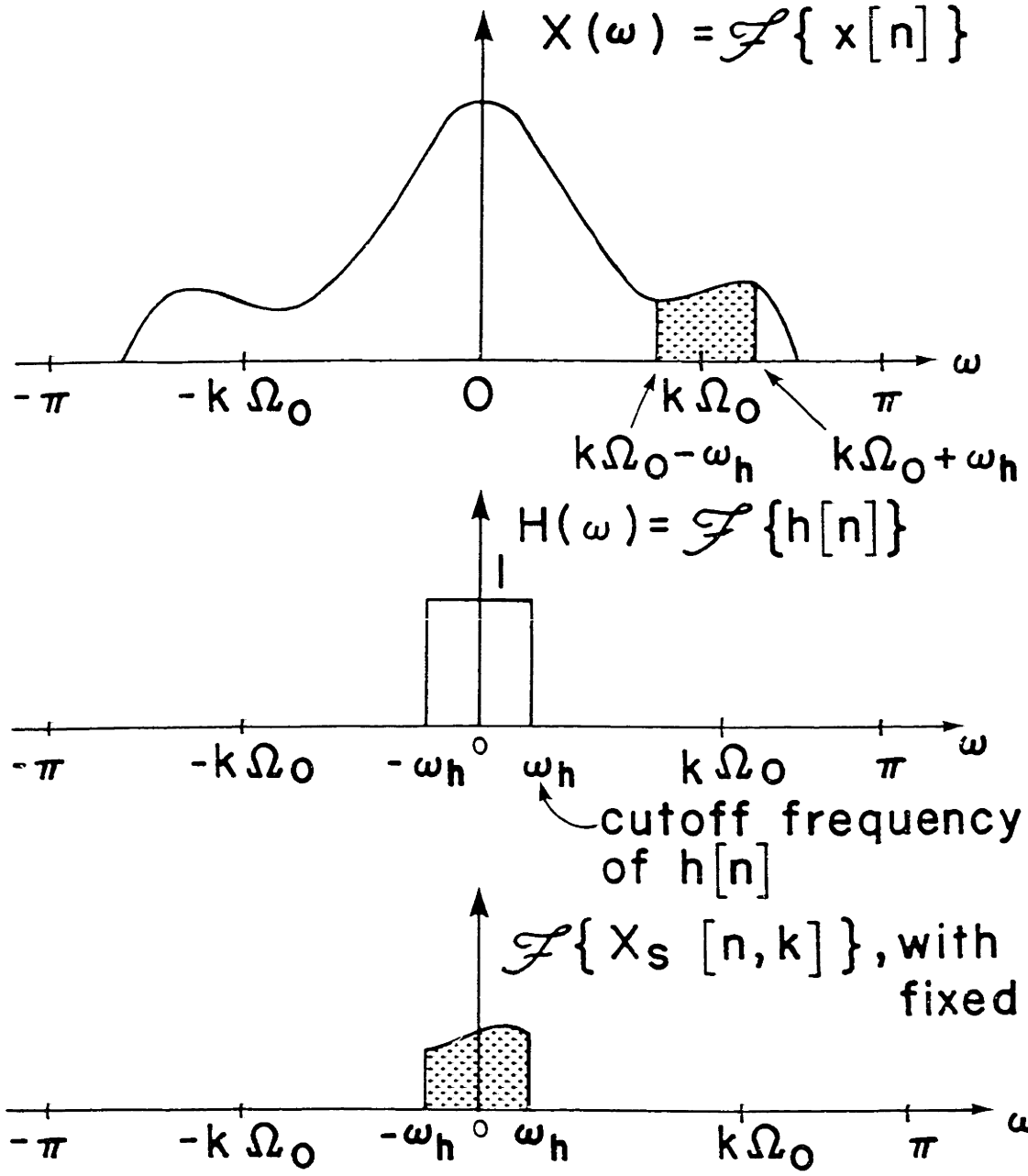
a) Analysis -- equation (3.17)



b) Synthesis -- equation (3.18)

Figure 3.3

Modulating and Filtering x[n]



If k is held fixed, X[n,k] is a frequency shifted band-pass section of x[n].

Figure 3.4

$$X_s[pR,k] = X_s[n,k]\Big|_{n=pR} \tag{3.21}$$

for any integer p.

Thus, the decimated DSTFT of x[n], $X_s[pR,k]$, consists of a set, indexed by k, of M sequences in the decimated time variable p. Alternatively, $X_s[pR,k]$ can be seen as a succession of local spectra of x[n], separated in time by R samples. Each one of these local spectra is obtained by letting the index k vary from 0 to M-1 while p is held fixed. Viewing $X_s[pR,k]$ as a set of M decimated time sequences corresponds to the filter-bank interpretation of the DSTFT shown in figure 3.3, while viewing it as a succession of local spectra of x[n] corresponds to the sliding window interpretation illustrated in figure 3.1

Both interpretations will be used throughout this thesis. The choice between the two during any specific derivation will be one of convenience and simplicity.

To avoid any confusion that might arise later, we shall adopt the convention that any variable that is underlined is being held fixed. Thus, $X_s[\underline{p}R,k]$ is a succession of local spectra (sliding window interpretation -- p held constant), while $X_s[pR,\underline{k}]$ is a set of M decimated time sequences (filter bank interpretation -- k held constant).

Now, we define $\mathcal{F}\{X_s[pR,\underline{k}]\}$ as the Fourier transform of $X_s[pR,\underline{k}]$. Note that $\mathcal{F}\{X_s[pR,\underline{k}]\}$ is the Fourier transform of a sequence in time, so its image is in the frequency domain. It

is useful to relate $\mathcal{F}\{X_s[pR,\underline{k}]\}$ to $\mathcal{F}\{X_s[n,\underline{k}]\}$, the Fourier transform along the same time variable prior to down-sampling:

$$\mathcal{F}\{X_s[pR,\underline{k}]\} = \sum_{p=-\infty}^{+\infty} X_s[pR,\underline{k}]e^{-j\gamma p}$$

$$= \sum_{n=-\infty}^{+\infty} X_s[n,k] \left\{ \frac{1}{R} \sum_{r=0}^{R-1} e^{j2\pi rn/R} \right\} e^{-j\gamma n/R}$$

where the term in brackets is the discrete Fourier series representation of a periodic sequence that equals unity when n is an integer multiple of R, and is zero otherwise.

$$= \frac{1}{R} \sum_{r=0}^{R-1} \sum_{n=-\infty}^{+\infty} X_s[n,\underline{k}]e^{-jn(\gamma-2\pi r)/R}$$

$$= \frac{1}{R} \sum_{r=0}^{R-1} \mathcal{F}\{X_s[n,\underline{k}]\}\Big|_{\omega = (\gamma-2\pi r)/R} \qquad (3.22)$$

In other words, $\mathcal{F}\{X_s[pR,\underline{k}]\}$ is obtained by adding together R replicas of $\mathcal{F}\{X_s[n,\underline{k}]\}$, separated by $2\pi/R$ from each other, and with the frequency dimension scaled by 1/R.

As long as R is small enough to ensure that the sampling frequency of the decimated transform exceeds the Nyquist rate of the analysis filter, the complete DSTFT sequence, $X_s[n,k]$, can be recovered from its decimated version by bandlimited interpolation. Schafer and Rabiner [1973(b)] treat the problem of bandlimited

decimation and interpolation in detail.

The decimation of $X_s[n,k]$ to obtain $X_s[pR,k]$ is only a conceptual operation. In practice, only those samples of the DSTFT which will be retained need be computed. The resulting decimated DSTFT relations are:

$$X_s[pR,k] = \sum_{n=-\infty}^{+\infty} x[n]h[pR-n]e^{-jk\Omega_0 n} \tag{3.23}$$

$$x[n] = \frac{1}{M} \sum_{k=0}^{M-1} \sum_{p=-\infty}^{+\infty} f[n-pR]X_s[pR,k]e^{jk\Omega_0 n} \tag{3.24}$$

with the constraint:

$$\sum_{p=-\infty}^{+\infty} f[n-pR]h[pR-n+qM] = \delta[q], \text{ for all integers } n \tag{3.25}$$

The synthesis filter f[n] in equations (3.24) and (3.25) is chosen to be a 1-to-R interpolating FIR filter, as defined by Schafer and Rabiner [1973(b)]. This choice of f[n] must satisfy equation (3.25). To show that this is the case, we can interpret equation (3.25) as an interpolation sum. Schafer and Rabiner [1973(b)] have shown that the sequence h[n] can be recovered from its decimated version h[pR] by the formula:

$$h[n] = \sum_{p=-\infty}^{+\infty} f[n-pR]h[pR] \tag{3.26}$$

This assumes, of course, that R is small enough not to cause aliasing in the sequence h[pR].

For the specific sequences f[n] and h[n] that we have chosen, the left-hand side of equation (3.25) can be evaluated using equation (3.26). Thus:

$$\sum_{p=-\infty}^{+\infty} f[n-pR]h[pR+(qM-n)] = h[n+(qM-n)]$$

$$= h[qM] \qquad (3.27)$$

Therefore, for equation (3.25) to be satisfied by f[n] and h[n] when the former is an interpolating filter, we must choose h[m] such that:

$$h[qM] = \delta[q] \qquad (3.28)$$

In Section 3.1, we restricted h[m] to be a Hamming window. Equation (3.28) will be satisfied if we restrict the length of h[m] to be less than 2M. Portnoff [1976, 1978] considered the possibility of using analysis filter lengths greater than 2M. In this case, equation (3.28) requires that h[n] be zero for n equal to a non-zero multiple of M. For this reason, Portnoff considered analysis filter shapes other than Hamming windows (e.g., truncated sinc functions). Experimental results have shown, however, that when the DSTFT representation of a signal is modified to effect time-scale modification

of the signal x[n] (as will be described in Chapter 4), analysis window lengths greater than M cause noticeable reverberation in the output signal $x^{s}[n]$. Thus, for the remainder of the thesis, the length of h[n] will be assumed to be less than or equal to the number M of DSTFT frequency samples.

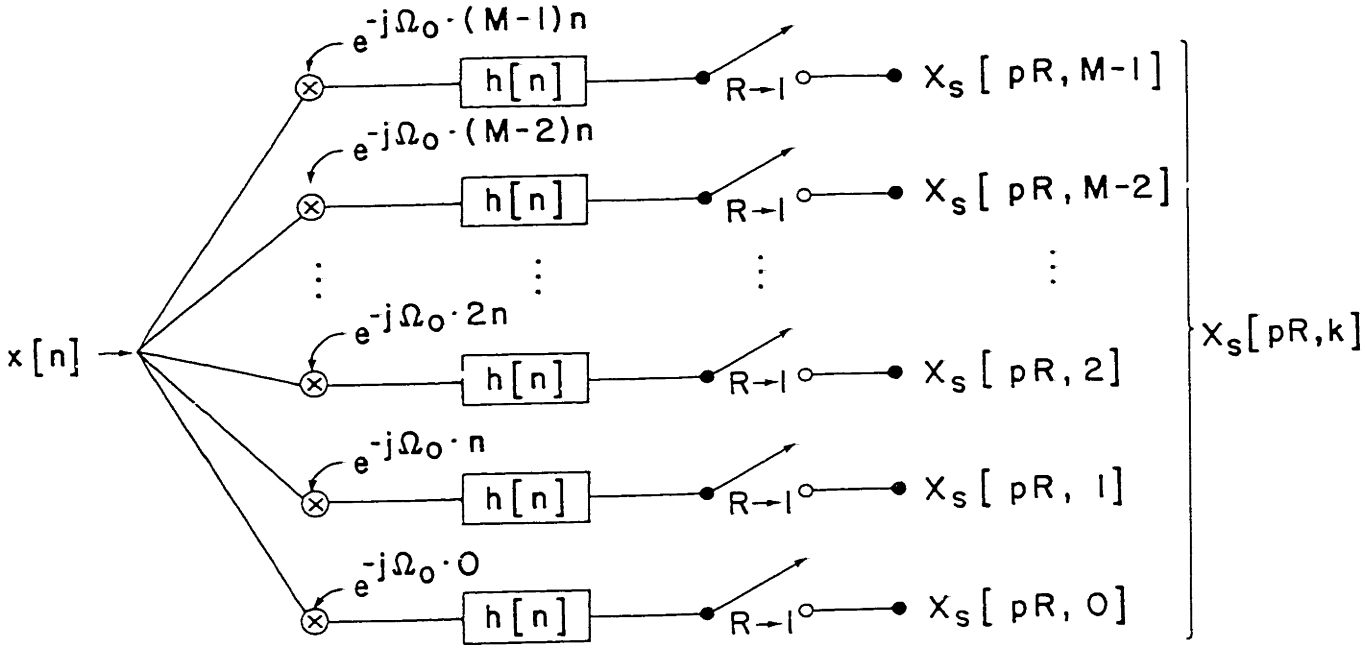
Equations (3.23) and (3.24) define the DSTFT. In addition, for equation (3.24) to hold, the sequences h[n] and f[n], together with the number M, must satisfy the constraint imposed by equation (3.25). In particular, when f[n] is a 1-to-R interpolating filter, the constraint will be satisfied if h[n] is a Hamming window whose length is less than M. The filter bank interpretation of equations (3.23) and (3.24) is illustrated in figure 3.5.

This definition of the DSTFT is still of limited use computationally because the limits on the summations over n and p remain infinite. It is a simple matter, however, to replace these limits by finite ones.

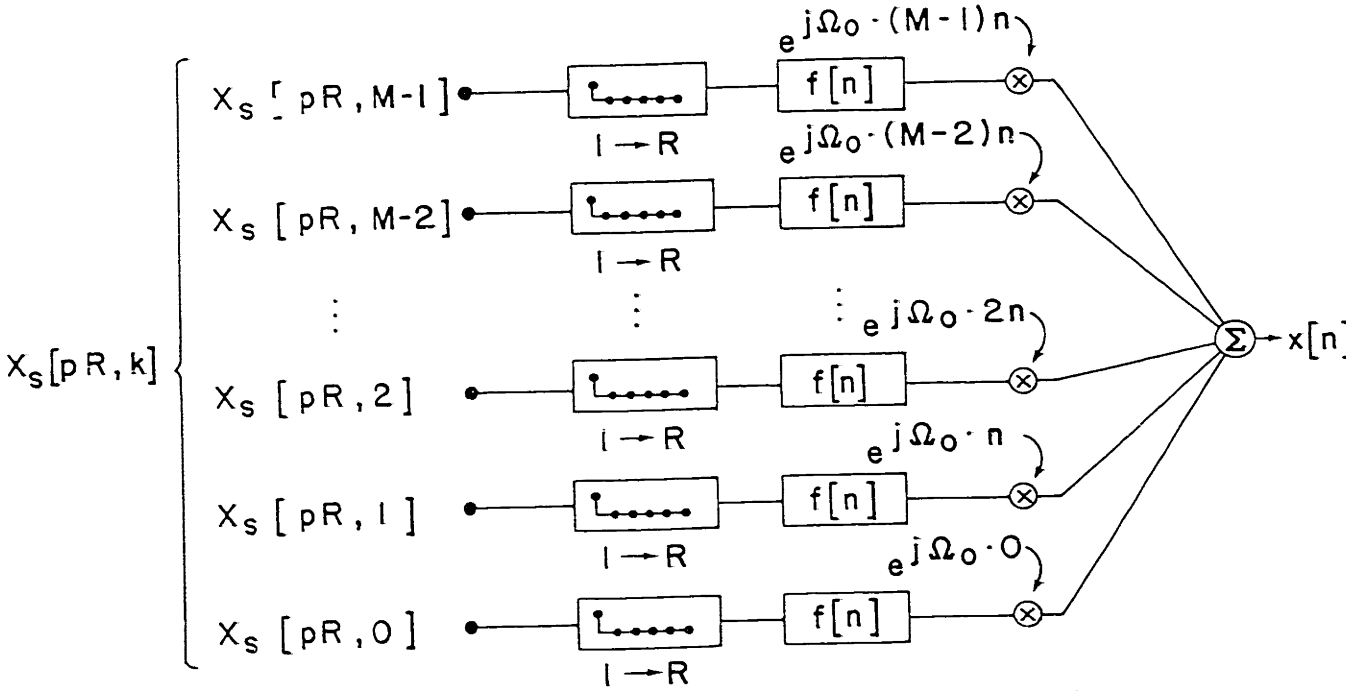
Consider equation (3.23). In general, the summation limits over n are infinite. In practice, however, the length of the window h[n] is finite and, thus, the tails of the summation make no contribution to $X_s[pR,k]$. Let h[n] be non-zero only for values of n in the range from $-h_1$ to h_2. That is, h[n] is $h_1 + h_2 + 1$ points long. Equation (3.23) then becomes:

$$X_s[pR,k] = \sum_{n=pR-h_2}^{pR+h_1} x[n]h[pR-n]e^{-jk\Omega_0 n} \qquad (3.29)$$

The Decimated DSTFT Seen as a Filter Bank



a) Analysis -- equation (3.23)



b) Synthesis -- equation (3.24)

Figure 3.5

Now, consider equation (3.24). The infinite summation limits cannot be replaced by finite ones as in equations (3.23) and (3.29) because ideal interpolating filters are infinitely long. Excellent, albeit not ideal, interpolation filters can be designed with finite lengths [Schafer and Rabiner, 1973(b); Oetken, Parks and Schuessler, 1975]. The length of these filters is measured in terms of the number of points from the original sequence that are involved in the computation of a given point of the interpolated sequence. An interpolating filter is said to be of order Q if it requires 2Q input points to compute an output point. (An even number of input points is required because, in general, interpolating filters are odd symmetric sequences).

Portnoff [1978] has shown that if f[n] is a 1-to-R interpolating filter of order Q, then the actual summation limits on p in equation (3.24) are:

$$L^{+}[n] = \lceil n/R \rceil + Q \tag{3.30}$$

$$L^{-}[n] = \lceil n/R \rceil - Q - 1 \tag{3.31}$$

where $\lceil a \rceil$ denotes the largest integer that is less than or equal to a.

Equation (3.24) can then be rewritten with finite limits:

$$x[n] = \frac{1}{M} \sum_{k=0}^{M-1} \sum_{p=L^{-}[n]}^{L^{+}[n]} f[n-pR] X_s[pR,k] e^{jk\Omega_0 n} \tag{3.32}$$

Equations (3.29) and (3.32) constitute a definition of the decimated DSTFT which can be implemented as a computer algorithm. All of the variables in the two equations are discrete, and their summation limits are finite. If they are implemented literally, however, their computational efficiency is low. The next, and last, section of this chapter presents efficient DSTFT analysis and synthesis procedures.

3.3 DSTFT Analysis and Synthesis Algorithms

If the number M of DSTFT frequency samples is chosen to be an integer power of 2, then equations (3.29) and (3.32) can be implemented very efficiently using a Fast Fourier Transform (FFT) algorithm. To conform to the notation commonly used in describing FFT algorithms, let us define the number W_M^{km}:

$$W_M^{km} = e^{-j2\pi km/M}$$

$$= e^{-j\Omega_0 km} \tag{3.33}$$

Equations (3.29) and (3.32) can then be rewritten using the FFT notation:

$$X_s[pR,k] = \sum_{n=pR-h_2}^{pR+h_1} x[n]h[pR-n]W_M^{kn} \tag{3.34}$$

$$x[n] = \frac{1}{M} \sum_{k=0}^{M-1} \sum_{p=L^-[n]}^{L^+[n]} f[n-pR]X_s[pR,k]W_M^{-km} \tag{3.35}$$

Efficient algorithms for the implementation of DSTFT Analysis (equation (3.34)) and DSTFT Synthesis (equation (3.35)) are derived separately in the remainder of this chapter.

The analysis algorithm presented here follows closely the one derived by Portnoff [1976, 1978]. It is simpler than its predecessor, however, because it takes advantage of our assumption that the length of h[n] is less than the number M of frequency samples.

Portnoff's synthesis algorithm is also described here. It will be shown to have very large storage requirements. An alternative algorithm will be presented which needs less than one-twentieth of the storage used by Portnoff's algorithm, without sacrificing its computational efficiency.

3.3.1 Short-Time Analysis Algorithm

Equation (3.34) can be implemented using an FFT algorithm by a simple change of variable.

Letting $r = n - pR + h_2$, equation (3.24) can be rewritten using r as the summation index:

$$X_s[pR,k] = \sum_{r=0}^{h_1+h_2} x[r+pR-h_2]h[h_2-r]W_M^{k(r+pR-h_2)} \tag{3.36}$$

For convenience, define $x_p[r]$ as follows:

$$x_p[r] = x[r+pR-h_2]h[h_2-r] \tag{3.37}$$

Then, equation (3.36) can be rewritten:

$$X_s[pR,k] = W^{k(pR-h_2)} \sum_{r=0}^{h_1+h_2} x_p[r] W_M^{kr} \tag{3.38}$$

Equation (3.38) expresses $X_s[pR,k]$ as the product of a phase term, $W_M^{k(pR-h_2)}$, and a summation which is in fact the Discrete Fourier Transform (DFT) of the sequence $x_p[r]$. If M is chosen to be an integer power of 2, this equation can be implemented with an FFT algorithm. Let us assume, then, that $M = 2^m$, for some integer m.

Define the M-point sequence $\tilde{x}_p[r]$ as follows:

$$\tilde{x}_p[r] = \begin{cases} x_p[r] & , \text{ for } r = 0 \text{ to } h_1+h_2 \\ 0 & , \text{ for } r = h_2+h_1+1 \text{ to } M-1 \end{cases} \tag{3.39}$$

In terms of $\tilde{x}_p[r]$, equation (3.38) becomes:

$$X_s[pR,k] = W_M^{k(pR-h_2)} \sum_{r=0}^{M-1} \tilde{x}_p[r] W_M^{kr} \tag{3.40}$$

The summation in equation (3.40) can be implemented using an FFT algorithm. This equation can be further simplified by replacing the product of the DFT of $\tilde{x}_p[r]$ and the linear phase term $W_M^{k(pR-h_2)}$ with a circular left shift of the sequence $\tilde{x}_p[r]$ in r

by the amount $pR - h_2$. This can be done because, for the Discrete Fourier Transform operator, multiplication by a complex exponential in the frequency domain corresponds to a circular shift in the time domain.

Equation (3.41) is equivalent to equation (3.40) with the phase term $W_M^{k(pR-h_2)}$ replaced by a circular shift. Together with equation (3.39), which is repeated here as equation (3.42), equation (3.41) constitutes an implementable and efficient DSTFT analysis algorithm, as shown in Table II.

The algorithm defined by equations (3.41) and (3.42) is shown in flowchart form in figure 3.6. The figure depicts the computation of a single DSTFT frame, for a given value of p (that is, the computation of $X_s[pR,k]$). Other frames are computed by repeating the same procedure for other values of p.

It is important to note that x[n] is assumed to be infinite. In practice, this means that x[n] is "padded" with zeros at both ends when the first and last few frames are analyzed.

3.3.2 Short-Time Synthesis Algorithm

Equation (3.35) synthesizes the signal x[n] from its DSTFT representation, $X_s[pR,k]$. It implements the inverse of the analysis procedure expressed by equation (3.34). Therefore, the two equations together form a transform pair.

The DSTFT analysis and synthesis equations can also be considered independently. In this sense, equation (3.34) takes two one-dimensional sequences and generates a two-dimensional sequence.

DSTFT Analysis Algorithm (see equations 3.41 & 3.42)

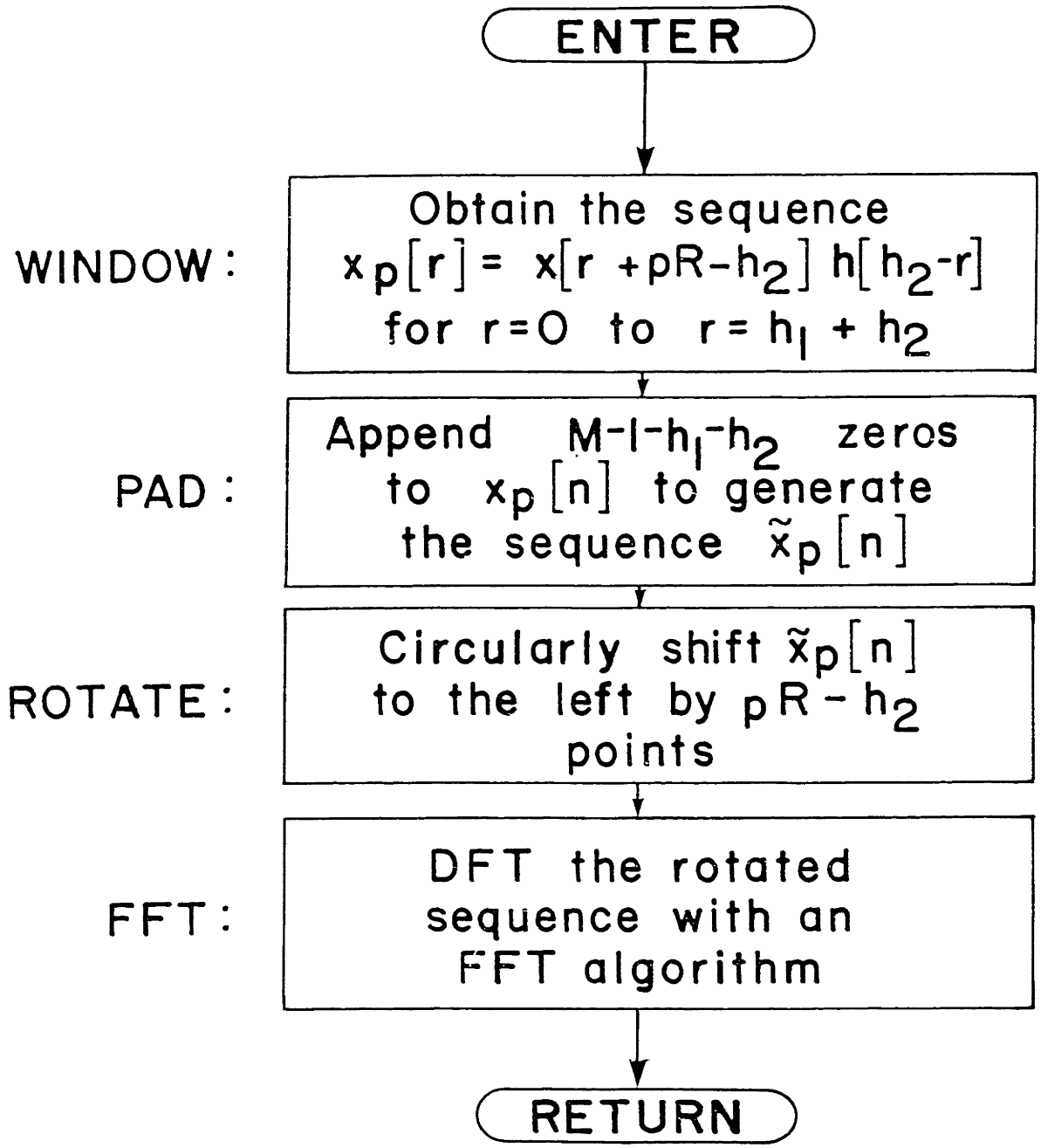


Figure 3.6

Table II

DSTFT Analysis Algorithm

$$X_s[pR,k] = \sum_{r=0}^{M-1} \tilde{x}_p[((r-pR+h_2))_M]W_M^{kr} \tag{3.41}$$

where:

$$\tilde{x}_p(r) = \begin{cases} x[r+pR-h_2]h[h_2-r] & , \quad r = 0 \text{ to } h_1 + h_2 \\ 0 & , \text{ otherwise} \end{cases} \tag{3.42}$$

and where: $((r))_M$ denotes "r modulo M."

In turn, equation (3.35) generates a one-dimensional sequence from two others with one and two dimensions, respectively.

In the next chapter, the DSTFT analysis and synthesis algorithms are used separately to design a TSM system. For this purpose, the input and output sequences of the synthesis equation will be renamed to distinguish them from their counterparts in the analysis equation.

Let $Y_s[pR,k]$ denote the input sequence in equation (3.35) and let $y[n]$ denote the corresponding output sequence. Equation (3.35) then becomes:

$$y[n] = \frac{1}{M} \sum_{k=0}^{M-1} \sum_{p=L^-[n]}^{L^+[n]} f[n-pR] Y_s[pR,k] W_M^{-kn} \qquad (3.43)$$

Clearly, if we choose $Y_s[pR,k]$ to equal $X_s[pR,k]$, equations (3.42) and (3.43) will be identical. In this case, $y[n]$ will equal $x[n]$, and equations (3.41) and (3.43) will constitute a transform pair.

The efficiency of equation (3.43) can be greatly improved by using an FFT algorithm to perform the bulk of the computation. However, in order to use an FFT, we must change the form of equation (3.43).

Interchanging the order of summation and rearranging terms, we have:

$$y[n] = \sum_{p=L^-[n]}^{L^+[n]} f[n-pR] \left\{ \frac{1}{M} \sum_{k=0}^{M-1} Y_s[pR,k] \; W_M^{-kn} \right\} \tag{3.44}$$

The term in brackets in equation (3.44) is the inverse DFT of the one-dimensional sequence $Y_s[pR,k]$, and thus it can be computed using an FFT algorithm.

The inverse DFT of $Y_s[pR,k]$ will yield a sequence $y[pR,k]$ which is analogous to the sequence $x[n,m]$ defined in equation (3.3), decimated at a rate R along the index n. In fact, if no modification is effected, the sequence $y[pR,m]$ will be equal to $x[pR,m]$.

Equation (3.44) can therefore be rewritten as two separate equations in terms of the sequence $y[pR,m]$:

$$y[n] = \sum_{p=L^-[n]}^{L^+[n]} f[n-pR]y[pR,n] \tag{3.45}$$

where:

$$y[pR,n] = \frac{1}{M} \sum_{k=0}^{M-1} Y_s[pR,k]W_M^{-kn} \tag{3.46}$$

A useful interpretation of equations (3.45) and (3.46) can be obtained by making use of the fact that W_M^{kn} is periodic in n with period M. Thus, the sequence $y[pR,((n))_M]$ is identical to the sequence $y[pR,n]$.

Equations (3.45) and (3.46) can then be rewritten in terms of $y[pR,((n))_M]$:

$$y[n] = \sum_{p=L^-[n]}^{L^+[n]} f[n-pR]\,y[pR,((n))_M] \tag{3.47}$$

where:

$$y[pR,((n))_M] = \frac{1}{M} \sum_{k=0}^{M-1} Y_s[pR,k]\,W_M^{kn} \tag{3.48}$$

Figure 3.7 illustrates the synthesis procedure described by equations (3.47) and (3.48). This figure emphasizes the derivation of the sequence y[n] from y[pR,n]. Portnoff [1976, 1978] has shown that:

$$y[n] = y[n,m]\Big|_{m = ((n))_M} \tag{3.49}$$

The sequence y[n,m] must be recovered from y[pR,m] by interpolation. However, it is not necessary to obtain y[m,n] in its entirety. Since y[n] is equal to $y[n,((n))_M]$, only those points of y[n,m] which lie along the helical path determined by the relation $m = ((n))_M$ need to be recovered.

The DSTFT synthesis algorithm described in equations (3.47) and (3.48) is implementable in a computer and has been shown to be computationally efficient [Portnoff, 1978]. Still, it has one major disadvantage. Since several frames of y[pR,m] have to be accessed simultaneously to interpolate y[n,m], its storage requirements are very large. Therefore, this procedure is difficult, or even

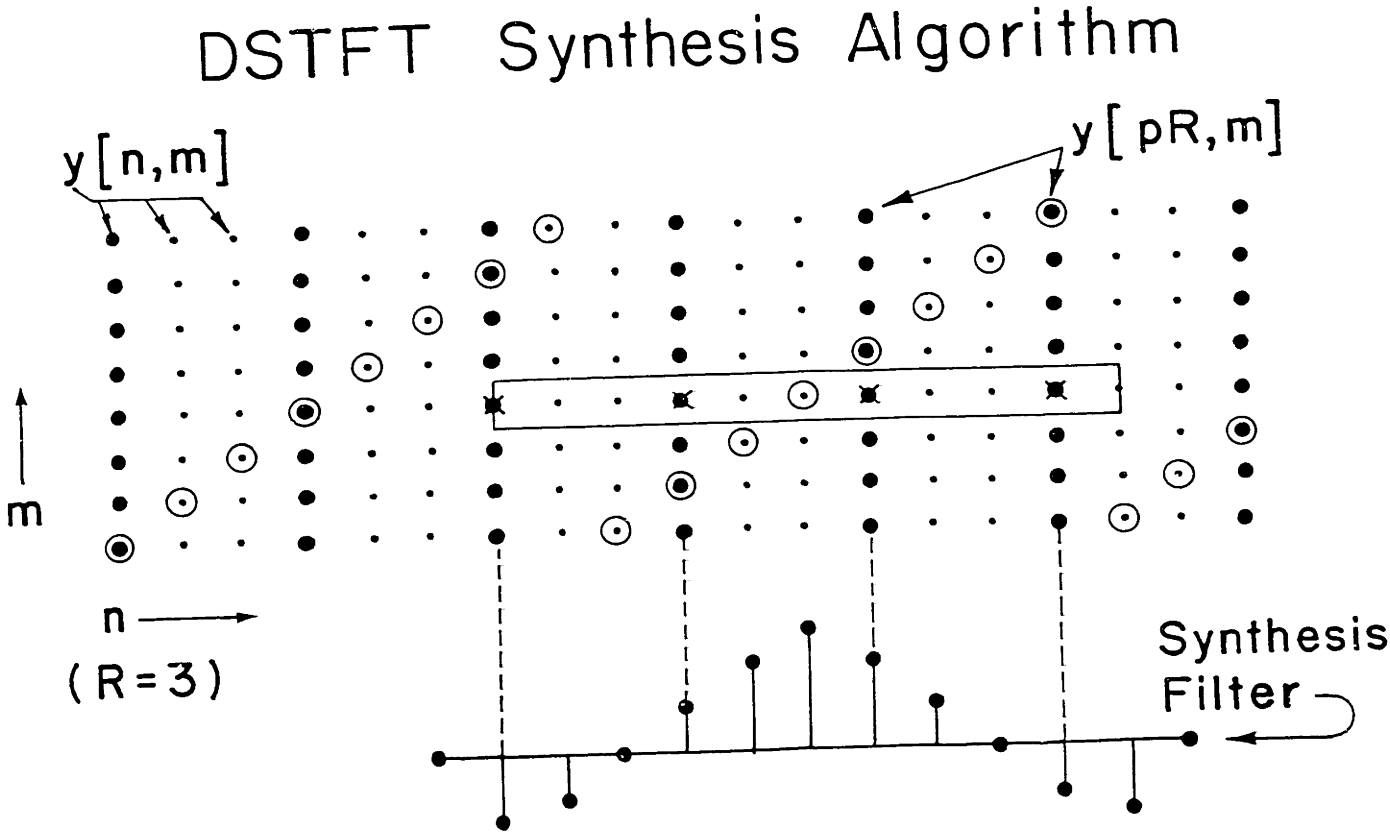


Figure 3.7

impossible, to implement in a small or medium sized computer.

In general, the order Q of a good interpolating filter will be between 10 and 20. The number of frames that such a filter would need to access simultaneously is 2Q + 1. Therefore, 21 ∿ 41 frames will usually be required in memory at any given time.

An alternative interpretation of equation (3.47) can reduce the number of frames that need to be accessed simultaneously from 2Q + 1 to one.

Let us define the sequence $\psi_r[n]$ as follows:

$$\psi_r[n] = f[n-r]y[r,((n))_M] \tag{3.50}$$

Equation (3.48) can then be rewritten in terms of $\psi_r[n]$:

$$y[n] = \sum_{P=L^-[n]}^{L+[n]} \psi_{pR}[n] \tag{3.51}$$

As a function of the variable n, for a given fixed value of r, $\psi_r[n]$ is the product of the interpolating filter f[n] and the sequence y[r,m] evaluated along the path $m = ((n))_M$. Without loss of generality, we can let f[0] correspond to the center value of the (odd) sequence f[n]. Then, since we have defined f[n] as a 1-to-R interpolating filter of order Q, we know that f[n] is equal to zero for values of n outside the range from -RQ to RQ. Consequently, $\psi_r[n]$ will equal zero if n-r < -RQ or n-r > RQ. In particular for r = pR:

$$n \notin [pR-RQ, pR+RQ] \Rightarrow \psi_{pR} = 0 \tag{3.52}$$

The sequence $\psi_{pR}[n]$ can be interpreted as the contribution to the output sequence y[n] from the p^{th} frame of the decimated sequence y[pR,m]. The fact that this contribution is zero beyond the length of the interpolating filter makes sense because one would not expect a given frame of y[pR,m] to contain any information about y[n] for values of n far away from pR.

Figure 3.8 shows how y[n] is constructed from the sequences $\psi_{pR}[n]$. Since these sequences are summed together and they overlap in time, we will refer to this synthesis algorithm as the OVERLAP-ADD synthesis algorithm; its structure is similar to that of the well-known fast convolution algorithm that bears this name [Stockham, 1966]. For the purpose of illustration, f[n] is assumed to be a sinc function with three pairs of zero crossings (too short for actual use), and the values of y[pR,m] are assumed to be equal to unity for all n and m, so that the familiar shape of the filter f[n] stands out in figure 3.8.

To describe the overlap-add synthesis algorithm, let us define the sequence $y_p[n]$ as the sum of the sequences $\psi_{pR}[n]$ for p ranging from $-\infty$ to P:

$$y_p[n] = \sum_{p=-\infty}^{P} \psi_{pR}[n] \tag{3.53}$$

DSTFT Overlap-Add Synthesis Procedure

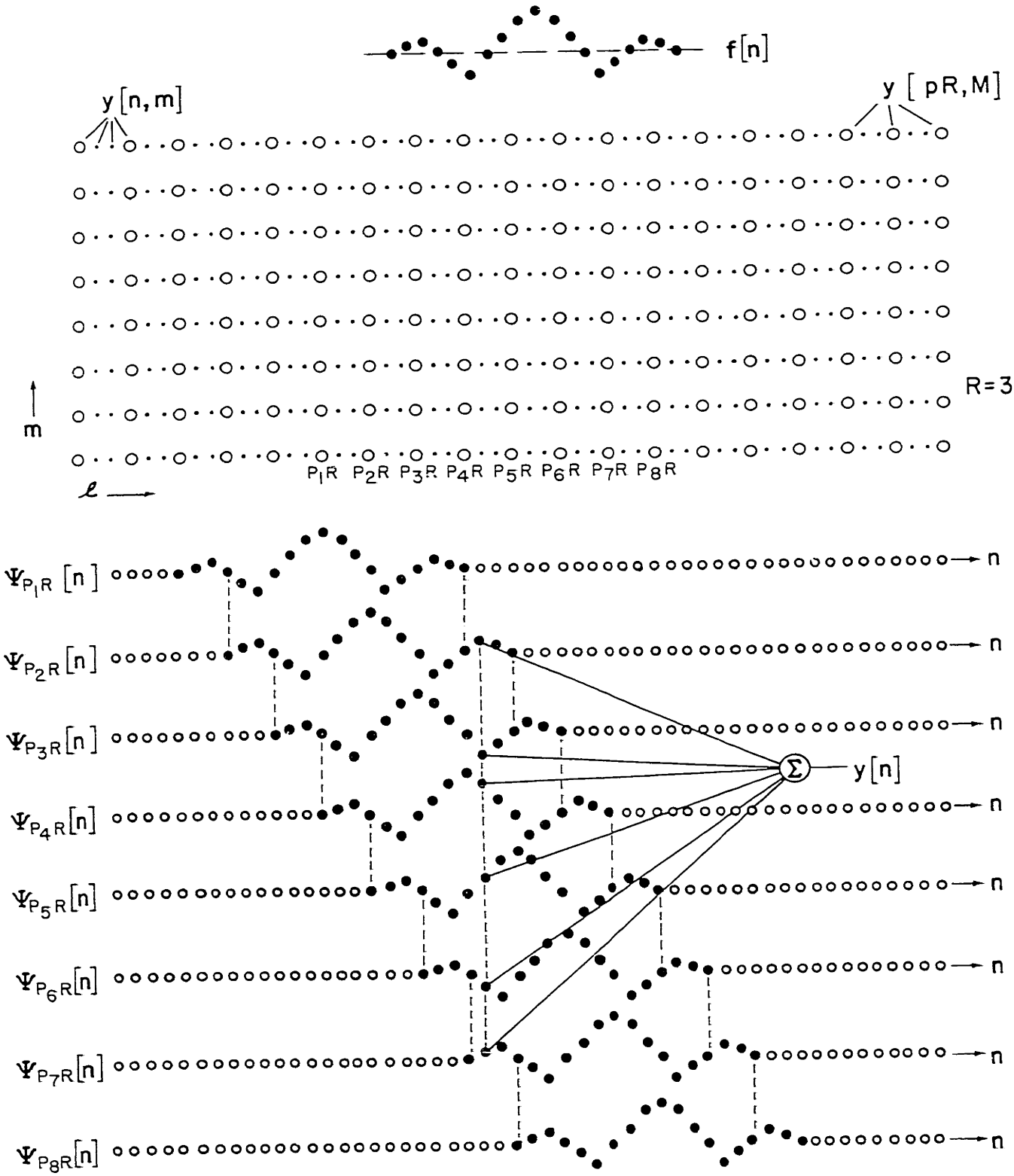


Figure 3.8

Since in practice the sequence y[n] has to start somewhere, we can define a time origin, without loss of generality. Let us assume, therefore, that y[n] begins at n = QR + 1. From this fact, and referring to equation (3.52), we can deduce that $\psi_{pR}[n]$ will equal zero, for all values of n, when p is less than zero. Thus, equation (3.53) becomes:

$$y_p[n] = \sum_{p=0}^{P} \psi_{pR}[n] \tag{3.54}$$

The sequence $y_p[n]$ can then be formed recursively:

$$y_p[n] = \begin{cases} 0 & , \text{ for } p < 0 \\ y_{p-1}[n] + \psi_{pR}[n] & , \text{ for } p \geq 0 \end{cases} \tag{3.55}$$

Equations (3.52) and (3.55) show that $\psi_{pR}[n]$ will only contribute to $y_p[n]$ for values of n greater than or equal to pR - RQ. Therefore, every time p increases by one, there are R points of $y_p[n]$ that will receive no further contributions from future frames of y[pR,m]. Consequently, these are completely formed points of the sequence y[n]. This fact can be seen in figure 3.8.

The sequence $y_p[n]$ is, therefore, a partially formed version of y[n]. It is constructed by overlap-adding the individual frame contributions $\psi_{pR}[n]$. The complete output sequence y[n] can thus

Table III

DSTFT Overlap-Add Synthesis Algorithm

Let $y_{-1}[n] = 0$, for all integers n. (3.57)

For p = 0 to ∞:

1. $y[pR,((n))_M] = IDFT\{Y_s[pR,k]\}$ (3.58)

2. $\psi_{pR}[n] = f[n-pR]y[pR,((n))_M]$ (3.59)

3. $y_p[n] = y_{p-1}[n] + \psi_{pR}[n]$ (3.60)

4. The leftmost pR points of $y_p[n]$ are completely formed points of y[n].

be obtained from $y_p[n]$ as p increases. Specifically:

$$y[n] = y_p[n] \quad , \quad \text{for } n \leq p - QR + 1 \tag{3.56}$$

In summary, the DSTFT overlap-add synthesis algorithm generates the sequence, y[n], from its decimated DSTFT representation, as shown in Table III.

In a computer implementation, $y_p[n]$ can consist of a finite length buffer that is initialized to zero (equation (3.57)) and whose contents, at each iteration on p, are replaced by the sum of its previous contents and the sequence $\psi_{pR}[n]$ (equation (3.60)). The buffer can then be shifted to the left by R points, and those R points that fall off its left end can be output as the next R points of the sequence y[n].

Figure 3.9 is a flowchart description of the DSTFT overlap-add synthesis algorithm. At any given time, only the frame of y[pR,m] for which $\psi_{pR}[n]$ is being computed has to be in main memory. In comparison, the procedure described by Portnoff [1976, 1978] requires about 20 such frames.

3.3.3 The Analysis/Synthesis System

The analysis and synthesis procedures described in the previous two sections constitute an efficient implementation of the Discrete Short-Time Fourier Transform. Without any modification between the analysis and synthesis stages, and as long as the constraint imposed on f[n], h[n] and M by equation (3.25) is

DSTFT Overlap-Add Synthesis Algorithm

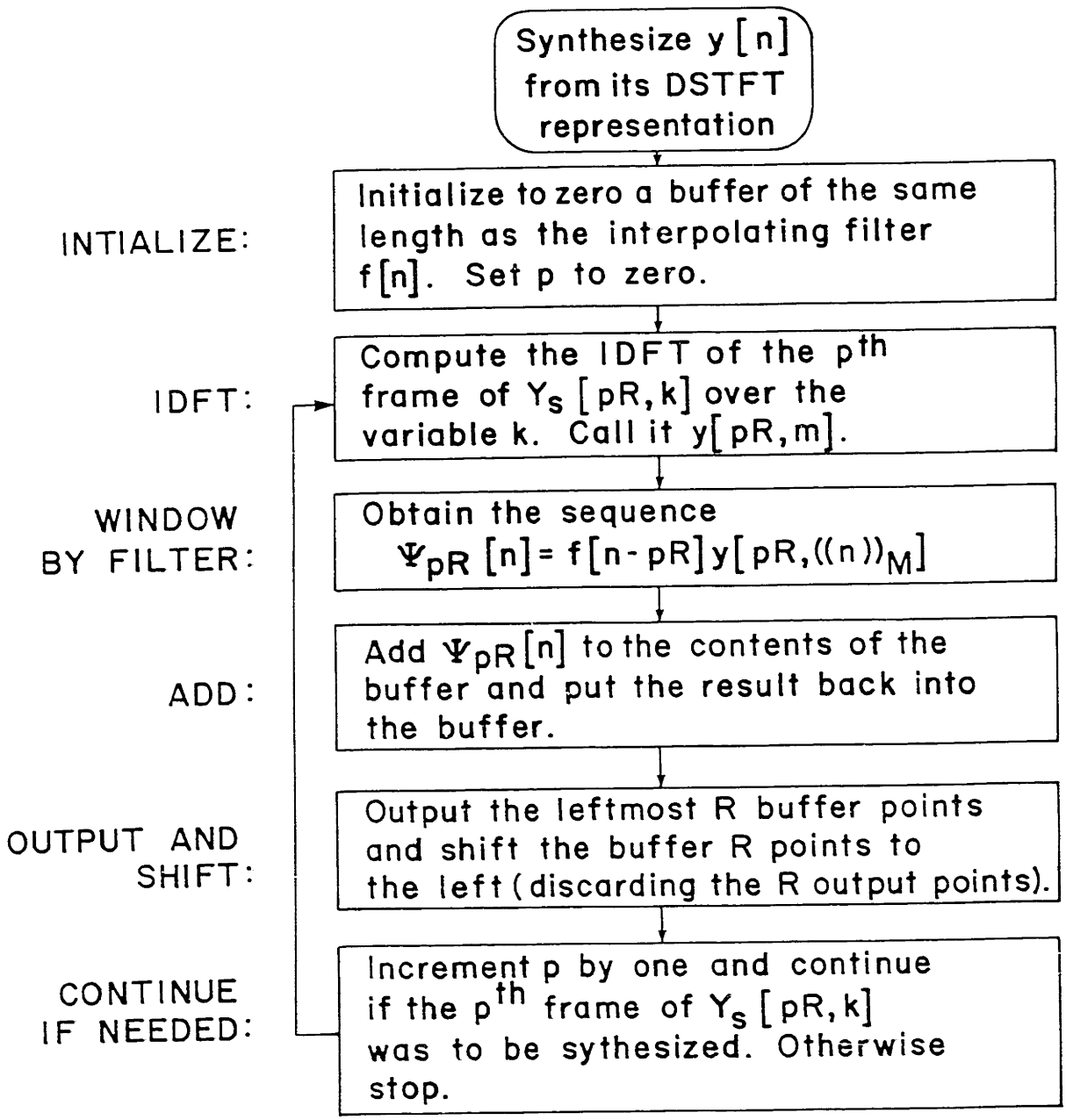


Figure 3.9

satisfied, the analysis/synthesis system is an identity system. That is, y[n] equals x[n] up to quantization and computational errors which, experimentally, have been shown to be negligibly small.

It is interesting and useful, however, to modify the sequence $x_s[pR,k]$ in some form, and to synthesize y[n] from the modified sequence $Y_s[pR,k]$. One possible modification consists of coding $X_s[pR,k]$ at a low bit rate for bandwidth compression. Schafer and Rabiner [1973(a)] have studied various coding schemes for this purpose. Bandwidth compression of this kind is based on the empirical result that the ear is less sensitive to quantization degradation when the quantization occurs in the frequency domain than when it occurs in the time domain.

Another possible way of modifying $X_s[pR,k]$ is to generate a sequence $Y_s[pR,k]$ that is approximately equal to the decimated DSTFT of x[n] had it been originally spoken more slowly or more rapidly. This modification is the subject of the following chapter.

CHAPTER 4

TIME-SCALE MODIFICATION OF SPEECH BASED ON THE DSTFT

A time-scale modification system is described in this chapter. From Chapter 2, we know that a TSM system must perform four operations on the speech signal x[n] to obtain the time-scale modified sequence $x^{\beta}[n]$. These four operations, which are carried out in stages, are referred to as Analysis, Linear Time-Scaling, Phase Modification and Synthesis. The Analysis stage estimates the time-varying values of the parameters which specify the model of speech developed in Chapter 2. The time variation of these parameters is then linearly time-scaled by the second stage of the system (Linear Time-Scaling). This operation produces the parametric representation of the sequence x[βn]. The third stage, Phase Modification, estimates the time-unwrapped phase of x[βn] and divides it by β, yielding the parametric representation of the desired time-scale modified sequence $x^{\beta}[n]$. The actual sequence $x^{\beta}[n]$ is then synthesized from its parameters by the fourth and final stage of the TSM system (Synthesis).

Portnoff [1978] has shown that the DSTFT representation of a speech signal x[n] constitutes an estimate of the values of the parameters in the speech model. In fact, if the analysis filter h[n] and the number M of frequency samples are appropriately chosen, this estimate is excellent. Therefore, the analysis and synthesis stages of the TSM system are efficiently implemented by means of

the DSTFT analysis and synthesis algorithms developed in Chapter 3.

Section 4.1 interprets the DSTFT representation of a speech signal x[n] in terms of the parameters of the speech. Both the estimation of the values of these parameters (analysis) and the generation of the signal from its parametric description (synthesis) are discussed.

The linear time-scaling stage is described in Section 4.2. This stage is implemented as a bandlimited decimation/interpolation operation, based on the technique developed by Schafer and Rabiner [1973(b)].

Section 4.3 describes the phase modification stage. The bound imposed on the growth of the time-unwrapped phase, $\phi[n]$, serves to develop an algorithm to estimate $\phi[n]$ from its principal value. The estimate of the time-unwrapped phase is then divided by β, thus completing the calculation of the parametric description of $x^{\beta}[n]$.

A somewhat surprising result is presented in Section 4.4. The decimation and interpolation steps which occur, respectively, within the DSTFT analysis and synthesis algorithms can be used to implicitly carry out the linear time-scaling operation described in Section 4.2. Thus, the explicit linear time-scaling stage can be effectively eliminated.

This finding has two important consequences. First, the need to perform a bandlimited interpolation to obtain $x[\beta n]$ from $x[n]$ no longer exists. Thus, the only interpolation operation remaining in the TSM system is the one that occurs as part of the sysnthesis

algorithm. In Chapter 3 we went to great lengths to reduce the storage needs of this interpolation. Therefore, implicit implementation of the linear time-scaling stage significantly reduces the storage requirements of the TSM system, while at the same time it increases its computational efficiency. In Portnoff's system [1978], the linear time-scaling was performed explicitly, and the synthesis was performed with the storage inefficient algorithm described in figure 3.7. The overall storage needed by the TSM system described here is about 50 times smaller than the storage needed by Portnoff's system (assuming interpolating filters of order 12).

The second consequence of implicit linear time-scaling is that the resulting structure of the TSM system can be easily modified to allow the TSM rate to vary at runtime. The advantages of a variable TSM rate were discussed in Section 1.3.

Section 4.5 presents a variable rate (non-uniform) TSM system, based on the uniform system described in Sections 4.1 - 4.4. The non-uniform TSM system is developed by altering the structure of the uniform TSM algorithm with an implicit linear time-scaling stage.

Section 4.6 describes the implementation of a non-uniform TSM system and, finally, Section 4.7 compares the system developed in this chapter with Portnoff's system.

As discussed in Chapter 2, the processing required to time-scale modify the unvoiced segments of the speech sequence x[n] is the same as that used during its voiced segments. Therefore, x[n] will be assumed to consist solely of voiced segments.

4.1 Interpretation of the DSTFT Of Voiced Speech (TSM Analysis and Synthesis)

The DSTFT representation $X_s[n,k]$, of a sequence $x[n]$ windowed by $h[n]$ is given by equation (3.17), which is repeated here for convenience:

$$X_s[n,k] = \sum_{m=-\infty}^{+\infty} x[m]h[n-m]e^{-j\Omega_0 km} \tag{4.1}$$

The speech sequence $x[n]$ can, in turn, be modeled as a sum of harmonically related complex exponentials. In Chapter 2, we described this model in equation (2.20):

$$x[n] = \sum_{r=0}^{p[n]-1} c_r[n]e^{jr\phi[n]} \tag{4.2}$$

The parameters in this model represent specific speech features. The sequence $p[n]$ is the time-varying pitch period of $x[n]$. The sequences $c_r[n]$ are lumped parameters, which describe the frequency response of the vocal tract. Finally, $\phi[n]$ is the time-unwrapped phase of the fundamental of the voiced excitation of $x[n]$.

Replacing $x[n]$ in equation (4.1) by its harmonic representation expressed in equation (4.2), we have:

$$X_s[n,k] = \sum_{m=-\infty}^{+\infty} \sum_{r=0}^{p[m]-1} c_r[m]e^{jr\phi[m]}h[n-m]e^{-jk\Omega_0 m} \tag{4.3}$$

In Chapter 3, we determined that the length of h[n] must be short enough to ensure that X[n,k] contains information about x[n] only in the vicinity of the point n. Therefore, we can assume that p[n] is constant for the duration of h[n]. Equation (4.3) can then be rewritten with p[m] replaced by p[n]. For the same reason, $\phi[m]$ can be replaced by its local approximation, given by equation (2.6). Equation (4.3) then becomes:

$$X_s[n,k] = \sum_{m=-\infty}^{+\infty} \sum_{r=0}^{p[n]-1} c_r[m] e^{jr(\phi[n]+(m-n)\Omega[n])} h[n-m] e^{-jk\Omega_0 m} \qquad (4.4)$$

Interchanging the order of summation and rearranging terms:

$$X_s[n,k] = \sum_{r=0}^{p[n]-1} \sum_{m=-\infty}^{+\infty} c_r[m] h[n-m] e^{jr(\phi[n]-n\Omega[n]+m\Omega[n])-jk\Omega_0 m}$$

$$= \sum_{r=0}^{p[n]-1} e^{jr(\phi[n]-n\Omega[n])} \cdot \left\{ \sum_{m=-\infty}^{+\infty} c_r[m] h[n-m] e^{jm(r\Omega[n]-k\Omega_0)} \right\} \qquad (4.5)$$

Referring to equation (3.1), we may recognize the term in brackets in equation (4.5) as the DSTFT of the sequence $c_r[n]$, windowed with h[n], with the frequency variable ω evaluated at $\omega = k\Omega_0 - r\Omega[n]$. This particular DSTFT analysis equation has the property that the bandwidth of the sequence $c_r[n]$ is significantly narrower than the bandwidth of h[n].

The difference in the bandwidths of $c_r[n]$ and $h[n]$ can be derived as follows. First, we saw in equation (2.21) that the complex harmonic amplitudes $c_r[n]$ of speech contain non-negligible frequency components only up to about ten Hertz. Second, for the pitch period $p[n]$ to be nearly constant under the window $h[n]$, the length of $h[n]$ cannot correspond in real time to more than about 30 milliseconds. For a Hamming window, this maximum length implies that the cutoff frequency of its spectrum cannot lie below about 70 Hertz. We will consider a 10 Hertz signal to be narrow-band compared to a 70 Hertz low-pass filter.

The large difference between the bandwidths of $c_r[n]$ and $h[n]$ can be exploited to simplify equation (4.5). Portnoff [1978] has shown that if a signal $z[n]$ is narrow-band compared to a filter $h[n]$, then the STFT of $z[n]$ windowed by $h[n]$ is given by:

$$Z_s[n,\omega] = z[n]H(\omega_0-\omega)\,e^{-j\omega n} + \varepsilon \qquad (4.6)$$

where: ω_0 is the center frequency of the spectrum of $z[n]$,

$H(\omega)$ is the Fourier transform of $h[n]$

and: ε is an error term that approaches zero as the spectrum of $z[n]$ approaches an impulse and $h[n]$ approaches an ideal low-pass filter.

The term in brackets in equation (4.5) can therefore be simplified by the result in equation (4.6), where we can assume that the error term ε is negligible. Equation (4.5) becomes:

$$X_s[n,k] = \sum_{r=0}^{p[n]-1} e^{jr(\phi[n]-n\Omega[n])} \cdot \left\{ c_r[n]H[0-(k\Omega_0-r\Omega[n])]e^{-j(k\Omega_0-r\Omega[n])n} \right.$$

(4.7)

The terms $e^{jrn\Omega[n]}$ and $e^{-jrn\Omega[n]}$ conveniently cancel in equation (4.7) which can be rewritten:

$$X_s[n,k] = \sum_{r=0}^{p[n]-1} c_r[n]H[r\Omega[n]-k\Omega_0]e^{j(r\phi[n]-kn\Omega_0)} \qquad (4.8)$$

Recall that the index k in $X_s[n,k]$ represents the continuous frequency ω sampled at intervals of Ω_0. A useful interpretation of $X_s[n,k]$ can be obtained by rewriting equation (4.8) in terms of the continuous frequency variable ω. The STFT of the speech signal x[n] can then be written by replacing $k\Omega_0$ by ω in equation (4.8):

$$X_s[n,\omega] = \sum_{r=0}^{p[n]-1} c_r[n]H[r\Omega[n]-\omega]e^{j(r\phi[n]-n\omega)} \qquad (4.9)$$

Equation (4.9) expresses the STFT of x[n] as a sum of p[n] images of $H(\omega)$, each shifted in frequency by $r\Omega[n]$ and scaled by $c_r[n]e^{j(r\phi[n]-n\omega)}$. In general, there will be some overlapping of the shifted images of $H(\omega)$. However, if the bandwidth of $H(\omega)$ is less than the pitch frequency $\Omega[n]$, then these images do not overlap.

For most speakers, the value of the pitch frequency rarely falls below 100 Hz. When we compared the bandwidths of $c_r[n]$ and $h[n]$, we found that the bandwidth of $H(\omega)$ must be greater than 70 Hertz. Therefore, if $H(\omega)$ is chosen to have a bandwidth between 70 and 100 Hertz, then equation (4.9) takes the form:

$$X_s[n,\omega] = \begin{cases} c_r[n]H\left[r\Omega[n]-\omega\right]e^{j(r\phi[n]-n\omega)} & , \text{ for } \left|r\Omega[n]-\omega\right| < \omega_h \\ 0 & , \text{ otherwise} \end{cases} \tag{4.10}$$

where ω_h denots the cutoff frequency of $H(\omega)$, and r varies between 0 and $p[n]-1$.

The DSTFT of $x[n]$ can then be obtained from equation (4.10) by sampling ω with a period Ω_0:

$$X_s[n,k] = \begin{cases} c_r[n]H\left[r\Omega[n]-k\Omega_0\right]e^{j(r\phi[n]-kn\Omega_0)} & , \text{ for } \left|r\Omega[n]-k\Omega_0\right| < \omega_h \\ 0 & , \text{ otherwise} \end{cases} \tag{4.11}$$

Equation (4.11) can be simplified further by assuming that $H(\omega)$ is close to the spectrum of an ideal low-pass filter. Therefore, $H(\omega)$ is equal to 1 over its entire non-zero region. Replacing $H\left[r\Omega[n]-k\Omega_0\right]$ by 1 in equation (4.11), we have:

$$X_s[n,k] = \begin{cases} c_r[n]e^{j(r\phi[n]-kn\Omega_0)} & , \text{ for } |r\Omega[n]-k\Omega_0| < \omega_h \\ 0 & , \text{ otherwise} \end{cases} \tag{4.12}$$

The DSTFT of x[n], as expressed in equation (4.12), is very similar to the speech model given by equation (4.2). In fact, by appropriately selecting the bandwidth of $H(\omega)$, and if Ω_0 is less than this bandwidth, then the DSTFT analysis equation separates the terms of the summation in equation (4.2), and multiplies them by a known constant, $e^{-jkn\Omega_0}$. Therefore, the sequence $X_s[n,k]$ is an estimate of the time-varying parameters in the model of speech given in equation (4.2). The TSM analysis stage can thus be implemented using the DSTFT analysis algorithm described in Chapter 3.

Let us now assume that the speech parameters have been modified, and that the sequence $Y_s[n,k]$ denotes the modified transform from which the time-scale modified speech, $y[n] = x^{\beta}[n]$, is to be synthesized. For simplicity, let us assume that the frequency variable is continuous. From equation (4.10), $Y_s[n,\omega]$ can be written as:

$$Y_s[n,\omega] = \begin{cases} c_r[\beta n]H[r\Omega[\beta n]-\omega]e^{j(r\phi[\beta n]/\beta-\omega n)} & , \text{ for } |r\Omega[\beta n]-\omega| < \omega_h \\ 0 & , \text{ otherwise} \end{cases} \tag{4.13}$$

where ω_h is the cutoff frequency of $H(\omega)$, and r varies between 0 and $p[\beta n]-1$.

Since the shifted images of $H(\omega)$ do not overlap:

$$Y_s[n,\omega] = \sum_{r=0}^{p[\beta n]-1} c_r[\beta n] H\left[r\Omega[\beta n]-\omega\right] e^{j(r\phi[\beta n]/\beta-\omega n)} \tag{4.14}$$

The sequence $y[n]$ is synthesized from its STFT by means of equation (3.15):

$$y[n] = \frac{1}{2\pi} \int_{-\pi}^{\pi} \sum_{m=-\infty}^{+\infty} f[n-m] Y_s[m,\omega] e^{j\omega n} \tag{4.15}$$

Replacing $Y_s[n,\omega]$ by its description in terms of speech parameters, given by equation (4.14), interchanging the order of summation and integration, and rearranging terms:

$$y[n] = \sum_{m=-\infty}^{+\infty} \sum_{r=0}^{p[\beta m]-1} f[n-m] c_r[\beta m] e^{jr\phi[\beta m]/\beta}$$

$$\cdot \frac{1}{2\pi} \int_{-\pi}^{\pi} H\left[r\Omega[\beta m]-\omega\right] e^{-j\omega(m-n)} d\omega \tag{4.16}$$

The integral over ω in equation (4.16) is the inverse Fourier transform of $H\left[r\Omega[\beta m]-\omega\right] e^{-j\omega m}$. This integral can be evaluated using the shift properties of the Fourier transform [Oppenheim and Schafer, 1975]:

$$\frac{1}{2\pi}\int_{-\pi}^{\pi} H\left[r\Omega[\beta m]-\omega\right]e^{-j\omega(m-n)}\,d\omega = h[m-n]e^{-j(m-n)r\Omega[\beta m]} \tag{4.17}$$

Replacing the integral in equation (4.16) by its value in equation (4.17), we have:

$$y[n] = \sum_{m=-\infty}^{+\infty}\ \sum_{r=0}^{p[\beta m]-1} f[n-m]c_r[\beta m]h[m-n]e^{jr(\phi[\beta m]/\beta-(m-n)\Omega[\beta n])} \tag{4.18}$$

The exponent in equation (4.18) is the local representation of $\phi[\beta n]/\beta$, as expressed by equation (2.6). Since h[n] is chosen to be short enough so that p[n] can be assumed to be constant under the window, we can substitute $\phi[\beta n]/\beta$ for its local representation in equation (4.18), and we can replace p[βm] by p[βn].

$$y[n] = \sum_{m=-\infty}^{+\infty}\ \sum_{r=0}^{p[\beta n]-1} f[n-m]c_r[\beta m]h[m-n]e^{jr\phi[\beta n]/\beta} \tag{4.19}$$

Interchanging the order of summation, and rearranging terms:

$$y[n] = \sum_{r=0}^{p[\beta n]-1}\left\{\sum_{m=-\infty}^{+\infty} c_r[\beta m]f[n-m]h[-(n-m)]\right\}e^{jr\phi[\beta n]/\beta} \tag{4.20}$$

The expression in brackets in equation (4.20) is the convolution of $c_r[\beta n]$ with the composite filter f[n]h[-n]. Since h[n] and f[n] are both low-pass filters, the bandwidth of the composite

filter is of the order of the sum of the bandwidths of the individual filters. The bandwidth of h[n] has been chosen to be significantly larger than the highest frequency component of $c_r[n]$. Therefore, the bandwidth of the composite filter f[n]h[-n] must be much greater than the highest frequency in $c_r[\beta n]$. Thus, the sequence $c_r[\beta n]$ is passed by the composite filter with negligible distortion. Equation (4.20) can therefore be rewritten with the expression in brackets replaced by $c_r[\beta n]$:

$$y[n] = \sum_{r=0}^{p[\beta n]-1} c_r[\beta n] e^{jr\phi[\beta n]/\beta} \tag{4.21}$$

Equation (4.21) is identical to equation (2.44). Therefore:

$$y[n] = x^{\beta}[n] \tag{4.22}$$

The TSM synthesis stage can thus be implemented using the DSTFT synthesis algorithm developed in Chapter 3.

4.2 Linear Time-Scaling

Equation (4.12) expresses $X_s[n,k]$ as an estimate of the time-varying parameters of the speech signal x[n]. Having estimated these parameters, the next stage of the TSM system must linearly time-scale the time variation of $X_s[n,k]$ by β, to obtain the new sequence $X_s[\beta n,k]$. Referring to equation (4.12), $X_s[\beta n,k]$ can be written in terms of the time-varying speech parameters:

$$X_s[\beta n,k] = \begin{cases} c_r[\beta n]\, e^{j(r\phi[\beta n]-k\beta n\Omega_0)} & , \text{ for } |r\Omega[\beta n]-k\Omega_0| < \omega_h \\ 0 & , \text{ otherwise} \end{cases} \tag{4.23}$$

In Chapter 2, β was restricted to be a rational number. Although, in general, β can be any real number, we justified this restriction by the fact that a real number can always be approximated by a rational one with arbitrary precision. If β is rational, it can be written as the ratio of two intergers:

$$\beta = D/I \tag{4.24}$$

Schafer and Rabiner [1973(b)] have shown that, if z[n] is an appropriately bandlimited signal, then z[βn] can be obtained from z[n] by the decimation/interpolation formula:

$$z[\beta n] = \sum_{r=-\infty}^{+\infty} g[nD-rI]\, z[r] \tag{4.25}$$

where g[n] is a 1-to-I interpolating FIR filter of order Q.

We can use equations (3.30) and (3.31) to replace the infinite summation limits in equation (4.25) by finite ones:

Let
$$\begin{cases} L^{+}[n] = \lceil n/I \rceil + Q & (4.26) \\ L^{-}[n] = \lceil n/I \rceil - Q + 1 & (4.27) \end{cases}$$

where $\lceil a \rceil$ denotes the largest integer that is less than or equal to a.

Then:

$$z[\beta n] = \sum_{r=L^{-}[n]}^{L^{+}[n]} g[nD-rI]z[r] \qquad (4.28)$$

Equation (4.28) can be used to obtain $X_s[\beta n,k]$ from $X_s[n,k]$. Referring to the filter bank interpretation of the DSTFT (figure 3.3), we may recall that the (one-dimensional) time sequence $X_s[n,\underline{k}]$ is bandlimited because it is the output of the low-pass filter h[n]. Since $X_s[n,\underline{k}]$ is bandlimited, it can be linearly time-scaled with the formula given in equation (4.28). Consequently:

$$X_s[\beta n,\underline{k}] = \sum_{r=L^{-}[n]}^{L^{+}[n]} g[nD-rI]X_s[n,\underline{k}] \qquad (4.29)$$

In practice, the output of the DSTFT analysis alogrithm consists of the sequence $X_s[n,k]$ in terms of its real and imaginary parts:

$$X_s[n,k] = \mathrm{Re}\{X_s[n,k]\} + j\ \mathrm{Im}\{X_s[n,k]\} \qquad (4.30)$$

Therefore, equation (4.29) is implemented as two parallel equations:

$$\mathrm{Re}\{X_s[\beta n,\underline{k}]\} = \sum_{r=L^-[n]}^{L^+[n]} g[nD-rI]\ \mathrm{Re}\{X_s[n,\underline{k}]\} \tag{4.31}$$

$$\mathrm{Im}\{X_s[\beta n,\underline{k}]\} = \sum_{r=L^-[n]}^{L^+[n]} g[nD-rI]\ \mathrm{Im}\{X_s[n,\underline{k}]\} \tag{4.32}$$

$$\text{where:}\quad X_s[\beta n,\underline{k}] = \mathrm{Re}\{X_s[\beta n,\underline{k}]\} + j\ \mathrm{Im}\{X_s[\beta n,\underline{k}]\} \tag{4.33}$$

The linear time-scaling stage of the TSM system can be implemented by applying equations (4.31) and (4.32) to $X_s[n,\underline{k}]$ for each of the M values of k. However, note that for equation (4.29) to hold, the decimation rate D cannot be arbitrarily large. In fact, D must be chosen to be small enough so that the sequences $X_s[Dn,\underline{k}]$ are not degraded by frequency domain aliasing.

The actual bound on the size of D is simple to obtain. Let us assume that the original continuous-time speech signal x(t) contains frequencies up to f_x = 5 KHz. By design, the analysis filter h[n] has a cutoff frequency, f_h, that lies between 70 and 100 Hertz. The sequence x[n] is assumed to be a discrete-time version of x(t), sampled at a rate f_s, which is greater than the Nyquist rate of x(t). $X_s[n,k]$ is sampled as often as x[n], but it contains frequencies only up to $\omega_h = 2\pi f_h/f_s$. The Sampling Theorem states that X[n,k] can be decimated at a rate D that must satisfy the constraint:

$$D < \frac{f_s}{2f_h} \tag{4.34}$$

In terms of the time-scale modification system, this restriction implies that the signal x[n] cannot be compressed at an arbitrary rate. In fact, in Section 4.3, the decimation rate D will be restricted to lie below $f_s/2.5\ f_h$. However, this restriction is not very significant because it places a bound on the compression rate that lies well beyond the point where the speech becomes unintelligibly fast. The number $f_s/2.5\ f_h$ is at least equal to 40*. Compressing speech more than 40 times is basically a useless operation.

In this section, we developed a technique to explicitly implement the linear time-scaling stage of the TSM system. In Section 4.4, however, we will show that $X_s[n,k]$ can be linearly time-scaled without explicitly carrying out the decimation/interpolation operation. The implicit implementation of the linear time-scaling stage will increase the computational efficiency of the TSM algorithm, and will significantly reduce its storage requirements. More importantly, implicit linear time-scaling will allow the TSM system to be modified in order to make the scale factor β variable at runtime.

4.3 Phase Modification

Equation (4.23) expresses the linearly time-scaled DSTFT sequence $X_s[\beta n,k]$ in terms of speech parameters. It is useful to analyze $X_s[\beta n,k]$ with a fixed value of the index k for which the

---

*10000/(2.5 x 100) = 40

sequence does not vanish. In this case, $X_s[\beta n,\underline{k}]$ can be reduced to:

$$X_s[\beta n,\underline{k}] = c_r[\beta n]e^{j(r\phi[\beta n]-\underline{k}\beta n\Omega_0)} \tag{4.35}$$

for some value of r between 0 and $p[\beta n]-1$

The phase modification stage of the TSM system must first estimate the time-unwrapped phase of $X_s[\beta n,\underline{k}]$. Then, it must divide this phase by β, and substitute the modified phase value for the unmodified value in $X_s[\beta n,\underline{k}]$. In this way, the sequence $Y_s[n,\underline{k}] = X_s^{\beta}[n,\underline{k}]$ is generated.

When the value of k is such that $X_s[\beta n,\underline{k}]$ vanishes, the phase of the sequence $X_s[\beta n,\underline{k}]$ is not defined. Experimental results show, however, that the phase modification alogrithm should treat $X_s[\beta n,\underline{k}]$ as if it did not vanish. This result is a consequence of the fact that the magnitude of the vanishing $X_s[\beta n,\underline{k}]$ is negligibly small. Therefore, any discontinuity that might appear in the estimated time-unwrapped phase of $Y_s[n,\underline{k}]$ will occur when its magnitude vanishes and, thus, when its contribution to y[n] is imperceptible. We shall therefore assume that the sequence $X_s[\beta n,\underline{k}]$ can be expressed by equation (4.35) for all values of k.

The linearly time-scaled DSTFT of x[n], expressed in equation (4.35), is a complex quantity, which can be rewritten in polar form:

$$X_s[\beta n,\underline{k}] = M[\beta n,\underline{k}]e^{j\Theta[\beta n,\underline{k}]} \tag{4.36}$$

where: $$M[\beta n,\underline{k}] = |X_s[\beta n,\underline{k}]| \tag{4.37}$$

and: $$\Theta[\beta n,\underline{k}] = \arg\{X_s[\beta n,\underline{k}]\} \tag{4.38}$$

The magnitude and phase of $X_s[\beta n,k]$ can be obtained from equation (4.35):

$$M[\beta n,\underline{k}] = |c_r[\beta n]| \tag{4.39}$$

$$\Theta[\beta n,\underline{k}] = \arg\{c_r[\beta n]\} + r\phi[\beta n] - \underline{k}\beta n\Omega_0 \tag{4.40}$$

The phase of $X_s[\beta n,\underline{k}]$ is equal to the phase of $c_r[\beta n]$ plus the term $r\phi[\beta n] - \underline{k}\beta n\Omega_0$. We can write $\Theta[\beta n,\underline{k}]$ explicitly as the sum of its two components:

$$\Theta[\beta n,\underline{k}] = \arg\{c_r[\beta n]\} + \zeta[\beta n,\underline{k}] \tag{4.41}$$

where: $$\zeta[\beta n,\underline{k}] = r\phi[\beta n] - \underline{k}\beta n\Omega_0 \tag{4.42}$$

Portnoff [1978] has shown that each complex harmonic amplitude, $c_r[\beta n]$, is a slowly varying function of n, whose phase is also slowly varying. In contrast, the sequence $\zeta[\beta n,\underline{k}]$ varies rapidly as a function of n, as can be seen in equation (4.42). The difference

in the variability of $\arg\{c_r[\beta n]\}$ and $\zeta[\beta n,\underline{k}]$ as functions of n will be used later to estimate $\zeta[\beta n,\underline{k}]$ from $\Theta[\beta n,\underline{k}]$.

Now, let us restate the objective of the phase modification stage of the TSM system in terms of the DSTFT representation of the speech. Given the linearly time-scaled DSTFT of x[n], $X_s[\beta n,k]$, expressed in equation (4.35) for a fixed value of k, the phase modification stage must form the sequence:

$$Y_s[n,k] = c_r[\beta n]e^{j(r\phi[\beta n]/\beta - kn\Omega_0)} \tag{4.43}$$

The desired time-scale modified sequence, y[n], can be synthesized from $Y_s[n,k]$ with the DSTFT synthesis algorithm, as shown in equations (4.14)-(4.22).

In order to form $Y_s[n,k]$, two distinct steps must be carried out:

1. Estimation of the sequence $\zeta[\beta n,\underline{k}]$ from the values of $X_S[\beta n,\underline{k}]$, which are the output of the linear time-scaling stage.

2. Substitution of the quantity:

   $$\hat{\zeta}[\beta n,\underline{k}]/\beta$$

   for the unmodified phase of $X_S[\beta n,\underline{k}]$, where $\hat{\zeta}[\beta n,\underline{k}]$ denotes the estimate of $\zeta[\beta n,\underline{k}]$.

For the sake of clarity, the phase estimation and substitution procedures are treated separately in Subsections 4.3.1 and 4.3.2. Finally, this section concludes with a summary of the complete phase modification algorithm.

4.3.1 Estimation of the Time-Unwrapped Phase

At the output of the linear time-scaling stage, the complex sequence $X_s[\beta n,\underline{k}]$ is obtained in terms of its real and imaginary parts (equation (4.33)).

The first step in the estimation of $\zeta[\beta n,\underline{k}]$ is the conversion of $X_s[\beta n,\underline{k}]$ to polar coordinates:

$$M[\beta n,\underline{k}] = \sqrt{(\mathrm{Re}\{X_s[\beta n,\underline{k}]\})^2 + (\mathrm{Im}\{X_s[\beta n,\underline{k}]\})^2} \tag{4.44}$$

$$\theta_{PV}[\beta n,\underline{k}] = \mathrm{Arg}(\mathrm{Re}\{X_s[\beta n,\underline{k}]\},\ \mathrm{Im}\{X_s[\beta n,\underline{k}]\}) \tag{4.45}$$

The subscript "PV" denotes that $\theta_{PV}[\beta n,\underline{k}]$ is the principal value of $\theta[\beta n,\underline{k}]$. The function Arg(a,b) is defined as follows:

$$\mathrm{Arg}(a,b) = \begin{cases} \tan^{-1}(b/a) & , a > 0, \text{ for all } b \\ \pi/2 & , a = 0,\ b > 0 \\ 0 & , a = 0,\ b = 0 \\ -\pi/2 & , a = 0,\ b < 0 \\ \tan^{-1}(b/a) + \pi & , a < 0,\ b \geq 0 \\ \tan^{-1}(b/a) - \pi & , a < 0,\ b < 0 \end{cases} \tag{4.46}$$

where $\tan^{-1}(b/a)$ is assumed to be the single-valued arctangent function of b/a whose range is the open interval, $(-\pi/2,\ \pi/2)$.

The function Arg(a,b) is often supplied as a standard function in high-level mathematically oriented programming languages.

The second step in the estimation of $\zeta[\beta n,\underline{k}]$ is to obtain $\theta[\beta n,\underline{k}]$ from $\theta_{PV}[\beta n,\underline{k}]$. In other words, we need to time-unwrap the principal value phase sequence $\theta_{PV}[\beta n,\underline{k}]$. In fact, we will find that $\zeta[\beta n,k]$ can be estimated directly from $\theta_{PV}[\beta n,\underline{k}]$.

We can relate the time-unwrapped phase of $X_s[\beta n,\underline{k}]$ to its principal value phase as follows:

$$\theta_{PV}[\beta n,\underline{k}] = ((\theta[\beta n,\underline{k}]))_{2\pi} \tag{4.47}$$

where $((\alpha))_{2\pi}$ denotes modulo 2π.

Equation (4.47) can be expressed in terms of an integer function $I[\beta n,\underline{k}]$ that denotes the integer number of jumps of 2π by which $\theta[\beta n,\underline{k}]$ and $\theta_{PV}[\beta n,\underline{k}]$ differ:

$$\theta_{PV}[\beta n,\underline{k}] = \theta[\beta n,\underline{k}] - 2\pi I[\beta n,\underline{k}] \tag{4.48}$$

The first backward difference operator Δ, applied to some sequence $z[n]$, is defined as follows:

$$\Delta z[n] = z[n] - z[n-1] \tag{4.49}$$

Taking the first backward difference of both sides of equation (4.48):

$$\Delta\theta_{PV}[\beta n,\underline{k}] = \Delta\theta[\beta n,\underline{k}] - 2\pi\Delta I[\beta n,\underline{k}] \tag{4.50}$$

For simplicity, define the sequence $i[\beta n,\underline{k}]$:

$$i[\beta n,\underline{k}] = \Delta I[\beta n,k] \tag{4.51}$$

The first backward difference of an integer function is also an integer function, so $\Delta\theta_{PV}[\beta n,\underline{k}]$ differs from $\Delta\theta[\beta n,\underline{k}]$ by an integer multiple of 2π.

The term $\Delta\theta[\beta n,\underline{k}]$ in equation (4.50) has a meaningful interpretation in terms of the filter bank interpretation of the DSTFT (figure 3.3). The sequence $\theta[n,\underline{k}]$, prior to the linear time-scaling operation, is the time-unwrapped phase of $X_s[n,\underline{k}]$ which, in turn, is the output of the low-pass filter $h[n]$. If ω_h is the cutoff frequency of $h[n]$, then it is also the highest frequency of $X_s[n,\underline{k}]$. Equation (2.8) expresses the local frequency, $\Omega[n]$, of a sequence as the first backward difference of the time-unwrapped phase of that sequence. Therefore, $\Delta\theta[n,\underline{k}]$ is equal to the local frequency of $X_s[n,k]$, which cannot be greater than ω_h. Consequently, we can set a bound on the size of $\Delta\theta[n,\underline{k}]$:

$$|\Delta\theta[n,\underline{k}]| < \omega_h \tag{4.52}$$

As shown in Section 4.2, $\omega_h = 2\pi f_h/f_s$, where f_h is the effective cutoff frequency (in Hertz) of the filter $h[n]$, and f_s is the sampling rate of $x[n]$. The actual values of f_h and f_s

were also discussed in Section 4.2: the sampling rate, f_s, was assumed to be about 10 KHz, while the cutoff frequency f_h was chosen to lie between 70 and 100 Hz. Therefore:

$$0.014\pi \leq \omega_h \leq 0.02\pi \tag{4.53}$$

We shall conservatively assume that:

$$\omega_h \leq 0.025\pi \tag{4.54}$$

Combining equations (4.52) and (4.54):

$$|\Delta\theta[n,\underline{k}]| < 0.025\pi \tag{4.55a}$$

Then, assuming that β is small enough so that $\Delta\theta[\beta n,\underline{k}]$ is equal to $\Delta\theta[\beta(n-1),\underline{k}]$ -- a very safe assumption -- we can write equation (4.55a) for the linearly time-scaled phase, $\theta[\beta n,\underline{k}]$:

$$|\Delta\theta[\beta n,\underline{k}]| < 0.025\pi\beta \tag{4.55b}$$

Now, let us rewrite equation (4.50) with $\Delta I[\beta n,\underline{k}]$ replaced by $i[\beta n,\underline{k}]$:

$$\Delta\theta_{PV}[\beta n,\underline{k}] = \Delta\theta[\beta n,\underline{k}] - 2\pi i[\beta n,\underline{k}] \tag{4.56}$$

If we choose β so that $0.025\pi\beta$ is less than π, then equations (4.55b)

and (4.56) show that $\Delta\theta[\beta n,\underline{k}]$ can be estimated from $\Delta\theta_{PV}[\beta n,\underline{k}]$ simply by adding some integer multiple, $i[\beta n,\underline{k}]$, of 2π to $\Delta\theta_{PV}[\beta n,\underline{k}]$ such that:

$$\left|\Delta\theta_{PV}[\beta n,\underline{k}] + 2\pi i[\beta n,\underline{k}]\right| < \pi \tag{4.57}$$

Actually, since $\theta_{PV}[\beta n,\underline{k}]$ is a principal-value phase, it ranges between $-\pi$ and π. Therefore, $\Delta\theta_{PV}[\beta n,\underline{k}]$ must lie between -2π and 2π. This means that $i[\beta n,\underline{k}]$ can only be equal to -1, 0, or 1. Once we know the value of $i[\beta n,\underline{k}]$ we can then find $\theta[\beta n,\underline{k}]$ by the recursive relation:

$$\theta[\beta n,k] = \begin{cases} \theta_{PV}[0,\underline{k}] & , \ n = 0 \\ \theta[\beta(n-1),\underline{k}] + \Delta\theta_{PV}[\beta n,\underline{k}] + 2\pi i[\beta n,\underline{k}] & , \ n > 0 \end{cases} \tag{4.58}$$

As indicated in the beginning of this subsection, however, $\zeta[\beta n,\underline{k}]$ can be estimated directly from $\theta_{PV}[\beta n,\underline{k}]$, without the need to form $\theta[\beta n,\underline{k}]$. Equation (4.41) expresses $\theta[\beta n,\underline{k}]$ in terms of $\zeta[\beta n,\underline{k}]$. Taking the first backward difference of both sides of this equation:

$$\Delta\theta[\beta n,\underline{k}] = \Delta \arg\{c_r[\beta n]\} + \Delta\zeta[\beta n,\underline{k}] \tag{4.59}$$

As mentioned earlier, Portnoff [1978] has shown that the phase of $c_r[\beta n]$ varies much more slowly than $\zeta[\beta n,\underline{k}]$ as a function of n. Therefore

$$\Delta \arg\{c_r[\beta n]\} << \Delta\zeta[\beta n,\underline{k}] \tag{4.60}$$

Equations (4.62) and (4.63) imply that, with a negligibly small error:

$$\Delta\theta[\beta n,\underline{k}] = \Delta\zeta[\beta n,\underline{k}] \tag{4.61}$$

Combining equations (4.56) and (4.61), and rearranging terms:

$$\Delta\hat{\zeta}[\beta n,\underline{k}] = \Delta\theta_{PV}[\beta n,\underline{k}] + 2\pi i[\beta n,\underline{k}] \tag{4.62}$$

where $i[\beta n,\underline{k}]$ is determined by equation (4.57)

Finally, the estimate $\hat{\zeta}[\beta n,\underline{k}]$ of the quantity $\zeta[\beta n,\underline{k}]$ is obtained recursively as follows:

$$\hat{\zeta}[\beta n,\underline{k}] = \begin{cases} 0 & , n = 0 \\ \hat{\zeta}[\beta(n-1),\underline{k}] + \Delta\theta_{PV}[\beta n,\underline{k}] + 2\pi i[\beta n,\underline{k}] & , n > 0 \end{cases} \tag{4.63}$$

The sequence $\hat{\zeta}[\beta n,k]$ can then be constructed by obtaining $\hat{\zeta}[\beta n,\underline{k}]$ for each of the M possible values of k.

4.3.2 Phase Substitution

Once the M sequences $\hat{\zeta}[\beta n,\underline{k}]$ have been obtained, the sequence $Y_s[n,\underline{k}]$ (as expressed in equation (4.43)) can be obtained as follows:

$$|Y_s[n,\underline{k}]| = M[\beta n,\underline{k}] \tag{4.64}$$

$$\arg\{Y_s[n,\underline{k}]\} = \arg\{c_r[\beta n]\} + \zeta[\beta n,\underline{k}]/\beta \tag{4.65}$$

Equation (4.64) is implemented directly, but equation (4.65) must be expressed in terms of $\hat{\zeta}[\beta n,\underline{k}]$, and can be considerably simplified. The phase of $Y_s[n,\underline{k}]$ is obtained from $\theta[\beta n,\underline{k}]$ and $\hat{\zeta}[\beta n,\underline{k}]$ by the formula:

$$\arg\{Y_s[n,\underline{k}]\} = \theta[\beta n,\underline{k}] + \left(\frac{1}{\beta} - 1\right)\hat{\zeta}[\beta n,\underline{k}] \tag{4.66}$$

The effect of equation (4.66) can be seen by substituting the value of $\theta[\beta n,\underline{k}]$, given by equation (4.41), in equation (4.66):

$$\arg\{Y_s[\beta n,\underline{k}]\} = \arg\{c_r[\beta n]\} + \zeta[\beta n,\underline{k}] + \left(\frac{1}{\beta} - 1\right)\hat{\zeta}[\beta n,\underline{k}]$$

$$= \arg\{c_r[\beta n]\} + \hat{\zeta}[\beta n,\underline{k}]/\beta \tag{4.67}$$

Although the phase modification algorithm given by equation (4.66) can be implemented directly, $\arg\{Y_s[n,\underline{k}]\}$ can be expressed recursively in a form that does not require the estimation of $\theta[\beta n,k]$. Combining equations (4.61), (4.63) and (4.66), we have:

$$\arg\{Y_s[n,\underline{k}]\} = \theta[\beta n,\underline{k}] + \left(\frac{1}{\beta} - 1\right) \sum_{p=1}^{n} \Delta\theta[\beta p,\underline{k}] \tag{4.68a}$$

Consequently:

$$\arg\{Y_s[n-1,\underline{k}]\} = \theta[\beta(n-1),\underline{k}] + \left(\frac{1}{\beta} - 1\right) \sum_{p=1}^{n-1} \Delta\theta[\beta p,\underline{k}] \tag{4.68b}$$

Subtracting equation (4.68b) from equation (4.68a), and rearranging terms:

$$\arg\{Y_s[n,\underline{k}]\} = \arg\{Y_s[n-1,\underline{k}]\} + \Delta\theta[\beta n,\underline{k}] + \left(\frac{1}{\beta} - 1\right)\Delta\theta[\beta n,\underline{k}]$$

$$= \arg\{Y_s[n-1,\underline{k}]\} + \frac{1}{\beta}\Delta\theta[\beta n,\underline{k}] \tag{4.69}$$

Replacing $\Delta\hat{\zeta}[\beta n,\underline{k}]$ for $\Delta\theta[\beta n,\underline{k}]$ in equation (4.69):

$$\arg\{Y_s[n,\underline{k}]\} = \arg\{Y_s[n-1,\underline{k}]\} + \frac{1}{\beta}\Delta\hat{\zeta}[\beta n,\underline{k}] \tag{4.70}$$

From equations (4.58), (4.63) and (4.66) we obtain the boundary condition:

$$\arg\{Y_s[0,\underline{k}]\} = \theta_{PV}[0,\underline{k}] \tag{4.71}$$

Finally, equations (4.64), (4.70) and (4.71) provide the desired procedure for forming the sequence $Y_s[n,\underline{k}]$ from $X_s[\beta n,\underline{k}]$ and the estimate $\hat{\zeta}[\beta n,\underline{k}]$ of the unmodified time-unwrapped phase:

$$Y_s[n,\underline{k}] = |Y_s[n,\underline{k}]| e^{j \arg|Y_s[n,\underline{k}]|} \tag{4.72}$$

where:

$$|Y_s[n,\underline{k}]| = M[\beta n,\underline{k}] \tag{4.73}$$

$$\arg\{Y_s[n,\underline{k}]\} = \begin{cases} \Theta_{PV}[0,\underline{k}] & , \quad n = 0 \\ \arg\{Y_s[n-1,\underline{k}]\} + \frac{1}{\beta}\Delta\hat{\zeta}[\beta n,\underline{k}] & , \quad n > 0 \end{cases} \tag{4.74}$$

As before, the sequence $Y_s[n,k]$ is obtained by evaluating $Y_s[n,\underline{k}]$ for each of the M possible values of k. To complete the phase modification stage, $Y_s[n,k]$ must be converted to rectangular coordinates before it is given to the synthesis stage.

4.3.3 Phase Modification Algorithm

The above subsections have developed an algorithm to modify the time-unwrapped phase of $X_s[\beta n,\underline{k}]$. The complete phase modification algorithm, which generates $Y_s[n,\underline{k}]$ from $X_s[\beta n,\underline{k}]$ (both given in rectangular coordinates) is summarized here, in Table IV.

The algorithm expressed in equations (4.75) - (4.81) computes $Y_s[n,\underline{k}]$ for a single value of k. To obtain the complete sequence $Y_s[n,k]$ this algorithm should be carried out for $k = 0,1,\ldots,M-1$.

Table IV

TSM Phase Modification Algorithm

1. $$|Y_s[n,\underline{k}]| = \sqrt{(\mathrm{Re}\{X_s[\beta n,\underline{k}]\})^2 + (\mathrm{Im}\{X_s[\beta n,\underline{k}]\})^2} \tag{4.75}$$

2. $$\theta_{PV}[\beta n,\underline{k}] = \mathrm{Arg}(\mathrm{Re}\{X_s[\beta n,\underline{k}]\},\ \mathrm{Im}\{X_s[\beta n,\underline{k}]\}) \tag{4.76}$$

where Arg(a,b) is defined in equation (4.46)

3. If n equals 0, skip to step 6.

4. Set $i[\beta n,\underline{k}]$ equal to -1, 0 or 1, so that:

$$|\Delta\theta_{PV}[\beta n,\underline{k}] + 2\pi i[\beta n,\underline{k}]| < \pi \tag{4.77}$$

5. $$\Delta\hat{\zeta}[\beta n,\underline{k}] = \Delta\theta_{PV}[\beta n,\underline{k}] + 2\pi i[\beta n,\underline{k}] \tag{4.78}$$

6. $$\arg\{Y_s[n,\underline{k}]\} = \begin{cases} \theta_{PV}[0,\underline{k}] & n = 0 \\ \arg\{Y_s[n-1,\underline{k}]\} + \frac{1}{\beta}\Delta\hat{\zeta}[\beta n,\underline{k}], & n > 0 \end{cases} \tag{4.79}$$

7. $$\mathrm{Re}\{Y_s[n,\underline{k}]\} = |Y_s[n,\underline{k}]| \cdot \cos(\arg\{Y_s[n,\underline{k}\}) \tag{4.80}$$

8. $$\mathrm{Im}\{Y_s[n,\underline{k}]\} = |Y_s[n,\underline{k}]| \cdot \sin(\arg\{Y_s[n,\underline{k}]\}) \tag{4.81}$$

4.4 Implicit Linear Time-Scaling

A procedure for implementing the linear time-scaling stage of the TSM system was described in Section 4.2. This procedure consists of explicitly decimating and interpolating the M band-limited one-dimensional sequences $X_s[n,\underline{k}]$ to obtain the two-dimensional sequence $X_s[\beta n,k]$. There is, however, an alternative implementation scheme for the linear time-scaling stage which is significantly more efficient (both computationally and in terms of storage requirements) than the method described in Section 4.2. This alternative scheme is referred to as Implicit Linear Time-Scaling.

In order to present the implicit linear time-scaling method, it is useful to view the TSM system as a whole. This is done by describing each of the four TSM stages in terms of their input/output behavior, and then analyzing the structure of the system as a set of four subsystems which communicate with each other through their respective inputs and outputs.

At the front end, the analysis stage takes a one-dimensional signal x[n], a window h[n] and a decimation rate R_A, and produces the two-dimensional decimated DSTFT sequence $X_s[pR_A,k]$. At the other end, the synthesis stage takes a two-dimensional sequence $Y_s[pR_S,k]$, a Q^{th} order 1-to-R_S interpolating filter f[n] and an interpolation rate R_S, and produces the one-dimensional sequence y[n] (figure 4.1).

In this chapter, we have implicitly assumed that:

$$R_A = R_S = 1 \tag{4.82}$$

The Explicit TSM System

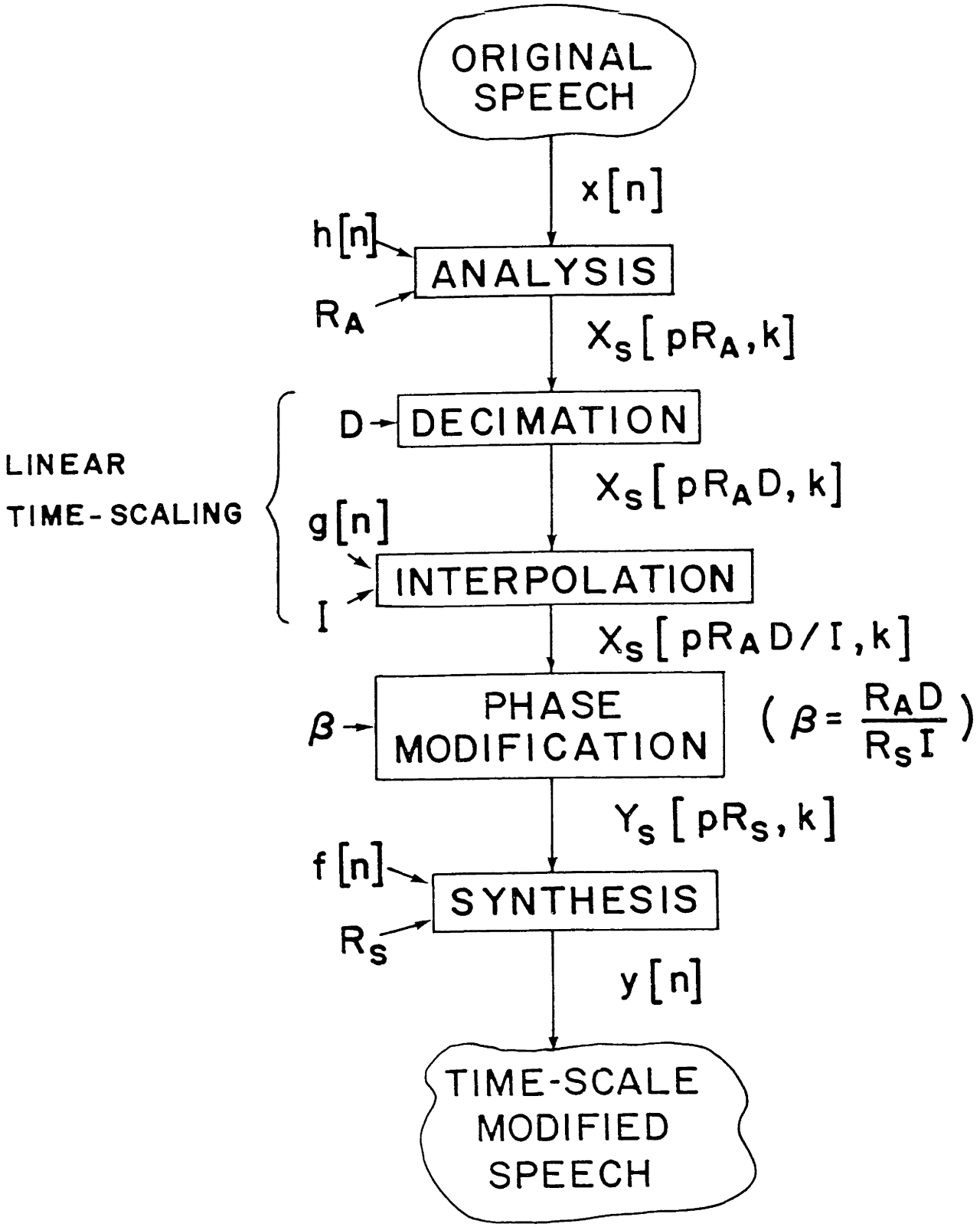


Figure 4.1

This assumption has been made only to simplify the notation and, as shown below, it is now useful to remove it.

Between the analysis and the synthesis stages, ths sequence $X_s[pR_A,k]$ is modified to form the sequence $Y_s[pR_S,k]$. This modification takes place in the linear time-scaling and phase modification stages. The linear time-scaling stage performs two operations on $X_s[pR_A,k]$: the sequence is decimated by the rate D and is interpolated by the rate I. These two operations are quite distinct although, as indicated by equation (4.29), they are carried out simultaneously. The inputs of the decimation operation are $X_s[pR_A,k]$ and the rate D, and its output is the decimated sequence $X_s[pR_AD,k]$. In turn, the interpolation operation takes $X_s[pR_AD,k]$, a Q_I^{th} order 1-to-I interpolating filter g[n] and the rate I, and produces the sequence $X_s[pR_AD/I,k]$. Therefore, the complete linear time-scaling stage generates $X_s[pR_AD/I,k]$ from $X_s[pR_A,k]$, g[n], D and I.

Finally, the phase modification stage accepts as inputs the sequence $X_s[pR_AD/I,k]$, and the value of the scale factor β, which is now given by:

$$\beta = (R_A \cdot D)/(I \cdot R_S) \tag{4.83}$$

The output of the phase modification stage is the sequence $Y_s[pR_S,k]$, from which the synthesis stage can generate the desired sequence $y[n] = x^{\beta}[n]$.

The structure of the TSM system in terms of its stages is shown in figure 4.1, where the linear time-scaling stage is shown as the cascade of two separate sub-stages: decimation and interpolation.

The implicit method of implementing the linear time-scaling stage can now be obtained from equation (4.83), by simply setting:

$$D = I = 1 \tag{4.84}$$

and letting R_A and R_S vary. In this case, equation (4.83) becomes:

$$\beta = R_A/R_S \tag{4.85}$$

The value of R_S can theoretically be any integer greater than or equal to 1, although Portnoff [1978] has shown that the value of β cannot be arbitrarily small due to the non-linear nature of the phase modification procedure. Experimentally, β can take values down to about 0.25 for speech signals. In Chapter 6, the TSM system is shown to work for music signals, in which case β can be as low as 0.1 without too much degradation.

The value of R_A is subject to the constraint imposed on D by equation 4.34:

$$R_A < f_s/2f_h \tag{4.86}$$

Therefore, R_A can range from 1 to about 40.

Equation (4.84) effectively eliminates the linear time-scaling stage by turning it into an identity. The resulting implicit TSM system is depicted in figure 4.2. We have already shown that the analysis stage can generate $X_s[pR_A,k]$ from x[n], h[n] and R_A, and that the synthesis stage can take $Y_s[pR_S,k]$, f[n] and R_S, and produce y[n]. However, the fact that $X_s[pR_A,k]$ can be transformed into $Y_s[pR_S,k]$ by the phase modification stage needs some clarification.

The input and output of the phase modification stage are given, respectively, by equations (4.35) and (4.43). Thus, the phase modification (PM) stage, given β, can be specified in operator form by the relation:

$$PM\{X_s[n,k],\beta\} = Y_s[n/\beta,k] \tag{4.87}$$

Replacing β in equation (4.87) by its value, given in equation (4.85):

$$PM\{X_s[n,k],\ R_A/R_S\} = Y_s[nR_S/R_A,k] \tag{4.88}$$

Decimating both sides of equation (4.88) along the index n at a rate R_A gives the desired result:

$$PM\{X_s[pR_A,k],\ R_A/R_S\} = Y_s[pR_S,k] \tag{4.89}$$

The Implicit TSM System

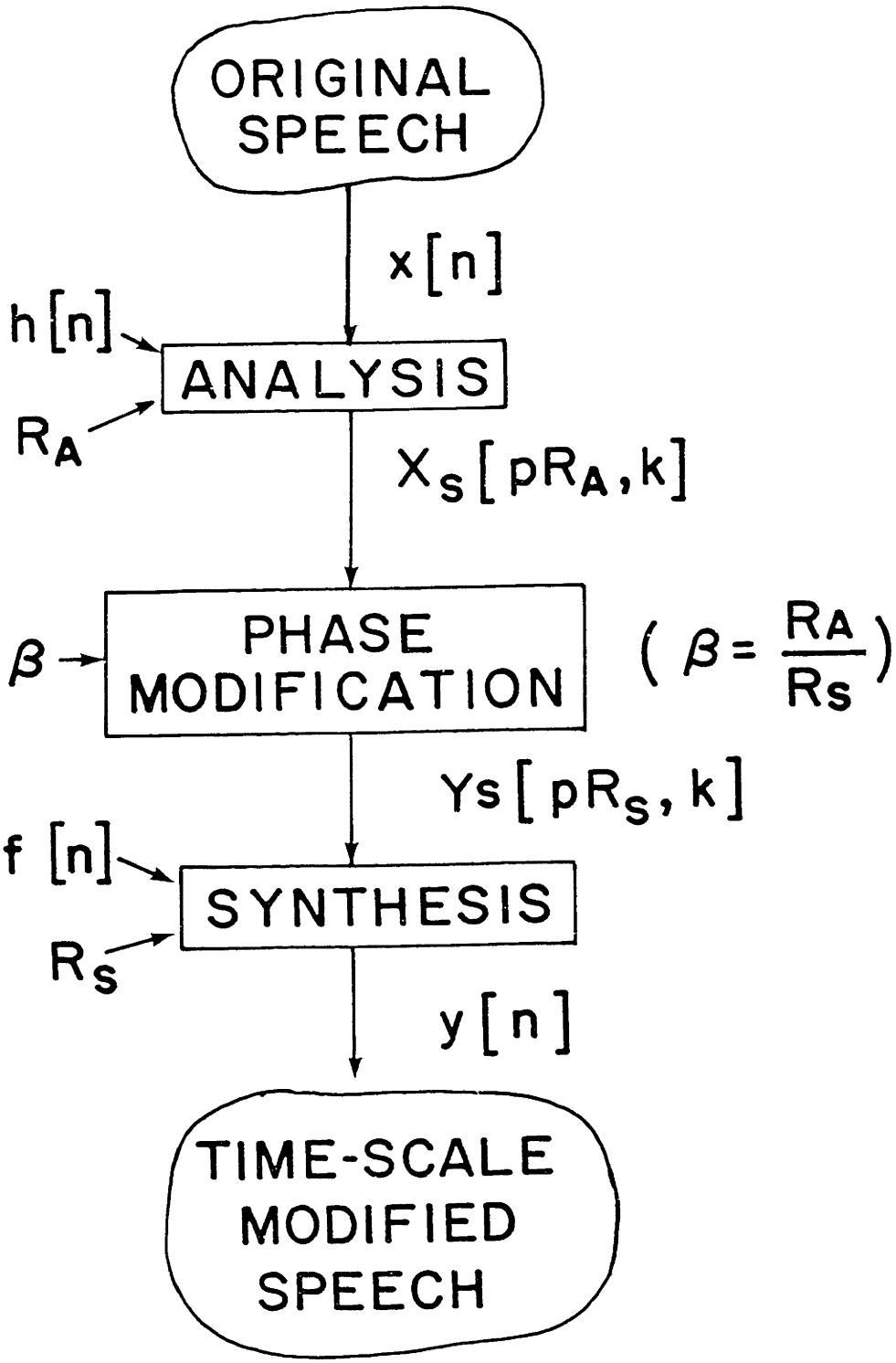


Figure 4.2

The implicit TSM system described in figure 4.2 is significantly more efficient than the explicit system shown in figure 4.1. Computationally, its increased efficiency is primarily due to the fact that the decimation stage, in the explicit TSM configuration, effectively discards D-1 out of every D spectral frames (i.e., sequences of the form $X_s[\underline{n},k]$) that are generated by the analysis stage. Therefore, most of the computational work of the analysis stage is wasted. The implicit TSM system does not waste any computations in this manner.

The elimination of the explicit interpolation stage also reduces the computational burden of the TSM system, but only by a negligible amount. In terms of storage requirements, however, the difference between explicit and implicit interpolation is very large. If g[n] is a Q_I^{th} order interpolating filter, then 2Q+1 spectral frames of $X_s[nD,k]$ are needed to perform the explicit 1-to-I interpolation. A typical value of Q is 12 and, as demonstrated by Portnoff [1978], the number M of frequency samples must at least equal $2^9 = 512$. Thus, $(2Q+1)M = 12800$ complex numbers must typically be stored simultaneously for use by the explicit linear time-scaling stage. In the implicit TSM system, the interpolation is carried out by the synthesis stage. This stage can be implemented with the overlap-add method described in Chapter 3, which stores only one frame at a time (512 complex numbers).

Portnoff [1978] developed a TSM system which uses an explicit linear time-scaling stage, and which carries out the synthesis by the method described in figure 3.7. By using implicit linear

time-scaling, and by implementing the TSM synthesis stage with an overlap-add DSTFT synthesis algorithm, the storage needs of the TSM system are reduced more than 50 times.

Up to this point in the thesis, the scale factor β has been assumed to be constant. In the next section of this chapter, the structure of the implicit TSM system, shown in figure 4.2, is modified in a way that allows the parameter β to be variable at runtime.

4.5 Non-Uniform Time-Scale Modification

The structure of the implicit TSM system, shown in figure 4.2, is a cascade of three distinct subsystems (stages): Analysis (A), Phase modification (PM) and Synthesis (S). Equation (4.87) specifies the phase modification stage in terms of its input/output behavior. The analysis and synthesis stages can be similarly specified:

$$A\{x[n],h[n],R_A\} = X_s[pR_A,k] \tag{4.90}$$

$$S\{Y_s[n,k],f[n],R_S]\} = y[n/R_S] \tag{4.91}$$

In terms of this operator notation, the implicit TSM system can be described as follows:

$$\begin{aligned} TSM\{x[n],\beta\} &= x^{\beta}[n] \\ &= S\{PM\{A\{x[n],h[n],R_A'\},R_A/R_S\},f[n],R_S\} \end{aligned} \tag{4.92}$$

where $\beta = R_A/R_S$

The scale factor β in equation (4.92) has no time dependence. In order for the value of β to vary in time, one or both of the rates R_A and R_S must vary in time. From equation (4.85), $\beta[m]$ can be expressed as:

$$\beta[m] = R_A[m]/R_S[m] \tag{4.93}$$

The index m has been used instead of the time index n in equation (4.93) to indicate that $\beta[m]$ does not correspond to samples of some continuous signal $\beta(t)$ but, rather, is an ordered succession of independent values. The feasibility of a TSM system which can accept variable analysis and synthesis time rates depends on whether the respective variable rate stages can be implemented.

The analysis algorithm is described in figure 3.6. The algorithm consists of the cascade of four operations: "Window," "Pad," "Rotate" and "FFT." Of these, only the first and third operations depend on the rate R_A. The operation labeled "Window" computes the sequence:

$$x_p[r] = x[r+pR_A-h_2]h[h_2-r] \tag{4.94}$$

To emphasize the role of the rate R_A, equation (4.94) can be written as:

$$x_p[r] = x[r + \sum_{m=1}^{p} R_A-h_2]h[h_2-r] \tag{4.95}$$

The variable rate $R_A[m]$ can be substituted directly for the constant rate R_A in equation (4.95):

$$x_p[r] = x[r + \sum_{m=1}^{p} R_A[m] - h_2]h[h_2 - r] \tag{4.96}$$

Now, let N_p be defined as follows:

$$N_p = \sum_{m=1}^{p} R_A[m] \tag{4.97}$$

In terms of N_p, equation (4.96) becomes:

$$x_p[r] = x[r + N_p - h_2]h[h_2 - r] \tag{4.98}$$

Therefore, $x_p[r]$ is equal to the sequence $x[r]$ windowed by $h[r]$, with the window positioned so that $h[0]$ lies directly above $x[N_p]$. The window position N_p can be defined recursively:

$$N_p = \begin{cases} N_{p+1} - R_A[p] & , \quad p < 0 \\ 0 & , \quad p = 0 \\ N_{p-1} + R_A[p] & , \quad p > 0 \end{cases} \tag{4.99}$$

Thus, the position of the window $h[n]$ for the p^{th} analyzed frame is $R_A[p]$ samples to the right (further in time) than the position of

Table V

Non-Uniform TSM Analysis Algorithm

For all integers $p \geq 0$:

1. $$N_p = \begin{cases} 0 & p = 0 \\ N_{p-1} + R_A[p] & , \quad p > 0 \end{cases} \tag{4.100}$$

2. $$\tilde{x}_p[r] = \begin{cases} x[r+N_p-h_2]h[h_2-r] & , \quad r=0 \text{ to } h_1+h_2 \\ 0 & , \quad \text{otherwise} \end{cases} \tag{4.101}$$

3. $$X_s[N_p,k] = \sum_{r=0}^{M-1} \tilde{x}_p[((r-N_p+h_2))_M]W_M^{kr} \tag{4.102}$$

where $((r))_M$ denotes "r modulo M"

the window for the p-1st frame. If R_A is a constant, then the right shifts of the window from one frame to the next are all of the same size. The window can also be shifted by different amounts from one analyzed frame to the next, so that the windowing operation can be implemented when R_A varies in time.

The only other operation in the analysis algorithm that depends on the value of R_A is the third one, labeled "Rotate." This operation circularly shifts the sequence $\tilde{x}_p[n]$ to the left by $pR_A - h_2$ points. If R_A varies in time, the "Rotate" operation merely performs the circular shift by $pR_A[p] - h_2$ points.

The non-uniform TSM analysis algorithm is shown in Table V. It has been assumed, for simplicity, that the integer index p can only take positive values.

The TSM phase modification algorithm is described in equations (4.75) - (4.81). The only place where the rates R_A and R_S enter into the algorithm is in equation (4.79), where the division by $\beta = R_A/R_S$ occurs. Clearly, this equation does not have to be modified at all to allow β to vary in time except, perhaps, to explicitly show this time variation; if β varies in time, equation (4.79) becomes:

$$\arg\{Y_s[p,\underline{k}]\} = \begin{cases} \theta_{PV}[0,\underline{k}] & , \quad p = 0 \\ \arg\{Y_s[p-1,\underline{k}]\} + \dfrac{R_S[p]}{R_A[p]} \Delta\hat{\zeta}[p \dfrac{R_A[p]}{R_S[p]}, \underline{k}] & , \quad p > 0 \end{cases} \tag{4.103}$$

Therefore, the TSM phase modification algorithm designed for a time-independent value of β can be used directly when β varies in time.

We have seen that R_A can vary in the analysis and phase modification algorithms. A simple argument will show, however, that the TSM synthesis algorithm cannot easily adapt to a variable rate R_S. As described in equations (3.57) - (3.60), the overlap-add synthesis algorithm is based on a running sum of the sequences $\psi_{pR_S}[n]$. These sequences correspond to the contribution of the p^{th} spectral frame of $Y_s[pR_S,k]$ to the sequence $y[n]$. By adding the sequences $\psi_{pR_S}[n]$ together, we are effectively interpolating selected points of the sequence $y[n,m]$ (which is the fully sampled version of the sequence $y[pR_S,n] = y[pR_S,((n))_M]$, defined in equation (3.57)). This interpolation is performed by means of the 1-to-R_S interpolating filter $f[n]$. If R_S were to vary in time, the filter $f[n]$ would itself have to vary, with one sequence $f_{R_S}[n]$ corresponding to each possible value of the rate R_S. Each of these filters would have to be stored in memory if a variable rate R_S were used. Furthermore, there is no simple way to construct $y[n]$ from a set of sequences $\psi_{pR_S}[n]$ obtained with different interpolating filters. Therefore, if the overlap-add synthesis algorithm is used, the rate R_S cannot vary in time. Implementation of the variable rate synthesis stage with the direct algorithm (shown in figure 3.7) was not considered in this thesis because the increased storage requirements made this an uninteresting alternative.

The variable scale factor $\beta[m]$ can therefore be implemented by selecting a constant denominator R_S, and letting $R_A[m]$ vary in time:

$$\beta[m] = R_A[m]/R_S \tag{4.104}$$

The selection of R_S must be done while realizing that $R_A[m]$ can only vary from 1 to about 40, as discussed in Sections 4.2 and 4.3. Therefore, $\beta[m]$ can only take values between $1/R_S$ and $40/R_S$.

In terms of the operator notation developed at the beginning of this section, the non-uniform TSM analysis algorithm can be described as follows:

$$A\{x[n],h[n],R_A[p]\} = X_s[N_p,k] \tag{4.105}$$

$$\text{where} \quad N_p = \sum_{m=1}^{p} R_A[m]$$

Replacing β in equation (4.87) by its value given in equation (4.104), and replacing equation (4.79) in the uniform-rate phase modification algorithm by equation (4.103), the non-uniform TSM phase modification algorithm can be described as follows:

$$PM\{X_s[n,k], R_A[p]/R_S\} = Y_s\{npR_s / \sum_{m=1}^{p} R_A[m],k]\}$$

$$= Y_s\{npR_S/N_p,k]\} \tag{4.106}$$

Since R_S remains constant for non-uniform TSM, the synthesis algorithm in this case is described by equation (4.91).

Table VI

Non-Uniform TSM Algorithm

$$x^{\beta[m]}[n] = TSM\{x[n],\beta[p]\}$$

$$= S\{PM\{A\{x[n],h[n],R_A[p]\},R_A[p]/R_S\},f[n],R_S\} \qquad (4.107)$$

The non-uniform TSM algorithm is described in Table VI in terms of the operators defined in equations (4.91), (4.105) and (4.106).

Figure 4.3 describes the structure of the non-uniform TSM system based on the algorithm described in equation (4.107). Comparison of figures 4.2 and 4.3 will show the differences between the uniform and non-uniform TSM systems. The uniform analysis and phase modification stages (shown in figure 4.2) have been replaced by the corresponding non-uniform stages in figure 4.3. In addition, the non-uniform TSM system includes an input port for the analysis rate sequence $R_A[p]$. It is assumed that the sequence $R_A[p]$ is long enough so that the value of the analysis window position marker, $N_p = \sum_{m=1}^{p} R_A[m]$, will eventually become larger than the length of x[n].

The next section in this chapter describes the implementation of the non-uniform TSM system described in figure 4.3.

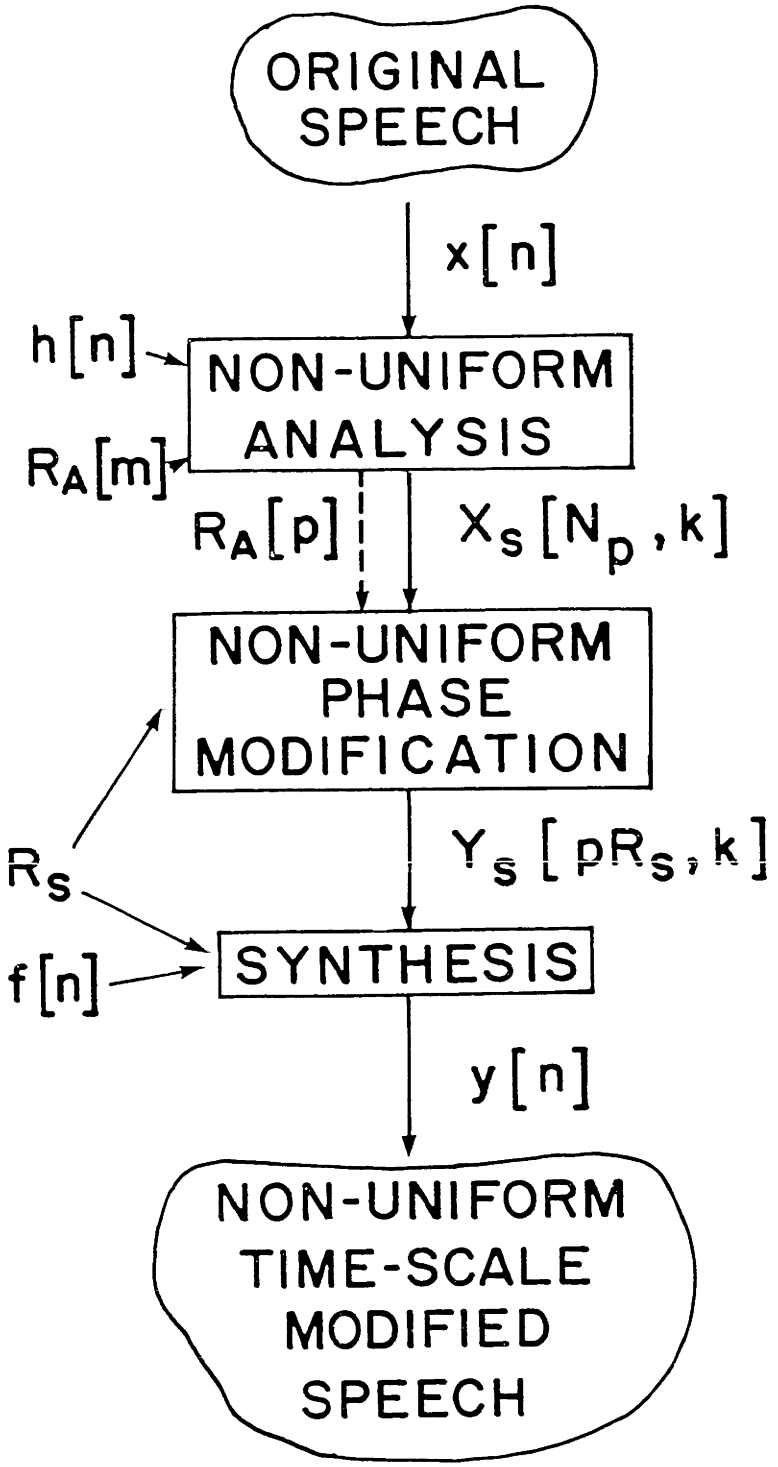
4.6 Implementation of a Non-Uniform TSM System

The non-uniform TSM system developed in this chapter was implemented in a general purpose PDP-11/50 minicomputer.

The sequence x[n] was obtained from the speech signal x(t). First, x(t) was low-pass filtered at 4.98 KHz and sampled at 10 KHz to form the sequence x'[n]. Second, x[n] was generated by pre-emphasizing x'[n] as follows:

$$x[n] = x'[n] - 0.995x'[n-1] \tag{4.108}$$

The Non-Uniform TSM System



$$\left(\beta[m] = \frac{R_A[m]}{R_s}\right)$$

Figure 4.3

The speech was pre-emphasized in order to decrease the dynamic range of the signal by flattening its spectral envelope [Holtzman Dantus, 1977]. In some cases, the signal x(t) was pre-emphasized prior to the filtering and sampling operation using the Speech Pre-emphasis filter designed by Holtzman Dantus [1977].

The analysis filter was chosen as a Hamming window. Several window lengths were used, but the best results were obtained with a 256-point window. The cutoff frequency of an N-point Hamming window is approximately equal to $4\pi/N$ [Oppenheim and Schafer, 1975]. Therefore, the analysis filter had a cutoff frequency ω_h equal to $\pi/64$ which, for a 10 KHz sampling rate, corresponds to an effective cutoff frequency of 78 Hz. This cutoff frequency falls within the 70 Hz - 100 Hz range discussed in sections 4.2 and 4.3.

The number M of frequency samples was chosen to be equal to 512. Portnoff [1978] has shown that values of M smaller than 512 will severly decrease the frequency resolution of the DSTFT. Larger values of M were not feasible due to memory size limitations. The M-point Discrete Fourier Transform was performed with an assembly language version of the decimation-in-time FFT algorithm described by Rabiner and Gold [1975].

The synthesis filter f[n] was designed using the method described by Oetken [1975]. In general, the filter order was chosen to be about 16, with a cutoff frequency of 0.75π.

The bound imposed on $R_A[m]$ by equation (4.86), with the parameter values chosen above, reduces to $R_A[m] < 64$, for all m. The analysis rate was conservatively allowed to vary between 1 and 48. To avoid

aliasing due to the non-linearity of the phase modification algorithm [Portnoff, 1978], the synthesis rate R_S was also chosen to be between 1 and 48:

$$1 \leq R_A, R_S \leq 48 \tag{4.109}$$

Finally, the output of the non-uniform TSM system was de-emphasized using the inverse of equation (4.108):

$$y'[n] = \sum_{m=0}^{n} (0.995)^m y[n] \tag{4.110}$$

The non-uniformly time-scale modified signal $y(\beta[m]t)$ was obtained by filtering $y'[n]$ with a 4.98 KHz low-pass filter. As in the case of the conversion of $x(t)$ to digital form, the de-emphasis and filtering operations were sometimes performed in reverse order. In this case, the de-emphasis was carried out by the de-emphasis filter designed by Holtzman Dantus [1977].

4.7 Comparison with Portnoff's System

The system developed by Portnoff [1978] is described in figure 4.1, and the non-uniform system developed in this thesis is described in figure 4.3. This section compares and contrasts the two systems.

In Portnoff's system there are five separate stages: Analysis, Decimation, Interpolation, Phase Modification and Synthesis.

In the new system, the Decimation and Interpolation stages have been eliminated. The remaining three stages carry out the same functions as the corresponding stages in Portnoff's system, but perform their tasks somewhat differently.

The non-uniform analysis stage differs from Portnoff's analysis stage in two respects: first, it allows a variable analysis rate, $R_A[m]$ and, second, because it assumes that the length of the analysis window h[n] is less than the number M of DSTFT frequency samples, it performs fewer computations. The phase modification stages in both systems are the same, except that in the non-uniform system, the scale factor β need not be constant. Finally, the synthesis stage in the new system is computationally identical to Portnoff's synthesis stage but, as discussed in Chapter 3, the overlap-add synthesis algorithm used in the new system is significantly more efficient in terms of storage requirements. It is important to note, however, that each of the stages in the new system could be substituted directly for the corresponding stage in Portnoff's system.

The performance of the two systems was compared by assuming a constant scale factor β for the non-uniform system. When the speech is expanded ($\beta < 1$), or mildly compressed ($1 < \beta < 3$), the two systems perform almost identically. When the speech is severely compressed ($\beta > 3$), however, the new system retains a higher pitch quality than Portnoff's system. As an identity system ($\beta = 1$), both TSM systems recover the original signal exactly.

This part of the thesis has described the development of a non-uniform TSM system. The second part evaluates the performance and the usefulness of this system.

PART II

USING THE TSM SYSTEM

CHAPTER 5

FEATURE-DEPENDENT TIME-SCALE MODIFICATION OF SPEECH

The model of the speech signal, x[n], developed in Chapter 2, consists of a sum of harmonically related exponentials. This model is shown in equation (2.20), which is repeated here for convenience:

$$x[n] = \sum_{r=0}^{p[n]-1} c_r[n] e^{jr\phi[n]} \tag{5.1}$$

The TSM system presented in Part I of the thesis was designed under the assumption that, within this model, the parameters p[n] and $c_r[n]$ vary much more slowly (as a function of n) than the time-unwrapped phase $\phi[n]$. While this is a valid assumption over most of the length of x[n], it is sometimes violated during speech transition segments (such as stop consonants, voiced-to-unvoiced transitions and rapid intervowel glides).

The speech is particularly degraded when the value of the scale factor β is much greater than one. When β is equal to one, however, the TSM system becomes an identity system, and does not degrade the signal. A possible improvement to a uniform TSM system might then be to adapt the value of the scale factor β to the local structure of the speech by making it approach unity during transition

segments. This technique is referred to as feature-dependent TSM.

In the case of compression, Toong [1974] has shown (using a Fairbanks type TSM system) that feature-dependent TSM may decrease the overall degradation of the time-scale modified speech.

This chapter develops a feature-dependent TSM system, based on the non-uniform TSM system developed in part I of the thesis. A possible strategy for performing feature-dependent TSM is to severely time-scale modify pause segments in the speech, in order to reduce the severity of the TSM during non-pause segments. This and other such trivial schemes are not considered in this thesis. The dependency on speech features is obtained by pre-processing the speech sequence, $x[n]$, to generate a sequence, $\beta[m]$, of TSM scale factors that varies in response to local speech features, such as transitions. The sequence $\beta[m]$ is then passed to the non-uniform TSM system in the form of a fixed synthesis rate R_S, and a sequence of analysis rates $R_A[m]$, such that:

$$\beta[m] = \frac{R_A[m]}{R_S} \tag{5.2}$$

The generation of the rates R_S and $R_A[m]$ from the speech sequence $x[n]$ is carried out in two stages. First, the speech is segmented into regions of similar local structure and, second, the segment characteristics (type and boundaries) are translated into a common synthesis rate, R_S, and a sequence of feature-dependent analysis rates, $R_A[m]$.

The remainder of this chapter presents three speech segmentation algorithms (two of which are automatic, and one of which is manual), discusses the translation process from segment characteristics to actual rate values, and compares feature-dependent and uniform TSM of speech.

Section 5.1 develops two different automatic speech segmentation algorithms. A manual speech segmentation scheme is described in Section 5.2. This procedure is based on the visual recognizability of certain speech features when the signal is plotted graphically (amplitude versus time). Next, the generation of the rates R_S and $R_A[m]$ from segment characteristics is discussed in Section 5.3. Finally, in Section 5.4, the feature-dependent TSM system is compared to a uniform TSM system.

5.1 Automatic Speech Segmentation Algorithms

This section presents two algorithms to automatically segment the speech signal. Although these algorithms select segment boundaries differently, they are based on a common set of three statistical measures of the speech.

For clarity, this section is divided into three subsections. First, the statistical measures that form the common ground for the segmentation algorithms are described in subsection 5.1.1 and, then, the two algorithms are presented, respectively, in subsections 5.1.2 and 5.1.3.

5.1.1 Speech Statistics

As discussed earlier in this chapter, it is desirable for the value of the TSM scale factor β to approach unity during speech transition periods. It is therefore necessary to develop a means of detecting the transition periods.

Stevens [1971] has shown that rapid transients in the speech signal can be identified with rapid changes in the magnitude of its short-time spectrum. Speech transitions can therefore be detected by measuring the difference between short-time magnitude spectra some R_M samples apart. The discrete short-time magnitude spectrum of the signal x[n], windowed by h[n] and sampled every R_M points, can be obtained from equation (3.34) as follows:

$$\left|X_s[pR_M,k]\right| = \left| \sum_{n=pR_M-h_2}^{pR_M+h_1} x[n]h[pR_M-n]W_M^{kn} \right| \qquad (5.3)$$

If Δ_p denotes the first backward difference operator along the index p, then a speech transition can be identified by large values of the length of the M-dimensional vector $\Delta_p\left|X_s[pR_M,k]\right|$. This length can be obtained with the Euclidean distance operator (ED_p) on the M-element vector $\left|X_s[pR_M,k]\right|$, along the index p, which is defined as follows:

$$ED_p\{\left|X_s[pR_M,k]\right|\} = \sum_{k=0}^{M-1} \left(\Delta_p\left|X_s[pR_M,k]\right|\right)^2 \qquad (5.4)$$

As a means of detecting speech transitions, however, this measure has the undesirable effect of becoming large when the overall energy of the speech signal changes abruptly. Thus, any emphasis or accentuation that might occur in the speech could be erroneously detected as a transition. To avoid this problem, we define the normalized Euclidean distance (NED_p) on the M-element vector $|X_s[pR_M,k]|$, along the index p, as follows:

$$NED_p\{|X_s[pR_M,k]|\} = ED_p\left(\frac{|X_s[pR_M,k]|}{E_k\{|X_s[pR_M,k]|\}}\right) \tag{5.5}$$

where $E_k\{|X_s[pR_M,k]|\}$ is the energy of the vector $|X_s[pR_M,k]|$ along the index k, and is defined by:

$$E_k\{|X_s[pR_M,k|\} = \sqrt{\sum_{k=0}^{M-1} (X_s[pR_M,k])^2} \tag{5.6}$$

The NED_p operator can be used to detect speech transitions because it is independent of the energy of the signal. In addition, this operator has a useful geometric interpretation. Combining equations (5.4) - (5.6), $NED_p\{|X_s[pR_M,k]|\}$ can be easily shown [Wiener, 1949] to be given by the relation:

$$NED_p\{|X_s[pR_M,k]|\} = 2\left(1-\cos\{|X_s[\underline{p}R_M,k]|,|X_s[(p-1)R_M,k]|\}\right) \tag{5.7}$$

Since the M scalars that comprise $|X_s[\underline{p}R_M,k]|$ are all positive, the cosine of the angle formed by the vectors $|X_s[\underline{p}R_M,k]|$ and

$|X_s[(p-1)R_M,k]|$ must be in the range:

$$0 \leq \cos\{|X_s[pR_M,k]|,|X_s[(p-1)R_M,k]|\} \leq 1 \tag{5.8}$$

Therefore, $NED_p\{|X_s[pR_M,k]|\}$ satisfies the relation:

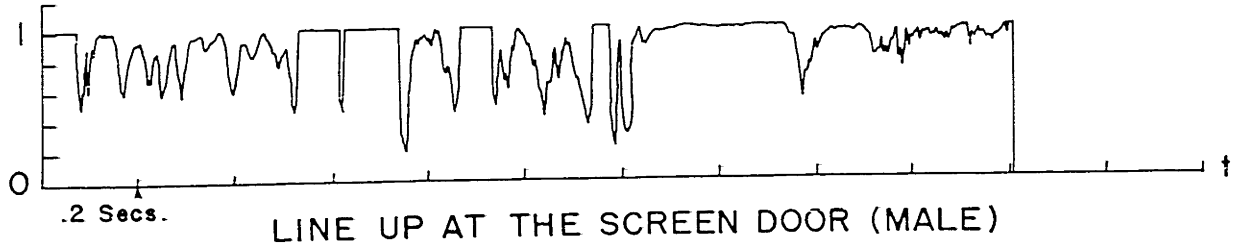
$$0 \leq \frac{1}{2} \cdot NED_p\{|X_s[pR_M,k]|\} \leq 1 \tag{5.9}$$

Consequently, speech transitions can be detected by comparing the quantity $NED_p\{|X_s[pR_M,k]|\}/2$ to a fixed threshold, NED_{min}, such that:

$$NED_p\{|X_s[pR_M,k]|\}/2 \begin{cases} < NED_{min} \rightarrow \text{transition} \\ \geq NED_{min} \rightarrow \text{normal} \end{cases} \tag{5.10}$$

The actual choice of the measure rate R_M, the window $h[n]$ and the threshold NED_{min} is experimental. According to Stevens [1971] and Klatt [1979], $h[n]$ should approximate the impulse response of a low-pass filter with a 300 Hz cutoff frequency, which is the case for a 64-point Hamming window at a 10 KHz sampling rate. With a window of this kind, Stevens [1971] has shown that R_M should be equal to 300 samples, again assuming a 10 KHz sampling rate. Figure 5.1 shows two typical NED_p measures. The value of the threshold, NED_{min}, will be determined in the next two subsections

Typical NED_p Measures



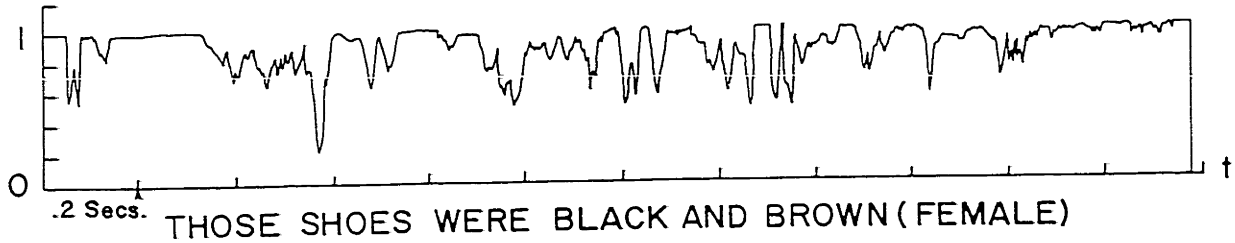


Figure 5.1

when the individual segmentation algorithms are discussed.

In addition to obtaining the normalized Euclidean distance (NED_p) measure of the speech, it is useful to estimate the local normalized log-energy value (NRG_p) and the local normalized zero crossing count (ZXS_p) of the speech signal, as a function of the time index p [Agrawal and Lin, 1975]. These measures are defined by the relations:

$$NRG_p\{x[n]\} = \frac{1}{5}\left[\log_{10}\left[\frac{\sum_{n=pR_N-h_2}^{pR_N+h_1}(x[n]h_N[pR_N-n])^2}{x_{max}^2 \cdot (E_n\{h_N[n]\})^2}\right]\right] + 1 \qquad (5.11)$$

$$ZXS_p\{x[n]\} = \frac{\sum_{n=pR_Z-h_2}^{pR_Z+h_1-1}\left[|SGN\{x[n]\}\cdot SGN\{x[n+1]\}| \cdot (1-SGN\{x[n]\}\cdot SGN\{x[n+1]\})\right]}{2 \cdot (h_1+h_2)} \qquad (5.12)$$

The various parameters in equations (5.11) and (5.12) are defined as follows:

- $h_N[n]$ is a 256-point Hamming window, from $-h_1 = -63$ to $h_2 = 64$.
- x_{max} is the maximum possible value of the sequence x[n], usually normalized to unity.

- SGN{α} is the sign function of α. It is defined as follows:

$$SGN\{\alpha\} = \begin{cases} -1 & , \quad \alpha < 0 \\ 0 & , \quad \alpha = 0 \\ 1 & , \quad \alpha > 0 \end{cases} \tag{5.13}$$

- The sampling rates R_N and R_Z have been experimentally determined to be (for best results):

$$R_N = R_Z = 20 \text{ samples.}$$

The measures defined in equations (5.11) and (5.12) are shown in figure 5.2 for the sentence: "Line up at the screen door."

The next two subsections use the NED_p, NRG_p and ZXS_p measures to segment the speech signal x[n]. The first segmentation algorithm, presented in subsection 5.1.2, is primarily based on the NED_p measure. The second algorithm, described in subsection 5.1.3, uses all three measures to segment the speech. Both segmentation algorithms consist of a decision procedure to determine the nature of the local structure of x[n]. These decision procedures are based on a set of comparisons between the local values of the speech measures and experimentally chosen thresholds.

5.1.2 Segmentation by Spectral Similarity

The speech segmentation algorithm described in this subsection categorizes speech into five segment types:

1. - Pause.
2. - Stationary.
3. - Slowly varying spectrum.
4. - Rapidly varying spectrum.
5. - Very rapidly varying spectrum.

(5.14)

Log-energy and Zero Crossing Count Measures

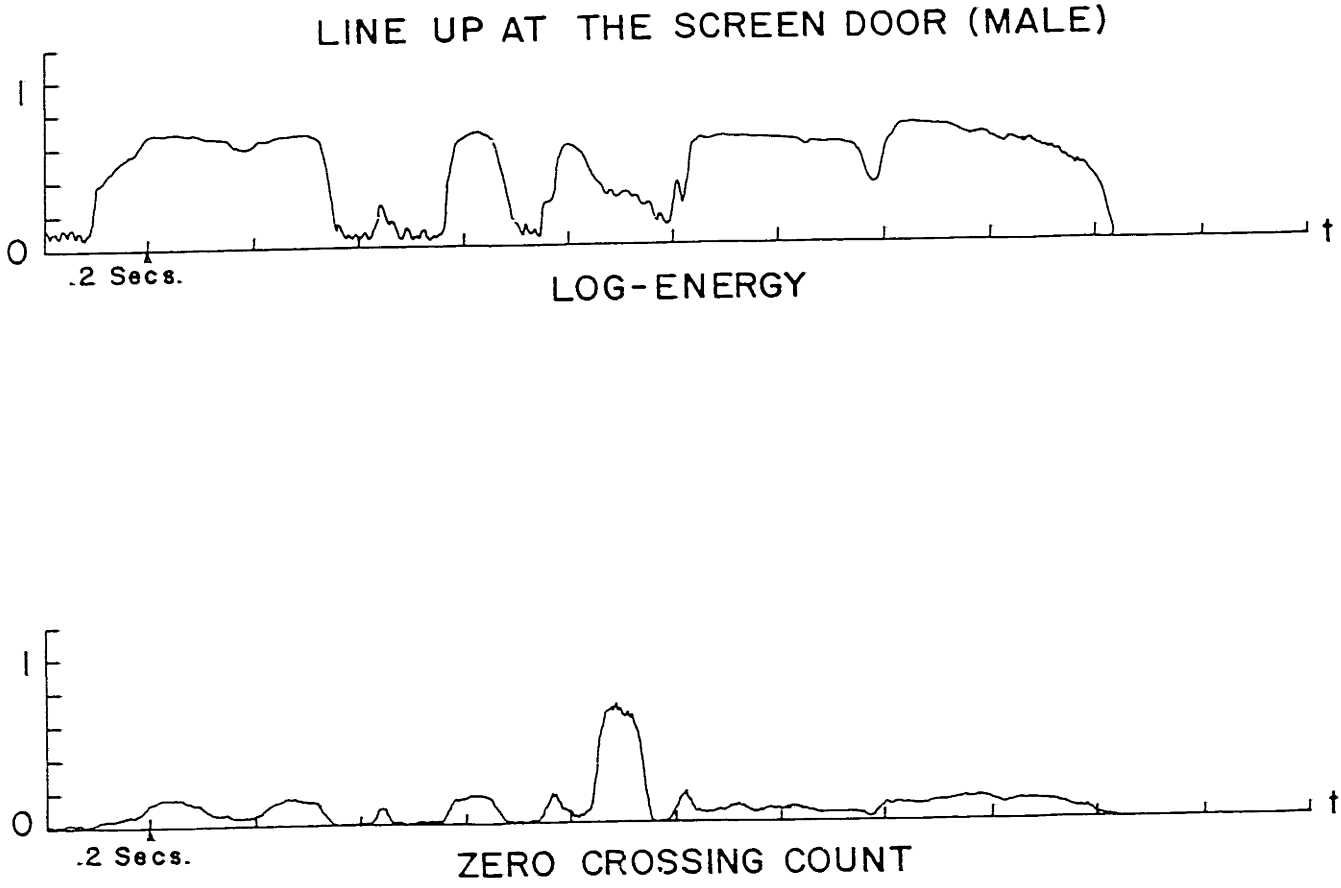


Figure 5.2

Although the decision procedure used by this segmentation algorithm involves all three speech measures, the segment type selection is based primarily on the NED_p measure. Equation (5.10) shows that the normal versus transition decision can be made by means of a threshold (NED_{min}). This single threshold is replaced here by a set of three thresholds: NED_{Low}, NED_{Middle} and NED_{High}.

The NRG_p and ZXS_p measures are used for pause detection. If each measure falls below its threshold (NRG_{min} and ZXS_{min}, respectively), a pause is detected [Agrawal and Lin, 1975].

Finally, the ZXS_p measure is used to improve the accuracy of the NED_p measure. As Agrawal and Lin [1975] have shown, if the zero crossing count exceeds 0.25, then the speech is locally unvoiced. In this case, due to the random nature of the speech waveform, the NED_p measure will take relatively high values.

For this reason, an additional threshold for the ZXS_p measure is defined:

$$ZXS_{Unvoiced} = 0.25 \tag{5.15}$$

When the ZXS_p measure exceeds $ZXS_{Unvoiced}$, the deviation from unity of the NED_p measure is attenuated. An attenuation of 30% has been experimentally shown to be adequate.

The decision procedure to segment the speech by spectral similarity is given in Table VII. The commonly accepted "Pidgin ALGOL" convention [Aho, Hopcroft and Ullman, 1974] for specifying algorithms is used in this thesis.

Table VII

Segmentation by Spectral Similarity

BEGIN

IF (ZXS_p > $ZXS_{Unvoiced}$) THEN: $NED_p \leftarrow NED_p - [(1-NED_p) \cdot 0.3]$ (5.16)

IF (NRG_p < NRG_{min}) and (ZXS_p < ZXS_{min}) THEN: "Pause"

ELSE IF (NED_p > NED_{High}) THEN: "Stationary"

ELSE IF (NED_p > NED_{Middle}) THEN: "Slowly varying"

ELSE IF (NED_p > NED_{Low}) THEN: "Rapidly varying"

ELSE: "Very rapidly varying"

END (5.17)

Typical values for the thresholds have been determined (by trial and error) to be:

$$NED_{High} = 0.83 \tag{5.18}$$

$$NED_{Middle} = 0.74 \tag{5.19}$$

$$NED_{Low} = 0.66 \tag{5.20}$$

$$NRG_{min} = 0.20 \tag{5.21}$$

$$ZXS_{min} = 0.055 \tag{5.22}$$

$$ZXS_{Unvoiced} = 0.25 \tag{5.23}$$

The decision procedure shown in Table VII categorizes x[n] in the vicinity of n = pR, for some sampling rate R. Since R may not (and generally does not) equal the actual sampling rates of the measures (R_M, R_N and R_Z), intermediary measure values are obtained by linear interpolation. A more accurate interpolation procedure is not necessary because the measures vary slowly with respect to the time index p.

Figure 5.3 shows the result of segmenting the sentence "You are the biggest man" by spectral similarity. The three speech measures from which the segmentation was obtained are shown in the figure for reference.

Segmentation by Spectral Similarity

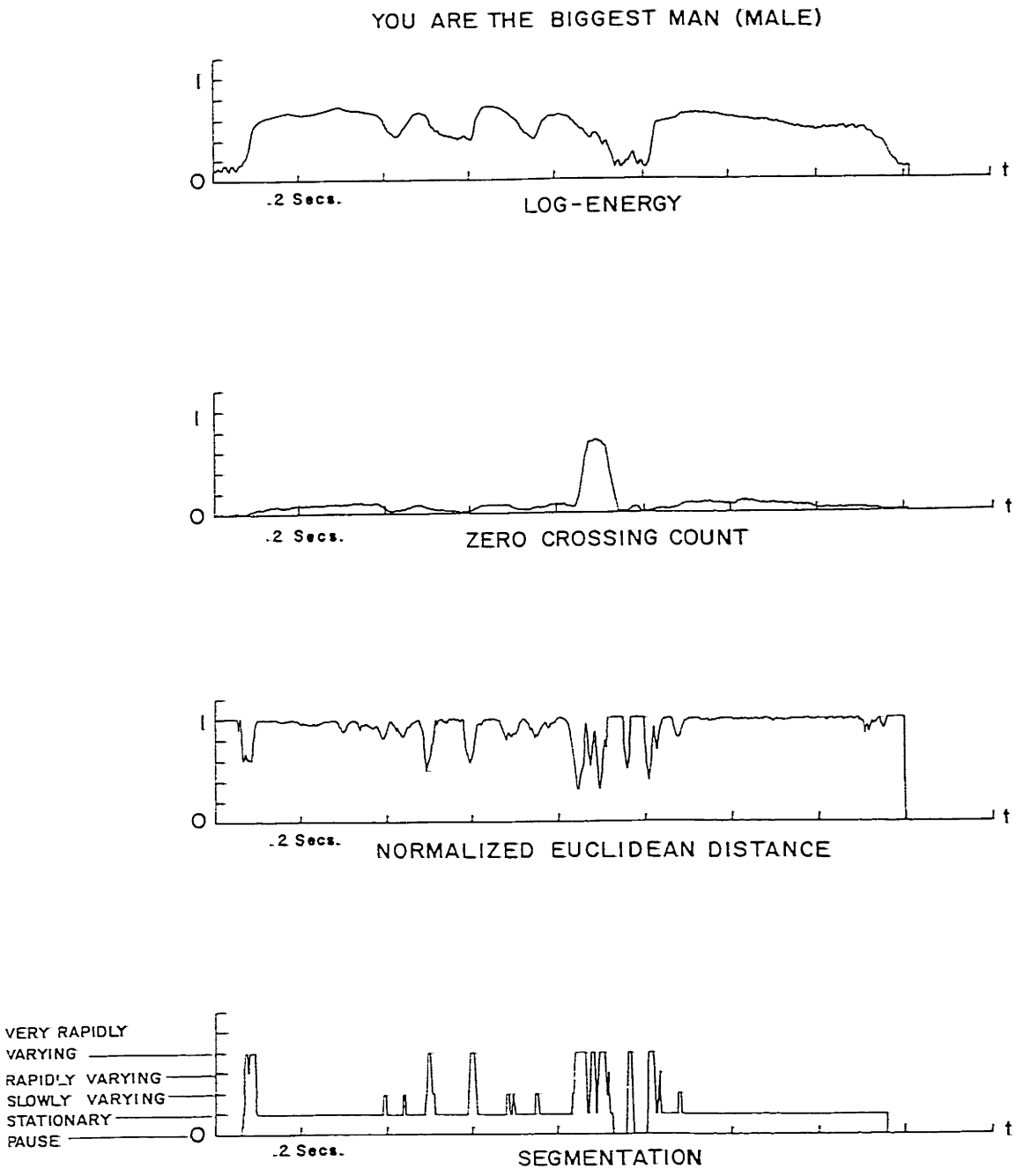


Figure 5.3

5.1.3 Speech-Specific Segmentation

An alternative segmentation algorithm categorizes the speech into four segment types which are specific to speech signals:

1. - Pause
2. - Voiced
3. - Unvoiced
4. - Transition (5.24)

The speech-specific segmentation algorithm uses a single transition threshold for comparison with the NED_p measure, as shown in equation (5.10). The voiced/unvoiced/pause decision is then made based on the values of the NRG_p and ZXS_p measures [Agrawal and Lin, 1975]. As in the case of the spectral similarity segmentation algorithm, when an unvoiced region is detected, the deviation from unity of the NED_p measure is attenuated by 30%.

The decision procedure used to segment the speech into "Pause," "Voiced," "Unvoiced" and "Transition" regions is shown in Table VIII.

Typical values for the thresholds in equations (5.29) and (5.30) have been experimentally determined to be as follows:

$$NED_{min} = 0.73 \quad (5.25)$$

$$NRG_{min} = 0.20 \quad (5.26)$$

$$ZXS_{min} = 0.055 \quad (5.27)$$

$$ZXS_{Unvoiced} = 0.15 \quad (5.28)$$

In the same manner as the spectral similarity segmentation algorithm, the measures are linearly interpolated to adjust to the sampling rate of the segmentation.

Table VIII

Speech-Specific Segmentation

BEGIN

IF (ZXS_p > $ZXS_{Unvoiced}$) THEN: $NED_p \leftarrow NED_p - [(1-NED_p) \cdot 0.3]$ (5.29)

IF ($NRG_p < NRG_{min}$) and ($ZXS_p < ZXS_{min}$) THEN: "Pause"

ELSE IF ($NED_p < NED_{min}$) THEN: "Transition"

ELSE IF ($ZXS_p > ZXS_{Unvoiced}$) THEN: "Unvoiced"

ELSE: "Voiced" (5.30)

END

Figure 5.4 shows the result of segmenting the sentence "You are the biggest man" with the speech-specific algorithm. The differences between the two segmentation algorithms can be seen by comparing figures 5.3 and 5.4.

These two schemes for speech segmentation can be carried out in a completely automated fashion. Section 5.2 describes an alternative segmentation scheme which requires human intervention, but which is likely to be more accurate than either of the two algorithms presented in this section.

5.2 Manual Speech Segmentation

The basis for the manual segmentation scheme is that voiced, unvoiced, transition and pause regions can usually be recognized visually in an amplitude versus time plot of the speech signal. Figure 5.5 shows the typical appearance of these regions. Voiced speech, shown in figure 5.5(a), is a quasi-periodic signal. Unvoiced speech, shown in figure 5.5(b), can be recognized by its noise-like (random) appearance. An example of a speech transition region is shown in figure 5.5(c). The characterizing feature of transition regions is rapid change in the nature of the signal, occurring over a short time interval (usually in the order of 100 milliseconds). Finally, a pause is shown in figure 5.5(d).

The speech signal can be readily segmented by visual identification of the type and boundaries of each consecutive region. Figure 5.6 shows a portion of the sentence "Line up at the screen door" (around the "s" in screen), with its manually selected segment boundaries.

Speech-specific Segmentation

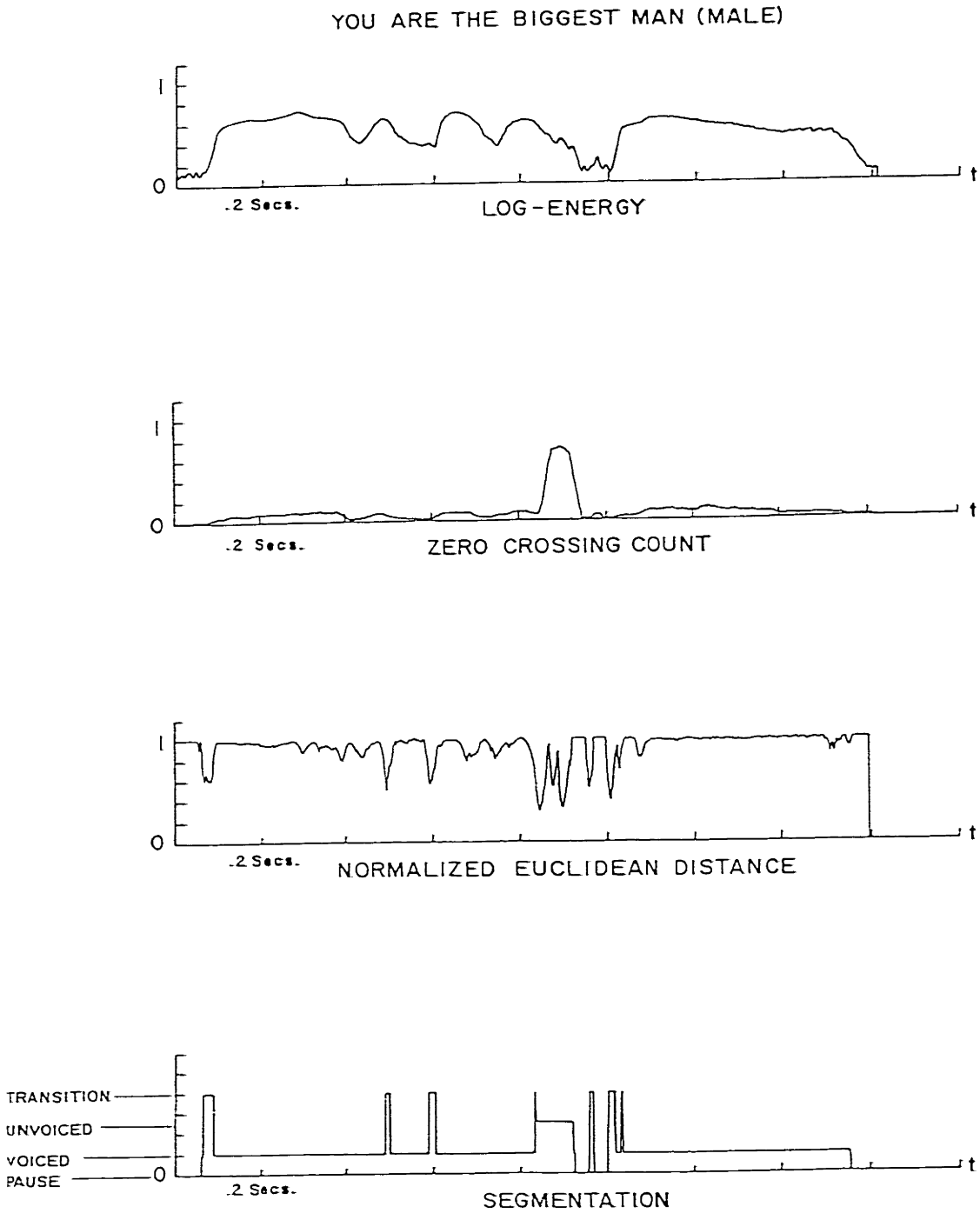


Figure 5.4

Typical Appearance of Speech Regions

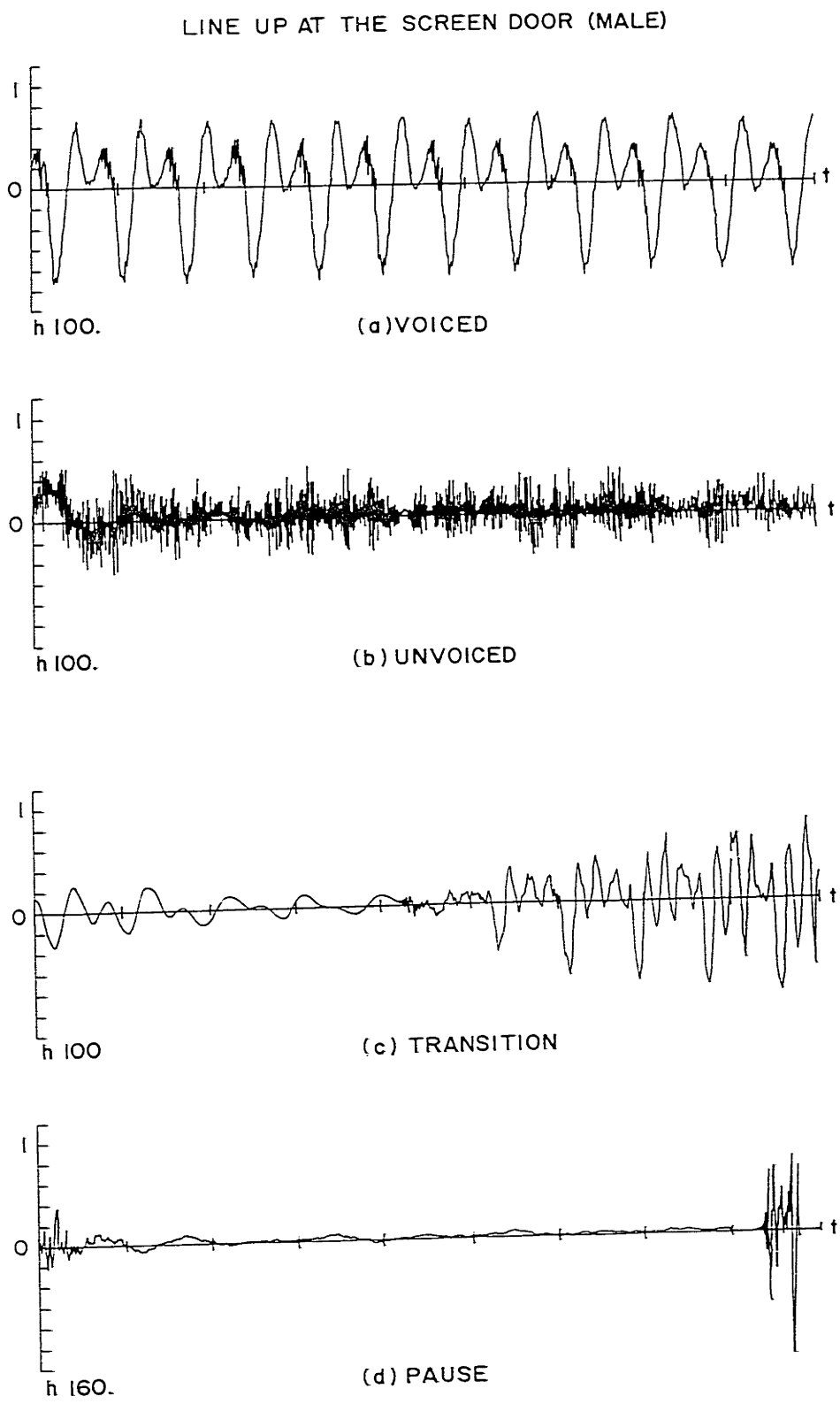


Figure 5.5

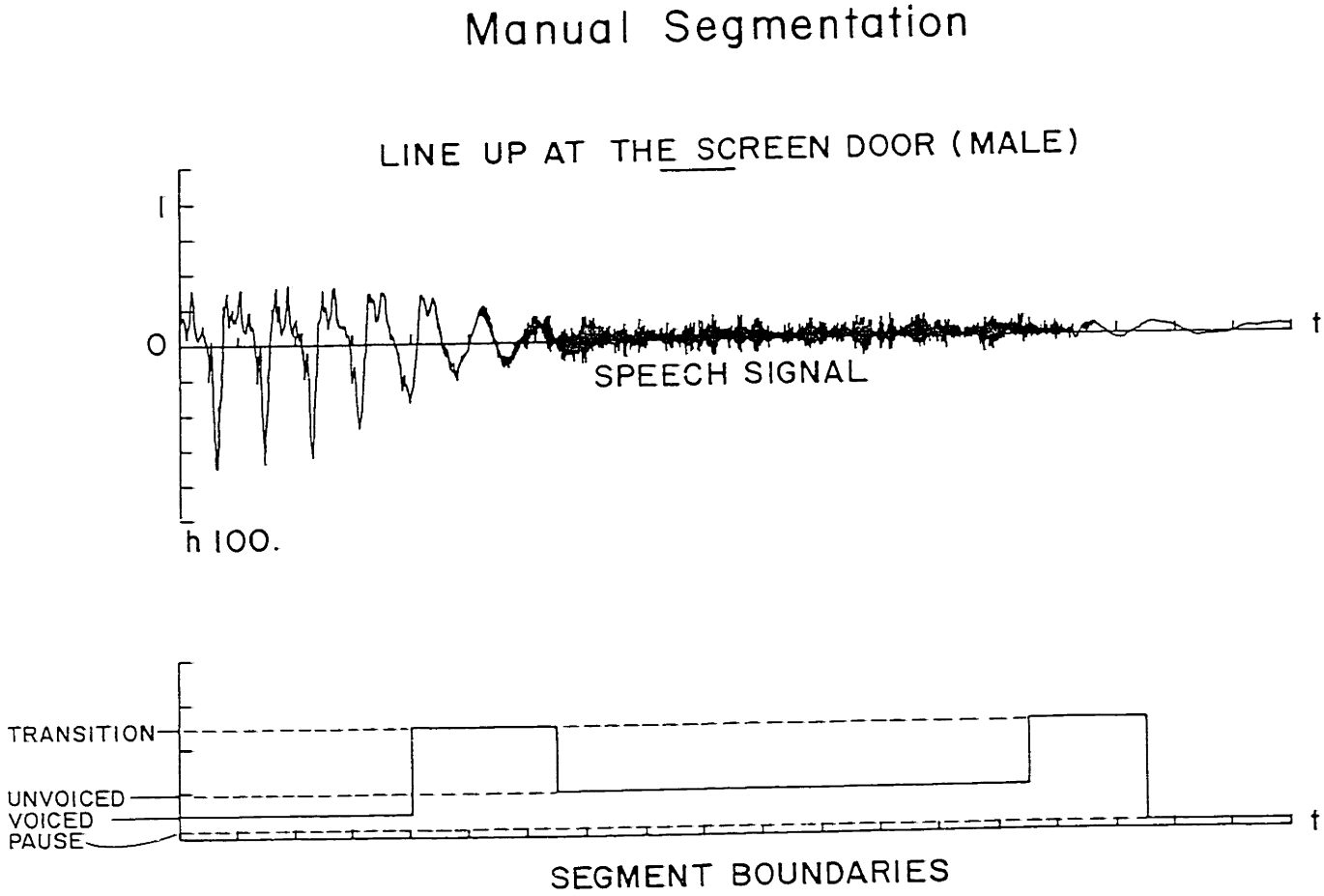

Manual Segmentation
LINE UP AT THE SCREEN DOOR (MALE)
1
0
h 100.
t
SPEECH SIGNAL
TRANSITION
UNVOICED
VOICED
PAUSE
t
SEGMENT BOUNDARIES

Figure 5.6

Regardless of the segmentation algorithm used, the resulting set of segments must be translated into a fixed synthesis rate, R_S, and a sequence of analysis rates, $R_A[m]$. These rates can then be used by the non-uniform TSM system to complete the feature-dependent TSM operation. The generation of R_S and $R_A[m]$ is discussed in Section 5.3.

5.3 Input to the Non-Uniform TSM System

The output of the segmentation algorithms described in the previous two sections consists of a sequence of segments, each defined by three parameters: beginning sample, ending sample and segment type. This sequence of segments must be transformed into a fixed synthesis rate, R_S, and a sequence of analysis rates, $R_A[m]$, which define the desired feature-dependent scale factor sequence, $\beta[m] = R_A[m]/R_S$.

Let S_q denote the q^{th} segment of the signal $x[n]$. Thus, S_q represents the triplet

$$S_q = \{b[q], e[q], T[q]\} \tag{5.31}$$

where $b[q]$ is the sample of $x[n]$ at which S_q begins, $e[q]$ is the end sample of S_q, and $T[q]$ is the type of S_q (e.g. "Voiced"). For convenience, we shall assume that the $q+1^{st}$ segment immediately follows the q^{th} segment. Thus, excluding the first and last segments, we have:

$$b[q] = e[q-1] + 1 \tag{5.32}$$

In addition, let us assume that there are Q segments (q = 0,1,...,Q-1), which span the whole signal x[n]. Consequently:

$$b[0] = 0 \tag{5.33}$$

$$\text{and} \quad e[Q-1] = \langle \text{last sample number in } x[n] \rangle \tag{5.34}$$

To obtain an overall scale factor β_0 (defined as the ratio of the length of the input speech signal over the length of the output time-scale modified signal), a feature-dependent scale factor sequence, $\beta[m]$, must be generated from the information contained in the set of Q triplets S_q. To generate $\beta[m]$, an analysis rate R_A^r must be assigned to the r^{th} segment type (there are either 4 or 5 different segment types, depending on the segmentation scheme used) and a fixed synthesis rate R_S must be selected. An overall scale factor β_0 can then be obtained by choosing values for $R_A[m]$ and R_S that minimize the error term in the formula:

$$\beta_0 = \left[\sum_{r=1}^{4 \text{ or } 5} R_A^r \left(\sum_{q \in Q_r} e[q]-b[q]+1 \right) \right] / R_S + \varepsilon \tag{5.35}$$

$$Q_r = \{q \mid T[q]=r\}$$

The error term ε in equation (5.34) is due to the fact that β_0 cannot, in general, be obtained exactly by selecting values for

$R_A[m]$ and R_S, since these rates are restricted to be integers between 1 and about 40, depending on the analysis filter used in the TSM system. Since there is no closed form solution to the problem of minimizing ε in equation (5.35), the selection of the rates R_A^r and R_S is heuristic. The selection can be accomplished either by user trial and error, or automatically by a "hill climbing" search algorithm [Nilsson, 1980].

Once the rates R_A^r and R_S have been determined, the sequence $R_A[m]$ (and consequently the sequence $\beta[m]$) can be easily generated. Equation (4.97) defines the quantity N_p:

$$N_p = \sum_{m=1}^{p} R_A[m] \tag{5.36}$$

The sequence $R_A[m]$ can then be generated by the following algorithm:

```
BEGIN
   p ← 0,  N_0 ← 0
   FOR q ← 0  UNTIL Q-1 DO
      UNTIL (N_p ≥ e[q]) DO
         BEGIN
            R_A[p] ← R_A^T[q]
            p ← p + 1
         END
END
```

(5.37)

Since e[Q-1] is equal to the last sample number in x[n] (equation (5.34)), the sequence $R_A[m]$ generated by equation (5.37) will be long enough so that the whole sequence x[n] will be processed by the TSM system.

Figure 5.7 shows the complete feature-dependent TSM system. The option of automatic or manual segmentation schemes is explicitly shown in the figure but, for simplicity, the choice between the two automatic segmentation algorithms discussed in this chapter is not shown.

Section 5.4 concludes this chapter with a comparison of uniform and feature-dependent TSM.

5.4 Evaluation of Feature-Dependent TSM

A careful, though somewhat informal, comparison of uniform and feature-dependent TSM shows that the feature-dependent system does not improve upon the uniform system. A set of ten test sentences, shown in the Appendix, was used to compare the two TSM schemes. All ten sentences were processed several times with each of the two segmentation algorithms described in Section 5.1, and with varying threshold values. Sentence #5 (which has a particularly rich phonetic structure) was also segmented manually.

In general, the feature-dependent time-scale modified speech was either similar, or slightly worse, than the uniformly time-scale modified speech. This can be attributed to several factors. For simplicity, only the case of speech-specific segmentation (either automatic or manual) is discussed.

The Feature-Dependent TSM System

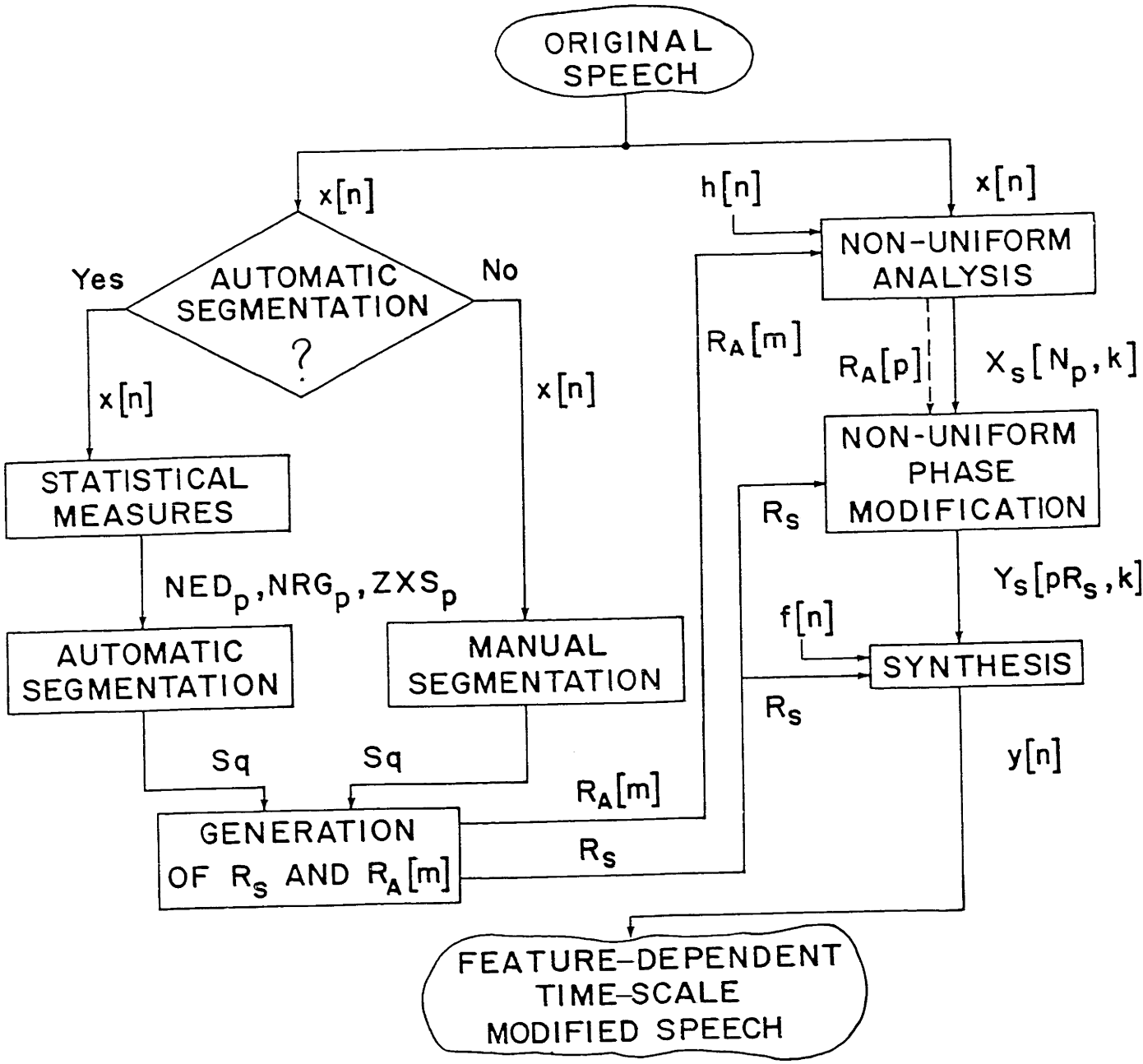


Figure 5.7

First, for the difference between uniform and feature-dependent TSM to be noticeable, the local scale factor during transition segments must be very close to unity. To obtain an overall scale factor β_0, the local scale factors during pause, voiced and unvoiced segments must be significantly farther from unity than β_0. The large difference in local scale factors causes the output speech to sound as if the speaker had stuttered. In addition, as the local scale factor departs from unity, the degradation of the output signal increases. Therefore, any gain in the quality of time-scale modified transitions is offset by a loss of quality during voiced and unvoiced portions of the output speech. Most listeners stated that the losses were greater than the gains.

Second, the perceived quality of the time-scale modified speech is somewhat different for different values of the scale factor β. The same effect is observed in Portnoff's system [1978, 1980]. Although this effect is difficult to notice between two time-scale modified sentences with different scale factors, it is easily perceivable when the value of the local scale factor is changing rapidly.

The experimental results obtained in this chapter indicate that feature-dependent TSM is not necessarily an improvement over uniform TSM. Clearly, not all possible schemes for making the scale factor β vary with changes in the local signal structure were evaluated. Whether an alternative feature-dependent TSM scheme that constitutes an improvement over uniform TSM can be

found, remains an open research question.

The fact that feature-dependent TSM is not necessarily an improvement over uniform TSM contradicts the results obtained by Toong [1974]. Using a Fairbanks (time-domain) type TSM system, and using manual segmentation, Toong found that feature-dependent TSM produced higher quality speech than uniform TSM. The experiments carried out by Toong, however, were different from the ones described in this thesis. First, the TSM technique used by Toong was a time-domain technique, which does not generate time-scale modified speech of a quality as high as that obtained with the TSM system based on the DSTFT. Therefore, Toong did not have a high-quality uniformly time-scale modified speech signal for comparison. Second, Toong did not compensate the approaches of the scale factor β to unity during transitions with corresponding departures of β from unity during the rest of the signal, to obtain an overall scale factor β_0. Therefore, the quality of the non-transition segments processed by Toong was identical in uniform and feature-dependent TSM but, in the latter case, the overall scale factor was $\beta_0 + \varepsilon$. Feature-dependent TSM using the system described in Part I of the thesis was carried out without this compensation, for comparison with Toong's results. In this case, feature-dependent TSM was slightly better than uniform TSM.

In Chapter 6, several other possible uses of the non-uniform TSM system are discussed.

CHAPTER 6

OTHER APPLICATIONS OF THE TSM SYSTEM

This chapter discusses three trial applications of the TSM system described in Part I. The purpose of this chapter is to suggest the overall usefulness of the TSM process, but not to derive any precise measurements of the performance of the TSM system.

Section 6.1 describes the result of time-scale modifying speech segments whose lengths are approximately one minute long, as opposed to the single sentences (about three seconds long) used to evaluate feature-dependent TSM. The overall quality of time-scale modified speech is informally evaluated and, in addition, the long-term stability of the phase modification algorithm developed in Section 4.3 is examined.

A simulation of the compression/expansion communications scheme suggested in Section 1.1 is discussed in Section 6.2. The overall quality of the recovered speech is shown to be poor beyond a 4-to-1 initial compression, although the speech retains some intelligibility up to a 10-to-1 initial compression.

Finally, Section 6.3 discusses the TSM process for signals other than speech. In particular, the result of applying the TSM system to music signals is evaluated.

6.1 TSM of Long Speech Segments

Although the TSM system developed in Chapter 4 is very efficient, its computational requirements are large. In its present implementation, the system processes a speech segment τ seconds long in approximately $10^3 \cdot \tau$ seconds, depending on the actual analysis and synthesis rates used. Therefore, the performance of the TSM system has been evaluated using single sentences which are about three seconds long. It is useful, however, to apply the TSM system to longer speech recordings, as they provide a better test of the overall quality of the time-scale modified speech. In addition, processing long speech segments allows us to verify the stability of the phase estimation algorithm (equations (4.75)-(4.81) and (4.103)) which, since it is implemented as a running sum, could become unstable.

In order to test the stability of the phase estimation procedure, the original algorithm is compared with a slightly modified version of itself. In the modified algorithm, the estimated time-unwrapped phase of $Y_s[pR_S,k]$ (given by equation (4.103)) is reset to the principal value phase of $X_s[N_p,k]$ whenever a pause is detected, as if the signal had just begun at the pause. Since the energy of the signal is negligibly small during a pause, any discontinuity in the phase of $Y_s[pR_S,k]$ that might be caused by the reset operation will have an imperceptible effect on the output signal. The detection of pauses can be performed either manually or automatically, as discussed in Chapter 5.

The time-scale modification of long speech segments was carried out both uniformly and non-uniformly. In the non-uniform case, the scale factor β was made to vary arbitrarily, to imitate the effect of a user speeding up or slowing down the speech at will. The varying scale factor β was implemented by means of the manual speech segmentation scheme developed in Chapter 5. In this case, however, the segmentation did not correspond to speech features but, rather, consisted of arbitrarily located regions where the scale factor gradually increased or decreased. In some cases, the scale factor β was even changed abruptly (from severe compression to severe expansion, or vice versa) to test the response of the system.

From the above experimental results, several observations can be made. First, the phase estimation algorithm is stable and need not be reset during pauses. Resetting the phase generally had no effect on the speech quality. In fact, in some cases, resetting the phase actually decreased the overall quality of the speech slightly. Second, it was found that semantically rich passages begin to be difficult to understand at twice their original speed. However, the level of understanding increases significantly when the listener has had previous experience listening to speeded-up speech. Finally, non-uniform TSM produces very high quality speech, despite the discontinuities in the scale factor β caused by the fact that the analysis and synthesis rates must take integer values between 1 and 40.

6.2 Compression and Expansion as a Communications Scheme

A communications scheme was proposed in Section 1.1 in which speech is compressed for transmission and then re-expanded at the receiver. The purpose of this scheme is to allow the simultaneous transmission of several speech signals over a channel that would otherwise be able to carry only one signal. This can be done by time-multiplexing a set of N signals, all of which have been previously compressed with a uniform scale factor $\beta = N$. The receiver can recover the transmitted signal by de-multiplexing, and expanding the speech with a scale factor $1/N$.

The compression/expansion part of this system was simulated. The original compression was carried out with N = 2,3,4,6, and 10. It was found that the quality of the recovered speech was very good for $N \leq 3$. Increasing to N = 4 and 6, the pitch information in the speech was quickly lost. However, even at N = 10 the recovered speech retained some intelligibility, although it sounded whispered. Since the compression/expansion technique works well only for $N \leq 3$, this communications scheme appears to have limited practical usefulness.

As an experiment, the speech was also processed in a reverse way; that is, the speech was first expanded and then compressed. Expansion rates of 2, 4 and 10 were tried. In this case, the quality of the recovered speech was excellent, regardless of the original expansion rate, although at the highest rate some reverberation was introduced.

6.3 Time-Scale Modification of Music Signals

Although the TSM system was designed for speech signals, it may be used to process signals other than speech. An obvious application is to music signals. Two different musical passages, each about 30 seconds long, were processed. The first passage was a classical guitar solo (Theme from the Etude No. 2 in B minor for guitar by Fernando Sor). All the notes were distinctly separated and, therefore, the signal was similar to voiced speech in that it was the output of a time-varying resonator (the guitar strings and body) excited with a quasi-periodic train of "impulses" (the action of the player's fingers). The second passage (Theme from the "Promenade" in "Pictures at an Exhibition" by Modeste Moussorgsky, orchestrated by Ravel) is played by a full orchestra and was chosen since it was very unlike speech.

The quality of the output music for both uniform and non-uniform TSM was excellent. However, it was found that the analysis window used for speech (a 256-point long Hamming window) did not have enough frequency resolution for music, so a 512-point Hamming window was used.

Two interesting effects were observed in time-scale modifying music signals. First, since the length of the analysis window was increased from 256 points to 512 points, its time resolution was decreased and, thus, the TSM process added some reverberation to the output music, particularly during expansion by a factor greater than 3. During compression, the timbre of the instruments changed slightly. This is due to the fact that the resonating time of the

instruments was compressed by the same scale factor as the overall signal. Moorer [1976, 1979] performed time-scale compression of signals generated by single musical instruments using a similar TSM system to the one developed in this thesis. Using linear predictive coding (LPC) techniques, however, he separated the resonant characteristics of the instruments from the action of the player. He then time-scale modified the player action by itself, and left the instrument characteristics intact, eliminating the timbre distortion problem.

Both music passages were also non-uniformly time-scale modified using the manual segmentation algorithm to determine the local value of the scale factor β (as in Section 6.1). Again, the integer nature of the analysis and synthesis rates did not cause perceptible discontinuities in the output signal, and the time-scale modified music was of high quality.

The next, and last, chapter summarizes the major results of this thesis and outlines suggestions for further research.

CHAPTER 7

CONCLUSIONS

7.1 Summary of Major Results

Part I of this thesis develops a non-uniform TSM system based on Portnoff's [1978] design. This system has several practical applications, including: a variable-speed speech playback system for the blind; a time-adjusting machine for the advertising industry; and normalization of the length of speech segments for voice recognition systems.

A general outline of the time-scale modification (TSM) process for speech signals is presented in Chapter 2. Using Portnoff's results, the speech is assumed to consist solely of voiced segments. The model of voiced speech is given by equations (2.20) and (2.21). Based on this model, the TSM process is described as a four-step procedure (Table I): Analysis, Linear Time-Scaling, Phase Modification and Synthesis.

Chapter 4 shows that the analysis and synthesis stages of the TSM system can be implemented by the corresponding DSTFT analysis and synthesis algorithms developed in Chapter 3 (Tables II and III). An explicit implementation of the linear time-scaling stage is developed in Section 4.2; later, in Section 4.4, it is shown that this stage can be implemented implicitly by allowing the analysis and synthesis rates to be different from one another. Section 4.3

then presents an algorithm for implementing the phase modification stage (Table IV).

The TSM system developed up to this point (figure 4.2) is a uniform system. This system is quite similar to Portnoff's system (figure 4.1), except that it has significantly smaller storage requirements since it uses the overlap-add synthesis algorithm developed in Chapter 3 (figure 3.8) and omits the explicit linear time-scaling stage.

A non-uniform TSM system is developed in Section 4.5. It is shown that the constant analysis rate can be replaced by a sequence of analysis rates which need not be the same; the synthesis rate, however, must remain constant. The analysis and phase modification algorithms are modified to accommodate the variable analysis rate. The non-uniform TSM analysis algorithm is given in Table V, and the complete non-uniform TSM system is shown in Table VI in operator notation. Figure 4.3 illustrates the structure of the non-uniform TSM system. This system was implemented in a general purpose mini-computer; this implementation is described in Section 4.6.

Part II of this thesis investigates several uses of the non-uniform TSM system developed in Part I. Feature-dependent TSM of speech signals is developed in Chapter 5. Both automatic and manual detection of speech features are described. It is shown that, in general, feature-dependent TSM is not an improvement over uniform TSM.

Finally, Chapter 6 evaluates the overall usefulness of the TSM system in three trial applications. Section 6.1 shows that

uniform and non-uniform time-scale modification of long speech segments produces high quality speech. The stability of the phase modification algorithm is also confirmed. Section 6.2 evaluates a compression/expansion communications scheme; the scheme is found to have limited practical usefulness. Finally, an application of the TSM system to music signals is discussed in Section 6.3.

7.2 Suggestions for Further Research

The non-uniform TSM system developed in Part I of this thesis has been shown to be successful for performing time-scale modification of both speech and music signals. However, there are several possible extensions to the system.

First, the current implementation of the system in a high-level language on a minicomputer is very slow; 1000 τ seconds are required to process a sentence τ seconds long. A much faster system could be implemented using a dedicated array processor or an assembly language program in a faster computer [Seneff, 1980]. The system could even be implemented in real-time (where a τ second long sentence is processed in τ seconds or less) with special purpose hardware and a pipelined structure.

Second, time-scale modification is only one possible application of the DSTFT analysis/synthesis algorithm. Other schemes for modifying the analyzed signal can be developed. For example, Seneff [1980] has shown that the spectral characteristics of the input signal can be modified by estimating the signal spectrum from its DSTFT representation and approximating the DSTFT

representation of the desired signal, which can then be synthesized by the DSTFT synthesis algorithm. Other modifications, such as low-bit coding of the DSTFT and feature extraction, could also be made.

Finally, alternative strategies for performing feature-dependent TSM can be developed. In particular, the scale factor β can be controlled by a heuristic algorithm that takes into account the semantic structure of the speech.

REFERENCES

Agrawal, A. and W. C. Lin, "Effect of Voiced Speech Parameters on the Intelligibility of PB Words," J. Acoust. Soc. Amer., vol. 57, no. 1, pp. 217-2, January 1975.

Aho, A. V., J. E. Hopcroft, and J. D. Ullman, The Design of Computer Algorithms, Reading, Mass.: Addison-Wesley Publishing Co., 1974.

Allen, J. B. and L. R. Rabiner, "A Unified Approach to Short-Time Fourier Analysis and Synthesis," Proc. IEEE, vol. 65, no. 11, pp. 1558-1564, November 1977.

Bloomfield, P., Fourier Analysis of Time Series: An Introduction, New York: John Wiley and Sons, 1976.

Fairbanks, G., W. L. Everitt, and R. P. Jaeger, "Method for Time or Frequency Compression-Expansion of Speech," IRE Trans., Professional Group on Audio, vol. AU-2, no. 1, pp. 7-12, January - February 1954.

Fairbanks, G., W. L. Everitt, and R. P. Jaeger, Recording Device, Washington, D.C., May 12, 1959, U.S. Patent No. 2,886,650.

Flanagan, J. L. and R. M. Golden, "Phase Vocoder," Bell Syst. Tech. J., 2nd ed., Berlin: Springer-Berlag, 1972.

Holtzman Dantus, S., "A Speech Pre-Emphasis Filter," S. B. Thesis, Dept. of Electrical Engineering and Computer Science, M.I.T., 1977.

Huggins, A. W. F., "More Temporally Segmented Speech: Is Duration or Speech Content the Critical Variable in its Loss of Intelligibility?" Quarterly Progress Report No. 114, Research Laboratory of Electronics, M.I.T., July 15, 1974, pp. 185-193.

Klatt, D. H., S. Holtzman Dantus, Private Communication, M.I.T., Cambridge, Mass., 1979.

Lee, F. F., "Time Compression and Expansion of Speech by the Sampling Method," J. Audio Eng. Soc., vol. 20, no. 9, pp. 738-742, November 1972.

Moorer, J. A., "The Use of the Phase Vocoder in Computer Music Applications," J. Audio Eng. Soc., vol. 26, no. 1/2, pp. 42-45, January/February, 1978.

Moorer, J. A., "The Use of Linear Prediction of Speech in Computer Music Applications," J. Audio Eng. Soc., vol. 27, no. 3, pp. 134-140, March, 1979.

Neuburg, E. P., "Simple Pitch-Dependent Algorithm for High-Quality Speech Rate-Change," 93rd Meeting Acoust. Soc. Amer., June 1977. Abstract, J. Acoust. Soc. Amer., vol. 61, suppl. no. 1, Spring 1977.

Nilsson, N. J., Principles of Artificial Intelligence, Palo Alto, Calif.: Tioga Publishing Co., 1980.

Oetken, G., T. W. Parks, and H. W. Scheussler, "New Results in the Design of Digital Interpolators," IEEE Trans. Acoust., Speech, and Sig. Proc., vol. ASSP-23, no. 3, pp. 301-309, June, 1975.

Oppenheim, A. V. and R. W. Schafer, Digital Signal Processing, Englewood Cliffs: Prentice-Hall, 1975.

Parsons, T. W., "Separation of Speech from Interfering Speech by Means of Harmonic Selection," J. Acoust. Soc. Amer., vol. 60, no. 4, pp. 911-918, October 1976.

Portnoff, M. R., "Implementation of the Digital Phase Vocoder Using the Fast Fourier Transform," IEEE Trans. Acoust., Speech, and Sig. Proc., vol. ASSP-24, no. 3, pp. 243-248, June, 1976.

Portnoff, M. R., "A Mathematical Framework for Time-Scale Modification of Speech," 93rd Meeting Acoust. Soc. Amer., June 1977. Abstract, J. Acoust. Soc. Amer., vol. 61, suppl. no. 1, Spring 1977.

Portnoff, M. R., "Time-Scale Modification of Speech Based on Short-Time Fourier Analysis," Sc.D. dissertation, Dept. of Electrical Engineering and Computer Science, M.I.T., 1978.

Portnoff, M. R., S. Holtzman Dantus, Private Communication, M.I.T., Cambridge, Mass., 1980.

Quatrieri, T. F., Jr., "Phase Estimation with Application to Speech Analysis-Synthesis," Sc.D. dissertation, Dept. of Electrical Engineering and Computer Science, M.I.T., 1979.

Rabiner, L. R. and B. Gold, Theory and Application of Digital Signal Processing, Englewood Cliffs: Prentice-Hall, 1975.

Rabiner, L. R. and R. W. Schafer, Digital Processing of Speech Signals, Englewood Cliffs: Prentice-Hall, 1978.

Schafer, R. W. and J. D. Markel (editors), Speech Analysis, New York: IEEE Press, 1979.

Schafer, R. W. and L. R. Rabiner, "Design and Simulation of a Speech Analysis-Synthesis Sytem Based on Short-Time Fourier Analysis," IEEE Trans. Audio Electroacoust., vol. AU-21, pp. 165-174, June 1973(a).

Schafer, R. W. and L. R. Rabiner, "A Digital Signal Processing Approach to Interpolation," Proc. IEEE, vol. 61, no. 6, pp. 692-702, June, 1973(b).

Scott, R. J. and S. E. Gerber, "Pitch Synchronous Time Compression of Speech," Proc. Conf. Speech Comm. Processing, pp. 63-65, April, 1972.

Seneff, S., "Speech Transformation System (Spectrum and/or Excitation) without Pitch Extraction," S.M.-E.E. Thesis, Dept. of Electrical Engineering and Computer Science, M.I.T., 1980.

Stevens, K. N., "The Role of Rapid Spectrum Changes in the Production and Perception of Speech," in Form and Substance, pp. 95-101, Copenhaguen: Akademisk Forlag, 1971.

Stockham, T. G., Jr., "High-Speed Convolution and Correlation," AFIPS Conf. Proc., 1966, Spring Joint Computer Conf. Reprinted in Digital Signal Processing, L. R. Rabiner and C. M. Rader (editors), New York: IEEE Press, 1972.

Toong, H. D., "A Study of Time-Compressed Speech," Ph.D. dissertation, Dept. of Electrical Engineering and Computer Science, M.I.T., 1974.

Tribolet, J. M. N. S., "Seismic Applications of Homomorphic Signal Processing," Sc.D. dissertation, Dept. of Electrical Engineering and Computer Science, M.I.T., 1974.

Weinstein, C. J., "Short-Time Fourier Analysis and Its Inverse," S.M. Thesis, Electrical Engineering Dept., M.I.T., 1966.

Wiener, N., Time Series, Cambridge, Mass.: The M.I.T. Press, 1949.

APPENDIX

TEST SPEECH PASSAGES

a) Sentences:

1. We made some fine brownies. (Female)
2. They took the crosstown bus. (Male)
3. The bowl dropped from his hand. (Female)
4. The chef made lots of stew. (Male)
5. Line up at the screen door. (Male)
6. He has the bluest eyes. (Male)
7. You are the biggest man. (Male)
8. Stuff those with soft feathers. (Female)
9. Those shoes were black and brown. (Male)
10. That shirt seems much too long. (Female)

b) Long speech passage:

When the sunlight strikes raindrops in the air, they act like a prism and form a rainbow. The rainbow is a division of white light into many beautiful colors. 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. People look, but no one ever finds it.

When a man looks for something beyond his reach, his friends say he is looking for the pot of gold at the end of the rainbow.

Throughout the centuries, men have explained the rainbow in various ways. Some have accepted it as a miracle without physical explanation. To the Hebrews, it was a token that there would be no more universal floods. The Greeks used to

imagine that it was a sign from the gods to foretell war or heavy rains. The Norsemen considered the rainbow as a bridge over which the gods passed from Earth to their home in the sky.

ND OF FILM

EASE REWIND


FILM
WIND