**Interruption techniques for efficient speech transmission**

This item was submitted to Loughborough University's Institutional Repository by the/an author.

**Additional Information:**


**Metadata Record:** [https://dspace.lboro.ac.uk/2134/34829](https://dspace.lboro.ac.uk/2134/34829)

**Publisher:** © Jeremy Scott Severwright

**Rights:** This work is made available according to the conditions of the Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International (CC BY-NC-ND 4.0) licence. Full details of this licence are available at: [https://creativecommons.org/licenses/by-nc-nd/4.0/](https://creativecommons.org/licenses/by-nc-nd/4.0/)

Please cite the published version.
INTERRUPTION TECHNIQUES
FOR EFFICIENT SPEECH TRANSMISSION
(PART 1 - INVESTIGATIONS)

by

JEREMY SCOTT SEVERWRIGHT, B.Tech.

A Doctoral Thesis submitted in partial fulfilment
of the requirements for the award of
Doctor of Philosophy of the Loughborough University of Technology

Supervisor: M. H. Ackroyd, Ph.D.
Department of Electronic and Electrical Engineering.
Acknowledgements

The author expresses his sincere thanks to Dr. M. H. Ackroyd, and Dr. G. B. Lockhart for their guidance and assistance during this work. The author also thanks Professor J. W. R. Griffiths, Head of the Department of Electronic and Electrical Engineering, for providing research facilities.
Summary

This thesis describes investigations into the use of interruption techniques for efficient speech transmission. Bit-rate compressions of up to about four times are achieved by regular transmission of short sections of digitally encoded speech - for example, sections of 10ms duration separated by intervals of 20 or 30 ms. An initial attraction of these techniques is that the interrupted speech is itself highly intelligible, although of very poor quality. At the receiver, the transmitted speech sections must be processed in order to synthesise a continuous speech signal of adequate quality.

In Part 1, investigations are described into simple synthesis techniques which operate by repeatedly outputting each transmitted section during the following interrupted period. From the effects of these techniques on speech quality, it is concluded that interruption is perceived aurally from the resulting temporal discontinuities in signal power, as well as from the associated spectral distortions.

Synthesis techniques using adaptive waveform prediction are then considered, to achieve further improvements in the reproduced quality of interrupted speech. These operate by creating an adaptive model of the speech signal from the received speech sections, and then using this to predict outputs during the interrupted periods. The predictor is described, and its performance illustrated by waveforms of processed speech and by informal judgements of its quality.

Digital computer simulation techniques have been used widely in these investigations, and provide valuable experimental flexibility. Their use is described in Part 2 of this thesis.
## Contents

Chapter 1. Introduction.  
1.1. Efficient speech transmission  
1.2. Interruption processing  
1.3. Previous work  
1.4. Scope of this project  

Chapter 2. Interruption processing  
2.1. Introduction  
2.1.2. Bit-rate reduction obtainable  
2.2. Theoretical discussion  
2.2.1. Non-rectangular time windows  
2.2.2. Non-recursive reiteration synthesis  
2.2.3. Recursive reiteration synthesis  
2.2.4. Reversed reiteration  
2.3. Experimental investigations  
2.3.1. Experimental details  
2.3.2. Interruption  
2.3.3. Non-recursive reiteration  
2.3.4. Recursive reiteration  
2.3.5. Reversed reiteration  
2.3.6. Preference evaluation tests  
2.4. Discussion  
2.5. Summary  

Chapter 3. Reiteration synthesis using adaptive waveform prediction  

Chapter 4. Pitch-synchronous reiteration synthesis  
4.2. 'Peakedness' pitch measurement  
4.2.2. Method of operation  
4.2.3. Performance testing  
4.2.4. Simulation of pitch-synchronous reiteration  
4.2.5. Results  
4.3. 'Periodicity' pitch measurement  
4.3.2. Autocorrelation methods  
4.3.3. Cepstrum methods  
4.3.4. Performance testing  
4.3.5. Simulation of pitch-synchronous reiteration  
4.3.6. Results  
4.4. Summary
Chapter 5. Prediction of speech amplitude

5.2. Minimum mean-square estimation
5.3. Rectified sum estimation
5.3.2. Results
5.4. Interpolation of speech amplitude
5.4.2. Results
5.5. Summary

Chapter 6. Prediction of amplitude spectrum

6.2. Prediction filter design
6.3. Comparison with least-squares error prediction
6.4. Summary

Chapter 7. Conclusion

References

Appendix A
Chapter I Introduction

1.1. Efficient Speech Transmission

Speech is Man's primary form of communication. In consequence, there has been a rapid growth in recent times in the use of telecommunication systems that directly convey speech. The provision of these facilities can, however, involve considerable expenditure. The 'CANTAT-2' submarine cable currently planned between Britain and Canada will cost £22M, to provide only 1840 speech circuits\(^1\), while the world-wide demand for telecommunications is currently growing at about 30% per annum\(^2\). Although it is hoped that satellite communication systems will compare favourably in cost with cables, these figures still provide a very powerful incentive to ensure that speech communication channels are used efficiently.

Conventionally the bandwidth allocated to a speech channel is approximately 3kHz, from 0.3kHz to 3.3kHz. However, Dudley\(^3\) in 1939 described the "vocoder", a coding device by which intelligible speech could be transmitted over a small fraction of this bandwidth. This demonstration implied that much of the speech signal is redundant. Flanagan\(^4\) has obtained an estimate of the degree of redundancy by comparing the information rate of the phonemic components of English speech (about 50 bits/s), with the information capacity of a typical speech channel (about 30k bits/s), which is about 600 times greater. Although such estimates of information rates are difficult to interpret meaningfully, the large disparity between these figures suggests that the speech time waveform is a highly redundant code for speech information.

Many speech coding systems have been suggested to achieve more efficient and economic communication. Perhaps the best known is the channel vocoder\(^5\) which achieves bandwidth compressions of the order of 10:1 by transmitting the speech spectral envelope, with separate 'pitch' information to allow the fine spectral detail in the original speech to be reinserted at the receiver.

1.2. Interruption Processing

This thesis describes investigations into a relatively simple class of speech
transmission systems using interruption techniques \(^{(52)}\). These reduce channel occupancy by periodically interrupting the speech signal at a low rate in comparison to its bandwidth, and are particularly suited to digital transmission methods. Fig. 1.2.1 illustrates the waveforms occurring during interruption of 100ms of speech. Simple time-domain techniques such as these suggest themselves because, although speech signals are complex, their characteristics tend to change relatively slowly. This implies that suitable waveforms for interrupted periods may be generated from the remaining signal sections. These gradual changes may be seen in Fig. 1.2.2, illustrating a 1s section of speech waveform. At the receiver, suitable waveforms would be generated to fill the interrupted periods, in order to reproduce speech signals of adequate quality.

An initial attraction of interruption techniques for speech transmission is that the interrupted speech is highly intelligible in itself, although of very poor quality. In an early paper describing the effects of interruption on speech, Miller and Licklider \(^{(6)}\) reported intelligibility scores above 90% under similar processing conditions to those employed currently, which suggests that these techniques might provide the basis for simple speech communication systems. In fact, as will be shown later, their use can avoid many of the performance problems encountered in more complex systems.

Two important parameters of the interruption process are the durations of the transmitted and interrupted sections of speech. These are respectively named the transmission period \(T_t\), and the interruption period, \(T_i\). The processing period, \(T_p\), refers to the duration of a complete cycle of the interruption process, and may be expressed:

\[
T_p = T_t + T_i \quad (1.1)
\]

The corresponding processing frequency, \(F_p\), may be written:

\[
F_p = \frac{1}{T_p} \text{ Hz} \quad (1.2)
\]

The transmission ratio, \(R\), expresses the proportion of signal retained during interruption:

\[
R = \frac{T_t}{T_p} \quad (1.3)
\]
Fig. 1.2.1. Waveforms during interruption of 100 ms Speech.

[a] Original waveform.
[b] Transmission function (Tt, Ti = 10 ms).
[c] Interrupted waveform.
Fig. 12.2. Waveform of 1 second of Speech.
The quantities $T_p$, $T_t$, and $T_i$ are illustrated in Fig. 1.2.1. Typically, the transmission period is of the order of 10mSec, and a transmission ratio of $\frac{1}{4}$ is apparently feasible. The bit-rate compression achieved by interruption processing is equal to the transmission ratio, and could be exploited in a number of ways. The speech signal in a transmission period might be stored for slower transmission over a whole processing period, for example, or other signals might be multiplexed through the transmission channel during interruption.

1.3. Previous Work

The application of interruption techniques for efficient speech transmission was considered in a number of early papers. Poirson (7) in 1920, and Marro (8) in 1936 suggested the use of a single channel for two-way communication by regular alternation between transmission and reception about 20 times per second. Montani (9) proposed a similar communication system in 1946 with a switching frequency of 15 Hz. In these systems, the interrupted speech signals were output from the receiver without any further processing. Montani reported that interruption had little effect on speech quality because of the 'persistence of hearing'. This finding has unfortunately not been substantiated in current investigations, in which low-frequency interruption has been found to cause severe degradation in speech quality.

Later in 1946, Gabor (10) proposed an interrupted speech system in which the transmitted speech sections were weighted with a Gaussian window to reduce spectral distortion, and were also re-output, or 'reiterated' during interrupted periods to reduce the discontinuities in signal power resulting from interruption. This was the first paper to consider the effects of interruption processing in any depth, but because of severe problems of implementation only limited practical investigations were performed. Gabor's system has since been simulated on a digital computer (11).

The first detailed practical investigations into interruption processing were reported by Miller and Licklider (6) in 1950. They found that the intelligibility
of interrupted speech could exceed 90% when a processing frequency in the range 30 to 100 Hz was used, with a transmission ratio of 1/2. However, they also noted the severe subjective distortions introduced during interruption, and described this as a 'warble'-like effect, tending to a 'buzz' at higher processing frequencies. In 1954 Fairbanks, Everitt and Jaeger(12) described the application of interruption techniques for 'compressed' tape recording of speech. Magnetic tape records of speech were manually interrupted by regularly splicing out short pieces, to produce tapes in which the retained pieces corresponded to the transmitted sections of interrupted speech. These were reproduced by a tape machine with a rotating head assembly which reiterated each portion of the tape several times to fill the time originally 'interrupted' by splicing. "Fair" quality was reported after tape compressions of 40%. This was the earliest practical investigation of the effects of reiteration synthesis on interrupted speech.

In 1956, David and McDonald(14) investigated 'pitch-synchronous' interruption and reiteration of speech. The transmission period was made equal to the fundamental, or 'pitch' period of voiced speech, and the transmitted sections were repeatedly reiterated after delays of one pitch period. It was shown that periodic waveforms could be reproduced without distortion by this technique, and the authors reported good quality reproduction with a transmission ratio of only 1/6. However, the scope of this investigation, and of most previous ones, was limited by practical problems. In particular, the audio-frequency delays required for reiteration synthesis were difficult to implement by existing analogue techniques. In this investigation, it was necessary to employ constant-pitch speech signals synthesised by a channel vocoder, as variable-delay devices were not available.

In 1959, Sharf(15) published detailed findings on the effects of reiteration synthesis techniques on interrupted speech. He found that reiteration had little effect on the intelligibility of interrupted speech, but included only very little information about its effect on speech quality. In these investigations it was
found necessary to use special magnetic and electrostatic recorders to obtain the signal delays required in the reiteration process. A novel interruption method was reported by Subrahmanyam and Peterson (16), also in 1959, in which successive frequency bands of the speech signal were transmitted sequentially.

In the transmitter the speech signal was passed through a continuously swept band-pass filter, and the filtered signal heterodyned down for narrow band transmission. In the receiver it was modulated to its original frequency band with a variable frequency oscillator swept in synchronism with the band-pass filter in the transmitter. This 'time-frequency scanning' is comparable with interruption processing as particular frequencies are only transmitted for a small fraction of time, but are transmitted sequentially rather than together as in the interruption processes considered here. An attraction of this scheme was that speech was converted directly into a baseband signal for simple transmission through conventional channels. However, the authors reported poor performance with speech signals.

In 1963, Indiresan (17) gave a good résumé of the current state of interruption processing for speech communication. Up to that time, many basic forms of interruption processing had been investigated but no systems had been developed to the level of general practical application. Many papers had reported detailed measurements of speech intelligibility after various forms of interruption processing, and had shown that it could exceed 90%. However, very few details were available about the quality of the processed speech, which was apparently generally poor. Since that time interest in interruption processing has apparently lapsed although some associated forms of processing have been investigated more recently.

In 1967, Stover (18) described a speech communication system in which only the first 3ms approximately of about every fifth pitch period was transmitted after infinite clipping, and accompanied by pitch information. The transmitted sections were pitch-synchronously reiterated in the receiver, and it was claimed
that bit-rate compressions of 20:1 could be achieved with speech intelligibility of about 70%. This coding technique is comparable in some aspects with the channel vocoder: information about the speech short-term amplitude spectrum is sent in both cases, either within a 3ms section of time waveform or as a set of channel signals conveying instantaneous spectral amplitudes. Pitch information is also transmitted, and used to reinsert periodicity, and corresponding fine spectral detail, within the receiver. In 1968, Beetle and Chapman\(^\text{19}\) described a random interruption scheme for efficient speech storage within a digital computer. Magnetic records of interrupted speech were prepared, containing information about the durations of successive transmission and interruption periods, and were reproduced by a reiteration process. The system could apparently reproduce speech of good quality after 30% compression.

Associated time-domain techniques have been investigated to improve the intelligibility of 'helium speech' from deepsea divers. Because of the increased speed of sound in the high pressure helium-oxygen breathing mixtures used, the formant frequencies of the diver's speech may be raised by two or three times above their usual values, causing a severe 'Donald Duck'-like distortion. Stover\(^\text{20}\) and Gill, Morris and Edwards\(^\text{21}\) have proposed systems which retain only the initial part of each pitch period of the diver's speech, and output this at a slower rate over the whole pitch period to effectively reduce the formant frequencies by the required ratio.

 Interruption techniques have been used for time-compression of speech recordings\(^\text{22}\), through which the average word rate may be increased by about 100% while leaving the character of the original speech largely unaltered. The compression process involves interruption with a transmission period of about 30ms, and transmission ratio between 1 and about 0.5 depending on the compression required. The retained speech sections are then re-recorded without interruptions to produce a compressed-speech recording of reduced overall duration. Compressed-speech records have been used for programs of rapid learning, and to enable blind people
to 'speed hear' recorded books.

Very low frequency interruption techniques have been applied successfully in the TASI (Time Assignment Speech Interpolation) system (23), currently in use on some transatlantic submarine cables. The system takes advantage of the fact that during telephone conversation a speaker might talk for only 40% of the time, and occupy the remaining 60% by listening or waiting for a call to be connected. A speaker is only allocated a transmission channel while actually speaking, and the channel is disconnected for re-allocation during pauses. By this method, 36 conversations may be satisfactorily transmitted through only 20 channels.

1.4. Scope of this Project

These investigations were conducted to study new techniques for interruption-processing, and to assess their use for efficient speech transmission.

In Chapter 2, the effects of some simple forms of interruption and reiteration are discussed theoretically, and are compared with the corresponding subjective distortions introduced in speech signals. From these comparisons, an improved type of reiteration synthesis using adaptive techniques is proposed.

Chapter 3 introduces these adaptive reiteration techniques. The reiteration process is considered as a form of waveform prediction, whereby signals for each interrupted period are generated from adjacent sections of transmitted signal. The requirements of the adaptive waveform predictor are discussed in terms of a model of the vocal mechanism. Chapters 4, 5 and 6 describe various aspects of the predictor design which take into account the speech fundamental period, and changes in the amplitude of glottal excitation and in the short-term speech spectrum.

This work deals with interruption processes, and the means by which a continuous speech signal might be reproduced; the techniques by which the interrupted speech might be transmitted are not considered. However, to facilitate its transmission, all the systems investigated here operate with fixed transmission and interruption periods. This eliminates the need to buffer the retained sections of waveform before transmission, as was necessary in some
earlier forms of pitch-synchronous interruption processing which used variable
transmission periods \( (10, 11, 14) \).

These investigations have become feasible with the advent of digital
signal processing techniques. Operations such as delay, and multiplication by
transmission windows may now be implemented conveniently and accurately. The
processes investigated here have all been considered in terms of sampled data
operations, with a view to digital implementation. An advantage of this approach
is that these processes may be readily investigated by simulation on a digital
computer, and all experimental investigations have currently been performed in
this way.

Simulations have been performed mostly on a C.T.L. Modular-I computer in the
Department of Electronic and Electrical Engineering. It has on-line analog-to-
digital and digital-to-analog converters, and can operate directly with analogue
input and output signals. Most of this work has been programmed in 16-bit fixed
point arithmetic for greatest computational speed. This allowed many forms of
interruption processing to be simulated in real time. It was necessary to develop
special-purpose programs and subroutines for these applications. These execute
signal processing functions, and also arrange efficient input and output of
analogue signals. Also a set of programs has been developed to allow large
amounts of sampled data to be transferred efficiently on paper tape to the
University Computer Centre, which has an ICL 1904A computer. These programs
enable speech signals to be digitised on the Modular-I for convenient off-line
processing in the Computer Centre, and subsequent conversion to analogue form
again on the Modular-I computer. Further details of these facilities are given
in Appendix A, and some of the computer programs are documented in Part 2 of this
thesis.
Chapter 2  

**Interruption Processing**

2.1. Introduction

This chapter describes investigations into some simple speech transmission systems employing time-invariant interruption and reiteration processes. These are considered theoretically in section 2, and the distortions introduced in speech signals by these processes are described in section 3. By comparison of the theoretical and subjective effects, a deeper insight is gained into the aspects of the interruption process most responsible for its subjective distortions. These findings lead to the proposal of an improved form of adaptive reiteration synthesis, which is then considered in Chapter 3. In this work, emphasis is placed on the 'quality' of processed speech rather than its intelligibility. As mentioned previously, intelligibility is normally well retained during interruption processing whereas quality can differ widely, and is of greater importance when assessing practical applicability. In this connection, the use of simple interrupted-speech transmission systems would be very attractive if adequate quality was obtained because of the possible economy and simplicity of implementation by modern digital techniques.

The investigations described in this chapter deal mainly with the effects of non-rectangular transmission windows, and various reiteration synthesis techniques on the quality of interrupted speech. The transmission window employed in the interruption process controls the nature of the spectral distortion introduced. As will be shown later, its extent can be reduced by using smooth-shaped transmission windows having narrow spectra with most energy concentrated around zero frequency. The effects of non-rectangular transmission windows have been only briefly considered previously. Trapezoidal windows have been investigated,\(^{(6, 15)}\) but the possible advantages of smooth transmission windows for reducing interruption distortion have been discussed only by Gabor\(^{(10, 11)}\) in an early work. Although they can lessen the spectral distortion resulting from interruption, the regular alternations in signal power also introduced by interruption still remain. Therefore, variations in subjective speech quality
obtained by use of non-rectangular transmission windows may be associated largely with the corresponding variations in spectral distortion, rather than temporal distortions of signal power. Fig. 2.1.1. illustrates the effect on a speech waveform of interruption with a Hamming transmission window.

In contrast, the effects of time-invariant reiteration synthesis techniques are to reduce the temporal variations in signal power introduced by interruption, rather than reduce the resulting spectral distortion. Any subjective improvements achieved by reiteration techniques may therefore be associated with the corresponding modifications in these temporal effects. Three types of reiteration synthesis have been investigated: a non-recursive type as employed in most previous investigations (15), from which each transmitted signal section is only output a finite number of times; a recursive, infinite memory type in which the output section is formed by adding the transmitted signal section, if present, to the last attenuated output section; and a 'reversed' type which is comparable to non-recursive reiteration but in which the output signal section is time-reversed about its centre between successive reiterations. Fig. 2.1.2. illustrates the effects of these reiteration processes on a section of interrupted speech waveform.

In previous investigations the subjective distortions resulting from interruption processing have been explained very largely in terms of the accompanying spectral distortion. The temporal distortion of signal power introduced by these processes, however, can vary greatly according to the interruption and reiteration syntheses technique employed, yet its subjective effect has been only briefly considered (10,15). Current findings suggest that this must be taken into account if the varying subjective distortions of different interruption processes are to be better understood, and further improved systems proposed.

Before considering these processes in detail, the following section considers some limits on the bit-rate reduction which may be practically obtained by interruption processing.
Fig. 2.1.1. Speech Waveforms during Interruption with Rectangular and Hamming Transmission Functions.

[a]. Original speech waveform (0.1 s).
[b]. Rectangular transmission function (T1, Ti = 20 ms).
[c]. Speech waveform interrupted by (b).
[d]. Hamming transmission function.
[e]. Speech waveform interrupted by (d).
Fig. 2.1.2. Speech Waveforms during Interruption and Reiteration Synthesis.

[a]. Original waveform (0.1s).
[b]. Interrupted waveform (Tt = 10ms, Ti = 20ms).
[c]. Waveform [b] after non-recursive reiteration.
[d]. Waveform [b] after recursive reiteration (A = 0.5).
[e]. Waveform [b] after reversed reiteration.
2.1.2. Bit-Rate Reduction Obtainable

The bit-rate reduction obtained by interruption processing is equal to 
(Transmission period)/(Processing period), i.e. the transmission ratio R. The 
greatest bit-rate reduction is therefore achieved by use of the shortest 
transmission period with the largest useable processing period. Some lower limits 
on the transmission period will now be considered.

The interrupted speech signal $s_i(t)$ may be considered to be the speech 
signal $s(t)$ multiplied with a periodic low frequency transmission function $m(t)$ 
which is zero during interrupted periods. The interrupted signal $s_i(t)$ is given by:

$$s_i(t) = s(t) \cdot m(t) \quad (2.1.1.)$$

This operation results in convolution of the Fourier Transform of the speech 
signal, $S(f)$, with the relatively narrow Fourier Transform of the periodic 
transmission function, $M(f)$. The Fourier Transform of the interrupted speech 
signal, $S_i(f)$, is given by:

$$S_i(f) = \int_{-\infty}^{\infty} S(f-f') M(f') \, df'$$

or

$$S_i(f) = S(f) \star M(f) \quad (2.1.2.)$$

$S(f)$ suffers a limited 'spreading', depending on the form of $M(f)$. To 
estimate $M(f)$, it is useful to consider the transmission function $m(t)$ as a 
succession of individual transmission windows $w(t)$ having duration of the 
transmission period $T_t$, and being applied regularly to the speech signal at 
intervals of the processing period $T_p$. The transmission function $m(t)$ may be 
expressed as the convolution of $w(t)$ with an impulse train of period $T_p$. That is:

$$m(t) = w(t) \star \sum_{n = -\infty}^{\infty} \delta(t - n T_p) \quad (2.1.3.)$$

where $\delta$ signifies the Dirac delta function. Taking Fourier transforms of 
equation (2.1.3.),
\[ M(f) = W(f) \sum_{n = -\infty}^{\infty} \delta(f - n/T_p) \quad (2.1.4.) \]

\( M(f) \) consists of impulses at intervals of \( 1/T_p \) Hz, whose areas are specified by \( W(f) \). As \( T_p \) is normally at least twice \( T_t \), several spectral impulses would lie within the frequency range for which \( W(f) \) exhibited significant magnitude, and the envelope of their areas would have approximately the same form as the Fourier transform of a signal transmission window, \( W(f) \). Therefore the width of the spectral 'spreading' indicated by equation (2.1.2.) depends largely on the spectral width of the transmission window employed. In the case of a rectangular window, for example, whose Fourier transform may be expressed:

\[ W_r(f) = \sin(\pi f T_t) / \pi f, \quad (2.1.5.) \]

where \( w_r(t) = 1, \quad |t| \leq T_t/2 \)

\[ = 0, \quad |t| > T_t/2 \]

the main spectral peak is roughly \( 1/T_t \) Hz wide, and spectral detail within such an interval is largely lost. Short transmission periods, therefore, tend to cause severe distortion of the speech spectrum, and consequential degradation of the processed speech.

Miller and Licklider\(^{(6)}\) reported that the intelligibility of interrupted speech rapidly declined if the transmission period was reduced below 3ms. More recently, Stover\(^{(18)}\) has also found that a transmission period of 3ms was necessary for reproduction of intelligible speech. This figure constitutes an approximate lower bound on the transmission period if intelligibility is to be retained. Under such processing conditions, the gross characteristics of the speech spectrum which apparently convey intelligibility\(^{(5)}\) would be only slightly affected. However, any spectral detail within an interval of about 300Hz, including individual harmonics of the glottal excitation, would be lost. In the current application, it is required that pitch information is retained in the transmitted sections, so that speech pitch and 'character' may be reproduced to some extent by simple
synthesis techniques. The width of the main spectral peak of the transmission window must therefore be less than the fundamental pitch frequency. In the case of a rectangular window this may be approximately stated:

$$\frac{1}{T_p} < \frac{1}{T_{pitch}} \text{ Hz}$$

i.e. $$T_p > T_{pitch} \text{ s}$$

where $$T_{pitch}$$ = period of pitch excitation.

As approximately 99% of pitch periods in conversational speech are shorter than 12 ms (24) (see Chapter 4.1), a transmission period of this value would satisfy condition (2.1.6.) under almost all conditions of speech. The values of transmission period employed in current investigations have mostly been of the order of 12 ms.

The upper practical limit on the processing period depends on the rate at which speech characteristics are modified. Rader (25) has measured the power spectra of vocoder channel signals, and found that 99% of the channel signal power lay below 25 Hz, implying that a sampling frequency of 50 Hz would be adequate for their transmission. This suggests that a processing frequency of 50 Hz would also allow adequate reproduction of speech characteristics from successive transmitted speech sections. Some distortions would still occur at this processing frequency if simple reiteration synthesis techniques were used as the reproduced speech spectrum would not be changed continuously but in steps coinciding with the transmission of successive signal sections. These effects would usually be minor, however, as speech spectra are largely controlled by the comparatively slowly changing mechanical configuration of the vocal tract and would not normally change significantly within a 20 ms period. Furthermore, it is feasible that even longer processing periods could be used without great reduction in quality. Miller and Licklider (6) found that interrupted speech retained high intelligibility even when the processing period was as long as 100ms. The high word intelligibility under these conditions implies furthermore that the intelligibility of individual phonemes also remains high. In the case of voiced phonemes, which have average duration of about 300 ms (26) or about three processing periods, this might be expected.
However, plosive phonemes, whose perception is also important, have much shorter durations of about 25 ms\(^{(27)}\). Some would therefore be completely eliminated by interruption with 100ms processing period. Their intelligibility under these conditions may be explained by the findings of Sharf and Hemyer\(^{(28)}\) which indicate that perception of consonants depends on the much slower formant transitions between those of the consonant, and the adjoining phonemes. Longer processing periods than 20 ms may therefore be feasible, the figure of 100ms setting an approximate upper limit if intelligibility is to be retained. With a transmission period of 12ms these figures indicate that a bit-rate compression of \(\frac{1}{3}\) could be achieved without significant impairment of speech quality, while bit-rate compressions of about \(\frac{1}{4}\) might be feasible. This figure has not been attained during current investigations, however. The largest useable processing period with currently investigated synthesis techniques is about 50ms, enabling bit-rate compressions of about \(\frac{1}{4}\) to be achieved.

2.2. Theoretical Discussion

2.2.1. Non-Rectangular Time Windows

As shown in the last section (equation 2.1.2), the effect of interruption is convolutive in the frequency domain. The Fourier transform \(S(f)\) of the speech signal becomes convolved with the Fourier transform \(M(f)\) of the transmission function. The Fourier transform \(S_f(f)\) of the interrupted speech signal may be written:

\[
S_f(f) = S(f) \ast M(f)
\]  

It was also shown that the spectral envelope of the transmission function was similar in form to the spectrum of the transmission window used. To minimise the spectral distortion resulting from interruption, therefore, the transmission function should employ a window whose spectral energy is concentrated at low frequencies, and as far as possible in the main peak at zero frequency. The following windows were used in transmission functions (listed here with Fourier transforms):
1) **Rectangular**

\[ w_r(t) = \begin{cases} 1, & |t| \leq \frac{T}{2} \\ 0, & |t| > \frac{T}{2} \end{cases} \]  

(2.2.2.)

\[ W_r(f) = \frac{T}{\pi f T} \left( \sin(\pi f T) / \pi f T \right) \]

2) **Triangular**

\[ w_t(t) = \begin{cases} 1 - 2 \frac{|t|}{T_t}, & |t| \leq \frac{T}{2} \\ 0, & |t| > \frac{T}{2} \end{cases} \]  

(2.2.3.)

\[ W_t(f) = \frac{T}{2} \times \left( \sin(\pi f T_t/2) / \pi f T_t \right)^2 \]

3) **Raised Cosine**

\[ w_c(t) = 0.5 \left( 1 + \cos(2\pi t / T_t) \right), \quad |t| \leq \frac{T_t}{2} \]

(2.2.4.)

\[ W_c(f) = \frac{T}{2} \times \left( \sin(\pi f T_t/2) / \pi f T_t \right) \left( 1 - (f T_t)^2 \right)^{-1} \]

4) **Hamming Raised Cosine**

\[ w_h(t) = 0.54 \times 0.46 \cos(2\pi t / T_t), \quad |t| \leq \frac{T_t}{2} \]

(2.2.5.)

\[ W_h(f) \text{ is similar to } W_c(f) \text{ above, but has higher concentration of energy (99.96%) in the main peak.} \]

These transmission windows, with their Fourier transforms, are shown in Fig. 2.2.1. The spectra of the non-rectangular windows (2) - (4) are more concentrated around their main peaks than for rectangular window (1). Their main peaks are wider, however, having widths of about $2/T_t$ Hz in comparison to about $1/T_t$ Hz for the rectangular window. These non-rectangular transmission
Log_{10} (magnitude of Fourier Transform).

Fig. 2.2.1. Transmission windows, with log (magnitude) of their Fourier transforms.
windows could therefore cause greater distortion of fine spectral details such as the individual harmonics of the pitch excitation function, but would greatly reduce the wide-band spectral distortion introduced by a rectangular window. This might reduce the 'harshness' in interrupted speech. Fig. 2.2.2. illustrates the nature of the spectral distortion introduced during interruption. It shows the amplitude spectra obtained by interruption of a sinusoidal signal at increasing frequencies, with rectangular and Hamming raised cosine transmission windows. The effect of the Hamming transmission window in limiting the spread of the spectral distortion may be clearly seen.

2.2.2. Non-Recursive Reiteration Synthesis

Non-recursive reiteration synthesis, as employed in most previous investigations of reiteration techniques, operates by re-outputting transmitted signal sections repeatedly to fill immediately following interrupted periods. The power in the reproduced signal is then made approximately continuous in time, rather than localised in the transmitted sections of the interrupted signal. If transmission and interruption periods of $T_t$ and $T_i$ respectively are considered, where

$$T_t = N_t T_s$$  
(2.2.6.)

and

$$T_i = N_i T_s = MN_t T_s$$  
(2.2.7.)

the reiteration process involves the repeated output of each transmitted section $M$ times, after successive delays of $T_t$. This is performed by a linear non-recursive filter having transfer function $H(z)$ where:

$$H(z) = 1 + z^{-N_t} + z^{-2N_t} + \ldots \ldots + z^{-MN_t}$$  
(2.2.8.)

where $z^{-1}$ denotes a delay of a sample interval $T_s$. This expression may be factorised:

$$H(z) = (1 - a(1)z^{-1})(1 - a(2)z^{-1}) \ldots (1 - a(MN_t)z^{-1})$$  
(2.2.9.)

It is useful here to examine the locations of the zeros of $H(z)$, given by $a(1)$ to $a(MN_t)$. Their locations may be simply established, as follows.

If $|z|$ is less than 1, the transfer function $H_1(z)$ where
Fig. 2.2.2. Illustration of Spectral Distortion introduced during Interruption \( T_l, T_i = 12 \text{ms} \).
may be expanded:

\[ H_1(z) = 1 + z^{-N_t} + z^{-2N_t} + \ldots \quad (2.2.11.) \]

The required transfer function \( H(z) \), of equation 2.2.8., consists of the first \((M+1)\) terms of \( H_1(z) \). The higher order terms may be cancelled out by introducing an extra term:

\[ H(z) = H_1(z) - z^{-N_t}(M+1)H_1(z) \quad (2.2.12.) \]

\( H(z) \) can therefore be expressed:

\[ H(z) = \frac{1 - z^{-N_t}(M+1)}{1 - z^{-N_t}} \quad (2.2.13.) \]

The numerator and denominator polynomials both have the same general form:

\[ H_2(z) = 1 - z^{-n} \quad (2.2.14.) \]

The zeros of \( H_2(z) \) may be located by substituting:

\[ z_1^n = z \quad (2.2.15.) \]

Then

\[ H_2(z_1) = 1 - z_1^{-1} \quad (2.2.16.) \]

\( H_2(z_1) \) exhibits a single zero in the \( z_1 \)-plane at \( z_1 = (1,0,0) \). The zero locations of \( H_2(z) \) are then found by noting that:

\[ n \cdot z = z_1 \quad (2.2.17.) \]

The angular interval of \( 2 \pi \) in the \( z_1 \)-plane is mapped into the angular interval \( 2 \pi/n \) in the \( z \)-plane. \( H_2(z) \) therefore exhibits zeros in the unit circle, as did \( H_2(z_1) \), but at angles \( p(\ell) \) where

\[ p(\ell) = 2 \pi/n, \quad \ell = 0,1,2, \ldots (n-1) \quad (2.2.18.) \]

This result allows the zero locations of \( H(z) \) to be established from equation 2.2.13. The numerator exhibits zeros at angles \( q(\ell) \) where

\[ q(\ell) = 2 \pi/(M+1)N_t, \quad \ell = 0,1, \ldots ((M+1)N_t -1) \quad (2.2.19.) \]

Similarly, the denominator exhibits zeros at angles \( r(\ell) \) where

\[ r(\ell) = 2 \pi/N_t, \quad \ell = 0,1, \ldots (N_t-1) \quad (2.2.20.) \]

This shows that the zeros of \( H(z) \) lie on the unit circle of the \( z \)-plane at angles
given by \( q(i) \), except for every \( i \) in \((M+1)\) which is cancelled by the denominator zeros \( r(i) \). By combining these results, the angles \( \phi \) of the zeros of \( H(z) \) may be written:

\[
\phi(i,j) = 2\pi/N_t \cdot \left( 1 + j/(M+1) \right)
\]

where \( j = 0, 1, 2, \ldots, (N_t-1) \) and \( i = 1, 2, 3, \ldots, M \) (2.2.21)

A typical pattern of these zero locations is illustrated in Fig. 2.2.3.

The reiteration process exhibits response peaks at angular frequencies \( r(i) \) of the cancelled zeros, but with little response in between. These peaks become narrower as \( M \) is increased. The peaks occur at frequencies \( f(i) \) where:

\[
f(i) = r(i) \cdot (1/2\pi T_s)
\]

(2.2.22.)

As the reiteration process exhibits this periodic 'comb' frequency response, its effect on a transmitted signal section depends on the nature of the waveform within it. It may be deduced intuitively that if a transmitted section contained a whole number of cycles of a periodic waveform, the signal sections would all join up exactly during reiteration and the waveform would be reproduced without distortion. Otherwise serious distortion could result.

These effects may be best examined in the frequency domain. From equation 2.2.19, the zeros of the reiteration process lie at frequencies \( f_z(i) \) where

\[
f_z(i) = q(i) \cdot (1/2\pi T_s)
\]

(2.2.23.)

except at frequencies \( f(i) \) of the cancelled zeros, given by equation 2.2.22.

During periodic interruption with transmission and interruption periods of \( T_s \) and \( M T_s \) respectively, the processing period is \((M+1)T_s\). Distortion sidebands are generally introduced around signal frequencies with spacing \( \Delta f \) where

\[
\Delta f = 1/(M+1) T_s
\]

(2.2.24.)

By comparison with equation 2.2.23, the spacing of the interruption distortion sidebands is seen to be equal to the spacing of the zeros of the reiteration process.
Z - Transfer function of non-recursive reiteration process:

\[ H(z) = 1 + z^{-N} + z^{-2N} + \cdots + z^{-MN} \]

Zeros of \( H(z) \) lie on unit circle of \( z \)-plane at angles:

\[ \Theta_{i,j} = \frac{2\pi}{N} \left( i + \frac{j}{(M+1)} \right), \quad i = 0, 1, 2, \ldots N-1 \]
\[ j = 1, 2, 3, \ldots M \]

Fig. 2.2.3. Zero Locations of Non-recursive Reiteration Process.
If a sinusoidal signal of frequency $1/T_t$ (1 integer), corresponding to a response peak $f(i)$ from equation 2.2.22, were interrupted and reiterated the frequencies of
the interruption sidebands would therefore coincide with the zeros of the
reiteration process. The distortion sidebands would be eliminated, except at
frequencies $f(i)$ of the other response peaks where the zeros are cancelled.
However, the effects of the other response peaks may be avoided if a rectangular
transmission window is used in the interruption process, whose Fourier transform is
$$\sin(\pi f T_t)/\pi f.$$ This has zeros at multiple frequencies of $1/T_t$ Hz, except at
zero frequency. As a result, distortion sidebands do not appear at intervals of
$1/T_t$ Hz to pass through the adjacent response peaks of the reiteration process.
Under this condition, the spectral distortion introduced by interruption is
completely eliminated by reiteration and the original signal, of frequency $1/T_t$, is
reproduced without distortion. In the time domain, the transmitted signal sections
contain a whole number of periods which therefore join up exactly after reiteration.

If the signal frequency were not an integer multiple of $1/T_t$ Hz, or a non-
rectangular transmission window were used for interruption, the reiteration process
would not eliminate the interruption distortion sidebands and the original signal
would not be reproduced exactly. In particular, if the signal frequency coincided
with one of the $M$ zeros between each response peak of the reiteration process the
signal would be completely lost and only the interruption sidebands would remain.

A time-invariant reiteration process of this kind could not eliminate
interruption distortion in speech signals, whose frequency spectrum is constantly
changing and which is not limited to multiples of $1/T_t$ Hz. The use of non-
rectangular transmission windows might therefore be advantageous with reiteration
synthesis, to limit the spectral distortion introduced by the interruption process.
The spectral distortions resulting from rectangular interruption and reiteration
are illustrated in Fig. 2.2.4, in which the 'comb' filtering effect of the
reiteration process may be seen - some frequencies are reproduced without distortion
while others are completely eliminated.
Fig. 2.2.4. Non-recursive reiteration.

Fig. 2.2.5. Recursive reiteration ($A = 0.5$)

Illustrations of Spectral Distortion introduced during Interruption ($T_t, T_i = 12$ ms) and Reiteration Synthesis -

Frequency spectrum of processed signal.
The non-recursive reiteration process considered so far repeatedly outputs a transmitted signal section at its original amplitude, to fill only the subsequent interrupted section. The weights of the equivalent linear filter are therefore all unity, and span a delay of one interruption period. A more generalised reiteration process may be considered, however, in which each transmitted section is reiterated via non-unity coefficients and over more than one interruption period. Its transfer function \(H(z)\) may be written:

\[
H(z) = \sum_{i=0}^{M'} a_i z^{-i N_t} \tag{2.2.25.}
\]

The weighting sequence corresponding to \(H(z)\) spans \(M'N_t\) sample intervals, or \(M'T_t\) seconds. If this is greater than the interruption period, the synthesiser requires storage for more than one signal section. Also, as non-unity coefficients are involved, the implementation of this reiteration process becomes more complex than for the simple reiteration process described earlier. The effects of this generalised reiteration have been briefly considered previously for the case when the \(a_i\) weights lie within a triangular window spanning two interruption periods. Its transfer function \(H_t(z)\) is then:

\[
H_t(z) = z^{-(M+1)N_t} \left\{ \sum_{i=1}^{M} \left(1-i/(M+1)\right) \left(1 + z^{-i N_t} + z^{-2i N_t}\right) + 1 \right\} \tag{2.2.26.}
\]

where \(M = N_t / N_t\). For comparison, the transfer function \(H_r(z)\) of the simple reiteration process was:

\[
H_r(z) = \sum_{i=0}^{M} z^{-i N_t} \tag{2.2.27.}
\]

The reiteration process corresponding to \(H_t(z)\) may be considered to form its output signal by adding linearly varying proportions of the last two transmitted signal sections. In each interruption period the most recent transmitted section is reiterated successively by weights lying on the rising edge of the triangular window while the previous section is reiterated by weights on the falling edge. To achieve approximate continuity of power in the reproduced signal, the sum of all the weights actually reiterating a transmitted section at any time should be
approximately 1. That is:

\[
\sum_n a \left( N_+ + n N_p \right) \sim 1 \quad \text{for} \quad 0 \leq i \leq M 
\] (2.2.28.)

where \( a \left( N_+ \right) \), \( i = 0, 1, \ldots, M \), includes all the non-zero weights in the first processing period in the delay of the reiteration process, and \( n \) is varied to include all other non-zero weights at intervals of the processing period. This condition is observed by the trianularly weighted reiteration process of equation 2.2.26.

In common with the simple reiteration process of equation 2.2.27, the generalised reiteration process of equation 2.2.25 also exhibits a 'comb' frequency response with peaks at frequency intervals of \( 1/T_1 \) Hz. However, as it has a longer weighting sequence, the width of these peaks for a given transmission ratio is less than those of the simple reiteration process, and might give rise to a more pronounced 'reverberant' effect in the processed speech.

The triangular type of reiteration might have advantages over the simple reiteration process for reproduction of interrupted speech with long processing periods. By the smooth, rather than instantaneous transition obtained between successive transmitted sections the impression of 'discontinuity' evident after simple reiteration might be reduced. However, the longer reiteration weighting sequence might cause detrimental speech 'slurring' effects.

2.2.3. Recursive Reiteration Synthesis

As an alternative to the non-recursive reiteration process described above, it is possible to implement reiteration by a recursive process although this has apparently not been considered previously. In this infinite-memory process the output is formed by adding the interrupted signal input to a fraction \( A \) of the output delayed by the transmission period \( T_1 \). Its block diagram is shown in Fig. 2.2.6. Its transfer function \( H(z) \) may be expressed:

\[
H(z) = \frac{1}{1 - Az^{-N_+}}, \quad 0 \leq A < 1 
\] (2.2.29.)

The feedback coefficient \( A \) typically has a value in the range 0.5 - 0.7. It controls the rate of decay of the reproduced signal during reiteration. The
Interrupted signal

Delay of $N_t$ sample intervals
$= N_t \cdot T_s \cdot s.$

$0 < A < 1$

Output.

Fig. 2.2.6. Block Diagram of Recursive Reiteration.
process exhibits $N_t$ poles in the $z$-plane on a circle of radius $A/N_t$ at angles $t$ where

$$t(i) = 2\pi i N_t, \ i = 0, 1, \ldots, N_t - 1.$$  \hspace{1cm} (2.3.30.)

Response peaks occur at these angular frequencies, and become more pronounced as $A$ tends towards 1. The corresponding frequency response $H(f)$ may be shown to be:

$$H(f) = \frac{1}{1 - 2A\cos(2\pi f T + \ldots + A^2)^i}.$$  \hspace{1cm} (2.3.31.)

This process therefore has a 'comb' frequency response in which the peaks are spaced by a frequency interval of $1/T_t$ Hz. For typical values of $A$, this has some similarity to that of the non-recursive reiteration process. As this recursive process has no zeros it could never eliminate all the distortion introduced by the interruption process, as non-recursive reiteration could under some circumstances. However, to its advantage, it would not completely eliminate any signal frequencies as non-recursive reiteration could. These points are illustrated in Fig. 2.2.5, showing the spectral distortion resulting from interruption and recursive reiteration synthesis.

Comparisons between non-recursive and recursive reiteration processes are of interest, as they exhibit similarities in spite of their fundamental differences. For example, their frequency responses can have similar forms. Intuitively, the non-recursive reiteration process might appear more suited to this application as it fills the interrupted period with the most recent signal available, from the last transmitted section. However, subjective assessments of different processes such as these can provide valuable insight into the basic requirements of the reiteration process.

2.2.4. Reversed Reiteration

A further reiteration technique under current investigation is 'reversed' reiteration in which each transmitted section is reiterated a finite number of times at its original amplitude, but is time-reversed about its centre between successive reiterations. This technique can greatly reduce the 'comb' filtering effect of non-recursive reiteration and the consequent reverberant nature of the reproduced speech.
The process of reversed reiteration may not apparently be represented by a time-invariant model. The form of the spectral distortion introduced is quite complex, as illustrated in Fig. 2.2.7. These spectral effects may be approximately estimated by means of the 'stationary phase' \(^{29}\) method. This assumes that the largest contributions to the energy frequency spectrum of a signal occur at frequencies for which the phase within the Fourier integral is stationary. For the analysis, a cosinusoidal signal \(s(t)\) is considered, where:

\[
s(t) = \cos(2\pi F_t)
\]  

(2.2.32.)

A transmission ratio of \(\frac{1}{2}\) is assumed, although the method may be extended to other cases. During interruption and reversed reiteration of signal \(s(t)\) with a transmission ratio of \(\frac{1}{2}\), alternate \(T\)-long signal sections are transmitted and then reiterated once after time reversal. The Fourier transform of the \(n\)th transmitted signal section, \(S_{tn}(f)\), is:

\[
S_{tn}(f) = \int_{(n-\frac{1}{2})T}^{(n+\frac{1}{2})T} s(t) e^{-j2\pi f t} dt
\]

(2.2.33.)

Substituting for \(s(t)\), and also \(t' = t-n\),

\[
S_{tn}(f) = \int_{-T/2}^{T/2} \cos(2\pi F(t'+nT)) e^{-j2\pi f(t'+nT)} dt'
\]

(2.2.34.)

After evaluation,

\[
S_{tn}(f) = e^{-j2\pi nT(f-F)} \frac{\sin(\pi(f-F)T)}{2\pi(f-F)} + e^{-j2\pi nT(f+F)} \frac{\sin(\pi(f+F)T)}{2\pi(f+F)}
\]

(2.2.34.)

The two terms on the right side of equation 2.2.34 contribute to \(S_{tn}(f)\) largely in positive, and negative frequencies respectively. The contribution from the second term will be ignored here, and the phase effects examined only at positive frequencies. As the signal frequencies of interest are typically much higher than the reciprocal of the transmission period, the effect of the second term over these positive frequencies may be neglected without significant error.
Fig. 2.2.7. Illustration of Spectral Distortion introduced during Interruption (\(T_1, T_i = 8\text{ ms}\)) and Reversed Reiteration Synthesis.
The phase, $P_{+}(n)$, of the positive-frequency term of equation 2.2.34, is:

$$P_{+}(n) = 2\pi n (f + F) T$$  \hspace{1cm} (2.2.35)

The Fourier transform of the $n$th signal section after reversed reiteration into the $(n + 1)$ time section, $S_{rn}(f)$, may be expressed similarly. The time-reversal and shift are effected by the substitution $(2n + 1)T - t'$ for $t$.

$$S_{rn}(f) = \int_{(n+3/2)T}^{(n+1/2)T} \cos \left[ 2\pi F \left( (2n + 1)T - t' \right) \right] e^{-j2\pi ft'} dt' \hspace{1cm} (2.2.36.)$$

After evaluation, this may be shown to be:

$$S_{rn}(f) = e^{-j2\pi fT} \frac{\sin(\pi T(f+F))}{2\pi(f+F)} + e^{-j2\pi nT(f+F)} \frac{\sin(\pi T(f+F))}{2\pi(t+F)} \hspace{1cm} (2.2.37.)$$

Again, only the first term on the right side of equation 2.2.37, contributing to positive frequencies, will be considered. Its phase, $P_{r}(n)$, is:

$$P_{r}(n) = -2\pi n(f + F)T + fT \hspace{1cm} (2.2.38.)$$

The Fourier transform of the complete reverse-reiterated signal, $S'(f)$ is the sum of $S_{tn}(f)$ and $S_{rn}(f)$ over all alternate values of $n$. That is:

$$S'(f) = \sum_{n=-\infty}^{\infty} S_{tn}(f) + S_{rn}(f) \hspace{1cm} (2.2.39.)$$

In steps of 2

The prominent spectral characteristics of the complete reverse-reiterated signal may be estimated by examining the successive values of $S_{tn}(f)$, and $S_{rn}(f)$, independently for phase stationarity. The Fourier transforms $S_{tn}(f)$ of the transmitted signal sections will be considered here first. These might be considered collectively as an interrupted signal with transmission and interruption periods of $T$. The spectral distortion introduced during interruption was described earlier in section 2.2.1. Similar results may be obtained here, as follows.

Phase stationarity is achieved when the phases of the Fourier transforms of
successive transmitted signal sections differ by an integer multiple of $2\pi$. 
That is:

$$P_t(n + 2) - P_t(n) = \pm 2\pi l$$  \hspace{1cm} (2.2.40.)

where $l = 0, 1, 2, \ldots$. Substituting the expression from equation 2.2.35. for $P_t$:

$$- 4\pi (f - F)T = \pm 2\pi l$$

$$\therefore f = F \pm 1/2T, \; i = 0, 1, 2, \ldots$$  \hspace{1cm} (2.2.42a.)

Spectral energy could therefore appear at the frequencies $f$ of equation 2.2.41. which satisfy the phase stationarity condition. These are spaced at intervals of $\frac{1}{2}T$ from the signal frequency $F$. As the processing frequency in this example is $\frac{1}{2}T$, this corresponds with previous results. The power appearing at a given frequency depends on the energy at that frequency in the Fourier transforms of the individual signal sections. The energy per Hz, $E(f)$, of those Fourier transforms over positive frequencies may be approximately obtained from equation 2.2.34. as follows:

$$E(f) = \left( \frac{\sin(\pi (f - F)T)}{2\pi (f - F)} \right)^2$$  \hspace{1cm} (2.2.42b.)

This indicates two important points. Firstly, because of the sinc$^2$ nature of $E(f)$, the spectral distortion is centered about the signal frequency $F$ and is practically limited in extent about $F$. Secondly, it is seen that $E(f) = 0$ when $(f - F) = \pm j/2T$ where $j = 1, 2, 3, \ldots$. These frequencies are the locations of the zeros of the Fourier transform of the rectangular signal section which was assumed in this example. Under this condition, the interruption distortion sidebands indicated by equation 2.2.41. would not appear for even values of $l$. The remaining distortion components appear in the display of Fig. 2.2.7. as the lines with positive gradient.

The spectral contribution from the Fourier transforms $S_{rn}(f)$ of the reverse-reiterated signal section may be investigated similarly. From equation 2.2.38., the phase, $P_r(n)$, of the Fourier transform of the $n$th signal section after reverse reiteration was approximately:
The condition for phase stationarity may again be expressed:

\[ P_r(n + 2) - P_r(n) = \pm 2\pi i, \quad i = 0, 1, 2, \ldots \]  

(2.2.44.)

Substituting for \( P_r(n) \) from equation 2.2.43,

\[ -4\pi(f + F)T = 2\pi i \]

\[ \therefore f = i/2T - F, \quad i = 0, 1, 2, \ldots \]  

(2.2.45.)

As the signal frequency, \( F \), is increased, the frequencies satisfying this stationary phase condition are seen to decrease. This effect gives rise to the lines with negative gradient in Fig. 2.2.7. The energy spectra of the reverse-reiterated signal sections are identical to \( E(f) \) from equation 2.2.42. As a result, most power appears at frequencies \( f \) close to \( F \).

It is possible to examine the two conditions for phase stationarity, given by equation 2.2.41 and 2.2.45, together to determine the frequencies \( f \) for which they are identical. These are the frequencies at which the positive- and negative-gradient lines intersect in the spectral distortion display of Fig. 2.2.7. They are found to be:

\[ f = i/4T, \quad i = 0, \pm 1, \pm 2, \pm 3, \ldots \]  

(2.2.46.)

This condition specifies that a \( T \)-long signal section should contain a whole number of quarter periods of a periodic signal. This condition is necessary, but not sufficient, for perfect reproduction. To achieve this, the reversal performed on the signal section before each reiteration must either leave it unaltered, or transform it into the next naturally occurring section of signal. Stringent phase conditions must therefore be satisfied, which become more restrictive as the transmission ratio is reduced below \( \frac{1}{2} \). For perfect reproduction to be possible, signals must exhibit some even or odd symmetry within a fundamental period. Apparently, general periodic signals with no such symmetry could never be reproduced without distortion after interruption and reversed reiteration. However, if the Fourier transforms \( S_{tn}(f) \) and \( S_{rn}(f) \) of individual transmitted, and reverse-reiterated signal sections are examined, they are seen to consist of the Fourier transform of a signal section of duration \( T \) in conjunction with
with complex exponential coefficients of constant magnitude. The width of the spectra of individual signal sections is therefore controlled by the Fourier transform of the transmission window used for the interruption process. Therefore, although the spectral distortion introduced by reversed reiteration has a very different form to that of other reiteration processes, its overall width is comparable in extent.

2.3. Experimental Investigations

2.3.1. Experimental Details

The interruption and reiteration techniques under investigation were all simulated on a small C.T.L. Modular – I digital computer, in real time. In these experiments, analogue speech signals were input and output via 11-bit analog-to-digital and digital-to-analog converters at a 10 kHz clock rate, and the peak signal amplitudes were set to use at least half the working voltage range of these devices. The speech samples used consisted of phrases and short sentences, extracted from tape recordings of normal conversations made in a non-reverberant environment, and were band-limited to the frequency range 300-3300 Hz.

The tape machines in these experiments included a Sangamo 3564 in FM recording mode, and a Revox A 77. Both had record-replay half-power bandwidths in excess of 0.1 - 20kHz, and signal/noise ratios greater than 50db with RMS total harmonic distortion less than 2%. During experiments the processed signals output from the digital computer were filtered by a sharp cutoff 5kHz low pass filter and were either re-recorded for further use or evaluated in listening tests with AKG K-60 headphones. Further details of the digital computer, and its application in these experiments, appear in Appendix A.

The following sub-sections describe some subjective impressions of speech quality after interruption and reiteration processing. These were obtained informally from three subjects. Also the results of a more detailed preference evaluation test are described.
2.3.2. Interruption

Programs were written to simulate interruption processing with rectangular, triangular, raised-cosine and Hamming transmission windows, in which the duration of the transmission and interruption periods were independently variable. These programs were used in listening tests to gain some impressions of the subjective distortions introduced by interruption, with transmission and interruption periods of the order of 10ms.

After rectangular interruption with transmission ratio of 0.5 and fairly short transmission period of 5 ms, speech sounded very noisy, harsh and 'buzz-like', although remaining easily intelligible. The individual character and pitch of speakers' voices were almost completely masked. The perceived pitch of the 'buzz' distortion effect was found to depend on the processing frequency, and its effect became more severe as this was increased.

As the duration of the transmission period was increased this 'buzz' distortion became less noticeable, and pitch intonations in the original speech became easily discernible. At the same time a 'warbling' or 'gargling' effect began to appear, creating the impression within the listener that individual transmitted sections were becoming separately discernible. These distortions were considered to be least objectionable when transmission periods of about 10ms were used. When increased to about 20ms the 'warbling' distortion became very severe, as it became possible for a listener to easily discriminate individual transmitted signal sections. However, the distortion of voiced pitch and speech character became less significant.

When the transmission ratio was reduced below ½, to ¼ or ⅛ for example, the distortions described above became far more severe. With a transmission ratio of ¼ and transmission period of 5ms the processed speech sounded completely monotonic as if spoken in a harsh, buzzing voice. As the transmission period was increased to 10ms individual interruptions became easily discernible, and when further increased to 20ms the processed speech took on a very irritating 'staccato' quality.
When interruption was performed with non-rectangular transmission windows, it was found that 'buzz' and 'warble' distortions were still introduced although the processed speech sounded less harsh and 'noisy'. It was readily detectable from the character of the processed speech whether a rectangular or non-rectangular window was in use, but not possible to recognise a particular non-rectangular transmission window as all those investigated were found to have very similar effects on speech quality.

2.3.3. Non-Recursive Reiteration

Experiments were conducted to assess the effects of non-recursive reiteration synthesis on the quality of interrupted speech. Programs were written to simulate these processes, in which the transmission ratio and duration of the transmission period were variable. The intelligibility of interrupted speech after non-recursive reiteration has been measured by Sharf (15). He found that speech intelligibility was generally high after this processing, but not significantly different from that of interrupted speech itself, and concluded that these reiteration techniques were of little value. However, it has been found currently that reiteration techniques can significantly improve the quality of interrupted speech.

During initial tests using 10ms transmission period and transmission ratio of ½, non-recursive reiteration was found to largely eliminate the 'buzz' and 'warble' distortions of interrupted speech, although the processed speech still sounded harsh and without individual character. It sounded monotonic, and somewhat as if it had been spoken in a very reverberant environment, probably as a consequence of the periodic 'comb' frequency response of the reiteration process. The apparent pitch, and severity of the monotonic effect depended upon the transmission period used. If reduced to 5ms, pitch intonation and character in the original speech were almost completely masked. Best quality was probably obtained with a transmission period of about 10ms - if increased to 20ms the processed speech quality deteriorated as individual reiterations became separately detectable, imparting an impression of 'discontinuity' to the listener.
If the transmission ratio were reduced below $\frac{1}{2}$, for example to $\frac{1}{3}$ or $\frac{1}{4}$, causing each signal section to be reiterated two or three times respectively, these distortions increased greatly. In particular the 'monotonic' effect became far more severe. With transmission periods below about 10 ms, the processed speech sounded rather as if it had been sung, at constant pitch.

The effects were also investigated of weighting transmitted signal sections with a Hamming window. This had little effect if the transmission period was short, but when longer transmission periods of about 20 ms were used this transmission window markedly modified the character of the processed speech, lessening the impression of 'discontinuity' and introducing a slight 'warble'.

Brief investigations were also conducted into the effects of more complex non-recursive reiteration processes with impulse responses spanning more than one interruption period, and with non-unity reiteration coefficients. The reiteration coefficients of the particular reiteration process used fell within a triangular window spanning two interruption periods, as described by equation 2.2.17.

With interrupted speech of transmission ratio 0.5, the subjective effects of this triangular reiteration process were found to be indistinguishable from those of the simple non-recursive reiteration process described previously over the range of transmission period of 5 to 20 ms. A similar 'monotonic' effect resulted when short transmission periods were used, while an impression of discontinuity was still obtained from the processed speech with longer transmission periods. Differences in subjective effect became apparent, however, if lower transmission ratios were employed, especially with a long transmission period. With a transmission ratio of $\frac{1}{3}$ and transmission period of 20 ms the discontinuous 'staccato' effect in interrupted speech synthesised by the simple reiteration process was noticeably reduced by this triangular reiteration process although the processed speech took on a markedly 'slurred' character. It was concluded that the triangular reiteration process has few advantages over simple reiteration to justify its greater complexity, the disturbing nature of the 'slurring' effects being comparable to that of the 'discontinuous' effects of simple reiteration when a low
transmission ratio and long transmission period are used.

2.3.4. Recursive Reiteration

Experiments were conducted into the effects of recursive reiteration on interrupted speech signals. As well as the transmission period and transmission ratio, the feedback coefficient $A$ (equation 2.2.19.) was also variable in the range 0 to 1. This controlled the signal decay between successive reiterations. Recursive reiteration was investigated in conjunction with rectangular interruption. In general, this form of processing was found to improve the quality of interrupted speech, although the improvement was less than that achieved by non-recursive reiteration. If the feedback coefficient were less than 0.5 the 'buzz' and 'warble' distortions of the interrupted speech remained, and the reiteration process had little apparent effect on quality. Best results were obtained with values of feedback coefficient between 0.5 and 0.7 - interruption distortions were noticeably reduced, although the processed speech began to assume a monotonic 'buzz' or 'whistle' - like character. This effect was more severe than the monotonic effect obtained from non-recursive reiteration, probably because the peaks in the frequency response of the recursive reiteration process (Fig. 2.2.6.) are narrower, under comparable processing conditions. The masking effect over the pitch and character of the original was also greater. If the feedback coefficient was made larger than 0.7 these effects became severe although some 'warble' interruption distortion still remained. The 'buzz' and 'whistle' distortion produced with higher values of feedback coefficient increased greatly in severity as the transmission ratio was reduced below $\frac{1}{2}$. As the transmission period was varied with transmission ratio of $\frac{1}{2}$, the best processed speech quality was probably obtained with a value of 10ms.

2.3.5. Reversed Reiteration

Finally, the effects of reversed reiteration processing on interrupted speech were investigated. This was found, in general, to produce comparable improvements to non-recursive reiteration in the quality of interrupted speech. The 'reverberant' effect introduced by non-recursive reiteration was almost absent after this form of
processing, although the processed speech sounded slightly more harsh. Reversed reiteration was investigated in conjunction with rectangular interruption.

Subjectively 'best' quality was probably obtained when a transmission period of 10ms was used. With a transmission ratio of \( \frac{1}{4} \) the previous remarks were found to apply. The pitch and character of the original speech were retained to some extent, and suffered less distortion than during non-recursive reiteration. If a shorter transmission period, of about 5ms, were used the harshness of the processed speech became worse, together with the distortion of voiced pitch. Pitch inflections were audible; but at times became inverted. When the pitch of the original speech decreased, for example, the pitch of the reproduced speech might sometimes increase. This effect may be considered against the spectral distortion display for reversed reiteration (fig. 2.2.7.) The downward-sloping lines indicate that an increase in signal frequency can cause a decrease in frequency of some spectral components. The movement of spectral distortion components against the corresponding change in signal frequency probably gives rise to the phenomenon of inverted pitch inflection. The effects of reversed reiteration with longer transmission periods, of about 20ms, were very similar to those of non-recursive reiteration under comparable conditions - the processed speech gave a similar impression of 'discontinuity'.

When the transmission ratio was reduced to \( \frac{1}{3} \), some 'reverberation' effects became evident, although they were less noticeable than from non-recursive reiteration. As with the previously described reiteration techniques, however, the processed speech quality rapidly deteriorated as the transmission ratio was reduced below \( \frac{1}{4} \).

From these listening tests it was informally judged that the reiteration techniques described have achieved a useful improvement in the quality of interrupted speech. These improvements are investigated in greater detail in the following subsection.
2.3.6. Preference Evaluation Tests

To obtain a more reliable indication of the effects of non-rectangular transmission windows, and reiteration synthesis techniques on interrupted speech quality, some speech samples were processed and tape-recorded for a preference evaluation test. These were prepared from the speech samples, programs and equipment described earlier. Because of the time-consuming nature of such tests, only four types of processing were investigated. These comprised interruption with and without Hamming transmission window, and with and without non-recursive reiteration synthesis. A transmission ratio of \( \frac{1}{4} \) was employed throughout.

Speech samples were prepared from each of these forms of processing with ten values of transmission period spaced approximately geometrically between 3.3ms and 25ms, corresponding to a range of processing frequency of 150Hz - 20Hz.

The speech samples were recorded in pairs – firstly unprocessed, and then after the required processing. A pause of approximately five seconds was included between each pair of samples. The transmission period, and type of processing were distributed randomly throughout the test which, including only one presentation of each combination of processing parameters, ran for approximately 20 minutes.

The test procedure was based on the 'rating method' \( (30,31) \). Before the test, subjects were told that they would hear pairs of short sentences and were asked to rate the second of the pair "on the basis of overall quality on an equal interval scale, with 10 defined as the quality of the first sentence and 0 defined as just unintelligible". An alternative testing method would have been the commonly used 'direct comparison' \( (30,31) \) procedure in which subjects are asked to compare a given sample of processed speech with a reference sample having known distortions, and to give a binary 'better/worse' response. The subjects would then compare a sample with a single standard, rather than rate it in an interval between two standards, and the results of the comparison test could be related directly to the reference distortion processes employed. The results from a rating test could not be; the rating of a particular processed speech sample would depend far more on
the judgement of the subject, as well as the upper and lower standards provided. However the rating method has the great advantage of allowing a subject a choice of a large number of responses (in this case 11) to each test sample, rather than 2 as in a comparative test. The subject can therefore provide more information per response, enabling useful results to be obtained far more rapidly from fewer responses.

The test was performed in a small, quiet, furnished room in which the test samples were reproduced via a high quality loudspeaker. It was conducted with four adult English, technically naive subjects who had not previously heard speech processed by these techniques. The test was conducted without pause, but none of the subjects complained of tiredness or fatigue afterwards. The average ratings of the different forms of processing are displayed in Fig. 2.3.1., against transmission period. There are several sources of error in the results, including:

(a) Subjects' variations in judgement.
(b) Bias resulting from test environment.
(c) Errors resulting from the limited number of test samples.

Subjective judgement errors within type (a) would include a random component, and a fixed bias resulting from differing interpretations of the test instructions and differences in mood and personality between the subjects. To estimate the consistency between the ratings of different subjects, the standard deviations were computed of the individual subjects' ratings at each test sample. Their average value over the whole test was 1.03. Some bias was evident between subjects' ratings, as the separate averages of individual subjects' ratings over the whole test were found to differ between 3.8 and 4.9.

However when these averages were separately eliminated from individual subjects' ratings and their standard deviations again computed at each test condition, their average value for the whole test decreased from 1.03 only to 0.95. Differences of judgement between subjects were therefore taken to be largely random, rather than caused by bias. Also, as the averaged ratings of the test samples ranged from below 2 to above 6, i.e. more than four times the mean inter-
Graph 1: Interruption, rectangular transmission function.  
Graph 2: Interruption, Hamming transmission function.  
Graph 3: Interruption and reiteration, rectangular transmission function.  
Graph 4: Interruption and reiteration, Hamming transmission function.  

Fig. 2.3.1. Results of Preference Evaluation Test for Interruption and Non-recursive Reiteration.
subject standard deviation, the subjects' ratings were considered usefully consistent.

The ratings in these tests would have been biased by associated environmental conditions, introducing errors of type (b). The facts that the test samples were reproduced by loudspeaker, and that the subjects were technically naive, for example, probably had significant effects on the ratings. These effects are not important here, however, as the test results are required for comparison, rather than absolute assessment, of these forms of processing.

The quality of individual processed samples might have been affected by the suitability of the particular speech sample used, introducing random errors of type (c). Ratings might also have been biased by contextual effects - for example, by the quality of previous test samples. As each combination of processing parameters only appeared once in the test, it was not possible to estimate the magnitude of these effects or to reduce them in the ratings of individual test conditions. However the errors in successive ratings would be uncorrelated while the ratings themselves would be expected to change smoothly with processing parameters. The effects of these errors could therefore be reduced by smoothing the available data. In this case, it was necessary to smooth the averaged ratings taken at equal logarithmic intervals of transmission period, by a 3-point filter having weighting sequence ($\frac{1}{4}, \frac{1}{2}, \frac{1}{4}$).

The test results will now be considered. As shown in Fig. 2.3.1., the ratings of all the forms of processing investigated exhibited maxima for transmission periods of about 10ms, the peaks being slightly more pronounced in the ratings of interruption processes. The figure of 10ms probably represents a compromise between the distortion of speech pitch and character at shorter transmission periods, and the irritating 'discontinuous' nature of speech after interruption processing at longer transmission periods.

The effects upon processed speech quality of Hamming weighted interruption is difficult to discern from these results. It produced a slight improvement in ratings, except for interrupted speech at shorter transmission periods.
However, the greatest improvement in ratings was approximately 1. This is comparable with the average standard deviation between individual subjects' ratings, and therefore cannot be considered significant. Reiteration synthesis produced a more marked improvement, however, increasing the ratings of interrupted speech by about 50% for a transmission of 10ms, and by larger percentages towards the upper and lower extremes of transmission period investigated. The ratings of interrupted speech after reiteration synthesis were noticeably less dependent on transmission period, their graphs (Fig. 2.3.1.) being less peaked than those for interruption processing alone.

These results show that reiteration synthesis can achieve a significant improvement in the overall quality of interrupted speech. Also the improvement is greater than that obtained from use of a Hamming transmission window. These points will be discussed in the following section.

2.4. Discussion

In the preceding section it is noted that the reiteration synthesis techniques investigated can achieve significant improvements in the perceived quality of interrupted speech, the greatest improvements being obtained from non-recursive and reversed reiteration techniques. It is also noted that the order of improvement obtained is greater than that achieved by use of non-rectangular transmission windows, chosen for more concentrated spectral distortion from the interruption process. As pointed out earlier, time-invariant reiteration processes modify the spectral distortion resulting from interruption but cannot in general reduce it. These processes can, however, reduce the discontinuities in signal power introduced by interruption. The best performance in this respect would be given by non-recursive and reversed reiteration techniques, as the signal amplitude during recursive reiteration would decrease exponentially in successive reiterations and some of the signal discontinuities introduced by interruption would remain. As noted above, the non-recursive and reversed reiteration techniques are also found to produce the greatest improvements in the quality of
interrupted speech. The success of these reiteration techniques suggests that the subjective distortion associated with interruption processing result mainly from aural discrimination of interruptions in time, rather than from the associated spectral distortion.

Miller and Licklider\(^{(6)}\) provide some supporting evidence. They reported that the harshness, and 'buzz' or 'warble' distortions of interrupted speech could be significantly reduced by the addition of white noise during interrupted periods. This finding has been confirmed during current investigations. The addition of noise in this way would not, of course, reduce the spectral distortion introduced by interruption but would produce a signal having less variation of power with time. The fact that a subjective improvement is achieved implies further that the ear responds adversely to the localisations in time of signal power after interruption.

It is of interest to consider the duration of the shortest interruptions which may be perceived by temporal discrimination only. Mowbray, Gebhard and Byham\(^{(32)}\) have found that regular interruptions of duration as small as 2ms may be perceived, even when the signal spectrum is unaltered by the interruption. They reported that the interruption of white noise at processing frequencies up to 250Hz (corresponding to transmission and interruption period of 2ms) produced clearly audible effects, and that it was possible for listeners to match the pitch of a variable frequency sinusoidal signal to the apparent pitch perceived from the interrupted noise. These findings indicate not only that the ear is sensitive to short interruptions in a signal, but also that a sensation of pitch can result from periodic power variations rather than from periodicity in the frequency spectrum.

More detailed investigations of temporal discrimination in aural perception were reported by Matthes and Miller\(^{(33)}\) from experiments on the phase sensitivity of the ear at audio frequencies. Three sinusoidal signals equispaced in frequency, eg 900, 1000, 1100Hz, were added to produce beat effects. Their amplitudes and phases were adjusted so that the composite signal exhibited 100% amplitude modulation. If the envelope frequency was less than about 400Hz they reported that this AM-type composite signal was perceived as 'raucous'. However if the phase of the centre frequency was shifted by 90°, without altering the
signal amplitudes, to produce an approximate FM-type signal having only about 25% envelope variation, the composite signal was found to sound 'smoother'. Two of their findings are of particular interest here. Firstly, they reported that the 'raucous' nature of the AM-type signal was most pronounced for beat frequencies in the range 24-75Hz which unfortunately coincides with the range of processing frequencies employed in this type of interruption processing. The sound of interrupted speech might be described similarly as 'raucous'. Secondly it was found that the perceived pitch of the AM-type signal was related to its envelope frequency rather than to a fundamental frequency, if any. These results again indicate that pitch may be perceived from temporal rather than spectral characteristics of a signal. These effects have been considered more recently by Plomp (34) who also concluded that the pitch of complex signals was related to the frequency of occurrence of auditory stimuli, rather than any harmonic relationship between the spectral components of the signal.

Comparable effects were encountered currently in interrupted speech which, with short transmission periods, assumed a 'buzz'-like quality whose pitch was closely related to the processing frequency. This connection was established by a simple experiment in which either a speech signal or a direct voltage could be interrupted. When the direct voltage was processed in this way a buzz sound was produced whose pitch corresponded to the processing frequency. When a speech signal was then applied the 'buzz' sound introduced was apparently identical in pitch, implying that this effect resulted directly from temporal perception of interruptions.

These results indicate that processes of temporal discrimination play an important part in the aural perception of interruption distortion. However the spectral effects of interruption processing are also significant. The monotonic character of interrupted speech after non-recursive reiteration may not be explained in terms of temporal discontinuities of signal power as these are largely eliminated by the reiteration process. It most probably results from the
periodic 'comb' frequency response of the reiteration process, as power in the reproduced signal is concentrated at the frequencies of the regularly spaced response peaks. To estimate briefly the degree to which a pitch could be perceived from the periodicity of the power spectrum of the reiterated signal a simple non-recursive reiteration process was simulated and fed with white noise. Its transfer function $H(z)$ was:

$$H(z) = 1 + z^{-N_t} \quad (2.4.1.)$$

The reiteration process corresponding to this transfer function would produce one reiteration of a transmitted signal section of duration $N_t T_s$ where $T_s$ is the sampling period. As explained in section 2.2.2., a non-recursive reiteration process with only one delay such as this exhibits the broadest frequency response peaks. Its frequency response $H(f)$ is:

$$H(f) = 2 e^{-j\pi f N_t T_s} \cos(\pi f N_t T_s) \quad (2.4.2.)$$

$$\left|H(f)\right| = 2 \left|\cos(f N_t T_s)\right| \quad (2.4.3.)$$

$\left|H(f)\right|$ therefore has the form of a rectified cosine function with peaks spaced at frequency intervals of $1/N_t T_s$ Hz. The filtering effect of this process on continuous white noise was found to be quite audible, and gave an impression of a fixed musical pitch. This was obtained for values of filter delay up to about 20ms, corresponding to a response peak interval of only 50 Hz. When shorter delays were used, however, the perceived pitch was found to vary in a complex way as the filter delay, and thus the response peak frequency interval, was altered. The perception of pitch from the periodicity of a power spectrum is apparently not well understood\(^{(34)}\), and further investigations could be usefully performed to explain these phenomena. Similar impressions of a fixed, monotonic pitch were obtained if continuous speech signals were filtered in this way, although the effects of long filter delays were not clear as any pitch effects were masked by the temporal echo effects introduced.

These results indicate that the temporal effects of interruption with transmission periods as short as 2ms may be perceived, while the spectral effects of
reiteration with transmission periods as long as 20ms may also be resolved aurally. In the range of transmission period employed in this type of interruption processing, of 5ms-20ms approximately, the ear apparently perceives interruption distortion to a significant degree by both its temporal and spectral effects. In the reproduction of interrupted speech, therefore, it is necessary to limit both discontinuities in signal power, and spectral distortion. It is considered unlikely that any worthwhile improvements in processed speech quality may be achieved from the simple synthesis techniques described so far. To gain further improvements in performance some more complex adaptive reiteration techniques are proposed in the next chapter.

An important aspect of interruption processing techniques is their practical applicability for speech transmission. Any reliable assessment of practical usefulness is very difficult - it involves consideration not only of the subjective distortions introduced, but also the ability of a potential user to accept them. However during the listening tests described in section 2.3, the subjects were able to offer informal comments on the practical importance of these techniques. The forms of processing judged to reproduce interrupted speech with best quality were non-recursive reiteration with or without Hamming transmission weighting, and reversed reiteration processes. The optimum processing conditions were judged to be a transmission ratio of $\frac{1}{4}$ and a transmission period of 10-15ms. The speech reproduced by these processes was considered to be of satisfactory quality for telephone conversation, although slightly fatiguing over long periods. All the subjects commented on its reverberant, or 'mechanical' nature, and felt this to be the most immediately noticeable distortion effect. Speaker identification was not found easy, although better if the longer transmission period of 15ms were used.

It is felt that these simple interruption techniques could satisfy some speech transmission applications, and could usefully double the transmission capacity of a digital speech channel. They are particularly attractive in view of their simplicity, as non-recursive reiteration (without Hamming weighting) or
reversed reiteration techniques merely require signal delay and switching operations. They may therefore be implemented conveniently by current digital techniques.

2.5. Summary

The theoretical effects of interruption and reiteration synthesis techniques have been considered. The subjective effects of these forms of processing on speech signals have also been estimated, from which it was concluded that reiteration synthesis techniques can achieve a useful improvement in the quality of interrupted speech.

From considerations of the subjective effects of different forms of interruption processing it was also concluded that the subjective distortions are perceived not only through the spectral effects of interruption processing but also through the temporal discontinuities of signal power introduced. To achieve further improvements in the quality of the processed speech it is apparently necessary to limit both temporal and spectral distortion effects. In view of the stringency of these requirements it is considered unlikely that further improvements in performance may be achieved from the time-invariant interruption and reiteration techniques considered here, and some improved adaptive reiteration synthesis techniques are considered in the following Chapter.
Chapter 3. Reiteration Synthesis using Adaptive Waveform Prediction

It was concluded in Chapter 2 that the ear perceives the effects of interruption processing from the accompanying discontinuities in signal power, as well as from the spectral distortion introduced. The magnitude of the temporal effects may be reduced by simple reiteration synthesis techniques, but the spectral distortion still remains. During non-recursive reiteration of interrupted voiced speech signals the transmitted speech sections do not generally contain a whole number of fundamental 'pitch' periods and the reproduced speech does not then retain its original periodicity. The distortions of pitch and the characterless nature of speech after interruption and non-recursive reiteration were noted prominently in Chapter 2.3. However it was also noted (Chapter 2.2.) that a periodic signal may be interrupted and synthesised by non-recursive reiteration without distortion if the transmitted signal sections contained a whole number of signal periods. Under this condition the reiterated sections join exactly with preceding and following transmitted sections and phase continuity is maintained throughout. A large proportion of speech is 'voiced', being generated by regular excitation of the vocal tract by short 'puffs' of air through the glottis. As this type of speech is approximately periodic some previous workers (10,11,14) have suggested that the distortions introduced during its interruption and reiteration might also be reduced if the transmitted sections always contained a whole number of periods, typically only one. This has been described as 'pitch-synchronous' processing. Because of the attendant practical problems, however, only one investigation (11) has used real speech signals. Instead of employing variable length transmission periods, with consequential problems of multiplexing and transmission, this process may be rearranged so that the transmission period is fixed, and only the duration of the reiterated signal section is varied. A block diagram of this synthesis method, employing a variable delay line, is shown in Fig. 3.1.1. Its operation, with the transmitted sections of an interrupted periodic signal, is now described.
Switch selects signal (a) during transmission periods, and (b) during interruption periods.

Delay duration $mT_s$ set equal to whole number of signal periods.

Fig. 3.1.1. Block Diagram of Reiteration Synthesis Process incorporating a Variable Delay Line.
Each transmitted signal section is output as it is received, and is also fed into the delay line. The delay is adjusted to span a whole number, $n$, of signal periods and to be less than or equal to the transmission period. At the end of the transmission period, therefore, the delay line stores the most recent $n$ signal periods. During the following interrupted period the delay line supplies the output signal, continuous in phase with the last transmitted section, which is still reapplied to its input. The stored signal section of $n$ periods is repeatedly reiterated as it circulates through the delay line, and fills the interrupted period. As the original signal was periodic the reiterated signal will be identical to that originally interrupted at the transmitter, and phase continuity will also be achieved with the following transmitted signal section. The interrupted periodic signal is therefore reproduced without distortion. The reiteration cycle is stopped when the next transmitted signal section arrives, which is also processed as described above. It should be noted that the total number of reiterations performed in an interrupted period depends on the signal period and is not in general a whole number.

This modified reiteration technique is attractive for pitch-synchronous processing as information is not required about the precise epochs of glottal pulses, but only about their period. It may also be usefully considered from a more general viewpoint. In this application the delay line may be considered to perform waveform prediction. Its transfer function $H_p(z)$ is:

$$H_p(z) = z^{-m}$$  \hspace{1cm} (3.1.1.)

where the delay $m T_s$ spans a whole number of signal periods. When provided with a section of periodic signal and with information about its period, the predictor generates exactly the next sample value of the signal. If this output value is reapplied to the predictor input it may continue to generate successive sample values of the periodic waveform. To operate in this way the maximum predictor delay, and the transmission period must be greater than or equal to the signal period.

The only signal characteristic assumed by the predictor $H_p(z)$ is its
periodicity. Voiced speech signals are, however, only approximately periodic. As well as their fundamental frequency their other characteristics such as their short-term amplitude spectrum all change with time. Changes in the fundamental frequency of speech with time would impose a limit on the accuracy of waveform prediction obtainable with pitch-synchronous processing, in which the prediction could only be adjusted conveniently during receipt of a transmitted speech section, and before starting reiteration. However, as discussed later in Chapter 4, pitch-synchronous processing is found to achieve a great improvement in the quality of speech after interruption and reiteration. Furthermore it is possible that this improvement would be still further increased if, as well as approximately maintaining speech periodicity between transmission periods, other speech characteristics could also be maintained to achieve more accurate waveform prediction.

The possible prediction techniques for this application are limited by the fact that the predictor should be able to operate, and compute its parameters, as far as possible from the signals in successive transmitted speech sections. For efficiency of communication, the computation of parameters at the transmitter to be subsequently sent through the transmission channel must be limited. It is for this reason that predictors designed from very general signal descriptions, such as Wiener\(^{(35)}\) least-squares error predictors, are mostly unsuitable as the waveforms of the interrupted signal sections are needed for their computation. This therefore has to be performed at the transmitter. As the weighting sequence of the non-recursive predictor would be approximately equal in length to the number of signal samples in each transmitted section the bit-rate reduction obtainable would be roughly halved. Although general least-square error predictors have not been considered for this application, they can provide interesting performance comparisons with the predictors investigated here. These points are discussed further in Chapter 6.

The configurations of the predictors employed currently have been simplified by assuming some speech characteristics in their design. They are derived from a model of the vocal mechanism in which the more subjectively significant aspects
of speech are described. As well as simplicity of implementation, this constraint brings a further advantage. When fed with a speech signal, the predictor will generate a signal having speechlike characteristics, although these characteristics may be in error. By comparison, a general predictor with no inherent constraints might generate signals having distinctly non-speechlike characteristics which could be more subjectively disconcerting. The speech model used in the derivation of the predictor will now be described.

The vocal mechanism is represented here in the frequency domain by the model shown in Fig. 3.1.2. As the accurate reproduction of voiced speech is apparently of greatest subjective importance, the model is oriented towards this. The vocal excitation is represented by source $S(f)$ multiplied by $H_g(f)$. $S(f)$, whose spectral envelope is approximately flat, describes the fine spectral detail of the source while the broad spectral characteristics are included in $H_g(f)$. This permits a convenient description of voiced excitation, in which case $S(f)$ is the Fourier transform of an approximately periodic unit-area impulse source, and $H_g(f)$ accounts for the practically encountered non-impulsive nature of the glottal excitation. The excitation amplitude is separately controlled by coefficient $A$. $H_t(f)$ represents the acoustic transfer function of the vocal tract. The effects of the broad, slowly varying spectra $H_g(f)$ and $H_t(f)$, controlling the short-term speech spectrum, may be conveniently combined in $H_s(f)$ where:

$$H_s(f) = H_g(f) \times H_t(f) \quad (3.1.2.)$$

All the elements of this model are time-varying. The short-term spectrum of the speech signal about time $t$, $S_0(f,t)$, is:

$$S_0(f,t) = A(t) \times S(f,t) \times H_s(f,t) \quad (3.1.3.)$$

The waveform prediction must therefore be able to continue changes in speech amplitude and short-term spectrum, governed by $A(t)$ and $H_s(f,t)$ respectively, during interrupted periods. This necessitates that either the characteristics of the interrupted speech section must be predictable from those of the transmitted sections, or that the errors involved in prediction are not subjectively significant.
Approximately periodic impulse source.

Amplitude.

Accounts for broad spectral characteristics of vocal source - i.e. its practical non-impulsive nature.

Vocal tract transfer function.

- Fourier transform of speech signal.

\[ S_0(f) = S(f) \cdot A \cdot H_g(f) \cdot H_t(f) \]

Fig. 3.1.2. Frequency-domain Model of Vocal Mechanism.
As prediction errors increase as the interruption period is increased, they effectively impose an upper limit on it. An interruption period of 40ms is apparently feasible with the waveform prediction techniques investigated here, enabling a transmission ratio of about \(1/2\) to be achieved.

To predict the speech amplitude \(A(t)\) approximately during interrupted periods a gain factor \(B\) may be incorporated in the previously described predictor \(H_p(z)\). Its new transfer function \(H_a(z)\) becomes:

\[
H_a(z) = B z^{-m} \tag{3.1.4}
\]

During reiteration, the reiterated signal is scaled by coefficient \(B\) each time it circulates through the predictor. The coefficient \(B\) is set equal to the amplitude growth of the speech over a time of \(mT\), i.e., the delay of the predictor. A first-order predictor of this kind could only reproduce a speech signal with correct amplitude if it were varying exponentially with time. This may be illustrated by considering the variation in amplitude, \(A(t)\), of a transmitted speech section during reiteration. The signal amplitude during the \(n\)th reiteration, \(a_r(t + nT)\), becomes:

\[
a_r(t + nT) = A(t) B^n \tag{3.1.5}
\]

where \(0 \leq t < T\), the predictor delay in this example. The function \(a_r(t)\) changes geometrically on each reiteration. If it is to be reproduced exactly the signal amplitude \(A(t)\) must similarly exhibit changes by a factor \(B\) over time \(T\). That is:

\[
A(t) = B A(t-T) \tag{3.1.6}
\]

For exact reproduction, therefore, the function \(A(t)\) requires the general form:

\[
A(t) = p(t) e^{kt} \tag{3.1.7}
\]

where \(p(t)\) is a function periodic in \(T\), and \(k = (\log B)/T\). In the simplest case, for example, \(p(t) = C\), a constant. Then, for exact reproduction,

\[
A(t) = Ce^{kt} \tag{3.1.8}
\]

This type of prediction will therefore only perfectly track exponentially varying amplitudes. The amplitude of a speech signal would only display an approximate exponential characteristic over short periods. To obtain more accurate
prediction a second-order predictor $H_{a2}(z)$ might be considered:

$$H_{a2}(z) = B_1 z^{-m} + B_2 z^{-2m} \quad (3.1.9)$$

Although it could predict more complex variations in amplitude, this approach has not been investigated as the predictor weighting sequence has a span of $2m$ sample intervals, which is twice as long as that of the first order predictor described by equation 3.1.4. As this weighting sequence must also span at least two pitch periods for correct operation, the transmission period would have to be twice as long. It is unlikely that this predictor would enable the interruption period to be doubled, and so the bit-rate compression obtainable would most probably be reduced. A further disadvantage of this type of predictor is that it adds two signal samples separated nominally by an integer number of signal periods. If they were not, through non-stationarity of the speech fundamental frequency or from errors in its estimation, serious waveform distortions could result. An alternative approach to reduce the errors of first-order prediction is to compute the value of $B$ to achieve smooth amplitude transition between successive transmitted sections, rather than to continue the amplitude trend of the current transmitted section. These points are discussed further in Chapter 5.

Changes in the short-term speech spectrum, represented by $H_s(f)$ in the model of equation 3.1.3., result largely from modifications in the mechanical configuration of the vocal tract during speech. They may be predicted approximately by a simple non-recursive zero-phase filter included in cascade with the predictor described so far. In this way, small changes would be made to the short-term spectrum of the speech section stored in the predictor on each reiteration. Again, such a filter could only perfectly predict exponential changes in spectral magnitudes. A zero-phase filter is used so that the speech amplitude spectrum may be modified during reiteration without significantly altering its phase and waveform characteristics such as its form factor. The transfer function $H_{cz}(z)$ of this zero-phase filter is:

$$H_{cz}(z) = a(0) + \sum_{i=1}^{N-1} a(i) (z^i + z^{-i}) \quad (3.1.10)$$
The transfer function $H_c(z)$ of the complete waveform predictor becomes:

$$H_c(z) = B \cdot z^{-m} \left( a(0) + \sum_{i=1}^{N-1} a(i) (z^i + z^{-i}) \right)$$

(3.1.11.)

The weighting sequence of $H_c(z)$ spans $2N-1$ samples. It should be noted that this filter is required to predict changes in speech amplitude spectra between transmitted sections, rather than to model the speech characteristics at that time. If the amplitude spectrum was stationary at a particular time, no filtering action would be required during reiteration and the $a(i)$ coefficients of equation 3.1.11. would be zero except $a(0)$. The filter frequency response $H_c(f)$ is set equal to the fractional change in speech amplitude spectrum occurring over the delay $m T_s$ of the predictor. As the vocal tract is an infinite-memory system the frequency response required by the waveform predictor is not in general time-limited. Its frequency response, $H_c(f)$, is however time-limited to $2NT_s$. The value chosen for $N$ therefore depends on the accuracy with which speech spectra are to be predicted during interrupted periods. Values of approximately 10 have been used in current investigations. These considerations are discussed further in Chapter 6.

The processing performed on the transmitted speech sections during reiteration by the waveform predictor of equation 3.1.11. includes delay, multiplication and simple zero-phase filtering. As the reproduced signal is obtained from sections of the original speech signal by this simple processing and without intermediate parameterisation, subtle speech characteristics, particularly relating to its excitation, may be well reproduced. In particular the 'peakedness' of voiced speech signals is retained reliably. As concluded in Chapter 2, the ear is sensitive to temporal signal characteristics over durations of normally encountered pitch periods, and several workers\(^{(36, 37, 38)}\) have suggested that correct reproduction of phase within pitch periods is important if the character of speech is to be reproduced. A further advantage of this reproduction technique is that, as the transmitted speech sections would normally contain several pitch periods, minor random variations or quasiperiodicity\(^{(39)}\) in the vocal excitation would be reproduced to some extent from the original speech. Many speech coding schemes of
higher transmission efficiency assume a model of the vocal mechanism and operate by encoding speech signals into the parameters of this model. Any speech characteristic not represented in the model is therefore lost. In the channel vocoder, for example, the vocal excitation is normally coded into only two parameters specifying its period and the degree of periodicity. Although speech of good quality may be reproduced, it may sound different to the original speech as a result of the suppression of minor speech characteristics not represented in the model.

An important property of these waveform prediction techniques is their ability to reiterate speech-like waveforms when small errors exist in the estimated speech characteristics, particularly in the pitch period. This is very necessary as speech characteristics are non-stationary and may therefore only be estimated approximately. If the pitch period were wrongly estimated, for example, some periodicity would still be retained during reiteration as the predictor delay spans all pitch periods in a transmitted speech section. The continuous variation of the pitch period during speech effectively prevents the use of some previously proposed synthesis techniques. For example, David and McDonald(14) have described a synthesis procedure in which transmitted speech sections were delayed, to allow pitch-synchronous reiteration relatively forwards and backwards in time over about two adjacent interrupted periods. The output signal during interrupted periods was a weighted sum of all the available reiterated signals. This type of generalised reiteration process was considered in Chapter 2.2. Its effect was to modify the characteristics of the synthesised signal fairly smoothly between transmitted sections, without the need for prediction of speech amplitude or short-term spectrum. This scheme was originally tested with vocoder-synthesised, constant-pitch speech which assured that the reiterated signals, added to form the output signal, were always in phase. This condition is not guaranteed with real speech, however, whose pitch period is constantly changing. As a result, the characteristics of the reproduced waveform
could become seriously distorted.

A further advantage of interruption techniques for speech transmission arises from the fact that the interrupted speech signal is itself intelligible, although of very poor quality. A transmission channel employing interruption techniques could be used for emergency communications if all the reiteration synthesis equipment failed. Interruption techniques therefore become more attractive than most other methods for efficient speech communication where reliability is of major importance.

Applications of waveform prediction in speech transmission systems have been considered elsewhere. Atal and Hanauer\textsuperscript{(40)}, and Weinstein and Oppenheim\textsuperscript{(41)} have described systems in which the coefficients of a linear predictor for the speech waveform are regularly sent to the receiver, with pitch information. The predictor generates the speech output recursively by repeatedly predicting one sample value ahead and feeding this back to its input together with the specified vocal excitation function. Bit-rate compressions of about 20:1 were achieved. Schroeder and Atal\textsuperscript{(42)} earlier investigated the use of a more complex predictor based upon a more comprehensive model of the vocal mechanism, which accounted for periodicity in the speech waveform, as well as non-uniformity in its short-term spectrum. This was employed in a predictive coding scheme in which the error between each speech sample and its predicted value was quantised and transmitted. As well as the error signal, the predictor coefficients also had to be transmitted regularly to the receiver. As a result, lower bit-rate compressions of about 5:1 were obtained. The predictors employed in all these investigations were based on models of the vocal mechanism, and were designed within the constraints of these models for least mean square prediction error.

In the following chapters the stages are described in the development of the complete waveform predictor of equation 3.1.11. These deal with the prediction of the phase of the vocal excitation, used for pitch-synchronous reiteration, together with prediction of speech amplitude and short-term spectrum. Graphical
Illustrations are provided of the waveforms of reproduced speech signals, and informal judgements are given of their quality. Because of the complex nature of these schemes, the time involved in arranging the processing of enough speech for quality tests would have been very large. It was decided instead to base subjective judgements on short, 10s sections of processed speech which were conveniently processed off-line, and for which programs could be written in high-level languages. Details of this processing scheme appear in Part 2 of this thesis. Although the results of quality tests would be very valuable, the short processed speech sections obtained in this way provide useful indications of the performance of these schemes.
Chapter 4. Pitch-Synchronous Reiteration Synthesis

Chapter 3 introduced a reiteration synthesis technique for interrupted speech which could maintain phase continuity of the vocal excitation during reproduction of voiced speech. This 'pitch-synchronous' reiteration is performed by a recursively connected adaptive waveform predictor whose transfer function \( H_p(z) \) is:

\[
H_p(z) = z^{-m}
\]

where the delay \( mT_s \) includes a whole number of pitch periods. This chapter considers some techniques for estimation of the speech pitch period, and describes the performance of this waveform predictor for reiteration synthesis of interrupted speech.

For this type of reiteration to function correctly, the transmitted speech sections must contain at least one period of a voiced speech signal. The transmission period therefore imposes a constraint on the largest speech pitch period which may be reproduced. It has to be chosen to allow correct operation with an adequately large proportion of speech signals. To choose the transmission period, therefore, information is required about the frequency of occurrence of different durations of pitch period in conversational speech. The author is not aware of any published data of this kind - certainly the collection of such statistics would involve analysis of very large amount of speech from a wide range of subjects. However, Holmes (24) has recently compiled histograms of the distribution of pitch periods in approximately 15 minutes of male conversational speech. The speech samples were edited to remove long silences, and their pitch was then estimated at 20ms intervals by an algorithm based upon the cepstrum transform (43) which could reliably measure periodicity up to 15ms. A normalised histogram and cumulative histogram of these results is shown in Fig. 4.1.1. The cumulative histogram shows that 99% of pitch periods in the test samples were shorter than 12ms. The synthesis techniques investigated here were accordingly designed to operate with voiced speech of period up to 12ms.

The improvement achieved by pitch-synchronous synthesis of interrupted speech
Fig. 4.1.1. Distribution of Vocal Pitch Period in Conversational Speech.
would depend on the accuracy of the method employed for pitch measurement. The problem of pitch measurement has received much attention (43, 44, 45, 46, 47) as it is required in many existing speech processing systems. In practice it has been found difficult to achieve reliable pitch measurement, which in some cases has led to the use of less efficient alternatives - for example 'voice excitation'(48) for vocoders. The requirements in this application are fortunately less stringent.

A voiced/unvoiced decision is not required, as the pitch estimated during unvoiced speech segments has been found to be relatively unimportant in its reproduction. Also, the waveform predictor of equation 4.1.1. may still operate satisfactorily if the estimated pitch period is a small integer multiple of the correct value, as long as the estimated value is less than the transmission period. In the event of a completely erroneous decision, causing phase discontinuities to be introduced during reiteration, some correct pitch periodicity would usually be retained as a transmitted speech section would normally contain several pitch periods.

The pitch measurement schemes investigated here can be broadly classified into two types, according to the speech waveform characteristic used - either its 'peakedness' or its approximate periodicity. These types are introduced below.

As voiced speech is produced by excitation of the resonant vocal tract by relatively low frequency 'puffs' of air through the glottis, the speech waveform generally exhibits a sharp rise in amplitude after each excitation. These regular peaks are usually clearly visible (see Fig. 1.2.2.) and various amplitude peak-picking schemes have been proposed (44) to locate them. These schemes are attractive because of their possible practical simplicity, but are liable to give erroneous results with noisy or multi-peaked speech waveforms, and also with insufficiently peaked speech which often occurs. These errors may be reduced by employing several simple peak detectors - for example, with different time constants or operating separately on the positive and negative peaks of the waveform and its differential (45), and using coincidence detection to obtain a best result.

Pitch measurement schemes have also been described which operate on the
periodicity of the speech waveform by computing its autocorrelation\(^{(46)}\) or cepstrum\(^{(43)}\), for example, and then locating the peak which is found at a delay equal to the pitch period. These produce more accurate results in general than the peak-picking schemes described above, but are also more complex. By using all the available waveform rather than merely its peaks, these methods are less affected by noise, and previous signal filtering and clipping operations than the peak-picking methods and can in general operate over wider ranges of pitch. The cepstrum of voiced speech, through its ability to separate the spectral contributions of the vocal source and tract, usually displays a far sharper peak at the pitch period than the autocorrelation, thereby enabling more accurate and reliable pitch estimation. Some interesting pitch measurement techniques have recently been suggested by Schroeder\(^{(47)}\) which attempt to locate harmonic frequencies of the vocal excitation from the speech amplitude spectrum, and then take their highest common factor to be the pitch frequency. This method has apparently produced encouraging results.

The following sections describe investigations into the waveform predictor of equation 4.1.1. for pitch-synchronous reiteration. The applications of pitch measurement schemes operating on speech 'peakedness', and its periodicity are described separately in sections 2 and 3.

4.2. 'Peakedness' Pitch Measurement

As mentioned in the last section, voiced speech waveforms usually display sudden rises in amplitude at the epochs of glottal 'pitch' pulses which may be simply located by peak-picking methods. Although of limited reliability, these methods offer the possibility of simple implementation and were used to gain initial impressions of the improvement in performance achieved by pitch-synchronous reiteration of interrupted speech.

These peak-picking schemes have to neglect close waveform peaks resulting from a single pitch pulse - the interval between detected pulses therefore has to be effectively constrained above a certain minimum value. The selection of this minimum detection interval is a compromise, as normally encountered pitch periods
can vary widely from about 2.5ms in female speech up to the currently assumed maximum of 12ms. A simple detection scheme which worked satisfactorily with shorter pitch periods would probably detect spurious minor peaks in longer pitch periods. The use of amplitude thresholds to overcome these problems is undesirable as the pitch detector should operate over a wide range of speech amplitudes, and also during rapid amplitude variations. Errors are also caused through insufficient peakedness in the speech signal, resulting either from the characteristics of the original speech or from signal distortions during transmission.

The pitch measurement scheme selected for these investigations operated simply by full-wave rectifying and low-pass filtering the speech signal, and taking the peaks of the resulting waveform to indicate the epochs of pitch pulses. This scheme was found to have a relatively short detection delay in comparison with normally encountered pitch periods, of about 2ms. It could therefore operate satisfactorily merely with the waveform in the transmitted speech sections and was incorporated in the receiver for these investigations.

4.2.2. Method of Operation

A block-diagram of the scheme is shown in Fig. 4.2.1. It operates by detecting the peaks of the full-wave rectified, low-pass filtered speech waveform. The low-pass filter is required to eliminate multiple peaks arising from individual pitch pulses while introducing as little delay as possible, and a 2-pole, 2-zero filter cutting off at 170 Hz was finally chosen. Its transfer function $H_f(z)$ for a 10 kHz sample rate was:

$$H_f(z) = 0.00251 \left( \frac{1 + z^{-2}}{1 - 1.8857 z^{-1} + 0.89190 z^{-2}} \right)$$

The two numerator zeros, at $\pm j$, were found to improve the performance of the filter in this application and were included as they involved only simple computation. The peaks in the processed signal were located by detecting when its differential became negative. A small hysteresis threshold was included in this
Fig. 4.2.1. Block Diagram of Peak-Picking Pitch Detector.
detection to reduce the effect of any small remaining waveform inflections, and of arithmetic roundoff errors, of about 1% of the peak value of the processed signal. Fig. 4.2.2 illustrates the waveforms appearing during the operation of this scheme with a speech signal of rapidly changing amplitude.

4.2.3. Performance Testing

Performance tests were conducted with this pitch measurement scheme to enable optimum values of parameters to be chosen, and to establish the types of speech for which it operated correctly.

Initial tests were conveniently performed by off-line computer simulation on short (e.g. 2s) sections of speech. Graphplots were produced of speech waveforms on which were marked the positions of detected pitch pulses. It was possible to estimate from these suitable values of detection parameters and also to measure the approximate detection delay. This varied between approximately 0.9ms and 2.5ms, depending on the degree of peakedness of the speech waveform. However for more general tests with more widely varying types of speech the following method was employed.

By means of an FM tape recorder, recordings of speech were reproduced at one eighth of real speed. Sections of voiced speech then sounded like very slow, deep 'growls' in which individual pitch pulses were separately audible. The pitch measurement scheme was simulated on the Modular-1 computer to operate with the one-eighth speed analogue speech signals and to output a narrow voltage pulse whenever a pitch pulse was detected. The pitch detection algorithm was programmed in assembly language, using fixed point arithmetic for greatest computational speed. Although not necessary here, this speed would be required later when the algorithm was used in simulations of pitch-synchronous reiteration. The detection pulses were added to the slowed speech signal and applied to headphones. In this way the timing of the detection pulses could be easily compared with the original pitch pulses. Multiple detection of individual pitch pulses, and individual detection failures, were audible.

Parameters were finally chosen for reliable pitch detection at longer pitch
After full-wave rectification

After low-pass filtering

Detected pitch pulses

Fig. 4.2.2. Waveforms appearing in Peak-picking Pitch Detector.
periods, under which condition pitch pulses in shorter periods tended to be missed. This compromise was chosen after initial experiments in which it was noted that the greatest improvement relative to asynchronous reiteration was achieved in the reproduced quality of lower pitch, rather than high-pitched speech. This was most probably because a transmitted speech section would contain fewer periods of a low-pitch speech signal and the disturbance of its periodicity by asynchronous reiteration would be greater.

The pitch detector tended to miss pulses in very quiet speech, probably because of the small amplitude threshold included in the detection process. When fed with unvoiced speech signals or noise, pitch pulses were produced randomly at a very low rate. The detection accuracy, however, was considered adequate to allow initial judgements of pitch-synchronous processing.

4.2.4. Simulation of Pitch-Synchronous Reiteration

The reiteration procedure was designed to operate from the location of pitch pulses in the transmitted speech sections, as this information was provided directly by the pitch detector. As illustrated by equation 4.1.1. the predictor $H_p(z)$ consists of a delay of an integer number of pitch periods, and the method of simulation is described below.

As each speech section was received the first and last pitch pulses were located. If less than two pulses were detected, asynchronous reiteration was performed. Otherwise the interval between these pulses, $N_p$ samples, was assumed to contain an integer number of pitch periods. The last $N_p$ locations of the buffer storing the speech section were then used to simulate the delay required for the waveform prediction operation. Rather than repeatedly moving signal samples from the end of the buffer, and moving all the remaining samples along one storage location, the delay line was simulated by a 'circular store' technique. At each clock interval, a particular buffer location was taken to represent the end, and start of the delay line. The output sample was removed from this location and the new input sample stored away in it. This reference point was moved by one
location at each clock interval, and reset to the start of the buffer when it reached the end. Instead of moving the entire contents of the buffer at each clock interval, therefore, merely its start and finishing point were moved. This technique was used to simulate delay throughout these investigations.

To obtain best results from this pitch detector, it was necessary to minimise the delay at the start of each transmitted signal section before it began to operate correctly. This was effectively reduced by leaving the state of the low-pass filter simulation unaltered between the end of each transmission period and the start of the next, rather than resetting the variables to zero.

4.2.5: Results

All experiments with speech signals were performed with equal transmission and interruption periods, to allow the subjective improvement obtainable by pitch-synchronous operation to be judged. The processes of interruption and pitch-synchronous reiteration were simulated in real time with a 10 KHz sample rate on the Modular-I computer, using the 'telephone bandwidth' speech samples and ancillary equipment described earlier. From informal judgements during listening tests, best results were obtained with a transmission period of approximately 20ms. Under these conditions the reproduction of pitch and 'character' was greatly improved by pitch-synchronous processing, although the reproduced speech retained the harsh and 'discontinuous' nature of asynchronously reiterated speech. The improvement was particularly noted with male, rather than higher-pitched female speech. The transmission period of 20ms, longer than previously found optimum for asynchronous reiteration, was necessary because each transmitted speech section had to contain at least two pitch pulses to enable pitch-synchronous reiteration to be performed. If the transmission period was not an integer multiple of the speech pitch period, the number of pitch pulses within a transmitted section would depend on the phase of the signal within it - that is, whether the incomplete period at the end of the transmitted section happened to contain a pitch pulse. To guarantee that the transmitted speech section contained at least two pitch pulses, therefore, the transmission period had to have at least twice the duration
of the longest encountered pitch period. The transmission period then required twice the duration needed for the waveform predictor to operate. This indicates that pitch extraction would in fact be best performed on the continuous speech signal available at the transmitter, and the pitch information then sent through the transmission channel.

Occasional pitch detection errors were not separately audible, unless only two pitch pulses were erroneously detected very close together. In this case a small part of the transmitted section was rapidly reiterated, producing a short 'squeak' sound in the processed speech. These were not prevalent, however, as this pitch detector tended to miss pitch pulses rather than to detect spurious ones responsible for this effect. In general it was considered that pitch-synchronous operation achieved significant improvements in the performance of reiteration synthesis. The reproduction of speech pitch and 'character' was improved, in spite of imperfections in the pitch detection strategy, and although the harshness of asynchronous reiteration still remained. This effect may have been accentuated by the long transmission period, of about 20ms, which was necessary for pitch detection.

The following section describes some more accurate and reliable methods of pitch measurement which operate on waveform periodicity rather than its 'peakedness'.

4.3. 'Periodicity' Pitch Measurement

A more reliable and accurate pitch measurement scheme operating from the periodicity of a signal will now be considered. Schemes of this kind typically operate by computing the autocorrelation or cepstrum of short sections of speech, and searching the resulting data for a peak which appears at a delay equal to the pitch period. As these methods do not use special waveform characteristics, for example, peakedness, but only rely on its approximate periodicity they are far less affected by commonly encountered signal impairments such as amplitude limiting and noise than the techniques described in the last section. The computational effort involved, however, is greater.
Methods based upon the autocorrelation function are initially attractive because much work has been done previously on simple methods of implementation (46). However, as explained later, the cepstrum (43) function of a section of voiced speech usually displays a sharper, more pronounced peak at the pitch period and allows more reliable and accurate measurements. The pitch measurement method developed for the current investigations is based upon the cepstrum, and was found to produce results of more than adequate accuracy for these investigations.

Pitch measurement schemes of this type tend to produce erroneous results if the pitch period is changing very rapidly, or alternates between two values (39). Errors in the first case have not been found important here, as such conditions apparently only occur rarely and for short periods. In the second case the computed pitch period tends to be a multiple of the average value, which is satisfactory here.

Pitch measurement schemes of this type have to be incorporated in the transmitting end of the channel where computations may be performed on the uninterrupted speech signal. The pitch information for a transmitted speech section may be sent in place of one speech sample. As the transmitted speech sections would typically contain over 100 samples, this information would occupy less than one per cent of transmission time.

In the following subsections the use of autocorrelation, and cepstrum techniques for pitch measurement is considered. The algorithm used currently for pitch measurement is then described, together with details of performance of a pitch-synchronous reiteration scheme in which it is used.

4.3.2. Autocorrelation Methods

The power spectrum of a short section of voiced speech displays periodic peaks, spaced by an interval of \( \Delta f \) where

\[
\Delta f = \frac{1}{T}
\]  

(4.3.1.)

where \( T \) is the period of the vocal excitation. As a result, the Fourier transform of this power spectrum - that is, autocorrelation of the section of speech waveform -
generally displays a peak at a delay T. As the observed peak is usually fairly pronounced, the autocorrelation has been suggested as a basis for pitch measurement methods. Typically the discrete autocorrelation of a section of speech of approximately 25ms duration would be computed, and searched for a peak over the approximate range of 2-12ms, the delay of the peak being taken as the pitch period. From examination of such autocorrelation of sections of voiced speech, however, it is noticed that the required peaks are often very broad and that spurious peaks occasionally exhibit higher amplitude than the required one.

An explanation of these poor results is given below, in terms of the vocal model discussed in Chapter 3: s(t) represents a periodic impulse source whose frequency is equal to the glottal excitation frequency; h_s(t) represents the convolved effects of the glottal volume flow waveform and the impulse response of the vocal tract, and s_o(t) designates the resulting speech signal. That is:

\[ s_o(t) = s(t) \star h_s(t) \]  

(4.3.2.)

The speech power spectrum may be expressed in terms of the squared moduli of the Fourier transforms of these quantities, as follows:

\[ |S_o(f)|^2 = |S(f)|^2 \times |H_s(f)|^2 \]  

(4.3.3.)

The autocorrelation, r_{so}(\gamma), of the speech signal is the Fourier transform of its power spectrum:

\[ r_{so}(\gamma) = F \left( |S_o(f)|^2 \right) \]  

(4.3.4.)

\[ = F \left( |S(f)|^2 \times |H_s(f)|^2 \right) \]

Therefore

\[ r_{so}(\gamma) = F \left( |S(f)|^2 \right) \star F \left( |H_s(f)|^2 \right) \]  

(4.3.5.)

\[ = r_s(\gamma) \star r_h(\gamma) \]

The effects of the vocal excitation and vocal tract therefore become convolved in the autocorrelation of the speech signal, causing broad and multiple peaks. As a result, autocorrelation pitch measurement schemes are only capable of limited accuracy and reliability.
4.3.3. Cepstrum Methods

The cepstrum overcomes the disadvantages of the autocorrelation function through its ability to separate the effects of vocal excitation and tract. It is defined as the Fourier transform of the logarithm of the power spectrum and its advantages are briefly explained below. From equation 4.3.3:

\[ |S_o(f)|^2 = |S(f)|^2 \times |H_s(f)|^2 \]  

(4.3.6.)

Taking logarithms,

\[ \log |S_o(f)|^2 = \log |S(f)|^2 + \log |H_s(f)|^2 \]  

(4.3.7.)

In the log (power spectrum) the effects of vocal source and tract become additive, instead of multiplicative. \(|H_s(f)|^2\) introduces a slow smooth wave in the log (power spectrum), while the power spectrum \(|S(f)|^2\) of the relatively low frequency periodic vocal source adds close ripples spaced by \(\Delta f = 1/T\). The interval between these ripples may be found by taking the Fourier transform of the log (power spectrum) to yield the cepstrum:

\[ F(\log |S_o(f)|^2) = F(\log |S(f)|^2) + F(\log |H_s(f)|^2) \]  

(4.3.8.)

In the cepstrum, a function of time, the effects of vocal source and tract are added instead of convolved as after autocorrelation. The close ripples in the log (power spectrum) from the vocal excitation produce a sharp peak at a delay equal to the pitch period, \(T\). The power transfer function \(|H_s(f)|^2\), representing the spectral effects of the vocal tract and glottal volume flow function, produce a separate broader peak at the origin. Examples of cepstra of speech, with further discussion of these points, is given by Noll(43).

The pitch measurement algorithm used currently is shown in block form in Fig. 4.3.1. The log (amplitude spectrum) is first computed of a 25.6ms section of speech, sampled at a 10 KHz rate. Hamming weighting is applied before spectrum analysis to reduce 'leakage' in the computed spectra. This is found to reduce the random noise in the computed cepstra, and to produce more pronounced peaks at the pitch period. The amplitude spectrum is computed, rather than power spectrum as...
Fig. 4.3.1. Block Diagram of Cepstrum Pitch Estimation.
it requires less dynamic range and is more amenable to fixed point computation. The log (amplitude spectrum) is then Fourier transformed to yield the cepstrum, a real and even function. As the log (amplitude spectrum) is non-band limited and contains aliasing distortion this last operation may only be performed approximately by discrete methods. The cepstrum may be computed as exactly as required by interpolating extra values in the amplitude spectrum, between those initially computed, before taking logarithms and Fourier transforming. This was not found necessary here, however, as the only characteristic of interest in the cepstrum is the location of the sharp peak at the speech pitch period which is not disturbed by these aliasing effects.

It is necessary for computational convenience to limit values of speech amplitude spectra above a minimum positive value before computing their logarithms. Currently this lower value was set far below normally encountered values of amplitude spectra. It is possible, however, that higher values of this lower limit might be usefully employed to reduce the effects of noise in the amplitude spectra. At higher frequencies the log (amplitude spectrum) does not exhibit any periodicity as signal power becomes lower than noise power. These frequency ranges apparently have the effect of adding noise throughout the computed cepstrum. For the application of pitch measurement, cleaner cepstra might be computed if the lower positive limit for the spectral amplitudes was raised to eliminate these low values at high frequencies.

The pitch period is estimated from the cepstrum by searching it for a peak over the range 1.0 - 12.7ms after application of linear weighting (1 at 1.0ms, 5 at 12.7ms)\(^{(43)}\). This linear weighting produces improved results because close ripples in the amplitude spectrum, resulting from longer pitch periods, tend to have low amplitude because of the initial Hamming weighting applied. Cepstral peaks obtained from speech with long pitch periods tend to have lower amplitude as a result, and may fall below the level of spurious peaks nearer the origin. This linear weighting, although only an approximate solution, allows reliable results to be obtained and is simple to implement.
4.3.4. Performance Testing

The performance of this pitch measurement scheme was tested with normal speech, to estimate its accuracy and also the range of pitch period over which it could operate.

Programmes were written to simulate the pitch measurement scheme on the Modular-1 computer. These were written to use 16-bit fixed point arithmetic. The programming techniques employed for the computation of discrete Fourier transforms, moduli of Fourier coefficients and logarithms are discussed in Part 2 of this thesis. Graphplots of cepstra computed in 16-bit fixed point arithmetic in this way were compared with those computed in floating point arithmetic in this way were compared with those computed in floating point arithmetic of higher precision. Minor differences between the results were easily visible, although the peak at the pitch period remained accurately located and of high amplitude. The computer accepted analogue speech signals as input, and output a voltage proportional to the estimated pitch period. Speech signals at 1/32 real time were used, and were displayed together with the 'pitch' output voltage on oscilloscopes. The pitch period was computed at intervals corresponding to 12.8ms in real time. It was possible to recognise segments of voiced speech and to closely examine the computed pitch period. At this speed reduction, the pitch period tended to change slowly and smoothly during voiced speech, and individual erroneous pitch estimates were clearly visible.

This pitch detection scheme was found to produce very reliable results over the range of pitch period encountered in the test speech samples of about 2.5ms - 11ms. The large peak about the origin in the cepstrum was found always to be substantially below 1 ms, and peak-picking could be reliably performed over the range of 1.0 - 12.7 ms. It was noticed that the pitch measurement scheme would sometimes begin to indicate correct results during the gradual onset of voicing before any waveform periodicity had become visible to an observer. During voiced segments of both male and female speech the estimated pitch period was apparently
only rarely in error, and the scheme was considered very suitable for these investigations.

To allow a closer examination of the errors in pitch estimates, a speech section of 10 s duration was digitised at a 10KHz rate and the sample values were stored on paper tape. Its fundamental frequency was estimated at 12.8ms intervals by this pitch measurement scheme. Also, using the Modular-1 computer, sections of this speech sample were displayed at the same intervals, from which the pitch period was estimated visually. These visual estimates relied upon temporal signal characteristics such as its peakedness which were related directly to individual glottal excitations. The existence of any voicing was also noted in each interval. These two sets of pitch data are displayed in Fig. 4.3.2. The errors between the computed, and visually estimated values during the periods of voicing are displayed as a histogram and cumulative histogram in Fig. 4.3.3, which shows the accuracy of this pitch measurement scheme. This result is perhaps also of wider significance: this pitch measurement scheme, employing the cepstrum transform, operates from the spectral periodicity of voiced speech, whereas the visual pitch estimation was temporal signal characteristics. It is of interest that these two methods, relying on different signal characteristics, can produce results in such close agreement in spite of the non-stationarity of speech.

4.3.5. Simulation of Pitch-Synchronous Reiteration

Speech sections are transmitted to the receiver together with the estimated number of signal samples Mg in a current pitch period, and the entire transmitted section is stored in a buffer in the receiver. The delay of m signal samples required by the predictor is computed to include all complete pitch periods within the transmitted section:

\[ m = \lfloor i \times Mg \rfloor \]

where \( i = \text{truncated integer} (N_s/Mg) \) (4.3.10)

The last m storage locations of the buffer are used to simulate the waveform predictor. This is performed by the 'circular store' technique described in the
Vocal pitch period estimated by:
(A) - Cepstrum method.
(B) - Visual method.

Fig. 4.3.2. Vocal Pitch Period of Conversational Speech, estimated by Cepstrum and Visual methods.
Fig. 4.3.3. Distribution of Errors in Estimation of Vocal Pitch Period by Cepstrum method.
last section.

It has been assumed so far that the waveform predictor delay would span all complete pitch periods in a transmitted speech section, to enable its characteristics to be reproduced as faithfully as possible and also to minimise the effects of pitch measurement errors. However some simulations were performed in which only the last pitch period was reiterated. In this case the interrupted period would be filled with the most recent signal available and, when no other speech characteristics were predicted, might be expected to join up with the preceding and following transmitted sections with less discontinuity.

These experiments were performed mostly by simulation on the Modular-1 computer, and used the fixed-point cepstrum pitch measurement programmes mentioned earlier. These programmes operated with analogue speech signals at 1/16 real speed, obtained from an FM tape recorder. The processed speech was again recorded on another channel of the tape machine, and then played back at full speed for subjective evaluations.

4.3.6. Results

Experiments were performed throughout with a transmission period of 12.8 ms, using the cepstrum pitch measurement algorithm described above.

When an interruption period of 12.8 ms was used (transmission ratio = 1/2) the reproduction of the pitch and individual character of speech after processing was informally judged to be very good, and it was possible to easily identify individual male and female speakers. When only the final pitch period of each transmitted section was reiterated some roughness was introduced into the processed speech, which also contained occasional short 'squeak' sounds resulting from erroneous estimation of a short pitch period. These effects varied in extent between different speech samples. The processed speech was considered to be improved in quality by the reiteration of all complete pitch periods in a transmitted section, as this reduced the roughness and eliminated the 'squeak' effects. A very slight 'warble' distortion sometimes appeared, however. The decrease in processed
speech quality when only the last pitch period was reiterated probably results from minor waveform discontinuities between the finish and start of the reiterated signal section. As the pitch period of voiced speech varies continuously, the pitch period of the reiterated signal section would in general differ slightly from the computed value over the whole processing period and introduce these discontinuities. This indicates the need to reiterate all pitch periods, even when other speech characteristics are not predicted, to utilise its periodicity and to reduce the frequency of these discontinuities during reiteration.

Experiments with an interruption period of 25.6 ms (transmission ratio = \( \frac{1}{3} \)) revealed a general worsening in quality, in which an effect of 'discontinuity' became apparent. This most probably resulted from the discontinuities in signal characteristics, especially its amplitude, between the end of reiteration and the start of the following transmitted speech section, which would have become larger as longer processing periods were used.

4.4. Summary

These experiments have shown that the quality of speech reproduced from an interrupted speech signal by reiteration synthesis is significantly improved by pitch-synchronous operation. Worthwhile improvements may be obtained even with very simple pitch detection.

It has also been shown that pitch-synchronous reiteration may be performed by a recursively connected waveform predictor. This allows fixed transmission and interruption periods to be used.
Chapter 5. Prediction of Speech Amplitude

This chapter describes the further development of the speech waveform predictor to continue changes in speech amplitude during reiteration. In the short-term model of the vocal mechanism of Chapter 3 (equation 3.1.3.), the amplitude of the vocal excitation was described by $A(t)$. Its time-varying nature may be seen from Fig. 1.2.2., illustrating the waveform of $I_s$ of speech. The speech amplitude may change significantly within about 30 or 40ms, which is of the order of the processing period employed for these investigations. The investigations described in the last chapter demonstrated the significant improvements in the synthesised quality of interrupted speech when the periodicity of the vocal excitation was continued by the reiteration process. These investigations were conducted to determine whether further improvements could be achieved by similarly continuing trends in another prominent characteristic of the vocal excitation, namely its amplitude $A(t)$. It was noted in the last chapter that interrupted speech, synthesised by pitch-synchronous reiteration techniques, still retained a rough quality although the pitch of voiced speech was reproduced well. During that type of reiteration synthesis the amplitude of the reproduced speech was not changed smoothly, but in steps coinciding with the transmission of new speech sections. The resulting amplitude discontinuities most probably contributed to this 'roughness' effect.

The transfer function $H_a(z)$ of the proposed prediction is:

$$ H_a(z) = B z^{-m} \quad (5.1.1.) $$

The coefficient $B$ is set equal to the fractional amplitude growth of the speech signal in the current processing period over the predictor delay of $m$ samples. As before, the predictor generates the signal to fill an interrupted period by repeatedly predicting the next speech sample ahead, each predicted sample also being fed back to its input as it is obtained. During this process the waveform stored within the predictor repeatedly circulates through it, and is scaled in amplitude by $B$ on each pass. Trends in amplitude of the transmitted speech
section are thus approximately continued during the remainder of the processing period. However, as discussed in Chapter 3, the amplitude of the reiterated signal varies approximately exponentially with time as it is scaled by a constant factor on each circulation. Therefore this waveform predictor can only perfectly predict signals whose amplitudes \( A(nT_s) \) vary exponentially. That is:

\[
A(nT_s) = A_0 \exp (knT_s)
\]

(5.1.2.)

where 

\[
k = \frac{\text{loge} B}{mT_s}
\]

In the case of real speech signals, \( A(t) \) only approximates to this exponential model over short periods of time. It is therefore necessary to calculate the value of \( B \) so that the exponential amplitude variation of the predicted signal will be a subjectively acceptable alternative to that of the original signal. In spite of these limitations, the prediction of speech amplitude by the predictor of equation 5.1.1. has been found to achieve a useful improvement in the performance of the reiteration process, especially with longer processing periods.

5.2. Minimum Mean-Square Estimation

Initial investigations were conducted into the computation of \( B \) by a minimum mean square prediction error criterion. The method is outlined below.

From equation 5.1.2. the predicted value \( s_p(n) \) of the \( n \)th speech sample is given by:

\[
s_p(n) = B s(n - m)
\]

(5.2.2.)

where \( s(n) \) are the sample values of the original speech signal, and \( m \) is the predictor delay. The error \( E(n) \) in the prediction of the \( n \)th speech sample is:

\[
E(n) = s(n) - s_p(n) = s(n) - B s(n - m)
\]

(5.2.2.)

It is assumed here for convenience that the predictor receives actual speech samples as input, and only the errors \( E(n) \) resulting from a single prediction operation are considered. During reiteration, however, the predictor would usually receive previously predicted signal samples as input and prediction errors
would accumulate. The results obtained from this analysis are of value, however, as it would be expected that when the errors $E(n)$ are minimised by a mean square criterion, the cumulative errors incurred during reiteration would also be about optimum by the same criterion.

The mean square prediction error $E_{ms}$ is given by:

$$E_{ms} = \frac{1}{N_i} \sum_{n} E^2(n)$$  \hspace{1cm} (5.2.3.)

where the sum extends over all samples in the interruption period, of $N_i$ sample intervals, in which the predictor is to be optimum. $E_{ms}$ may be further expressed:

$$E_{ms} = \frac{1}{N_i} \sum_{n} (s(n) - B s(n-m))^2$$  \hspace{1cm} (5.2.4.)

The optimum value of $B$ is obtained by equating the partial derivative of $E_{ms}$ with respect to $B$ to zero:

$$\frac{E_{ms}}{B} = \frac{1}{N_i} \sum_{n} (s(n) - B s(n-m)) s(n-m)$$  \hspace{1cm} (5.2.5.)

$$= 0$$

On solving for $B$,

$$B = \frac{\sum_{n} s(n) s(n-m)}{\sum_{n} s^2(n-m)}$$  \hspace{1cm} (5.2.6.)

This equation states that the optimum value of $B$ is equal to the normalised autocorrelation of the speech signal for a delay of $m$, computed over the interrupted period. The corresponding value of $m$ may be obtained by substituting the value of $B$ from equation 5.2.6. into equation 5.2.4. After rearrangement of terms:

$$E_{ms} = \frac{1}{N_i} \sum_{n} s^2(n) - \frac{1}{N_i} \frac{\sum_{n} s(n) s(n-m))^2}{\sum_{n} s^2(n-m)}$$  \hspace{1cm} (5.2.7.)

When finding the value of $m$ for which $E_{ms}$ is minimum, the first term on the right side of equation 5.2.7. may be neglected as it is independent of $m$. The minimum of $E_{ms}$ occurs when the second term on the right side of equation 5.2.7. is maximum, and the square of the numerator summation may be ignored without altering its location. The optimum value of $m$ is therefore given by the location of the maximum of the normalised autocorrelation $r(m)$ of the speech signal over the
interrupted period, given by:

\[
    r(m) = \frac{\sum_n s(n) s(n - m)}{\sum_n s^2(n - m)}
\]  

(5.2.8)

for values of \( m > 0 \). Equations 5.2.6. and 5.2.8. state that for minimum mean-square prediction error the values of \( B \) and \( m \) are equal to the magnitude and delay respectively of the peak in the normalised autocorrelation of the original speech computed over the interrupted period.

In this application, however, the value of \( m \) is not arbitrary but is chosen so that the predictor delay \( mT_s \) spans a whole number of pitch periods in voiced speech. As mentioned in Chapter 4, the delay of the peak of the normalised autocorrelation above a minimum of about 2ms is usually approximately equal to the pitch period, but is very liable to errors and is not suitable for pitch period measurement. As the value of \( m \) calculated by these techniques cannot be used here, the value of \( B \) is also inapplicable. Because of the oscillatory nature of the autocorrelation of speech, the value of \( B \) could not in general be calculated from an independent value of \( m \). This finding illustrates the clash between subjective and objective criteria in the design of speech processing systems. The value of \( m \) specified for the predictor has to be chosen to reproduce the periodicity of voiced speech, because the distortion of pitch by interruption processing was noted in Chapter 2 to be one of its most disturbing subjective effects. The form of the predictor, and indeed the model of the vocal mechanism (equation 3.1.3.) from which it is derived, are based upon speech characteristics of subjective significance. Although the predictor designed objectively by equation 5.2.6. and 5.2.8. would predict with minimum mean-square error, it would not reproduce well the speech characteristics, such as pitch, which are considered to be subjectively important here. As the final test of a speech processing system is one of subjective acceptance, it has to be designed to optimise fairly vague subjective criteria which apparently may not be easily expressed in engineering terms.
5.3. Rectified Sum Estimation

In this application a method is required for the calculation of $B$ whereby $m$ may be separately specified and which can also operate satisfactorily with unvoiced speech waveforms. The following method was investigated, in which the rectified sum was computed of two sections of speech signal in the current processing period which were relatively displaced by $m$ samples. The value of $B$ is estimated by dividing the rectified sum of the forward-shifted signal section by that of the other section. That is:

$$B = \frac{\sum_{n=1}^{N} s(n + m)}{\sum_{n=1}^{N} s(n)}$$  \hspace{1cm} (5.3.1.)

This would be computed at the transmitter from the uninterrupted speech waveform within a processing period. This expression is simple to compute as, apart from one division, the arithmetic only involves the absolute addition of speech sample values and is amenable to fixed point computation. If the speech amplitude varies exponentially, i.e. $s(n + m) = B \cdot s(n)$, then the value of $B$ given by equation 5.3.1. is exact. In other cases, of non-exponential amplitude variation and in unvoiced signals, this value of $B$ is still found to be satisfactory. The estimated value of $B$ varies smoothly with $m$ and, as required, involves no constraints on $m$.

The summations in equation 5.2.1. have to be performed over complete pitch periods to obtain accurate results, as the power in varied speech signals is not distributed uniformly throughout each pitch period. After each opening of the glottis, at which a puff of air is emitted into the bottom of the vocal tract, the speech waveform displays a sudden rise in amplitude which then decays throughout the pitch period up to the next excitation. The power is concentrated in the initial part of each pitch period - this is illustrated by the peakedness of speech waveforms, which may be seen in Fig. 1.2.2. The value of $N$ in equation 5.3.1. is conveniently set equal to $m$, the predictor delay which also spans a whole
number of pitch periods.

The values estimated for $B$ by equation 5.3.1. from the section of speech waveform of Fig. 1.2.2. are displayed graphically in Fig. 5.3.1. The speech amplitude, $A$, is also displayed:

$$A = \frac{1}{m} \sum_{n=1}^{m} s(n)$$  \hspace{1cm} (5.3.2.)

This is equal to the denominator of equation 5.3.1., normalised by $m$. The values of $B$ and $A$ were estimated at 12.8ms intervals. Fig. 5.3.1. shows that the speech amplitude $A$ varies quite smoothly. Also, by comparison with the speech waveform from Fig. 1.2.2., it shows that the value of $B$ estimated during voiced speech also varies in a smooth fashion. However, during periods of quiet, and during the sudden onset of voicing the value of $B$ can suddenly adopt very large values. This probably results from a noise impulse, or the start of a section of voiced speech appearing in the forward-shifted signal section summed in the numerator of equation 5.3.1. but not in the section summed in the denominator. Of course, in these cases the signal amplitude variation deviates widely from the exponential model assumed here. However, large values of $B$ could cause short bursts of large amplitude signals to be generated during reiteration, especially at the onset of voicing. These effects would be expected to worsen as longer processing periods were used.

5.3.2. Results

The processes of interruption and reiteration synthesis incorporating this waveform predictor were simulated to allow their effect upon speech quality to be judged. The simulations were performed on sections of speech of 10s duration. A sample rate of 10KHz, and transmission period of 12.8ms were used. The value of $B$ was computed from the signal at the start of each processing period, including the transmitted speech section, so that it described the rate of amplitude variation in the transmitted speech section most accurately. This ensured the best continuity of signal amplitude during reiteration. The simulation was
Fig. 5.3.1. Graph showing Variation of Speech Amplitude and Growth Coefficient with time. (Growth coefficient calculated to continue trend in last transmitted speech section)
programmed similarly to that of pitch-synchronous reiteration described in the last chapter, except that signal samples were multiplied by B as they were removed from the predictor delay, before being restored.

This prediction of speech amplitude was informally judged to achieve a significant improvement in the quality of the synthesised speech. By comparison with the reiteration techniques discussed in the last chapter, the processed speech was generally less rough and 'noisy'. When an interruption period of 12.8ms was used (transmission ratio = \( \frac{1}{2} \)) the speech only suffered minor degradation during processing. With longer interruption periods of 25.6ms or 38.4ms (transmission ratios of \( \frac{1}{3} \), \( \frac{1}{4} \)) some roughness again became apparent. In particular, with an interruption period of 38.4ms a very rough, 'warble' effect was introduced. From visual examination of the reproduced waveform it was noticed that the predicted speech amplitude often contained large errors towards the end of the interruption period. Occasionally very large signal amplitudes were reproduced, of several times the original peak amplitude, which were individually audible as loud 'clicks'. As a result this amplitude prediction technique achieved little improvement with longer interruption periods. The requirement for amplitude prediction was indicated by the subjective improvement achieved in the reproduced speech quality with shorter processing periods. However an important potential advantage of reiteration techniques incorporating these waveform prediction methods is that they might enable longer interruption periods to be used, with consequentially greater transmission efficiency. To achieve this, an improved technique for the estimation of B is required. In the next section, a technique is described whereby the value of B is calculated to achieve amplitude continuity between the reiterated signal and the following transmitted section.

5.4. Interpolation of Speech Amplitude

The effects are now described of calculating the value of B to achieve approximate amplitude continuity between the reiterated signal and the following transmitted section. Rather than to continue the amplitude trends in the current
transmitted sections, the value of $B$ is calculated here from the amplitudes of the current and following transmitted sections. This is estimated so that the amplitude of the reiterated signal is changed to that of the following transmitted section during the course of the interrupted period. In the case of purely exponential amplitude variation, of course, both of these approaches are identical. Effectively, this value of $B$ causes the amplitude of the reiterated signal to follow an exponential function which is interpolated between the amplitudes of the current and following transmitted sections.

This method necessitates extra signal storage within the receiver as the waveform predictor for the current reiteration cannot be designed until the following transmitted speech section is available. The synthesis process therefore involves a delay of one processing period. However, to its advantage, this computation of $B$ may be performed in the receiver and does not therefore occupy the transmission channel.

The ratio, $C$, of the amplitude of successive transmitted speech sections is estimated from their rectified sums:

$$C = \frac{\sum_{n=1}^{m} s(n + N_p)}{\sum_{n=1}^{m} s(n)}$$

(5.4.2.)

where a processing period contains $N_p$ samples, (i.e. $s(n)$ and $s(n + N_p)$ are the signals of successive transmitted sections) and $m$ samples, the predictor delay, spans a whole number of pitch periods. $B$ must be set equal to the fractional change in speech amplitude over the predictor delay of $m$ sample intervals. As the amplitude of the reiterated signal is varied geometrically with time, the value of $B$ may be expressed:

$$B = C^{(m/N_p)}$$

(5.4.2.)

Then, from equation 5.4.1,

$$B = \left[ \frac{\sum_{n=1}^{m} s(n + N_p)}{\sum_{n=1}^{m} s(n)} \right]^{m/N_p}$$

(5.4.3.)
Fig. 5.4.1. shows the graph of B calculated in this way at intervals of 12.8 ms over the section of speech signal shown in Fig. 1.2.2. The processing period was 25.6 ms ($N_p = 256$). The speech amplitude, A of equation 5.3.2, is also displayed. The graph shows that the values of B are more consistent, and less prone to errors, than those computed by equation 5.3.1. (see Fig. 5.3.1.) This is especially so during the onset of voicing, when good amplitude prediction is most desirable.

When this value of B is used in the waveform predictor of equation 5.1.1, the amplitude of the reiterated signal will be changed in an exponential fashion throughout the interrupted period up to that of the following transmitted section. The errors in the amplitude of the reiterated signal resulting from its exponential growth were investigated in the following way. A sample of conversational speech approximately 3 s in length was split into sections of 12.8 ms duration. The amplitudes $A(i)$ and $A(i + 2)$ of the $i$th and $(i + 2)$th section were calculated from equation 5.3.2. From these, a value $A_p(i + 1)$ was estimated for the amplitude of the $(i + 1)$th section on the assumption of geometric growth:

$$A_p(i + 1) = A(i) \times \left( \frac{A(i + 2)}{A(i)} \right)^{1/2}$$

(5.4.4.)

The mean fractional amplitude error, $E$, between $A_p(i + 1)$ and $A(i + 1)$ was computed over all $L$ sections of the speech sample, as below:

$$E = \frac{1}{L-2} \sum_{i=2}^{L-1} \frac{A_p(i + 1) - A(i + 1)}{A(i + 1)}$$

(5.4.5.)

The amplitude error $E$ was less than 20% for the speech sample investigated. This was taken to indicate that the exponential model for speech amplitude was valid in practical terms within the period spanned by two signal sections, of 25.6 ms. The experiment was then repeated to predict the value of $A(i + 2)$ from $A(i)$ and $A(i + 4)$ as follows:

$$A_p(i + 2) = (A(i) \times A(i + 4))^{1/2}$$

(5.4.6.)

The mean amplitude error between $A_p(i + 2)$ and $A(i + 2)$, computed as before, was
Fig. 5.4.1. Graph showing Variation of Speech Amplitude and Growth Coefficient with time. (Growth coefficient calculated for amplitude interpolation between last and next transmitted speech sections)
found to increase to approximately 60%. The relative displacement of the signal section on the right side of equation 5.4.6 is 51.2ms. The large amplitude estimation error in this case indicates that the exponential model is inadequate in objective terms over this interval. However the errors result largely from the smoothing effect of this estimation technique on sudden changes in speech amplitude. In particular, the value estimated by equation 5.4.6. might be several times the correct value for one or two signal sections following a sudden reduction in amplitude. The fractional error summed in equation 5.4.5. would then exceed 1 for short periods. However, as the amplitude of the synthesised speech would still vary smoothly this prediction technique might possibly satisfy subjective criteria.

5.4.2. Results

The processes of interruption and reiteration were simulated with speech signals, to permit some judgements of the quality of the reproduced speech. The reiteration process incorporated the predictor of equation 5.1.1. and the value of B was computed from equation 5.4.3. for approximate amplitude continuity. The simulation was performed with a speech sample of about 10 s duration. A sample rate of 10KHz was employed, with a transmission period of 12.8ms. The reproduced quality of speech processed in this way with an interruption period of 12.8ms was considered to be good, and the speech only suffered minor distortion. When a 25.6ms interruption period was used (transmission ratio = 1/2), the processed speech again began to take on a rough quality, as described in the last section. When the interruption period was extended to 38.4ms (transmission ratio = 1/3), this distortion worsened, although still being much less disturbing than the effects encountered with the previous amplitude prediction techniques. Perhaps the most noticeable effect of this processing with an interruption period of 38.4ms was to 'smooth out' some phonemes. Plosives in particular sounded less distinct in the reproduced speech. The individual processing periods began to be separately audible, imparting an effect of discontinuity, as mentioned previously in cases where long processing periods were investigated. Although the reproduced
speech was intelligible, its quality was poor. The impressions of 'discontinuity' imparted by the reproduced speech might be reduced by prediction of the short-term amplitude spectrum, to be described in the next chapter. However, the 'smoothing' of the reproduced speech amplitude which became evident here, results from the simple exponential variation of characteristics produced by this type of first-order predictor, and cannot be simply avoided. This suggests that these prediction techniques are only applicable when the processing period is less than approximately 50ms. Further work is required to fully assess the subjective effects of these prediction errors.

Fig. 5.4.2. illustrates the waveform of 0.1s of speech before and after interruption and reiteration synthesis with the techniques described here. Transmission and interruption periods of 12.8ms were used. The accurate reproduction of periodicity, and the smooth amplitude variation in the reproduced signal may be clearly seen.

5.5. Summary

Methods for prediction of speech amplitude during reiteration synthesis of interrupted speech are described.

Simulation of these processes with speech signals have produced encouraging results, and demonstrated the need for them. However these investigations have also shown the limitation of the exponential model assumed for speech amplitude, and have suggested that predictors of the type investigated here could only be employed with processing periods less than about 50ms.
Waveforms (12.8 mSec. transmission and interruption periods):
(a) Original speech signal.
(b) After interruption.
(c) After interruption, and reiteration.
(d) After interruption, and adaptive prediction.

Fig. 5.4.2. Graphs of Reproduced Speech Waveforms after Interruption Processing.
Prediction of Amplitude Spectrum

The short-term model of the vocal mechanism described in Chapter 3 (equation 3.1.3.) may be reconsidered here:

\[ S_0(f) = A \cdot S(f) \times H_s(f) \]  
(6.1.1.)

During generation of a speech signal whose Fourier transform is \( S_0(f) \), \( A \) and \( S(f) \) represent the slowly varying amplitude and fine spectral characteristics of the vocal excitation. \( H_s(f) \) represents the combined spectral effects of the vocal tract and of the non-Impulsive nature of the glottal acoustic excitation during voiced speech. This chapter describes the development of the speech waveform predictor from this model to account for its remaining characteristic - the short-term spectrum \( H_s(f) \). Investigations in this direction have reached an advanced stage, although subjective assessments of the performance of the complete predictor in interruption and reiteration of speech signals have yet to be made. These results are presented to complete the derivation of the waveform predictor, and to allow comparisons to be made with other prediction techniques.

As described in Chapter 3, the changes in speech amplitude spectrum within an interrupted period may be approximately predicted by including a simple zero-phase non-recursive filter \( H_{cz}(z) \) in cascade with the previously described predictor \( H_a(z) \) (equation 5.1.1.). These are expressed:

\[ H_{cz}(z) = a(0) + \sum_{i=1}^{N-1} a(i) \ (z^i + z^{-i}) \]  
(6.1.1.)

and

\[ H_a(z) = B \ z^m \]  
(6.1.2.)

The weighting sequence of \( H_{cz} \) spans \( 2N-1 \) sample intervals, and \( N \) is typically 8 or 16. The transfer function \( H_c(z) \) of the complete predictor is the product of \( H_a(z) \) and \( H_{cz}(z) \):

\[ H_c(z) = B \ z^{-m} \left[ a(0) + \sum_{i=1}^{N-1} a(i) \ (z^i + z^{-i}) \right] \]  
(6.1.3.)

By this means, small changes are made in the amplitude spectrum of the reiterated section of speech signal on each circulation through the predictor, to continue trends between adjacent transmitted speech sections. The amplitude
growth coefficient \( B \) may be conveniently incorporated in the filter coefficients:

\[
H_C(z) = z^{-m} \left( b(0) + \sum_{i=1}^{N-1} b(i) (z^i + z^{-i}) \right) \tag{6.1.4.}
\]

where \( b(i) = B \times a(i) \).

The prediction of the amplitude spectrum may be considered as a generalisation of the amplitude prediction discussed in the last chapter. Instead of the amplitude of the speech signal as a whole, the amplitudes of separate frequency bands are now predicted independently. A zero phase filter is used in this application so that the speech amplitude spectrum may be modified without significantly altering its phase, and waveform characteristics. In particular, this allows the envelope periodicity of voiced speech to be reproduced faithfully \((37, 38)\).

In common with the amplitude prediction discussed in the last chapter, this prediction process modifies spectral amplitudes exponentially in the reiterated signal. Its satisfactory operation relies on the validity of an exponential model for their variation with time. It was noted that speech amplitude could not usefully be approximated by a function of the form \( y = a \cdot \exp(bt) \) over periods greater than about 50ms. As variations in both speech amplitude and short-term spectra result largely from motions of the musculature within the vocal mechanism, it might be expected that speech amplitude spectra may similarly only be considered exponential over periods of about 50ms. Further work is required to estimate the subjective effects of the exponential modifications of signal characteristics which are produced by the predictor.

In each processing period the predictor coefficients are computed so that the amplitude spectrum of the reiterated signal varies approximately continuously between those of the preceding and following transmitted section. The amplitude spectrum of the following transmitted section is divided by that of the preceding one to obtain the fractional change over the processing period. For each frequency this ratio is raised to the power \((m/N_p)\) to obtain the fractional change of amplitude spectrum over the predictor delay of \( m \) sample intervals. The non-recursive filter \( H_{cz} \) is then designed to achieve this frequency response.
design is described in the following section. Finally section 3 compares the complete predictor designed by these techniques with predictors designed for the same application by minimum mean square error criteria.

6.2. Prediction Filter Design

The design of the filter $H_{cz}$ for prediction of the speech amplitude spectrum is now discussed.

Firstly, the $N$-point amplitude spectra are calculated of the transmitted speech sections preceding and following the current interrupted period at equispaced frequencies $f(i)$ where

$$f(i) = i / (2NT_s); i = 0 \ldots (N-1) \quad (6.2.1.)$$

As in the estimation of speech amplitude discussed in the last chapter, the estimation of the amplitude spectrum has to be performed over a whole number of periods of voiced speech signals because of the significant non-uniformity of the energy distribution within the fundamental period. In this case it was performed over all complete periods in the transmitted section, spanned by $m$ sample intervals. The number of samples in this signal section was increased to 128 by the addition of zeros. A transmission period of 12.8ms is assumed; that is, 128 sample intervals at a 10KHz sample rate. This enabled its discrete Fourier transform to be calculated by the efficient 'Fast Fourier Transform' (49) algorithm to yield the Fourier coefficients $F(i)$ at 64 positive frequencies. The values $S(i)$ of the $N$-point amplitude spectrum were obtained as follows:

$$S(i) = (128/m) \times 1/k \sum_{n=k(i-\frac{1}{2})+1}^{k(i+\frac{1}{2})} |F(n)| \quad (6.2.2.)$$

where $i = 0, 1, \ldots (N-1)$

$$k = 64/N.$$  

The scaling factor $(128/m)$ was included to account for the zeros added before computation of the Fourier coefficients. As the impulse response of the vocal mechanism is not time-limited, the sampled amplitude spectrum $S(i)$ generally includes aliasing distortion.
Accordingly the prediction of spectral amplitude may only be performed approximately. Fig. 6.2.1 illustrates the amplitude spectra of successive 12.8ms sections through a 1s sample of speech, estimated by equation 6.2.2.

The ratio \( R(I, L) \) of the amplitude spectra of the speech sections in and following the \( L \) th processing period may now be expressed:

\[
R(I, L) = \frac{S(I, L+1)}{S(I, L)}, \quad I = 0, 1, \ldots (N-1) \quad (6.2.3)
\]

\( R(I, L) \) expresses the fractional change in the speech amplitude spectrum over the \( L \) th processing period of \( N_p \) samples. For the predictor design, the fractional change over the predictor delay of \( m \) samples, \( R_p(1, L) \), is required. As the spectral amplitudes of the reiterated signal section are modified geometrically on each circulation through the predictor, the values of \( R_p(I, L) \) may be expressed:

\[
R_p(I, L) = \left( R(I, L) \right)^{m/N_p} \quad (6.2.4)
\]

The spectral ratios \( R_p(I) \) for the predictor design were specified at frequencies \( f(I) \) where

\[
f(I) = \frac{I}{2NT_s}, \quad I = 0, 1, \ldots (N-1) \quad (6.2.5)
\]

The zero phase non-recursive filter \( H_{cz} \) is designed as follows to match this frequency response. It may be shown\(^{(49)}\) that the filter \( H(z) \) having transfer function:

\[
H(z) = a/2 \times (z^n + z^{-n}) \quad (6.2.6)
\]

exhibits a periodic real frequency response \( H_f(f) \) where

\[
H_f(f) = a \cos \left( 2\pi f n T_s \right) \quad (6.2.7)
\]

by means of the substitution \( z^{-1} = \exp(-j2\pi fT_s) \), which is the Fourier transform of unit delay \( z^{-1} \). A general zero-phase frequency response \( H_{of}(f) \) may be synthesised by adding filters of this type. That is:

\[
H_{of}(z) = a_o + \sum_{i=1}^{N-1} a_i (z^i + z^{-i}) \quad (6.2.8)
\]

and

\[
H_{of}(f) = a_o + \sum_{i=1}^{N-1} a_i \cos \left( 2\pi f i T_s \right) \quad (6.2.9)
\]
Fig. 6.2.1. Short-time Amplitude Spectra during 1 second of Speech (from equation 6.2.2; $N = 16$).
This shows that the \( a_i \) coefficients required in the design of filter \( H_0(z) \) are the Fourier coefficients of the frequency response \( H_{of}(f) \). In the present application the required frequency response is specified by the \( N \) sampled values \( R_p(i) \). The coefficients \( b(i) \) required in the prediction filter \( H_{cz}(z) \) may now be conveniently obtained by computing the discrete Fourier transform of the \( R_p(i) \) values. The resulting filter weighting sequence, containing \( 2N-1 \) values, is a periodic, truncated version of the infinite length weighting sequence which would match the required frequency response. This truncation is equivalent to multiplying the required weighting sequence by a symmetrical, rectangular window of \( 2N \) samples duration. The frequency response of the filter, \( H_{of}(f) \), becomes the convolution of the required frequency response, \( R_{pf}(f) \), with the Fourier transform \( H_r(f) \) of the rectangular window. That is:

\[
H_{of}(f) = R_{pf}(f) \ast H_r(f)
\]

where

\[
H_r(f) = \frac{\sin \left( \pi fT \right)}{\pi f}
\]

\[
h_r(t) = \begin{cases} 
1, & -T/2 \leq t \leq T/2 \\
0, & \text{elsewhere}.
\end{cases}
\]

Here the rectangular window has a duration of \( 2NT_s \), and the zeros of its Fourier transform \( H_r(f) \) occur at frequencies \( 1/(2NT_s) \), \( i = 1, 2, \ldots \). If the convolution of equation 6.2.10 is evaluated at a filter design frequency, of \( k/2(2NT_s) \), where \( k = 0, 1, 2, \ldots (N-1) \), these zeros will coincide with all the other design frequencies. This rectangular truncation does not then alter the filter performance at its design frequencies, but can introduce significant deviations in between these because of the broadly spread sidelobes of \( H_r(f) \). These may be reduced by weighting the filter coefficients with a smoothly-shaped window, such as an even raised-cosine function. This has a smoother Fourier transform, of more limited extent. (50) This technique was employed here. Because of the broader central peak in the Fourier transform of a raised-cosine function, however, its use reduces the spectral resolution obtainable with a given value of \( N \).

As the impulse response of filter \( H_{cz} \) has a duration of only \( 2NT_s \), its frequency
response is time-limited. As mentioned earlier, its required frequency response is in general not time-limited. The accuracy achieved in the prediction of the speech amplitude spectrum will depend on the value of N employed. Speech signals typically have up to four resonances below about 4KHz\(^4\) and at least eight frequency samples would be required to resolve these, at intervals of approximately 500 Hz. With a sampling frequency of 100\(\mu\)s as currently employed in these investigations, equation 6.2.5. indicates that a value of N of about 10 is required to allow this frequency sampling interval. As the filter coefficients are weighted by a raised-cosine window, however, the weighting sequence has to be twice as long to achieve the same spectral resolution. An N-value of 16 has been used mostly in these investigations.

Some practical examples of this design procedure are shown in Fig. 6.2.2. The short-term spectra are illustrated of four successive 12.8ms section of voiced speech. These are estimated at 16 frequencies between 0 and 4.5KHz, as described earlier. The ratios of the successive amplitude spectra are then shown followed by the corresponding 31-point filter weighting sequences. The spectral ratios were estimated in this case over 12.8ms intervals, whereas the delay of the speech waveform predictor would generally be less than this. The filtering required would then be correspondingly less. The weighting sequences are shown to consist mainly of a large central peak, while the non-central weights are of much lower magnitude.

The weighting sequence of the complete waveform predictor of equation 6.1.4. spans \(m + N - 1\) sample intervals. A problem of implementation can arise if \(m + N - 1\) exceeds \(N_+\), the number of samples transmitted in each speech section. Inadequate data will then be available to fill the predictor delay. This situation would only exist at the commencement of reiteration as the extra data required would soon be created by the predictor. This problem could be avoided by imposing a maximum limit on \(m\) of \(N_p - N + 1\), instead of \(N_p\), but this would reduce the largest reproducible pitch period. However, as the predictor weighting
Fig. 6.2.2. Illustration of Design of Prediction Filter.

a(1-4): Discrete amplitude spectra of successive 12.8 ms speech sections.

b(1-3): Ratios of successive amplitude spectra.

c(1-3): Corresponding filter weighting sequences.
sequence consists largely of one main peak at a delay of m samples while the other weights are small, an approximate solution to this problem might well be adequate. One possibility would be to merely zeroise any predictor delay remaining unfilled after receipt of a transmitted speech section. Alternatively, as the pitch period of the speech in a transmitted section is known, the extra delay could be filled initially with waveform from an adjacent period in the transmitted section.

The performance of the complete waveform predictor of equation 6.1.4. is illustrated in Figs. 6.2.3. and 6.2.4. These show the waveforms and predictor impulse responses produced during the interruption and reiteration of a speech section of 100ms duration with 12.8ms transmission and interruption periods. Fig. 6.2.3. displays the section of voiced speech, with the impulse responses of the waveform predictors calculated successively for interrupted periods commencing at 12.8ms, 38.4ms, 64.0ms and 89.6ms. Fig. 6.2.4. illustrates this section of speech after interruption, followed by reiteration using these predictors in the successive interrupted periods. Two examples of these diagrams are shown, designated by the suffix (a) or (b), which illustrate different sections of speech waveform. The speech sample shown in Figs. 6.2.3.(b) and 6.2.4.(b) was of a short, unstressed 'a' sound from the phrase 'having a bash'. Of about 60ms duration, this was one of the shortest isolated voiced sounds encountered in the speech samples examined, and usefuly demonstrates the performance of these prediction techniques. The waveforms of the synthesised speech signals demonstrate that the characteristics of the original signal are well retained during reproduction. The smooth variation of signal characteristics between transmitted sections appears to be an adequate approximation to the variation in the original speech, even in the case of the rapid variations in sample (b). In particular, the periodicity and the 'peakedness' of the speech signals are well retained. It may be envisaged, however, that if longer processing periods were used the deviation between the characteristics of the original signal, and the 'smoothed' characteristics of the reiterated signal could
Fig. 6.2.3(a). Original speech section, and predictor weighting sequences at intervals of 25.6 ms.
Fig. 6.2.4(a). Illustration of synthesised speech signal obtained using predictors of Fig. 6.2.3(a).
Fig. 6.2.3(b). Original speech section, and predictor weighting sequences at intervals of 25.6 ms.
Fig. 6.2.4(b). Illustration of synthesised speech signal obtained using predictors of Fig. 6.2.3(b).
Increase rapidly.

6.3. Comparison with least-Squares Error Prediction

The complete waveform predictor of equation 6.1.4., for the reiteration synthesis of interrupted speech, has been derived through considerations of the simple speech model described in Chapter 3 (equation 3.1.3.). The speech characteristics incorporated in this model, however, are those which in previous speech research have been found to be of greatest subjective significance - the amplitude and frequency of the vocal excitation and the short-term spectrum.

There are two powerful incentives for the use of subjectively derived speech models in the design of processing systems. Their success ultimately depends on the accuracy with which they can reproduce these subjectively important characteristics, and the assumptions of a suitable signal model can often result in considerable reductions in the processing involved. However because of the time-varying nature, and complexity of speech signals these models may only be approximate and do not permit any useful objective assessment to be made of a processing system, or comparisons to be drawn between different ones. Indeed, a severe problem in speech research at present is that the objective criteria required for the design of a processing system can often be only vaguely specified in terms of subjective experience, or speech processing 'folklore'. These constraints prompted the design of some linear non-recursive waveform predictors by an entirely objective criterion - namely, for minimum mean square prediction error. Comparisons between them and the subjectively derived predictors described in the last section would enable some judgement to be made of the importance of subjective design criteria in this application.

The minimum mean square error predictor was designed with a weighting sequence of $N_t$ values, equal to the length of a transmitted speech section. It was to predict one sample value ahead with minimum mean square error in the interrupted period for which it was designed. It could then be used for reiteration synthesis in the same way as the predictors just described. Its design will now
be considered. When given the \(N_t\) previous speech samples, \(S(n), S(n-1), \ldots S(n-N_t + 1)\), the predictor estimates the value of \(S(n+1)\). The predicted value, \(S_p(n)\), of \(S(n+1)\) is:

\[
S_p(n) = \sum_{i=0}^{N_t-1} a(i) S(n-i)
\]  
(6.3.1.)

where \(a(i), i=0, 1, \ldots N_t-1\) are the predictor coefficients. The prediction error, \(E(n)\), at the \(n\)th sample is:

\[
E(n) = S(n+1) - S_p(n)
\]  
(6.3.2.)

It is assumed here that the predictor is fed with sampled values of the original speech, rather than previously predicted values as would normally occur during reiteration. Therefore the effects of cumulative prediction errors are neglected here. It is required to minimise the mean square error \(E_{ms}\) over the interrupted period where:

\[
E_{ms} = \frac{1}{N_t} \sum_{n=0}^{N_t-1} E^2(n)
\]  
(6.3.3.)

or

\[
E_{ms} = I (E^2(n))
\]

where \(I\) denotes the operation of averaging over the current interrupted period. When \(E_{ms}\) is minimum, its partial derivatives with respect to the predictor weights \(a_i\) are zero. That is:

\[
\frac{\partial}{\partial (a(j))} \left( I (E^2(n)) \right) = 0, \ \ 0 \leq j \leq N_t
\]  
(6.3.4.)

\(I (E^2(n))\) may be expressed:

\[
I (E^2(n)) = I \left[ S(n+1) - \sum_{i=0}^{N_t-1} a(i) S(n-i) \right]^2
\]  
(6.3.5.)

Then

\[
\frac{\partial}{\partial (a(j))} \left( I (E^2(n)) \right) = I \left[ 2 \left( S(n+1) - \sum_{i=0}^{N_t-1} a(i) S(n-i) \right) S(n-j) \right]
\]  
(6.3.6.)

Rearranging terms,

\[
\left[ \sum_{i=0}^{N_t-1} a(i) S(n-i) S(n-j) \right] = I \left[ S(n+1) S(n-j) \right]
\]  
(6.3.7.)

If a correlation coefficient \(r(1, m)\) is defined as follows:
then equation 6.3.7. may be expressed:

\[ \sum_{l=0}^{N_t-1} a(l) r(l,j) = r(-1,j) \] (6.3.9.)

This is expanded below as a matrix equation for the case of \( N_t = 4 \):

\[
\begin{bmatrix}
r(0,0) & r(1,0) & r(2,0) & r(3,0) \\
r(0,1) & r(1,1) & r(2,1) & r(3,1) \\
r(0,2) & r(1,2) & r(2,2) & r(3,2) \\
r(0,3) & r(1,3) & r(2,3) & r(3,3)
\end{bmatrix}
\begin{bmatrix}
a(0) \\
a(1) \\
a(2) \\
a(3)
\end{bmatrix}
=
\begin{bmatrix}
r(-1,0) \\
r(-1,1) \\
r(-1,2) \\
r(-1,3)
\end{bmatrix}
\] (6.3.10)

As speech signals are generally non-stationary the square matrix of correlation coefficients on the left side of equation 6.3.10 generally exhibits no symmetry, and all the coefficients have to be calculated individually. Furthermore the correlation matrix, of dimension \( N_t \times N_t \), must be inverted to obtain the predictor coefficients. As \( N_t \) is of the order of 100, these tasks would involve considerable computational effort. A solution becomes feasible, however, if the speech is assumed to be practically stationary over the operating period of the predictor. The correlation coefficient of equation 6.3.8. now becomes:

\[ r(l) = I \left( S(n) S(n-l) \right) \] (6.3.11.)

and is an even function. Equation 6.3.9. now becomes a discrete form of the Wiener (35) equation:

\[ \sum_{l=0}^{N_t-1} a(l) r(-1+j) = r(1+j), \quad 0 \leq j < N_t \] (6.3.12.)

Its matrix equation now becomes (\( N_t = 4 \)):

\[
\begin{bmatrix}
r(0) & r(1) & r(2) & r(3) \\
r(1) & r(0) & r(1) & r(2) \\
r(2) & r(1) & r(0) & r(1) \\
r(3) & r(2) & r(1) & r(0)
\end{bmatrix}
\begin{bmatrix}
a(0) \\
a(1) \\
a(2) \\
a(3)
\end{bmatrix}
=
\begin{bmatrix}
r(1) \\
r(2) \\
r(3) \\
r(4)
\end{bmatrix}
\] (6.3.13.)
The matrix on the left side of equation 6.3.13 is highly symmetrical and may be inverted relatively rapidly. However, the equation 6.3.12 assumes that the speech signal is stationary, and would not be applicable during rapid changes of speech characteristics.

Waveform predictors designed by minimum mean-square error criteria have been used previously in speech processing systems, for predictive coding\(^{(42)}\) and for predicting the response of the vocal tract\(^{(40)}\). The predictors in these systems were designed to predict one sample ahead with minimum mean square error. Larger amounts of signal were predicted by recirculating the predicted signals back into the predictor input, as performed in these current investigations. As a digression, an alternative approach would have involved the design of several predictors to individually predict 1, 2, 3, \ldots samples ahead with minimum mean square error. The signal generated from a single predictor by recirculating its output back to its input, and that constructed from the individually predicted samples from several predictors are apparently not identical except in trivial cases such as of periodic waveforms where exact prediction may be performed. Therefore, the signal generated by a single recirculating predictor designed in this way to predict one sample ahead is in any case apparently not the best linear estimate, by a minimum mean square criterion.

Wiener predictors were designed for each of the interrupted periods of the 100ms section of speech shown in Fig. 6.2.3a. The weighting sequences, of length 128, were calculated from equation 6.3.12. As the characteristics of the speech section shown in Fig. 6.2.3a did not appear to vary rapidly, this design procedure was expected to be approximately valid. The weighting sequences of the resulting predictors are shown in Fig. 6.3.1. By comparison with Fig. 6.2.4a, showing the weighting sequences of the predictors described in the last section for this speech sample, the Wiener predictors have a very different form. Rather than a large peak at the delay of the pitch period, their weights are spread over the entire filter delay. The largest weights occur around zero delay, and no peaks
Fig. 6.3.1. Original speech section, and weighting sequences of Wiener predictors at intervals of 25.6 ms.
are visible at the pitch period.

The waveform synthesised from interrupted speech by these predictors is shown in Fig. 6.3.2. The reproduction of periodicity and waveform characteristics is immediately seen to be worse than that achieved with the previously described predictors. The accuracy of reproduction of the original waveform characteristics varied widely between different reiterated sections. The most visually noticeable distortion in the reiterated waveforms is perhaps their reduced amplitude. In all cases this decayed throughout reiteration, this being most marked during periods of greatest non-stationarity of the original speech. The fundamental envelope periodicity was reproduced in varying degrees, but did not appear as distinct by visual inspection as in the original speech. These errors are probably increased by the slow changes in speech characteristics throughout the speech sections employed. Under these conditions, the design procedure of equation 6.3.12 is only approximate. By visual examination the waveforms reiterated in the second and fourth interrupted periods show greatest similarity to the original waveforms. These also were periods of slowest change in the original speech characteristics.

These results show that predictors designed by these minimum mean square error criteria have a very different form to those described in the last section. They can achieve somewhat similar results in reiteration, however, during periods when the speech characteristics are almost stationary. Prediction errors caused the reproduced signal to have low amplitude, with less pronounced fundamental envelope periodicity. The subjective disturbance caused by these distortions could well be increased by the fact that the signal characteristics introduced would not normally be encountered in speech. This indicates an advantage of the previously described predictor whose structure is derived from a speech model - if given a section of speech signal, it will predict a signal with speech-like characteristics even though these may contain errors. The waveform generated by a Wiener predictor, although it might contain less mean square error, might contain
Fig. 6.3.2. Illustration of synthesised speech signal obtained using Wiener predictors of Fig. 6.3.1.
distortions which were subjectively worse. These errors might have been reduced if the predictor had been designed by equation 6.3.9., using the correlation coefficients calculated by equation 6.3.8. which do not assume stationarity in speech characteristics. Further research in this area would be worthwhile. However, because of the work involved in the individual computation of the correlation coefficients, and in the inversion of the non-symmetric correlation matrix, the calculation of a single predictor would apparently occupy more than one hour on an ICL 1904A computer. Improved computational techniques are required before detailed research will be feasible in this direction. By comparison, the design of a predictor as described in the last section occupies less than two seconds on the same computer.

6.4. Summary

This chapter described the waveform predictor derived from the speech model of equation 3.1.3. This continues the periodicity of varied speech, as well as trends in amplitude spectrum, during the reiteration synthesis of interrupted speech. Waveforms of speech signals after processing by these techniques demonstrate their feasibility, and show that speech characteristics may be reproduced well.

The predictor, designed from a simple speech model, is compared with a predictor designed without such constraints, for minimum mean square error. This latter predictor is found to give worse performance. Prediction errors can cause non-speechlike signals to be generated, and considerable more computational effort is required in their design.
Chapter 7. Conclusion

This thesis has described investigations into the use of interruption techniques for efficient speech transmission. Investigation of various synthesis methods have demonstrated that speech of good quality may be reproduced from interrupted signals, and have shown that these techniques are practically feasible.

Initial investigations were performed into simple processing schemes. These operated by repeatedly outputting, or reiterating, each transmitted speech section in various ways to fill the following interrupted period. It was found that speech signals synthesised in this way always contained significant distortions. From the nature of these distortions, it was concluded (Chapter 2) that, over the range of processing parameters employed in interruption processing, interruption distortion is perceived through both its temporal and spectral effects. It is considered that further useful improvements are unlikely to be achieved in the performance of simple interruption and reiteration synthesis schemes.

To obtain further improvements, a new reiteration synthesis technique was proposed (Chapter 3) which employs adaptive waveform prediction. The speech signal for interrupted periods is predicted from the transmitted sections. Speech periodicity, and trends in short-term spectrum, are continued. The performance of this reiteration method has been demonstrated by simulations with speech signals. Speech of good quality may be reproduced after bit-rate compressions of 2:1 relative to that required for conventional PCM transmission, while bit-rate compressions of 4:1 may be feasible. Detailed subjective assessments of the complete synthesis scheme have still to be performed. This synthesis scheme, incorporating a waveform predictor, is far more general in concept than previous proposals for pitch-synchronous interruption and reiteration (11, 14) with which it may be compared. Also, unlike these, it permits fixed transmission and interruption periods to be
employed for convenience of transmission.

The advantages of interruption techniques arise largely from the fact that sections of the original speech signal are conveyed to the receiver, where they may be employed in the reproduction process. This enables the characteristics of the vocal excitation - important for speech quality - to be reproduced accurately. Also, as the interrupted speech is itself intelligible, a channel may be used for emergency communications if the synthesis equipment fails in the receiver.

The complexity of the system lies largely in the predictor, and in the methods used to compute its parameters. However, interruption techniques have no clear advantages over some other speech communication systems which also employ predictors of comparable complexity. They are inefficient in that the speech signal interrupted at the transmitter is completely discarded. Predictive coding schemes,\(^\text{42}\) for example, transmit the coarsely quantised error between the speech signal and its predicted value, together with regularly estimated predictor parameters. They employ predictors of comparable complexity to those required here, and may use all of the speech waveform in the computation of their parameters. Speech of good quality may apparently be reproduced with about double the bit-rate compression obtainable with interruption processing.

Interruption techniques might be more useful in other areas of speech research - for example, into the processes of speech perception. As they operate in the time, rather than frequency domain the distortion of speech characteristics introduced are very different to those from most other forms of speech processing. They are inherently simple and their effects are widely variable. They might well also provide a useful tool for further development of speech prediction techniques.
References

22. Center for Rate Controlled Recordings, University of Louisville, Louisville, Kentucky, USA.
34. R. Plomp, "Pitch of complex tones", Jnl.A.S.A. vol. 41, no. 6 (1967)


Appendix A Computer Simulation

The experimental investigations described in this thesis have been performed very largely by simulation on a small digital computer. This appendix describes why simulation methods were used in preference to hardware experimentation techniques.

Computer simulation permits great improvements in convenience and experimental flexibility. These improvements arise largely through the modularity which can be achieved in computer programs. Sections of a large program, performing particular functions, may be developed and tested individually. They may also be incorporated in other programs where their particular functions are required. By comparison, some unwanted interactions generally occur when sections of a hardware system are connected, making it very difficult to design such sections in isolation. Because of the need for compatibility of signal and power supply voltages and impedances, it is also difficult to re-use hardware sections in other systems.

A further reason for the use of simulation techniques here was that the processing schemes investigated were described in terms of sampled-data operations. Many of the operations involved - signal delay, for example - are implemented more conveniently by digital techniques working with sampled data than by analogue techniques. As digital computers also operate on sampled data, these simulations were then straightforward to program.

Simulations were mostly performed on a small CTL Modular-1 digital computer. This was equipped with analogue/digital and digital/analogue converters for input and output of analogue signals. It was used on-line, and its speed was adequate to simulate many forms of interruption processing in real time.

Values of processing parameters could be input via a teletype as a simulation program was run. The programs could generally accept very wide ranges of processing parameters - the processing period could be set between 200 s and several seconds. Such a wide range could not be obtained simply in an
experimental hardware system. Although such a wide range might be required only rarely, this demonstrates that the experimental constraints imposed by computer simulation are in general less stringent than those imposed by hardware experimental methods.

**Details of Computer**

A small CTL Modular-1 computer was employed for these investigations. It operated with 16-bit words. Programs were mainly written in assembly language, to gain speed and also to permit efficient use of the A/D and D/A converters. The processor contained five accumulators, two of which were particularly suited for arithmetic operations. The core contained 16 K words, and had a cycle time of 0.75μs. In general, the processor speed was limited by this cycle time. No bulk storage was available. A fast paper tape reader was used for input of programs and data, and two teletypes were available for running and controlling programs.

For this simulation work, the most useful peripherals were probably the 8-channel A/D and D/A converters. These converted between 11-bit (10-bit + sign) numbers, and an analogue voltage range of ±5.12 volts. The analogue step size was 5 mV. The conversion time of the A/D converter was 16μs, and that of the D/A converter was 1μs. Although the maximum speed of a program would be limited by these figures, the program was not held up during these conversions. The program commands controlling these devices merely initiated conversions, after which program execution could continue. 'Clock' facilities were available whereby a conversion could be delayed until a pulse was received from an external clock oscillator.

The processor contained hardware for performing fixed-point multiplication and division, in about 3μs. Floating-point arithmetic, of higher accuracy, was performed by software and required about 100-200μs. To achieve greatest speed in these simulation programs, fixed-point arithmetic was used almost entirely. Results accurate to 0.1% could generally be obtained which was more than adequate for these purposes.
As no bulk storage was available, all processing had to be performed on-line. At each sampling instant, the last processed value would be output from the D/A converter, and a new value input via the A/D converter. In this way, it was only necessary to store a small number of recent sample values and simulation variables in core store. However, the processing of each sample value had to be completed before the next sampling instant. In a sample period of 100μs, for example, the computer could execute about 60-100 instructions which was adequate for many of the simpler forms of interruption processing. These could be simulated in real time – the signal to be processed would be taken from one channel of an analogue tape recorder, and the processed signal immediately recorded on another channel. When more complex schemes were investigated, it became necessary to reduce the sampling rate to allow adequate time for the processing of each sample value. This was made possible by the use of an FM tape recorder, by which speech signals could be slowed down to 1/32 of real speed. This permitted the sample interval to be increased by a factor of up to 32.

This form of processing was found to be very convenient for speech signals. The signals recorded on analogue tape could be examined at all stages of processing by listening tests or visually with an oscilloscope. Such examination would be far more difficult if the signals were stored digitally on magnetic tape for off-line processing.

The computer incorporated multiprogramming facilities. Several programs could reside in store, each with hardware protection against store corruption by other programs. An 'executive' program was always resident in store and controlled the loading, running and deletion of programs. Simulation programs were almost always run by themselves, however, as they generally consumed a large proportion of processor time and had to access the A/D and D/A converters at regular intervals.

Simulation programs were normally arranged to read in any required parameters, and then to enter a continuous processing loop. The programs were stopped by
typing a control character on the teletype to re-enter the executive program. Another program could then be run.

Programming Techniques

The flow of simulation programs was arranged, as far as possible in keeping with computing efficiency, to model the physical system under investigation. This facilitated program development, and also any later modifications required. To illustrate these points, program INT9 is briefly described here. This performed on-line simulation of rectangular interruption and non-recursive reiteration, and its flow diagram is shown in Fig. A.1. It operated on analogue speech signals via A/D and D/A converters, and simulation parameters were typed in on a teletype console. Further documentation of this, and some other simulation programs, appears in Part 2 of this thesis.

The interruption process modelled by INT9 operates in two parts:

- Transmission - signal sample values are regularly transmitted to the receiver where they are output, and also stored successively in a buffer. This continues until \( N_t \) sample values have been conveyed, after which transmission ceases.

- Reiteration - sample values are accessed successively from the start of the buffer and output regularly. The contents of the buffer - that is, the last transmitted section, is reiterated the required number of times to fill the interrupted period. The transmission phase is then recommenced.

These functions are performed by program INT9 as follows. When run, it first requests values of the number of signal samples in each transmission period, \( N_t \), and the number of reiterations to be performed, \( K_A \). When the user has typed these, the program enters a continuous processing loop. The transmission phase described above is simulated by processing loop 1. Signal sample values are regularly accessed from the A/D converter, and retained in processor accumulator 'A'.
NUT – number of samples in each transmitted section.

KA – number of reiterations per transmitted section.

Processor accumulators:
A – used to transfer input and output sample values.

M – holds sample count during interruption and reiterations.

Fig. A1. Flow diagram of simulation program INT9
(rectangular interruption, non-recursive reiteration).
Each sample value is immediately output by the D/A converter, and also stored in consecutive locations of a buffer array AREA. This continues until NT samples have been processed. The count of processed samples is kept in accumulator 'M'.

Reiteration is then commenced. Storage location KT is loaded with -Ka - the negated number of reiterations to be performed. The number is controlled by loop 2. Before each reiteration, KT is tested for equality with zero, and also incremented by 1. When its value is found to be zero, the initial transmission phase (loop 1) is re-entered. While KT remains less than zero, a reiteration is performed by loop 3. Sample values are accessed from successive locations of buffer AREA, and output via the D/A converter. When all NT values have been output to complete that reiteration, loop 2 is re-entered to determine whether any further reiterations are required or whether the transmission phase of loop 1 is to be recommenced.
INTERRUPTION TECHNIQUES
FOR EFFICIENT SPEECH TRANSMISSION
(PART 2 - COMPUTER PROGRAMS)

by
JEREMY SCOTT SEVERWRIGHT, B.Tech.

A Doctoral Thesis submitted in partial fulfilment
of the requirements for the award of
Doctor of Philosophy of the Loughborough University of Technology

Supervisor: M. H. Ackroyd, Ph.D.

Department of Electronic and Electrical Engineering.
Summary.

Some computer programs, and programming techniques for simulation of signal processing schemes are described. These were developed for use in the investigations described in the first part of this thesis.
Contents.

Chapter 1. Introduction. 1

Chapter 2. Utility subroutines. 3

2.2 ALPHAl 5
2.3 DAT1 7
2.4 DATA3 9
2.5 FORM1 12
2.6 FORM2 14
2.7 NUM1 16
2.8 NUM2 18
2.9 NUM3 19
2.10 NUM3 21
2.11 PTSET, PTREAD 22

Chapter 3. Fixed-point signal processing subroutines. 24

3.2 ATAB 26
3.3 COSF 28
3.4 FFT1 30
3.5 HAMM 33
3.6 LOG 36
3.7 MOD 38

Chapter 4. Display subroutines for analogue peripherals on Modular-1 computer. 42

STR1, STR2 44
PLT1, PLT2, PLT3 45

Chapter 5. Collection and reproduction of analogue signals for off-line processing. 46

5.2 Paper tape code. 47
5.2.1 TOP1 48
5.2.2 DATA2 49
5.3 Data collection and reproduction. 50

Chapter 6. Simulation programs. 52

6.2 Assembly language programs.
INT5 52
INT9 53
S207I 54

6.3 FORTRAN programs.
Magnetic tape subroutines.
PE21 58
PROC14 60

61
This part of the thesis describes some of the computer programs which were developed for this project. These fall mainly into two groups:

1) Programs in assembly language for the CTL Modular-1 computer. These were written to operate directly with analogue input and output signals, interfaced with analogue-to-digital and digital-to-analogue converters. Single-word (16 bit) fixed point arithmetic was used throughout, as processing speed was of primary importance. These programming techniques were used for on-line processing of analogue speech signals, and acquisition of sampled speech data for off-line processing.

2) Programs in FORTRAN for convenient off-line processing of small amounts of sampled speech data (eg up to 10s of speech). These programming techniques were very useful for initial investigations of processing schemes, and for detailed examinations of their effects.

Some special-purpose assembly language subroutines were developed to execute commonly required operations in signal processing. By using these, programs could be rapidly developed to perform a wide variety of processes. These were particularly convenient for assembly-language programming. The subroutines executed basic operations such as weighting an array of numbers with a Hamming window, calculation of the discrete Fourier transform by the FFT method, and calculation of the modulus of complex numbers. As the individual subroutines only performed simple basic operations of this kind, it was possible to make them efficient and the instructions required to call them from a main program were simple. Complex forms of processing were executed by calling several subroutines sequentially to operate upon data in the required way. The calculation of the cepstrum transform on the CTL Modular-1 computer, for example, required five or six of these assembly-language subroutines.

The reader is assumed here to have a working knowledge of FORTRAN, and also of the CTL Modular-1 computer and its assembly language. These programs are not presented in any way as examples of good programming, but to illustrate
two points. Firstly, they provide examples of the programming techniques used to simulate the processing schemes investigated in this project. Secondly, they illustrate one approach to the problem of programming signal processing operations which the author has found to be convenient, flexible and efficient.

Many of the programs described here deviate widely from usual 'good programming' practice. For example, none of the signal processing subroutines provide any indication of arithmetic overflow, or of erroneous calling instructions. Such indications are generally unnecessary here, as the processed data is generally displayed graphically or reconstructed into an analogue signal to be judged in a listening test. Programming and processing errors generally become immediately obvious, and there is no need to incorporate error indication which would make programs longer and most probably slower.

Programs are described in the following chapters:

2) 'Utility' Subroutines for CTL Modular-1 computer for routine data input and output, and program control.

3) Signal processing subroutines forCTL Modular-1 computer, for efficient processing of large amounts of data.

4) Graph display subroutines for CTL Modular-1 computer, to drive Tektronix 611 VDU, and analogue X-Y plotter.

5) Programs for data acquisition from analogue signals, and for reconstruction of analogue signals from sampled data.

6) Simulation programs for interruption processing, non-recursive reiteration and synthesis using adaptive predictors.
Chapter 2

Utility Subroutines

These subroutines were written in assembly language, for use with assembly language programs. They provide convenient means for performing commonly required data input and output operations. Their functions include input and output of single-word integer numbers, printout of text for headings and character input for control of program branching and execution.

The subroutines are listed below with brief descriptions:

- **ALPHA1**: for program control from teletype.
- **DAT1**: input of decimal integer from teletype.
- **DATA3**: input of decimal integer from fast paper tape reader.
- **FORM1**: text output on teletype for headings.
- **FORM2**: " " " " " "
- **NUM1**: output of decimal integer on teletype (leading spaces).
- **NUM1A**: " " " " " " (no leading spaces).
- **NUM2**: teletype 'graphplot' output of integer data array.
- **NUM3**: teletype printout of integer data array.
- **PTSET**
- **PTREAD**

These subroutines are written in modular assembly language for CTL assembler MAS1, and link loader FORMAL. These comments apply in all cases:

1) These subroutines all comprise individual LIBRARY modules.
2) The calling program must declare ADX external address constants for all subroutines to be called.
3) The calling program must load the W-register with a suitable offset for LOCAL subroutine variables before calling.
4) The subroutines are entered by ENT instructions, referring to the required ADX constant as operand.
5) All register values remain unaltered on exit, except in the case of some input subroutines which leave the required value in the A-register.
6) The subroutines use READ, WRITE, SUS, TERM and DEBUG executive program commands, and will operate under E1 or E2 executive programs.
7) Most of these subroutines use storage locations in the Z-segment. The locations are declared as required by the subroutines, and can be located anywhere in the Z-segment. The user must remember to allocate a Z-segment of adequate size at LOAD-time (normally only 1 page).

An example of a typical program:

```
TRY: PROGRAM ; EXAMPLE OF PROGRAM TO ILLUSTRATE
PRN: PAR + 3: GLOBAL ; SUBROUTINE CALLING.

.......
.......
LDW X WOFF ; LOAD W-REG WITH 'LOCAL' OFFSET.
.......
ENT X NUM ; PRINT OUT CONTENTS OF A-REG
.......
.......
WOFF:WBASE ; 'LOCAL' VARIABLE OFFSET.
NUM:ADX NUM1 ; EXTERNAL ADDRESS CONSTANT FOR
ENDMODULE ; NUM1 SUBROUTINE.
PROGEND
```

The subroutines will now be described separately.
2.2. ALPHA1

Purpose: Program branching, and control of execution, from teletype.

Description.

On entry, ALPHA1 rings the bell in the specified teletype. The user then types in two characters (without echo). ALPHA1 then compares these with a table of character pairs provided by the calling program. If correspondence is found with the nth pair (n starting at 0), ALPHA1 will exit to the calling program and will increment the P-register value by n while doing so. That is, n instructions following the calling instruction for subroutine ALPHA1 will be skipped. If these contain JMP instructions to various parts of the program, these parts of the program may be entered as required by typing the required character pairs.

If the typed character pair is not found in the list, the teletype bell is rung again and the user may type in another pair.

Details

Before calling, the A,B and M registers must be loaded as follows:

A-register loaded with number of character pairs in list.
B-register loaded with output device number of required teletype.
M-register loaded with M-modifiable indirection constant for first location of the table of character pairs. Each word of the table contains a pair of packed ISO-7 characters.

Storage Requirements

X segment: 33 words.
Y " 5 words (LOCAL).
Z " 17 words.

Programming Example

TEST: PROGRAM ; EXAMPLE PROGRAM FOR ALPHA1 SUBROUTINE.
PRM: CHARRPAIRS+2: IND CHAR: ..............: GLOBAL
USEY
RELSTART CHARRPAIRS
'P2 ; STORE AWAY CHARACTER
'P3 ; PAIRS
'P1
IND YM CHARPAIRS ; INDIRECTION CONSTANT FOR CHAR. LIST.
USEX
......
LDW X LOCL ; SET OFFSET FOR LOCAL VARIABLES.
......
LDA L 3 ; LENGTH OF CHARACTER LIST.
LDB L 1 ; O/P NUMBER (EXECUTIVE TELETYP).
LDM Y INDCHARS ; IND. CONST. FOR CHARACTER LIST.
ENT X ALPHA ; CALL ALPHA1.
JMP PART2 ; EXIT HERE IF "P2" TYPED.
JMP PART3 ; HERE IF "P3" TYPED.
PART1: ...... ; HERE IF "P1" TYPED.
......
PART2: ......
......
PART3: ......
......
LOCL: WBASE ; OFFSET FOR LOCAL VARIABLES.
ALPHA: ADX ALPHA1 ; EXTERNAL ADDRESS OF ALPHA1.
ENDMODULE
PROGEND
; ALPHA1 PROCESS - FOR PROGRAM CONTROL
; A: NO. OF EXIT BRANCHES
; B: T/T OP DEVICE NO.
; M: M-IND. CONST. OF TEXT TABLE
; J. S. S. MARCH 1972. EDN. 3

ALPHA1: LIBRARY
ALPHA1: MODULE

ZAL+16: FILE
USEZ
RESTART ZAL+2

ADF ZAL+7: TO RING BELL
1
0
1
13S

ADF ZAL+14: TO READ CHARACTERS
3
0
1

USEX
AL1: PROCESS
LDA YW .2
LDB X AL4
STA ZB .0; STORE DEVICE NOS.
ADA L 1
STA ZB 7

AL2: LDB X AL4; RING BELL
WRITE
TERM
SUS

LDB X AL4+1
READ; READ TWO CHARACTERS
TERM
SUS

LDA ZB .6
SFTL S L L 8
ADA ZB 7
STA YW 4; PACK CHAR. INTO (W+4)

Listing of ALPHA1 process.
Listing of ALPHAl (continued).
2.3 DAT1

Purpose: Input of decimal integer from teletype.

Description:

On entry, DAT1 rings the bell in the specified teletype. The user may then type in the successive digits of the required integer (without echo). These input characters are recognised:

- Digits 0-9
- + and - signs for the first character only (+ sign optional).
- Space, and semi-colon to terminate number and cause subroutine to exit.

If an invalid character is typed, or if the number causes overflow out of a 16-bit word, the teletype bell is rung and the user may retype the whole number.

On exit, the number just input is left in the A-register. If the semi-colon terminator is typed, program execution will continue from the calling instruction for DAT1 subroutine. However, if 'space' is typed the immediately following instruction will be skipped.

Details.

Before calling, the B-register must be loaded with the output device number of the required teletype.

On exit, the number must input is left in the A-register, whose previous contents are lost. If the semi-colon terminator is used, DAT1 subroutine will exit normally to the calling program with the P-register containing its pre-entry value. If the 'space' terminator is used, however, the P-register is loaded with its pre-entry value +1 on return. This causes the following instruction to be skipped. Simple program branching and control can thus be arranged. This facility is very useful where it is necessary to type in an arbitrary number of values for a program. The 'space' terminator might be used throughout, except for the very last number which may be terminated by 'semi colon', for example. A JMP instruction to the next section of program may then be included in the location immediately following the call to subroutine DAT1.
See Chapter 2.1 for details of calling instructions.

**Programming Example.**

TEST: PROGRAM ; EXAMPLE PROGRAM TO ILLUSTRATE
PRN: FRED:VAL:GLOBAL ; USE OF DAT1 SUBROUTINE

....

LOW X LOCL ; OFFSET FOR LOCAL VARIABLES.

....

LDB L 1 ; B-REG. HOLDS O/P DEVICE NO.
ENT X DAT ; CALL DAT1 - VALUE PUT IN A-REG.
JMP NEXT ; SEMI-COLON TERMINATOR.
STA Y VAL ; SPACE TERMINATOR.

....

NEXT: STA Y FRED

....

LOCL: WBASE
DAT: ADX DAT1 ; EXTERNAL ADDRESS OF DAT1 SUBROUTINE.
ENDMODULE
Listing of DAT1 process.
Listing of DAT1 (continued).
2.4 DATA3

Purpose: Input of decimal integer from paper tape via fast reader.

Description:

DATA3 subroutine reads in decimal integer numbers from paper tape via the fast reader. One number is read in on each call. It is very useful for programs which require a large number of data values or control parameters, and permits great flexibility of input number formats. The following characters are recognised:

- Digits 0 - 9
- + and - signs for the first number character only (+ sign optional).

The following characters may be used to terminate numbers:
- Space
- Semi-colon
- Line-feed

The following characters are ignored:

- Blank
- Rub-out
- Carriage-return

DATA3 reads through a paper tape until a sign or a digit character is encountered. Following digits are read in until a terminator character is found. The subroutine then exits to the calling program. Numbers may therefore be separated on paper tape by any number of terminator or ignored characters, and the numbers may be of any length as long as their value is representable in 16 bits (-32768 to 32767). During a normal exit of this kind, the input number is left in the A-register and the pre-entry P-register value is incremented by 1 before return. This causes the following instruction to be skipped.

If an invalid character is encountered, or an input number causes overflow out of 16 bits, the subroutine exists to the calling program but does not increment the P-register value. The instruction following the call to DATA3 is then executed. The A-register contains the 8-bit value of the character
which caused the error condition.

Details.

DATA3 subroutine calls PTREAD subroutine to read character values from paper tape. Before calling DATA3, the program must call PTSET subroutine (no arguments) to initialise PTREAD.

The register values on entry to DATA3 are unimportant. On exit the A-register contains the input number, or erroneous character value in case of error. During a normal exit, the instruction following the call to DATA3 is skipped.

See Chapter 2.1 for details of calling instructions.

Storage Requirements.

X segment: 81 words.
Y " : 7 (LOCAL).
Z " : 0

Programming example:

TRY: PROGRAM ; ILLUSTRATES USE OF
PRN: VAL...GLOBAL ; DATA3 SUBROUTINE.
....
....
LOW X LOCL ; SET OFFSET FOR LOCAL VARIABLES.
....
ENT X SET ; CALL PTSET INITIALLY
....
....
ENT X DAT ; CALL DATA3
JMP ERROR ; ERROR EXIT
STA Y VAL ; NORMAL EXIT
....
....
ERROR: ... ; TAKE SUITABLE ERROR ACTION
LOCL: WBASE ; LOCAL OFFSET CONSTANT
SET: PTSET ; EXTERNAL ADDRESSES FOR
DAT: DATA3 ; PTSET AND DATA 3
ENDMODULE
PROGEND
Listing of DATA3 process.

..DATA3.PROCESS...READS DECIMAL INTEGER FROM PAPER TAPE
;IF O.K., VALUE LEFT IN A; EXITS TO (P+2)
;IF ERROR, LAST CHAR. LEFT IN A; EXITS TO (P+1)
;CALLS PTREAD.
;J.S.S. MARCH 1972. EDN. 2

DATA3:LIBRARY
DATA3:MODULE
DATA3:PROCESS

LDA L 0; ZEROISES:
STA YW 1; OP. NUMBER
STA YW 5; SIGN

DB1 ADW L 7.
ENT X PTRD
SBA L ?
STA YW 4; STORE NEXT P/T CHAR. IN (W+4)

TSTL A=0; IGNORES:
JMP DB1; RUNOUT
SBA L 141
TSTL A=0
JMP DB1; CAR. RET
SBA L 114
TSTL A=0
JMP DB1; RUNOUT

LDA YW 4
SBA L 10; TERMINATORS:
TSTL A=0
JMP DB4; LINE FEED
SBA L 150
TSTL A=0
JMP DB4; SPACE
SBA L 27
TSTL A=0
JMP DB4; SEMI-COLON

LDB L 1
LDA YW 5
TSTL A<0,A>0; FIRST DIGIT?
JMP DB2
STA YW 5; YES, SIGN=1
LDA YN 4
SBA L 43
TSTL A=0; +SIGN?
JMP DB1
SBA L 2

Listing of DATA3 process.
TSTL A<0, A>0; -SIGN?
JMP DB2; NO
SBB L 2; YES
STA YW 5; SIGN=-1
JMP DB1

DB2: LDA YW 4; CHECK PARITY
SFTL S R 8
CPYL RA B1 EQV NOT1
STA YW 6
SFTL S A R 1
SFTL S A L 1
SBA YW 6
TSTL A<0, A>0
JMP DB5; PARITY ERROR

LDA YW 4; DECODE ISO-7 CHAR.
SFTL S L L 9
SFTL S L R 9
SBA L 48
STA YW 6; VALUE LEFT IN (W+6)

TSTL A<0; CHECK FOR NON-DIGIT CHARS.
JMP DB5; ERROR
SBA L 9
TSTL A>0
JMP DB5; ERROR

LDA YW 6; FORM NEW OP. NUMBER
MLS YW 5
STA YW 6
LDA YW 4
MLS L 10
ADA YW 6
TST X DB3; OVERFLOW?
JMP DB5; YES
STA YW 1; NO - READ NEXT CHAR.
JMP DB1

DB3: TSTC A0YR

DB4: LDA YW 5; EXIT CODE
TSTL A=0; SIGN ZERO?
JMP DB1
LDA YW 0
AAR L 1
STA YW 0; NO - STEP P; AND EXIT
EXT

DB5: LDA YW 4; ERROR_EXIT

Listing of DATA3 (continued).
STA YH 1
EXT

PTRD:ADX PTREAD

END

ENDMODULE

Listing of DATA3 (continued).
2.5 Form1

Purpose: Convenient text output on teletype for headings.

Descriptions.

Form1 subroutine allows the output of standard headings, etc, to be programmed simply. The required text may be specified conveniently with the facilities provided in CTL ASS assemblers. The blocks of text for all the required headings are stored in packed form in the Z-segment starting from word 1. Each block is terminated by a storage location containing zero value. To output the nth block of text (n starts at 1), the A-register is loaded with the value n before calling Form1. No knowledge is required about the length or starting address of each block of text.

As the text has to be loaded from word 1 in the Z-segment, i.e. a fixed address, it must be loaded into the Z-segment before any other data. In practice, therefore, it is only of use in program segments, which are always loaded first. Form2 subroutine, described later, provides similar facilities for subroutine segments.

Details.

Before calling, the A and B registers must be loaded as follows:

A loaded with number of text block (n = 1, 2, 3, ...)

B loaded with output number of required teletype.

In the assembly-language program, the required blocks of text (including carriage returns and line feeds) are typed within inverted commas. The CTL assemblers will then store the text in packed form. Each block of text must be terminated with a zero-value word.

Programming Example.

```
FORMTEST: PROGRAM ; TO ILLUSTRATE FORM1
PRN: VAL:.....: GLOBAL ; SUBROUTINE
ZRNR: TEXT + 20: FILE ; ALLOCATE Z SEGMENT FOR TEXT
USEZ
0 ; START AT Z-WORD 1
```
START MESSAGE WITH \texttt{CAR.RET} AND \texttt{LINE FEED} IF REQUIRED

FINISH WITH ZERO

SECOND MESSAGE

\texttt{USEX}

\texttt{LDW X LOCL} \texttt{; SET OFFSET OF LOCAL VARIABLES}

\texttt{LDB L 1} \texttt{; O/P TELETYPING NUMBER}

\texttt{LDA L 1}

\texttt{ENT X FORM} \texttt{; PRINT FIRST MESSAGE}

\texttt{LDA L 2}

\texttt{ENT X FORM} \texttt{; PRINT SECOND MESSAGE}

\texttt{\ldots\ldots}

\texttt{LOCL: WBASE}

\texttt{FORM: ADX FORM1} \texttt{; EXTERNAL ADDRESS OF FORM1}

\texttt{ENMOD}

\texttt{PROGEND}
Listing of FORM1 process.
2.6 FORM2

Purpose. Text output on teletype.

Description.

FORM2 Subroutine is a more generalised version of FORM1 (see Chapter 2.5). It outputs strings of packed text in the same way, but the block of text is located in the Y-segment and may start at any offset. Each block must be terminated by a zero-value word. To output a block of text, the M-register is loaded with the offset of its first word before calling FORM2. As the text may be stored anywhere in the Y-segment, FORM2 may be conveniently called by subroutine segments. However, in order to store the text efficiently with CTL ASS assemblers, the programmer must count up the length in words of each block of text. These points are illustrated below.

Details.

Before calling, the B and M registers must be loaded as follows:

B loaded with output number of required teletype.

M loaded with offset of start of required text block.

In the assembly-language program, the required blocks of text (including carriage returns and line feeds if required) are typed within inverted commas. The CTL assemblers will then store the text in packed form. Each block of text must be terminated with a zero value word.

Storage Requirements.

<table>
<thead>
<tr>
<th>Segment</th>
<th>Words</th>
</tr>
</thead>
<tbody>
<tr>
<td>X</td>
<td>17</td>
</tr>
<tr>
<td>Y (LOCAL)</td>
<td>4</td>
</tr>
<tr>
<td>Z</td>
<td>9</td>
</tr>
</tbody>
</table>

Programming Example.

```
FORMTEST: PROGRAM ; TO ILLUSTRATE USE OF FORM2
PRN: FRED+99: TEXT1+8: TEXT2+9: ....:GLOBAL

USEY

RELSTART TEXT1
; LOAD IN TEXT1
" ; CAR/RET AND LINE FEED
```
FIRST MESSAGE".

0

RESTART TEXT2

"SECOND MESSAGE".

0

USEX

....

LOW X LOCL

....

LDB L 1

....

LDM X T1

ENT X FORM

....

LDM X T2

ENT X FORM

....

T1:ADG TEXT1

T2:ADG TEXT2

LOCL: WBASE

FORM: ADX FORM2

ENDMODULE

PROGEND
:FORM2: FOR TEXT OUTPUT (PACKED) FROM Y-SEGMENT
:O := DEVICE NO.
:K := BUFFER OFFSET (TERMINATED BY ZERO)
:J S S APRIL 1972 EDN 1

FORM2:LIBRARY
FORM2:MODULE

DUM:PARAM+5:BUF+1:FILE
USE2
RESTART_PARAM+1
ADF BUF
0
0
1

USEX
FORM2:PROCESS
P:A:DEV:TEXTOFF:LOCAL

LDB X ZOFF; TRANSFER DEVICE NO.
LDA_YH_DEV
STA ZB 0

LDM YW TEXTOFF
LOOP:CPYL_RB M1 EQU NOT1
LDA YB 0; GET NEXT PACKED TEXT WORD
TSI A=0, INC; END OF TEXT?
EXT

LDB X ZOFF
STA ZB_7
SFTL S L L 8
STA ZB_6
WRITE; WRITE UNPACKED TEXT
7720; _DEBG DIY
SUS
JMP LOOP

ZOFF: ADF_PARAM

END
ENDMODULE

Listing of FORM2 process.
2.7. NUM1

Purpose: Output of decimal integer, with leading spaces, on teletype.

Description.

NUM1 prints out the pre-entry value of the A-register as a decimal integer on a specified teletype. Six characters are always output. These consist of leading spaces, a negative sign if required and the digits of the required number. NUM1 will correctly output any 16-bit number. See also Chapter 2.8 for details of NUM1A subroutine. This performs a similar function but does not output leading spaces.

Details.

Before calling, the A and B registers must be loaded as follows:

A-register loaded with the 16-bit integer to be output.

B-register loaded with output number of required teletype.

See Chapter 2.1 for details of calling instructions.

Storage Requirements.

X segment: 57 words

Y segment: 7 (LOCAL) words

Z segment: 13 words

Programming Example.

TESTNUM1 : PROGRAM ; PROGRAM TO ILLUSTRATE
PRN: FRED: VAL: .....GLOBAL ; USE OF NUM1 SUBROUTINE

....

....

LDW X LOCL ; SET LOCAL VARIABLE OFFSET

....

LDB L 1 ; SET TELETYPENumber

LDA Y VAL ; LOAD A-REG. WITH VALUE TO BE OUTPUT

ENT X NUM ; OUTPUT IT.

....

....
LOCAL: WBASE
NUM: ADX NUM1
ENDMODULE
PROGEND

; LOCAL VARIABLE OFFSET CONSTANT.

; EXTERNAL ADDRESS OF NUM1
; NUM1 PROCESS - FOR NUMBER PRINTOUT ON SPECIFIED DEVICE.
; A: = NUMBER, B: = OP DEVICE NO.
; J. S. APRIL 1972. EDN. 3

NUM1: LIBRARY
NUM1: MODULE

PY: PY+11: FILE; PRIVATE FILESPACE
USE2
RESTART_PY+1
ADF PY+6

USEX
PN2: PROCESS
LDA L 160
STA YH 5; STORE (SPACE) IN SIGN LOCATION
LDB L 0
STB_YH_4; OUTPUT BUFFER CNT
SBB L 6
TSIL B = 0, INCR
ADP L 4
ADB X PN4+3
STA ZB 11; LOAD OUTPUT BUFFER WITH SPACES.
SBB X PN4+3
SDP L 6

LDA YH 1
TSIL A<0; NUMBER -VE?
JMP PN3; YES.

PN2: LDB L 0; NO
DIY L 10; DIVIDE NUMBER RESIDUE BY 10
TST X PN4+1; REMAINDER -VE?
ADB L 10; YES
SBA L 1
STA YH 6; NO, STORE RESIDUE AWAY.

CPY_X PN4+2; FORM DIGIT IN (B) INTO ISO-7 CHAR.
SFTL T L R 4
SFTL D A L 7
CPYL RA EGY NOT1 B1 M1
ADA L 48; RESULT IN (A).

LDB YH 4
SBB L 1; DECREMENT BUFFER COUNT.
STB YH 4

Listing of NUM1 process.
ADB X PN4+3
STA ZB 12; STORE 150-7 CHAR. IN BUFFER.
LDA YW 6
TSL L A=0; IS NUMBER RESIDUE ZERO?
JMP PN6; YES.
JMP PN2; NO - LOOP.

PN3: LDA L 45; INPUT NUMBER -VE.
STA YW 5; STORE -VE SIGN.
LDA L 0
SBA YW 1; INVERT NUMBER.
TST X PN4; TEST OVERFLOW.
LDB L 2; YES.
JMP PN2+1
JMP PN2; NO.

PN6: LDB YW 4; OUTPUT.
ADB X PN4+3
LDA YW 5
STA ZB 11; STORE -VE SIGN IN BUFFER IF REQ.
LDB X PN4+3
LDA YW 2
STA ZB 0; STORE DEVICE NO. IN PARAM. AREA.
WRITE; OUTPUT BUFFER.
TERM
SUS.
EXIT; EXIT.

PN4: TSTC A0VR, SKP2
TSTC B0R, SKP2
CPYC RM RA B1 ADD
ADE PY

.END
ENDMODULE
2.8 NUM1A

Purpose: Output of decimal integer, without leading spaces, on teletype.

Description.

NUM1A subroutine functions similarly to NUM1 in all respects, except that no leading spaces are ever output. A negative sign is output if required, followed by the digits of the required number. This subroutine is useful for punching data values on paper tape in a compact format, as no unnecessary leading spaces are included. In general, however, a terminator character must be included to indicate the end of each number for any reading subroutine with which the data might be used.

Details.

See details of NUM1 subroutine (chapter 2.7)

Storage requirements.

- X segment - 58 words
- Y segment - 7 (LOCAL) words
- Z segment - 13 words
Listing of NUM1A process.
Listing of NUM1A (continued).
2.9 NUM2

*Purpose: Printout of integer array as a 'graphplot' on teletype.*

*Description.*

NUM2 accesses successively each location of the specified data array. It prints out the contents of the location as a decimal integer at the left-hand side of the printout. It then prints out an asterisk at a distance across the printout which is proportional to the contents. This is repeated for each location. The successive asterisks produce a simple 'graphplot' effect.

Before commencing output, NUM2 finds the maximum and minimum values in the specified data, and calculates scale factors so that the graphplot fills the available width of paper. This 'auto-ranging' facility permits the subroutine to be called very simply. It is very useful for diagnostic purposes during initial program development.

Two line feeds are output before starting, and after finishing, the graphplot printout. NUM2 calls NUM1 subroutine for printout of decimal integers.

*Details.*

Before calling, the A, B, and M registers must be loaded as follows:

- A-register loaded with the number of locations to be displayed.
- B-register loaded with the output number of the required teletype.
- M register loaded with the M-modifiable indirection constant of the first location to be displayed.

See Chapter 2.1 for details of calling instructions.

*Storage Requirements.*

- X segment: 103 words
- Y segment: 6 (LOCAL) words
- Z segment: 8 words

*Programming Example.*

```
TESTNUM2: PROGRAM ; PROGRAM TO ILLUSTRATE USE OF
PRN: FRED: IND: ARRAY + 99 GLOBAL; NUM2 SUBROUTINE
```
USEY
RELSSTART IND
IND YM ARRAY
USEX

LDM Y IND

ENT X NUM

LDA L 100
LDB L 1
LDM Y IND

ENT X NUM

LOCL: WBASE
NUM: ADX NUM2

ENDMODULE
PROGEND

; "IND" HOLDS M-MODIFIED INDIRECTION
; CONSTANT FOR "ARRAY"

; NO.POINTS TO BE OUTPUT
; SELECT TELETEYPE
; SELECT ARRAY
; CALL NUM2

; OFFSET CONSTANT FOR LOCAL VARIABLES
; EXTERNAL ADDRESS OF NUM2
; NUM2 PROCESS - FOR ARRAY PLOUT ON SPECIFIED TELETYPE. 
; A: ARRAY LENGTH
; B: OP. DEVICE NO.
; M: H-IND. CONST.
; CALLS NUM1 PROCESS. EDN. 2. APRIL 1972. J. S. S.

NUM2:LIBRARY
NUM2:MODULE

ZPZ+7:FILE; PRIVATE_FILESPACE
USE2
RESTART_ZPZ+2; FOR I/I OP.
ADF ZPZ+7
A
0
0
1

USEX
PZ1:PROCESS
LDA YW 2
LDB X PZ+1; MOVE DEVICE NO. TO_FILESPACE
STA ZB 0

LDM L 0; FIND MAX AND MIN OF DATA.
LDA_YW_1
STA YW 4
STA YW 5
LDM L 1
PZ2:SBN_YW_1
TSTL M=0
JMP PZ5
ADM YW 1
LDA_YW_3
SBA YW 4
TST_X PZ+1
JMP PZ3
ADA YW 4
STA YW 4; STORE MAX VALUE
PZ3:LDA_YW_3
SBA YW 5
IST_X PZ+2
JMP PZ2
ADA YW 5
STA YW 5; STORE MIN VALUE
JMP PZ2

PZ4: ADF ZPZ+1; CONSTANTS
TSTC AOVR, A=0, R<0
TSTC AOVR, A=0, A>0, INCM

Listing of NUM2 process.
P25: LDA YW 5; FIND DATA SPAN AND MID POINT
SFTL 5 A R 1
STA YW 5
LDA YW 4
SFTL 5 A R 1
STA YW 4
SBA YW 5
LDB YW 4
ADB YW 5
STA YW 4; DATA SPAN
STB YW 5; MID_POINT

ADW L 6; POSITION PAPER
LDA L 141
ENT X P26
LDA L 10
ENT X P26
ENT X P26
SBB L 6

LDM L 0; START PLOTTING
P27: SBB YW 1
SSTL M=0
JMP P210
ADW YW 1
LDA YW 3
LDB YW 2
ADW L 6
ENT X XNUM1; PRINT OUT NUMBER
SBB L 6

LDB YW 3
LDA L 0
SBB YW 5; SHIFT DATA VALUE
TST X P211; OVERFLOW?
LDB X P211+1
SFTL D A R 1
DIY YW 4; SCALE DATA VALUE
LDB L 0
MLD L 62
ADW L 32; B: HOLDS OFFSET (1-62)

ADW L 6
LDA L 160
P28: SSTL B=0
JMP P29
ENT X P26
SBB L 1
JMP P28

Listing of NUM2 (continued).
.P29: LDA L 170; OUTPUT *,CHR L/F
ENT X PZ6
LDA L 141
ENT X PZ6
LDA L 10
ENT X PZ6
SBW L 6
ADM L 1
JMP PZ7

.P210: ADM L 6
LDA L 141
ENT X PZ6
LDA L 10
ENT X PZ6
ENT X PZ6
SBW L 6
EXT; EXIT FROM PROCESS

.P211: TSTC BOVR; CONSTANTS
32767
XNUM1: ADX NUM1

.P26A: PROCESS; TELETYPewriter PROCESS - OUTPUTS CHAR FROM A-REC
LDB X PZ6+1
LDA YN 1
STA ZB 6
WRITE
?=20; DEBUG
SUS
EXT
PZ6: ADS PZ6A
ADF ZP2+1

END
ENDMODULE

Listing of NUM2 (continued).
2.10. NUM3

Purpose: Printout of integer array on teletype.

Description:

NUM3 prints out the contents of successive array locations as decimal integers. Ten numbers are output on each line. NUM3 permits rapid and compact output of large amounts of data.

A line feed is output before starting data output. NUM3 calls NUM1 subroutine for printout of decimal integers.

Details.

Before calling, the A, B and M registers must be loaded as follows:

A-register loaded with the number of locations to be output.
B-register loaded with the output number of the required teletype.
M-register loaded with the M-modified indirection constant of the first location to be output.

See chapter 2.1 for details of calling instructions.

Storage requirements:

X segment: 36 words
Y segment: 5 (LOCAL) words
Z segment: 16 words
NUM3 PROCESS - FOR ARRAY PRINTOUT ON SPECIFIED DEVICE.
A:=LENGTH B:=OP_DEVICE_M:=M-IND_CONST.
CALLS NUM1 PROCESS.
J. S. S. APRIL 1972. EDN. 3

NUM3:LIBRARY
NUM3:MODULE

ZPR+15:FILE; PRIVATE FILE DECLARATION
USEZ
RESTART ZPR+2
ADF ZPR+15; FOR C/R, L/F OP
2
0
0
1
1

ADF ZPR+15; FOR (SPACE)OP
1
0
0
1
1

USEX
PR1:PROCESS
LDA_YW 2
LDB X PR3
STA_ZB 0; STORE_OP_DEVICE_NO.
LDB X PR3+1
STA_ZB 0

LDM L 0
LDB L 0
PR2:SBM_YW 1
TSL M=0; END OF ARRAY?
EXIT
ADD_YW 1
TSL B=0, INCB; END OF LINE?
JMP PR4

STB_YW 4; NO - OP (SPACE)
LDB_X PR3+1
WRITE
TERM
SUS
JMP PR5

Listing of NUM3 process.
Listing of NUM3 (continued).
2.11. PTSET, PTREAD.

Purpose: Buffered input via fast paper tape reader.

Description.

For many kinds of data input, it is most convenient to read characters singly or in small groups of variable length. Practical problems are encountered, however, if paper tape data is to be read in this way through the C.T.L. fast reader. The rapid application of drive wheel and brake can cause severe scuffing and wear of the tape.

These problems are overcome by PTREAD subroutine. This contains a 32-long buffer for paper tape characters. A program may obtain successive character values by calling PTREAD subroutine. On each call, the next character value is returned in the A-register. When the buffer becomes empty, PTREAD reads in the next block of 32 characters. In this way, a program may obtain paper tape characters singly or as required, while the tape itself is read in blocks of 32 characters for greatly reduced wear.

PTSET subroutine must be called once before any calls to PTREAD subroutine to initialise PTREAD. Paper tape is read in the forward direction.

Details.

Subroutine PTSET must be called initially. It requires no arguments. If the fast reader is off-line, it will wait. Subroutine PTREAD may then be called as required. It requires no arguments. The tape character value is returned in the A-register. If the fast reader is off-line, the program will terminate.

Storage requirements.

X segment: 25 words
Y segment: 4 (LOCAL) words
Z segment: 39 words
Programming Example.

TESTPT: PROGRAM
PRN: FRED:VAL: .....GLOBAL
....
....
LDW X LOCL
....
ENT X SET
....
....
ENT X READ
STA Y FRED
....
....
LOCL: WBASE
SET: ADX PTSET
READ: ADX PTREAD
ENDMODULE
PROGEND

; TO ILLUSTRATE PTSET AND
; PTREAD SUBROUTINE

; CALL PTSET FIRST TO INITIALISE

; CALL PTREAD TO GET A CHARACTER VALUE

; VALUE LEFT IN A-REG.

; LOCAL OFFSET CONSTANT

; EXTERNAL ADDRESSES OF PTSET AND

; PTREAD SUBROUTINES
Listing of PTSET and PTREAD processes.
LDA L 0
SBA L 29; RESET BUFFER COUNT
JMP_PT2

CONST:ADF_ZPT+1

END
ENDMODULE

Listing of PTSET and PTREAD (continued).
Chapter 3.

Fixed-Point Signal Processing Subroutines

This chapter describes some assembly-language subroutines for signal processing applications. They are intended for use with assembly-language programs. The subroutines, operating with 16 bit fixed-point data, are intended for applications where computational speed is of primary importance. They are listed below with brief descriptions:

- **ATAB** - generates a table of sine values required by FFT1.
- **COSF** - calculates cosine values.
- **FFT1** - calculates D.F.T. by FFT algorithm.
- **Hamm** - multiplies a data array by Hamming window.
- **LOG** - calculates log₂ (X) by linear approximation.
- **MOD** - calculates \((x^2 + y^2)^{\frac{3}{2}}\) by polynomial approximation.

The use of fixed-point arithmetic imposes practical limits on the accuracy of these subroutines. Typical applications include on-line spectrum analysis where the results are for "human consumption", e.g., by graphical display. The subroutines are written in modular assembly language for CTL assembler MAS1, and link loader FORMAL. The following comments apply in all cases:

1. These subroutines all comprise individual LIBRARY modules.
2. The calling program must declare ADX external address constants for all subroutines to be called.
3. The calling program must load the W-register with a suitable offset for LOCAL subroutine variables before calling.
4. The subroutines are entered by ENT instructions, referring to the required ADX constant.
5. All register values remain unaltered on exit, except in the case of COSF, LOG, and MOD which return calculated values in the A-register.
6. On exit, execution of the calling program always continues from the instruction following the subroutine call.
(7) None of these subroutines contain any executive instructions, or access any Z-segment storage locations.

(8) No 'execution error' indications are provided.

An example of a typical program:

```plaintext
TRY: PROGRAM          ; EXAMPLE OF PROGRAM TO ILLUSTRATE
PRN: PAR+3: GLOBAL    ; SUBROUTINE CALLING

.....

.....

LDW X WOFF            ; LOAD W-REG WITH 'LOCAL' OFFSET

.....

ENT X COS             ; CALL COSF SUBROUTINE

.....

.....

WOFF: WBASE           ; OFFSET FOR 'LOCAL' VARIABLES

COS: ADX COSF         ; EXTERNAL ADDRESS CONSTANT FOR

ENDMODULE             ; COSF SUBROUTINE

PROGEND
```

The subroutines will now be described separately.
3.2. ATAB

Purpose: Calculations of a table of sine values as required by FFT1 subroutine.

Description.

When computing the DFT of an N-long number sequences, FFT1 subroutine requires a table of $3N/4$ values as follows:

$$\sin (i) = 32767 \cdot \sin \left(\frac{2 \pi i}{N}\right),$$

$$0 \leq i < \frac{3N}{4}$$

ATAB subroutine calculates this table. Subroutine COSF is called to calculate the individual sine values.

Details.

Before entry, the A and M registers must be loaded as follows:

A register loaded with N.

M register loaded with M-modifiable indirection constant of first location of sine table.

See Chapter 3.1 for details of calling instructions.

Storage Requirements.

X segment: 29 words.

Y segment: 6 (LOCAL) words.

Executive Time:

$$T_{exec} = 150 \cdot N \text{ s approximately}$$

Programming Example.

```
TESTATAB: PROGRAM ; TO ILLUSTRATE USE OF ATAB SUBROUTINE
PRN:NUM: INDSIN: SIN+191 ......: GLOBAL
USEY
RELSSTART NUM
256 ; SET VALUE OF NUM (256). SIN TABLE
IND YM SIN ; IS ONLY 192 WORDS LONG THOUGH.
USEX
.....
.....
```
LDW X LOCL

......

......

LDA Y NUM

......

......

......

LDM Y INDSIN

......

......

ENT X TAB

......

......

......

LOCL: WBASE

TAB: ADX ATAB

......

ENDMODULE

PROGEND

; SET OFFSET FOR LOCAL VARIABLES.

; SET NUM IN A.

; SET INDICTION CONSTANT IN M.

; CALL ATAB, TO CALCULATE SINE VALUES.

; EXTERNAL ADDRESS OF ATAB SUBROUTINE
; ATAB PROCESS - COMPUTES SINE ARRAY FOR FFT1 PROCESS
; N := NUMBER OF POINTS FOR FFT
; M := M-IND. CONST. OF SINE ARRAY (LENGTH 3*N/4)
; ARRAY[i] = 32767 * SIN(2*PI*I/N), I = 0 .. 3*N/4-1
; CALLS COSF PROCESS
; J. S. SEYERWRIGHT, DEPT. OF ELECTRONIC AND ELECTRICAL ENGINEERING,
; LOUGHBOURGH UNIVERSITY OF TECHNOLOGY.
; MARCH 1972, EDITION 2.

ATAB: LIBRARY
ATAB: MODULE
LDA L 0
SBA YW 1
SETL 5 A R 2
STA YW 4; -N/4

ADA YW 1
SBA L 1
STA YW 5; 3*N/4-1, UPPER ARRAY COUNT

LDM L 0
AT4_CPYL_RA M1 EOY NOT1
ADA YW 4; A := COUNT-N/4 (QUARTER CYCLE)
LDB L 0
MLD X AT3
DIV YW 1; SCALE BY 2*PI/N

CPYL RA A1 EOY NOT1
ADW L 6
ENT X CSE; WORK OUT COSINE(X)
SEW L 6

LDB L 0
MLD X AT5
DIV X AT6; SCALE VALUE BY 32767/4096
STA YW 1; STORE IN SINE ARRAY

SBM_YW 5
TSTL M=0, INCH; TEST COUNT
EXT
ADM YW 5
JMP AT4

AT3: 3.1415926535897932
AT5: 32767
AT6: 4096

CSE: ADX COSF

ENDMODULE

Listing of ATAB process.
3.3. **COSF**

**Purpose:** Calculation of fixed-point cosine values.

**Description.**

COSF requires an argument in radians $\times \frac{4096}{2^{12}}$. Scaling by $4096$ permits useful accuracy to be obtained by fixed point arithmetic. The resulting cosine value is also scaled by $4096$, and errors do not exceed $\pm 1$. Fixed-point arguments in the range $-32768$ to $32767$ correspond to over two and a half cycles of the cosine function.

Initially COSF modifies the argument to make full use of symmetries in the cosine function. The sign of the argument is checked and it is inverted if negative. If the resulting argument is greater than $2 \pi \times 4096$ this value is subtracted. After these manipulations, the actual cosine calculations need only operate over a smaller range of arguments and are simpler to program in fixed-point arithmetic.

The argument is then shifted by $\pi - \pi \times 4096$ is subtracted and the result inverted if negative. Finally $\frac{\pi}{2} \times 4096$ is subtracted to produce an argument in the range $\pm \frac{\pi}{2} \times 4096$. This is $3\pi/2 \times 4096$ less than the original argument, and conveniently limited in range. Its sine is calculated by a Maclaurin expansion to obtain the cosine of the original arguments. The Maclaurin expansion for $\sin (x)$ is:

$$
\sin (x) = x + \frac{x^3}{3!} + \frac{x^5}{5!} + \frac{x^7}{7!} + \ldots
$$

The successive terms of this expansion are calculated until they become too small to alter the fixed-point sum.

**Details.**

Before calling, the argument (radians $\times 4096$) is loaded into the B-register. On exit, the A-register contains the calculated cosine value $\times 4096$. See chapter 3.1 for details of calling instructions.

**Storage Requirements.**

- $X$ segment: 41 words.
- $Y$ segment: 7 (LOCAL) words.
Execution time.

\[ T_{exec} = 100 - 150 \mu s, \text{ depending on argument.} \]

Programming Example.

TRYCOSF: PROGRAM ; TO ILLUSTRATE COSF SUBROUTINE
PRN: VAL: COSVAL:.....GLOBAL

.....
LDW X LOCL ; SET OFFSET FOR LOCAL VARIABLES
.....
LDB Y VAL ; LOAD ARGUMENT INTO B
ENT X COS ; CALCULATE COSINE
STA Y COSAL ; RESULT LEFT IN A
.....
LOCL: WBASE
COS: ADX COSF ; EXTERNAL ADDRESS OF COSF SUBROUTINE
ENDMODULE

PROGEND
: COSF_PROCESS - COMPUTES COSINE BY MACLAURIN SERIES
A: = 4096 * COS(B: / 4096)
I. S. SEVERHRIET, DEPT. OF ELECTRONIC AND ELECTRICAL ENGINEERING
LOUGHBOURGH UNIVERSITY OF TECHNOLOGY.
MARCH 1972, EDITION 3.

COSF:LIBRARY
COSF:MODULE
LDA YW 2
TSTL A<0
CPY X YZ1: INVERT_ARG. IF < 0
SBA X YZ2
TSTL A<0
ADA X YZ2; SHIFT_ARG. TO 0 - 2*PI
SBA X YZ3
TSTL A<0
CPY X YZ1; FOLD_ARG. ABOUT PI
SBA X YZ4; SUBTRACT PI/2
STA YW 6; WORK OUT SINE OF (W+6)
STA YW 4; CURRENT ITERATION INCREMENT
STA YW 1: TOTAL
LDM L 1

YZ6:ADM L 1
STM YW 5: NEXT_DENOMINATOR_DIVIDEND
CPYL RA RB AND
SBA YW 4
MLD YW 6
SFTL D A L 3; SHIFT_PRODUCT INTO B:
CPYL RA B2 EQV NOT1
LDB L 0
DIV YW 5

ADM L 1
STM YW 5
LDB L 0
MLD YW 6
SFTL D A L 3
CPYL RA B2 EQV NOT1
LDB L 0
DIV YW 5; NEW_INCREMENT_IN A:

TSTL A=0?; INCREMENT-0?
EXT; IF SO, EXIT
STA YW 4; NO - LOOP
ADA YW 1
STA YW 1
JMP YZ6

Listing of COSF process.
Listing of COSF (continued).
3.4. FFT1

Purpose: Calculations of D.F.T. by FFT method.

Description.

FFT1 subroutine calculates the direct or inverse D.F.T. of complex fixed-point number sequences by an FFT algorithm (1). The algorithm uses a radix -2, 'decimation in time' method. The following transforms are performed:

Direct: \[ X(k) = \frac{1}{N} \sum_{i=0}^{N-1} x(i) \exp(-j2\pi ik/N), \quad 0 \leq k < N \]

Inverse: \[ x(i) = \sum_{k=0}^{N-1} X(k) \exp(j2\pi ik/n), \quad 0 \leq i < N \]

Division by N is performed during the direct, rather than the inverse, transform to ease problems of arithmetic overflow.

The N complex data values must be stored as follows:

<table>
<thead>
<tr>
<th>Real parts</th>
<th>Imaginary parts</th>
</tr>
</thead>
<tbody>
<tr>
<td>0, 1, 2, 3, ..... (N-1)</td>
<td>0, 1, 2, 3, ..... (N-1)</td>
</tr>
</tbody>
</table>

The real values are located consecutively. The corresponding imaginary values are located similarly, immediately above in store. The separation of real and imaginary parts in this way simplifies the handling of purely real data.

When calculating the DFT of a set of N complex numbers, FFT1 requires a table of \( \frac{\pi}{N} \) sine values as follows:

\[ \text{Sin}(i) = 32767 \times \sin\left(2\pi i/N\right), \quad 0 \leq i < \frac{\pi}{N} \]

This table of sine values may be conveniently produced by ATAB subroutine (see Chapter 3.2). It is not altered by FFT1.

During execution, FFT1 first performs bit-reversed 'shuffling' of the complex data. The FFT calculations are then performed. The complex
exponential coefficients required are obtained from the sine table described above. The orders of complex multiplications, additions and subtractions are chosen for greatest accuracy with fixed-point arithmetic.

The length, N, of the complex number sequence to be transformed must be an integer exponent of 2.

Details.

Before calling, a 4-word parameter area in the Y-segment must be filled as follows:

Word 0: N, number of complex data values.

1: 0 for direct transform, > 0 for inverse transform.

2: Indirection constant (for M-modification) of starting address of sine table (see above).

3: Indirection constant (for M-modification) of starting address of data.

The offset of this parameter area must be loaded into the S-register before calling FFT1 subroutine.

Storage Requirements.

X segment : 158 words

Y segment : 20 (LOCAL) words

Execution Time.

\[ T_{exec} = 50 N \log_2(N) \text{ us approximately.} \]

Arithmetic Errors.

After direct, and inverse transform of 32 data values, RMS error between initial and final values is about 5.

Overflow.

Cannot occur on direct transform if all data values (real and imaginary parts) are within the range ± 16384.

Programming Example.

TRYFFT1: PROGRAM ; TO ILLUSTRATE USE OF FFT1

; SUBROUTINE
PRN: NUM: DIR: INDAT: GLOBAL
SINTAB + 191: DATR + 255: DATI +255: .......: GLOBAL
USEY
    RELSTART NUM ; SET UP PARAMETER AREA (TYPICAL).
    256 ; 256 DATA VALUES, E.G.
    0 ; DIRECT TRANSFORM
    IND YM SINTAB ; IND CONSTANTS FOR SIN TABLE
    IND YM DATR ; AND DATA
    USEX
    ......
    LDW X LOCL ; SET OFFSET FOR LOCAL VARIABLES
    ......
    LDA Y NUM ; CALL ATAB SUBROUTINE
    LDM Y INDAT ; TO GENERATE SINE TABLE
    ENT X TAB ; FOR FFT1
    ......
    ......
    LDB L NUM ; LOAD B WITH OFFSET OF PARAMETER AREA
    ENT X FFT ; CALL FFT1
    ......
    ......
    ......
    LOCL: WBASE
    TAB: ADX ATAB ; EXTERNAL ADDRESSES FOR ATAB
    FFT: ADX FFT1 ; AND FFT1 SUBROUTINE
    ......

ENDMODULE

PROGEND

Reference (1):
G-AE Sub-committee on Measurement Concepts, "What is the Fast Fourier
LISTING OF FFT1 PROCESS

; FFT1_PROCESS - COMPUTES DFT BY FFT ALGORITHM
; B := OFFSET OF Y-PARAMETER AREA (4 WORDS)
; PARAMETERS: NO_POINTS, DIR, IND_SIN_TABLE, IND_DATA(REAL)
; DIR = 0 (DIRECT TRANSFORM); > 0 (INVERSE TRANSFORM)
; J. S. SEVERWRIGHT, DEPT. ELECTRONIC AND ELECTRICAL ENGINEERING,
; LOUGHBOROUGH UNIVERSITY OF TECHNOLOGY.
; MARCH 1972, EDITION 3.

FFT1:LIBRARY
FFT1:MODULE

LDB YW 2; TRANSFER ARGUMENTS
LDA YB 0
STA YW 16; NUM
LDA YB 1
STA YW 17; DIR
LDA YB 2
STA YW 18; IND_SIN
LDA YB 3
STA YW 19; IND_DATA(REAL)

LDA YW 16
SETL J_L_R_15
SBB L 1
SIB YW 7 7;_LOG2(NUM)

LDA YW 16
ADA YW 19
STA YW 4; _IND_DATA(IMAG)

LDA YW 16; START SHUFFLING
SBA L 2
STA YW 12; UPPER_COUNT_VALUE
LDA L 0
STA YW 8; ZEROISE_COUNT
STA YW 9

FF0: LDA YW 8
SBA YW 12
TSTL A=0, INCA; TEST COUNT
JMP FF9

ADA YW 12
STA YW 8

LDB L 0
STB YW 10
LDM YW 7

FF6: SFTL T L R 1; DO BIT-REVERSAL OF A:
EXC YW 9
ADJ X FF7; (SHIFT CONSTANT)
STA YW 11
SBM X FF7
SFT YW 11; DYNAMIC LEFT SHIFT

Listing of FFT1 process.
Listing of FFT1 (continued).
Listing of FFT1 (continued).
LDA YW 11
ADA YW 10; INCREMENT CEXP OFFSET
SBA YW 6
TSTL A<0; <NUM/2?
ADA YW 6
STA YW 11; NEXT CEXP OFFSET

LDA YW 9
ADA L 1
STA YW 8; DATA INCREMENTS SINCE LAST SKIP
SBA YW 7; DATA OFFSET INCREMENT
TSTL A<0
JMP FF5
STA YW 9
LDB YW 8
ADE YW 7
STB YW 8; NEW LOWER DATA OFFSET

FF5:LDA YW 8
ADA L 1
STA YW 8; INCREMENT DATA OFFSET
SBA YW 16; NUM
TSTL A<0; END OF THIS LOOP?
JMP FF3

LDA YW 10; YES-HALVE CEXP OFFSET INCREMENTS
SFTL S A R 1
TSTL A=0
EXT; EXIT IF ZERO
STA YW 10

LDA YW 7
SFTL S A L 1; DOUBLE DATA OFFSET INCREMENTS
STA YW 7
LDA L 0
JMP FF2

FF4:CPYC RA SUB A1
ENDMODULE

Listing of FFT1 (continued).
3.5. HAMM

Purpose: Multiplication of number sequence by Hamming window.

Description.

HAMM subroutine weights a real fixed-point number sequence $s(n)$, where $n = 0, 1, \ldots, N-1$, with a Hamming window $h(n)$:

$$h(n) = 0.54 - 0.46 \cos \left( \frac{2 \pi n}{N} \right),$$

The weighted number sequence $S_w(n)$, where $n = 0, 1, \ldots, N-1$ is formed:

$$S_w(n) = s(n) \cdot h(n)$$

The values $S_w(n)$ are re-stored over the original values of $s(n)$. HAMM requires a table of $3N/4$ sine values as follows:

$$\sin(i) = 32767 \cdot \sin \left( \frac{2\pi i}{N} \right)$$

$\quad i = 0, 1, \ldots, 3N/4$

HAMM subroutine is intended for spectrum analysis applications in conjunction with FFT1 (see Chapter 3.4). When used in this way, the table of sine values required by HAMM is identical to that required by FFT1, and both subroutines may conveniently use the same table. To retain accuracy, double length fixed-point multiplication is used in the weighing process. The number sequences to be processed may contain any 16-bit values.

Details.

Before calling, the A, B and M registers must be loaded as follows:

A - register loaded with $N$, the length of the number sequence.

B - register loaded with the M-modifiable indirection constant of the sine table (see above).

M - register loaded with the M-modifiable indirection constant for the start of the number sequence. The original number sequence is overwritten with the weighted values.

See Chapter 3.1 for details of calling instructions.

N.B. HAMM operates only on a real number sequence. If it is required to weight a complex number sequence, HAMM must be called twice to separately weight the real and imaginary parts.
Storage Requirements.

X - 35 words.
Y - 9 (LOCAL) words.

Execution Time:

T_{exec} = 30 x N s approximately.

Programming Examples.

TESTHamm: PROGRAM ; ILLUSTRATION OF USE OF
 ; HAMM SUBROUTINE

USEY
RELSTART NUM
256 ; NUM SEQUENCE IS 256 LONG HERE EG.
IND YM SIN ; IND CONST FOR SIN TABLE
IND YM DAT ; IND CONST FOR DATA
USEX
......
......
LDW X LOCL ; SET OFFSET FOR LOCAL VARIABLES
......
......
LDA Y NUM
LDM Y INDSIN ; CALL ATAB SUBROUTINE TO SET UP
ENT X TAB ; SINE TABLE
......
......
LDA Y NUM ; CALL HAMM SUBROUTINE TO
LDB Y INDSIN ; WEIGHT ARRAY 'DATA' USING
LDM Y INDDAT ; SINE TABLE IN ARRAY 'SIN'
ENT X HAM
......
.....
LOCL: WBASE
TAB: ADX ATAB ; EXTERNAL ADDRESSES OF ATAB
HAM: ADX HAMM ; AND HAMM SUBROUTINES
ENDMODULE
PROGEND
HAMM_PROCESS - APPLIES HAMMING WINDOW TO ARRAY
\[ A := \text{ARRAY LENGTH} \]
\[ B := \text{M-IND CONST. OF SINE TABLE (FOR FFT PROCESS)} \]
\[ M := \text{N-IND CONST. OF DATA} \]
\[ J.S.\text{ SEVERAIGHT, DEPT. OF ELECTRONIC AND ELECTRICAL ENGINEERING} \]
\[ \text{LOUGHBOROUGH UNIVERSITY OF TECHNOLOGY} \]
\[ \text{MARCH 1972, EDITION 2} \]

HAMM_LIBRARY

HAMM_MODULE

LDA YW 1
SET S_A_R 1
STA YW 5; (NUM/2)
SET S_A_R 1
ADA YW 2
ADA YW 5
SBA L 1
STA YW 4; WORKING IND. CONST. FOR COS PART OF SINE TABLE

LDA YW 3
ADA YW 5
SBA L 1
STA YW 6; WORKING IND. CONST. FOR DATA

LDM L 0
SBN YW 5; SET UP COUNT
HA2:TSTL M=6, INCM
EXT

LDB L 0
LDA YW 1; GET COS VALUE
MLD X HA4
ADB X HA4+1
STB YW 7; WINDOW VALUE

LDA YW 6
MLD YW 7
STB YW 6; WEIGHT DATA (FROM START)

SIM YW 8
CPY X HA4+2
ADM L 1
LDA YW 6
MLD YW 7
STB YW 6; WEIGHT DATA (FROM END)

LDM YW 8
JMP HA2

HA4: -15074; -0.46
17689; 0.54
CPY_RN_M1_SUB

ENDMODULE

Listing of HAMM process.
3.6. **LOG**

**Purpose:** Calculation of \( \log_2 (X) \) by linear approximations.

**Description.**

LOG subroutine calculates the logarithm to base 2 of an integer \( X \) by linear approximation. The calculated value is scaled by \( 1024 \) to permit accurate representation in fixed-point form. The integer value \( X \) is considered in the following form:

\[
X = 2^n \cdot x_1
\]

where \( n \) is an integer, and \( x_1 \) lies between 1 and 2. The logarithm of the first part, \( 2^n \), is merely equal to \( n \). It is found simply by locating the position of the highest set bit in \( X \). The logarithm of the second part, \( x_1 \), is approximated as follows: As \( x_1 \) varies from 1 to 2, its logarithm varies from 0 to 1. It is calculated by a linear approximation:

\[
\log_2 (x_1) \approx x_1 - 1
\]

The logarithm of \( X \) is found by adding the logarithms of the first and second parts:

\[
\log_2 (X) \approx n + x_1 - 1
\]

The necessary calculations may be performed very efficiently on the Modular-1 processor, by use of the 'shift' functions.

The logarithms of zero, and negative numbers are returned as zero.

The subroutine returns exact values for the logarithm when \( X \) is an integer exponent of 2, i.e. \( 2^n \) where \( n = 0, 1, 2, \ldots \). Errors are great when \( X \) is small, and lies between these values. The largest error, when calculating the logarithm of 3, is less than 6%. \( X \) can have any 16 bit value.

This subroutine has been found very suitable for the on-line calculation of \( \log \) (power spectra), and cepstrum functions.

**Details.**

Before calling, the B-register must be loaded with integer \( X \). The calculated value of its logarithm to base 2, multiplied by \( 1024 \), is returned in the A-register.
Storage Requirements.

X segment: 18 words.

Y segment: 4 (LOCAL) words.

Execution Time.

$T_{exec} = 25 \mu s$ approximately
LOG_PROCESS_-_COMPUTES LOG2(B.) * 1024

RESULT LEFT IN A: ON EXIT

J. J. SEVERNWRIGHT, DEPT. OF ELECTRONIC AND ELECTRICAL ENGINEERING
LOUGHBOROUGH UNIVERSITY OF TECHNOLOGY.
MARCH 1972, EDITION 3.

LOG_LIBRARY

LOG_MODULE

LDA YW 2
JSFL A20; ARGUMENT +YE?
JMP L02
LDA L 0; NO - EXIT
STA YW 1
EXT

L02: SFTL J A L 15
STB YW 1
LDB L 14
SBB YW 1
STB YH 1; OFFSET OF UPPER BIT

SFTL S L L 2; REMOVE UPPER BIT
SFTL S L R 6; SHIFT REST DOWN
EXC YH 1
SFTL S A L 10; SHIFT (UPPER BIT POSN) UP
ADA YW 1; ADD
STA YW 1
EXT

ENDMODULE

Listing of LOG process.
3.7. MOD.

Purpose: Calculation of $(x^2 + y^2)^{1/2}$ by polynomial approximation.

Description.

In many kinds of signal processing, it is necessary to calculate the moduli of complex numbers as follows:

$$\text{Mod}(x) = \left(\text{Re}^2(x) + \text{Im}^2(x)\right)^{1/2}$$

The direct calculation of this function may not be performed conveniently in fixed-point arithmetic. When 16-bit data values are squared, 31-bit values result which cannot be manipulated by processor functions on the Modular-1 computer. All arithmetic operations have to be performed by software, at least an order of magnitude slower than the corresponding 16-bit processor operations. The square-root operation, in particular, becomes very slow when operating with 31-bit data values. For these reasons, an alternative method was developed for more efficient calculation of the modulus function in fixed-point arithmetic. The method, used by MOD subroutine is described below.

For purposes of notation, the modulus function will be described:

$$M = (x^2 + y^2)^{1/2}$$

Initially the numbers $X$ and $Y$ are checked separately and inverted if negative. The larger of the numbers $X$ and $Y$ is then found. It will be assumed that the number $X$ is larger. $M$ may be re-expressed as follows:

$$M = X \left(1 + \left(\frac{Y}{X}\right)^2\right)^{1/2}$$

$$= X A$$

where $A = (1 + k^2)^{1/2}$ and $k = Y/X$

If $Y$ is smaller than or equal to $X$, $k$ must lie between 0 and 1. $A$ lies between 1 and $\sqrt{2}$, and varies smoothly with $k$ within this range. $A$ can therefore be calculated conveniently from $k$ by means of a polynomial approximation. This may be performed entirely in 16-bit fixed-point arithmetic, and involves far less computational effort than the square root operation required for direct calculations of $M$. When the value of $A$ has been calculated in this way, it is multiplied by $X$ to yield the required value $M$. 
The coefficients of the polynomial approximation for $A(k)$ were calculated for minimum mean-square error over the range $0 \leq k \leq 1$. The following third order polynomial was found to give adequate accuracy.

$$A(k) = 1.0003 + 0.55205k - 0.0076k^2 - 0.13029k^3$$

Subroutine MOD calculates the modulus $M$ of numbers $X$ and $Y$ by the following steps:

1. Invert $X$ or $Y$ if negative, and swap over if necessary to make $X > Y$.
2. Calculate $k = Y/X$.
3. Calculate $A$ from third order polynomial approximation in $k$.
4. Calculate modulus $M$ by multiplying $A$ by $X$.

The arguments $X$ and $Y$ may have any values for which the modulus does not exceed 32767.

Details.

Before calling MOD subroutine, the arguments $X$ and $Y$ must be loaded into the A and B registers. The calculated value of the modulus $M$ is returned in the A register. See Chapter 3.1 for details of calling instructions.

Storage Requirements.

- $X$ segment: 39 words.
- $Y$ segment: 5 (LOCAL) words.

Execution Time.

$$T_{exec} = 50 \mu s$$ approximately

Arithmetic Error.

Maximum error in calculated value of $M$ is $\pm 2$.

Programming Example.

```plaintext
TESTMOD: PROGRAM  ; ILLUSTRATES USE OF MOD SUBROUTINE

PRN: VAL: REAL: IMAG: .......: GLOBAL

.....

.....

LDW X LOCL  ; SET OFFSET FOR LOCAL VARIABLES
```
LDA Y REAL ; LOAD ARGUMENTS INTO REGISTERS
LDB Y IMAG
ENT X MODS ; CALCULATE MODULUS
STA Y VAL ; RESULT LEFT IN A REGISTER

LOCL: WBASE
MODS: AIX MOD ; EXTERNAL ADDRESS OF MOD SUBROUTINE

NEDMODULE
PROGEND
MODPROCESS: COMPUTES (A^2 + B^2)^0.5 BY POLYNOMIAL APPROXIMATION.
LOAD ARGUMENTS IN A: AND B: RESULT LEFT IN A:
J. S. SEVERAIGHT, DEPT. OF ELECTRONIC AND ELECTRICAL ENGINEERING.
LOUGHBOURGH UNIVERSITY OF TECHNOLOGY.
MARCH 1972, EDITION 3

MOD:LIBRARY
MOD:MODULE

LDA YW 1
TSIL A<0
CPY X MO2; INVERT ARG. 1 IF -YE
STA YW 1

LDA YW 2
TSTL A<0
CPY X .MO2; INVERT ARG. 2 IF -YE
STA YW 2

SBA YW 1
TSIL A<0; ARG. 1 > ARG. 2?
JMP M03
LDA YW 2; NO, SO_SWAP
EXC YW 1
STA YW 2

M03: LDA YW 2
DIV YW 1
TSIX MO5; OVERFLO~/2?
LDA X .MO6
STA YW 4; ARG. 2/ARG. 1

MLD X .MO4
ADB X .MO4+1
CPYL RA.B2 .EQV .NOT1

MLD YW 4
SBB L 127; 0.0076X14
CPYL RA.B2 .EQV .NOT1

MLD .YW 4
ADB X .MO4+2
CPYL RA.B2 .EQV .NOT1

LDB L 0
MLD YW 1; MULTIPLY BY ARG. 1
SFIL D A L 1
STB YW 1
EXT

Listing of MOD process.
Listing of MOD (continued).
This chapter describes subroutines for displaying straight lines on two types of analogue peripheral, namely:

a) Tektronix 611 Video Display Unit.

b) Bryans 24000 A4 X-Y Plotter.

The following subroutines are described:

STR1, STR2 - for video display unit.

PLT1, PLT2, PLT3 - for X-Y plotter.

These may be used to draw straight lines between specified coordinates on their respective peripherals. They form a convenient basis for more comprehensive display subroutines. Both sets of subroutines are written in assembly language for the CTL MAS1 assembler program, and form complete modules. They are intended to be called from assembly language programs. Linking subroutines are available which allow them to be called from Modular-1 FORTRAN programs. These subroutines are perhaps most useful with FORTRAN programs - these may be rapidly written to investigate different processing schemes, and all the waveforms involved may be displayed for detailed examination.

The following comments apply to both sets of display subroutines:

1) The 1.024 D/A Converter is used for all output to display peripherals.

2) The calling program must declare ADX constants for all subroutines required.

3) The calling program must load the W-register with a suitable offset for LOCAL variables used in the subroutines.

4) Arguments, where required, consist of an X and Y coordinate which must be loaded into the A and B registers respectively before calling. The display axes are spanned by a coordinate range of ± 1023; values outside this range are limited to it within the subroutines.

5) The subroutines are intered by ENT program instructions.
6) On exit, program execution always continues from the following instruction. Pre-entry register values are always restored on exit.

7) The subroutines do not access any Z-segment storage locations.

8) The subroutines include references to CHACCESS and TERI executive instructions.

N.B.: The Y : X aspect ratio of both the display peripherals mentioned previously is about 4 : 3. The displacement corresponding to a given Y-coordinate increment is therefore about one third greater than that resulting from a similar X-coordinate increment.
Subroutines STR1, STR2.

- to draw straight lines on Tektronix 611 V.D.U. (Edition 2).

Programming points:

STR1 is called to set the starting coordinates of a line, these being transferred from the calling program as described in the introduction. No output is produced.

STR2 is then called to specify the coordinates to which a line is to be drawn on the V.D.U. It is drawn between the coordinates specified in the last call to either STR1 or STR2, and those specified in the current call. STR2 may be called as many times as required to generate a series of connected lines.

The line is formed from a large number of closely spaced dots, the writing beam being turned on for approximately 12 μS at each dot. When writing, the V.D.U. switches to 'VIEW' mode, in which it then remains for approximately 90 seconds.

Storage requirements:

\[
\begin{align*}
X & \quad 101 \text{ words} \\
Y & \quad 12 \text{ (10 LOCAL) words.}
\end{align*}
\]

External Connections:

External X, Y and Z signals are obtained directly from D/A converter channels 3, 4 and 5 respectively, and vary within the range \( \pm 5.115 \text{ volts} \). No 'clocking' signals are required, and no other control signals are output.
STR1: PROCESS - TO SET FIRST PLOTTING COORDINATE
STR2: PROCESS - DRAWS LINE TO NEXT COORDINATE
A:, B: HOLD X, Y CO-ORDS BEFORE CALLING, +/−1023.
J.S.S. APRIL 1972, EDN.2

STR1:STR2:LIBRARY
STR1:MODULE
XLAST:YLAST:GLOBAL

STR1:PROCESS; FOR SETTING UP

LDA YW XIN
SRE X ST2
LDB X GLOB
STA YB 0; STORE X-VALUE
LDA YW YIN
SRE X ST2
LDB X GLOB
STA YB 1; STORE Y-VALUE

EXT

STR2:CUE; FOR DRAWING LINE

LDA YW XIN
SRE X ST2
STA YW XLIM; GET X-VALUE
LDB X GLOB
SBA YB 0
STA YW XDIFF; GET XDIFF
LDA YW XLIM
STA YB 0; STORE NEW XLAST

LDA YW YIN
SRE X ST2
STA YW YLIM; GET Y-VALUE
LDB X GLOB
SBA YB 1
STA YW YDIFF; GET YDIFF
LDA YW YLIM
STA YB 1; STORE NEW YLAST

LDA YW XDIFF
TSTL A<0
CPY X AINV
STA YW NINC; ABS(XDIFF)
LDA YW YDIFF
TSTL A<0
CPY X AINV; ABS(YDIFF)
SBA YW NINC
TSTL A<0
JMP NEXT
ADA YW NINC
STA YW NINC; NO. INCREMENTS

Listing of STR1 and STR2 processes.
Listing of STR1 and STR2 (continued).
GLOB: ADG XLAST
AINV: CPYC RA SUB A1
BOVF: TSTC BOVR
AMAX: 32767

ST2A: PROCESS; CHECKS COORDINATES
STB YW LINK
SBA X ST2+2
TSTL A>0; TOO POSITIVE?
LDA L 0
ADA X ST2+1
TSTL A<0; TOO NEGATIVE?
LDA L 0
SBA X ST2+2
LDP YW LINK; EXIT
ST2: ADS ST2A
2046
1023

END
ENDMODULE

**

Listing of STR1 and STR2 (continued).
Subroutines PLT1, PLT2, PLT3.
- to draw straight lines on Dryan's 24000 X-Y Plotter.

Programming points:

PLT1 is called to set the starting coordinates of a line, these being transferred from the calling program as described in the introduction. The plotter pen is positioned over this coordinate for about 0.5 seconds.

PLT2 is then called to specify the coordinates to which a line is to be drawn on the plotter. It is drawn between the coordinates specified in the previous call to either PLT1 or PLT2, and those specified in the current call. PLT2 may be called as many times as required to draw a series of connected lines.

At each call, PLT2 lowers the plotter pen (if raised), and moves it at roughly constant speed to the specified coordinate. PLT2 then exits, leaving the pen lowered to obtain a 'clean' plot. The next call to the plotter routines must occur within about 20 ms, however, to prevent drift in the D/A converter output from causing 'jagged' lines. A further call may be made to PLT2, or PLT3 may be called to raise the pen.

PLT3 requires no arguments. It raises the plotter pen, and waits approximately 50 ms to allow it to lift clear of the paper. This prevents short 'tails' being drawn on lines if a program call to PLT1 follows immediately, causing the pen to move rapidly towards a new coordinate.

Storage requirements: X - 131 words
Y - 15 (13 LOCAL) words.

External connections:

X, Y and Z output signals are obtained from D/A converter channels 0, 1 and 2 respectively. The X and Y outputs are connected to the "Y" and "X" inputs on the plotter respectively, to obtain a Y : X display aspect ratio of about 4 : 3; i.e. similar to a printed page. A4 - size paper is very suitable.

The Z (pen shift) output is connected via the pen-lift amplifier to the remote pen control inputs on the plotter.

The D/A converter must be clocked at a frequency of 300-400 Hz.

The plotter must be calibrated to cover the required area of display with X, Y input voltages of ± 5.115 volts.
;PLT1 - TO SET UP PLOTTER Routines.
;PLT2 - TO DRAW LINE.
;A, B: HOLD X, Y CO-ORDS (+-1023)
;PLT3 - TO RAISE PEN AFTER LINE.
;J.S.S. April 1972, EDN.3

Listing of PLT1, PLT2 and PLT3 processes.
Listing of PLT1, PLT2 and PLT3 (continued).
PLT3:CUE; TO LIFT PEN
LDB X XLAST
LDA YB 0
STA YW XDP; GET XOP
LDA YB 1
STA YW YOP; GET YOP
LDA L 0
STA YW ZOP; RAISE PEN
LDM L 0
SBM L 50
LOOP3:TSTL M=0,INCM
EXT
SRE X PL2; OUTPUT
JMP LOOP3
XLAST:ADG LAST

PL2A:PROCESS; TO OUTPUT X,Y,Z VOLTAGES
STB YW LINK
LDA L 8
CHACCESS
TERM
LDA YW XOP
LDB L 0
STA ZB 24; CH.0
SBP L 3
LDA L 0
SBA YW YOP; INVERT Y-OUTPUT SO PLOTTER GOES IN CORRECT DIRECTION
LDB L 0
STA ZB 9; CH.1
SBP L 3
LDA X CONST
LDB YW ZOP
TSTL B=0
SBA X CONST+1
LDB L 0
STA ZB 10; CH.2
SBP L 3
LDA L 0
CHACCESS
TERM
LDP YW LINK
PL2:ADS PL2A

Listing of PLT1, PLT2 and PLT3 (continued).
PL3A: PROCESS; TO CHECK AND LIMIT CO-ORDS
STB YW LINK
SBA X CONST
TSTL A>0
LDA L 0
ADA X CONST+1
TSTL A<0
LDA L 0
SBA X CONST
LDP YW LINK
PL3: ADS PL3A

CONST: 1023
2046

END
ENDMODULE

**

Listing of PLT1, PLT2 and PLT3 (continued).
Chapter 5. Collection and Reproduction of Analogue Signals for Off-line Processing

The University Computer Centre housed a large ICL 1904A digital computer. This facility was very attractive for simulation programming use. The computer incorporated fast, hardware floating-point arithmetic, far more core storage than the Modular-1 computer, as well as magnetic tape and disc bulk storage. Also, good compiling systems were available for high-level languages, such as FORTRAN, and incorporated excellent diagnostics. However, this computer had no peripherals for handling analogue signals and in any case was oriented towards off-line batch processing operation. To make use of these facilities, therefore, it was necessary to collect and reproduce analogue signals using the analogue peripherals in the Modular-1 computer.

Processing was performed in three stages:

1) Analogue signal sampled and stored on Modular-1 computer.
2) Data processed off-line by FORTRAN simulation program on 1904A computer.
3) Analogue signal reproduced from sampled data on Modular-1 computer.

For data transfer to the 1904A computer, the Modular-1 had only paper tape peripherals (1 inch, 8 hole), and this medium was employed. It has several advantages - it is extremely cheap, and read/write operations may be programmed simply. Unlike magnetic tape, paper tape can be spliced and requires no special storage conditions. A disadvantage is that large amounts of tape are required. Using an efficient tape code which was developed for this purpose, 100,000 sample values (about 10 seconds of speech) occupied almost two eight-inch reels of paper tape. The resulting physical inconvenience limits the use of this processing method to short sections of speech, of up to about 10 s in length. Off-line processing was accordingly most useful for initial trials of new processing schemes. A 10 s segment of processed speech was adequate for initial performance estimates.
In the following sections, the paper tape code used for the data transfer operations is described, together with the methods employed on the Modular-1 computer to perform the signal collection and reproduction operations. Some of the FORTRAN simulation programs are described in the following chapter. Many establishments house small computers equipped with analogue peripherals, as well as larger, more powerful computers for off-line operation. The off-line processing techniques described here may have application in such establishments.

5.2. Paper Tape Code.

To minimise the bulk of paper tape involved, the paper tape code had to be efficient. At the same time, its form had to be such that it could be processed easily by a FORTRAN program, in a large computer with peripherals of limited flexibility.

The analogue peripherals on the Modular-1 operated with 11-bit signed integer numbers - in the range $-1024$ to $1023$. These could be encoded onto tape as decimal integers. Each data value would require at least four tape characters on average - an average of three decimal digits and a sign or separator character. Each tape character would then convey $11/4$ bits - less than three bits. This indicates very inefficient usage of 8-hole paper tape.

The choice of possible codes is limited by hardware on the 1904A computer, as well as by FORTRAN software. In common with many large computers, the various peripherals decoded character representations from their respective media into a common internal machine form. The paper tape reader and punch could not handle all of the 256 patterns possible in 8-hole tape. They would only recognise the set of sixty-four 'printable' character codes used by the FORTRAN language, together with a few control codes. This set includes the digits 0–9, the alphabet and punctuation characters, and the paper tape code was based upon it. By use of a 64-member set, each paper tape character can convey $\log_2 (64) = 6$ bits. This is more than twice that conveyed by characters in a decimal representation, and the 11-bit numbers handled by the analogue peripherals on the Modular-1 computer could be represented in only two paper tape characters.
The eight-hole paper tape patterns used by most English computers follow the ISO-7 code. The American 8-hole ASCII code is similar, except in the eighth (parity) hole. In the ISO-7 code discussed here, this eighth hole is set in each character for even parity, and the binary values of the lower seven holes corresponding to the 64 printable characters run consecutively from 32 to 95. To be compatible with the 1904A computer, therefore, six-bit values (0-63) were represented as tape characters by adding 32 to obtain the decimal equivalent of the lower seven holes, and then setting the eighth hole for even parity.

Twelve-bit numbers - in the range -2048 to 2047 - were encoded into two tape characters. Negative data values were made positive by adding 4096 ($2^{12}$) to obtain a 2's complement representation. The upper six bits of the resulting twelve-bit pattern were encoded and output on tape first, followed by the lower six bits. This process was performed in reverse order to decode data values from pairs of tape characters. To illustrate these processes further, the FORTRAN and CTL Assembly language subroutines are now described for decoding this paper tape data.

5.2.1. FORTRAN Subroutine IOP1.

Subroutine IOP1 is written as a FORTRAN integer function, and is listed in Fig 5.1. Decoding is performed in two steps. Initially, a block of data values is read from paper tape into an integer array, IT, by a standard FORTRAN 'READ' command. A2 format is used so that each storage location holds the internal machine representations of two characters - i.e. a character pair which holds one data value. To use a 'READ' command in this way, the data has to be in blocks of known size, each terminated with an 'end of record' (line feed) character. It is convenient to convey about 1000 data values in each block, and to separate these with a few inches of blank tape for easy identification. Function IOP1 is then used to decode data values. The Ith value in the current block is obtained:

$$N = \text{IOP1}(\text{IT}(I))$$
FUNCTION IOP1(I1)
C DECODES VALUE OF I1 FROM 12-BIT TEXT FORM.
C TEXT CHARACTERS IN TOP TWO BYTES OF WORD I1.
C IOP1 HAS VALUE IN RANGE -2048 TO 2047.
C J.S.SEVERWRIGHT. NOV. 1971. EDN.2.
C SHIFT TOP TWO BYTES TO BOTTOM OF 'I'.
I=I1/4096
C IF TOP BIT WAS SET (I.E. NEGATIVE) CLEAR BITS 12-23.
IF (I.LT.0) I=I+4096
C EXTRACT UPPER AND LOWER DATA BYTES.
IUP=I/64
ILOW=I-IUP*64
C CONVERT INTERNAL MACHINE CHARACTER VALUES TO
C SIX-BIT DATA VALUES.
IUP=IOP1A(IUP)
ILOW=IOP1A(ILOW)
C EXTEND SIGN BIT IN 'IUP'.
IF (IUP.GT.31) IUP=IUP-64
C COMBINE UPPER AND LOWER SIX-BIT PATTERNS.
IOP1=IUP*64+ILOW
RETURN
END
C FUNCTION IOP1A(I)
C THIS FUNCTION CONVERTS AN INTERNAL MACHINE CHARACTER
C VALUE 'I' TO CORRESPONDING SIX-BIT DATA VALUE.
IF (I-32) 1,1,0
IOP1A=I
RETURN
1 IF (I-16) 0,2,2
IOP1A=I+16
RETURN
2 IOP1A=I-16
RETURN
END

Listing of IOP1 subroutine.
The subroutine is machine-dependent in that its form depends on how and where characters are stored in array IT by the READ command. In ICL 1900 computers, the internal machine values (six bits) of characters are stored from the top of storage locations. In A2 format, the first (most significant) character of a pair is stored in the topmost six-bit byte, and the second character is stored in the next. IOP1 extracts the separate character values from each 24-bit word by division. It is then necessary to obtain the six-bit data values corresponding to the characters from their 1900 internal machine forms. In fact, this is done very simply by function IOP1A, included with IOP1.

The listing of IOP1 in Fig 5.1. illustrates the simplicity of this approach with FORTRAN programming. It would be possible to write a version of IOP1 in FORTRAN which was entirely independent of characteristics of the computer hardware. Only one character value would be read into each storage location of array IT, and would be compared with a table of Hollerith constants of all 64 printable characters to obtain the corresponding six-bit data value. Such a subroutine would occupy more storage space, and would be far less efficient than the version described here.

Similar techniques are employed for data output.

5.2.2. Assembly language subroutine OATA2.

Subroutine DATA2 is written in assembly language for the Modular-1 computer. It is programmed similarly to subroutine DATA3, for reading decimal integers, which is described in Chapter 2. DATA2 controls the reading, as well as decoding, of individual character pairs. Initial blank tape characters are ignored by the instructions below label DT3.

When a non-zero tape character is found, it is decoded to the corresponding six-bit data value by DT2A subroutine. This checks parity, and that the decoded data value is valid - in the range 0-63. The resulting value is shifted left by six places, and its sign extended to the top of the sixteen-bit word. In assembly language, unlike FORTRAN, shifting operations permit these data manipulations to be performed very efficiently. The second character is similarly read and decoded, and added to the previous result to yield the required data value.
;DATA2 PROCESS - READS DATA VALUE (12-BIT). LEFT IN A: ON EXIT.
;IF_ERROR, EXITS TO (P+1) IF_OK, EXITS TO (P+2)
;J. S. MARCH 1972. EDN. 2
;CALLS PTREAD.

DATA2:LIBRARY
DATA2:MODULE
DT1:PROCESS
DT3:ADW L 8
ENT X PTRD
SBW L 8
TSTL A=0; IS_CHAR, BLANK?
JMP DT3

SRE X DT2; DECODE CHAR.
SFTL S L L 10
SFTL S A R 4
STA YH 7
ADW L 8
ENT X PTRD
SBW L 8
SRE X DT2; DECODE NEXT CHAR.
ADA YH 7
STA YH 1
LDA YH 0
ADA L 1; INCREMENT P-VALUE
STA YH 0
EXIT; EXIT TO (P+2)

DT2A:PROCESS; DECODES AND CHECKS CHARACTERS.
STA YH 4
STA YH 5

SFTL T L R 8
STB YH 6
CPYL RA B1 EQY NOT1
SFTL S L R 1
SFTL S L L 1
SBA YH 6
TSTL A<0, A>0; CHECK FOR PARITY ERROR.
EXIT

LDA YW 5
SFTL S L L 9
SFTL S L R 9
SBA L 32; DECODE.

Listing of DATA2 process.
Listing of DATA2 (continued).

TSTL A<0;  CHECK LOWER LIMIT.
EXT
SBA L 63
TSTL A>0;  CHECK UPPER LIMIT.
EXT
ADA L 63

LDP YW 4
DT2:ADS DT2A

PTRD:ADX PTREAD

END
ENDMODULE
As this subroutine reads, as well as decodes, data it must inform the calling program of data errors. It does this, like TAPR3, by skipping the following instruction in the calling program if no errors are encountered. The skipped instruction, executed only after an error, may be a jump instruction to another section of program. In particular, this feature is useful for detecting the ends of data blocks if an erroneous character is purposely included at the end of each one.

5.3. Data Collection and Reproduction.

The Modular-1 computer, with which the operations of signal collection and reproduction were performed, had no bulk storage. In general, when programs were loaded, the core storage available for data was limited to about 10,000 words - about 1 s of speech. It was therefore necessary to digitise, and reproduce speech signals as a succession of one-second segments. In order to process longer sections of continuous speech, an analogue tape recorder was employed which incorporated full remote control facilities (Sangamo 3560). Its operation was controlled via five channels of the D/A converter. In use, one tape channel was used to record or replay the required speech signal. An FM channel was used, to obtain best signal/noise ratio. Another tape channel was used to store marker pulses, at intervals of approximately 1 s. This was recorded on a direct channel, in bursts of high frequency sine wave of approximately 100 ms duration. Marker pulses of approximately 50 ms rise time were obtained by rectifying, and low-pass filtering this signal. This low-pass filtering eliminated the effects of impulsive noise and tape drop-outs from the marker pulses. An FM channel was not used as it would not operate while the tape transport was speeding up, or slowing down. The marker pulses had a peak amplitude of approximately 1 v, and were considered to occur by programs when their level increased through 0.5 v. The detection 'jitter' error was found to be less than 200 µ s.

Analogue signals were reproduced from sampled data in the following way. The tape was rewound to a point before the first marker pulse. The program then requested data, which was read from paper tape. This data was in blocks of 1000 words, so that a block could be re-read if an error occurred. Reading
continued until the data buffer could accept no further complete blocks. The operator could then instruct the computer to output a section of signal onto the tape recorder. The computer started the tape recorder. When a marker pulse was detected, the speech channel was switched to 'record' mode, and output of sampled data via a D/A converter channel was commenced. This continued until the next marker pulse was found. Data output was ceased, the recorder was switched out of 'record' mode and into 'reverse' for a short time. The tape transport was then stopped. When it came to rest, the transport was positioned just ahead of the last encountered marker pulse. The spacing ahead was controlled by the programmed duration of the previous 'reverse' motion so that when the transport was started to record the next block, it could attain full speed before reaching that marker pulse.

Any remaining data in the program buffer was moved down to its start, and further data was read from paper tape to refill it. The computer was then instructed to output a section of signal onto the tape recorder, as before. This time, however, as the transport was initially positioned in front of the second marker pulse, the signal section was recorded between the next pair of marker pulses. This process, of reading in more data and recording this on tape between successive pairs of marker pulses, was repeated as many times as required.

It was found necessary to check each section of tape as it was recorded. In case of error, a 'repeat' command was necessary to backspace the tape over one section, and to re-record it. The process of data collection, although the reverse of that described above, employed similar techniques.
Chapter 6. Simulation Programs

This chapter describes some of the simulation programs which were written for these project investigations. They are presented here with brief descriptions to illustrate the programming techniques employed, rather than the processes being simulated. Assembly-language programs for a Modular-1 computer, and FORTRAN programs for an ICL 1904A computer, are described separately.

6.2. Assembly language programs.

These were written for the Modular-1 computer. They operated on-line, using the A/D and D/A converters to input and output analogue signals. The following programs are described here, with listings:

- INT5 - rectangular interruption.
- INT9 - interruption and non-recursive reiteration synthesis.
- S2071 - interruption and pitch-synchronous reiteration, incorporating amplitude prediction.

These programs operate under the E1 executive program, and are run from the executive teletype console.
When this program is run DAT1 subroutine is entered initially. It is not used here to read in a number but because, as described in Chapter 2, it can skip the following program instruction on exit if a 'space' terminator character is typed instead of 'semi-colon.' If a 'space' is typed, the program prints messages requesting new values of the number of speech samples to be transmitted and interrupted NUT and NUI, during each processing period. Subroutines FORM1, DAT1 and NUM1 are used for this. If, however, 'semi-colon' were typed initially, execution jumps to label 'START', and processing commences immediately.

Following the 'START' label, the 'CHACCESS' executive subroutine is called to shift the program's range of Z-segment addresses to cover the A/D and D/A converters. The processing loop is entered at label 'GO'. The 'M' register counts the number of sample values processed in each processing period. Following label 'RET', a sample value is input to the 'A' register from the A/D converter operating in clocked mode. If the sample count in 'M' exceeds the value of NUT, the A-register is zeroised - i.e. interrupted. The value left in the A-register is then output via the D/A converter. The count in 'M' is then checked. When this is equal to (NUT+NUI), it is zeroised, and this process is repeated.
LISTING OF INT5 PROGRAM

INT5: PROGRAM
; IMPLEMENTS PERIODIC INTERRUPTION
; J. S. S. 1/2/72: EDN 2.

PRN: NUT: NUI: GLOBAL

USEZ ; STORE TEXT.

NO. SAMPLES TRANSMITTED: "
"
"; INTERRUPTED: "

USEX ; START OF PROGRAM.
LDA X LOCL ; SET W: FOR LOCAL WORKSPACE.
ENT X DAT1 ; TYPE IN NEW PARAMETERS?
JMP START ; NO - JUMP STRAIGHT TO PROCESSING LOOP.

LDA L 1 ; YES - READ IN NEW DATA VALUES.
ENT X FORM1 ; READ IN NUT.
ENT X DAT1

STA Y NUT
ENT X NUM1

LDA L 2
ENT X FORM1 ; READ IN NUI.
ENT X DAT1

STA Y NUI
ENT X NUM1

START: LDA L 8
7800 ; OPEN A/D, D/A CHANNELS.
TERM

GO: LDM L 0 ; M: HOLDS SAMPLE COUNT IN A PROCESSING PERIOD.
RET: LDB X AA
LDA 2B 31 ; INPUT A SAMPLE FROM A/D CONVERTER.
SBP L 3 ; (CHANNEL 7 CLOCKED)

SBM Y NUT
TSTL M=0, M>0, INCM ; INTERRUPT IT IF M: > NUT.
LDA L 0

LDP L 0
STA 2B 15 ; OUTPUT PROCESSED SAMPLE VIA D/A CONVERTER.
SBP L 3

Listing of INT5 program.
._SBM_Y_NUI
TSTL M=0 ; END OF PROCESSING PERIOD YET ?
JMP.GO ; YES.
ADM_Y_NUI
JMP RET ; NO.

LOC: HBASE
AA: 1024

FORM1: ADX_FORM1
DAT1: ADX_DAT1 ; EXTERNAL SUBROUTINE ADDRESS CONSTANTS.
NUM1: ADX_NUM1

ENDMODULE
PROGEND

Listing of INTS (continued).
The initial section of this program is similar to that of INT5. The values of NUT, the number of samples in each transmitted section, and KA, the number of reiteration to be performed may be typed in on the executive teletype or left unaltered. The ability to run a program rapidly without having to set its parameter values is useful when rapidly comparing the effects of several processing programs.

The processing loop is entered at label 'NE3'. The M-register is used to count samples during transmission and reiteration processes. During reiteration, the A/D converter is not used. At the start of a transmission period, therefore it is necessary to update the internal buffer of the A/D converter by the dummy command following label NE3.

The section of code following label NE2 simulates the transmission process. NUT sample values are input via the A/D converter, stored in a buffer and output via the D/A converter in clocked mode. When NUT samples have been transmitted and stored, reiteration is commenced. The code following label NE1 counts the number of reiterations actually performed in this processing period. Each reiteration is performed by the code following label NE5.
INI9: PROGRAM
; IMPLEMENTS INTERRUPTION AND NON-RECURSIVE REITERATION.
; J.S.S. 1/2/1972._EDN.1. (FOR E1 EXEC)

FRED+50: FILE

USEY
RESTART_INDP ; STORE _INDIRECTION_CONSTANT_FOR BUFFER
IND YH AREA

USEZ ; STORE TEXT.

"/

BLOCK LENGTH: "
";

": NO. REITERATIONS: "

USEX ; START OF PROGRAM.
LDW L 1 ; WORKSPACE AT BOTTOM END OF Y-SEG.

ENT X DAT1 ; TYPE IN NEW PARAMETERS ?
JMP START ; NO _ JUMP TO PROCESSING LOOP

LDA L 1 ; YES _ NEW PARAMETERS.
ENT X FORM1 ; READ IN NUT.
ENT X DAT1
0
STA Y NUT
ENT X NUM1

LDA L 2 ; READ IN KA.
ENT X FORM1
ENT X DAT1
0
STA Y KA
ENT X NUM1

START: LDA L 8
7880 ; OPEN A/D AND D/A CHANNELS.

TERM

NEG: LDM L 0 ; CLEAR HIELDS BUFFER _WORD_COUNT.
LDV X L3
LDA 28.15 ; CLEAR A/D CONVERTER BUFFER.
SBP L 3

Listing of INT9 program.
NE2:SEM Y NUT ;INPUT, STORE AND OUTPUT
TSL\text{\textunderscore}M=0 ;A\text{\textunderscore}TRANSMITTED\text{\textunderscore}SECTION
JMP NE1
STM Y NUT
LDB X L3
LDA ZB 15 ;INPUT\text{\textunderscore}UNCLOCKED, CH. 7
S\text{\textunderscore}BP L 3
STA YM INDP ;STORE IT
LDB L 0
STA ZB 31 ;OUTPUT\text{\textunderscore}CLOCKED, CH. 7
S\text{\textunderscore}BP L 3
STM L 1
JMP NE2.

NE1:LDB L 0 ;SET UP FOR REITERATIONS.
SBB Y KA
STB Y KT
NE4:LDB Y KT
TSL\text{\textunderscore}B=0, INCB ;REITERATIONS ALL COMPLETED ?
JMP NE3 ;YES.
STB Y KT
NE5:SEM Y NUT ;NO - DO A REITERATION.
TSL\text{\textunderscore}M=0
JMP NE4
STM Y NUT
LDA YM INDP
LDB L 0
STA ZB 31
S\text{\textunderscore}BP L 3
STM L 1
JMP NE5

L3:1024

END10
don

Listing of INT9 (continued).
This program simulates interruption, with reiteration synthesis incorporating adaptive prediction. The predictor maintains pitch synchronism, and continues the amplitude growth of the current transmitted section. The predictor weighting sequence is:

\[ H_p(z) = Bz^{-m} \]

The value of \( m \) is estimated by the cepstrum method described in Part 1 of this thesis. The value of \( B \) is calculated by a comparison of rectified signal sums:

\[
B = \frac{m-1}{m} \sum_{i=0}^{m-1} \left| S_i + m \right|
\]

\[
\sum_{i=0}^{m-1} S_i
\]

The calculations of the value of \( m \) and \( B \) presented problems for on-line simulation because an entire transmitted section was required before these calculations could commence. These consumed a large proportion of processor time, and would have imposed a delay before reiteration could commence. Because of this delay, processing could not be performed sample-by-sample. Instead, it was necessary to process the whole transmitted speech section as a block.

On-line processing of analogue data required regular, continuous data sampling even while blocks of data were being processed. This was achieved by outputting processed signals and inputting new signals under regular interrupt control, while the simulation program processed the current transmitted speech section. It was not then tied to real time, except inasmuch as it had to process data blocks at a faster rate than the input/output interrupt program.

The input/output interrupt program to operate the A/D and D/A converters was incorporated in the E1 executive program. A transfer operation was initiated by the command '7920', while the B-register contained the offset of a GLOBAL 2-word parameter area, as follows:

Word 0 = Length of buffer.

1 = Offset in Y-segment.
S2071: PROGRAM
; SIMULATES RECTANGULAR INTERRUPTION AND ADAPTIVE REITERATION
; PREDICTOR CONTINUES PITCH, AND GROWTH/DECAY OF CURRENT
; TRANSMITTED SECTION.

PRN+20: NUM: DIR: INDT: INDR: INDI: INCS+1; PA+1; PB+1; LINK+2; GLOBAL

USEY
RESTART_NUM
236 ; LOAD PARAMETER AREA FOR FFT1 PROCESS.
0
IND YM TRIG
IND YM_REAL

IND YM_REAL+255
IND YM_IMAG+255 ; WORKING CONSTANTS - PROGRAM WORKS WITH
256 ; 128-WORD TRANSMIT AND INTERRUPT SECTIONS.
ADG DATA1
256
ADG DATA2

USEX ; START OF PROGRAM.
LDW L 1 ; WORKSPACE AT BOTTOM OF Y-SEG.
LDA Y NUM
LDM Y INDT
ENT X ATAB ; GENERATE SINE TABLE FOR FFT1 PROCESS.

GO: LDB L PA ; CONTROL PROGRAM.
7920 ; INPUT/OUTPUT BUFFER "DATA1".
SBP L 3
LDA Y PA+1
STA Y LINK
SRE X SPEC ; NON_PROCESS CONTENTS OF "DATA2".
LDB L PB
7920 ; NON_SWAP_BUFFERS, AND CONTINUE.
SBP L 3
LDA Y PA+1
STA Y LINK
SRE X SPEC
JMP GO

SP1: PROCESS ; PROCESSING PROGRAM.
STB Y LINK+1
LDA Y LINK ; LINK HOLDS ADDRESS OF CURRENT BUFFER.
ADA L 255
STA Y LINK+2

Listing of S2071 program.
Listing of S207I (continued).

LDB L 0; MOVE DATA TO A COMPLEX ARRAY.
LDM L 0
SBM Y NUM
SP9:TSTL M=0, INCM
JMP SP8
LDA YI LINK+2
SFTL S A L 4; SCALE DATA INTO REAL PART.
STA YI INDS
STB YI INDS+1; ZEROISE IMAG. PART.
JMP SP9

SP8: LDM Y INDR; HAMMING WEIGHT REAL PART OF DATA.
LDB Y INDT
LDA Y NUM
ENT X HAMM
LDB L NUM
ENT X FFT1; FFT IT.

LDM L 0; CALCULATE LOG(MOD(SPECTRUM)).
SBM Y NUM
SP2:TSTL M=0, INCM
JMP SP3
LDA YI INDS
LDD YI INDS+1
ENT X MOD; GET MODULUS.
CPYL RB A1 EQV NOT1
ENT X LOG; GET LOG.
STA YI INDS; STORE IN REAL PART.
LDA L 0
STA YI INDS+1; ZEROISE IMAG. PART.
JMP SP2

SP3: LDB L NUM
ENT X FFT1; FFT IT.
LDM L 0
STM Y 19; FIND PITCH PERIOD.
SIM Y 18
LDM L 10
SP4: SBM L 128
TSTL M=0
JMP SP5
ADM L 128
LDA YI INDR; WORK OUT SCALING FACTOR.
SFTL S A L 1
SIA Y 20
CPYL RA M1 EQV NOT1
SFTL S A L 7
ADA X SPEC+3
MLD Y 20; SCALE CEPSTRUM VALUE.
CPYL RA B1 EQV NOT1
SBA Y 18
TSTL A=0, A<0, INCM; NEW MAXIMUM?
JMP SP4; NO.
Listing of S207I (continued).
Listing of S207I (continued).
As with other executive commands '7920' had a reject branch. The command was rejected if a previous transfer operation was not completed. When accepted, however, the interrupt program accessed successive locations of the buffer, output the values via the D/A converter and simultaneously replaced these with new values via the A/D converter. The timings of the interrupts were controlled by an external variable-frequency oscillator. In operation, therefore, the contents of one buffer were processed as a block by the main program while the contents of another buffer were output and renewed by the interrupt program. When the interrupt program finished transferring the contents of its buffer, the buffers were swapped. The data just processed was thus output and renewed, while the main program processed the last block of data input.

The program may be considered in two parts. The control section, below label 'GO', selects which of the two buffers DATA1 and DATA2 should be input/output and which should be processed. The processing section, below label SP1, simulates interruption and adaptive reiteration. Initially, the pitch period of the speech in the buffer (spanning one processing period) is estimated by the cepstrum method. The cepstrum transform is calculated by the fixed-point subroutines described in Chapter 3. When the predictor delay, \( m \), has been calculated the gain coefficient B is estimated. Reiteration is then performed by the section of code below label SP6.

This deposits the signal values predicted from the transmitted section into the latter part of the buffer corresponding to the interrupted signal section. The processing of the buffer is then completed, and it is ready to be output.

Unlike programs INT5 and INT9, program S207I could not operate at a 10 kHz sample rate. The speech was slowed down to 1/32 of real speed, to allow a longer sample interval of 3.2 ms to be used. This allowed adequate time to perform the processing involved.

6.3. FORTRAN programs.

These programs were written for off-line processing on an ICL 1904A computer. They were used in conjunction with the data collection and repro-
duction facilities provided by the Modular-1 computer, described in the last chapter. As an intermediate step in its processing, the paper tape data output by the Modular-1 was always stored on digital magnetic tape in the 1904A computer. The processing programs then read data from this magnetic tape, and wrote the processed data values on another one, prior to output on paper tape. The use of intermediate magnetic tapes was found to increase convenience and reliability of processing especially when one speech section was to be processed several times. It also reduced the inconvenience caused when a paper tape was damaged.

The following programs are described:

Magnetic tape subroutines - for convenient storage of serial data on magnetic tape.

PE21 - pitch measurement by cepstrum method, as described in Part 1 of this thesis.

PRO14 - rectangular interruption and reiteration incorporating adaptive prediction.
Magnetic tape subroutines.

For efficiency of storage, data has to be written on magnetic tape in blocks, typically containing several hundred words. For signal processing, however, data values are usually required singly although in serial order. These subroutines were written to permit data values to be written and accessed singly, yet stored on magnetic tape in records of 400 REAL variables. The tape is given an input or output channel number by an initial statement in the program description segment of the type:

```
INPUT 5 = HTO (DATAFILE0001)/810
```

The magnetic tape DATAFILE0001, with data block length of 810 words, is allocated input channel number 5, for example.

Data is written as follows. Initially, subroutine TPINIT is called:

```
CALL TPINIT (I)
```

Its argument, I, is the channel number of the magnetic tape to be written on. Successive REAL data values are written by subroutine TPWRITE:

```
CALL TPWRITE (X)
```

This value X is stored in a buffer within these subroutines. When this is full, containing 400 values, it is written onto magnetic tape. Also written are the counts of the first and last values written in this record. In the first record, these values would be 1 and 400; in the next, 401 and 800, and so on. These values enable the reading subroutines to locate required data values efficiently.

When all data values have been written, subroutine TPTERM is called. This fills any empty data locations in the buffer with zeros, and writes this onto tape. It is called:

```
CALL TPTERM (N)
```

N is set by TPTERM to the total number of data values which have been written onto tape.

Data values are read as follows. Initially, subroutine TPSET is called, to allocate an input magnetic tape:

```
CALL TPSET (I)
```

where I is the channel number of the magnetic tape to be read. The N th data
value is read by function TPREAD;

\[ X = \text{TPREAD}(N) \]

TPREAD examines the current record in its buffer, which was read from tape. If \( N \) lies between the lower and upper data count values written with that record, TPREAD accesses the required value from its buffer and returns this to the program. If \( N \) lies outside this range, TPREAD reads forward or rewinds the tape to find the required record. If the value of \( N \) exceeds the number of values written on tape (i.e. the subroutine attempts to read beyond the end of data) zero value is returned. This feature is convenient, as the number of samples in a file may not comprise a whole number of processing periods, or the number of samples may not be known accurately. In both cases, the simplest solution is often to arrange for the simulation program to read slightly beyond the end of data to ensure that all is processed.
TAPE READING SUBROUTINES.

INITIALLY CALL TPSET(ICHANNEL).

THEN (I)TH VALUE IS OBTAINED:

VALUE=TPREAD(I)


SUBROUTINE TPSET(I)

DIMENSION A(400)

COMMON/TREAD/N, IEND, ILOWER, IUPPER, A

N=I

IEND=0

REWIND N

READ (N) ILOWER, IUPPER, A

IF (ILOWER.EQ.1.AND. IUPPER.EQ.400) RETURN

WRITE (2, 100) N

100 FORMAT(/, 'WRONG TAPE IN CHANNEL NO. ', I3)

STOP

END

FUNCTION TPREAD(I)

DIMENSION A(400)

COMMON/TREAD/N, IEND, ILOWER, IUPPER, A

IF (I, LT, 1) GO TO 8

1 IF (I, LT, ILOWER) GO TO 2

2 IF (I, GT, IUPPER) GO TO 5

TPREAD = A(I-ILower+1)

RETURN

3 IF (ILower-I, LT, 401) GO TO 3

REWIND N

GO TO 4

BACKSPACE N

4 IEND=0

GO TO 6

5 IF (IEND, EQ, 1) GO TO 8

6 READ (N, END=7) ILOWER, IUPPER, A

GO TO 1

7 IEND=1

8 TPREAD = 0.0

RETURN

END

Listing of subroutines TPSET, TPREAD.
TAPE WRITING SUBROUTINES.
INITIALLY CALL TPINIT(ICHANNEL).
CALL TPWRITE(X) TO STORE X.
CALL TPTERM(NUMSTORED) TO TERMINATE WRITE PROCESS.
'NUMSTORED' HOLDS NO. VALUES WRITTEN.
J.S. SEVER Wright. OCT. 1971. EDN. 1.

Listing of subroutines TPINIT, TWRITE, TTERM.

SUBROUTINE TPINIT(I)
DIMENSION A(400)
COMMON/THRITE/A,N,ILOWER,IUPPER,ICOUNT,IWRITE
N=I
REVIEW N
ILower=1
IUPPER=400
ICOUNT=0
WRITE=0
RETURN
END

SUBROUTINE TWRITE(X)
DIMENSION A(400)
COMMON/THRITE/A,N,ILOWER,IUPPER,ICOUNT,IWRITE
WRITE=WRITE+1
ICOUNT=ICOUNT+1
A(IWRITE)=X
IF (WRITE.LT.400) RETURN
WRITE (N) ILOWER,IUPPER,A
ILower=ILower+400
IUPPER=IUPPER+400
WRITE=0
RETURN
END

SUBROUTINE TTERM(NUM)
DIMENSION A(400)
COMMON/THRITE/A,N,ILOWER,IUPPER,ICOUNT,IWRITE
DO 1 I=WRITE+1,400
1 A(I)=0.0
WRITE (N) ILOWER,IUPPER,A
ENDFILE N
RETURN
END
This program estimates the pitch period of a section of speech at intervals of 12.8 ms, by the cepstrum method described in Part 1 of this thesis. The estimation of pitch period in this way consumed by far the largest proportion of processor time when simulation programs were run. When several simulations were performed with the same section of speech, therefore, the pitch data was estimated separately by program PE21 and stored for repeated use. On the ICL 1904A computer, this program required approximately thirty minutes to estimate the pitch period in ten seconds of speech.

Speech sampled data was read from a magnetic tape, and the calculated value of pitch period were written on another tape, using the magnetic tape subroutine just described. The subroutines EDINIT, EDWRITE and EDTERM, used to write the estimated pitch values, were similar to the subroutines TPINIT, TPWRITE and TPTERM just described. They wrote data onto magnetic tape in blocks of 60, instead of 400, data variables. They made less efficient use of magnetic tape, but used a far smaller buffer store than the TP - subroutines and were adequate when only small amounts of data were involved.

The subroutine NLOGN (N, X, LX, DIR) is used here to calculate the discrete Fourier transform of complex array X(LX) by the part Fourier transform method.
TO COMPUTE PITCH PERIOD OF SPEECH BY CEPSTRUM METHOD.

COMPLEX C
DIMENSION A(256),C(256),WEIGHT(256)

SET CHANNEL NO. OF INPUT DATA MAG. TAPE.
CALL TPSET(3)

SET CHANNEL NO. OF OUTPUT (PITCH) MAG. TAPE.
CALL EDINIT(4)

GENERATE TABLE OF COEFFICIENTS FOR HAMMING WEIGHTING.
CONST=2.0*3.1415926535/255.0

DO 1 I=1,256
1 WEIGHT(I)=0.54-0.46*COS(FLOAT(I-1)*CONST)

ENTER LOOP TO PROCESS 100,000 SAMPLES IN 128-LONG BLOCKS.
DO 2 IC=0,101000,128

MOVE DATA DOWN ARRAY A, AND READ NEW DATA IN TO TOP.
DO 3 J=1,128
A(J)=A(J+128)
3 A(J+128)=TFREAD(IC+J)

WEIGHT CONTENTS OF 'A', AND CONVERT TO COMPLEX.
DO 4 J=1,256
4 C(J)=CMPLX(A(J)*HEIGHT(J),0.0)

CALCULATE DFT OF ARRAY C BY FFT METHOD.
CALL NLOGN(8,C,256,-1.0)

FIND LOG(AMPLITUDE SPECTRUM) FROM ARRAY C.
DO 5 J=1,256
5 C(J)=CMPLX(ALOG(CABS(C(J))+1.0E-30),0.0)

CALCULATE ITS FFT, TO OBTAIN CEPSTRUM.
CALL NLOGN(8,C,256,-1.0)

NOW FIND LOCATION OF MAXIMUM.
IND=0
VAL=0.0

DO 6 J=11,129
6 CONINUE
GET A CEPSTRUM VALUE AND APPLY LINEAR WEIGHTING.
X=REAL(C(J))*(1.0+FLOAT(J)/30.0)
IF (X.LE.VAL) GO TO 6
IND=J-1
VAL=X

STORE VALUE OF IND.
XIND=FLOAT(IND)
2 CALL EDWRITE(XIND)

END OF PROGRAM NOW.
CALL EDTERM(N)
WRITE (2,100) N
100 FORMAT(//////,40X,110)
STOP
END
This program simulates interruption and reiteration incorporating adaptive prediction. The prediction process accounts for the pitch period and rate of amplitude growth of the speech signal, by the methods described in Chapter 5.4 in Part 1 of this thesis. The program operates with transmitted speech sections of 128 samples, and processing periods of 256 sample intervals (12.8 ms and 25.6 ms in real time). The reiteration process requires the preceding and current transmitted speech section, in arrays A and B, when calculating the sampled data for the last interrupted section.

The program within the DO-loop to statement number 2 simulates a complete processing period. The DO-loop to statement number 3 shifts the contents of array B to array A, and reads the sample values of the current transmitted section to array B. The rectified sum of the previous B-array, in location BSUM, is also moved to location ASUM. This value is required in the calculation of growth coefficient COEFF, performed later. The pitch period is also input. These values are read from magnetic tapes by the subroutines described in previous sections.

The DO-loop to statement 31 calculates the rectified sum of the new B array and stores it in BSUM. This value, together with the value of ASUM, is used to calculate the growth coefficient, COEFF, for the prediction process.

The reiteration process is performed by the DO-loop to statement number 7. The count of predicted samples (128 are required in all) is stored in NBASE. The value of variable NREIT was previously set to the number of samples which spanned all complete pitch periods of the last transmitted section in array A. The contents of array A between A (NREIT+1) and A (128) are now used to simulate the predictor delay line. Instead of repeatedly removing the contents of A (NREIT+1), shifting down the remaining contents of the array and restoring the scaled value at A (128), a 'circular store' simulation method is employed. Instead of moving the contents of the array down, the address of the accessed location is successively incremented upwards from (NREIT+1) to 128. At each stage the contents of the location are accessed, scaled, output and restored in the
MASTER PR014

SIMULATES INTERRUPTION WITH ADAPTIVE REITERATION SYNTHESIS.
PREDICTOR CONTINUES PITCH AND AMPLITUDE GROWTH.
AMP. GROWTH COEFFICIENT COMPUTED FOR BEST JOINUP.
PROCESSING PERIOD - 256 SAMPLES. TRANSMISSION PERIOD - 128.

DIMENSION A(128),B(128)
SET SPEECH INPUT, PITCH AND OUTPUT CHANNELS (3,4 AND 5).
CALL TPS(3)
CALL EDSET(4)
CALL TPINIT(5)

SET UP INITIALLY.
BSUM=500.0
DO 1 I=1,128

B(I)=TPREAD(I)
ENTER PROCESSING LOOP.

DO 2 IC=256,101000,256

MOVE LAST TRANS. SECT. FROM B TO A.
INPUT NEW SECTION TO B.
AND OUTPUT IT AS WELL.
ASUM=BSUM
BSUM=0.0
DO 3 I=1,128

A(I)=B(I)
B(I)=TPREAD(I+IC)

CALL TPWRITE(A(I))

FIND NO. SAMPLES CONTAINING ALL COMPLETE PITCH PERIODS IN A.
IPITCH=IFIX(EDREAD(IC/128-1))
NREIT=(128/IPITCH)*IPITCH
CALCULATE AMPLITUDE GROWTH COEFFICIENT.

DO 31 I=1,NREIT

BSUM=BSUM+ABS(B(I))
BSUM=BSUM/FLOAT(NREIT)
COEFF=(BSUM/ASUM)**(FLOAT(NREIT/256.0))

OUTPUT LAST NREIT SAMPLES IN A UNTIL 128 OUTPUT.
NBASE=0

6 DO 7 I=129-NREIT,128

X,A(I)=A(I)*COEFF
CALL TPWRITE(X)
NBASE=NBASE+1
IF (NBASE.EQ.128) GO TO 2
7 CONTINUE
GO TO 6
2 CONTINUE

END OF PROCESSING.
CALL TPTER(N)
WRITE (2,100) N
100 FORMAT(///,40X,110)
STOP
END

Listing of program PR014.
same buffer location. When the address of the accessed location exceeds 128, it is reset to \((NREIT+1)\) again and the process continues until \(NBASE\) sample values have been predicted. This approach involves far less computational effort than the direct simulation of delay, although its effect is identical. The predicted sample values are written onto magnetic tape by the subroutines described earlier.