AN IMPROVED SOLID STATE TERMINAL
ANALOG SPEECH SYNTHESIZER

by

RAYMOND SAMUEL TOMLINSON

B.S.E.E.
Rensselaer Polytechnic Institute (1963)

SUBMITTED IN PARTIAL FULFILLMENT OF THE
REQUIREMENTS FOR THE DEGREE OF
MASTER OF SCIENCE

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY
June, 1965

Signature of Author
Department of Electrical Engineering, May 28, 1961

Certified by
Thesis Supervisor

Accepted by
Chairman, Departmental Committee on Graduate Students
AN IMPROVED SOLID STATE TERMINAL
ANALOG SPEECH SYNTHESIZER

by

RAYMOND SAMUEL TOMLINSON

Submitted to the Department of Electrical Engineering on May 28, 1965 in partial fulfillment of the requirements for the degree of Master of Science.

ABSTRACT

This report discusses the application of modern technology to developing an improved terminal analog or resonance speech synthesizer. The concept of hybrid computation is used to combine the speed of analog circuitry with the precision and flexibility of digital control. The familiar cascaded pole analog is realized with analog computer techniques in which the circuit parameters are controlled with digital signals from a small computer. The experimenter exercises control over the synthesizer through on line commands typed in at a typewriter console. The ability to produce an utterance and almost instantly judge its quality, modify it, and try again, allows the desired result to be attained much more rapidly than has been previously possible. Flexible data input is provided through several sources, including a light pen with which the required control signal curves may be drawn on a CRT display. Provision is made for processing the input data in a general way to allow the specification of something other than formant frequency or glottal frequency.

Thesis Supervisor: Kenneth N. Stevens
Title: Professor of Electrical Engineering
ACKNOWLEDGEMENT

The author wishes to express his thanks to Professor Stevens for his aid, opinions and guidance. Thanks are also extended to the members of the Speech Group for their assistance in preparing this thesis. Particular thanks are due Mr. Keith North for his invaluable help in assembling the hardware, Mr. James Williams for designing and constructing some of the circuits we have used, and Mrs. Jane Heinz, Miss Eleanor River and Mr. Ned Renhaim for their aid in preparing the manuscript. This work would not have been possible without the support and facilities of the Speech Communications Group, the Research Laboratory of Electronics and their supporting agencies.
# TABLE OF CONTENTS

**Chapter One - Outline of the Problem**

1.1 Introduction ........................................ 1

1.2 What Makes a Speech Synthesizer ..................... 3

1.3 Hybrid Computation .................................. 4

1.4 Programming Considerations ........................... 7

**Chapter Two - Hardware Design**

2.1 Overall Synthesizer Description ................... 10

2.2 Formant Circuits .................................... 11

2.3 The Pole-Zero Pair Circuit ........................... 12

2.4 Perturbations Due to Parasitic Phase Shift .......... 16

2.5 Glottal Source ...................................... 20

2.6 Noise Generator and Modulator ....................... 21

2.7 Digital Attenuators .................................. 22

2.8 Interconnecting Switches .............................. 22

2.9 Control Logic ....................................... 23

2.10 Miscellaneous Circuits ............................... 25

2.10.1 Overload Detectors .............................. 25

2.10.2 Higher Pole Correction ......................... 26

2.10.3 Output Amplifier ................................ 27

**Chapter Three - The Synthesizer Control Program**

3.1 Overall Program Description ......................... 28

3.2 Description of Part One .............................. 28

3.2.1 Commands and Data Storage ...................... 29

3.2.2 Data Input ....................................... 31

3.2.3 Data Display ..................................... 34

3.2.4 Calibration of the Synthesizer .................. 35

3.2.5 Other Features ................................... 35

3.2.6 Error Comments ................................... 35

3.3 Operation of the Editor ............................. 36

3.4 Operation of Part Two ............................... 37

3.4.1 Expression Evaluation ............................ 39
3.4.2 Special Events Compiler ............... 40
3.4.3 Error Comments ....................... 41
3.5 Operation of Part Three .................. 42

Chapter Four - Performance Evaluation
4.1 The Scope of the Evaluation ............... 45
4.2 Technical Performance .................... 46
  4.2.1 Pole-Zero Pair Circuit ............... 46
  4.2.2 Formant Circuits ..................... 48
  4.2.3 Interconnecting Switches ............. 48
  4.2.4 The Overall Synthesizer ............. 49
4.3 Listening Evaluation ...................... 50

References
LIST OF FIGURES

2.1 Block Diagram of Complete Synthesizer.
2.2a Analog Computer Simulation of Resonator.
2.2b Realization of the Resonator Using Operational Amplifiers.
2.3 (does not exist)
2.4 First Realization of Pole-Zero Pair.
2.5 Modification of Figure 2.4 to Allow Pole Frequency to be Less Than the Zero Frequency.
2.6 Block Diagram of Final Pole-Zero Circuit.
2.7 Detailed Circuit for Pole-Zero Pair.
2.8 Simple Resonator Diagram.
2.9 Root Locus Diagram for Simple Resonator.
2.10 Differential Integrator.
2.11 Pole-Zero Plot for Non-Ideal Differential Integrator.
2.12 Phase Angle of Non-Ideal Integrator.
2.13 Root Locus Diagram for Resonator with Non-Ideal Integrator (no damping).
2.14 Determination of Phase Shift due to Extraneous Pole and Zero.
2.15 Attenuator Stage Operation
2.16 Interconnecting Switch Circuit.
2.17 Overload Circuit Schematics.
2.18 Output Amplifier.
3.1 (does not exist)
3.2 Pushdown Evaluation Flow Diagram.
3.3 Special Events Compiler Flow Diagram.
LIST OF FIGURES - con't

4.1 Pole Frequency Accuracy Curves.
4.2 Zero Frequency Accuracy Curves.
4.3 Bandwidth Accuracy Curves.
4.4 Synthesized Vowels and Diphthongs.
4.5 Synthesized "Where are you?"
4.6 Assorted Stop Consonants.
4.7 "Are you a good boy or a bad boy"

TABLES

4.1 D.c Evaluation Data.
4.2 Interconnecting Switch Data.
4.3 Signal to Noise Measurements.
Chapter One Outline of the Problem

1.1 Introduction

Before going into the "how" of speech synthesis, a few words on the "why" are in order. First, the generation of speech is an end in itself. Speech communication using bandwidth compression techniques requires some means of regenerating the speech at the receiving end from the transmitted signals (1,3). Man-machine communication using speech requires the machine to be able to generate speech.

Second, speech synthesis is an important means of testing theories of human speech production. A theory of speech production which is capable of guiding an accurate synthesis is better than one which cannot. Speech synthesizers are valuable tools for improving the understanding of the human speech production process. Speech synthesizers are also useful in studies of speech perception because they can generate well defined signals and manipulate its parameters to examine human responses to stimuli so generated.

Of the different types of speech synthesizers which have been constructed, the terminal analog or resonance
synthesizer is by far the most popular. The principal reason for this popularity is that the terminal analog synthesizer is relatively easy to use, and is able to synthesize most speech sounds. Furthermore, such a synthesizer is considerably less complex than most other types.

As more sophisticated and precise theories have been proposed, existing synthesizers have been found lacking in precision, accuracy, range, and stability. This thesis reports the development of a new solid-state terminal analog speech synthesizer with marked improvement in these areas. In addition, a great deal of flexibility has been achieved through the use of on-line computer control.

It is not our intention that this thesis be either a treatise on the general problem of speech synthesis, nor a detailed description of the many operational features of the synthesizer. Rather, we hope to present a description of the general capability of the synthesizer, plus a more detailed look at certain important aspects of the design. Separate documents are being prepared to serve as operation manuals for those who wish to use the synthesizer.
1.2 What Makes a Speech Synthesizer

The basis of all terminal analog speech synthesizers is that the spectrum of the signal can be separated into a source function and the transfer function of the vocal tract \((10,11)\). Furthermore, Fant \((11)\) has shown that for an acoustic tube driven at one end and open at the other, the transfer function can be approximated as closely as desired by a cascade of circuits containing only complex conjugate pole pairs. That such a model works is proven by the existence of many working speech synthesizers which use this principle.

A model consisting solely of cascaded complex conjugate pole pairs is only adequate for certain classes of sounds. The two notable examples where this model breaks down are sounds in which the tube is excited other than at the end, and when more than one acoustic tube is involved. An analysis of such acoustic configurations \((16,23,24)\) produces a transfer function which contains complex conjugate zero pairs as well as pole pairs. If a synthesizer is to produce sounds with spectra containing zeroes, a transfer function containing zeroes must be included.

There are two principal excitation sources in most speech: a quasi-periodic pulse source caused by vibration
of the vocal cords (glottal source), and random noise caused by turbulence as the airstream passes through a constriction in the vocal tract (noise source). A general purpose speech synthesizer should contain both kinds of sources. In addition, the noise source should be modulated by the glottal source (8) since, in the vocal tract, the intensity of the noise varies with the velocity of the airstream which, in turn, is varied by the vibrations of the vocal cords. Other sources to produce pops and clicks could be included, but this has not been done in this synthesizer.

In addition to the above, there are several other auxiliary features which might be included. These features are not essential to the synthesis of speech, but contribute to adjustment, testing, or monitoring of the synthesizer. Several such features are included in the present synthesizer, and will be discussed later.

1.3 Hybrid Computation

The motivation behind the use of hybrid computation (5,6,22) is that analog computation is superior (at the present time) to digital computation for certain tasks, and vice-versa. In particular, analog computation is especially suited to high speed linear processing of time functions.
The ease with which most filtering problems may be solved by bringing together the right assortment of electrical circuits is fairly well established. The solution to this same problem using high speed digital computers is conceptually easy also, but even the highest speed computers presently available cannot perform arithmetic at a rate fast enough to adequately process these time functions in real time.

When non-linear or time-variant computation is attempted, the picture starts to change. The complexity of analog circuitry instantly increases. Good, high speed, wide range multipliers and similar circuitry is prohibitively expensive. The complexity of digital computation increases somewhat, but there is no additional cost. The problem of computation speed still exists however. When the problem of communicating with the user is considered, analog computation is an extremely weak contender. Effective communication cannot be achieved with analog techniques. About the best that can be done is by means of a kind of "sign language" in which the user and synthesizer draw pictures to each other. Such a scheme cannot compare to the tremendous flexibility offered by a digital computer capable of handling both "sign language" communication, as well as symbols which can be combined into higher level languages including languages composed of a subset of English.
The obvious next step is to combine the better features of both digital and analog computation. Such a combination is referred to as hybrid computation. The task of communication with the user can then be assigned to the computer along with certain information processing tasks, and the signal processing performed by the analog circuitry.

In the case of speech synthesis, the signal processing is linear and time-variant. The time variation is what must be controlled by the digital computer. Henke (4) has described a method of controlling a terminal analog speech synthesizer through a digital computer. Because the analog circuits he used were previously built to accept analog control voltages, he achieved control over the synthesizer with analog signals from a digital to analog converter.

If the intention is to control the synthesizer from digital signals, a more direct approach is possible. Such a scheme has been described by Whitman (5). The parameters of the analog circuit are set with "coefficient multipliers" at particular points in the signal path. These coefficient multipliers simply multiply (amplify or attenuate) the signal by a particular amount. In the method used in this synthesizer, the amount of multiplication is varied in discrete steps by the use of transistor switches controlled by digital signals.
In bypassing the intermediate analog control signal, one major deterrent to high quality synthesis is removed. The high speed analog multiplier necessary in a strictly analog circuit is eliminated, and in its place a digital multiplier can be used. Such a multiplier (7) can be built to perform with high precision at very moderate cost.

1.4 Programming Considerations

There currently exist two prevalent modes of computer usage. Closed shop operation in which the user's program is run by a machine operator and the results returned later is characterized by turn-around times (the delay before the results are received) ranging from one or two hours to several days. On-line operation in which the user runs his own program is characterized by a fast turn-around time of minutes or seconds at the expense of higher computer usage.

For an application such as the speech synthesis being discussed, the advantages of on-line operation outweigh any disadvantages. The availability of small computers at MIT makes such operation even more attractive. The PDP-1 computer in the Research Laboratory of Electronics was chosen for four reasons. First, a cathode-ray tube display is available to present pertinent data in a meaningful and
easily interpreted way. Second, a high speed magnetic drum may be used for storing large amounts of data. Third, a special panel has been added to the computer to connect user's external I-O equipment in an almost trivial way. Fourth, it is relatively easy to schedule time on the computer.

The functions of the control program for the speech synthesizer may be grouped into three categories. The program must accept input data or assist in its production, provide information to the user when he requests it, and generate control signals for the synthesizer to produce an utterance.

To maintain generality, it is desirable to divorce the operation of the program, as it appears to the user, from the actual synthesizer as much as possible. And it should be possible to make alterations to the synthesizer without having to rewrite the program. The program utilizes a "mapping specification" to govern the relation between input and output data. This specification is typically prepared in advance to be read in by the user. By doing so, the peculiarities of the synthesizer are taken into consideration without the user having to worry about it. In addition, if the user wishes to change the appearance of the machine he may do so by altering the mapping specification. More will
be said about this feature in chapters three and four.

Since the intention is for the program to be usable, with a minimum of training, by anyone, it should not be necessary for a potential user to possess programming ability, or even to know how to operate a computer beyond such basic things as knowing what a sense switch is. To this end, the program has been written to be completely self-contained. The entire operation of the program is controlled through typed in commands. There is no necessity for manually starting the program or using any other program in any way. The program tries to recover from its own mistakes or mistakes made by the computer.

The command language has been made as concise as possible to allow rapid type-in of commands. Such a concise language is not quite as mnemonic as one which is less concise, but after using the program for a while, most users have no difficulty remembering which command does what. The command language is discussed in more detail in chapter three.
Chapter Two  Hardware Design

2.1 Overall Synthesizer Description

The synthesizer hardware can be divided into components having specific functions such as the noise generator or the pole-zero pair circuit. Mechanically, each of these components is entirely contained on one or more plug-in circuit cards. The sections which follow describe each of these various components in detail. This section describes the overall synthesizer.

A block diagram of the analog circuitry is illustrated in figure 2.1. There are several possible signal paths through the synthesizer according to the type of sound to be produced. The path to be used is selected by a set of transistor switches. For vowels, glides, and some consonants which can be characterized by a transfer function consisting of complex conjugate pairs of poles (CCP poles) driven by a pulse source at the glottis, the signal path is indicated with solid lines. Similarly, aspiration, frication (voiced and unvoiced), and nasalization are provided by other paths through the synthesizer. These paths are drawn on figure 2.1 according to the legend on the figure. It
should be noted that inputs to the output amplifier are mixed (added) together to produce the output.

2.2 Formant Circuits

The formant circuits used were originally built for an experimental forerunner of the present synthesizer. Details of the design are given in reference (6). The circuit used is shown in figure 2.2. The bandwidth of the circuit is not programmable, and must be set by a potentiometer (R'). The center frequency is determined by a digital attenuator between the first and second integrators. When the attenuator setting is 0 db., the center frequency is $G$, the gain factor of the integrators. As the attenuation is increased, the center frequency decreases.

The transfer function for the circuit is

$$H(s) = \frac{s^2 + \omega_n^2}{s^2 + 2\omega_n s + \omega_n^2} = \frac{AG^2}{s^2 + \sigma s + AG^2}$$

Thus, the natural frequency ($\omega_n$) varies as the square root of the attenuation. Since the maximum attenuator setting is one, the maximum frequency of resonance is $G$. In practice, the gain factors of the two integrators are not equal, and the maximum frequency is the geometric mean of the two gains.
The maximum frequencies chosen for the four formants are 1600 cps for F1, and 5000 cps for F2, F3, and F4. These frequencies are far above those normally encountered in speech, but the extra range might be useful for studying the speech of infants, or non-speech sounds. The theoretical minimum frequencies are 41 cps for F1, and 128 cps for F2, F3, and F4. When operating at their lower extremes, the formant circuits tend to become noisy due to the lower loop gain. A practical minimum for these center frequencies is twice the theoretical minimum.

2.3 The Pole-Zero Pair Circuit

From a consideration of the configuration of the vocal tract during the production of nasals and nasalized sounds, and from spectrum matching experiments for these sounds (23,24), it is evident that a transfer function containing only poles does not comprise a very accurate analog. The spectrum of the noise produced by turbulence at a constriction in the vocal tract in the production of fricatives also contains zeroes (16) due to the effect of the cavity behind the constriction. For the production of these sounds, it is necessary to introduce zeroes into the transfer function produced by the synthesizer.
In the synthesizer being described, a circuit has been incorporated to satisfy these needs. Since a circuit whose transfer function consists solely of a zero is physically impossible to build due to the high gain at high frequencies, an actual circuit must have a transfer function containing poles as well to cancel the effect of the zeroes at very high frequencies. There are two choices for the location of these poles. They may be at very high frequency where their effect on the signal is negligible, or at a lower frequency where their effect may be useful. The latter approach was deemed more satisfactory. Two factors which influenced this decision are the need for a CCP pole in the noise filter (16), and the fact that if the extra pole is not needed, it can be raised to a high frequency so that its effect is minimized. The circuit to be designed is to provide a pair of complex conjugate zeroes, and a pair of complex conjugate poles.

The first circuit considered for this role used the obvious method of building a feedback amplifier whose feedback path contained one of the circuits previously designed for producing a pair of poles. The pertinent equations are developed below. Referring to figure 2.4a, we
write

\[ e_{out} = A \left( e_{in} - \frac{e_{out} \omega_n^2}{s^2 + 2\delta \omega_n s + \omega_n^2} \right) \]

\[ e_{out} = \frac{A e_{in}}{1 + \frac{A \omega_n^2}{s^2 + 2\delta \omega_n s + \omega_n^2}} \]

\[ \frac{e_{out}}{e_{in}} = \frac{A \left( s^2 + 2\delta \omega_n s + \omega_n^2 \right)}{s^2 + 2\delta \omega_n s + (1+A) \omega_n^2} \]

The zeroes of the transfer function are the poles of the feedback function \( F(s) \), as we would expect, and the poles of the transfer function are those of the feedback function multiplied by square root \((1+A)\). Furthermore, the bandwidths of both the pole and the zero are the same. Two drawbacks are immediately apparent. First the transfer function goes to zero when the pole frequency is the same as the zero frequency \((A=0)\), and second, the pole frequency must lie above the zero frequency unless \(A\) can be made negative. Both these problems can theoretically be overcome by judiciously reordering the components and effectively making the gain go negative. The circuit for this is shown in figure 2.5. The relevant equations for this circuit are given below.

\[ e_{out} = Ge_{in} - \frac{\omega_n^2}{s^2 + 2\delta \omega_n s + \omega_n^2} \left\{ G \left( K - \frac{1}{G} \right) \right\} e_{out} \]

\[ e_{out} = \frac{Ge_{in}}{1 + \frac{(GK-1)\omega_n^2}{s^2 + 2\delta \omega_n s + \omega_n^2}} = \frac{G \left( s^2 + 2\delta \omega_n s + \omega_n^2 \right)}{s^2 + 2\delta \omega_n s + GK \omega_n^2} \]

An experimental circuit revealed that parasitic phase shift in various parts of the circuit introduced wild deviations
from theory in the real part of the poles and zeroes. While it may have been possible to compensate for the parasitic phase shift, it was felt that a different approach would be better.

Writing out the desired transfer function as a function of $1/s$, we get:

$$\frac{s^2 + a s + b}{s^2 + c s + d} = \frac{1 + \frac{a}{s} + b \frac{1}{s^2}}{1 + c \frac{1}{s} + d \frac{1}{s^2}} = \frac{e_{\text{out}}}{e_{\text{in}}}$$

$$e_{\text{in}} \left(1 + \frac{a}{s} + \frac{b}{s^2}\right) - e_{\text{out}} \left(\frac{c}{s} + \frac{d}{s^2}\right) = e_{\text{out}}$$

From the last equation, we can immediately draw a block diagram of the required circuit (figure 2.6) consisting of integrators, summers and coefficient multipliers. Since the coefficient multipliers can provide no gain, the gain is supplied by the integrators. The details of the circuit may then be worked out in a standard fashion. The problem is not completely solved though, since there is still parasitic phase shift to be compensated. The general problem of phase shift is discussed in the next section.

The final circuit, shown in figure 2.7, provides a transfer function consisting of a conjugate pair of poles, and a conjugate pair of zeroes. The center frequencies of the pole and the zero are individually variable from 255 cps to 10000 cps. The bandwidths are independently variable from 10 cps to 2000 cps. The bandwidth is constant $\pm$ 20 cps
over the entire frequency range with the greatest error occurring above 3000 cps.

2.4 Perturbations Due to Parasitic Phase Shift

Since the operating frequencies of the circuits used in the synthesizer are considerably higher than those in conventional operational amplifier circuitry, the problems introduced by parasitic phase shift are more acute. For purposes of discussion, we shall refer to the simple resonator section in figure 2.8 and consider the perturbation of the poles by the presence of unwanted phase shift.

For the circuit in figure 2.8, the root locus is as pictured in figure 2.9. As the loop gain (A) is increased, the poles move along a line parallel to the j-axis. If there is no damping (T= ), the locus coincides with the j-axis. If we now introduce a demon which produces a certain amount of phase shift without disturbing the amplitude of the loop gain, the closed loop poles will be perturbed. It can be shown (20), that at the frequency of any pole of the closed loop, the loop gain must be -1 for negative feedback. Graphically, this implies that the angles of all lines drawn to the closed loop pole from the open loop zeroes minus those from the open loop poles must be congruent modulo 360
to 180 degrees. If demons are present, their phase shift must be included in this sum as well. For the case of no damping, the closed loop poles must move along a circle centered at the origin as the phase shift produced by the demon is varied. The distance traveled along this circle must be such as to change the angle from the open loop poles at the origin by an amount which cancels the phase shift produced by the demon. If this distance is small, the circular arc may be approximated by a straight line of length $P$.

$$\frac{2\phi}{\omega} = \phi$$
$$P = \frac{\omega \phi}{2}$$

To get a feeling for the allowable phase shift, consider

$$\omega = 2\pi \left(10,000 \text{ cps}\right)$$

then

$$\phi = \frac{20}{10,000} = .002 \text{ radians or } .12 \text{ deg.}$$

Certain sources of phase error may be immediately enumerated as follows: digital attenuators (7), operational amplifiers (5,6), other components, wiring. All of these sources produce essentially linear phase lag by introducing very high frequency poles. The method of compensation is to introduce zeroes in the loop to provide cancelling phase lead. This is accomplished by adding series resistors (6) in the integrating capacitors.
If this were the only source of phase shift, there would be no problem. There is another source which is not obvious, and which is the most significant source of error. Consider the differential integrator of figure 2.10. Its transfer function is found as follows.

\[ e_o = A \left( e_2 \frac{1}{R_2 + \frac{sC_2}{R_1}} - e_1 \frac{1}{R_1 + \frac{sC_1}{R_1}} - e_o \frac{R_1}{R_1 + \frac{sC_1}{R_1}} \right) \]

Let \( \tau_1 = R_1C_1 \), \( \tau_2 = R_2C_2 \)

\[ e_o = A \left( e_2 \frac{1}{1 + s\tau_2} - e_1 \frac{1}{1 + s\tau_1} - e_o \frac{s\tau_1}{1 + s\tau_1} \right) \]

assume \( A \to \infty \) then

\[ \frac{e_2}{1 + s\tau_2} - \frac{e_1}{1 + s\tau_1} = \frac{s\tau_1}{1 + s\tau_1} e_o \]

assume \( e_1 = 0 \)

if \( \tau_1 = \tau_2 \)

then \( \frac{e_o}{e_2} = \frac{1}{s\tau_1} \)

and it is a simple integrator. If, \( \tau_1 \neq \tau_2 \), then there is an extraneous real axis pole and zero added to the integrator.

The pole-zero plot for this non-ideal integrator is shown in figure 2.11. The phase contribution for j-axis frequencies is shown in figure 2.12. If this non-ideal integrator is included in a loop with an ideal integrator, the root locus in figure 2.13 results. The asymptotes (20)
are at \( \pm 90 \) degrees and pass through the centroid of the open loop poles and zeroes. The location of the centroid then gives the maximum perturbation of the real part of the closed loop poles.

A rough estimate of the perturbation for a more realistic situation is found as follows. Refer to figure 2.14.

\[
\tau_1 = \tau, \quad \sigma_1 = \frac{1}{\tau_1}
\]
\[
\frac{1}{\tau_2} = \frac{1}{\tau_1} + \epsilon, \quad \sigma_2 = \sigma_1 + \epsilon
\]

The phase shift on the \( j \)-axis due to the pole and zero at frequency \( \omega \) is

\[
\phi = \Theta_1 - \Theta_2 = \arctan\left(\frac{\omega}{\sigma_1}\right) - \arctan\left(\frac{\omega}{\sigma_2}\right)
\]
\[
\phi = \arctan\left(\frac{\omega}{\sigma_1}\right) - \arctan\left(\frac{\omega}{\sigma_1 + \epsilon}\right)
\]

approximating the difference as a differential

\[
\frac{\phi}{\Delta \sigma} = -\frac{d}{d \sigma}\left\{\arctan\left(\frac{\omega}{\sigma}\right)\right\} \quad \phi = \frac{\omega}{\sigma_1^2 + \omega^2} \Delta \sigma
\]
\[
\Delta \sigma = \sigma_2 - \sigma_1 = \epsilon \quad \phi = \frac{\omega}{\sigma_1^2 - \omega^2} \epsilon
\]

for \( \omega = \sigma_1 \) (roughly the maximum resonant frequency)

\[
\phi = \frac{\epsilon \omega}{2 \omega^2} = \frac{\epsilon}{2\omega} = \frac{\epsilon}{2\sigma_1} = \frac{\epsilon \tau_1}{2}
\]

If \( \epsilon = 0.1 \sigma_1 \) (one percent error in time constant)
then \[ \phi = \frac{0.01\pi}{2} = 0.005 \text{ radians} \]

The perturbation for \( \omega_{\text{max}} = 2\pi \cdot 10^4 \)

\[ P = \frac{1}{2} \omega \phi = 2\pi \cdot 50 = 314 \text{ radians} = 50 \text{ cps} \]

To achieve an error of \( \pm 20 \text{ cps} \), requires adjusting the time constants to be equal within \( \pm 0.4 \% \) percent.

2.5 Glottal Source

The glottal source consists of a single shot triggered by the computer to produce a narrow rectangular pulse. The spectrum of this pulse is essentially flat over the audio band. A filter consisting of two low frequency real axis poles is used to shape the spectrum. Presently, this shaping filter is combined with the higher pole correction filter and the radiation compensation filter. No attempt was made to design any of these filters, and circuits developed for an exploratory model (6) were used. The final model will have these filters separated.

After being shaped, the glottal pulse is attenuated by a digital attenuator to determine its amplitude. The attenuated signal is connected to the input of the formant circuits by one of the interconnecting switches.
2.6 Noise Generator and Modulator

Our original intention was to include a noise circuit in the synthesizer as part of this thesis. This has not been possible, however, due to shortcomings in available noise sources. The scheme to be used for modulating the noise has been described elsewhere (8). Briefly, the idea is to hard limit wide-band noise with a limiting value determined by the modulating signal. Low-pass filtering the result yields essentially band-limited Gaussian noise.

The problem arises due to the production of a d.c. component in the signal if the noise source amplitude distribution is asymmetrical. That is, although the noise may have zero mean, it has one sign for more of the time than the other. The output of the limiting process will then have a d.c. component since it depends only on the sign of its input. The simple expedient of passing the signal through a capacitor will not work due to the variations in the modulating signal. The application of d.c. to any of the filters causes unwanted noise to be developed. Therefore, more work is necessary to eliminate the d.c. from the output.
2.7 Digital Attenuators

The design of the digital attenuators has been adequately described elsewhere (7). A brief description follows for completeness.

The basic principle of one stage of the attenuator is illustrated in figure 2.15. By opening or closing the switch, one of two attenuation values may be selected. When the switch is open, the signal passes through unattenuated. When the switch is closed, the attenuation is determined by the voltage divider ratio. Eight such stages followed by unity gain buffer amplifiers are cascaded to form a complete attenuator. The attenuation provided by the eight stages is 1/4, 1/2, 1, 2, 4, 8, 16, and 32 decibels. The attenuation of the complete attenuator is the sum of the attenuations (in decibels) of those stages with the switch closed. The switch consists of a single transistor which is either cut off or saturated according to a digital level. By using various combinations of digital levels, the attenuation may be varied from 0 to 63 3/4 db in 1/4 db steps.

2.8 Interconnecting Switches

The interconnection of the various parts of the synthe-
sizer as illustrated in figure 2.1 is done by a set of digitally operated transistor switches. Each switch has two states. In the "on" state, a connection is established from the input to output. In the "off" state, there is no connection.

When the switch is off, the switch should effectively disappear as far as the load is concerned. The output impedance should be high in the off position. This requires a more complicated series switch rather than a shunt type. In addition, the offset should be low (less than 5 mv.), and a minimum of control current should appear in either the source or the load. The circuit in figure 2.16 accomplishes this. Its operation is briefly as follows. The first four transistors operate as a phase splitter and push-pull amplifier for the digital signal. When zero volts is applied to the digital input, the switch transistors are back-biased to 6 volts and switch is open. When -3 volts is applied to the input, the transistors are forward biased through the 20k resistors closing the switch.

2.9 Control Logic

The functions of data buffering and routing, timing, and general control are provided by an assembly of logic modules. The control logic is composed of four parts. The
bulk of the logic is devoted to thirteen 8-bit flip-flop registers. The outputs of these flip-flops are used to drive the digital inputs of the attenuators and configuration switches. The contents of these registers are altered by commands from the computer.

An eight bit counter is used as a clock to keep track of time. The counter is wired to count down to zero. When the counter is stepped below zero, the overflow pulse is used to set a status flip-flop which signals the computer and inhibits further counting. The counting pulses are produced by a 100 kc oscillator. The purpose of the clock is to avoid the necessity of keeping track of time in the computer program.

The signals sent to the synthesizer consist of 18 bits from the IO register of the computer plus an i-o transfer pulse generated by the execution of a particular iot instruction in the program. A set of inverters is used to increase the driving capability of the data lines from the computer and to provide the inverted sense of the signals. The pulse from the computer is also cleaned up with a pulse amplifier to assure reliable operation.

An address decoder is used to decode bits 1-5 of the IO register. The decoded levels are used to gate the iot pulse at the input of a pulse amplifier. The output pulse from
the pulse amplifier is used to jam-transfer IO bits 10-17 into the selected buffer register. Three of these pulses are used to set the clock, generate glottal pulses, and generate marker pulses.

Two levels are sent back to the computer from the synthesizer. One signals that the clock has reached zero, the other signals that an overload has occurred. These two levels are taken as status bits in the sequence break or priority interrupt system of the computer. When the level is one, the normal program is interrupted to allow the proper action to be taken. The design of the logic is completely straightforward and will not be discussed further.

2.10 Miscellaneous Circuits

2.10.1 Overload Detectors

Due to the high Q of the tuned circuits in the synthesizer, the possibility of developing high signal levels exists. This is particularly true if two of the formant frequencies are close together. After passing through succeeding stages of the synthesizer, the distortion may no longer be discernible through casual listening.
In order to detect the occurrence of overload and signal the operator, an overload detector has been incorporated into the design. The circuit for accomplishing this was developed by Jim Williams of the Speech Communications group. The circuit is shown in figure 2.17 and operates as follows. The input signal is applied to the bases of two transistors operating as level discriminators. The emitters are biased at \( \pm 5 \) volts such that if the signal exceeds this voltage, a transistor will conduct developing a voltage across the common load resistor. This pulse-like signal then sets a flip-flop whose output serves to indicate that an overload has occurred. The flip-flop is reset by the computer program when it recognizes the overload condition.

A set of level discriminators is supplied for each critical point in the synthesizer. Currently fifteen are used.

2.10.2 Higher Pole Correction

Fant (2,11) has shown that if the transfer function of a uniform straight tube is approximated by a function including a finite number of poles, there will be an error in the level of the high frequency spectrum. A uniform tube closed at one end and open at the other has a transfer
function containing peaks at all odd multiples of the lowest
resonance. If the lowest resonance is at 500 cps and the
approximation contains the first four poles (500 cps, 1500
cps, 2500 cps, and 3500 cps), the error will be 41 db. at
4000 cps. If this error is not compensated, a noticeable
degradation in quality results.

The higher pole correction circuit used in this synthe-
sizer was developed for an exploratory synthesizer (6). The
higher pole correction is combined with other circuits to
shape the glottal spectrum, and correct for the radiation
impedance. If the higher pole correction is isolated from
these other circuits, the transfer function consists of a
CCP pole and two real axis zeroes (figure 2.18). The
transfer function obtained matches the curve given by Fant
to ± 4 db. up to 4 kcps.

2.10.3 Output Amplifier

The output amplifier serves to mix the signals from
different parts of the synthesizer and provide a low
impedance output to drive tape recorders and monitoring
equipment. The circuit used is shown in figure 2.19. The
mixing is accomplished by the input resistor network. An
emitter follower drives the output transformer whose bal-
anced output appears at the output connectors.
3.1 Overall Program Description

The program is logically and physically divided into six parts. One of these parts resides in core storage permanently and contains routines common to all other parts plus all of the variables which must be preserved as each part is run. The other five parts reside on the drum auxiliary storage unit and are read into core storage when they are to be run. Each of these parts runs as a procedure. That is, there is no modification of any part of the program while it runs. Thus, when it is finished there is no necessity to save it. The next time this part is to be used, a fresh copy is read from the drum. This saves a drum swap every time another part is brought in.

The terms input data and output data as used here refer to lists of numbers each element of which is a sample from a time function. The entire list specifies the time function for the duration of an utterance. The distinction between input data and output data is that output data lists contain numbers which are directly transferred to the synthesizer,
whereas input data is entered by the experimenter, and is never directly used to control the synthesizer.

The five parts are named part one, part two, part three, buffer display, and editor. The division of labor between them is such that no appreciable program swapping occurs. The tasks assigned to these five parts are as follows. Part one is responsible for accepting, generating, and manipulating input data and coordinating the activities of the other parts. Part two performs the translation of input data into output data according to a text string entered via the editor. Part three uses the output data generated by part two to actually control the synthesizer. This part also monitors the overload detector and timing of the whole operation. The editor is available to enter and alter the text string specifying the input-output translation. The buffer display is a short program that displays a graph of the current input data.

3.2 Description of Part One

3.2.1 Commands and Data Storage

All action within part one is initiated by typing in a command consisting of a single lower case letter or special
character which may be preceded by any number of numeric arguments. The number of arguments depends on the command and some commands do different things depending on the number of arguments. An argument is composed of one or more decimal integers separated by one of the operators "+", or "-". Typing a comma terminates the current argument and starts the next. Argument handling commands allow errors to be corrected anywhere in the string as well as wiping out the whole string.

The main data storage is provided in a number of tables on the magnetic drum. Each drum field (consisting of 4096 18 bit computer words) used for data is divided into several tables. The number of tables per field is set by a parameter typed in when the program is first started. Each element of a table is associated with a particular sample time in the utterance. Successive elements specify the value of a function at successive multiples of the sampling interval. Thus each table contains samples of a time function. Nominally, each drum field represents one utterance, i.e. the tables on that field contain all the functions necessary to specify one utterance. In practice, however, this arrangement is only necessary when that field is to be used directly for an utterance.
In the process of arranging all of the data needed to produce an utterance, it often becomes necessary to move data from one table to another. Several commands are provided to move data around on the drum. These commands also make it possible to examine (on the CRT display) various data tables. In order to simplify much of the data manipulation, the concept of a current table and current field is used. Manipulation commands which do not specify a table or field use the current table or field. Thus if all the data being maneuvered is on one field, that argument may be omitted, and the current field will be used. This reduces the amount of typing necessary and the possibility of error which it entails.

3.2.2 Data Input

Data may be entered in three ways: by drawing curves on the CRT display with the light pen, by programmed curve fitting to points entered from the typewriter, or from prepared paper tapes.

When the light pen input mode is active, the program displays a pattern of random points on the CRT until the light pen sees one of them. The program then draws a cross to the four extremes of the light pen's field of vision. From the x-axis average of these points, the closest time coordinate which is a multiple of the sampling interval is
found and the average y coordinate replaces the previous entry for that time in the core data table. Irregularities in the input are smoothed by passing the x and y coordinates through programmed single pole low pass filters (as though both x and y were sampled time functions) with a time constant of sixteen points. This filtering is applied only to the points seen by the pen and not to data already in the table.

The curve fitting routine uses the LaGrange interpolation formula (21) to exactly fit segments of n-th order polynomials to input points from the typewriter. The equations used are given below.

Let \( P_k(x) = \prod_{i=0}^{n} (x-x_i) \)

Then \( y = \sum_{k=0}^{m} \frac{y_k P_k(x)}{P_k(x_k)} \)

The Xi and Yk are the n points straddling X plus one more point fixed by the program to produce a continuous derivative at the transition between segments of the fitted curve. The elements of the sum are computed in single precision fixed point for speed. To avoid an intermediate overflow or underflow, the numerator product is carried to the point of imminent overflow, then a division is done. The multiplications are continued in this way until there are none left, then the remaining divisions are performed. When one of the x coordinate passes one of the Xi, the range of specified
points used in the fitting is changed so the interpolated
point is as nearly in the middle as possible.

For polynomials of degree one, a separate routine is
used to draw straight line segments between the specified
points. The formula used is

\[ y = y_1 + \frac{(x-x_1)(y_2-y_1)}{(x_2-x_1)} \]

Where \( X_1, Y_1, \) and \( X_2, Y_2 \) surround \( X \). In all cases the
curves are drawn only between and including the most extreme
points specified.

Paper tape input is done by reading binary data blocks
from which data is loaded point by point into the core data
table. The primary purpose of this input is for reloading
previously dumped data or to enter the output from another
program.
At any time, the core data table may be smoothed using a smoothing command. This command passes those points in the core data table which are currently in the horizontal range of the display (as determined by display manipulation commands) through a non-linear low pass filter. The equation used in delay notation is

\[ O_n = \frac{O_{n-1} + I_{n+1}}{2} + \frac{f(I_n - \frac{O_{n-1} + I_{n+1}}{2})}{4} \]

where \( O_k \) is the \( k \)th output sample point,
\( I_k \) is the \( k \)th input sample point

and \( f(x) = \begin{cases} x & \text{for } |x| < 16 \\ 15 & \text{for } x \geq 16 \\ -15 & \text{for } x \leq -16 \end{cases} \)

The use of the clipping function "f" is to minimize the effect of any artifacts in
the data. By including three points in determining On, the effect of time shift inherent in simpler low pass filters is minimized.

3.2.3 Data Display

Normally, while sitting in its listen loop waiting for type-in, the program displays the contents of the core data table in graphical form on the CRT display. This display includes a fully labelled coordinate grid plus data. The grid is calculated to have major divisions of 5, 10, 20, 50, 100, etc. and minor divisions of 1, 2, 5, 10, 20, etc. As the size of the display is changed, the grid is adjusted to always yield a reasonable display. The current table and field numbers are also displayed for quick reference.

Commands are available to move the display in any direction or to change the vertical or horizontal scale. This allows any portion of the data to be examined in detail. These commands affect only the display and not the data itself.
3.2.4 Calibration of the Synthesizer

To facilitate any necessary calibration or adjustment of the synthesizer, two commands may be typed into part one. One command causes 18 bits to be transferred from the test word toggle register of the computer to the synthesizer as data. This is useful for balancing the attenuators or generating steady tones for signal tracing. The second causes a comparison of the timing of the clock in the synthesizer to the computer's timing. This was intended for setting the 100kc clock frequency, but seems to indicate only that the computer is 5 percent slower than advertised.

3.2.5 Other Features

A command is available for setting the time of the first glottal pulse to within 10 usec. Commands are also used to initiate other parts of the program such as the editor and part two.

3.2.6 Error Comments

Whenever an inconsistent set of arguments is given to a command, an error comment is typed out. The comments are abbreviated to five letters such as "ildeg" (illegal degree) to avoid the time necessary to type out longer comments and
to conserve storage space. About 15 comments are used for various types of errors. The program also checks itself for errors and if impossible situations arise, the comment "program error" is typed. The computer performs a parity check on data read from the drum and if an error is detected, the program prints out the comment "drerr" (drum error). There are no halts in the program. All unusual conditions cause an error printout and the program recovers and continues running. The program is now sufficiently debugged to be able to attribute most stops or program errors to malfunction of the computer.

3.3 Operation of the Editor

The mapping specification is stored on the drum as a text string. This string is scanned by part two during the translation of input data to output data. To allow rapid changes to this string, an editor patterned after the PDP-1 Expensive Typewriter program (26) has been included. Lines of text in the buffer may be examined, deleted, changed or inserted using various commands. Provision is also made for reading and punching paper tape. In normal use these specifications would be prepared in advance, and the user would not worry about what it consists of.
Internally, the text is represented by a character set consisting of lower case letters, numbers, and the punctuation \(+ - ( ) = \times \vee \wedge \sim / . , \tab \space\) and carriage return. The necessary conversion is performed automatically for input and output. The different character set is employed to facilitate the operation of part two by making the classification of character types easier. The conversion is done in the editor because space is more plentiful.

The commands to the editor are also single letters preceded by arguments as in part one. Unfortunately, there are not enough letters to accommodate all the commands, so the same letters are used for commands to both part one and the editor. To help avoid confusion, commands typed to the editor are printed in red and require a carriage return to be typed after the letter.

3.4 Operation of Part Two

Upon command to part one to produce an utterance, part two is called to perform the input-output translation. The details of this translation are determined by scanning a text string which was previously entered into the program through the editor. In addition to producing the output data tables, other information is prepared for the utterance.
The output data is first compiled onto a drum field in the same format as the input data tables. The drum field used is not available for input data. The storage allocation on this output field is determined by an output variable declaration line in the beginning of the specification string.

The body of the specification consists of a series of algebraic statements written in terms of input variables, output variables, temporary variables, constants, and operators. Each statement is applied to every element of the variable tables.

Variable names consist of two parts, a letter (i, o, or t for input, output, or temporary) denoting the type of variable, followed by a number identifying the variable. Thus i2 refers to input data table 2, o4 refers to that output data table whose contents will eventually appear on channel 4. The storage for temporary variables is allocated automatically by the appearance of the variable name. Storage for temporary variables and for intermediate results is provided on drum fields not being used for any other purpose. All arithmetic is done in floating point to avoid scaling problems. Input data and output data are converted to floating point for the calculations, and then returned to fixed point.
3.4.1 Expression Evaluation

The process of evaluating the expression is divided into three parts. The evaluation is done using a full priority push-down scheme which manipulates operators and pointers to the data tables feeding them to a second routine which does the actual arithmetic. A third routine does the scanning of the text string assembling constants, function names, variables names etc. and providing pointers and operators to the main routine.

The evaluation is defined recursively as follows. The routine evaluates the expression up to but not including the operator whose precedence is not greater than the precedence of the operator which called it. When it can go no further, the routine returns a pointer to the table containing the results of the sub-expression just evaluated. The flow diagram for the process is illustrated in figure 3.2. Note that the scanning process does not penetrate a carriage return and effectively causes repeated returns to close all extra open parentheses and finally return to where it was called from. Unary operators, distinguished by being preceded by another operator, are done immediately regardless of priority. Omitted operators are taken to mean implied multiplication. That is, \(12(t1-3)\) is taken to be \(12x(t1-3)\).
Operators on the same level except for "=" are done from left to right since this generates less intermediate storage. Equal sign is done from right to left to follow conventional practice.

3.4.2 Special Events Compiler

After completing the output compilation, the rest of the text is scanned to compile the special events table. The special events table is used for changing the configuration switch settings during the utterance. Strings such as "at 0,100, set 1,2 reset 3" are processed. For each time specified, two words are stored in the special events table. The first gives the time, the second contains a mask indicating in one half, which switches are to be set and in the other half, which are to be reset at that time.

The algorithm for decoding these statements is very simple-minded. The flow chart in figure 3.3 should be self-explanatory. Note that the first letter of the words is used to distinguish them. Thus "antidisestablishmentarianism" would be the same as "at". Also space and comma as well as any other non-numeric and non-alphabetic character serve only as a separators, and 20 separators are as good as one. The intention is to provide an easily readable format with a fairly obvious meaning.
When all the text has been scanned, the output tables on the drum are squeezed into half their original space. That is, 4096 words are packed two to a word into 2048 words. As this is done, the range of the numbers is checked. Out of range numbers (which cannot be transferred to the synthesizer) cause an informative print-out indicating in which output channel they occurred. The packing is necessary so all the data can fit into core along with part three.

3.4.3 Error Comments

While performing the translation, the program checks for certain syntactic errors such as illegal punctuation or illegal function names. The arithmetic is also checked for such errors as division by zero, or logarithm of a non-positive number. If any such error is found, a comment is typed out indicating what line of text was being processed when the error was found. No attempt has been made to distinguish between various types of errors.
3.5 Operation of Part Three

The third part of the program takes care of the actual transfer of data to the synthesizer. The cycle of events in the main loop is as follows.

1) Set up the data. That is, unpack it from the table, tack a channel number on its front end, and put it into the output list. This is done for each output item that is to occur at the next sample point.

2) Wait for the right time to transfer the data to the synthesizer. (The sequence break routine is keeping track of the time.)

3) Transfer the data and return to step one unless there is no more data.

4) If there is no more data, reset the data pointers, delay for a while and return to step 1.

The sequence break routine is initiated when the hardware clock times out. It does the following.
1) If a glottal pulse is to be generated, do it.

2) If a glottal pulse will occur within the range of one countdown of the clock, go to step 8.

3) Set the clock to delay one millisecond.

4) Add the amount of the clock setting to the current time.

5) If it is time for more data and the main loop is ready to put it out, signal the main loop to go ahead.

6) If the main loop is not ready, type out error comment "too fast" and stop the utterance.

7) Return to the main loop.

8) Calculate the interval to the next glottal pulse, and set the clock for that delay. Set a flag indicating that a glottal pulse is to be generated at the next clock break.

9) Go to step 4.
The glottal pulse timing is determined from the frequency data at every sample point. The time left to the next pulse is adjusted to the new frequency, and the interval between subsequent glottal pulses is computed. After each glottal pulse, this is used to determine when to produce the next pulse. The updating process is inhibited until after the first glottal pulse is generated to allow this to be precisely specified. In the case of zero frequency pulses, a flag is set to indicate that regardless of any numbers which may be calculated, no glottal pulse (except the first) is to be generated. As soon as the frequency becomes non-zero, the flag is cleared.

While the program is waiting for the go ahead signal to transfer the next data, it checks to see if any characters have been typed on the typewriter. The characters so typed are taken as commands to either go back to part one, or to stop talking but not go back to part one. Naturally, there is also a command to start talking again.
4.1 The Scope of the Evaluation

The evaluation of the synthesizer consists of two parts. It is important to know how it performs technically as well as how good the speech it produces is. Technical performance can be measured in terms of signal to noise ratio, distortion, and accuracy. The quality of the speech it produces has been judged through informal listening and by comparing spectrograms of the synthesized versus the original speech.

In addition to the above two measures of performance, another measure is how efficiently an experimenter can use the synthesizer. At this time, we have not had an opportunity to let many people try their hand at using the device, so this aspect of its performance can only be judged from the author's rather prejudiced experience.
4.2 Technical Performance

4.2.1 Pole-Zero Pair Circuit

Since the pole-zero pair circuit is prone to more problems than the rest of the resonant circuits, extensive testing was performed on it. The tests were pole frequency accuracy (both real and imaginary parts), zero frequency accuracy, dc offset over several frequency settings and ambient temperatures, and interchangability measurements.

The worst results were obtained in the dc measurements at elevated temperatures. At low frequency settings, as much as 3 volts of dc offset appeared at the output. The cause of this is principally the low loop gain combined with offsets introduced by the attenuator. The feedback in the circuit tends to compensate for these offsets by offsetting the output. The low gain in the feedback path requires a large offset to effect a small compensation at the input. Fortunately, the output of the pole-zero circuit is not applied to any rapidly switching circuit. The results of the dc tests are tabulated in table 4.1.

The frequency accuracy tests were made by measuring the frequencies at which the amplitude response was 3 db. down from the resonant peak (3db. up points in the case of the zeroes). The average of these was taken as the center
frequency, and the difference divided by two as the bandwidth. Both the center frequency attenuators and bandwidth attenuators were varied to determine the accuracy and the orthogonality of the controls. The results of these measurements appear in graphs in figures 4.1, 4.2, and 4.3.

A comment about the center frequency accuracy curves is necessary. In the circuit, the quantity which is actually set by the attenuators is the so-called natural frequency of the circuit

$$\omega_n = \sqrt{\omega_0^2 + \sigma^2}$$

The curve which depicted is not $\omega_n$. As a result, for high damping where $\omega$ and $\omega_n$ are greatly different, a high error results. The curves for smaller bandwidths more accurately reflect the actual performance.

The bandwidth accuracy curves in figure 4.3 indicate a rather high error in the bandwidth of the zero. This is due to an error in choosing the zero bandwidth trimming resistor. The measurement of the bandwidth of the zero is very difficult due to the low signal level, and the resistor was chosen on the basis of incorrect measurements. After the resistor was re-trimmed, the bandwidth accuracy was improved to be about the same as that of the pole. Again the use of the 3 dB points is not an accurate measure of the bandwidth for low $Q$, and the error at low frequencies is not
accurately given by the curves.

The interchangability tests revealed that replacing operational amplifiers or attenuators caused essentially no change in the ac accuracy. The differences that were noted are completely accounted for by experimental errors such as meter interpolation. The dc performance was upset due to the external balancing adjustments on the operational amplifiers. After readjustment, the performance was equivalent to the original.

4.2.2 Formant Circuits

Since the formant circuits were not built as part of this thesis, no extensive testing has been done on them. Some measurements were made when the circuits were built, and are reported in reference (6). There is no reason to expect worse performance than that of the pole-zero pair circuit. The dc characteristics are much better due to the direct feedback from output to input. The offset is typically less than 5 mv.

4.2.3 Interconnecting Switches

The results of measurements on one of the interconnecting switches are tabulated in table 4.2.
4.2.4 The Overall Synthesizer

The electrical tests made on the complete synthesizer consisted chiefly of noise measurements. There are three principle sources of noise in the attenuator. Random noise is generated by the circuits themselves and produces a rushing sound with a tone-like quality due to the filtering action of the resonators. Non-random induced noise is produced mainly by ground potential differences. Hum caused by induction (usually magnetic) from the 60 cps power line is one of these sources. The other is ground shift caused by the logic signals from the computer. These appear as varying high pitched sounds. The third source is caused by switching of the digital attenuators (7). Three types of noise exist in this category. Additive noise appears due to crosstalk of the digital signals (due to capacitance in the switching transistors) into the analog signal paths. Multiplicative noise results from modulation effects of switching the analog signals. The third type is caused by switching the unwanted dc component in the analog signals.

The measurements and listening tests indicate the random noise and multiplicative noise are insignificant. In the static condition (no connection to the computer), the major offender is hum. With the computer connected and no utterance being generated, the most audible noise is that from the changing logic levels from the computer. If the
attenuator settings are changed as in speech but with no excitation applied, the switched dc noise becomes predominant. The results of the noise measurements are tabulated in table 4.3. The figure of most interest is the noise during speaking with no excitation. Compared with a typical vowel /o/, the S/N ratio is 35 db.

4.3 Listening Evaluation

The first utterances which were generated, were simple vowels and diphthongs. The data for generating these sounds were taken from references (13,14,17), as well as spectrograms of the author saying the sounds. A wiring error and inaccuracy in the spectrograms contributed to a poor result. After fixing the wiring error, and changing the mapping specification to compensate for the inaccuracy in the spectrograph, a very good result was obtained. The glides /w/ and /j/ were also produced with equally good results. Spectrograms of the synthesized speech are illustrated in figure 4.4.

The next step in difficulty was the production of connected speech. The sentenced "where are you" was recorded and a spectrogram made. Data from the spectrogram was entered into the program using piecewise linear curve-
fitting. The result was good. Spectrograms of the synthesized version are shown in figure 4.5.

The stops /b/, /d/, and /g/ were produced next. These consonants can be characterized by a rapid transition of the formants (typically in 35 msec.) (15,18). The thirty sounds produced by combining the 10 vowels and diphthongs /i/, /e/, /a/, /æ/, /ɨ/, /o/, /u/, /ɪ/, /æɨ/, and /æw/ with the above stops in the initial position were synthesized to produce test data for another experiment. The original data was taken from spectrograms of the sounds and augmented with the results of other researchers (13,15,17,18), and then hand-tailored to produce the final utterance. Good results were obtained with the exception of /di/ which could not be successfully synthesized. A representative collection of spectrograms of the results are shown in figure 4.6. It was noticed in producing the sounds, that very little "thump" (which has been noticed in other synthesizers) was present.

The most recent utterance produced is a sentence containing voiced stops, glides, diphthongs, and vowels. The sentence is "are you a good boy, or a bad boy?" Data for this utterance were taken directly from a spectrogram and the curves for the four formants and glottal pitch and amplitude drawn directly with the light pen input on the CRT display. It required 35 minutes to enter all the data and
produce the first draft of the utterance. The results were good but there was an error in copying part of the glottal frequency curve and the /g/ in "good" was not very clear. After a few more trials, a very good synthesis of the sentence was produced. The total time to prepare the utterance was 45 minutes.
Figure 2.1 Block Diagram of Complete Synthesizer.
Figure 2.2a Analog Computer Simulation of Resonator.

\[ \omega_0 = \frac{\sqrt{A}}{G} \quad G = \frac{1}{RC} \]

\[ BW = 2 \times \omega = \frac{1}{CR'} \]

Figure 2.2b Realization of the Resonator Using Operational Amplifiers.
Figure 2.4 First Realization of Pole-Zero Pair.

Figure 2.5 Modification of Figure 2.4 to Allow Pole Frequency to be Less Than the Zero Frequency.

Figure 2.6 Block Diagram of Final Pole-Zero Circuit.
**Figure 2.7 Detailed Circuit for Pole-Zero Pair.**

**Notes on figure 2.7:**

* R3 is augmented with a selected resistor to trim bandwidth of the zero.
* R4 is augmented with a selected resistor to trim bandwidth of pole. (R4 must be selected before R3)

R1 is selected to trim the maximum resonant frequency of the zero.

R2 is selected to trim the maximum resonant frequency of the pole.

C1 is matched to C2 (not necessary, but a good idea.).

C3 is matched to C4 (also not necessary.)

All resistor are high quality 1% 1/4 or 1/2 watt carbonfilm. C1-C4 are high quality silver-mica capacitors.

a₀, a₁, b₀, b₁ are digital attenuators.
Figure 2.8 Simple Resonator Diagram.

Figure 2.9 Root Locus Diagram for Simple Resonator.

Figure 2.10 Differential Integrator.
Figure 2.11 Pole-Zero Plot for Non-Ideal Differential Integrator.

Figure 2.12 Phase Angle of Non-Ideal Integrator.

Figure 2.13 Root-Locus Diagram for Resonator with Non-Ideal Integrator (no damping).
Fig. 2.14: Determination of Phase Shift due to Extraneous Pole and Zero
Fig. 2.15 Attenuator Stage Operation

(a) \[ A = \frac{R_2}{R_2 + R_1} \]

Basic Attenuator Principle

(b) Typical Attenuator Stage

Fig. 2.15 Attenuator Stage Operation
Fig. 2.16 Interconnecting Switch Circuit
Fig. 2.17a  Level Discriminator (Overload Detector)

Fig. 2.17b  Flip Flop and Indicator (Overload Detector)
Figure 2.18 Output Amplifier.
OP(X) is the operator whose priority is X.

PR(X) is the priority of operator X.

c.r. means carriage return.

SYL(N) is the N-th syllable in the expression.

PUSH'X means put X in the next free entry on the pushdown list.

POP'X restores X from the last entry on the pushdown list.

Figure 3.2 Pushdown Evaluation Flow Diagram.
CHR(N) is the N-th character in the string.

Figure 3.3 Special Events Compiler Flow Diagram.
Figure 4.3 Bandwidth Accuracy Curves (Pole-Zero Pair)
"roar" - original

/ðæ/ - synthesized

"roar" - synthesized

/e/ - synthesized

Fig. 4.4
Fig. 4.5  "Where are you?" - synthesized
Fig. 4.6 (cont)
Fig. 4.6 (cont)
Fig. 4.7 "Are you a good boy ...
Fig. 4.7 (cont) ... or a bad boy?"
Table 4.1

<table>
<thead>
<tr>
<th>$f_n$ of pole and zero</th>
<th>$T_{amb}$</th>
<th>d.c. output</th>
</tr>
</thead>
<tbody>
<tr>
<td>255 cps (63 3/4 db)</td>
<td>21.5°C</td>
<td>-0.92v</td>
</tr>
<tr>
<td>316 cps (50 db)</td>
<td>21.5°C</td>
<td>-0.54v</td>
</tr>
<tr>
<td>369 cps (56 db)</td>
<td>21.5°C</td>
<td>-0.32v</td>
</tr>
<tr>
<td>2000 cps (28 db)</td>
<td>21.5°C</td>
<td>+0.028v</td>
</tr>
<tr>
<td>255 cps</td>
<td>30.5°C</td>
<td>-3.7v</td>
</tr>
<tr>
<td>316 cps</td>
<td>30.5°C</td>
<td>-2.3v</td>
</tr>
<tr>
<td>369 cps</td>
<td>30.5°C</td>
<td>-1.37v</td>
</tr>
<tr>
<td>2000 cps</td>
<td>30.5°C</td>
<td>-0.5v</td>
</tr>
</tbody>
</table>

Table 4.2

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Input Impedance:</strong></td>
<td>10K in parallel with load</td>
</tr>
<tr>
<td><strong>Output Impedance:</strong></td>
<td>5 plus source</td>
</tr>
<tr>
<td><strong>Max. Output Current:</strong></td>
<td>15 ma. at ground</td>
</tr>
<tr>
<td><strong>Approx. Predicted Offset:</strong></td>
<td>1 mv. with 10 source</td>
</tr>
<tr>
<td>Vowel</td>
<td>/i/</td>
</tr>
<tr>
<td>-------</td>
<td>------</td>
</tr>
<tr>
<td>Signal/noise ratio</td>
<td>+47 db</td>
</tr>
<tr>
<td></td>
<td>+43 db</td>
</tr>
<tr>
<td></td>
<td>+31 db</td>
</tr>
</tbody>
</table>
References


26. Expensive Typewriter 11a, Memorandum PDP-22, PDP-1 Computer, Research Laboratory of Electronics, MIT.