PRELIMINARY EXPERIMENTS WITH SYNTHESIS BY RULE OF STANDARD DANISH

Peter Holtse

1. Introduction

Since the autumn of 1973 a system for synthesis by rule of Standard Danish has been under development as a joint project between the Telecommunications Research Laboratory (TFL) and the Institute of Phonetics, Copenhagen.[1]

The aim of the project is to develop a system which will generate acceptable spoken Danish from a table of stored parameter values. The system should provide a basis for testing hypotheses about the perceptual relevance of various acoustic features, and it is hoped that it will help in formulating hypotheses on subjects like temporal organization and intonation of Danish.

2. Hardware equipment

The rule system is implemented on the RC 4000 computer system of TFL. The hardware equipment consists of an RC 4000 with 32 k bytes of core and conventional peripheral equipment. The actual synthesis is performed by a digitally controlled OVE IIIc synthesizer. Control information from the central

[1] The practical problems of writing the computer programs are handled by B. Bagger Sørensen of TFL.

computer is transferred to a PDP-8/I minicomputer working as a buffer station. The buffer program controls the synthesizer by updating the fifteen control parameters of OVE once every 10 milliseconds. (Fig. 1.)

3. Synthesis strategy

The synthesis program consists of a complex of four independent computer programs communicating via the operating system of the central computer. This means that the synthesis programs are run in time sharing with any other jobs under execution. Therefore the time needed to synthesize a given string of speech may vary depending on the number of other jobs running. During the writing of the programs we have been more concerned with the development of a useful tool for phonetic research than with reducing execution time or program size.

The program synthesizes a sample of speech by chaining together a string of "segments" stored on the disc. A "segment" is a characteristic acoustic event, e.g. the explosion of a stop consonant, the steady state of a vowel, or the fricative phase of an [s]. Each segment has associated with it a table of information consisting of (1) a head and (2) a matrix of values for the central parameters of the synthesizer.

The control parameters are:

AO: amplitude of voice source
Ac: amplitude of friction noise
Ah: amplitude of hiss through formants
An: amplitude through nasal branch
F1: frequency of first formant
F2: " " second "
F3: " " third "
Ak: pole/zero ratio of fricative formants
K1: frequency of first fricative formant
K2: " " second " "

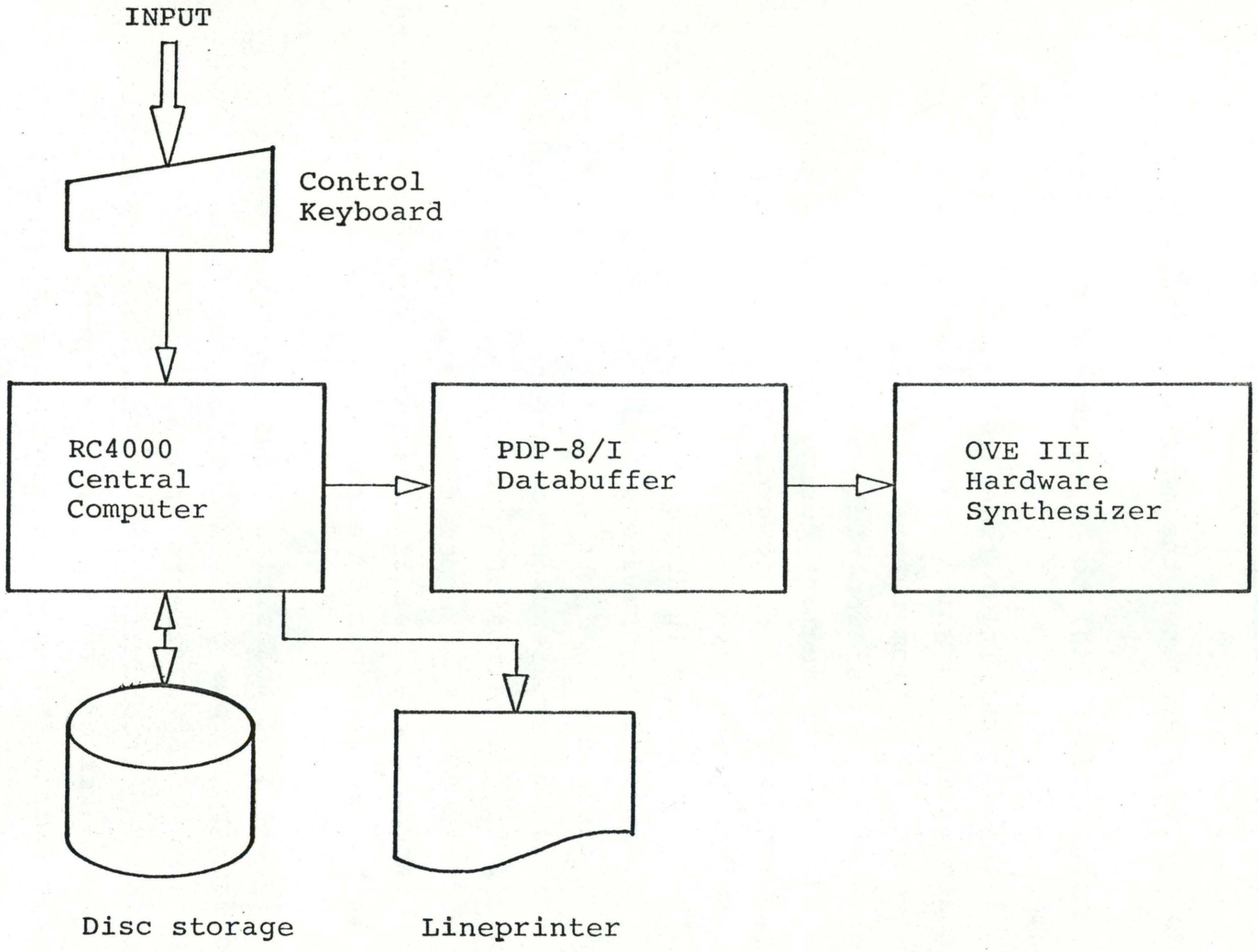


Figure 1

Hardware organization of the synthesis system.

The synthesizer needs a twelfth control parameter, FO: fundamental frequency, but the segment tables contain no information on this parameter. The band widths of the three (vowel) formants are kept at a constant value.

For each control parameter the matrix contains three entries:

(1) The target value (tg) to be reached during the segment

(2) The internal transition time (ti), i.e. the part of the segment during which the parameter is moving towards or away from the target value.

(3) The external transition time (tx), i.e. the part of the neighbouring segment during which the parameter is moving away from or towards the target value of the neighbouring segment.

All target values are listed in Hz for the frequency parameters and in dB for the amplitude parameters. The transition times are listed in centiseconds.

The transition times are used to interpolate straight lines between the target values of adjoining segments, as shown in figure 2. Straight line interpolation has been chosen as being easy to conceptualize, but if experience should demonstrate the need for it the program could be changed to use other strategies.

Figure 3 shows that transition times may have negative values. This feature allows the boundary between adjacent segments to be treated as a mere reference line.

The head of the information table for each segment tells the chaining procedures of the synthesis programs how the data of the parameter matrix should be treated. The first entry of the head is the name of the segment. The name must be two alphanumeric characters. Furthermore, the head contains the standard duration and the rank of the segment. When two segments

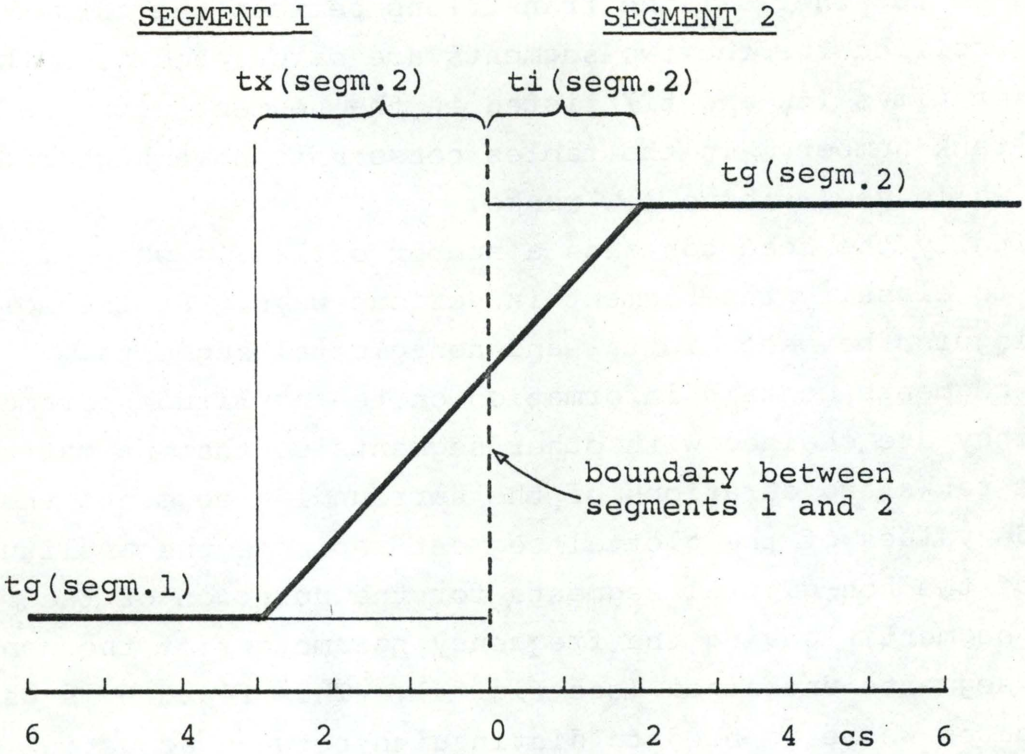


Figure 2

Straight line interpolation between two segments. Transition times in this example are taken from segment 2.

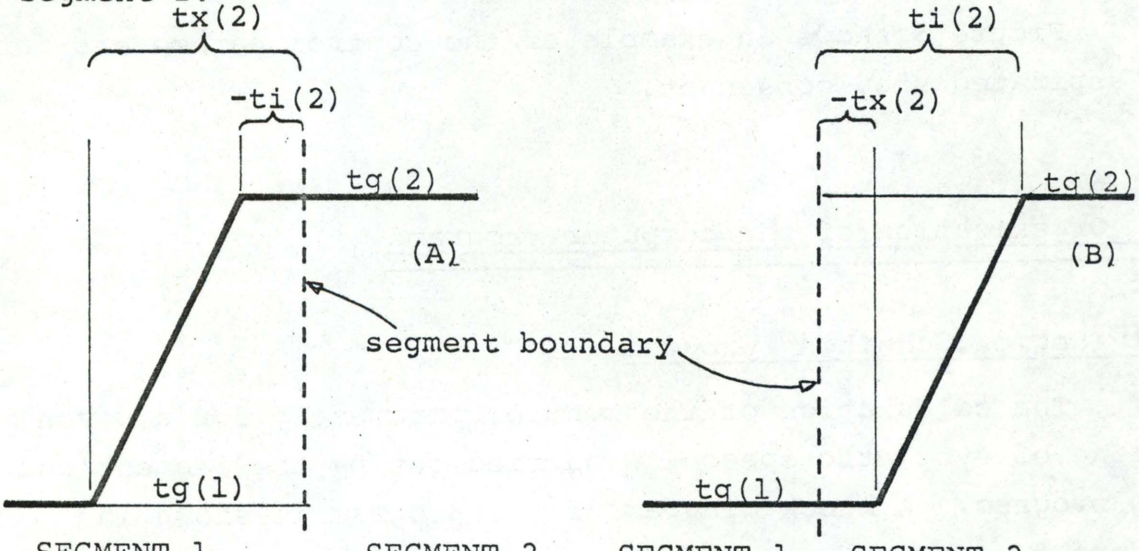


Figure 3

Examples of negative values of internal (A) and external (B) transition times. The entire transition may be moved to either side of the segment boundary line.

are chained together all the transitions between the target values specified for the two segments are calculated from the transition times (tx and ti) listed in the segment with the highest rank number. In the tables consonants have high rank numbers while vowels have low ranks.

Finally the head contains a number of labels which may be used to classify the segment in various ways. At the moment we distinguish between glottal and non-glottal segments. Glottal segments contain information on the amplitude parameters only. They are chained with other segments so that no matter what the ranks and durations of the surrounding segments the amplitude values of the glottal segment replaces the amplitude values of the non-glottal segments for the duration of the glottal segment, leaving the frequency parameters of the non-glottal segments unchanged (see fig. 4). This feature is useful in cases where we need to distinguish between acoustic parameters governed mainly by the upper articulators as opposed to acoustic parameters governed by the larynx. Thus aspiration of stop consonants and the Danish "stød" are, for the moment, represented as glottal segments.

Figure 5 shows an example of the control parameters for an aspirated stop consonant.

4. Organization of the computer programs

4.1 Segment chaining program

The calculation of the control parameters for a given string of synthetic speech is carried out by the segment chaining program. A block diagram of the program is shown in figure 6.

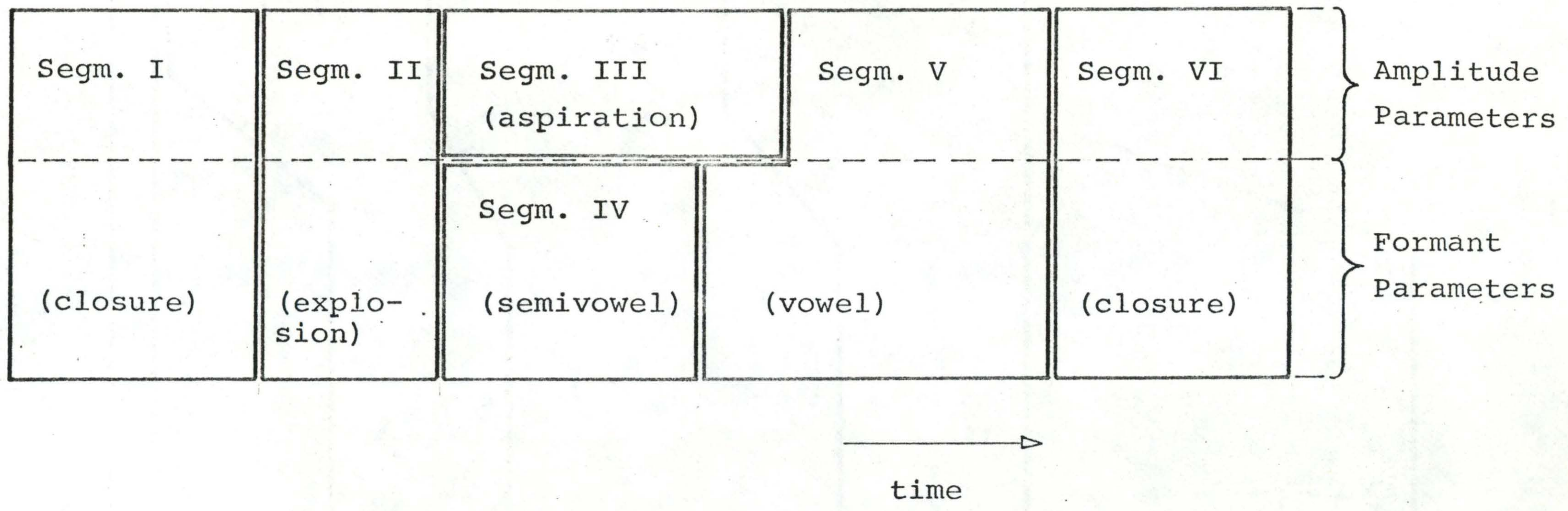


Figure 4

Schematic drawing showing how a glottal segment assumes control over the amplitude parameters and leaves the formants unchanged.

The example shows the chaining of the segments for the sequence /pjat/ = [pçad] . The segments will be listed in the input string as /closure/, /explosion/ /aspiration/,/semivowel/, /vowel/ etc.

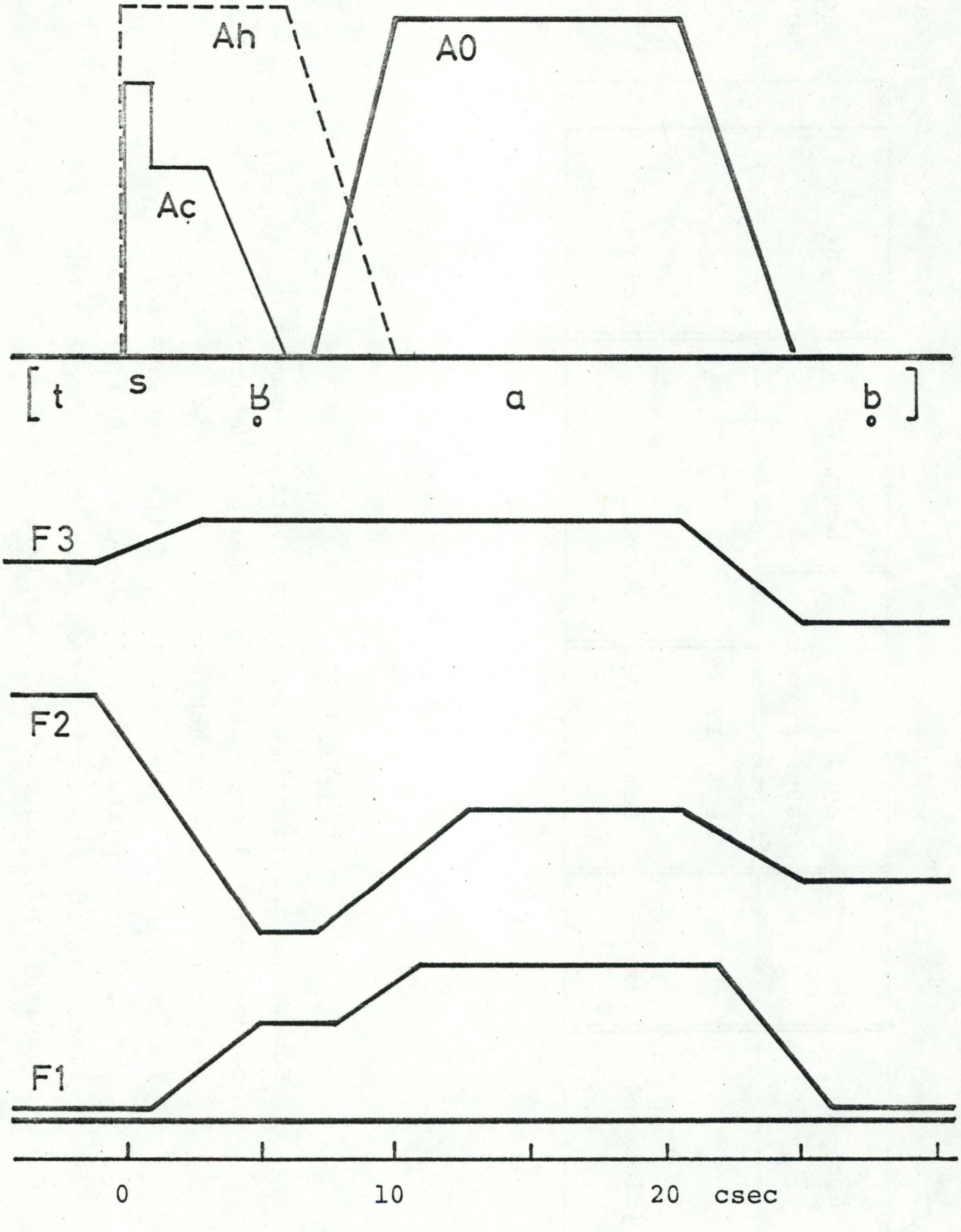
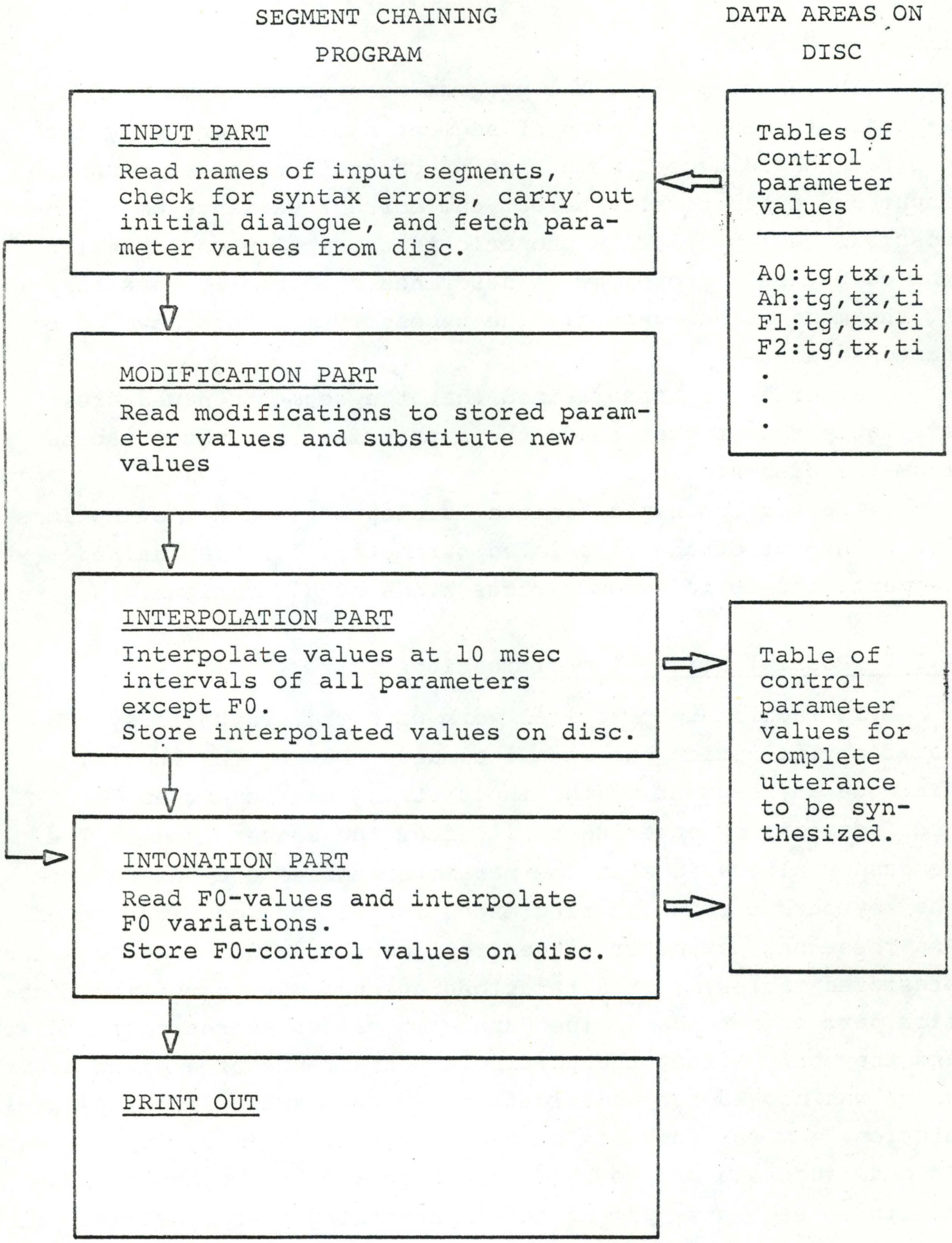


Figure 5

Example of the control parameters for the sequence /trab/.



SEGMENT CHAINING PROGRAM

DATA AREAS ON DISC

INPUT PART

Read names of input segments, check for syntax errors, carry out initial dialogue, and fetch parameter values from disc.

Tables of control parameter values

A0:tg,tx,ti
Ah:tg,tx,ti
Fl:tg,tx,ti
F2:tg,tx,ti
.
.
.

MODIFICATION PART

Read modifications to stored parameter values and substitute new values

INTERPOLATION PART

Interpolate values at 10 msec intervals of all parameters except F0.
Store interpolated values on disc.

Table of control parameter values for complete utterance to be synthesized.

INTONATION PART

Read F0-values and interpolate F0 variations.
Store F0-control values on disc.

PRINT OUT

Figure 6

Block diagram of synthesis program.

4.1.1 Input part

The input part of the program accepts as input from a keyboard terminal a string of segment names. Eventually the program should accept some kind of phonetic transcription as input and automatically select the correct variants among the possible segments. This phonetic transcription could well be the output of a phonology as described by Basbøll (this issue). At the moment, however, all the necessary segments have to be named separately.

After having ascertained that the segments named are available the program reads the appropriate parameter tables from the disc storage.

The section named "initial dialogue" is a set of options for print-out of the calculated parameters for the chained segments and options for various kinds of alterations.

4.1.2 Possibilities of modifications

To facilitate practical work with the synthesis system modification option has been included. During the initial dialogue any segment of the input string may be marked for modifications by entering a '!' after the segment name. The necessary alterations in the parameter tables are entered via the keyboard terminal during execution of the modification part. And these new parameter values replace the values read from the prestored tables. The alterations entered during the modification part do not change the parameter values stored on the disc, and they only affect the particular occurrence of a given segment which was marked for modification. Thus segment 'p2' ([p]-explosion) may for instance be used three times in an input string. If modifications are to have effect on all three occurrences all three segments 'p2' of the input string must be marked with a '!'. And the same alterations must be typed three times.

(In this way it would, of course, also be possible to make three different editions of the same segment in one synthesis run.)

When all modifications are entered the program interpolates straight lines between adjacent segments as described in section 3. The interpolated values, quantized in steps of 10 msec, for all control parameters except FO are stored on the disc.

4.1.3 Fundamental frequency control

The last part of the program calculates the values needed for the control of the fundamental frequency. At the moment there are two FO-options: Either a detailed description of the intonation pattern may be entered via the keyboard or an automatic intonation contour generator may be called. The automatic FO-variation is a slightly falling curve whose only justification is that it avoids a complete monotone which is uncomfortable.

Any detailed, or indeed natural sounding, FO-variation must be entered by hand. This is done by typing a series of points in time and their corresponding fundamental frequency values into the computer. The program then interpolates straight lines between the specified points. After interpolation the FO-control values are stored on the disc together with the other control parameters.

It is possible to make repeated changes to the FO-pattern of a given synthetic utterance without recalculating all the control parameters. This is done by typing an '=' as input string during the input part. Program control then goes directly to the intonation part of the program, bypassing the modification and interpolation parts. This allows the user to enter a new intonation pattern to the old edition of the synthesized string.

4.1.4 Print-out of control values

The last part of the program has options for printing out the complete list of control parameter values on the line printer. For visual reference a special option draws a diagram of the variations of the control parameters on the line printer.

4.2 Utility routines

Besides the segment chaining program there are three utility routines. One, the execution program, reads the output from the synthesis program on the disc, converts the frequency and amplitude values to the binary format demanded by the synthesizer, and transmits these control values to the PDP-8 buffer. Using the PDP-8 as a buffer allows quite long strings of continuous speech to be synthesized, the upper limit being given only by the storage capacity of the disc.

The other two utility routines are: "GETPAR" which has options for printing the contents of the permanent segment tables on the disc storage, and "READPAR" which enters modifications into the permanent tables.

5. Conclusion

So far the emphasis of our work has been on developing an adequate library of segment tables. This work is still in progress, and we are beginning to consider the problems of segment duration as a function of number and quality of surrounding segments. Later on we hope to develop algorithms for the control of intonation, stress, and rhythm. This work will be done in cooperation with the group working on a computer based phonology of Danish (cf. Basbøll, this issue).

References

Basbøll, H. and K. Kristensen 1974: "Preliminary work on computer testing of a generative phonology of Danish" (this issue p. 216-226)

Holmes, J.N., I.G. Mattingly, and J.N. Shearme 1964: "Speech synthesis by rule", *LS* 7, p. 127-143

Nooteboom, S.G., I.H. Slis, and L.F. Willems 1973: "Speech synthesis by rule: Why, what and how?", *IPO* 8, p. 3-13