To be published in:
Proceedings of the First West Coast Computer Faire,
San Francisco, 1977

WCCF - 1

Speech Synthesis by a Set of Rules (or Can a Set of Rules speak English?)

by D. Lloyd Rice

ABSTRACT

A description of a speech synthesis by rule system is presented which generates control parameters for a formant frequency synthesizer such as the Computalker model CT-1. The first 3 parts of the system are described in some detail, especially the rules themselves. The rules portion is described completely, with discussion of the phonetic effects of each rule. The fourth part of the system is the actual playback of the parameter data to the synthesizer. The output of the rule system is then compared with a similar set of data recovered from a human pronunciation of the same utterance. Similarities and differences between the two versions are discussed.

INTRODUCTION

Human speech is a complex and variable process. Every time you say something, it comes out a little bit differently than you ever said it before. Most of this variation tells the listener something about you, how you feel, whether you're in a hurry, whether you are tense, tired or thinking about something else. In addition to these kinds of variation there is another kind of variability, determined by the phonetic context. For example, a "p" following an "s" in the same word has little or no aspiration noise following the release of closure by the tongue, while a "p" alone at the beginning of a word may have as much as 60 or 70 msec of aspiration noise.

A computer can be programmed to speak in a cut-off, robot-like manner by fairly simple joining together of fixed sounds. Speech generated in this way is generally quite difficult to understand, even in the best of circumstances. To give the computer a better quality speaking voice, on the other hand, requires a more complex system of rules which express some of the patterns of variation observed in human speech. This paper describes such a set of rules and the linguistic mechanisms in which they operate.

A BASIC SYNTHESIS BY RULE SYSTEM

A top level flowchart of the system is given in figure 1, which shows the four basic steps in the synthesis process. First, the phonetic input string is parsed into a sequence of phonetic symbols, some of which may be marked with stress level values. A phonetic feature matrix is filled in by assigning a column of binary features for each of the phonetic symbols. In the second step, 3 groups of rules operate on the feature matrix, searching for certain patterns of features and modifying individual features in a column or perhaps by inserting or

WCCF - 2

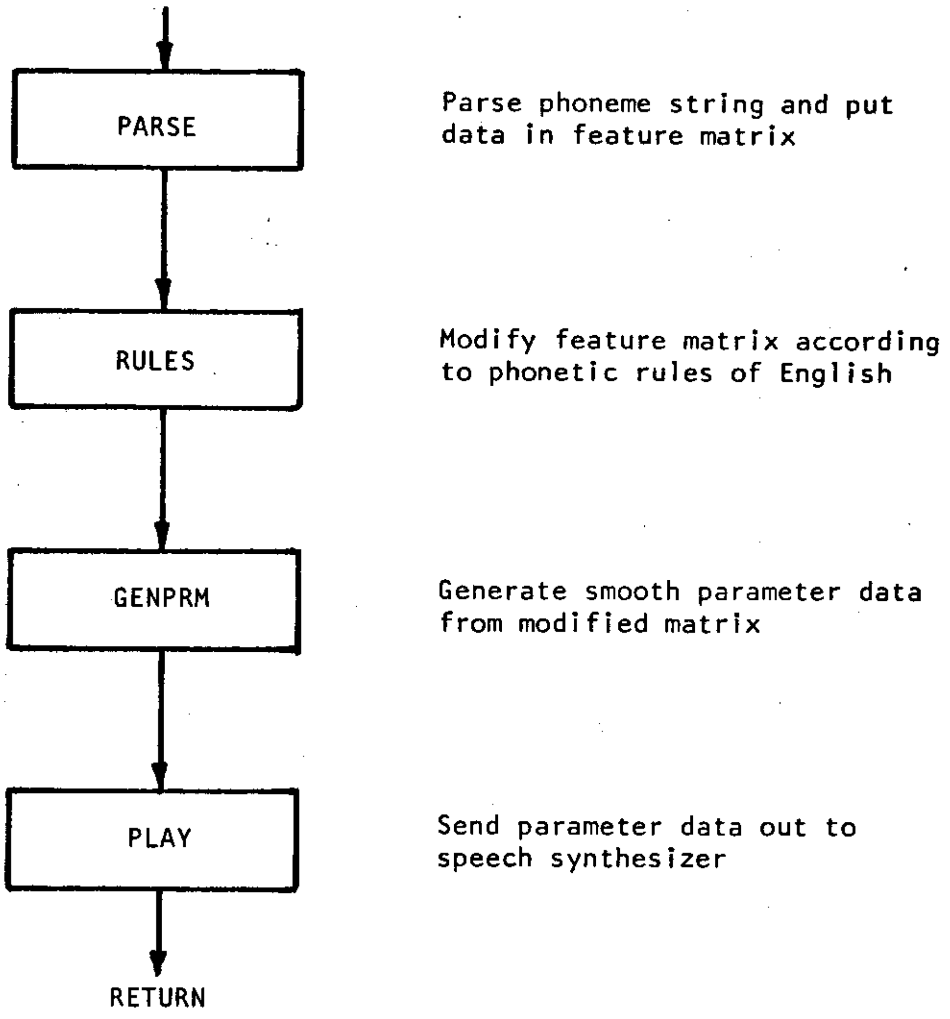
PARSE
Parse phoneme string and put data in feature matrix
RULES
Modify feature matrix according to phonetic rules of English
GENPRM
Generate smooth parameter data from modified matrix
PLAY
Send parameter data out to speech synthesizer
RETURN

Figure 1 Synthesis by Rule system top level flowchart

deleting entire columns. Rule group 1 affects only columns as a whole and in effect operates on the symbolic level prior to feature assignment. Its purpose is to revise certain spelling combinations to make the input form correspond more closely to normal spelling practices. Group 2 contains the actual phonetic readjustment rules. Rules in this group do things like deleting the aspiration of a "p" following an "s", softening a "t" or "d" sound when it comes between 2 vowels, and spreading assigned vowel stresses to adjacent consonants, which affects their durations. Group 3 rules are concerned with assignment and modification of durations to each column of the matrix. Examples are the lengthening of the last vowel of a word and readjustments to normalize the total time between stressed vowels. The third basic step in the synthesis process is the construction of smooth synthesis control parameters from the phonetic values in the feature matrix. Target parameter values for each phonetic segment are obtained from a table. These target values are then smoothed into continuous control sequences, governed by the computed durations and affected by a system of priorities assigned to each phonetic segment. The fourth and final step is the realization of the control parameters as speech output by the synthesizer. That step will not be discussed further in this paper. Reference 5 describes the synthesizer hardware and the phonetic nature of the control parameters.

THE PARSING SYSTEM

The alphabet used for the phonetic input to the rule system is ARPABET, a phonetic alphabet adapted to the ASCII character set. A complete listing of ARPABET with examples of each of the phonetic symbols is given in reference 4. The symbols are coded with either single or double ASCII characters in such a way that an unambiguous parse of a correct string can be done with a fairly simple state transition parser. Most input syntax errors can be detected.

In addition to separating the input string into a sequence of phonetic units, the subroutine PARSE builds up a matrix of phonetic information about the sequence. Each column of this phonetic matrix contains a packet of information about each of the phonetic units in the input. The rows represent 4 different items of information: First is the 6-bit code for the phonetic symbol itself; second is a cluster of 14 bits of feature information obtained from a table of feature values for each phonetic symbol; third is a location for any stress value that was specified in the input; and fourth is a location, unspecified at this time, for the duration value which is to be computed for each unit. The matrix, then, contains 5 bytes of information about each input unit. Since the primary purpose of this matrix is to contain the feature data about the phrase being synthesized, it is often called the feature matrix. At this point, it would be helpful to look more closely at these features.

WCCF - 4

		vowel	front	diphthong	consonant	stop	voiced	plosive	plosive aspirate	fricative	liquid	nasal	dental	boundary	ignore
CBOUND	space													1	1
CPER	.													1	1
CCOMMA	,														1
CQUEST	?													1	1
	+														1
CTERM	#													1	1
	:														1
	IY	1	1				1								
	IH	1	1				1								
	EH	1	1				1								
	AE	1	1				1								
	AA	1	1				1								
	AH	1	1				1								
	AO	1					1								
	OW	1		1			1								
	UH	1					1								
	UW	1		1			1								
	AX	1					1								
	IX	1					1								
	ER	1					1								
	UX	1					1								
	OH	1					1								
	AW	1		1			1								
	AY	1	1	1			1								
	OY	1	1	1			1								
	EY	1	1	1			1								
	RX	1					1								
	EL														
	EM														
	EN														
	Y				1		1								
	YX	1					1								

	vowel	front	diphthong	consonant	stop	voiced	plosive	plosive aspirate	fricative	liquid	nasal	dental	boundary	ignore
W				1		1				1				
WX	1					1								
R				1		1				1				
L				1		1				1				
LX	1					1								
WH				1		1								
M				1	1	1					1			
N				1	1	1					1	1		
NX				1	1	1					1			
P				1	1		1	1						
T				1	1		1	1				1		
K				1	1		1	1						
KX				1	1		1	1						
B				1	1	1	1							
D				1	1	1	1					1		
G				1	1	1	1							
GX				1	1	1	1							
DX				1	1							1		
F				1					1					
TH				1					1			1		
S				1					1			1		
SH				1					1					
V				1		1			1					
DH				1		1			1			1		
Z				1		1			1			1		
ZH				1		1			1					
CH														
JH														
HH				1										
Q				1	1	1								
-														

Table 1 Phonetic Feature values assigned to each phonetic symbol

feat A	vowel	cons	front	diphth		word bound	phrase bound	ignore
feat B	stop	voiced	plosive	plosive aspir.	fricative	liquid	nasal	dental

Table 1 presents a listing of the set of feature values corresponding to each of the phonetic symbols used in the system. These features represent a classification of each symbol according to linguistic or phonetic categories which are important to the pronunciation of that symbol. Identifying the pronunciation categories in this way simplifies the rules needed in the system, since most of the rules operate according to the categories and would have to be restated many times with different units if they operated directly on the symbolic units. Occasionally, tho, it will be more convenient for a rule to test the unit code directly rather than the feature values. For this reason, both are retained in the matrix.

THE RULES

The only complication to be considered in the rule control system is the order of application of the groups and the rules within a group. As the matrix is scanned to test for the conditions satisfying a rule, these conditions may be met or not depending on whether certain other rules have already applied and modified the matrix. In order to guarantee unambiguous application of all rules, a particular order of application must be defined, and that order kept in mind when the rules are devised. The order of application defined here is as follows:

> Within a group of rules, the matrix will be scanned once from left to right. As the scan comes to each column of the matrix, all rules in the group are tested in the order they are stated. The right hand side of each rule is tested by aligning the first "/" mark in that rule just to the left of the matrix column currently being scanned. Whenever a rule condition is met, the matrix is immediately modified. When the application of any rule involves changing the phonetic symbol at any matrix column, then all features in the column are set as defined for the new phonetic unit. Any stress value already assigned is left unchanged.

Since there are 3 separate groups of rules, the matrix will be scanned from left to right 3 times. In many cases, when a rule condition is met at a particular column of the matrix, and that rule has been executed, it is known that certain subsequent rules in the same group cannot apply. In those cases, some execution time can be saved by judicious branching to the best point from which to continue testing.

RULE GROUP 1

Of the three groups of rules in the system, the first group has the least dependence on the feature categories. Several of the rules in this group could in fact operate on the level of units before the features have been assigned. These are spelling convention rules. Six "dummy" symbols are defined as valid input units, which are changed into the correct phonetic sequences by these rules.

WCCF-6

done in LOOKUP

	Ø ⇐ . OR , OR ?/~~SPACE~~/
~~R1A~~	~~⇐ /,/~~
~~R1B~~:	AX,L ⇐ /EL/
~~R1C~~:	AX,M ⇐ /EM/
~~R1D~~:	AX,N ⇐ /EN/
~~R1E~~:	T,SH ⇐ /CH/
~~R1A~~:	D,ZH ⇐ /JH/

In reading these rules, A ⇐ /B/ indicates that if unit B is found in the unit sequence it is replaced by A. Note that in 5 of the 6 rules just listed, this involves the insertion of a new column into the matrix. When a new matrix column is inserted, the new feature values must of course be looked up and entered into the matrix as well as the unit itself. The "/" marks are used to bound the portion of the matrix currently open for modification.

The next rule in this group specifies that a glottal stop is to be inserted before a stressed vowel at the beginning of a word if the previous word also ends with a stressed vowel.

R1A	~~R1G:~~	Q ⇐ vowel stress>Ø,wordbound//vowel stress>Ø

In all rule specifications, capital letters indicate phonetic segments, while lower case letters indicate features or stress specifications. Commas or "/" marks separate references to different matrix columns. If the situation described by this rule is found in the matrix, a glottal stop and its features are inserted as a new column at the position indicated by the adjacent "/" marks in the rule. It is of interest to consider the large number of rules that would be needed to state this modification if they were expressed in terms of phonetic symbols rather than the feature "vowel".

The next 2 rules change the pronunciation of R and L when they occur after a vowel.

R1B	~~R1H:~~	RX ⇐ vowel/R/
R1C	~~R1I:~~	LX ⇐ vowel/L/

The final pair of rules in this group adds an end-glide to any diphthong depending on whether the diphthong is marked by the feature "front".

R1D	~~R1J:~~	YX ⇐ diphth front//
R1E	~~R1K:~~	WX ⇐ diphth -front//

These rules contain examples of combinations of features to be located in a matrix column. This can be interpreted as a bit masking problem, observing the bit polarities of the various features.

RULE GROUP 2

The rules in group 2 perform the major phonetic modifications which are necessary for the sound structure of English. The first 5 rules in this group set consonant stresses for certain consonant cluster patterns as a means of controlling the durations of these consonants. A scheme of marking negative stress values is used to communicate additional information to the duration rules in the next group.

```
                         stress≥Ø
                              ↓            ≠
R2A:       stress=1 ⇐ /cons/vowel stress =Ø  vowel
R2B:     stress=-1,-1 ⇐ /S,plos -voice/stress≠Ø
R2C:     stress=-1,-1 ⇐ /plos or (fric -voice),liquid or nasal
                                            /vowel stress≠Ø
R2D:   stress=-1,-1,-1 ⇐ /S,plos -voice,liquid/vowel stress≠Ø
R2E:     stress=-1,-1 ⇐ /T or D,SH or ZH/vowel stress≠Ø
```

Several of these rules contain multiple units within the slash marks, indicating that some modification is to occur at each of the enclosed columns providing all conditions are met. The parentheses here indicate logical operator priorities rather than optional units as in linguistic notational conventions.

Four more rules in this group modify the pronunciation of certain phonemes depending on the consonant or vowel environment. A T or D occurring between two vowels is softened to the phonetic unit known as a flap. This is a single, quick tap of the tongue indicated by the symbol DX. The following vowel must be unstressed unless it is at the beginning of the next word. The vowel UW is raised to UX when it follows a consonant in which the tongue is moved up to touch just behind the teeth. This has the effect of lowering formant 1 and raising formant 2 corresponding to a higher tongue position during the vowel. K and G are modified to alternate forms which have lower formant 2 and formant 3 when they are followed by a back vowel where the tongue must move to the back of the mouth.

```
R2F:       DX ⇐ vowel/T or D/~~optional~~(wordbound,vowel) or
                                       vowel·stress=Ø
R2G:       UX ⇐ dental/UW/
R2H:       KX ⇐ /K/vowel -front
R2I:       GX ⇐ /G/vowel -front
```

The next 2 rules in group 2 account for the fact that voiceless aspirated stop consonants lose their aspiration either when preceeded by an S or when they come at the end of a word. This is accomplished by changing them to their voiced counterparts, which do not have aspiration by definition.

```
R2J:       add 4 to phonetic unit code⇐ S/plos -voice/
R2K:       add 4 to phonetic unit code⇐ /plos -voice/wordbound
```

The phonetic unit codes are assigned so that the voiced unaspirated stops are located just after the voiceless stops which are detected by these rules. As with all rules, the features are changed along with the phonetic code.

The final cluster of 3 rules in group 2 has a similar effect to the previous pair, except that here only the plosion and aspiration features are changed instead of changing the phonetic unit as a whole. This is the first case encountered where a rule only changes a few features and not the whole unit. Later on, when the features are being interpreted to generate the synthesizer control parameter data, lack of the plosive feature will result in a shorter time being allocated to the aspiration portion of the consonant, while the aspiration feature will determine how much hiss sound is produced during that interval. These rules have the following effect: first, deleting both plosion and aspiration when the stop is followed by a second stop, whether or not the following stop is at the beginning of the next word; second, deleting the aspiration when the stop is followed by WH or HH, whether or not the WH or HH is at the beginning of the next word; and third, deleting the aspiration when the stop is not at the beginning of a word and is followed by an unstressed vowel.

```
R2L:    -plos -plosa ⇚ /plos/optional wordbound,stop cons
R2M:        -plosa ⇚ /plosa/optional wordbound,aspirant
R2N:        -plosa ⇚  -(wordbound or silence)/plosa/vowel -strs
```

RULE GROUP 3

The rules in group 3 are entirely concerned with the computation of sentence timing structure. Concerning intelligibility of the output, this is without doubt the most important part of the entire system, altho the pitch control parameter (FØ), generated in the next step, also has an important role in the perceived quality. The initial (or target) timing structure is determined by a dual table of duration values. Each phonetic symbol has associated both a long and short duration depending on whether that unit is stressed or unstressed. One of these duration values is inserted into each matrix column depending on the stress value currently stored at that column. This initial duration assignment is considered to be a part of group 3 altho it is not a rule in the usual sense.

The first rule of group 3 is concerned with the tendency in English pronunciation to meter out the timing of unstressed syllables so that stressed syllables occur at roughly equal intervals. In other words the durations are adjusted so that the major temporal pattern moves from stress point to stress point.

rule does not do what this description says. this whole area is still unsatisfactorily handled.

R3A: between wordbound or . or silence change the duration of each stressed vowel by (2*NV+3)/(5*NV) where NV is the number of vowels in between

The final vowel of a ~~word~~ phrase and all units from there to the end of the ~~word~~ phrase are lengthened.

R3B: dur*1.5 ⇚ /last vowel of a ~~word~~ phrase, ... /~~word~~phrasebound

Vowel duration is changed if followed by an unvoiced plosive, a voiced fricative, or by RX or LX followed by a consonant.

```
R3C:     dur*.6 ⇐ /vowel/plos -voice
R3D:     dur*1.25 ⇐ /vowel/fric voice
R3E:     dur*.5 ⇐ /vowel/RX or LX,cons
```

The duration of W, R or L, with a negative stress value and followed by a vowel is lengthened by 20 msec if it is preceeded by an unvoiced plosive and the length is set to 90 msec if it is preceeded by an S.

```
R3G:     dur+20msec ⇐ plos -voice/(W or R or L) stress<Ø/
                                                  vowel
R3F:     dur=90 msec ⇐ S/(W or R or L) stress<Ø/vowel
```

All consonants with negative stress values are shortened.

```
R3H:     dur*.8 ⇐ /cons stress<Ø/
```

The durations of certain stop,fricative combinations are fixed.

```
R3I:     dur=70,60 msec ⇐ /T stress>Ø,SH stress>Ø/
R3J:     dur=70,50 msec ⇐ /D stress>Ø,ZH stress>Ø/
R3K:     dur=60,40 msec ⇐ /T stress=Ø,SH stress=Ø/
R3L:     dur=40,30 msec ⇐ /D stress=Ø,ZH stress=Ø/
```

The durations of certain nasal,plosive combinations are fixed.

```
R3M:     dur=30,30 msec ⇐ /(M,P or B) or (N,T or D) or
                                   (NX,K or KX or G or GX)/
```

The durations of adjacent plosives are cut in half, even if a word boundary comes between them.

```
R3N:     dur of plosives*.5 ⇐ /plos/optional wordbound/plos/
```

It is difficult (no doubt impossible) to state a complex rule without an equally complex rule description language. In order to present the rules here without having to define an elaborate notational system, I have taken the liberty of expressing some things in English. In most of the rules which are presented here as clusters, there is some degree of commonality which should be used to advantage in writing code to realize these rules. I leave it as a problem for the interested reader to express these sets of rules most efficiently in flowchart form.

Up to this point in processing the speech sequence, the entire string had to be stored and processed as a whole because of the nature of the rule scan. These steps of building up, testing, and modifying the matrix need not occupy much memory space and should run fairly quickly. By the time this paper appears in print, the actual memory space and ~~memory~~ running time requirements for the 8080 version should be available.

GENERATING THE SYNTHESIZER CONTROLS

We will now discuss in somewhat less detail the mechanisms used to generate the synthesizer control parameters. The first control function to be produced is the pitch parameter (FØ). This is the only parameter which requires any scanning back and forth in the matrix; all the rest can be generated strictly from left to right. For this reason, the FØ parameter is handled separately from the rest, and is computed first. Once that parameter is completed and stored in the output buffer, and the remaining parameters have been computed as far as the first phonetic unit, it is at least theoretically possible to begin playing the speech data out to a hardware synthesizer using an interrupt-controlled update rate. Whether or not this is possible in reality depends on the processing speed at which the remaining parameters are being computed.

GENERATING THE FØ PARAMETER

The plan followed in this model of the FØ function is based on a few simple rules. The underlying principle is that each time the amplitude of voicing parameter (AV) is switched on by a voiced segment of speech, the FØ level starts off at a mid-range value of 120 Hz. For as long as the AV control remains on, the pitch drifts slowly downward in an exponential decay fashion toward 100 Hz. Superimposed on this basic downdrift pattern are "intrusions" which raise the FØ level in smooth upward arcs during all stressed vowels. The degree of pitch raising depends on the stress value up to maximum of 160 Hz at the peak of the arc for a vowel with stress=1. In addition, near the end of a phrase, a final rising, level or falling contour is added depending on the punctuation mark used to end the sequence, whether question mark, comma or period, respectively. A new end-point level is determined by adding 40 Hz for a question mark or subtracting 40 Hz for a period. A linear ramp is then constructed from the previously computed FØ level at a point 300 msec before the end of voicing to the newly determined end-point level. This linear ramp is then added to the original FØ curve. This is shown in figure 2.

It seems worthwhile to consider at this point the effect on the listener of different models for FØ. In human speech, the pitch contour is used to communicate a very wide variety of emotional and physical facts in addition to certain linguistic information about the stress structure of the utterance. Since the software system has no access to any of these esoteric,

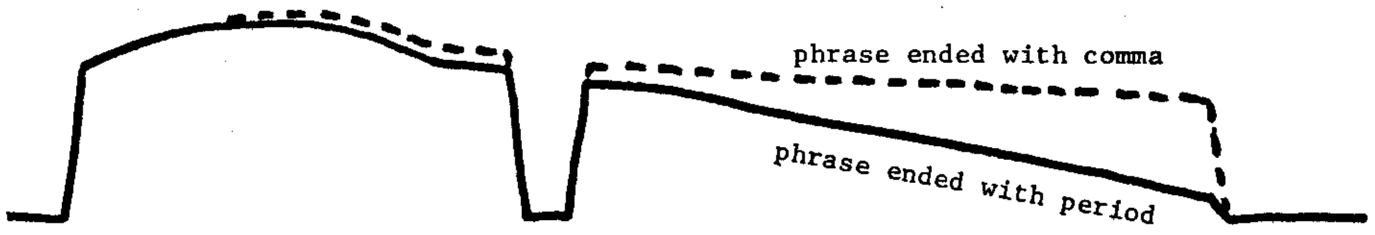

Figure 2   End-of-phrase pitch parameter (FØ) contours

external factors (even if we did know how to model the effects) the speech output must necessarily be rather bland in that respect. The tendency seems to be for linguistics researchers to overestimate the influence of the linguistically related variables such as stress patterns in trying to account for the wide range of variations caused by external, non-linguistic facts. In a synthesis by rule system, the influence of stress values on the FØ parameter can easily be modified by changing the frequencies mentioned in the last paragraph. Reducing the stress peak frequencies would give the output less of a sing-song quality. Alternatively, if so desired, the pitch output could be made completely flat by skipping this FØ calculation entirely and putting a constant value into the FØ parameter.

GENERATING THE OTHER PARAMETERS

The most essential of the remaining synthesizer controls are, by far, the three formant frequency controls (F1, F2 and F3). These must be relatively smooth curves during the entire phrase, at least with no abrupt jumps while the amplitude of voicing (AV) is turned on. These controls also bear the main load of determining the correct vowel quality at each instant during the utterance when there is voicing.

In order to achieve efficient coding within 8 bits and for minimum hardware complexity in the Computalker model CT-1 Synthesizer, the formant information on parameters F1, F2 and F3 is coded not as frequencies in hz, but as a reciprocal log function of frequency. Then, to eliminate complications of extra conversions, all formant information throughout the synthesis by rule system is coded using this reciprocal relationship. Keep in mind when reading graphs such as figures 3 and 4 that the formant scale is inverted from the usual formant frequency plots. Also, all three parameters are plotted overlaid on the same scale to save space. This convention requires some re-learning for those of us accustomed to reading plots of formant frequency vs. time, but it seems advisable to plot the parameters just as they are actually used in the system. In spite of this, I still think of formants as frequencies and referred to frequency scales in the preceeding discussions of rule behavior.

The basic pattern for the formant frequency parameters is determined by a table of formant settings for each of the phonetic units in the system. Values from this table are taken as targets to be approached during the duration of each phonetic segment. This sequence of target values is then smoothed across each segment boundary depending on a system of priorities assigned to each segment. Each segment is assigned an initial time constant and a terminal time constant, which, together with the relative priority of each, determine how it interacts with its neighboring segments. The neighbor with the greater priority imposes its time constants and boundary crossing points on the transition. In most cases, the higher ranking units actually have the weakest conditions, thus guaranteeing that they will be "overtaken" when they occur next to a lower priority segment. As an example of this, HH has a high priority, but no control over the boundary crossing value. As a result, when HH occurs next to any vowel, which would have a lower priority, the HH determines that the vowel will have full control over the transitions. This effect can be seen in the graphs in figure 3.

Once the formant parameters are computed for the duration of a segment, it remains only to fill in the various amplitude control parameters (AV, AH, AF and AN) and the fricative frequency setting (FF). All of these parameters are controlled more or less directly by bits set in the feature matrix. For example, when a consonant is encountered with the plosive feature set, a pulse is set up on the AH control. The time of release of the plosive burst is computed depending on the plosive aspirate feature. Provided the following consonant is not also a stop, this pulse is then stored at the computed release time. Voicing and nasal amplitudes simply move thru smooth arcs to predefined levels if the corresponding feature bits are set.

HOW GOOD IS IT?

As an illustration of the parameter output of a rule system of the sort described, figures 3 and 4 present sets of parameters generated by the program and recovered from actual human speech, respectively. The examples compare the two versions of the word "hello" as produced by these two methods. In the synthesized version in figure 3, the phonetic segments are marked with ASCII characters just as used to generate the word. The input to the program was the string "HHEH2LOW," which was followed by another phrase. In the human version in figure 4, the segments are marked using the International Phonetic Alphabet (IPA) symbols as well as the normal English spelling. The human version was also spoken as a part of a longer phrase.

While there is quite a lot of difference in detail, the overall shapes are reasonably similar. The major difference one sees immediately is in the timing structure. While the overall length of the human version is 130 msec shorter, the length of the OW vowel is the same within 10 msec. The first 3 segments make up this difference, particularly the ɛ, which is barely more than ⅓ the length of the computer generated EH. The primary reason for this is the 2 stress specified on the EH vowel. This was necessary to get even the small rise in pitch seen during that vowel. The generated FØ curve during the remainder of the word is disappointing to say the least, and shows nothing like the expressive contour seen in the human version. The differences in F3 during the L are probably not significant, altho it's interesting to note that the synthetic version accurately follows the "textbook" pattern for an L, while the human version shows the ℓ quality only in F1 and F2. Of greater interest is the dip in voicing amplitude during the human ℓ. No doubt the perception of an "ℓ" would be improved if the rule system generated such a dip
during an L sound.

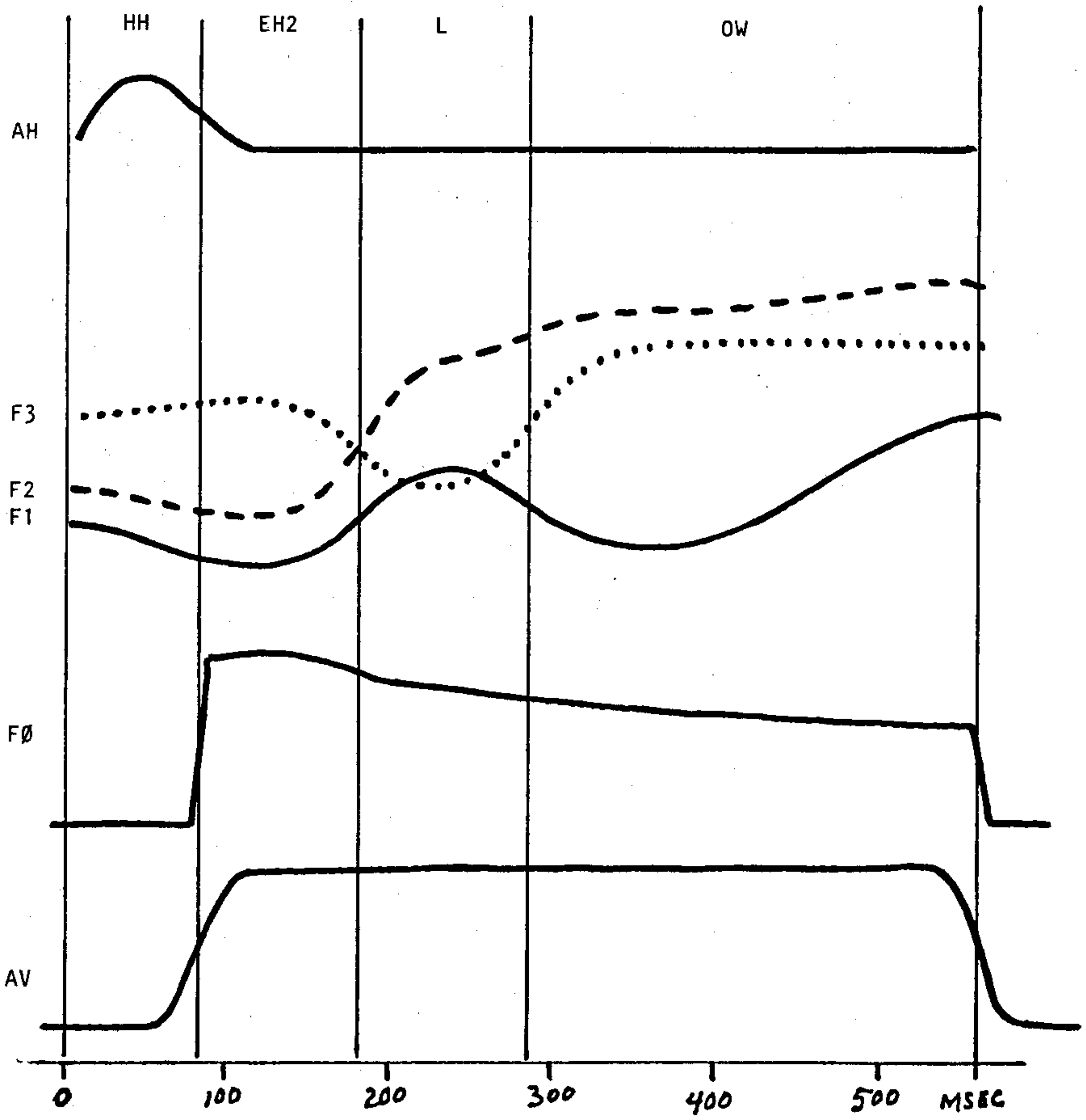

Figure 3 Synthesis parameters generated by Rule

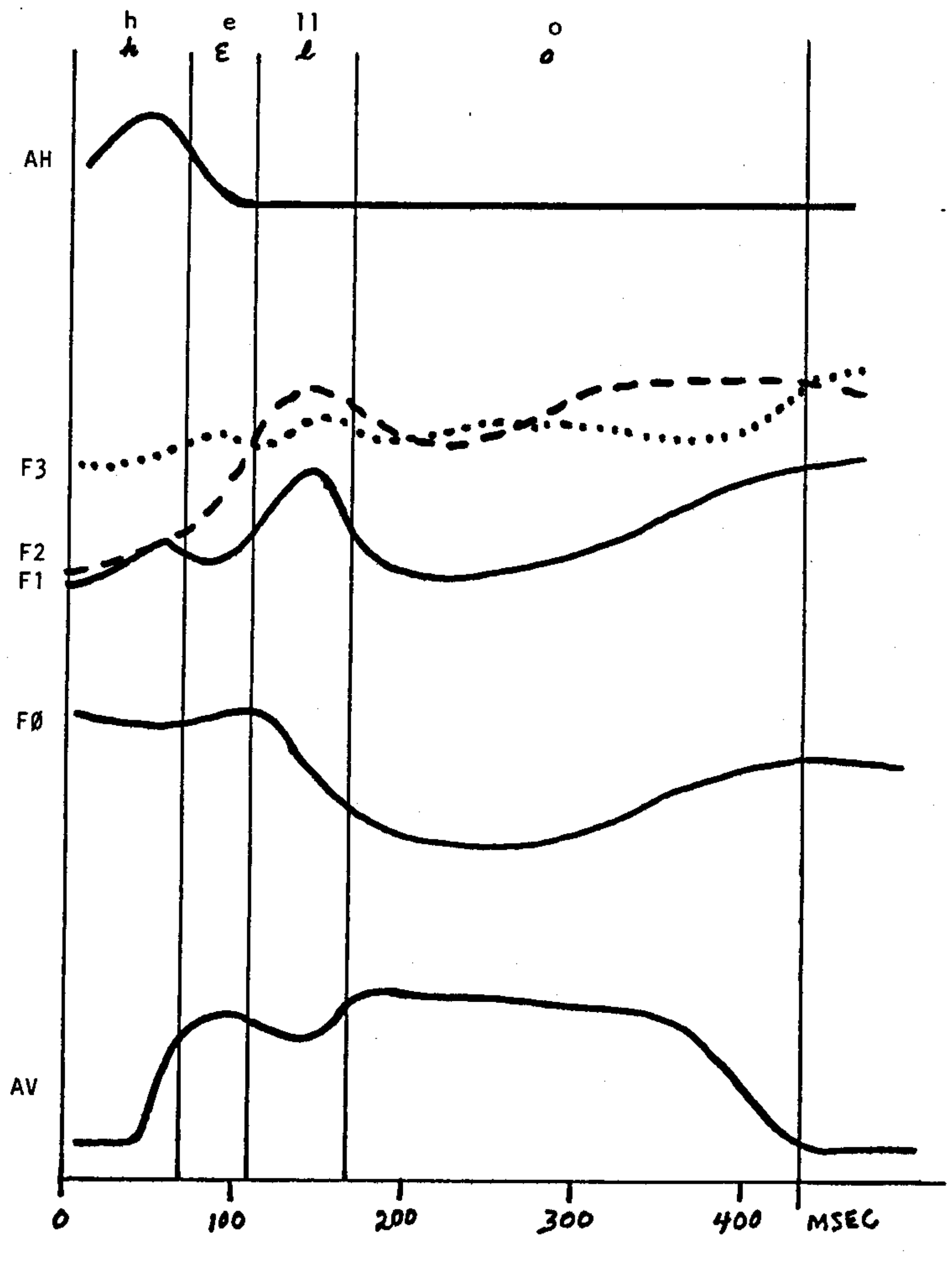

Figure 4 Parameters extracted from human speech "hello"

REFERENCES

1. R. Carlson and B. Granstrom, "A phonetically-oriented programming language for rule description of speech", Proc. Speech Communication Seminar, Stockholm, Sweden, August 1-3, 1974

2. D.H.Klatt, "The linguistic uses of segmental duration in English: Acoustic and perceptual evidence", J. Acoust. Soc. Amer.,vol 59(May 1976)pp. 1208-1221

3. D.H.Klatt, "Structure of a Phonological Component for a Synthesis-by-Rule Program", IEEE Trans on ASSP, vol ASSP-24(Oct 76)pp. 391-398

4. D.L. Rice,"Hardware and Software for Speech Synthesis", Dr. Dobbs J. of Computer Calisthenics & Orthodontia, vol 1, no 4 (April 76)

5. D.L. Rice, "Friends, Humans and Countryrobots: Lend me your Ears", BYTE no. 12 (Aug 76)

BIOGRAPHY

Mr. Rice holds a B.A. in Linguistics from UCLA (1969). He has been involved in research in acoustic phonetics for the past nine years, which included the opportunity to work in Sweden and Finland during part of 1974. He is currently a partner of Computalker Consultants in Santa Monica, CA, involved with the design of speech synthesis circuitry and related software. His interests include speech production, digital signal processing, microcomputers, and occasionally such pursuits as sailing and SCUBA diving.