CONCATENATIVE SPEECH SYNTHESIS USING SUB-PHONEME SEGMENTS

R.A.Sharman

Speech Research Group, IBM(UK) Labs, Hursley Park, Winchester SO23 9DR, UK

Abstract

This paper discusses a method of acoustic waveform synthesis, for use in a text-to-speech system, which employs the concatenation of a very large number of very small acoustic units. Such sub-phoneme sized audio segments, called *wavelets*, can be individually spectrally analysed and labelled as *fenones*. (Fenones are familiar from speech recognition methods, where they are clustered into logically related groups, called *fenemes*, sequences of which can be matched with individual *phonemes*, and hence *words*.) In the text-to-speech case the required phonemes are known from prior linguistic analysis of the input words in the text. Suitable sequences of fenemes can be predicted for each phoneme in its own context using standard *hidden markov modelling* techniques. A complete output waveform can be constructed by simply concatenating a very long sequence of wavelets, each corresponding to its respective feneme. The design and use of this approach to waveform generation is discussed in the context of a practical text-to-speech system. The advantages of using a feneme set extracted from a training script read by a single human speaker is that it might be possible to generate natural sounding speech, using a finite sized codebook.

1. INTRODUCTION

The objective of a text-to-speech (TTS) system is to convert a sequence of one or more *words* into an acoustic *waveform* which will be as acceptable to a human listener as if the words had been spoken naturally. Automatic synthesis of waveforms from text is required when the contents of an intended message are not known in advance, and so pre-recording the waveform is not possible. This can occur, for example, when electronic mail messages must be read over the telephone to a remote user. Many methods of converting text to speech have been proposed[7] and a wide variety of implementation techniques studied[2]. The standard methods[1] have been shown to have high intelligibility[12] using both rhyme and comprehension tests. However, most existing techniques, and thus also commercially available systems, produce sound which is less natural, and usually less acceptable than, human speech. In some applications, such as human orientated computer interaction, noisy, or stressful situations, the requirement for more natural sounding speech is thought to be important.

A favoured method of creating speech output is to use samples of speech taken from a recorded human voice. Since only a finite amount of recorded material can be obtained, it is thus necessary to segment and re-assemble the actual recordings to create the illusion of a new utterance. The most common units of segmentation used have been phonemes[11], diphones[3] and demi-syllables[9]. Waveform synthesis by the concatenation of segments of naturally spoken speech has the potential to improve over other methods of speech synthesis, such as formant-based methods[1], because it has the ability to precisely model the speech characteristics of a given human speaker and so achieve a more natural speech quality. Additionally, if the process of capturing a speaker's segmental and prosodic characteristics could be made sufficiently straightforward, it would assist in the rapid creation of new "voices" for a Text-to-Speech (TTS) system[6]. It might even enable a system to be trained on a given speaker or customised to perform in particular speaking styles or dialects.

The principal difficulty with concatenative synthesis is the decision of exactly what type of segment to select. Long phrases exactly reproduce the actual utterance originally spoken, and are widely used in Interactive-Voice-Response (IVR) systems. Such segments are very difficult to modify or extend for even quite trivial changes in the text. Phoneme sized segments can be extracted from aligned phonetic-acoustic data sequences, but simple phonemes alone cannot generally model the difficult transition periods between the more steady state central sections, leading to unnatural sounding speech . Diphone and demi-syllable segments have therefore been popular for text-to-speech systems precisely because they do attempt to capture these transition regions, and can conveniently yield locally intelligible acoustic waveforms.

An additional problem with the concatenation of phoneme-sized, or larger, units is the need to modify each segment according to the precise prosodic requirements of the context it is intended for. Some approaches have used an LPC representation of the audio signal so that its pitch can be easily modified[14]. Other approaches have used the pitch-synchronous-overlap-and-add (PSOLA) technique[4] to enable both pitch and duration to be modified for each segment of the complete output waveform[6]. Both of these approaches introduce the possibility of degrading the quality of the output waveform, for example by introducing perceptual effects related to the excitation chosen, in the LPC case, or other unwanted noise due to accidental discontinuities between segments, in the PSOLA case. A method of waveform generation which does not require such modification, or keeps it to a minimum, would therefore be advantageous.

In most concatenative synthesis systems the determination of the actual segments for the given voice is also a significant problem. If the segments are determined by hand the process is slow and tedious. If the segments are determined automatically they may contain errors which will degrade the voice quality. While automatic segmentation can be done with little or no operator intervention, for example by using a speech recognition engine in its phoneme recognising mode[8], the quality of segmentation at the phonetic level may not be adequate to isolate good units. In this case some hand tuning would still seem to be needed.

Consequently, this study examines an alternative idea for waveform synthesis: namely the concatenation of very small, sub-phoneme sized, units. These units are well-known from speech recognition techniques, and if correctly selected and concatenated into very long sequences, could form complete waveforms which could be acceptable as intelligible, natural sounding speech. The objective of this approach is to create a computationally viable method of capturing information about a particular speaker which will lead to an automatic method of producing intelligible, natural sounding speech, in short a *speaker dependent speech synthesis* technique. The principle of the method is outlined below, and a procedure for building such a system is described. The system has been prototyped and initial results suggest that the output can be acceptable. Some of the outstanding problems with the method are discussed.

2. THE DEFINITION AND DETERMINATION OF SUITABLE SUB-PHONEME SEGMENTS

Let an observed speech waveform, S, be denoted by a sequence of digital samples assuming some sampling rate suitable for establishing enough bandwidth to capture all the relevant frequencies, or $S = s_0, s_1, s_2, \ldots s_n$. For example, at a sampling rate of 11.025 Khz, as used in standard multimedia audio adapters, there will be 11025 samples per second, and (n/11025) is the length of S in seconds. Now assume that consecutive groups of m samples can be identified and labelled. There is no requirement for each group to be the same length, although a fixed length is usually chosen for speech recognition work. In this case arbitrary length groups are assumed. Each group of samples denotes a unique waveform segment, called a *wavelet*, and can be denoted by w_i. The waveform can be considered to be a sequence of adjacent, non-overlapping segments, or $S = W = w_0, w_1, \ldots, w_m$. Each wavelet can be uniquely labelled with an index number, I_i, giving a sequence of labels $L = l_0, l_1, \ldots, l_m$ which can be called *fenones*. A fenone has an associated spectrum obtained, for example, by using the fast fourier transform to obtain a vector of discrete fourier transform coefficients for the corresponding wavelet.

Now suppose a new sequence of fenones, $L' = l'_0, l'_1, \ldots, l'_m$, is constructed according to some principle. Clearly the corresponding waveform can be generated by selecting the wavelet, w'_i, which corresponds to each new fenone, l'_i, for each i=0,1,...,m. In principle the selection can be done by a simple lookup table, and the final waveform is obtained by concatenating the corresponding wavelets, giving $S' = W' = w'_0, w'_1, \ldots, w'_m$. In practice, discontinuities at wavelet boundaries can be smoothed by applying a window, say a Hanning window, to a larger section of the waveform, and performing pitch synchronous overlap-and-add of adjacent wavelets as in the PSOLA technique[4]. Since each wavelet may contain inherent traces of the fundamental frequency at which it was originally uttered, great care would need to be exercised in choosing the precise sequence of fenones so that realistic pitch movements are perceived in the final waveform. For that reason the following discussion will largely ignore the problem of correct pitch assignment for the sake of clarity. It should be noted that the pitch problem can, of course, be solved by either selecting the correct fenone from a set of similar fenones differing only by their pitch, or by modifying a single fenone by a standard LPC or PSOLA waveform modification technique.

The main drawback of the simple method just described is that a very large inventory of fenones would be required, even given the fact that identical ones (where they exist in the training corpus) can be coalesced. Because truly identical fenones do not often occur (except in silence or noise) there could be the order of a hundred thousand fenones in a typical training corpus of about 1 hour of speech, given the parameters specified above for sample rate and fenone rate. Another difficulty would be the choice of a strategy for deciding which fenone to select in order to build the desired output sequence. As such, the choice of fenone would represent an enormous search problem.

In order to simplify the search problem, and reduce the number of items which can be selected, the same strategy is proposed here that is commonly used in speech recognition systems[5], namely to cluster[10] the fenones into an equivalent set of labels, which may be called *fenemes*. Fenemes can be considered to be unique numbers, but it is also useful to associate mnemonic text labels for ease of understanding. Thus, for example, fenone number 126 might belong to the cluster denoted by feneme 35 which could be labelled AE1_2, indicating a portion of an AE1 phoneme.

The feneme is thus considered to be a generic sub-phoneme unit, used in potentially many different contexts, and which is typically of the order of a few milliseconds. The lower limit on the size of a feneme is a single fundamental frequency wave epoch, and is determined in part by the base pitch of the speaker being modeled. It is a basic assumption of this approach that there exists a useful clustering of fenones which will both significantly reduce the number of fenemes, and yet not introduce any unwanted perceptual effects which could degrade the output waveform. Of course in the limit, when the number of fenemes is equal to the number of fenones, a perfect waveform could conceptually be constructed (at least for utterances similar to the training corpus). A useful clustering of the fenones for speaker-dependent speech recognition might yield as few as 1000 fenemes, but that would be unlikely to be sufficient for a general purpose text-to-speech system. The determination of the best feneme set, or *codebook*, is still a matter of investigation, but it seems plausible that a size of the order of magnitude of 10,000 might suffice.

The fenones can be clustered into a set of fenemes by standard methods of vector quantisation, for example by the use of the k-means algorithm[10] to cluster wavelets with similar features, creating a codebook of labels of a fixed size. The fenemes are then said to be *trained* on the observed corpus. Each feneme represents the centroid of a group of fenones, and can thus be associated with a wavelet by either choosing one of the fenones in the set (perhaps the nearest) or simply by choosing an arbitrary member of the fenone set (for example the first, or the seed value for the given cluster).

3. CONSTRUCTION OF WAVEFORMS FROM SUB-PHONEME SEGMENTS

Assuming that a good set of fenemes can be generated, the problem remains to predict the sequence of fenemes required to achieve a particular output utterance. The method suggested here, and prototyped in a text-to-speech system, is to use conventional language modelling techniques[13] to predict the desired feneme sequence, given the required sequence of phonemes specified by the initial linguistic processing of the text-to-speech system. This can be done, for example, by using a *n-gram hidden markov model*(HMM) trained on the original speaker dependent speech corpus. The equation to be solved must relate the observed phonemes in terms of an underlying (hidden) feneme sequence. This is the maximisation of the conditional probability of a model producing the observed sequence of phonemes, $F = f_1, f_2, \ldots, f_n$, given any feneme label sequence, $L = l_1, l_2, \ldots, l_n$, or $\max_L\left[P(F|L)\right] = \max_L\left[P(L|F)P(L) / P(F)\right]$, which amounts to finding the $\max_L\left[P(L|F)P(L)\right]$. This model can be initialised by extracting counts from a previously aligned corpus of phoneme and feneme sequences. The model can be further trained by standard HMM modelling techniques to create a general purpose model for predicting fenemes from phonemes.

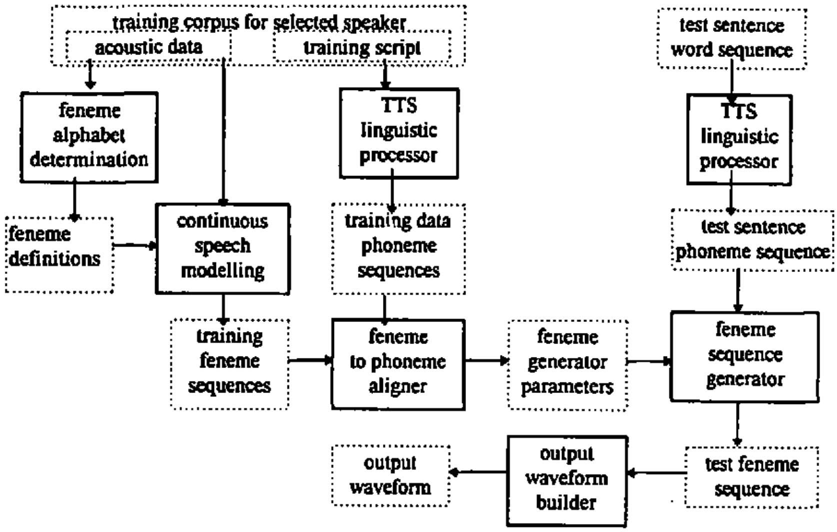
First, the training corpus is analysed to determine the equivalent feneme sequences in the way described above, and then these feneme sequences are aligned with the known phoneme sequences using the Forward-Backward algorithm to train a suitably chosen HMM. This is standard practice in speech recognition methods[5] to obtain a mechanism for labelling a speech corpus automatically with the corresponding phonemes. The process is not entirely error-free, but can be shown to be surprisingly accurate. Now that an HMM exists trained to model phoneme-to-feneme mappings, it is a simple matter to use the HMM generatively to produce typical feneme sequences when given an arbitrary phoneme input sequence. A further constraint on the model is the expected duration of the phoneme output; that is, the number of fenemes which must be processed in order to output a single phoneme. Without this constraint the model could only be expected to output rather short sequences.

The order of the phoneme output n-gram model needs to be considerably higher that the n=3 which is typical for language modelling in speech recognition, but since the parameter tables are usually sparse this does not present a great problem.

Finally, once a feneme sequence has been constructed, the actual output waveform can be generated in the way indicated above. If the wavelets are encoded as LPC coefficients they can be re-synthesised at different pitch values. Alternatively, direct waveform concatenation and modification by PSOLA methods, while more expensive in storage requirements, can be shown to be very efficient computationally if all the operations are carried out in the time-domain.

4. COMPONENTS OF SYNTHESIS SYSTEM

The method proposed in this study can be visualised as a working system by reference to the following diagram. The processing components of the diagram are described in more detail below. (In the diagram data items are denoted by boxes with dashed outlines, and processing algorithms are denoted by boxes with solid outlines. Arrows indicate the movement of data.)



A block diagram of the system components required for speaker-dependent speech synthesis

The processing components of the system pictured above are described in more detail here:

1. **Feneme alphabet determination**: The fenone size is chosen, typically of the order of 10ms, as for speech recognition. A training script read by a speaker is labelled with FFT coefficient vectors for each actual fenone. These vectors are clustered for the given speaker, and quantised to a finite set of fenemes which are unique to the speaker. The size of the feneme alphabet is important in determining the degree to which the fenemes will generalize.

2. **Continuous speech modelling:** The equivalent feneme sequences for the fenones observed in the training corpus are generated, using the fenone clusters determined.

3. **TTS linguistic processor**: The words of the training script are converted to phoneme sequences, using the linguistic processing stage of the text-to-speech system itself.

4. **Feneme to phoneme aligner:** The phonemes of the training script are aligned with the feneme sequences of the observed speech corpus by the viterbi alignment of a hidden markov model which has been trained by the forward-backward algorithm on this task. Note that this is very similar to the initial stages of analysis in some speech recognition systems. It is clearly essential to have a training script which is large enough to contain a sufficient number of examples of all common phonetic contexts so that the subsequent TTS synthesis will have a good coverage.

5. **Feneme sequence generator**: A model of phoneme to feneme conversion can be created using the training data generated in the previous step. It is worth noting one significant departure from general language modeling practice here, which is that no attempt to "smooth" the model should be made here, since, unlike normal n-gram modeling, it is required to generate actual fenemes sequences and not recognize unobserved new sequences.

6. **Output Waveform Build**: The expected feneme sequence for a given phoneme sequence is converted into an actual waveform by one of a variety of methods. The preferred method adopted here is to use direct concatenation of actual segments in the time domain by PSOLA techniques. It is at this stage that the desired pitch modification can take place. It should be noted that the durational modelling is automatically taken care of within the feneme sequence generation, although the more general conversion appropriate to overall speaking rate can be performed by PSOLA methods as well.

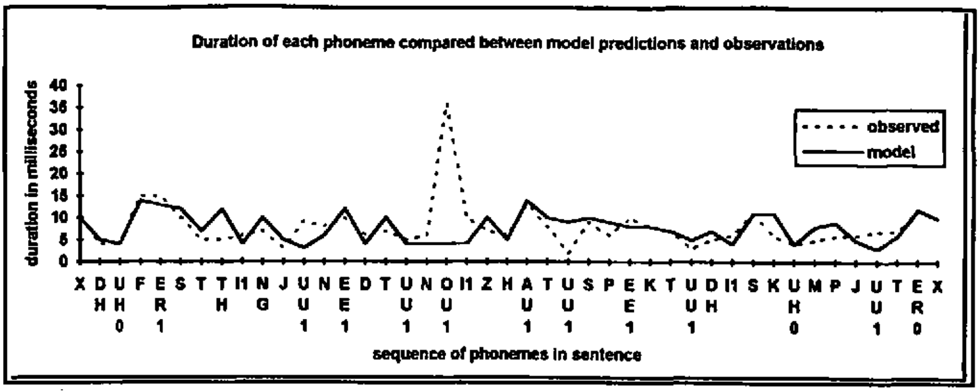
It will be noted that, in principle, the stages of processing described can be completely automated so that the steps could be repeated on any number of training corpus examples. Thus the process could be adapted at will to new speakers, or even, with suitable modifications, to new languages.

5. AN EXPERIMENT IN USING THE APPROACH

A corpus of 150 sentences of English was recorded from a single speaker, comprising about one hour of audio recording at 11khz. The sentences were divided into 100 training and 50 test sentences. Approximately 350,000 fenones were clustered into 320 fenemes, and each feneme labelled according to its typical occurrence in one of four positions for each corresponding phoneme, there being 80 phonemes (stressed and unstressed vowels being distinguished). The training sentences were generated in their fenemic form, and the phonemes and fenemes aligned by a hidden markov model. A feneme generator model was constructed as the inverse hidden markov model using the aligned data as training data.

For each of the test sentences a phoneme string, with associated duration and pitch values was generated, using language modelling techniques also derived from the same speaker. The accuracy of the generated duration values can be seen from the following example taken from the test set, which compares the calculated values to the actual ones. The major discrepancy in the example shown is the error in the central portion where a phrase final lengthening is not correctly modelled. The durational model is a closer approximation to the observed values where the local context gives a good indication of the likely segment duration, as would be expected from the type of modelling being undertaken. If higher level constraints, such as the grammatical structure of the sentence, were to be taken into account, then it might be possible to improve the accuracy of prediction for phenomena such as phrase final lengthening.

The example sentence is *The first thing you need to know is how to speak to this computer*, which has been transcribed as the phoneme sequence, X, DH, UH0, F, ER1, S, T, TH, I1, J, UU1, N, EE1, D, T, UU1, N, OU1, I1, Z, H, AU1, T, UU1, S, P, EE1, K, T, UU1, DH, I1, S, K, UH0, M, P, J, UU1, T, ER0, X. The phoneme symbols are principally taken from the International Phonetic Alphabet with modifications for marking stressed and unstressed syllables, and transcribing to printable characters for purposes of computer processing. The system is the same as that used in a speech recognition system[5].



Using the sequence of phonemes and their durations generated, the feneme sequence for each test sentence is then generated. An example of a typical phoneme-feneme alignment is shown below, for the words *the first thing you need to know* as the start of the test sentence, above.

```
(...:X):    DS_2 DS_3 DS_2 DS_3 DS_3 DS_3 DS_2 DS_2 DS_2 DS_2 DS_2
            DS_2 DS_2 DS_3 DS_3 DS_3 DS_3 DS_3 DS_3 DS_3 DS_2 ONF_3 PQ_2
(254:DH):   ONL_2 ONDH3 ONDH3 PQ_1 DH_1
(259:UH0):  UH0_1 UH0_1 UH0_1
(262:F):    B_2 B_2 TH_2 TREE2 TREE2 F_2 TREE3 F_3 F_3 F_3 DS_3 F_3 F_4 F_4
(276:ER1):  PQ_1 ER1_1 ER1_1 ER1_1 ER1_1 ER1_2 ER1_2 ER1_2 ER1_3 ER1_3 ER1_3 ER1_3 ER1_3 ER1_3
(290:S):    S_2 S_2 S_2 S_2 S_2 Z_3 S_3 S_4 S_4 Z_4 Z_4 T_1 TRS_2 TRS_2 TRS_3
(305:T):    T_4 T_4 P_4 V_1
(309:TH):   DH_1 DH_3 DH_1 DH_4 DH_4 DH_4
(315:I1):   I1_1 I1_2 I1_2 UH0_1
(319:NG):   NG_2 NG_2 NG_2 NG_3 NG_3 NV_3 N_3 N_3 N_3
(328:J):    EE1_2 EE1_2 EE1_2 J_2 J_2 J_2 J_3 J_3 J_3 J_3 J_4 J_4
(340:UU1):  UU0_1
(341:N):    NV_2 NV_2 NV_2 N_2 ONM_4 NV_3 ONM_4 ONM_4 M_1 N_1
(351:EE1):  EE1_1 EE1_1 EE1_1 EE1_2 EE1_2 EE1_2 EE1_2 EE1_2 EE1_2 EE1_3 EE1_3 I0_4 TREE1 K_2 TREE2
(366:D):    TREE2
(367:T):    TREE3 TRSH3 T_4 TQ_2 TQ_2 TQ_2 TQ_3 TQ_3 TQ_3 TQ_4 TQ_4
(378:UU1):  UH0_1 UH0_1 UH0_1
(381:N):    ONM_3 NV_2 NV_2 N_1 N_3 N_4 ONM_4 ONM_4 M_1
(390:OU1):  D_1 EH1_1 EH1_1 EH1_1 AU1_1 ?_1 ?_1 AU1_1 AU1_1 AE1_2 AE1_2 UG_2 UG_2 UG_2 IG_1 IG_1 UG_2
UG_2 UG_3 UG_3 UG_3 UG_3 UG_3 UG_3 UU1_3 UU1_3 UU1_3 UU1_3 L_4 TRL_2 TRL_2 TRL_2 TRL_2
(423:I1):   X_1 TRM_3 DS_3 DS_2 DS_3 DS_2 DS_2 DS_2 DS_2 DS_2 DS_3 X_3 ONI_1 ONEE3 ONI_1 ONI_3 B_1
B_1 I0_1 I0_1 I0_1
```

The waveforms were then generated from the feneme sequences. A further step planned is the evaluation of the intelligibility and naturalness of the resulting waveform, as measured by the standard type of perceptual tests[12].

6. CONCLUSIONS

The objective of this study has been to investigate a method of concatenative synthesis which has some desirable properties. It captures speaker-dependent characteristics so that a natural speaker quality may be attempted. Larger segments of speech, such as diphones, demi-syllables, etc. can be modelled by the concatenation of a long sequence of sub-phoneme sized segments. The selection of segments can be motivated by, and to some extent synchronized with, the types of segments used in speech recognition systems for speaker-dependent speech recognition.

The advantage of this method is that techniques used in speech recognition can be applied to speech synthesis resulting in a useful sharing of concepts, and algorithms. There are also various practical benefits in terms of the sharing of tools and corpora, which are also not insignificant. Anecdotal evidence of playing these utterances in uncontrolled circumstances is that the sound produced is intelligible, and does have a quality recognisably like that of the original speaker.

The disadvantage of this approach is that there are a number of difficult decisions to take, for which there is as yet little guidance. The size of the feneme, and the size of the feneme alphabet, are among the major concerns. The assumption is that it is possible to define some combination of feneme length and codebook size which will allow natural sounding quality to be achieved. A methodology which might enable the aim of a customisable, speaker-dependent, natural quality speech synthesis TTS system to be achieved have been described.

7. REFERENCES

1. J.Allen, M.S.Hunnicutt and D.Klatt, *From Text to Speech: The MITALK system,* Cambridge University Press, 1987.
2. G.Bailly, C.Benoit and T.R.Sawallis, Ed, *Talking Machines, Models and Designs,* Elsevier, 1992
3. F.Charpentier and M.Stella, *Diphone Synthesis using an overlap-add technique for speech waveforms concatenation,* in ICASSP 86 (Tokyo) pp 2015-2018
4. F.Charpentier and E.Moulines, *Pitch Synchronous waveform processing techniques for text-to-speech synthesis using diphones,* In Proceeding EuroSpeech 89, Paris 1989, pp 13-19.
5. F.Jelinek et al, *The design of a large vocabulary discrete word speaker dependent dictation system,* Proceedings of IEEE vol. 11, No 3, Nov, 1985
6. A.G.Hauptmann, *SpeakEZ: A first experiment in concatenation synthesis from a large corpus,* Eurospeech 93, Berlin, Sept. 1993, pp 1701-1704
7. J.N.Holmes, *Speech Synthesis and Recognition,* Van Norstrand Reinhold(UK), 1988
8. K.F.Lee, *Large Vocabulary Speaker Independent Continuous Speech Recognition: The Sphinx system,* Ph.D Thesis, Dept. of Computer Science, Carnegie Mellon University, Pittsburgh, PA, 1988.
9. M.Macchi and M.Speigel, *Using a demi-syllable inventory to synthesise names,* in Speech Tech 90, Proceedings Voice Input/Output Applications conference and exhibition, Media Dimensions Inc., New York, 1990.
10. J.Makhoul, S.Roukos, and H.Gish, *Vector Quantization in Speech Coding,* Proceedings of the IEEE, vol. 73 No. 11, Nov. 1985, pp 1551-1588.
11. J.Olive, *Speech Synthesis by Rule,* In: Speech Communication, Ed G.Fant, vol. 2, Proceedings of the speech communication seminar, Stockholm, 1974. J.N.Holmes, *Speech Synthesis and Recognition,* Van Norstrand Reinhold(UK), 1988
12. D.Pisoni, P Nusbaum and B.Greene, *Perception of synthetic speech generated by rule,* Proceedings of IEEE 73, 11 (1985), 1665-1676
13. R.A.Sharman, *An Introduction to Language Modelling,* IBM Technical Report No 205, UKSC, Hursley, Winchester, UK, 1991
14. M.Spiegel, *Orator system technical briefs No 1.,* Technical report, Bell Communications and Research Lab, March 1991.