On the automatic training of phonetic units for word recognition

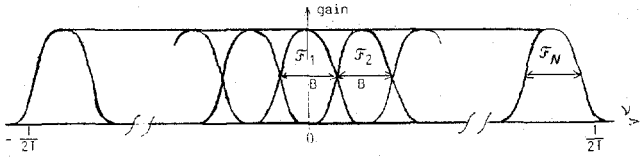

Fig. 2. The FSF bank.

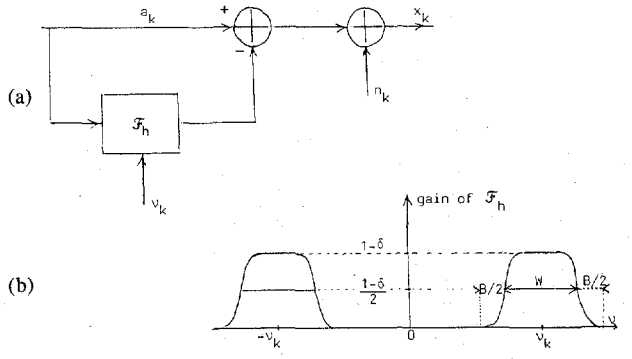

Fig. 3. The channel model.

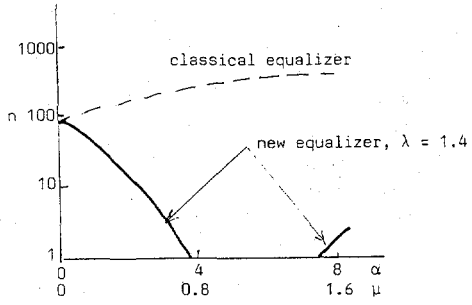

Fig. 4. Performance of the new equalizer in a case of a single moving notch.

These filters cover equally the Nyquist range of the sampled input signal.

The data a_k assume the four levels $\pm 1/\sqrt{5}$, $\pm 3/\sqrt{5}$ with equal probabilities; each data a_k encodes a group of two bits according to a Gray code. The fading channel model is shown in Fig. 3(a) where n_k is a white, zero-mean, Gaussian channel noise. The delayed and attenuated path is modeled by the filter F_h. We consider the case of a single moving notch in the band; F_h is a Nyquist filter [Fig. 3(b)] with a bandwidth W and a time-varying center frequency ν_k. The parameter δ is the channel gain at the frequency ν_k. The motion of ν is assumed sinusoidal, $\nu_k = \nu_o + \beta \sin(k\Delta f/\beta)$. However, the adaptation algorithm is not adjusted to any specific motion model. Then the conclusions reached via this model will hold for any motion having a similar speed for ν_k.

An important parameter that characterizes the fading velocity is the translation $\Delta\nu$ of the notch frequency during a time interval of length $1/W$, i.e., the duration of the impulse response of the fading channel. The dimensionless corresponding parameter $\Delta\nu/W = \Delta f/W^2T$ is an intrinsic measure of the nonstationarity degree of the fading. The other relevant parameters are evidently the fading depth δ and its bandwidth W.

Fig. 4 shows the performance of the new equalizer and of a classical frequency one (4) in the case $W = 3$ B, $\delta = -9$ dB, and $\Delta\nu W = 2$ percent that corresponds to a very rapid fading. Moreover, this fading is severe since it affects a third of the whole bandwith. In Fig. 4, n is the number of erroneous bits over a period of 1200 transmitted bits. The figure shows that the new equalizer with $\lambda =$ 1.4 and $3.8 \leq \alpha \leq 7.5$ ensures $n = 0$. On the other hand, the classical equalizer (4) gives results worse than no equalization at all ($\mu = 0$).

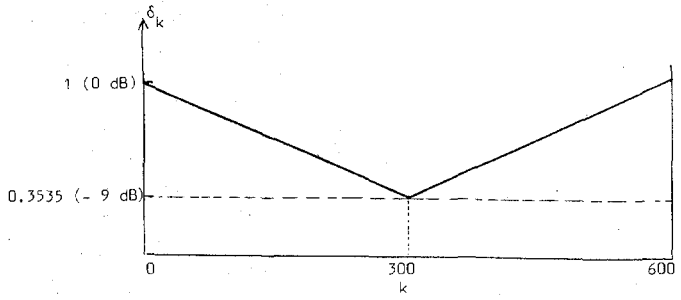

Fig. 5. Variation of δ_k with time.

In order to examine the ability of the new equalizer to track rapid variations in the depth δ of fading, we simulate the same channel model with a depth δ_k that varies according to Fig. 5. The results of the new equalizer with $\lambda = 1.4$ and $\alpha = 5$ are: $n = 0$ at SNR = 25 dB, and $n = 2$ at SNR = 18 dB. Hence, the new equalizer tracks rapid variations in both δ_k and ν_k.

References

[1] B. Golberg, "300 kHz–30 MHz MF/HF," *IEEE Trans. Commun. Technol.*, vol. COM-14, pp. 767–784, Dec. 1966.
[2] W. D. Rummler, "More on the multipath fading channel model," *IEEE Trans. Commun.*, vol. COM-29, pp. 346–352, Mar. 1981.
[3] F. Ling and J. G. Proakis, "Adaptive lattice decision-feedback equalizers—Their performance and application to time-variant multipath channels," *IEEE Trans. Commun.*, vol. COM-33, pp. 348–356, Apr. 1985.
[4] L. J. Griffiths, "Rapid measurements of digital instantaneous frequency," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-23, pp. 207–222, 1975.
[5] R. R. Bitmead and B. D. O. Anderson, "Adaptive frequency sampling filters," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-29, pp. 684–693, 1981.

On the Automatic Training of Phonetic Units for Word Recognition

HERMANN NEY, DIETER MERGEL, AND STEPHEN M. MARCUS

***Abstract*—In this correspondence, we present some preliminary results on using phonetic subword units in word recognition as compared to whole word templates. The phonetic subword units are specified as either phonelike units with and without temporal structure or as diphonelike units. The determination of these subword units requires segmentation, labeling, and parameter estimation at the same time, and is performed by an iterative two-stage algorithm consisting of nonlinear time alignment and parameter estimation.**

Experiments were carried out, using a connected digit recognition task, to study the usefulness of the subword unit representation and

Manuscript received June 6, 1984; revised March 22, 1985. This work was supported in part by the German Federal Ministry for Research and Technology (BMFT) under Grant 08 IT 15255, and was partly performed within a joint Siemens/Philips project. Only the authors are responsible for the contents of this work.

H. Ney and D. Mergel are with the Philips GmbH Forschungslaboratorium Hamburg, D-2000 Hamburg 54, West Germany.

S. M. Marcus is with the Institute for Perception Research (IPO), Eindhoven, The Netherlands.

IEEE Log Number 8406386

the effect on recognition performance of some versions of the subword specification. The best error rates for subword units are still, by a factor of 2 or more, larger than those for whole word templates.

I. Introduction

The purpose of this correspondence is to report on the first steps toward using phonetic subword units for automatic word recognition. Basing the recognition on units smaller than whole words is a major departure from our previous studies [1], [2] which relied exclusively upon whole word templates.

In principle, the use of subword units in recognition is expected to offer several advantages over whole word templates. First, it would ultimately lead to a discriminative, phonetic network as it is described by Klatt [3] and Moore *et al.* [4]. Thus, the indifferent summing up of local distances in the dynamic programming recursion for word recognition is avoided, and the appropriate weighting of discriminative segments within words is achieved automatically. As a result, an improvement in recognition accuracy should be achieved. The authors consider this the most important aspect of using phonetic subword units. A second advantage could be that the use of subword units would allow "synthetic training" of a word by building up the word artificially from the subword units according to the phonetic transcription of a lexicon. Third, as a byproduct, there could be a significant reduction of the computational expenditure in recognition because the pattern matching comparisons could be more or less restricted to the inventory of subword units. In practice, these advantages, especially the first one, may be hard to achieve due to idealized assumptions in the phonetic transcription and the mathematical difficulties in automatically determining the subword units.

There are a numer of systems that use subword units for recognition [5]–[7]. In the IBM system [5], the acoustic vectors are reduced to a finite set of phone labels by vector quantization, and a large amount of training is required to estimate the statistical parameters of the probabilistic finite state machines that specify the subword units. Other systems use demisyllables and either perform an explicit segmentation [6] or use subword units constructed beforehand by man [7].

The approach chosen here differs from the above approaches in a number of points; its main characteristics are as follows:

- no human interaction is required;
- there are two inputs to the training procedure: the standard phonetic transcription and the training utterances;
- the type of the subword units desired is easily changed by changing the rules for representing each phoneme or phoneme group as a sequence of acoustic vector labels at the 10-ms level;
- the technique for determining the subword units is based on dynamic programming and uses the same error criterion as the recognition algorithm does; and
- no explicit segmentation of the incoming utterance into subword units is performed; the subword boundaries are a byproduct of the recognition algorithm.

In order to test the performance of the method and the different types of subword units, recognition experiments were performed on connected digit strings. Admittedly, the recognition task and the vocabulary are rather simple, but these evaluations can serve as a first step toward using *a priori* phonetic knowledge in recognition and constructing a discriminative phonetic network. The ultimate aim is to test different types of phonetic subword units and to decide by experiment which type is the most suited.

II. The Function of the Subword Units and of the Lexicon

The essential function of the subword units and of the lexicon in our recognition system is to direct the construction of word reference patterns in the training phase. The recognition itself is, as in previous studies, based on whole word reference patterns and the corresponding matching scores. That means that the recognition is performed strictly within the lexical constraints.

TABLE I
Phonetic Transcription of the German Digit Vocabulary

eins: "A-J-N-S"
zwo: "T-S-W-OO"
drei: "D-R-A-J"
vier: "F-I-AR"
fuenf: "F-UE-N-F"
sechs: "Z-AE-K-S"
sieben: "Z-I-B-E-N"
acht: "A-CH-T"
neun: "N-O-J-N"
null: "N-U-L"

The phonetic transcription of the German digit vocabulary is from a German standard pronunciation dictionary and is shown in Table I. No additional phonetic or phonological rules are made use of. It is not checked to what degree each speaker's individual pronunciation corresponds to the pronunciation required by the lexicon. The phonetic transcription shown in Table I is given in terms of phonemes. In order to be able to study subword units like phonemes with a temporal structure or diphones in addition to "stationary" phonemes, we have to apply a conversion rule to the phonetic script. Since the final specification of the subword units must be given in terms of acoustic vectors, it is suitable to combine these two conversion steps as follows.

From the phonetic transcription, we derive a label script at the level of the acoustic vectors representing 10-ms segments of speech so that each vector label stands for an acoustic prototype vector. Different types of rules for converting the phonetic script into a vector label script are used, depending on what the subword units are expected to be. The duration of the elements represented by each prototype vector is not specified, but will be determined experimentally in the training phase. A temporal structure for phoneme X is introduced by the conversion rule:

$$-X- \rightarrow -x-x'-x''-$$

where x, x', x'' stand for three prototype vectors representing (quasi-)stationary portions at a subphonetic level. Diphonelike units are obtained for phoneme X in the context A–X–B by the context dependent conversion rule:

$$-X- \rightarrow -ax-ax'-x-xb-xb'-$$

where the labels ax, ax' and xb, xb' stand for (quasi-)stationary portions of the diphones A–X and X–B, respectively. Similarly, vector pairs were used in [8] to represent the temporal structure of a vector sequence. By increasing the number of independent prototype vectors, we implicitly take more and more account of phonetic context dependence. A completely different method would be a hierarchical strategy where first primary phonetic features are identified and then specific rules are used to refine the first recognition step by considering the phonetic context. Due to the use of a vector label script, subword units and phonetic transcription at the acoustic vector level are equivalent. Thus, we have achieved a mapping of the phonetic script on acoustic prototype vectors. What we need for a complete specification of the subword units, is to determine these prototype vectors and their durations. This problem is considered in the next section.

III. The Training Phase

A. The Estimation Criterion

The problem of determining the prototype vectors can be formulated as follows. Given the training utterances and the vector label script, the question is what the "best" estimates of the prototype vectors are.

To quantitatively specify the term "best" estimates, we need an estimation criterion that reflects the interdependence of segmenting and labeling the acoustic training vectors and estimating the pro-

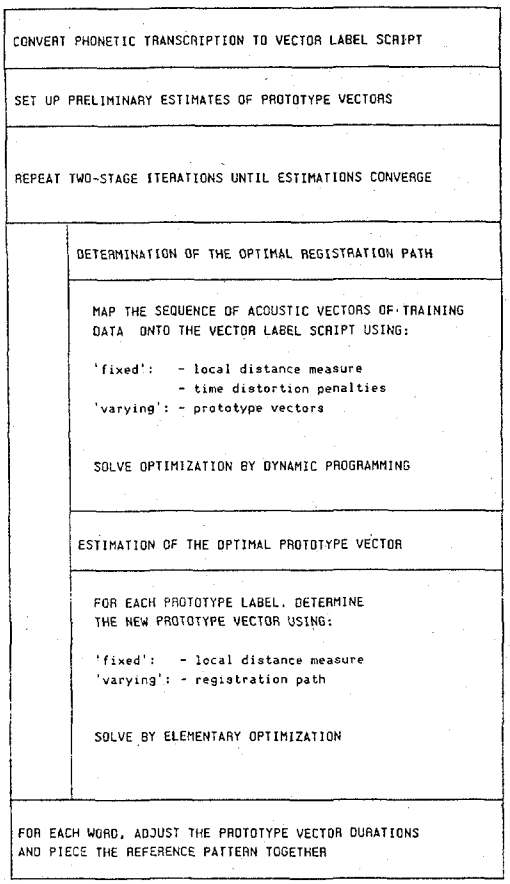

Fig. 1. Block diagram of the training algorithm.

totype vectors. Furthermore, this criterion should be consistent with the criterion used in the recognition phase, which guarantees a self-consistent set of subword units as opposed to subword units specified by human operators.

The task of segmenting and labeling can be formulated as finding a registration path j: $i \rightarrow j(i)$ that maps each acoustic training vector i on a prototype label $j(i)$. Consecutive acoustic training vectors can be mapped on the same prototype label as many times as necessary. The task of estimating the prototype vectors amounts to determining a set of 16 components for each prototype vector. Both tasks require a measure of dissimilarity or a distance measure between pairs of acoustic vectors. Using such a local distance, we define the estimation criterion as the global distance, i.e., the sum over the local distances along the registration path of all training utterances. The problem then is to determine both the registration path and the set of prototype estimates that minimize the global distance. This is quite a complex optimization problem due to the coupling between the registration path and the prototype estimates. We do not know how to solve this problem in such a way that attaining the global optimum can be guaranteed. Therefore, we use an iterative procedure that can be shown to be locally convergent and that is based on the same concepts as the Viterbi training in a probabilistic framework [5], [9], [10].

B. The Training Algorithm

A block diagram of the training algorithm is shown in Fig. 1. The algorithm starts with converting the phonetic script to a vector label script, as described in the previous section, and setting up some preliminary estimates of the prototype vectors which, in our implementation, is done by assuming equally long segment durations within a training utterance. Then we utilize a two-stage iteration procedure in order to improve the estimates first for the registration path and then for the prototype vectors. This iteration procedure will later be shown to be locally convergent.

In the first stage of the iteration, the sequence of acoustic training vectors is mapped onto the sequence of prototype vectors as given by the vector label script. The determination of this registration path generally requires a local distance measure between the acoustic vectors and a set of time distortion penalties. In our implementation, no time distortion penalties were used so that for each acoustic vector of the training data, exactly one local distance with no extra penalty contributes to the overall criterion. The registration path is determined by dynamic programming.

In the second stage of the iteration, the new-found registration path is used to update the estimates of the prototype vectors. For each prototype label, all acoustic training vectors registered with it are collected and, depending on the local distance measure, the components of the prototype vector must be estimated by an elementary optimization. For the squared Euclidean distance, this estimation means calculating the sample average of each component of the acoustic training vectors. For the absolute value distance, the estimation amounts to calculating the sample median of each vector component. What varies in this stage from iteration to iteration, is the registration path and thus the selection of acoustic training vectors used for refining the prototype estimates.

In each of the two optimization stages, the sum of all matching scores becomes smaller or at least remains the same. Thus, these iterations result in monotonically decreasing sequence of overall matching scores. On the other hand, it is clear that the overall matching scores have a lower bound of zero. According to an elementary theorem on sequences, the sequence must therefore converge. The requirement, of course, is that in each optimization step the operations performed consistently lead to a lower matching score.

In order to simplify the calculations, the sample median, which is the correct estimator for the absolute value distance, is replaced by the sample average. In informal comparative tests, this approximation has been found to result only in negligible differences. Despite this small inconsistency, we have encountered no convergence problem in the iterations. To initialize the training procedure, a linear segmentation was used for each training utterance. The number of segments is known from the conversion rules. Typically, convergence occurs after 15 iterations for a training set of 4 repetitions of the digit vocabulary, and the average distance score is decreased by a fraction of typically 5 percent. After the iterations have converged, the final step in the training algorithm is the explicit construction of the reference patterns. The prototype vectors are placed together according to the vector label scripts for each vocabulary word, and their duration at a given word position is chosen as the average duration observed in the training utterances in order to appropriately model the length of each word. Comparative tests show that nearly the same results are obtained when the duration of each segment is already estimated during the iterations and when the time distortion penalties are different from zero. This effect can be explained by the fact that the endpoints of the training utterances spoken in isolation serve as anchor points for the time registration path and heavily constrain its degrees of freedom.

IV. Experimental Results

A. The Recognition Task

To evaluate the usefulness of different subword representations, recognition experiments were carried out on a connected digit recognition task. The recognition conditions were speaker-dependent recognition and high-quality speech. The acoustic parameters were the cepstral parameters 1, $\cdots$, 15 and the relative intensity which was computed as the difference between successive zeroth cepstral coefficients. The recognition algorithm is described elsewhere [2]. The time distortion penalties in the nonlinear time alignment procedure were chosen in such a way that, on the average, they roughly reflected the speaking rate differences between training and test utterances.

The recognition algorithm that is based on whole word references remains unchanged. The introduction of subword units affects only

TABLE II
TOTAL OF ERRORS[=(DELETIONS, CONFUSIONS, INSERTIONS)] FOR 3500 DIGIT RECOGNITION TESTS

Method	Number of Prototypes	Errors $(D, C, I) = S$	Error Rate (Percent)
Whole Words	570	(3, 11, 3) = 17	0.5
38 Segments	38	(3, 62, 12) = 77	2.2
22 Segments	22	(5, 85, 29) = 119	3.4
22 Segments with Poor Initialization	22	(6, 328, 61) = 395	11.3

TABLE III
TOTAL OF ERRORS[=(DELETIONS, CONFUSIONS, INSERTIONS)] FOR 3500 DIGIT RECOGNITION TESTS

Substitution of X in A-X-B	Number of Prototypes	Errors $(D, C, I) = S$	Error Rate (Percent)
One Prototype: $-x-$	22	(5, 85, 29) = 119	3.4
Temporal Structure plosives: $-x-x'-x''-$ other: $-x-x-x-$	30	(2, 82, 18) = 102	2.9
Temporal Structure: $-x-x'-x''-$	66	(1, 33, 5) = 39	1.1
Diphone Model: $-ax-ax'-x-xb-xb'-$	74	(1, 23, 6) = 30	0.9

the way in which the whole word templates are constructed. The vocabulary was the German digits with the "2" being spoken as "zwo." The tests were performed on five speakers (three male, two female). The training utterances were obtained by having each person speaking the digits four times in isolation. Only these 40 utterances of each speaker were used as training utterances for the training algorithm described in Section III. As test utterances, 100 strings of seven digits were recorded for each speaker. Thus, the overall number of digits to be recognized is 5 speakers × 100 strings × 7 digits = 3500 digits. In connected word recognition, three types of word errors can occur: deletions, confusions, insertions. The recognition results presented in the next subsection are given in terms of the total of word errors = [deletions, confusions, insertions] for the 3500 digits to be recognized.

B. Recognition Using Subword Units

First, a control experiment with whole word patterns and no phonetic subword units was carried out by using time aligned and averaged reference patterns. In addition, recognition tests with several types of subword units were run. The results are shown in Tables II and III. For each recognition test, four columns are given: the type of subword units used, the resulting number of independent prototype vectors, the absolute number of recognition errors in terms of deletion, confusion and insertion errors and the word error rate. For whole word patterns, the word error rate is 0.5 percent (Table II), which is caused by 17 recognition errors in the 3500 recognition tests.

Using the phonetic transcription of Table I without any modification leads to a drastic increase of the errors as Table II shows. The phonetic transcription implies 38 phonetic segments, 22 of which are different. Recognition experiments were performed for both 38 and 22 segments, where each segment is represented by exactly one prototype vector. The use of 38 segments actually ignores the phonetic transcription completely and merely specifies the number of independent prototype vectors. As compared to no phonetic transcription, the numbers of errors were drastically increased by a factor of 4 and 7. An additional experiment using 22 prototype vectors was run to test the influence of the initialization technique on the recognition results. In the linear segmentation used for initialization, the segment boundaries were shifted to the middle of each segment, which resulted in a new segmentation and in poor initial estimates of the prototype vectors. As a matter of fact, the experiment shows a critical dependence on the initialization technique: the error rate went up by a factor of nearly 4 when compared to the original segmentation, whereas the optimization criterion, i.e., the overall distance, was increased by merely 5 percent.

The drastic increase of the error rate for subword units could indicate that the conversion rules from the phonetic level to the vector label level must better take account of the context dependence of the prototype vectors. In order to test these ideas, an additional series of experiments were performed using different types of conversion rules as introduced in Section II. The results are summarized in Table III.

The first result for 22 prototype vectors is included for the sake of completeness and is taken from Table II. Since a plosive is only poorly modeled by a stationary segment, a temporal structure for plosives is introduced. Each plosive consists of three stationary segments. There are 4 plosives in the vocabulary dictionary so that a total of 30 prototype vectors is obtained. This refinement of the plosives leads to only a slight improvement. A much higher improvement is achieved by extending this temporal structure to all phonemes and using 66 prototype vectors: the error rate is reduced from 2.9 to 1.1 percent. For the digit vocabulary, the diphone model results in 74 prototype vectors and produces a similar error rate, namely, 0.9 percent.

The results in Table III show that the recognition performance depends on the type of subword units and above all on the number of prototype vectors. For the best experiments, the number of errors is still clearly higher than in the case of whole word based recognition with no subword units. As can be seen from the number of prototype vectors shown in Table III, there is a clear correlation between the number of prototype vectors and the recognition accuracy: the higher the number of prototype vectors, the higher the recognition accuracy. For this small vocabulary, it is not possible to tell whether diphonelike units perform better than phonemelike units with a temporal structure, because both types of units result in nearly the same number of prototype vectors, 74 and 66, respectively. Roughly the same number of prototype vectors, namely, 60, was found optimal in recognition tests with vector quantization for the same speech database [11]. To achieve higher recognition accuracies, it could be useful to combine the approach presented with the hierarchical strategy that uses more context dependent and phonological rules.

V. CONCLUSIONS

We have presented a method for the determination of phonetic subword units that is based on a two-stage iteration. The method has an inherent flexibility to test different types of subword units. The preliminary experiments carried out on connected digit strings lead to an error rate which, for the best type of subword units, is typically double the error rate obtained for whole word patterns without any phonetic transcription. Further insights could be expected from tests on a larger vocabulary, from a better context-dependent phonetic transcription, from checking the speaker's pronunciation and from examining how good the local optimum obtained by the iteration procedure is.

REFERENCES

[1] M. H. Kuhn and H. Tomaschewski, "Improvements in isolated word recognition," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-31, pp. 157–167, Feb. 1983.
[2] H. Ney, "The use of a one-stage dynamic programming algorithm for connected word recognition," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-32, pp. 263–271, Apr. 1984.
[3] D. Klatt, "Speech perception: A model of acoustic-phonetic analysis and lexical access," *J. Phonet.*, vol. 7, pp. 279–312, 1979.
[4] R. K. Moore, M. J. Russell, and M. J. Tomlinson, "The discrimi-

native network: A mechanism for focusing recognition in whole word pattern matching," in *Proc. 1983 IEEE Int. Conf. Acoust., Speech, Signal Processing*, Boston, MA, Apr. 1983, pp. 1041–1044.
[5] L. R. Bahl, F. Jelinek, and R. L. Mercer, "A maximum likelihood approach to continuous speech recognition," *IEEE Trans. Pattern Anal., Machine Intell.*, vol. PAMI-5, pp. 179–190, Mar. 1983.
[6] G. Ruske and T. Schotola, "The efficiency of demisyllable segmentation in the recognition of spoken words," in *Proc. 1981 IEEE Int. Conf. Acoust., Speech, Signal Processing*, Atlanta, GA, Mar.–Apr. 1981, pp. 971–974.
[7] A. E. Rosenberg, L. R. Rabiner, S. E. Levinson, and J. G. Wilpon, "A preliminary study on the use of demisyllables in automatic speech recognition," in *Proc. 1981 IEEE Int. Conf. Acoust., Speech, Signal Processing*, Atlanta, GA, Mar.–Apr. 1981, pp. 967–970.
[8] S. M. Marcus, "Time in the process of speech recognition," in *Proc. 10th Int. Congr. Phonet. Sci.*, Utrecht, The Netherlands, Aug. 1983, pp. 307–313.
[9] P. F. Brown, C.-H. Lee, and J. C. Spohrer, "Bayesian adaptation in speech recognition," in *Proc. 1983 IEEE Int. Conf. Acoust., Speech, Signal Processing*, Boston, MA, Apr. 1984, pp. 761–764.
[10] Y. Kamp and C. J. Wellekens, "Viterbi training in speech recognition," personal communication, Oct. 1983.
[11] H. Bourlard, C. J. Wellekens, and H. Ney, "Connected digit recognition using vector quantization," in *Proc. 1984 IEEE Int. Conf. Acoust., Speech, Signal Processing*, San Diego, CA, Mar. 1984, pp. 26.10.1–4.

Estimating the Steady-State Frequency of a Sine-Wave Burst with Extremely Short Record Length

E. I. PLOTKIN, L. M. ROYTMAN, AND M. N. S. SWAMY

Abstract—**We present a technique based on modified linear prediction for accurate estimation of the steady-state frequency of a sine-wave burst with extremely short record length. The estimation procedure presented here makes use of automatic windowing of the burst location within the observation interval. Such a windowing technique eliminates the transition effects which accompany the estimators derived using the traditional LPC technique. To implement this windowing, a set of artificial state variables, given by the product of output samples and their delayed versions, is used. The advantages of this method are demonstrated using a computer simulation.**

I. Introduction

The goal of this work is to investigate the efficient algorithm for estimating a frequency of a pulse-shaped sine wave with extremely short record length. The term frequency refers to the "steady-state frequency" (SSF); this means that the pulse under consideration is artificially prolonged so that the frequency is associated with the reciprocal period. Without neglecting the many applications of SSF estimation of sine bursts in digital communication and radar systems, such an estimation may be viewed as a good test example for the evaluation of spectrum analysis methods.

The problem of estimation of parameters of sinusoidal signals in the presence of noise has been attracting the attention of researchers in the field for the last decade [2]–[7], and many algorithms for its solution have been presented so far in the literature.

Manuscript received September 12, 1984; revised August 20, 1985.

E. I. Plotkin is with the Department of Electrical and Computer Engineering, Ben-Gurion University, Beer-Sheva 84105, Israel.
L. M. Roytman is with the Department of Electrical Engineering, City College of New York, New York, NY 10031.
M. N. S. Swamy is with the Department of Electrical Engineering, Concordia University, Montreal, P.Q. H3G 1M8, Canada.

IEEE Log Number 8406029.

These algorithms are based on either numerous modifications of linear predictive techniques [2], [4], [7] or statistical methods such as maximization of likelihood ratio, spectral estimation via maximum entropy [6], or some special methods like singular value decomposition [3] or the Pisarenko and Prony methods [5].

In all of these methods occur the same difficulties if the signal under analysis has a short record length comparable to the length of one period and its location within the observation interval is unknown *a priori*.

One of the possible ways to solve the problem is to use an endpoint detector to detect a sine-burst location within the observation interval; then it is conceptually possible to process it in a rigorous manner as a continuous signal. Another one is a nonadaptive approach (presented in this correspondence) based on a special nonlinear transformation of a sine-wave burst with unknown location. Such an approach improves the performance of frequency estimation of an extremely short sine burst.

II. Problem Formulation

Consider a burst sine-wave signal $s(t)$ of amplitude A, phase ϕ, and steady-state frequency ω_0:

$$s(t) = AQ\left(\frac{t}{\Delta}\right) s_c(t) = AQ\left(\frac{t}{\Delta}\right) \cos(\omega_0 t + \phi) \tag{1}$$

where $Q(t/\Delta)$ is a rectangular window with duration Δ and unknown location along the t axis; $s_c(t)$ is a continuous sinusoidal function of frequency ω_0. With the addition of noise $w(t)$ in (1), the signal under analysis is

$$x(t) = s(t) + w(t) \tag{2}$$

where $w(t)$ is a background zero-mean band-limited white noise process of variance σ^2.

We shall assume that the signal $s_c(t)$ can be represented by a finite-dimensional (M) linear prediction model:

$$G(x) = \sum_{k=0}^{M} \hat{a}_k x_k \tag{3}$$

where $x_k = x(t - k\tau)$ is a delayed by $k\tau$ version of a signal $x(t)$.

For the sinusoidal signal, such a model may be restricted to $M = 2$ [2], [4] and $a_2 = a_0 = 1$, i.e., only one coefficient $\hat{a}_1 = \hat{a}$ has to be estimated.

In the presence of noise, the constant $\hat{a}$ is to be so chosen as to minimize the resulting mean square

$$\epsilon^2(t) = \langle G^2(x) \rangle, \qquad t \in (0, T)$$

where T is the record length of the observation interval.

Applying the LMS criterion to (3), we obtain

$$\hat{a}(\omega) = \left. \frac{\langle x_2 + x, x_1 \rangle}{\langle x_1, x_1 \rangle} \right|_{0,T} \tag{4}$$

where $\langle\ ,\ \rangle$ denotes an inner product of two real functions in the L^2 [8].

Note that such an estimator results in an LMS error in the coefficient $\hat{a}(\omega)$ estimation if T is significantly large ($T \gg T_0 = 2\pi/\omega$). If $x(t) = s_c(t)$, i.e., noise $w(t)$ is absent and $T \gg T_0$, then the estimator (4) gives us a true value of $a(\omega) = -2 \cos \omega_0 \tau$. On the other hand, if $x(t) = s(t)$ [a burst sine wave given by (1)], the estimator (4) would result in a certain error *even in the absence of noise*. The main reason for this error is a boundary effect in the estimation (4) due to the "burst" character of a signal (1).

Referring to Fig. 1, we can see that there are five distinct parts of the integrants in (4). We note that if (4) is applied within the range $(0', T')$, it would give a correct result because this case is equivalent to continuous signal processing (without transients). Thus, in the absence of noise ($w(t) \equiv 0$),