

THE APRIL 1983  
VOL. 62, NO. 4, PART 1



BELL SYSTEM  
TECHNICAL JOURNAL

---

<b>Analysis of Thermally Induced Loss in Fiber-Optic Ribbons</b> G. S. Brockway and M. R. Santana	993
<b>Adaptive Linearization of Power Amplifiers in Digital Radio Systems</b> A. A. M. Saleh and J. Salz	1019
<b>An Introduction to the Application of the Theory of Probabilistic Functions of a Markov Process in Automatic Speech Recognition</b> S. E. Levinson, L. R. Rabiner, and M. M. Sondhi	1035
<b>On the Application of Vector Quantization and Hidden Markov Models to Speaker-Independent, Isolated Word Recognition</b> L. R. Rabiner, S. E. Levinson, and M. M. Sondhi	1075
<b>A Lithographic Mask System for MOS Fine-Line Process Development</b> J. M. Andrews	1107
<b>A Circuit That Changes the Word Rate of Pulse Code Modulated Signals</b> J. C. Candy and O. J. Benjamin	1161
CONTRIBUTORS TO THIS ISSUE	1169
PAPERS BY BELL LABORATORIES AUTHORS	1173
CONTENTS, MAY-JUNE 1983	1179

# THE BELL SYSTEM TECHNICAL JOURNAL

## ADVISORY BOARD

D. E. PROCKNOW, *President*

I. M. ROSS, *President*

W. M. ELLINGHAUS, *President*

*Western Electric Company*

*Bell Telephone Laboratories, Incorporated*

*American Telephone and Telegraph Company*

## EDITORIAL COMMITTEE

A. A. PENZIAS, *Committee Chairman, Bell Laboratories*

M. M. BUCHNER, JR., *Bell Laboratories*

R. P. CLAGETT, *Western Electric*

T. H. CROWLEY, *Bell Laboratories*

B. R. DARNALL, *Bell Laboratories*

B. P. DONOHUE, III, *American Bell*

I. DORROS, *AT&T*

R. A. KELLEY, *Bell Laboratories*

R. W. LUCKY, *Bell Laboratories*

R. L. MARTIN, *Bell Laboratories*

J. S. NOWAK, *Bell Laboratories*

L. SCHENKER, *Bell Laboratories*

G. SPIRO, *Western Electric*

J. W. TIMKO, *American Bell*

## EDITORIAL STAFF

B. G. KING, *Editor*

PIERCE WHEELER, *Managing Editor*

LOUISE S. GOLLER, *Assistant Editor*

H. M. PURVIANCE, *Art Editor*

B. G. GRUBER, *Circulation*

**THE BELL SYSTEM TECHNICAL JOURNAL** (ISSN0005-8580) is published by the American Telephone and Telegraph Company, 195 Broadway, N. Y., N. Y. 10007; C. L. Brown, Chairman and Chief Executive Officer; W. M. Ellinghaus, President; V. A. Dwyer, Vice President and Treasurer; T. O. Davis, Secretary.

The Journal is published in three parts. Part 1, general subjects, is published ten times each year. Part 2, Computing Science and Systems, and Part 3, single-subject issues, are published with Part 1 as the papers become available.

The subscription price includes all three parts. Subscriptions: United States—1 year \$35; 2 years \$63; 3 years \$84; foreign—1 year \$45; 2 years \$73; 3 years \$94. Subscriptions to Part 2 only are \$10 (\$12 foreign). Single copies of the Journal are available at \$5 (\$6 foreign). Payment for foreign subscriptions or single copies must be made in United States funds, or by check drawn on a United States bank and made payable to The Bell System Technical Journal and sent to Bell Laboratories, Circulation Dept., Room 1E-335, 101 J. F. Kennedy Parkway, Short Hills, N. J. 07078.

Single copies of material from this issue of The Bell System Technical Journal may be reproduced for personal, noncommercial use. Permission to make multiple copies must be obtained from the editor.

Comments on the technical content of any article or brief are welcome. These and other editorial inquiries should be addressed to the Editor, The Bell System Technical Journal, Bell Laboratories, Room 1J-319, 101 J. F. Kennedy Parkway, Short Hills, N. J. 07078. Comments and inquiries, whether or not published, shall not be regarded as confidential or otherwise restricted in use and will become the property of the American Telephone and Telegraph Company. Comments selected for publication may be edited for brevity, subject to author approval.

Printed in U.S.A. Second-class postage paid at Short Hills, N. J. 07078 and additional mailing offices. Postmaster: Send address changes to The Bell System Technical Journal, Room 1E-335, 101 J. F. Kennedy Parkway, Short Hills, N. J. 07078.

© 1983 American Telephone and Telegraph Company.

# THE BELL SYSTEM TECHNICAL JOURNAL

DEVOTED TO THE SCIENTIFIC AND ENGINEERING  
ASPECTS OF ELECTRICAL COMMUNICATION

---

Volume 62

April 1983

Number 4, Part 1

---

Copyright © 1983 American Telephone and Telegraph Company. Printed in U. S. A.

## Analysis of Thermally Induced Loss in Fiber-Optic Ribbons

By G. S. BROCKWAY\* and M. R. SANTANA

(Manuscript received October 1, 1982)

*In this paper, added loss during temperature cycling in a given ribboned fiber is shown to be caused by thermally induced axial compressive strain imparted to the fiber. A microbending-sensitivity parameter  $\delta$  is introduced which reduces all loss-strain curves corresponding to different fibers to one characteristic master curve. Thermoviscoelasticity theory is used to calculate the time- and temperature-dependent compressive strain imparted to a ribboned fiber during a standard environmental cycle. Combining these analytical results with environmental data, the functional relationship between fiber-compressive strain and the added loss for a fiber of any given  $\delta$  in an Adhesive-Sandwich Ribbon (ASR) with Urethane-Acrylate (UA) coated fibers has been determined. Using this analysis, the added loss for a UA ASR can now be predicted for any environmental cycle. The critical material properties that dominate the environmental performance of ASRs are the tape shrinkback at elevated temperatures and the product  $\alpha EA$  of the coefficient  $\alpha$  of thermal expansion, the time- and temperature-dependent relaxation modulus  $E$ , and the area  $A$  of the coating.*

### I. INTRODUCTION

Unless special precautions are taken, fiber-optic cables installed in the outside plant could experience temperatures ranging from  $-45^{\circ}\text{F}$

---

\* Plastics Engineering Consultants, Inc., Lawrenceville, GA.

to +190°F. Thus, optical transmission loss resulting from this thermal history directly impacts the coating and ribbon choices for a particular system design. An understanding of the relationships among the thermally induced strains on the "ribboned" glass fibers, the resulting added transmission loss, and fiber parameters is crucial to properly evaluating candidate fiber-coating materials, ribbon structures, and/or ribbon-matrix materials. Moreover, it is desirable to be able to predict long-term behavior from short-term testing, thereby simplifying the environmental testing procedure.

The added loss in environmental testing is generally thought to be associated with the microbending of the axis of the fiber. Moreover, it has been shown by other investigators that when intimate contact is forced between a fiber and a microscopically rough surface, losses due to microbending can be substantial and are a function of the fiber geometric and optical parameters as well as the elastic modulus of the rough surface.<sup>1,2</sup> Therefore, thermally induced strains on the ribbon structure can indirectly result in added loss by increasing the contact forces between fibers. In addition, the sensitivity of a particular fiber to microbending loss is said to be associated with irregularities at the core-cladding interface, which may be due to core-diameter variations and/or refractive index variations.<sup>3</sup>

In this paper, a thermoviscoelastic analysis is used to compute the axial and transverse strains that are imparted to a fiber-optic ribbon when it is subjected to any given thermal cycle. This analysis shows that for the Bell System Adhesive-Sandwich-Ribbon<sup>4</sup> (ASR) construction with Urethane-Acrylate (UA) coated fibers, the transverse strain due to lowering the temperature results, contrary to one's expectation, in a reduction in the contact force between fibers. On the other hand, the resulting axial compressive strain evidently either increases this contact force or induces fiber buckling at the critical spatial wavelength, and thus creates added optical loss. Moreover, for a given ribboned fiber we establish herein the existence of a relationship between the environmental added loss and any measure of microbending sensitivity of the fiber. This equivalence enables us to predict the environmental performance of any ribboned fiber over a wide temperature-time span from data gathered over a relatively short time. In particular, the loss data collected from one excursion to -45°F can be combined with thermoviscoelastic data and analysis to predict the loss of any ribbon configuration with any choice of constituent materials, provided the tested fibers have a suitable range of microbending sensitivity.

## II. ENVIRONMENTAL TEST PROCEDURE

The intent of environmental testing is to determine the effect of thermal exposure on the optical performance of fiber-optic ribbons. All

Table I—Environmental test cycle

Cycle Number	Exposure Temperature, $T$ ( $^{\circ}\text{F}$ )	Exposure Time, $t$ (days)
I	75	2
	-45	2
	-15	2
	15	2
	75	2
	170	30
II	75	2
	-45	2
	-15	2
	15	2
	75	2
	190	34
III	75	2
	-45	2
	-15	2
	15	2
	75	2

the ribbons tested were placed in a 23-inch-diameter cardboard cylindrical container in a stem-pack fashion. The containers were then placed in a "walk-in" environmental chamber with both ends available for measuring purposes. Input and output array connectors<sup>5</sup> were fabricated to measure loss by the reference-fiber technique. Namely, the 0.63- $\mu\text{m}$  loss of a fiber is obtained by taking the ratio of its output power to the average output power of ten reference fibers that are short enough to have negligible loss.

Table I is a summary of the environmental testing cycle used for this investigation. The loss of the ribboned fibers\* was obtained at the end of each exposure. The change in loss for a typical ribboned fiber is plotted in Fig. 1 for each temperature in Cycles I, II, and III. The loss of the ribboned fibers increases with decreasing temperatures, the maximum loss occurring at the lowest temperature. As is evident in Fig. 1, the loss also increases with increasing cycle number. We subsequently show that this "pumping effect" is due to polyester-tape shrinkback<sup>†</sup> and stress relaxation effects during the high-temperature exposure between cycles.

It follows that for a 40-year design life, the worst-case environmental cycle would be a 40-year exposure at 190 $^{\circ}\text{F}$  followed by a low-temperature exposure. This worst-case cycle is summarized in Table II. Note

\* Fibers used in this study had cladding and core diameters of 110 and 55  $\mu\text{m}$ , respectively.

<sup>†</sup> Shrinkback is the recovery of process-induced strains.

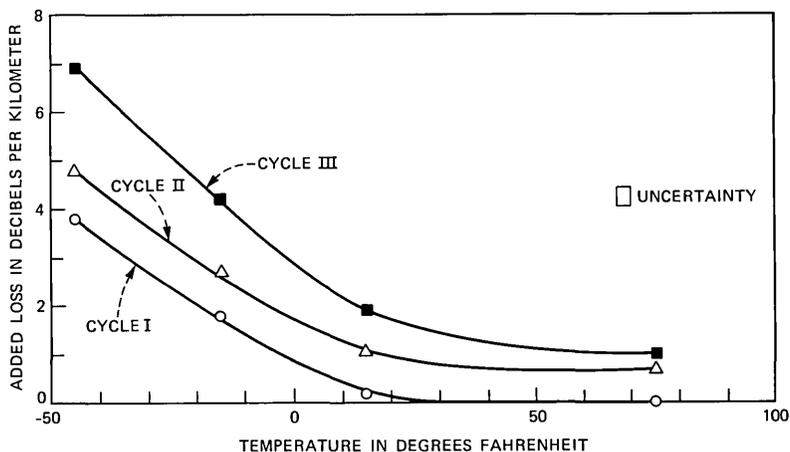


Fig. 1—Environmental added loss at 0.63  $\mu\text{m}$  vs. temperature (UA-coated fiber, ASR construction).

Table II—Worst-case environmental cycle

Exposure temperature, $T$ (°F)	Exposure Time, $t$ (days)
190	14,610 (40 years)
75	2
-45	2
-15	2
15	2
75	2

that the cycle of Table II is not suggested for testing purposes but will only be used to predict the most pessimistic estimates of performance.

### III. DUALITY BETWEEN MICROBENDING SENSITIVITY AND STRAIN

A convenient means for measuring the microbending sensitivity of different fibers is to determine the wavelength-independent loss coefficient<sup>6</sup> of the fiber when wound in several layers on a 6-inch-diameter reel under a tension fixed for all fibers. Let  $\delta$  denote the wavelength-independent loss coefficient measured under these conditions. This method was contrived to produce artificially high losses, thereby magnifying the loss contribution due to microbending. Of course, the losses for these same fibers will be substantially smaller when measured in a stress-free configuration.

The added-loss response to a given cycle (see Fig. 1) is characteristic of all fibers of a single given  $\delta$  (as defined above) in a given structure

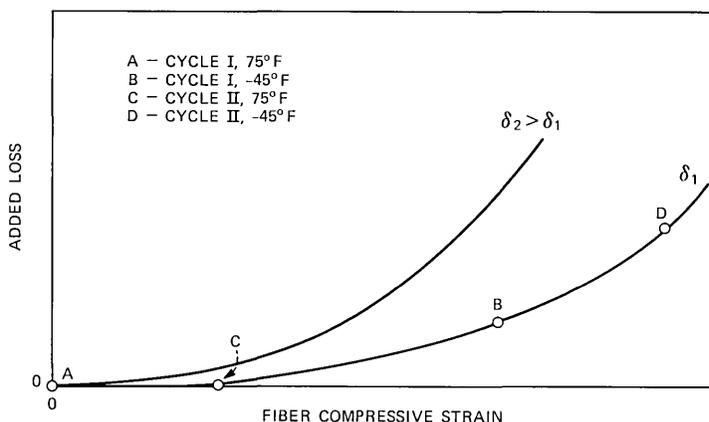


Fig. 2—Environmental added loss vs. fiber compressive strain.

but will change for fibers having different  $\delta$ 's. Moreover, examination of all the environmental data collected in this study indicates that the change in loss at a given point in any cycle increases with increasing  $\delta$ .

As is well known, microbending loss occurs when the axis of a fiber is bent into a curve whose spectrum contains a certain critical frequency.<sup>3</sup> Transverse pressure against a fiber on a microscopically rough surface has been shown to introduce loss in this way.<sup>1,2</sup> However, in the case of environmentally induced loss in an ASR ribbon, it is shown in the Appendix that the *transverse* strain due to thermal contraction *reduces* the contact forces between fibers. Thus, the added-loss response of Fig. 1 must be due to another mechanism. The only reasonable possibility is that the compressive axial strain induces microbending loss through fiber buckling. This loss would occur if the fiber buckling were at the critical wavelength or if buckling increases the contact forces between the fiber and its surroundings. It is not our intent to identify which of these means dominates the behavior of the fiber but only to characterize its manifestations. In other words, we simply correlate the added loss with the compressive strain imparted to the fiber.

Both the added loss and compressive strain increase with decreasing temperature; their relationship\* is shown schematically in Fig. 2. Here, the losses at 75°F and -45°F in Cycle I are denoted by A and B, respectively. After high-temperature exposure, the shrinkback of the polyester tape increases the strain on the fiber, so that C and D

\* If strain and added loss monotonically increase with decreasing temperature, then it is easy to show that the added loss monotonically increases with increasing strain.

represent 75°F and -45°F in the second cycle. This explains the increased loss with each cycle that is evidenced in Fig. 1. This phenomenon has been observed consistently in all environmental testing and, as previously noted, is called the pumping effect.

Since the loss increases with increasing  $\delta$ , the added-loss response to strain becomes steeper with increasing  $\delta$ , as illustrated in Fig. 2. The effect of  $\delta$  in this graph is obviously a contraction of the strain scale with increasing  $\delta$ . As to whether this contraction is uniform or not is readily tested by plotting the loss versus logarithmic strain, where a uniform contraction would then appear as a horizontal, rigid translation of the added-loss curves with  $\delta$ . As will be demonstrated in Section V, the loss-strain data may indeed be reduced in this fashion. Since the strain-scale contraction factor is a function of  $\delta$  alone (the contraction is uniform), it creates a one-to-one correspondence (duality) between  $\delta$  and compressive strain, changes in  $\delta$  being equivalent to changes in strain. With this in mind, we proceed to a calculation of the axial strain in the environmental cycle.

#### IV. THERMOMECHANICAL ANALYSIS

This section is devoted to the summary of the formulas needed to calculate the axial strain history to which a fiber-optic ribbon is subjected during any environmental cycle. All of the plastics used in an ASR structure possess time- and temperature-dependent moduli, two examples of which are shown in Fig. 3. If the plastic is instantaneously strained, the stress required to sustain that strain relaxes with time according to the given curve. Notice that the relaxation that occurs in the UA modulus at 140°F in one hour takes more than 40 years at room temperature. The relaxation at -45°F is slower yet, so that the UA coating is as stiff as the polyester tape over significant time periods at these low temperatures.

On the other hand, the coefficients of thermal expansion of these plastics are constant with time provided the temperature changes do not encompass their glass transition temperatures,  $T_g$ . Even so, by virtue of the time dependence of the plastic moduli, the ribbon itself exhibits time- and temperature-dependent thermal expansion and contraction during the thermal cycle. This situation is depicted in Fig. 4. Since the expansion coefficients  $\alpha_P$  of the plastics are much greater than that ( $\alpha_G$ ) for glass, the fibers restrain the contraction of the plastic, the net contraction  $\alpha\Delta T$  of the ribbon being determined by equilibrium (force balance) in the structure. The relatively high short-time modulus of the plastic (see Fig. 3) causes the initial contraction (expansion) to be high. As the modulus of the plastic relaxes, the energy stored in the glass causes the ribbon to recover some of this high initial strain. Of course, the rate of this recovery depends on the temperature at which it occurs.

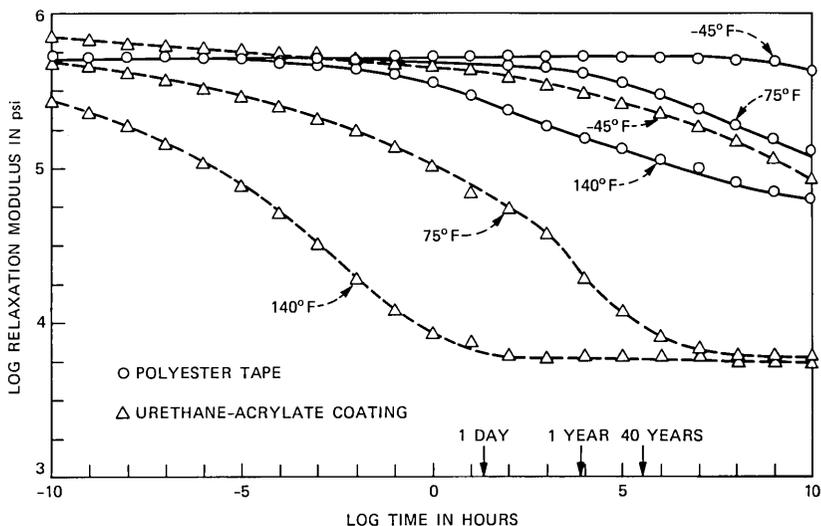


Fig. 3—Time dependence of moduli for ribbon constituents at various temperatures (generated by time-temperature superposition from data collected by R. P. DeFabritis).

The glass also resists shrinkback  $\epsilon_S^{TA}$  of the polyester tape during exposure to elevated temperatures. If it were unrestrained, the tape would shrink back according to the results in Fig. 5. As illustrated in Fig. 6, the shrinkback  $\epsilon_S$  that the ribbon experiences is much less than that of the free tape.

The compressive strain induced in the ribbon by its thermal contraction upon an excursion to  $-45^\circ\text{F}$  is thus increased in each subsequent cycle by the shrinkback during the high-temperature exposure. We now proceed to a calculation of these strains.

#### 4.1 Axial thermal expansion

If the plastic phases were considered to be elastic, the axial tensile modulus of a fiber-optic ribbon would be approximated well by the *rule of mixtures*

$$E_e = \frac{\sum_i E_i A_i}{\sum_i A_i}, \quad (1)$$

where  $E_i$  is the modulus of the  $i$ th constituent,  $A_i$  its area, and the summation is taken over all phases. In reality, (1) is a lower bound for the effective elastic modulus, being exact if all the constituents have the same Poisson's ratio.<sup>7</sup> Furthermore, eq. (1) can be derived in an elementary fashion by considering the equilibrium of the ribbon if one ignores Poisson's effect and assumes that all constituents are equally strained.

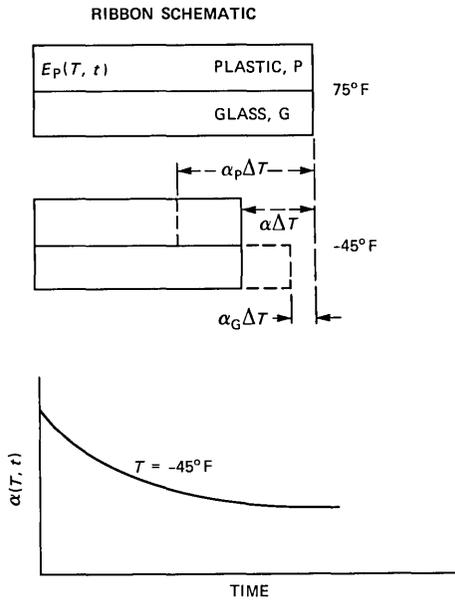


Fig. 4—Mechanics of ribbon thermal contraction.

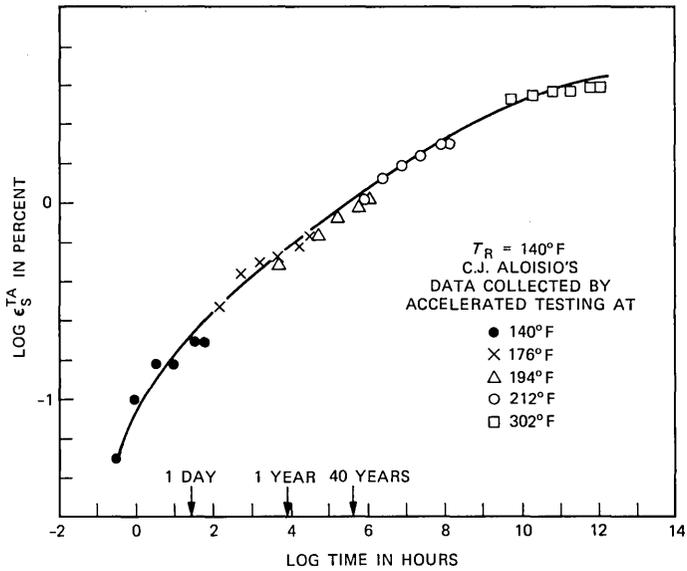


Fig. 5—Shrinkback strain vs. time for 3M No. 5 polyester tape.

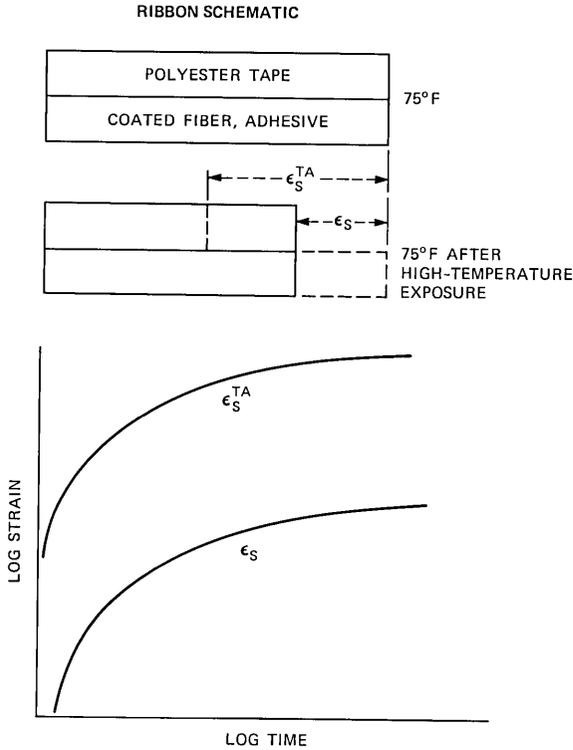


Fig. 6—Mechanics of ribbon shrinkback.

This same kind of elementary argument leads to the formula

$$\alpha_e = \frac{\sum_i \alpha_i E_i A_i}{\sum_i E_i A_i} \quad (2)$$

for the coefficient of thermal expansion of the ribbon, given elastic phases with moduli  $E_i$ , areas  $A_i$  and coefficients of thermal expansion  $\alpha_i$ . Schapery has shown that (2) is exact if each phase has the same Poisson's ratio and is a very good approximation in any case.<sup>7</sup> These elastic results may be used to generate their viscoelastic counterparts. To see this, assume that all constituents obey constitutive equations of the form<sup>8</sup>

$$\sigma(t) = \int_0^t E(T(t), t - \tau) \dot{\epsilon}_o(\tau) d\tau \quad (3)$$

$$E(T, t) = a_G(T) E(T_R, \xi), \quad \xi = \int_0^t \frac{d\tau}{a_T(T(\tau))}$$

and

$$\epsilon_s(t) = \epsilon(t) - \alpha[T(t) - T_0], \quad T_0 = T(0), \quad (4)$$

where  $\epsilon$ ,  $\sigma$ , and  $E(T, \cdot)$  are the strain, the stress and the relaxation modulus at temperature  $T$ , respectively. Here, a superposed dot on a function denotes differentiation with respect to its argument. On a log-log plot of modulus versus time,  $\log a_G(T)$  and  $\log a_T(T)$  represent, respectively, the vertical and horizontal distances that the curve for a temperature  $T$  must be translated to lay over that for the reference temperature  $T_R$ . All the materials considered here, including glass as a trivial case, conform to this hypothesis.

When (3) and (4) are met, it is readily shown that equation (1) continues to hold in the viscoelastic case, so that

$$E(T, t) = \frac{\sum_i E_i(T, t)A_i}{\sum_i A_i} \quad (5)$$

is exact (Poisson's effect being ignored). Moreover, as long as the curve of the logarithm of the modulus versus logarithmic time has small curvature, a good approximation to  $\alpha(T, t)$  is provided by the so-called quasi-elastic approximation<sup>8</sup>

$$\alpha(T, t) = \frac{\sum_i \alpha_i E_i(T, t)A_i}{\sum_i E_i(T, t)A_i}. \quad (6)$$

The strain in the ribbon due to a temperature history  $T$  in the absence of stress is characterized by

$$\epsilon(t) = \int_0^t \alpha(T(t), t - \tau) \dot{T}(\tau) d\tau. \quad (7)$$

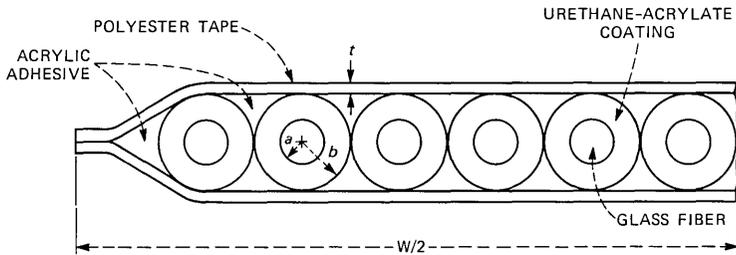
Notice that the ribbon has a time-dependent expansion coefficient even though we have supposed that the constituent plastics do not. This latter assumption is valid as long as the temperature excursion does not encompass the glass transition temperature  $T_g$  of any constituent. Equation (6) can easily be shown to continue to hold even if the constituents have time-dependent expansion coefficients. One need only replace each  $\alpha_i$  by  $\alpha_i(T, t)$ . For the purposes of this investigation, it is sufficient to assume each  $\alpha_i$  to be constant.

The data needed to calculate the effective modulus  $E$  and coefficient  $\alpha$  of thermal expansion of the ribbon at  $-45^\circ\text{F}$  and  $75^\circ\text{F}$  for a time of 10 hours is given in Table III. The geometric data of Table III was calculated using the nominal dimensions shown on the cross-sectional view of a typical ribbon in Fig. 7. Notice that although the glass makes far and away the most significant contribution to the stiffness of the

Table III—Geometric and material properties of ribbon constituents

	Ten-Hour Modulus							
	$A$ ( $10^{-4}$ in. <sup>2</sup> )	$\alpha$ ( $10^{-5}$ °F <sup>-1</sup> )	$E$ ( $10^5$ psi)		$EA$ ( $10^2$ lb)		$\alpha EA$ ( $10^{-3}$ lb/°F)	
			-45°F	75°F	-45°F	75°F	-45°F	75°F
3M No. 5 Polyester Tapes	3.00	1.06	5.27	4.67	1.58	1.40	1.67	1.48
3M No. 5 Acrylic Adhesive	3.60	7.72	0.00	0.00	0.00	0.00	0.00	0.00
Urethane-Acrylate Coatings	5.87	3.33	4.22	0.74	2.48	0.43	8.26	1.43
Glass Fibers	1.77	0.028	107	107	18.94	18.94	0.53	0.53
<b>Total</b>	<b>14.24</b>				<b>23.00</b>	<b>20.77</b>	<b>10.46</b>	<b>3.44</b>

Thus, eq. (6)  $\rightarrow \alpha = \frac{\sum \alpha EA}{\sum EA} = \frac{4.55 \times 10^{-6} \text{ °F}^{-1} \text{ @ } -45\text{°F}}{1.66 \times 10^{-6} \text{ °F}^{-1} \text{ @ } 75\text{°F}}$



DIMENSIONS
$a = 2.17$ mils
$b = 9.00$ mils
$t = 1.00$ mil
$w = 150$ mils

Fig. 7—Cross section of a twelve-fiber ribbon.

ribbon, other constituents play a major role in the calculation of  $\alpha$ . Indeed, at  $-45^\circ\text{F}$  the  $\alpha EA$  of the glass may well be neglected in comparison with that of the coating. At  $75^\circ\text{F}$ , on the other hand, the tape, coating and glass make comparable contributions to the effective thermal expansion coefficient.

In Fig. 8,  $\alpha$  and  $E$  are plotted logarithmically against the logarithm of time at  $75^\circ\text{F}$ . It is evident that the change in  $\alpha$  with respect to temperature and time is much more significant than the change in ribbon modulus. The coefficients of thermal expansion listed for each of the constituents in Table III are for temperatures below their respective glass transition temperatures, since our interests are in

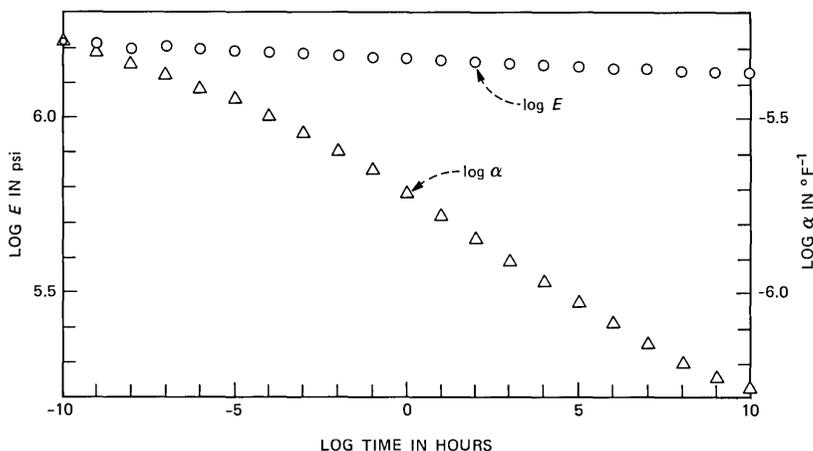


Fig. 8—Time dependence of the modulus and the coefficient of thermal expansion for the ribbon at 75°F (UA-coated fibers, ASR construction).

calculating the compressive strains imparted to the fiber during the low-temperature excursion.

If the ribbon is exposed to the temperature history

$$T(t) = T_0 + \sum_{k=1}^n (T_k - T_{k-1})H(t - t_k), \quad H(t) = \begin{cases} 0 & t < 0 \\ 1 & t \geq 0 \end{cases} \quad (8)$$

$0 < t_1 < t_2 < \dots < t_n < t$ , the resulting strain history is

$$\epsilon(t) = \sum_{k=1}^n (T_k - T_{k-1})\alpha(T_n, t - t_k), \quad (9)$$

as found from substituting (8) into (7). Observe that eq. (9) contains all relaxation effects associated with the temperature history in (8).

#### 4.2 Axial shrinkback

We now set out to calculate the shrinkback induced in the ribbon by the shrinkage of the polyester tape during high-temperature exposure. Denote by  $\epsilon_S^{TA}(T_R, t)$  the shrinkback (see Fig. 5) the tape would experience in the time period  $[0, t]$  at some reference temperature\*  $T_R$  if it were unrestrained. At any other temperature  $T$ , the unrestrained shrinkback in the same time period would be

$$\epsilon_S^{TA}(T, t) = \epsilon_S^{TA}(T_R, t/a_T(T)), \quad (10)$$

where  $a_T$  is the temperature-dependent scale-contraction factor. If the tape is subjected to temperature  $T_1$  for a time  $t_1$  and then to  $T_2$  for  $t_2$ , the total shrinkback would be

\*  $T_R = 60^\circ$  in Fig. 5.

$$\epsilon_S^{TA}(T_R, t_1/a_T(T_1) + t_2/a_T(T_2)).$$

For the polyester tape used in the ASR construction, shrinkback is observable in the environmental cycle only during the high-temperature exposures, the time scales being much too long at the lower temperatures.

An analysis similar to that outlined in Section 4.1 for the time-dependent modulus and coefficient of thermal expansion of the ribbon can be used to obtain the approximate formula for the shrinkback strain of the ribbon  $\epsilon_S$ .

$$\epsilon_S(t) \cong S(T, t)\epsilon_S^{TA}(t), \quad (11)$$

where

$$S(T, t) = \frac{E_{TA}(T, t)A_{TA}}{\sum_i E_i(T, t)A_i}. \quad (12)$$

A double logarithmic plot of the ribbon shrinkback function  $S$  as calculated from (12) is shown in Fig. 9 for  $T = 140^\circ\text{F}$ . Observe that the shape of  $\log S(T, \cdot)$  is very much like that of the logarithm of the polyester modulus in Fig. 3. The ribbon shrinkback during any high-temperature exposure can be calculated by substituting the appropriate mechanical and shrinkback data (e.g., Figs. 3 and 5) into eqs. (11) and (12). The error made in using eq. (11) at  $140^\circ\text{F}$  has been shown to decrease monotonically from 10 percent at two hours to 8.8 percent at ten hours. The approximation of eq. (11) increasingly improves as time goes on and as temperature increases. Since our interest is in times greater than forty-eight hours at temperatures over  $170^\circ\text{F}$ , (11) is quite acceptable.

#### 4.3 Calculation of fiber compressive strains in the environmental test cycle

As discussed in Section III, a compressive strain applied to a ribboned fiber induces added optical loss. This strain  $\epsilon_F$  for the temperature history  $T$  [see eq. (8)] is given by

$$\epsilon_F(t) = \epsilon_R(t) - \alpha_F[T(t) - T_0], \quad (13)$$

where  $\alpha_F$  is the linear coefficient of thermal expansion of the glass fiber and

$$\epsilon_R(t) = \epsilon(t) + \epsilon_S(t) \quad (14)$$

is the ribbon strain. In (14),  $\epsilon$  is the strain due to the thermal contraction of the ribbon [eq. (9)], and  $\epsilon_S$  is the ribbon shrinkback strain [eq. (11)].

Thermoviscoelastic data on each of the constituent materials were incorporated into computer-programmed versions of eqs. (6), (9), (11),

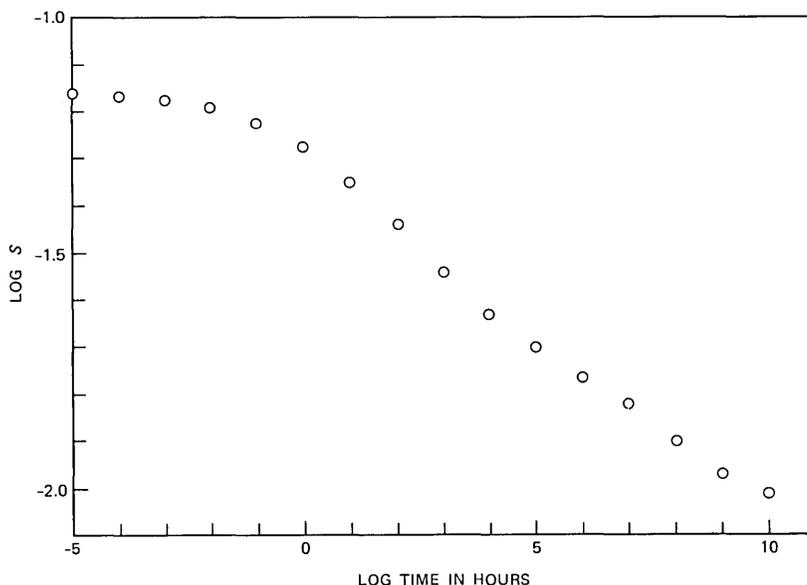


Fig. 9—Time dependence of ribbon-shrinkback function at 140°F (UA-coated fibers, ASR construction).

(12), and (13) to obtain the fiber strains on each excursion to  $-45^{\circ}\text{F}$ . The results are plotted in Fig. 10.

#### V. THE MASTER CURVE FOR ADDED LOSS VERSUS COMPRESSIVE STRAIN

The purpose of this section is to construct the curve of added loss versus compressive strain (master curve) for a fiber having an arbitrary value of the microbending sensitivity parameter  $\delta$ . This will be done by demonstrating that the effect of  $\delta$  on the loss-strain curve is to uniformly contract the strain scale, which appears as a rigid shift (translation) of the curves with  $\delta$  when the loss data are plotted against logarithmic strain. This master curve together with the  $\delta$ -shift curve can then be used to predict the performance of any fiber ( $\delta$  known) in any environmental cycle.

In Fig. 11, the environmental added-loss data for five fibers are plotted versus the logarithm of the axial compressive strain on the glass fiber as calculated in Section IV (Fig. 10). These data include all the measurements for Cycles I through III according to Table I. Neither of the points for  $75^{\circ}\text{F}$  in Cycle I appears in these figures because for the first the strain is zero and for the second it is tensile (negative), so that both logarithms are undefined.

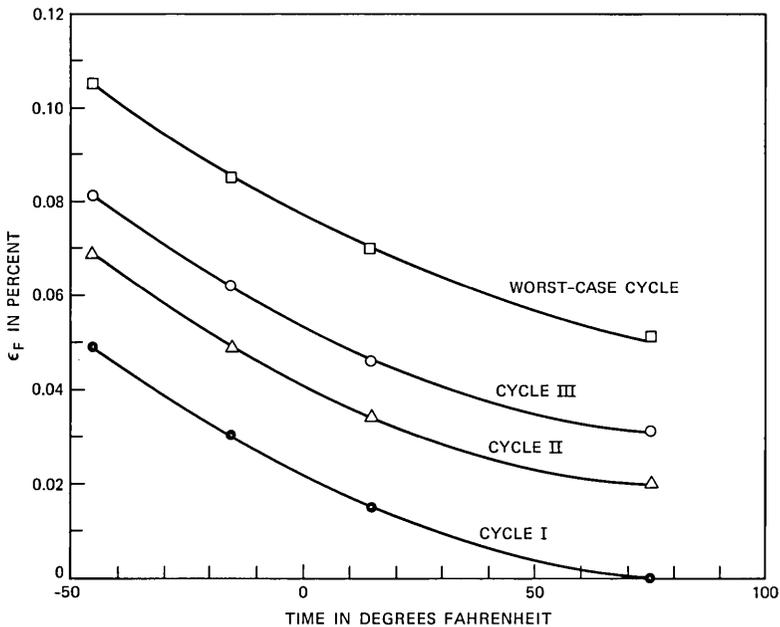


Fig. 10—Calculated fiber compressive strain vs. temperature in the environmental cycle (UA-coated fiber, ASR construction).

### 5.1 Shifting with $\delta$ to form the master curve

The plots in Fig. 11 were laid over one another and shifted horizontally by hand until they formed a single, smooth curve. The amount of shift  $\log a_{\epsilon}(\delta)$  required for each of the five values of  $\delta$  is shown in Fig. 12 referenced to  $\delta = 1.0$  dB/km. The linear, least-squares fit shown in Fig. 12 results in an excellent approximation of the shift data. For simplicity this linear approximation is used in the subsequent construction of the master curve.

The resulting master curve is shown in Fig. 13. The data points for the five different fibers are shown with different symbols on this plot. This master curve together with the  $\delta$ -shift curve of Fig. 12 can be used to obtain the added-loss-versus-strain profile for a fiber of any given  $\delta$ . A fourth-order polynomial has been fitted to the master curve data and is included in Fig. 13. Observe that the scatter about the best-fit polynomial is within the loss measurement uncertainty. Notice also that there is significant overlap in the data from fiber to fiber. On account of the functional relationship between  $a_{\epsilon}$  and  $\delta$  (illustrated in Fig. 12), we may view the master curve of Fig. 13 as added loss versus strain for a fixed value of  $\delta$  or added loss versus  $\delta$  for a fixed level of strain. This makes precise the  $\delta$ -strain duality alluded to in Section III.

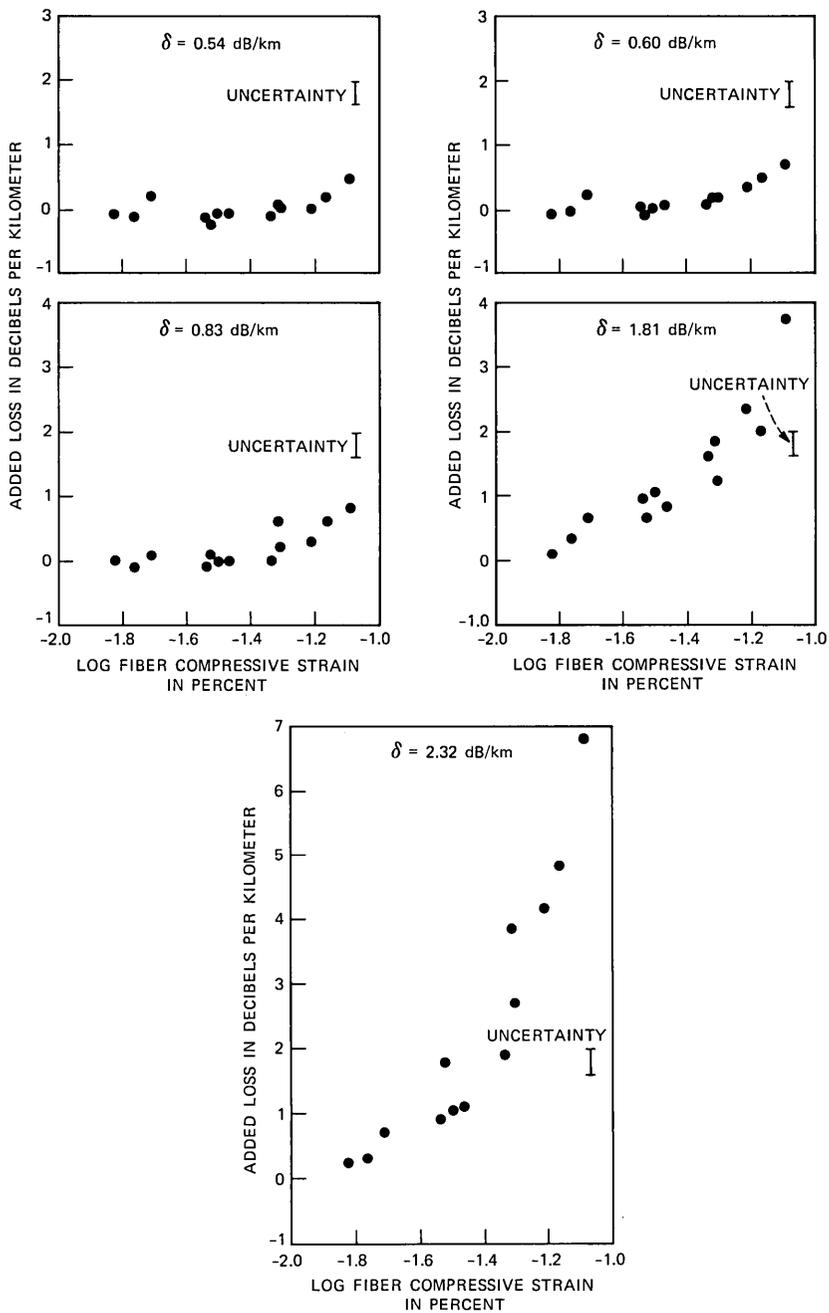


Fig. 11—Environmental added loss at  $0.63 \mu\text{m}$  vs. calculated fiber strain in the environmental test (UA-coated fibers, ASR construction).

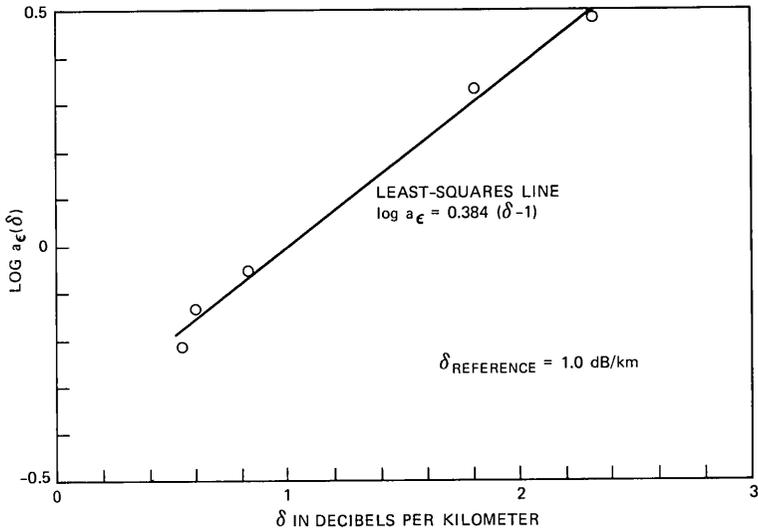


Fig. 12—Strain-scale-contraction factor vs. microbending sensitivity parameter  $\delta$  (referenced to  $\delta = 1.0 \text{ dB/km}$ , UA-coated fibers, ASR construction).

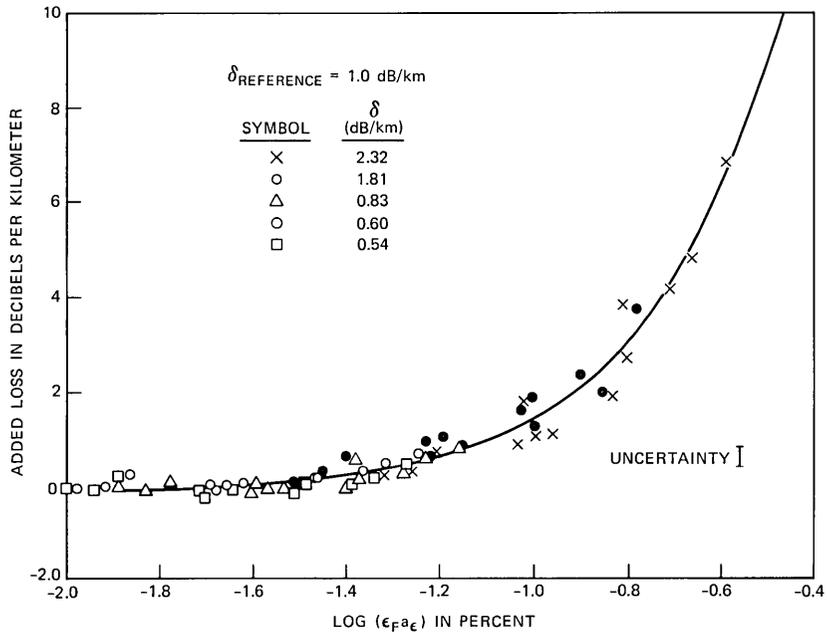


Fig. 13—Environmental added loss vs.  $\delta$ -reduced fiber strain (referenced to  $\delta = 1.0 \text{ dB/km}$ , UA-coated fibers, ASR construction).

Although the master curve of Fig. 13 has been constructed from data collected in all three cycles, this was not necessary. Indeed, a polynomial fit through Cycle I is virtually indistinguishable from the one shown in Fig. 13. Thus, we are led to conclude that only one low-temperature excursion is necessary to complete an analysis of the type presented here, provided fibers having a suitable range of  $\delta$ 's are tested. This observation greatly simplifies the environmental-testing procedure for fiber-optic ribbons.

### 5.2 Prediction of loss in the worst-case cycle

We are now in a position to predict the behavior of a UA ASR in the worst-case environmental cycle. To do this we need only add to the strain associated with each temperature and time (see Fig. 10) the shift factor associated with the appropriate value of  $\delta$  (see Fig. 12) for the fiber in question. The loss can then be read off the master curve (Fig. 13). For example, the strain  $\epsilon_F$  in the worst-case cycle at  $-45^\circ\text{F}$  obtained from Fig. 10 is 0.105%. The shift corresponding to a  $\delta$  of 2.32 dB/km from the equation in Fig. 12 is  $\log a_\epsilon(2.32) = 0.51$ . Thus, the log of the  $\delta$ -reduced strain [ $\log \epsilon_F a_\epsilon(2.32)$ ],  $-0.47$ , yields an added loss of 9.8 dB/km from Fig. 13. This procedure was employed to generate the curves in Fig. 14 for the ribbon behavior in the worst-case cycle for different values of  $\delta$ .

Notice for a  $\delta$  of 2.32 dB/km in Fig. 14 that the loss in the worst-case cycle is much worse than that of Cycle III shown in Fig. 1. This

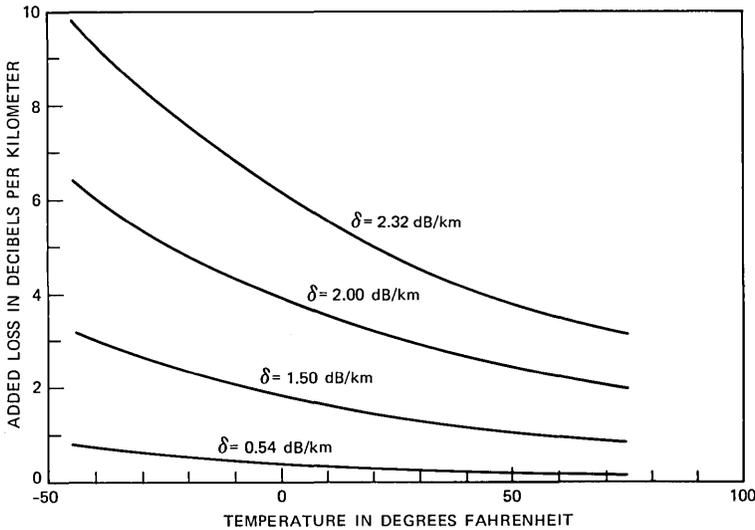


Fig. 14—Environmental added loss at  $0.63 \mu\text{m}$  vs. temperature in the worst-case cycle (UA-coated fibers, ASR construction).

of course is the shrinkback effect previously discussed. Also, observe from comparing Figs. 11 and 14 that a fiber with  $\delta = 0.54$  dB/km would meet a 0.5 dB/km added-loss criterion after Cycle III but not after the worst-case cycle. It is therefore clear that data sufficient to allow long-term predictions are essential to any meaningful ribbon evaluation. Finally, we remark that the added loss for a UA ASR can be predicted in the way demonstrated above for *any* environmental cycle provided one calculates the appropriate strain history by means of the formulas in Section IV.

## VI. APPLICATIONS

### 6.1 A Criterion for coating/ribbon-structure comparisons

If one were to choose a ribbon-performance requirement of 1.0 dB/km in the worst-case cycle, then it is clear from Fig. 14 that an ASR made with a population of UA-coated fibers having high  $\delta$ 's would be unacceptable. The same structure manufactured with low- $\delta$  fibers would satisfy the requirement. Thus, one can easily err out of ignorance of the strong effect arising from the sensitivity of the fiber to micro-bending (as reflected in  $\delta$ ). Obviously, for a given performance requirement there is a maximum permissible  $\delta$ -value,  $\beta$ . That is, if ribbons are manufactured from a fiber selection having  $\delta$ 's less than  $\beta$ , the given performance requirement will be met. Clearly then, it is not a question of whether the UA-ASR structure is acceptable or not, but rather how restrictive the value of  $\beta$  is. The parameter  $\beta$  is therefore an effective criterion for comparing various coating and ribbon-structure combinations.

To calculate this maximum permissible  $\delta$ -value,  $\beta$ , for the UA ASR given a 1.0 dB/km added loss in the worst-case cycle, we enter into the master curve (Fig. 13) at a loss of 1.0 dB/km and read off the associated reduced strain

$$\log \epsilon_F a_\epsilon(\beta) = -1.09. \quad (15)$$

Next, introduce the equation for the  $\delta$ -shift curve (Fig. 12),

$$\log a_\epsilon(\beta) = 0.384(\beta - 1)$$

into (15) to arrive at

$$\beta = 1 - \frac{1.09 + \log \epsilon_F}{0.384}. \quad (16)$$

Now the  $-45^\circ\text{F}$  strain in the worst-case cycle is 0.105 percent, so that (16) yields

$$\beta = 0.7 \text{ dB/km.}$$

Similarly, for a performance requirement of 0.5 dB/km, one finds

$$\beta = 0.2 \text{ dB/km.}$$

Thus, UA-coated fibers selected with  $\delta$ 's less than  $\beta$  could be used in an ASR and satisfy the chosen performance requirement for a 40-year design life.

### 6.2 Critical material and geometric properties

The maximum permissible  $\delta$ ,  $\beta$ , can also be used to evaluate how material and geometric properties might be altered to improve environmental performance. It is obvious that a reduction in polyester-tape shrinkback would reduce environmental added loss. This is illustrated dramatically in Fig. 15, where the parameter  $\beta$  is plotted versus shrinkback normalized with respect to that of the present tape in the worst-case cycle (see Fig. 5). The two curves are for performance standards of 0.5 and 1.0 dB/km in the worst-case cycle. Notice that a 50-percent reduction increases  $\beta$  from 0.2 to 0.5 dB/km for the 0.5 dB/km performance requirement, while elimination of the shrinkback

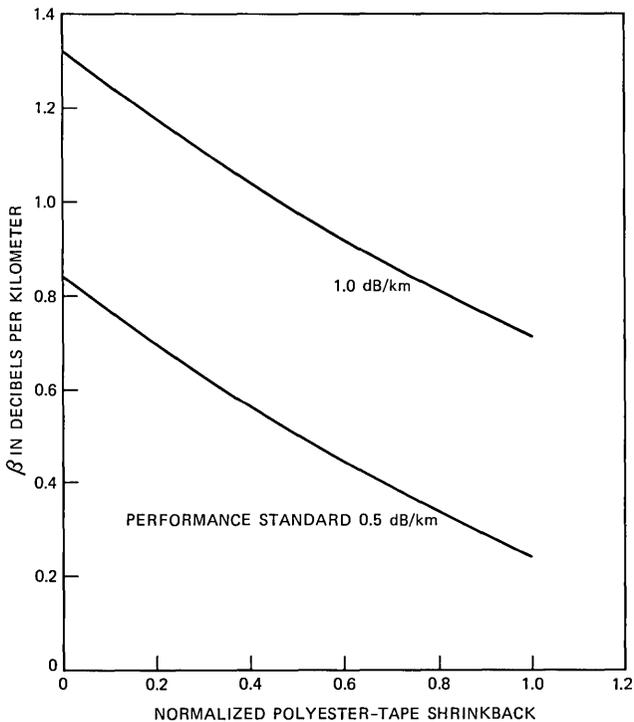


Fig. 15—Effect of shrinkback on the maximum permissible  $\delta$ -value for the worst-case cycle (UA-coated fibers, ASR construction).

altogether increases  $\beta$  to 0.8 dB/km. Observe from eqs. (11) and (12) that the ribbon shrinkback is linear with respect to the area of the tape. Thus, Fig. 15 can also be viewed as a plot of  $\beta$  versus reduced area normalized to that of the present tape (see Table III).

A consideration of Table III reveals that reducing the  $\alpha EA$  product of the coating will likewise diminish the strain on the fibers, since the coefficient of thermal expansion of the ribbon would then be reduced. The coefficients  $\alpha$  of thermal expansion are essentially the same for all plastics though the moduli  $E$  can vary substantially from one material to another. Notice further that the area  $A$  of the coating is greater than that of any other ribbon constituent. As to whether this reduction in fiber strain is accompanied by an environmental performance improvement depends on the extent to which the added-loss-versus-compressive-strain master curve and its associated  $\delta$ -shift characteristic are influenced by the changes adopted to reduce the fiber strain. Changing  $\alpha$  of the coating has no effect on these curves while changing coating modulus  $E$  and/or the ribbon geometry may have a substantial effect. The variation in the shape of these curves with changes in these various parameters can be ascertained only by completing a characterization of the type carried out here for the UA ASR.

Figure 16 shows how strong an effect reducing thermal contraction can be when the change in the added-loss-reduced-strain characteristic is negligible. Here we have plotted  $\beta$  against the coated-fiber outer diameter holding the fiber diameter constant. Observe that if the outer diameter of the coated fiber is reduced from 9 to 6 mils the effect on  $\beta$  is bigger than the effect of a 50-percent reduction in tape shrinkback. Letting the coating thickness tend to zero results in an increase in  $\beta$  from 0.2 to 1.0 dB/km for the 0.5 dB/km performance requirement. If a 50-percent reduction in tape shrinkback is combined with a reduction in coated-fiber outer diameter to 6.5 mils, one can show that  $\beta$  increases from 0.2 to 0.9 dB/km. Finally, we remark that these same dramatic effects should occur with reductions in the coating modulus.

Other investigators have established that dual-coated fibers having a soft primary coating are less sensitive to microbending than fibers having a hard, single coating.<sup>1</sup> Such dual-coated fibers in an ASR should also behave well environmentally. To see this, consider a secondary coating of UA over silicone where the area of the UA skin is  $1.56 \times 10^{-5}$  in.<sup>2</sup> Neglecting the stiffness of the silicone (it is three orders of magnitude smaller than that of UA), we obtained an effective single-coated UA diameter of 6.4 mils. The corresponding increase in  $\beta$  (Fig. 16) is from 0.2 to 0.6 dB/km for the 0.5 dB/km performance criterion. As previously mentioned these conclusions are predicated upon the assumption that the basic loss-strain characteristic of the ASR structure is relatively insensitive to changes in coating modulus.

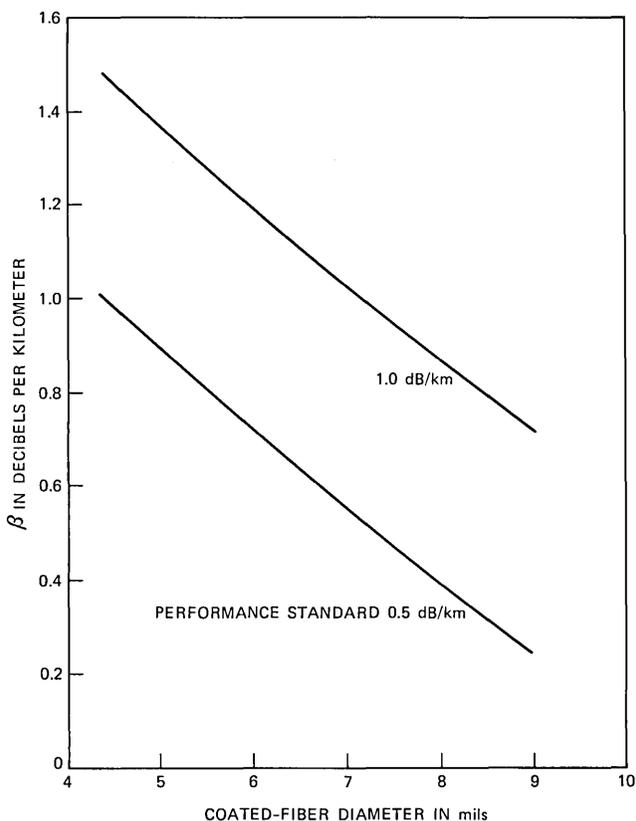


Fig. 16—Effect of coating thickness on the maximum permissible  $\delta$ -value for the worst-case cycle (UA-coated fibers, ASR construction, fiber diameter = 4.33 mils).

## VII. CONCLUSIONS AND RECOMMENDATIONS

We have shown that:

(i) Environmental added loss in an ASR is associated with axial compressive strain (not transverse strain) imparted to the fibers by thermal contraction of the ribbon in the low-temperature excursion and ribbon shrinkback together with relaxation during the high-temperature exposure.

(ii) There is a duality between a measure  $\delta$  of the microbending sensitivity of a fiber and the fiber-compressive strain; viz., the effect of  $\delta$  on the characteristic added-loss response to compressive strain is a uniform contraction of the strain scale.

(iii) Points (i) and (ii) above result in a master curve of environmental added loss versus  $\delta$ -reduced fiber strain together with a plot of the strain-scale contraction factor versus  $\delta$ -value. We are thus led to the algorithm depicted schematically in Fig. 17. Therefore, when

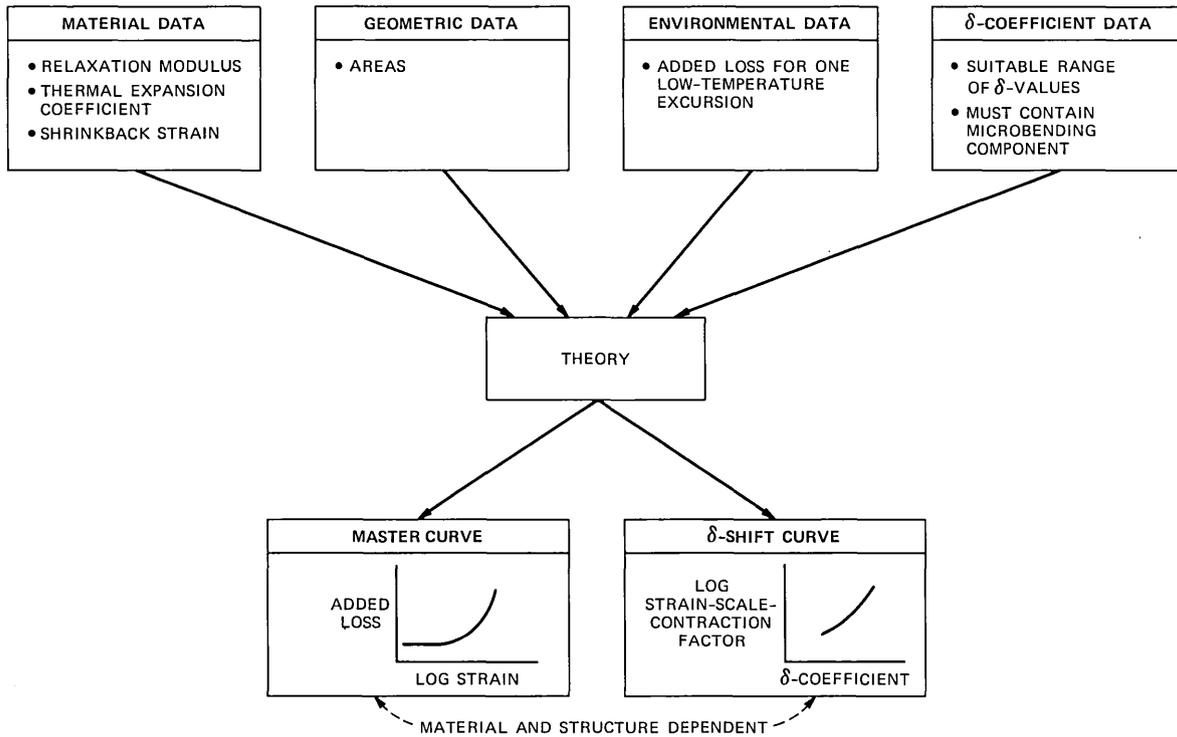


Fig. 17—Schematic of new environmental testing algorithm.

suitable material, geometric, environmental, and optical data are synthesized by way of our theory, we obtain a master curve and a  $\delta$ -shift curve that are sufficient to predict the environmental added loss for the candidate ribbon configuration for any desired environmental cycle. Notice that this environmental-testing scheme eliminates the necessity for long-term testing, only one low-temperature excursion being required.

(iv) There is a maximum value of  $\delta$  that a particular ribbon design can accommodate and still meet a given performance requirement. This maximum permissible  $\delta$ -value is an excellent criterion for making coating/ribbon-structure comparisons.

By application of these techniques, we have found:

(v) The critical material and geometric parameters in ASR design are the shrinkback of the polyester tape and the product  $\alpha EA$  of the coefficient  $\alpha$  of thermal expansion, the time- and temperature-dependent relaxation modulus  $E$ , and the area  $A$  associated with the fiber coating.

(vi) Any method of reducing the  $\alpha EA$  product of the coating such as introducing a soft, single coating or a dual system with a soft primary coating should improve environmental performance.

## VIII. ACKNOWLEDGMENTS

This work was performed jointly with P. F. Mahr. The authors greatly appreciate the invaluable assistance provided by C. J. Aloisio, M. J. Buckler, R. P. DeFabritis, B. J. Overton, D. N. Ridgway, and C. R. Taylor all of Bell Laboratories in Atlanta, Georgia.

## APPENDIX

### *Transverse Thermal Contraction of an Adhesive-Sandwich Ribbon*

Consider a section of an ASR containing two coated fibers as shown in Fig. 18. If the temperature is instantaneously lowered by an amount

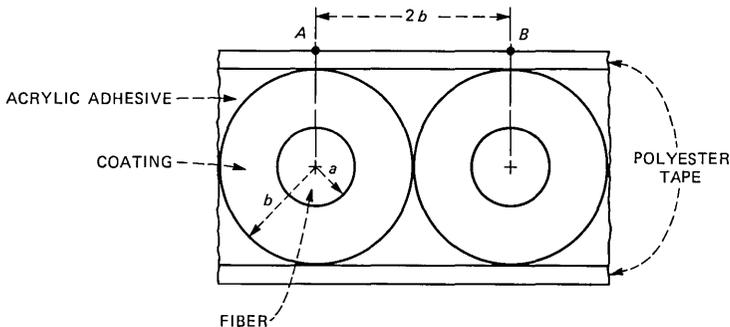


Fig. 18—Two-fiber portion of an ASR cross section prior to thermal contraction.

$|\Delta T|$  ( $\Delta T < 0$ ) from the temperature  $T_0$ , the section  $AB$  of tape would have an unstressed length  $l_{TA}$  of

$$l_{TA} = 2b(1 + \alpha_{TA}\Delta T), \quad (17)$$

where  $b$  is the outer radius of the coated fiber and  $\alpha_{TA}$  is the coefficient of transverse thermal expansion of the polyester tape.

A coated fiber would also shrink in diameter, the final radius  $b_f$  conforming to

$$b_f = b(1 + \alpha_*\Delta T), \quad (18)$$

where  $\alpha_*$  is a number somewhere between the coefficients of thermal expansion of the glass and the plastic coating. To estimate  $\alpha_*$  we appeal to linear thermoelasticity theory.

Denote by  $u$  and  $\sigma$  the radial displacement and stress fields (assumed axisymmetric) in the coating after the temperature is lowered. The stress field  $\sigma$  is to vanish at  $r = b$ , and it must match the stress in the glass fiber at  $r = a$ . The displacements must also match at this fiber-coating interface. Since the glass is many times stiffer than the plastic, we approximate the boundary conditions at  $r = a$  by supposing the glass-coating interface to shrink with the glass coefficient of thermal expansion. Thus,

$$u(a) = a\alpha_F\Delta T, \quad \sigma(b) = 0, \quad (19)$$

where  $\alpha_F$  is the coefficient of thermal expansion of the glass fiber.

When the axisymmetric solution to the requisite field equations<sup>9</sup> is substituted into (19), a pair of simultaneous, linear algebraic equations for two unknown constants result. Solving these equations and doing some algebra, one deduces that

$$u(b) = \alpha_*b\Delta T, \quad (20)$$

where

$$\alpha_* = \frac{4}{3b^2/a^2 + 1} [(b^2/a^2 - 1)\alpha + \alpha_F] \quad (21)$$

when Poisson's ratio of the plastic is taken as  $1/3$ .

For the case at hand, we use the values for  $a$  and  $b$  from the table in Fig. 7 and select  $\alpha$  and  $\alpha_F$  for the urethane-acrylate coating and the glass, respectively, from Table III. Substitution of these numbers into (21) gives

$$\alpha_* = 3.18 \times 10^{-5} \text{ } ^\circ F^{-1}, \quad (22)$$

which is 95 percent of the urethane-acrylate value.

If the final length  $l_{TA}$  of the tape segment is less than the final diameter  $2b_f$  of the coated fiber, pressure will be created between the

fibers and microbending loss may result. Otherwise, the fibers will separate. Thus, the critical quantity is

$$d = l_{TA} - 2b_f. \quad (23)$$

Incorporating eqs. (17) and (18) into (23) results in

$$d = 2b\Delta T(\alpha_{TA} - \alpha_*) . \quad (24)$$

For the case at hand  $\alpha_{TA} < \alpha_*$  since the transverse thermal expansion coefficient for the tape is even less than the axial value reported in Table III. We thus conclude from (24) that  $d$  is positive (recall that  $\Delta T < 0$ ).

Thus as the temperature is lowered the fibers separate and microbending loss from transverse contraction is therefore impossible. As the temperature is raised, however, the fibers obviously approach one another since at high temperatures  $\alpha_{TA}$  is still less than  $\alpha_*$ , but now  $\Delta T > 0$ . Moreover, there is transverse shrinkback at high temperatures. There is indeed environmental loss data which indicate increased loss at high temperatures.

## REFERENCES

1. D. Gloge, "Optical-Fiber Packaging and its Influence on Fiber Straightness and Loss," *B.S.T.J.*, 54, No. 2 (February 1975), pp. 245-262.
2. W. B. Gardner, "Microbending Loss in Optical Fibers," *B.S.T.J.*, 54, No. 2 (February 1975), pp. 457-465.
3. D. Marcuse, "Losses and Impulse Response of a Parabolic Index Fiber With Random Bends," *B.S.T.J.*, 52, No. 8 (October 1973), pp. 1423-1437.
4. M. J. Saunders and W. L. Parham, "Adhesive Sandwich Optical Fiber Ribbons," *B.S.T.J.*, 56, No. 6 (July-August 1977), pp. 1013-1014.
5. C. M. Miller, "Fiber-Optic Array Splicing With Etched Silicon Chips," *B.S.T.J.*, 57, No. 1 (January, 1978), pp. 75-90.
6. K. Inada, "A New Graphical Method Relating to Optical Fiber Attenuation," *Optics Commun.*, 19, No. 3 (December 1976), pp. 437-438.
7. R. A. Schapery, "Thermal Expansion Coefficients of Composite Materials Based on Energy Principles," *J. Composite Mater.*, 2 (July 1968), pp. 380-404.
8. R. A. Schapery, "Viscoelastic Behavior and Analysis of Composite Materials," *Composite Materials*, Vol. 2, *Mechanics of Composite Materials*, Ed. G. P. Sendeckyj, New York: Academic Press, 1974, pp. 85-168.
9. B. A. Boley and J. H. Weiner, *Theory of Thermal Stress*, New York: Wiley, 1960, p. 289.

## Adaptive Linearization of Power Amplifiers in Digital Radio Systems

By A. A. M. SALEH and J. SALZ

(Manuscript received October 5, 1982)

*High-frequency power amplifiers operate most efficiently at saturation, i.e., in the nonlinear range of their input/output characteristics. This phenomenon has traditionally dictated the use of constant envelope modulation methods for data transmission, resulting in circular signal constellations. This approach has inherently limited the admissible data rates in digital radio. In this paper we present a method for solving this problem without sacrificing amplifier power efficiency. We describe and analyze an adaptive linearizer that can automatically compensate for amplifier nonlinearity and thus make it possible to transmit multilevel quadrature amplitude modulated signals without incurring intolerable constellation distortions. The linearizer utilizes a real-time, data-directed, recursive algorithm for predistorting the signal constellation. Our analysis and computer simulations indicate that the algorithm is robust and converges rapidly from a blind start. Furthermore, the signal constellation and the average transmitted power can both be changed through software.*

### I. INTRODUCTION

Progress in high-speed data transmission over radio channels has lagged behind that of the voiceband channel. Inherent difficulties associated with implementing automatic equalizers are partially responsible, but the application of multilevel quadrature amplitude modulation (QAM) also has been inhibited by the amplitude (AM/AM) and phase (AM/PM) nonlinearities present in radio-frequency (RF) power amplifiers.

Recent work<sup>1</sup> has evolved design principles showing a possibility of substantial improvements in QAM performance over linear fading radio channels. A crucial obstacle to achieving these gains is the nonlinear distortion introduced by power amplifiers. Attempts to re-

alize high-speed data transmission over these channels force consideration of methods to cope with this nonlinear distortion.

One approach is to back off from saturation sufficiently so that the signal level is restricted to the linear range of amplification. The required amount of power back-off can be several decibels, resulting in an inefficient operation of the power amplifier. Moreover, the achievement of a given desired level of average transmitted power would require the use of a large, expensive, and high-power-consuming amplifier.

It has been realized<sup>2-7</sup> that some improvements in this regard can be obtained by using fixed signal predistortion circuits prior to amplification. Such circuits, however, cannot compensate for drifts in power amplifier nonlinearities caused by temperature changes, dc power variations, and component aging. These fluctuations can considerably degrade performance of systems employing constellations with large numbers of points, say 64 or higher, and the use of an adaptive technique is necessary in these applications.

Conceptually, the nonlinear distortion introduced by the power amplifier can be minimized at the receiver by an adaptive nonlinear equalizer. Such schemes have been proposed and studied in voiceband data transmission<sup>8</sup> and for filtered PSK signals operating over satellite channels.<sup>9</sup> This approach does not seem to be reasonable or necessary in our application, since the source of the nonlinearity is at the transmitter. Thus, it would appear logical to equalize the nonlinearity at the transmitter, where it occurs and where the transmitted bits are available.

Thus, this paper focuses on the problem of adaptive predistortion linearization. We describe a transmitter-based recursive algorithm for predistorting the signal constellation, thereby rendering a virtually linear transmitter. The algorithm operates in real time and is data directed. The predistortion is accomplished within a digital memory, which is used to generate the desired baseband signal. This maximizes the use of digital technology, and increases the reliability and flexibility of the system. For example, it is possible to change the signal constellation and the average transmitted power through software. Our treatment applies only to single-valued, memoryless nonlinearities.

The idea of adaptive predistortion of signal constellations has been previously suggested.<sup>10</sup> In this reference, the predistortion of each point of the constellation is accomplished by switching the RF or intermediate-frequency (IF) input signal to a separate path containing an adjustable diode attenuator and an adjustable diode phase shifter. The analog hardware required for such an implementation would be quite involved, especially for a large number of points. We have also found in the patent literature<sup>11</sup> a description of a digital adaptive

predistorter that appears to be similar to the one we present here. However, no detailed description of the algorithm or its behavior is provided.

Our basic ideas are described in Section II. Mathematical analysis is provided in Section III, and simulation results of the adaptive linearizer with finite-precision arithmetic are given in Section IV.

## II. GENERAL DESCRIPTION AND REQUIREMENTS OF OPERATION

A block diagram of a QAM transmitter with the proposed predistortion linearizer is shown in Fig. 1. A random access memory (RAM) contains the predistorted values of the in-phase and quadrature voltages of each point on the QAM constellation. A memory-lookup encoder obtains each input data symbol and generates the RAM addresses of the desired signal point. The corresponding stored, predistorted voltage values are converted to analog voltages using a pair of digital-to-analog (D/A) converters. These voltages drive a quadrature modulator, which generates the desired predistorted RF signal for the duration of the input symbol. That signal is then amplified, filtered, and transmitted. This part of the linearizer is similar in operation to a recently proposed memory-based encoder described in Ref. 12.

To update the RAM information, the amplifier output is sampled by a directional coupler and demodulated using the same local oscillator used for the modulator, which eliminates the need for carrier recovery. The output in-phase and quadrature voltages of the demodulator are converted to digital form using a pair of analog-to-digital (A/D) converters. A linearizing processor, which is the heart of the linearizer, receives this information, compares it with the input data, and computes the resulting error. A recursive algorithm, which is discussed in the next section, uses the error to update the voltage values in the RAM corresponding to the particular data point under consideration. Note that each point on the signal constellation is treated separately. Thus, the linearizer can support any desired constellation.

The memory-lookup encoder and the D/A converters have to operate, of course, at the full signaling rate. However, the linearizing processor and the A/D converters can operate at a much reduced rate, since the updating process is only needed to compensate for drifts that occur on a much slower time scale than the data rate.

We now emphasize a crucial point. The proper operation of the linearizer as described above assumes that the amplifier is memoryless, and requires that the signal not be filtered before the power amplifier. Thus, all pulse shaping and filtering must be performed by the combination of the post-amplifier, RF bandpass filter, and the receiver filters. The former filter should be designed just to meet FCC (or

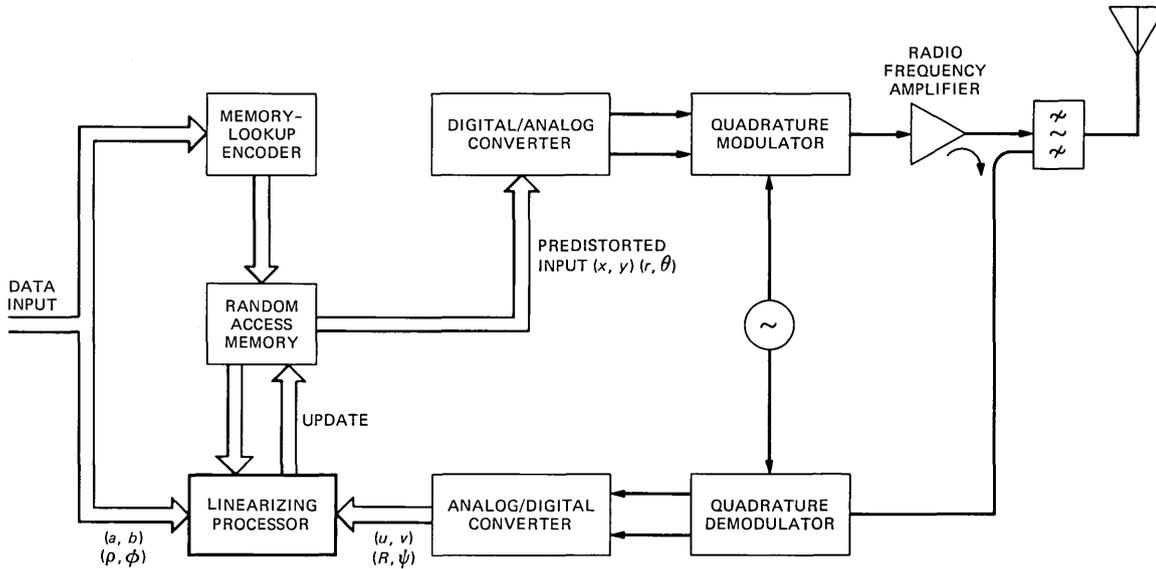


Fig. 1—Schematic representation of the adaptive, digital, predistortion linearizer.

other) emission rules for square input pulses, as described in Ref. 13. Such an implementation may require an automatic equalizer at the receiver to eliminate residual intersymbol interference. It was shown<sup>1</sup> that, even with ideal filtering, adaptive equalization would still be necessary to compensate for multipath fading using QAM signals with large numbers of levels. Thus, the elimination of pre-amplifier pulse shaping appears to be a mild constraint.

Because of the more stringent requirement on the post-amplifier, RF bandpass filter, its loss would of course be increased over that of the conventional case where preamplifier filtering does the spectrum shaping. However, computations<sup>14</sup> based on practical filters<sup>15</sup> operating in the 6-GHz band show that the RF filter loss would only increase from about 0.5 dB to 1.5 dB. However, we will see in Section IV that the linearizer in our system would allow the operation of the power amplifier to approach saturation. This would result in several decibels of power increase, which would more than compensate for the one-decibel increase in the filter loss.

### III. THE RECURSIVE ALGORITHM

As we already mentioned, our approach is applicable to any signal constellation. However, for clarity of exposition, we restrict our analysis to a rectangular constellation. Referring to Figs. 1 and 2, we denote a point on the rectangular QAM constellation by the complex number

$$a + ib = \rho e^{i\phi},$$

where, for an  $L^2$ -level system,  $a$  and  $b$  assume values on the lattice  $\pm 1 \pm 3 \pm 5 \pm \dots \pm (L - 1)$ . We denote the predistorted point by the

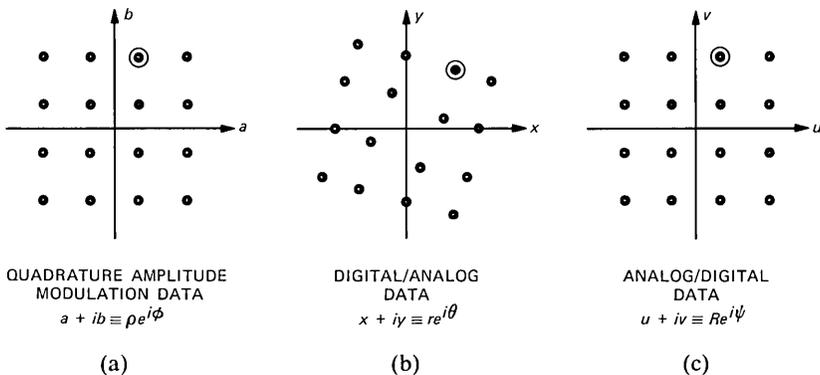


Fig. 2—The signal constellations at various points in the linearizer of Fig. 1. (a) Quadrature amplitude modulation data. (b) Digital/analog data. (c) Analog/digital data.

complex number

$$x + iy = re^{i\theta}.$$

A sequence of these points amplitude and phase modulates a carrier at frequency  $f_c$  and the resulting signal,

$$S(t) = \sum_m r^{(m)} p(t - mT) e^{i[2\pi f_c t + \theta^{(m)}]}, \quad (1)$$

is applied to the amplifier. In (1),  $p(t)$  is a rectangular pulse,  $1/T$  is the signaling rate, and  $r^{(m)}$ ,  $\theta^{(m)}$  represent the amplitude and phase of the  $m$ th data symbol.

The amplifier is customarily represented by a pair of memoryless nonlinear functions.<sup>16,17</sup> The amplitude function,  $A(r)$ , causes amplitude distortion (AM/AM) and the phase function,  $\Phi(r)$ , causes amplitude to phase conversion (AM/PM). Thus the signal, (1), after amplification becomes

$$S_o(t) = \sum_m A(r^{(m)}) p(t - mT) e^{i[2\pi f_c t + \theta^{(m)} + \Phi(r^{(m)})]}. \quad (2)$$

Figure 3 shows sketches of typical curves  $A(r)$  and  $\Phi(r)$  for a traveling-wave tube (TWT) power amplifier.

We remark that if the functions  $A(\cdot)$  and  $\Phi(\cdot)$  were known exactly we could choose transformations  $g(\cdot)$  and  $h(\cdot)$  from points  $(\rho, \phi)$  to

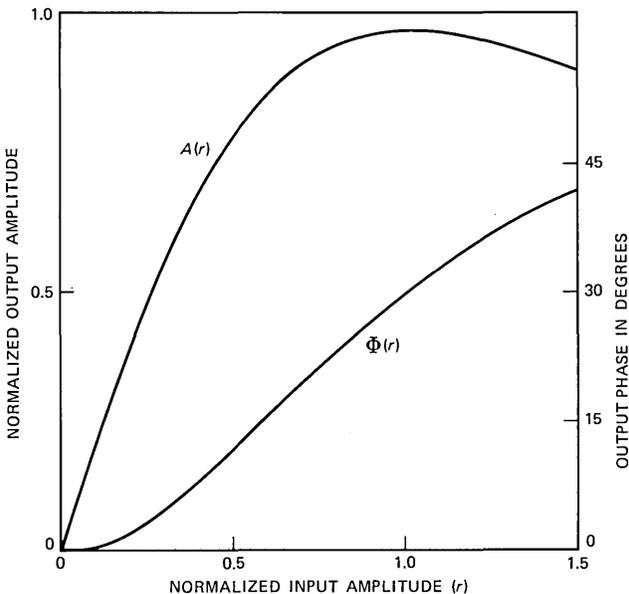


Fig. 3—Typical example of the input/output amplitude and phase characteristics of a TWT amplifier.

$(r, \theta)$  so that

$$A(r) = A[g(\rho)] = G\rho, \quad (3)$$

where

$$r = g(\rho)$$

and

$$\Phi(r) + h(\phi, \rho) = \phi + \zeta, \quad (4)$$

where

$$\theta = h(\phi, \rho).$$

The constant gain,  $G$ , in (3) is some desired gain, taking into account the linear gain of the power amplifier and the coupling factor of the sampling directional coupler. The fixed phase shift,  $\zeta$ , in (4) is arbitrary and can be set to zero without loss of generality.

Finally, let the measured data point from the A/D converter be

$$u + iv = Re^{i\psi}.$$

The object of the predistortion is to find a solution

$$r = \hat{r}$$

and

$$\theta = \hat{\theta},$$

such that

$$Re^{i\psi} = G\rho e^{i\phi}. \quad (5)$$

We now describe an iteration procedure that converges to (5).

Since data are usually scrambled, a specific data point  $(\rho, \phi)$  will occur at random. Let  $(r_n, \theta_n)$  be the predistorted RAM data point and  $(R_n, \psi_n)$  be the measured data point at the  $n$ th time the desired data point  $(\rho, \phi)$  occurs. The measured output radius  $R_n$  is then

$$R_n = A(r_n) + \nu_n \quad (6a)$$

while the measured angle is

$$\psi_n = \Phi(r_n) + \theta_n + \mu_n, \quad (6b)$$

where  $\nu_n$  and  $\mu_n$  are zero-mean measurement errors. If the measurements were perfect and noiseless, we would simply solve the following two nonlinear equations

$$f(\hat{r}) = A(\hat{r}) - G\rho = 0 \quad (7)$$

and

$$\Phi(\hat{r}) + \hat{\theta} - \phi = 0 \quad (8)$$

for  $\hat{r}$  and  $\hat{\theta}$ .

A great number of iterative procedures are known for solving the nonlinear equations given in (7) and (8). In the presence of measurement noise (with convex functions  $A$  and  $\Phi$ ), stochastic approximation algorithms<sup>18</sup> provide efficient methods. So, if one chooses step sizes  $\alpha_n$  and  $\beta_n$ , which behave as  $o(1/n)$ , the following recursions are known to converge to the true solutions  $\hat{r}$  and  $\hat{\theta}$  in a mean-square sense:

$$r_{n+1} = r_n - \alpha_n(R_n - G\rho) \quad (9a)$$

$$\theta_{n+1} = \theta_n - \beta_n(\psi_n - \phi). \quad (9b)$$

To provide a rationale for the above recursions we analyze, in some detail, the behavior of (9a). The behavior of (9b) is similar. Since  $f(X)$ , eq. (7), is continuous on  $(X_0, X_{\max})$  and if, by hypothesis, the derivative  $f'(X)$  of  $f(X)$  exists on this interval, then there is a  $\xi$  such that

$$\begin{aligned} X_0 &\leq \xi \leq X_{\max} \\ f'(\xi) &= \frac{f(X_{\max}) - f(X_0)}{X_{\max} - X_0}. \end{aligned} \quad (10)$$

Applying this mean-value theorem to  $f(r_n)$ , eq. (7), we get

$$f(r_n) = A(r_n) - G\rho = f(\hat{r}) + f'(\xi_n)(r_n - \hat{r}), \quad (11)$$

where  $\xi_n$  lies between  $r_n$  and  $\hat{r}$ . Substituting this into (9a) we get

$$r_{n+1} - \hat{r} = r_n - \hat{r} - \alpha_n[A'(\xi_n)(r_n - \hat{r}) + \nu_n]. \quad (12)$$

In (12) we made use of the fact that  $f(\hat{r}) = 0$  and subtracted  $\hat{r}$  from both sides of the equation.

Letting  $\alpha_n A'(\xi_n) = \gamma_n$ ,  $r_n - \hat{r} = \delta_n$  and then iterating (12) we obtain

$$\begin{aligned} \delta_{n+1} &= (1 - \gamma_n)\delta_n - \frac{\gamma_n \nu_n}{A'(\xi_n)} \\ &= \delta_1 \prod_{i=1}^n (1 - \gamma_i) - \sum_{k=1}^n \frac{\nu_k}{A'(\xi_k)} \frac{\gamma_k}{1 - \gamma_k} \prod_{i=k}^n (1 - \gamma_i). \end{aligned} \quad (13)$$

We now compute the mathematical expectation of  $\delta_n$  and the variance:

$$\begin{aligned} \bar{\delta}_{n+1} &= E\{\delta_{n+1}\} = E\{\delta_1\} \prod_{i=1}^n (1 - \gamma_i) \\ &\leq E\{\delta_1\} e^{-\sum_{i=1}^n \gamma_i} \end{aligned} \quad (14)$$

and

$$\sigma_{n+1}^2 = E\{\delta_{n+1} - \bar{\delta}_{n+1}\}^2 = \sigma_\nu^2 \sum_{k=1}^n \frac{\gamma_k^2}{[A'(\xi_k)]^2 (1 - \gamma_k)^2} \prod_{i=k}^n (1 - \gamma_i)^2, \quad (15)$$

where  $\sigma_v^2 = E\{\nu_n^2\}$ . We see from (14) and (15) that rapid convergence of the algorithm is critically dependent on the step-size sequence,  $\gamma_n$ , which in turn depends on the derivative of the nonlinearity in the neighborhood of the solution. While the derivative is unknown, in most applications it can be bounded away from zero and this makes it possible to estimate the best step-size sequence. We see from (14) that, even in the absence of noise, the algorithm is guaranteed to converge to the true solution only if the step-size series diverges, i.e.,

$$\sum_{i=1}^n \gamma_i = \sum_{i=1}^n \alpha_i A'(\xi_i) \rightarrow \infty, \quad n \rightarrow \infty. \quad (16)$$

If  $\alpha_i$  and  $A'(\xi_i)$  are restricted to be positive, (16) is equivalent to requiring  $\sum_{i=1}^n \alpha_i$  to diverge. So it follows that the structure of the sequence can be of the form

$$\alpha_n = \frac{a}{n^\eta}, \quad (17)$$

where  $a$  is a positive constant and  $0 \leq \eta \leq 1$ . It can be shown that for this choice of  $\alpha_n$ , and for  $c = aA'(\hat{r})$ ,

$$\frac{E\{\delta_{n+1}\}}{E\{\delta_1\}} = e^{-\frac{cn^{(1-\eta)}}{1-\eta} + cq(\eta)}, \quad (0 \leq \eta < 1) \quad (18)$$

and

$$[A'(\hat{r})]^2 \frac{\sigma_{n+1}^2}{\sigma_v^2} = \frac{c}{2n^\eta}, \quad (0 < \eta < 1), \quad (19)$$

where

$$q(\eta) = \lim_{n \rightarrow \infty} \left\{ \frac{n^{1-\eta}}{1-\eta} - \sum_{k=1}^n \frac{1}{k^\eta} \right\};$$

e.g.,  $q(0) = 0$  and  $q(1/2) \approx 1.46$ . Table I shows the behavior of the statistics for the special cases  $\eta = 0$  and  $\eta = 1$ , where (18) and (19) are not applicable.

Table I—Behavior of statistics for  $\eta = 0$  and  $\eta = 1$

$\eta$	$c = aA'(\hat{r})$	$\frac{\bar{\delta}_{n+1}}{\bar{\delta}_1}$	$\frac{\sigma_{n+1}^2}{\sigma_v^2} [A'(\hat{r})]^2$
0	$0 < c < 2$	$(1-c)^n < e^{-cn}$	$\frac{c}{2-c} [1 - (1-c)^{2n}] \sim \frac{c}{2-c}$
1	$c > \frac{1}{2}$	$e^{-0.577/n^c}$	$\frac{c^2}{2c-1} \times \frac{1}{n}$
1	1	$e^{-0.577/n}$	$\frac{1}{n}$
1	1/2	$\frac{0.577}{e^{2/\sqrt{n}}}$	$\frac{\ln(n)}{4n}$

It is noted from eqs. (17) through (19) and Table I that the choice of a fixed step size ( $\eta = 0$ ) gives the fastest convergence of the mean,  $\bar{\delta}_{n+1}$ , but results in a finite variance,  $\sigma_{n+1}^2$ , as  $n \rightarrow \infty$ . On the other hand, a variable, progressively smaller step size ( $\eta \neq 0$ ) gives a slower convergence of the mean, but results in a variance approaching zero as  $n \rightarrow \infty$ . Thus, a fixed step size should be used if the measurement error variance,  $\sigma_v^2$ , is very small, and a variable step size should be used if the variance is large. Actually, when finite-precision arithmetic is employed, the choice of a progressively smaller step size can lead to large errors.<sup>19</sup> Thus, in our particular application where the measurement error can be made small, and where the use of finite-precision arithmetic is necessary, a fixed step size is more suitable.

We note that our algorithm, (9), is based on the polar representation of the data symbols, while the hardware implementation of Fig. 1 is based on the rectangular representation. Thus, the linearizing processor is required to convert back and forth between the two representations. One could avoid this conversion by replacing (9) by the rectangular-based algorithm

$$x_{n+1} + iy_{n+1} = x_n + iy_n - (\alpha_n + i\beta_n)[u_n + iv_n - G(a + ib)].$$

Unfortunately, however, the convergence of this algorithm is not guaranteed, even for some well-behaved amplifier characteristics.

#### IV. SIMULATIONS

Here we present the results of computer simulations of the algorithm, (9), with fixed step sizes and finite-precision arithmetic. The precision is limited by the finite number of bits of the D/A and A/D converters (Fig. 1). As an example, we consider a TWT amplifier with normalized amplitude and phase nonlinearities of the form<sup>17</sup>

$$A(r) = \frac{2r}{1 + r^2}, \quad (20a)$$

$$\Phi(r) = 60^\circ \frac{r^2}{1 + r^2}, \quad (20b)$$

which are sketched in Fig. 3. Note that at saturation,  $r = 1$ ,  $A(r) = 1$ , and  $\Phi(r) = 30^\circ$ . Figure 4 shows the severe distortion of the output constellation obtained with such an amplifier for a 64-QAM input signal driven with its corner point at saturation.

Figures 5, 6, and 7 show the simulated results of the output signal constellations after the application of (9) for three different cases (explained below). In all cases the amplifier drive is maintained at saturation. The four quadrants in each figure represent different combinations of the numbers of bits of the D/A and A/D converters, as indicated in Table II.

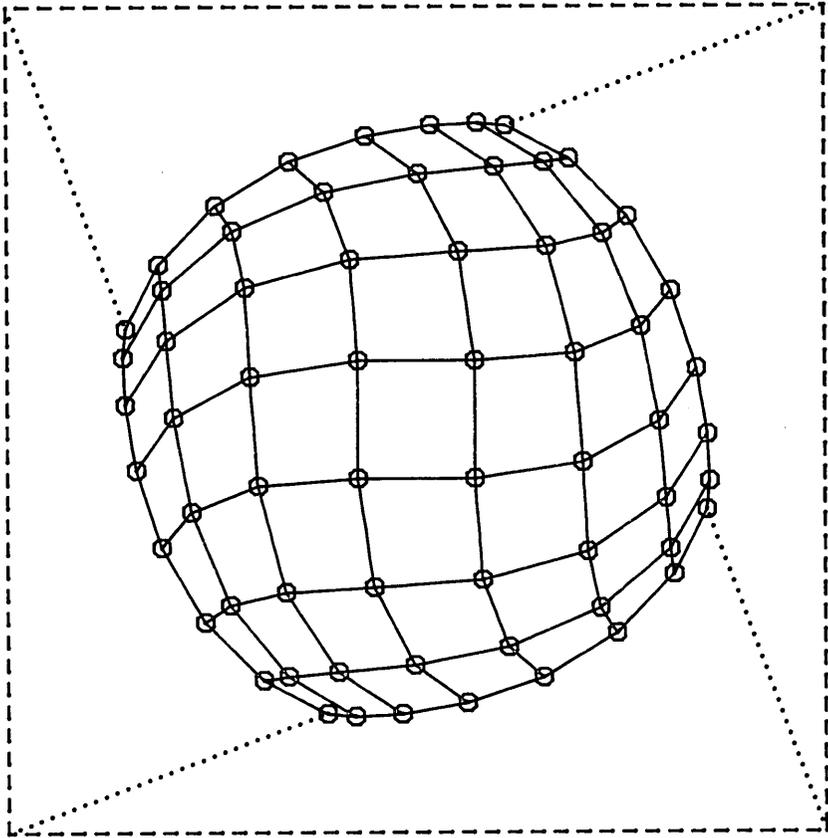


Fig. 4—The distortion of the signal constellation obtained with the amplifier of Fig. 3 for a 64-QAM input signal with its corner points at saturation.

Figure 5 corresponds to the case of no measurement error, i.e.,  $\nu_n = \mu_n = 0$  in (6). The step sizes used in (9) are  $\alpha_n = 0.5$  and  $\beta_n = 1.0$ . Since the normalized small-signal gain,  $A'(r)$ , of the amplifier is about 2, the choice of  $\alpha_n = 0.5$  results in a value for  $c$  of about 1. This results in the fastest convergence of (9a), as can be seen from the first row of Table I. Similarly, the choice of  $\beta_n = 1.0$  results in the fastest convergence of (9b) since that equation is linear in  $\theta$ . The initial guesses of  $r$  and  $\theta$  for each constellation point in Fig. 5 were chosen at random, i.e., from a blind start. The results indicated in the figure are those after 25 iterations for each constellation point. The point scatter shown is entirely due to the finite resolution of the D/A and A/D converters since no measurement errors were assumed.

It is clear from Fig. 5 that the use of 8 bits for both the D/A and A/D converters (fourth quadrant) gives quite an acceptable performance for 64 QAM. The use of 9 bits for each converter (first quadrant)

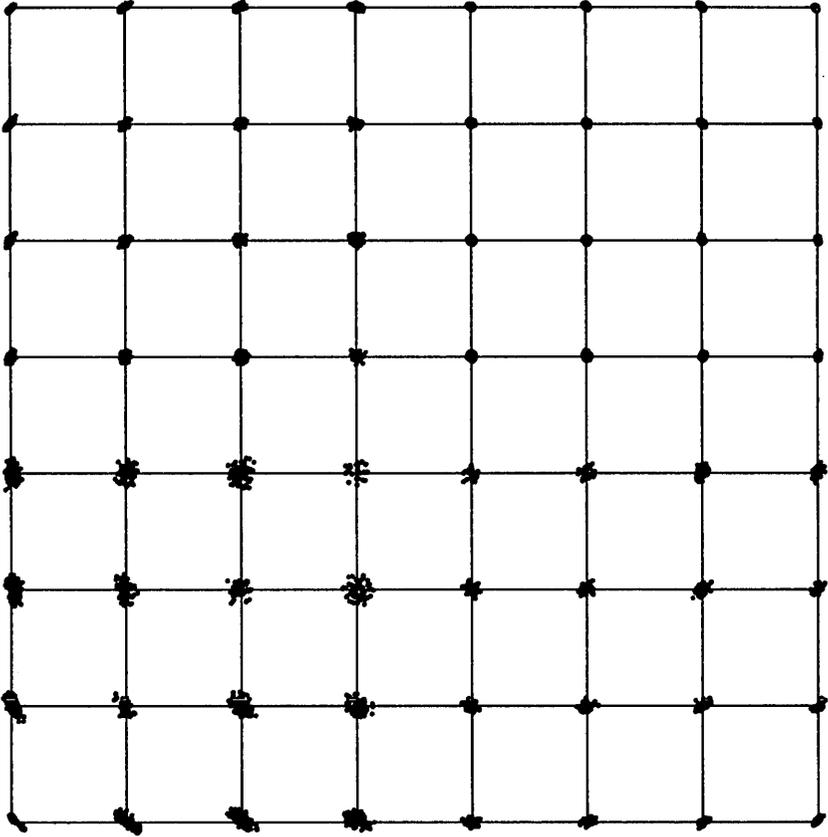


Fig. 5—Simulated results of the output signal constellation after 25 iterations with no measurement errors, and for step sizes  $\alpha_n = 0.5$  and  $\beta_n = 1.0$ . The number of bits of the D/A and A/D converters are different in each quadrant as indicated in Table 2.

results in an almost perfect output constellation. In general, for  $L^2$ -QAM, where  $L = 2^M$ , D/A and A/D converters with  $(M + 5)$  bits are needed for acceptable performance, and  $(M + 6)$  or more bits are needed for almost perfect performance.

In Fig. 6, the step sizes employed are  $\alpha_n = 0.5$  and  $\beta_n = 1.0$ , as in Fig. 5. However, a measurement error was introduced that is equivalent to a 30-dB signal-to-noise ratio at the corner points of the constellation. With the normalization used for  $A(r)$  in (20a), this noise corresponds to  $\sigma_v^2 = A^2(r)\sigma_\mu^2 = 0.0005$ , where  $\sigma_v^2$  and  $\sigma_\mu^2$  are the variances of the errors  $\nu_n$  and  $\mu_n$  defined in (6). It is clear from Fig. 6 that such a level of noise, in combination with the large step sizes employed, gives unacceptable results.

In Fig. 7, the same noise level as that of Fig. 6 is employed. However, the step sizes were reduced by a factor of 10, i.e.,  $\alpha_n = 0.05$  and  $\beta_n =$

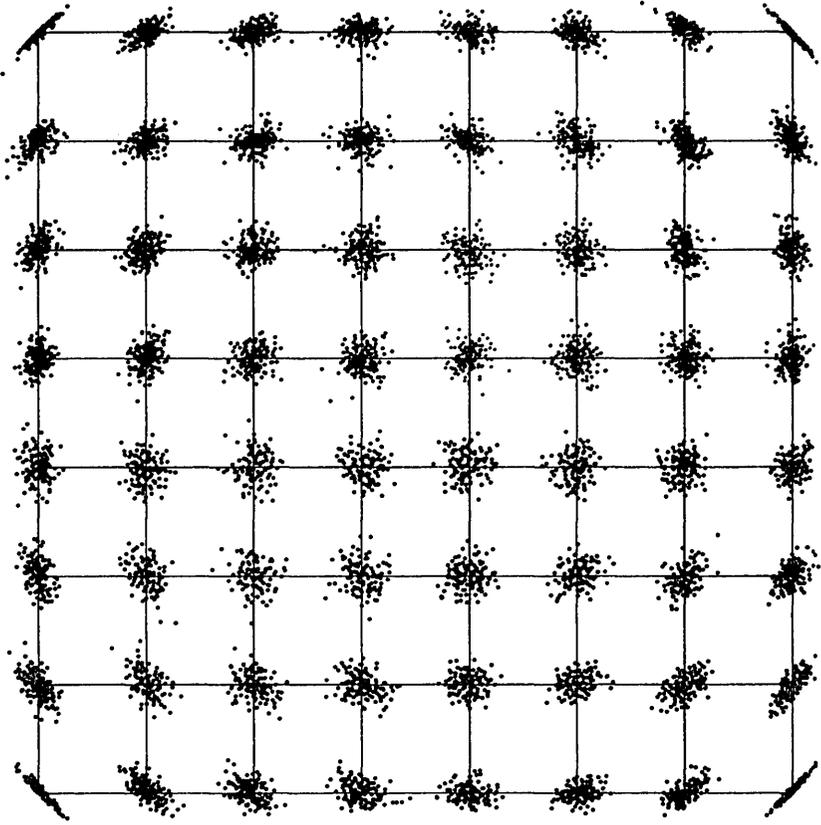


Fig. 6—Same as Fig. 5, but with a measurement error equivalent to 30-dB of signal-to-noise ratio at the corner points of the constellation.

0.1, resulting in a greatly improved performance. About 100 steps of iteration were needed in this case to reach convergence. Note that in spite of the measurement noise, the performance in the third quadrant of Fig. 7 is better than the corresponding performance in Fig. 5, where no noise is present. This is due to the reduced step sizes, and to the fact that the noise tends to smooth out quantization errors.

## V. SUMMARY AND CONCLUSIONS

We have proposed and analyzed a transmitter-based, adaptive linearizer, which automatically compensates for power amplifier nonlinearity in digital radio systems employing multilevel quadrature amplitude modulation. The linearizer utilizes a real-time, data-directed recursive algorithm for predistorting the signal constellation. The algorithm is robust and results in rapid convergence, even from a blind

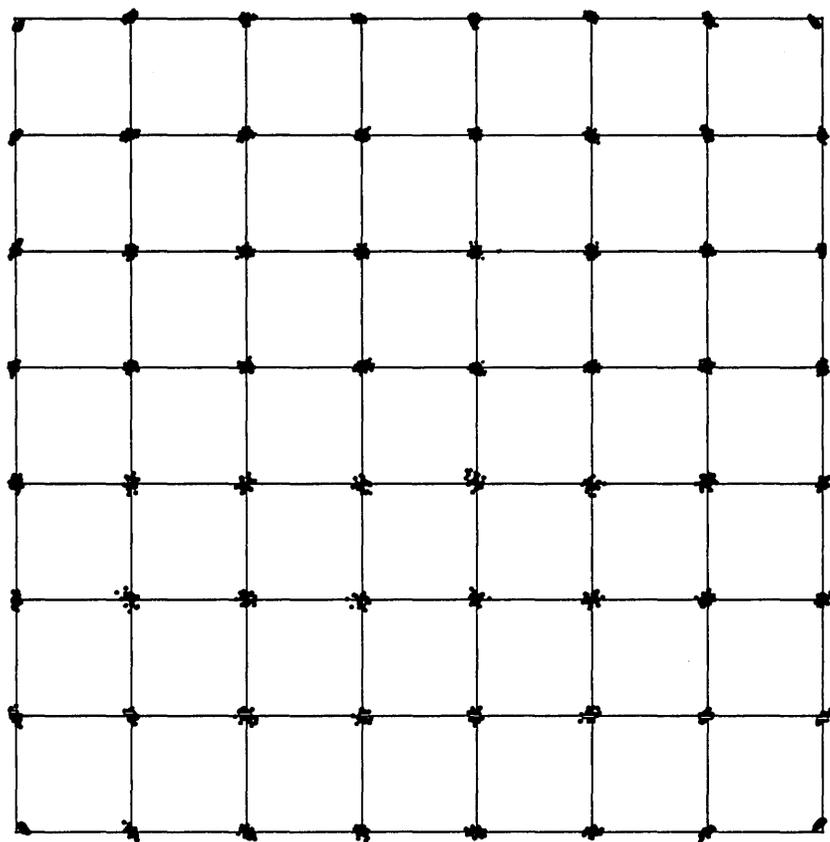


Fig. 7—Same as Fig. 6, but with the step sizes reduced by a factor of 10, i.e.,  $\alpha_n = 0.05$  and  $\beta_n = 0.1$ . About 100 iteration steps were needed for convergence.

Table II—Number of bits of the D/A and A/D converters for the four quadrants of Figs. 5, 6, and 7

Quadrant Number	Number of Bits of the D/A Converter	Number of Bits of the A/D Converter
1	9	9
2	9	7
3	8	6
4	8	8

start. The digital nature of the linearizer allows both the signal constellation and the average transmitted power to be changed through software.

The proposed linearizer is only suitable for amplifiers with essen-

tially memoryless and single-valued nonlinearities. Consequently, pulse shaping is not permitted before amplification. In our approach the task of pulse shaping is relegated to the combination of the transmit RF filter, a receiver-based filter, and an automatic equalizer. The latter is presumed to be required anyway to deal with multipath fading, especially for signals having a large number of levels.

Our principal conclusion is that high-power RF amplifiers can operate at saturation, where they are most efficient, provided that an adaptive predistortion linearizer is used. We have demonstrated that such a system is feasible and have provided a theory and a methodology for assessing its performance.

## REFERENCES

1. G. J. Foschini and J. Salz, "Digital Communications over Fading Radio Channels," *B.S.T.J.*, 62, No. 2, Part 1 (February 1983), pp. 429-56.
2. R. P. Hecken and R. C. Heidt, "Predistortion Linearizer of the AR6A Transmitter," *ICC '80 Conf. Rec.*, 2 (June 1980), pp. 33.1.1-6.
3. T. Nojima and Y. Okamoto, "Predistortion Nonlinear Compensator for Microwave SSB-AM System," *ICC '80 Conf. Rec.*, 2 (June 1980), pp. 33.2.1-6.
4. A. Egger, M. Horn, and T. Vien, "Broadband Linearization of Microwave Power Amplifiers," *Proc. of the 10th European Microwave Conf.*, September 1980, pp. 490-4.
5. C. Brenson, M. Palazo, and R. Neyer, "Linearizing T.W.T. Amplifiers in Satellite Transponders—System Aspects and Practical Implementation," *Proc. of the AIAA 8th Commun. Satellite Conf.*, April 1980, pp. 80-9.
6. G. Sato, H. Shibazaki, T. Asai, and T. Kurokawa, "A Linearizer for Satellite Communications," *ICC '80 Conf. Rec.*, 2 (June 1980), pp. 33.3.1-5.
7. G. Satoh and T. Mizuno, "Impact of a New TWTA Linearizer Upon QPSK/TDMA Transmission Performance," *IEEE J. Select. Areas in Comm.*, special issue on Digital Satellite Communications, *SAC-1*, No. 1 (January 1983), pp. 39-45.
8. D. D. Falconer, "Adaptive Equalization of Channel Nonlinearities in QAM Data Transmission Systems," *B.S.T.J.*, 57, No. 7 (September 1978), pp. 2589-611.
9. S. Benedetto and E. Biglieri, "Nonlinear Equalization of Digital Satellite Channels," *IEEE J. Select. Areas in Comm.*, special issue on Digital Satellite Communications, *SAC-1*, No. 1 (January 1983), pp. 57-62.
10. W. J. Weber, III, "The Use of TWT Amplifiers in *M*-Ary Amplitude and Phase-Shift Keying Systems," *ICC '75 Conf. Rec.*, 3 (June 1975), 36.17-21.
11. R. C. Davis and W. Boyd, "Adaptive Predistortion Technique for Linearizing a Power Amplifier for Digital Data Systems," U.S. Patent 4,291,277, issued September 22, 1981.
12. T. L. Osborne and P. D. Karabinis, unpublished work.
13. L. J. Greenstein and D. Vitello, "Required Transmit Filter Bandwidth in Digital Radio Systems," *IEEE Trans. Comm.*, *COM-29*, No. 9 (September 1981), pp. 1405-8.
14. H. Wang and C. Ren, private communications.
15. J. K. Plourde and C. Ren, "Application of Dielectric Resonators in Microwave Components," *IEEE Trans. Microwave Theory Tech.*, *MTT-29*, No. 8 (August 1981), pp. 754-70.
16. A. L. Berman and C. H. Mahle, "Nonlinear Phase Shift in Traveling-Wave Tubes as Applied to Multiple Access Communications Satellites," *IEEE Trans. Comm. Tech.*, *COM-18*, No. 1 (February 1970), pp. 37-48.
17. A. A. M. Saleh, "Frequency-Independent and Frequency-Dependent Nonlinear Models of TWT Amplifiers," *IEEE Trans. Comm.*, *COM-29*, No. 11 (November 1981), pp. 1715-20.
18. A. E. Albert and L. A. Gardner, Jr., "Stochastic Approximation and Nonlinear Regression," *Research Monograph No. 42*, Cambridge, MA: MIT Press, 1967.
19. R. D. Gitlin, J. E. Mazo, and M. G. Taylor, "On the Design of Gradient Algorithms for Digitally Implemented Adaptive Filters," *IEEE Trans. Circuit Theory*, *CT-20*, No. 2 (March 1973), pp. 125-36.



# An Introduction to the Application of the Theory of Probabilistic Functions of a Markov Process to Automatic Speech Recognition

By S. E. LEVINSON, L. R. RABINER, and M. M. SONDHI

(Manuscript received September 10, 1982)

*In this paper we present several of the salient theoretical and practical issues associated with modeling a speech signal as a probabilistic function of a (hidden) Markov chain. First we give a concise review of the literature with emphasis on the Baum-Welch algorithm. This is followed by a detailed discussion of three issues not treated in the literature: alternatives to the Baum-Welch algorithm; critical facets of the implementation of the algorithms, with emphasis on their numerical properties; and behavior of Markov models on certain artificial but realistic problems. Special attention is given to a particular class of Markov models, which we call "left-to-right" models. This class of models is especially appropriate for isolated word recognition. The results of the application of these methods to an isolated word, speaker-independent speech recognition experiment are given in a companion paper.*

## I. INTRODUCTION

It is generally agreed that information in the speech signal is encoded in the temporal variation of its short-duration power spectrum. To decode the signal, then, requires techniques for both estimation of power spectra and tracking their changes in time. This paper is concerned with the application of the theory of probabilistic functions of a (hidden) Markov chain to modeling the inherent nonstationarity of the speech signal for the purposes of automatic speech recognition (ASR).

The use of hidden Markov models for ASR was proposed by Baker<sup>1,2</sup> and, independently, by a group at IBM.<sup>3-12</sup> The theory on which their work rests is due to Baum et al.<sup>13-17</sup> Its first appearance in the literature occurred several years before Baker's studies and has since been

explored in some detail.<sup>18,19</sup> Our previous work in ASR has used temporal alignment procedures based on dynamic programming techniques,<sup>20</sup> and we hoped that through studying the new (to us) body of material we could improve the performance and/or capabilities of our present ASR systems.

Our initial goal, therefore, was to understand the theory of hidden Markov models sufficiently well to enable us to implement a new ASR system that could be compared directly to our existing ones. We have, in fact, been able to accomplish that goal, and a description and the results of our experiments are reported in a companion paper.<sup>21</sup> In the course of our studies, we have collected and integrated a number of loosely related mathematical techniques pertinent to Markov modeling. We have also modified and adapted these techniques to the specific ASR problems we wished to study. Our purpose in writing this tutorial, then, is to present this synthesis in a way that will be enlightening to those not familiar with the theory of hidden Markov models. We also hope that this treatment will provide for a better understanding of our companion paper. Finally, we hope to make the presentation general enough so that the theory is seen to be applicable to more than the problem of ASR.

We shall proceed as follows. We begin by defining probabilistic functions of a (hidden) Markov<sup>22</sup> chain and then show how they may be used in a natural way to model the speech signal. Once this is done, our task is reduced to solving two specific and well-defined mathematical problems: (i) computing the parameters of a proposed model conditioned on a sequence of observations assumed to have been generated by the model, and (ii) calculating the probability that a given set of observations was produced by a particular model.

First we review the solution to these problems as originally given by Baum,<sup>13</sup> who treated them as problems in statistical estimation. Lest the problems be too narrowly construed, we look at them as problems of classical constrained optimization. This allows us to give a partial geometrical interpretation to the Baum-Welch algorithm and to relate it to other studies of the problem by Baum and Eagon<sup>14</sup> and Baum and Sell.<sup>17</sup> It also makes clear that there are other methods of solution available that may, in certain instances, have advantages over the Baum-Welch algorithm. Finally, we discuss the dynamic programming algorithm of Viterbi<sup>23</sup> as an alternative to the so-called "forward-backward" method of Baum<sup>13</sup> for computing the probability of a sequence of observations conditioned on a specific model.

The treatment of these problems in the existing literature, and as recounted here, can lull a prospective user of the theory into a false sense of security. The equations look innocuous enough but, in reality, they overlook two problems that, though uninteresting from a theo-

retical standpoint, are of great significance for a robust implementation. We believe it is worthwhile to address, first, a numerical problem arising from the evaluation of certain frequently occurring algebraic expressions, and then an experimental difficulty precipitated by the inescapable reality of finite training-data set.

The numerical problem arises because, regardless of the method of solution chosen, one is required to evaluate a product of stochastic matrices involving a number of factors proportional to the number of observations. In any real computer, this will ultimately result in underflow. Fortunately, the computation can be scaled using a technique that subsequently will be seen to have some very useful properties.

The problem of insufficient training data can be ameliorated by changing the constraints on the optimization problem. This can be simply and directly accomplished in the classical methods. We show that the Baum-Welch algorithm, too, can be modified to produce the same result. Both of these methods appear to be simpler in implementation than the technique proposed by Jelinek and Mercer.<sup>9</sup>

Finally, under the heading of implementational considerations, we discuss techniques for model averaging. These can be used both for block processing of observations in case one is subject to storage limitations, and for increasing model stability under some circumstances.

The speech recognition experiment that we had in mind was on a speaker-independent, isolated word recognition system with a small vocabulary. Oddly enough, this is a simpler task than those to which the theory had already been applied by Baker<sup>1</sup> and the IBM group.<sup>3-12</sup> Perhaps our choice of a manageable problem is responsible for the degree of success reported in the companion paper.<sup>21</sup> We determined that for our ASR task it is advantageous to use a particular kind of hidden Markov model, which we call a left-to-right model. In such a model there are strong temporal constraints on the Markov chain. First, any state, once left, cannot be later revisited. Second, there is a final absorbing state in which all observation sequences are assumed to terminate. These restrictions on the sequences affect both parameter estimation and probability computation procedures. We show how the Baum-Welch algorithm, the Viterbi algorithm, and the classical methods can all be adapted for use with left-to-right models.

We conclude our presentation with several sample solutions to some artificial but nontrivial problems that illustrate concepts treated in the foregoing discussion.

## II. A REVIEW OF THE THEORY

A probabilistic function of a (hidden) Markov chain is a stochastic

process generated by two interrelated mechanisms, an underlying Markov chain having a finite number of states, and a set of random functions, one of which is associated with each state. At discrete instants of time, the process is assumed to be in some state and an observation is generated by the random function corresponding to the current state. The underlying Markov chain then changes states according to its transition probability matrix. The observer sees only the output of the random functions associated with each state and cannot directly observe the states of the underlying Markov chain; hence the term hidden Markov model.

In principle, the underlying Markov chain may be of any order and the outputs from its states may be multivariate random processes having some continuous joint probability density function. In this discussion, however, we shall restrict ourselves to consideration of Markov chains of order one, i.e., those for which the probability of transition to any state depends only upon that state and its predecessor. We shall also limit the discussion to processes whose observations are drawn from a discrete finite alphabet according to discrete probability distribution functions associated with the states.

It is quite natural to think of the speech signal as being generated by such a process. We can imagine the vocal tract as being in one of a finite number of articulatory configurations or states. In each state a short (in time) signal is produced that has one of a finite number of prototypical spectra depending, of course, on the state. Thus, the power spectra of short intervals of the speech signal are determined solely by the current state of the model, while the variation of the spectral composition of the signal with time is governed predominantly by the probabilistic state transition law of the underlying Markov chain. For speech signals derived from a small vocabulary of isolated words, the model is reasonably faithful. The foregoing is, of course, an oversimplification intended only for the purpose of motivating the following theoretical discussion.

Let us say that the underlying Markov chain has  $N$  states  $q_1, q_2, \dots, q_N$  and the observations are drawn from an alphabet,  $\mathbf{V}$ , of  $M$  prototypical spectra,  $v_1, v_2, \dots, v_M$ . The underlying Markov chain can then be specified in terms of an initial state distribution vector  $\pi' = (\pi_1, \pi_2, \dots, \pi_N)$  and a state transition matrix,  $A = [a_{ij}]$   $1 \leq i, j \leq N$ . Here,  $\pi_i$  is the probability of  $q_i$  at some arbitrary time,  $t = 0$ , and  $a_{ij}$  is the probability of transiting to state  $q_j$  given current state,  $q_i$ , that is  $a_{ij} = \text{prob}(q_j \text{ at } t + 1 | q_i \text{ at } t)$ .

The random processes associated with the states can be collectively represented by another stochastic matrix  $B = [b_{jk}]$  in which for  $1 \leq j \leq N$  and  $1 \leq k \leq M$ ,  $b_{jk}$  is the probability of observing symbol  $v_k$  given current state  $q_j$ . We denote this as  $b_{jk} = \text{prob}(v_k \text{ at } t | q_j \text{ at } t)$ . Thus a

hidden Markov model,  $\mathbf{M}$ , is identified with the parameter set  $(\pi, A, B)$ .

To use hidden Markov models to perform speech recognition we must solve two specific problems: observation sequence probability estimation, which will be used for classification of an utterance; and model parameter estimation, which will serve as a procedure for training models for each vocabulary word. Both problems proceed from a sequence,  $\mathbf{O}$ , of observations  $O_1 O_2 \dots O_T$  where each  $O_t$  for  $1 \leq t \leq T$  is some  $v_k \in \mathbf{V}$ .

Our particular classification problem is as follows. We wish to recognize utterances known to be selected from some vocabulary,  $W$ , of words  $w_1, w_2, \dots, w_V$ . We are given an observation sequence,  $\mathbf{O}$ , derived from the utterance of some unknown  $w_i \in W$  and a set of  $V$  models  $\mathbf{M}_1, \mathbf{M}_2, \dots, \mathbf{M}_V$ . We must compute  $P_i = \text{prob}(\mathbf{O} | \mathbf{M}_i)$  for  $1 \leq i \leq V$ . We will then classify the unknown utterance as  $w_i$  iff  $P_i \geq P_j$  for  $1 \leq j \leq V$ .

The training problem is simply that of determining the models  $\mathbf{M}_i = (\pi_i, A_i, B_i)$  for  $1 \leq i \leq V$  given training sequences  $\mathbf{O}^{(1)}, \mathbf{O}^{(2)}, \dots, \mathbf{O}^{(V)}$ , where  $\mathbf{O}^{(i)}$  is known to have been derived from an utterance of word  $w_i$  for  $1 \leq i \leq V$ .

One could, in principle, compute  $\text{prob}(\mathbf{O} | \mathbf{M})$  by computing the joint probability  $\text{prob}(\mathbf{O}, \mathbf{s} | \mathbf{M})$  for each state sequence,  $\mathbf{s}$ , of length  $T$ , and summing over all state sequences. Obviously this is computationally intractable. Fortunately, however, there is an efficient method for computing  $P$ . Let us define the function  $\alpha_t(i)$  for  $1 \leq t \leq T$  as  $\text{prob}(O_1 O_2 \dots O_t \text{ and } q_i \text{ at } t | \mathbf{M})$ . According to the definition  $\alpha_1(i) = \pi_i b_i(O_1)$ , where  $b_i(O_1)$  is understood to mean  $b_{ik}$  iff  $O_1 \equiv v_k$ ; then we have the following recursive relationship for the "forward probabilities":

$$\alpha_{t+1}(j) = \left[ \sum_{i=1}^N \alpha_t(i) a_{ij} \right] b_j(O_{t+1}) \quad 1 \leq t \leq T-1. \quad (1)$$

Similarly, we define another function,  $\beta_t(j) = \text{prob}(O_{t+1} O_{t+2} \dots O_T | q_j \text{ at } t \text{ and } \mathbf{M})$ . We set  $\beta_T(j) = 1 \forall j$  and then use the backward recursion

$$\beta_t(i) = \sum_{j=1}^N a_{ij} b_j(O_{t+1}) \beta_{t+1}(j) \quad T-1 \geq t \geq 1 \quad (2)$$

to compute the "backward probabilities."

The two functions can be used to compute  $P$  according to

$$P = \text{Prob}(\mathbf{O} | \mathbf{M}) = \sum_{i=1}^N \sum_{j=1}^N \alpha_t(i) a_{ij} b_j(O_{t+1}) \beta_{t+1}(j) \quad (3)$$

for any  $t$  such that  $1 \leq t \leq T-1$ . Equations (1) to (3) are from Baum<sup>13</sup> and are sometimes referred to as the "forward-backward" algorithm.

Setting  $t = T - 1$  in (3) gives

$$P = \sum_{i=1}^N \alpha_T(i) \quad (4)$$

so that  $P$  can be computed from the forward probabilities alone. A similar formula for  $P$  can be obtained from the backward probabilities by setting  $t = 1$ . These and several other formulas in this section may be compactly written in matrix notation (see Appendix A). For instance,

$$P = \pi' B_1 A B_2 A \dots A B_{T-1} \mathbf{1}, \quad (5)$$

where  $\mathbf{1}$  is the  $N$ -vector  $(1, 1, 1, \dots, 1)'$  and

$$B_t = \begin{pmatrix} b_1(O_t) & & & \\ & b_2(O_t) & & \mathbf{0} \\ & & \ddots & \\ \mathbf{0} & & & b_N(O_t) \end{pmatrix} \quad (6)$$

for  $1 \leq t \leq T$ . From (5) it is clear that  $P$  is a homogeneous polynomial in the  $\pi_i$ ,  $\alpha_{ij}$ , and  $b_{jk}$ . Any of eqs. (3) through (5) may be used to solve the classification problem. The forward and backward probabilities will prove to be convenient in other contexts.

When we compute  $P$  with the forward-backward algorithm, we are including the probabilities of all possible state sequences that may have generated  $\mathbf{O}$ . Alternatively, we may define  $P$  as the maximum over all state sequences  $i = i_0, i_1, \dots, i_T$  of the joint probability  $P(\mathbf{O}, i)$ . This distinguished state sequence and the corresponding probability of the observation sequence can be simultaneously computed by means of the Viterbi<sup>23</sup> algorithm. This dynamic programming technique proceeds as follows: Let  $\phi_1(i) = \pi_i b_i(O_1)$  for  $1 \leq i \leq N$ . Then we can perform the following recursion for  $2 \leq t \leq T$  and  $1 \leq j \leq N$

$$\phi_t(j) = \max_{1 \leq i \leq N} [\phi_{t-1}(i) \alpha_{ij}] b_j(O_t) \quad (7a)$$

and

$$\psi_t(j) = i^*, \quad (7b)$$

where  $i^*$  is a choice of an index  $i$  that maximizes  $\phi_{t-1}(i)$ .

The result is that  $P = \max_{1 \leq i \leq N} [\phi_T(i)]$ . Also the maximum likelihood state sequence can be recovered from  $\psi$  as follows. Let  $q_T = i^*$ , where  $i^*$  maximizes  $P$ . Then for  $T \geq t \geq 2$ ,  $q_{t-1} = \psi_t(q_t)$ . If one only wishes to compute  $P$ , the linked list,  $\psi$ , need not be maintained as in (7b). Only the recursion (7a) is required.

The problem of training a model, unfortunately, does not have such a simple solution. In fact, given any finite observation sequence as training data, we cannot optimally train the model. We can, however,

choose  $\pi$ ,  $A$ , and  $B$  such that  $\text{prob}(\mathbf{O}|\mathbf{M})$  is locally maximized. For an asymptotic analysis of the training problem the reader should consult Baum and Petrie.<sup>15</sup>

We can use the forward and backward probabilities to formulate a solution to the problem of training by parameter estimation. Given some estimates of the parameter values we can compute, for example, that the expected number of transitions,  $\gamma_{ij}$ , from  $q_i$  to  $q_j$ , conditioned on the observation sequence is just

$$\gamma_{ij} = \frac{1}{P} \sum_{t=1}^{T-1} \alpha_t(i) a_{ij} b_j(O_{t+1}) \beta_{t+1}(j). \quad (8)$$

Then, the expected number of transitions,  $\gamma_i$  out of  $q_i$ , given  $\mathbf{O}$ , is

$$\gamma_i = \sum_{j=1}^N \gamma_{ij} = \frac{1}{P} \sum_{t=1}^{T-1} \alpha_t(i) \beta_t(i), \quad (9)$$

the last step of which is based on (2). The ratio  $\gamma_{ij}/\gamma_i$  is then an estimate of the probability of state  $q_j$ , given that the previous state was  $q_i$ . This ratio may be taken as a new estimate,  $\bar{a}_{ij}$ , of  $a_{ij}$ . That is,

$$\bar{a}_{ij} = \frac{\gamma_{ij}}{\gamma_i} = \frac{\sum_{t=1}^{T-1} \alpha_t(i) a_{ij} b_j(O_{t+1}) \beta_{t+1}(j)}{\sum_{t=1}^{T-1} \alpha_t(i) \beta_t(i)}. \quad (10)$$

Similarly, we can make a new estimate of  $b_{jk}$  as the frequency of occurrence of  $v_k$  in  $q_j$  relative to the frequency of occurrence of any symbol in state  $q_j$ . Stated in terms of the forward and backward probabilities we have

$$\bar{b}_{jk} = \frac{\sum_{t \ni O_t = v_k} \alpha_t(j) \beta_t(j)}{\sum_{t=1}^T \alpha_t(j) \beta_t(j)}. \quad (11)$$

Finally, new values of the initial state probabilities may be obtained from

$$\bar{\pi}_i = \frac{1}{P} \alpha_1(i) \beta_1(i). \quad (12)$$

As we shall see in the next section, the reestimates are guaranteed to increase  $P$ , except at a critical point.

### 2.1 Proof of the reestimation formula

The reestimation formulas (10), (11), and (12) are instances of the Baum-Welch algorithm. Although it is not at all obvious, each application of the formulas is guaranteed to increase  $P$  except if we are at a critical point of  $P$ , in which case the new estimates will be identical

to their current values. Several proofs of this rather surprising fact are given in the literature.<sup>13,17</sup> Because we shall need to modify it later to cope with the finite sample size problem, we shall briefly sketch Baum's proof<sup>13</sup> here. The proof is based on the following two lemmas: *Lemma 1: Let  $u_i, i = 1, \dots, S$  be positive real numbers, and let  $v_i, i = 1, \dots, S$  be nonnegative real numbers such that  $\sum_i v_i > 0$ . Then from the concavity of the log function it follows that*

$$\begin{aligned} \ln \left( \frac{\sum v_i}{\sum u_i} \right) &= \ln \left[ \sum_i \left( \frac{u_i}{\sum_k u_k} \right) \cdot \frac{v_i}{u_i} \right] \\ &\geq \sum_i \frac{u_i}{\sum_k u_k} \ln \left( \frac{v_i}{u_i} \right) \\ &= \frac{1}{\sum_k u_k} \left[ \sum_i (u_i \ln v_i - u_i \ln u_i) \right]. \end{aligned} \quad (13)$$

Here every summation is from 1 to  $S$ .

*Lemma 2: If  $c_i > 0, i = 1, \dots, N$ , then subject to the constraint  $\sum_i x_i = 1$ , the function*

$$F(\mathbf{x}) = \sum_i c_i \ln x_i \quad (14)$$

*attains its unique global maximum when*

$$x_i = \frac{c_i}{\sum_i c_i}. \quad (15)$$

The proof follows from the observation that by the Lagrange method

$$\frac{\partial}{\partial x_i} \left[ F(\mathbf{x}) - \lambda \sum_i x_i \right] = \frac{c_i}{x_i} - \lambda = 0. \quad (16)$$

Multiplying by  $x_i$  and summing over  $i$  gives  $\lambda = \sum_i c_i$ , hence the result.

Now in Lemma 1, let  $S$  be the number of state sequences of length  $T$ . For the  $i$ th sequence let  $u_i$  be the joint probability

$$\begin{aligned} u_i &= \text{Prob}[\text{state sequence } i, \text{ observation } \mathbf{O} \mid \text{model } \mathbf{M}] \\ &= P(i, \mathbf{O} \mid \mathbf{M}). \end{aligned}$$

Let  $v_i$  be the same joint probability conditioned on model  $\bar{\mathbf{M}}$ . Then

$$\begin{aligned} \sum_i u_i &= p(\mathbf{O} \mid \mathbf{M}) \triangleq P(\mathbf{M}) \\ \sum_i v_i &= p(\mathbf{O} \mid \bar{\mathbf{M}}) \triangleq P(\bar{\mathbf{M}}) \end{aligned} \quad (17)$$

and the lemma gives

$$\ln \frac{P(\bar{\mathbf{M}})}{P(\mathbf{M})} \geq \frac{1}{P(\mathbf{M})} \cdot [Q(\mathbf{M}, \bar{\mathbf{M}}) - Q(\mathbf{M}, \mathbf{M})], \quad (18)$$

where

$$Q(\mathbf{M}, \bar{\mathbf{M}}) \triangleq \sum_i u_i \ln v_i. \quad (19)$$

Thus, if we can find a model  $\bar{\mathbf{M}}$  that makes the right-hand side of (18) positive, we have a way of improving the model  $\mathbf{M}$ . Clearly, the largest guaranteed improvement by this method results for  $\bar{\mathbf{M}}$ , which maximizes  $Q(\mathbf{M}, \bar{\mathbf{M}})$  [and hence maximizes the right-hand side of (18)]. The remarkable fact proven in Ref. 13 is that  $Q(\mathbf{M}, \bar{\mathbf{M}})$  attains its maximum when  $\bar{\mathbf{M}}$  is related to  $\mathbf{M}$  by the reestimation formulas (10) through (12). To show this let the  $s$ th-state sequence be  $s_0, s_1, \dots, s_T$ , and the given observation sequence be  $O_{k_1}, \dots, O_{k_T}$ . Then

$$\ln v_s = \ln p(s, \mathbf{O} | \bar{\mathbf{M}}) = \ln \bar{\pi}_{s_0} + \sum_{t=0}^{T-1} \ln \bar{a}_{s_t s_{t+1}} + \sum_{t=0}^{T-1} \ln \bar{b}_{s_{t+1}}(O_{t+1}). \quad (20)$$

Substituting this in (19) for  $Q(\mathbf{M}, \bar{\mathbf{M}})$ , and regrouping terms in the summations according to state transitions and observed symbols, it can be seen that

$$Q(\mathbf{M}, \bar{\mathbf{M}}) = \sum_{i=1}^N \sum_{j=1}^N c_{ij} \ln \bar{a}_{ij} + \sum_{j=1}^N \sum_{k=1}^M d_{jk} \ln \bar{b}_j(k) + \sum_{i=1}^N e_i \ln \bar{\pi}_i. \quad (21)$$

Here

$$c_{ij} = \sum_{s=1}^S p(s, \mathbf{O} | \mathbf{M}) n_{ij}(s) \quad (22a)$$

$$d_{jk} = \sum_{s=1}^S p(s, \mathbf{O} | \mathbf{M}) m_{jk}(s) \quad (22b)$$

$$e_i = \sum_{s=1}^S p(s, \mathbf{O} | \mathbf{M}) r_i(s), \quad (22c)$$

and for the  $s$ th-state sequence

$n_{ij}(s)$  = number of transitions from state  $q_i$  to  $q_j$

$m_{jk}(s)$  = number of times symbol  $k$  is generated in state  $q_j$

$r_i(s) = 1$  if initial state is  $q_i$

$= 0$  otherwise.

Thus,  $c_{ij}$ ,  $d_{jk}$ , and  $e_i$  are the expected values of  $n_{ij}$ ,  $m_{jk}$ ,  $r_i$ , respectively, based on model  $\mathbf{M}$ .

The expression (21) is now a sum of  $2N + 1$  independent expressions of the type maximized in Lemma 2. Hence,  $Q(\mathbf{M}, \bar{\mathbf{M}})$  is maximized if

$$\bar{a}_{ij} = \frac{c_{ij}}{\sum_j c_{ij}} \quad (23a)$$

$$\bar{b}_j(k) = \frac{d_{jk}}{\sum_k d_{jk}} \quad (23b)$$

$$\bar{\pi}_i = \frac{e_i}{\sum_i e_i}. \quad (23c)$$

These are recognized as the reestimation formulas.

## 2.2 Solution by optimization techniques

Lest the reader be led to believe that the reestimation formulas are peculiar to stochastic processes, we shall examine them briefly from several different points of view. Note that the reestimation formulas update the model in such a way that the constraints

$$\sum_{i=1}^N \pi_i = 1 \quad (24a)$$

$$\sum_{j=1}^N a_{ij} = 1 \quad \text{for } 1 \leq i \leq N \quad (24b)$$

and

$$\sum_{k=1}^M b_{jk} = 1 \quad \text{for } 1 \leq j \leq N \quad (24c)$$

are automatically satisfied at each iteration. The constraints are, of course, required to make the hidden Markov model well defined. It is thus natural to look at the training problem as a problem of constrained optimization of  $P$  and, at least formally, solve it by the classical method of Lagrange multipliers. For simplicity, we shall restrict the discussion to optimization with respect to  $A$ . Let  $Q$  be the Lagrangian of  $P$  with respect to the constraints (24b). We see that

$$Q = P + \sum_{i=1}^N \lambda_i \left( \sum_{j=1}^N a_{ij} - 1 \right), \quad (25)$$

where the  $\lambda_i$  are the as yet undetermined Lagrange multipliers.

At a critical point of  $P$  on the interior of the manifold defined by (24a through c), it will be the case that for  $1 \leq i, j \leq N$

$$\frac{\partial Q}{\partial a_{ij}} = \frac{\partial P}{\partial a_{ij}} + \lambda_i = 0. \quad (26)$$

Multiplying (26) by  $a_{ij}$  and summing over  $j$  we get

$$\sum_{j=1}^N a_{ij} \frac{\partial P}{\partial a_{ij}} = - \left[ \sum_{j=1}^N a_{ij} \right] \lambda_i = -\lambda_i = \frac{\partial P}{\partial a_{ij}}, \quad (27)$$

where the right-hand side of (27) follows from substituting (24b) for the sum of  $a_{ij}$  and then replacing  $\lambda_i$  according to (26). From (27) it may be seen that  $P$  is maximized when

$$a_{ij} = \frac{a_{ij} \frac{\partial P}{\partial a_{ij}}}{\sum_{k=1}^N a_{ik} \frac{\partial P}{\partial a_{ik}}}. \quad (28)$$

A similar argument can be made for the  $\pi$  and  $B$  parameters.

While it is true that solving (28) for  $a_{ij}$  is analytically intractable, it can be used to provide some useful insights into the Baum-Welch reestimation formulas and alternatives to them for solving the training problem. Let us begin by computing  $\partial P / \partial a_{ij}$  by differentiating (3), according to the formula for differentiating a product,

$$\frac{\partial P}{\partial a_{ij}} = \sum_{t=1}^{T-1} \alpha_t(i) b_j(O_{t+1}) \beta_{t+1}(j). \quad (29)$$

Substituting the right-hand side of (29) for  $\partial P / \partial a_{ij}$  in (28), we get

$$a_{ij} = \frac{\sum_{t=1}^{T-1} \alpha_t(i) a_{ij} b_j(O_{t+1}) \beta_{t+1}(j)}{\sum_{j=1}^N \sum_{t=1}^{T-1} \alpha_t(i) a_{ij} b_j(O_{t+1}) \beta_{t+1}(j)}. \quad (30)$$

Then changing the order of summation in the denominator of (30) and substituting in the right-hand side of (2) we get

$$a_{ij} = \frac{\sum_{t=1}^{T-1} \alpha_t(i) a_{ij} b_j(O_{t+1}) \beta_{t+1}(j)}{\sum_{t=1}^{T-1} \alpha_t(i) \beta_t(i)}. \quad (31)$$

The right-hand side of (31) is thus seen to be identical to that of the reestimation formula (10). Thus, at a critical point, the reestimation formula (10) solves the equations (31). Similarly, if we differentiate (3) with respect to  $\pi_i$  and  $b_{jk}$  we get

$$\begin{aligned} \frac{\partial P}{\partial \pi_i} &= \sum_{j=1}^N b_i(O_1) a_{ij} b_j(O_2) \beta_2(j) \\ &= b_i(O_1) \beta_1(i) \end{aligned} \quad (32)$$

and

$$\frac{\partial P}{\partial b_{jk}} = \sum_{t \ni O_t = v_k} \sum_{i=1}^N \alpha_t(i) a_{ij} \beta_{t+1}(j) + \delta(O_1, v_k) \pi_j \beta_1(j), \quad (33)$$

respectively. In (33)  $\delta$  is understood to be the Kronecker  $\delta$  function.

By substituting (32) and (33) into their respective analogs of (28), we obtain the reestimation formulas (12) and (11), respectively, at a critical point. Thus it appears that the reestimation formulas may have more general applications than might appear from their statistical motivation.

Equation (28) suggests that we define a transformation,  $\mathbf{T}$ , of the parameter space onto itself as

$$\mathbf{T}(x)_{ij} = \frac{x_{ij} \frac{\partial P}{\partial x_{ij}}}{\sum_{k=1}^N x_{ik} \frac{\partial P}{\partial x_{ik}}}, \quad (34)$$

where  $\mathbf{T}(x)_{ij}$  is understood to mean the  $i, j$ th coordinate of the image of  $\mathbf{x}$  under  $\mathbf{T}$ . The parameter space is restricted to be the manifold such that  $x_{ij} \geq 0$  for  $1 \leq i, j \leq N$  and  $\sum_{j=1}^N x_{ij} = 1$  for  $1 \leq i \leq N$ . Thus, the reestimation formulas (10), (11), and (12) are a special case of the transformation (34), with  $P$  a particular homogeneous polynomial in the  $x_{ij}$  having positive coefficients. Here the  $x_{ij}$  include the  $\pi_i$ , the  $a_{ij}$ , and the  $b_{jk}$ . Baum and Eagon<sup>14</sup> have shown that for any such polynomial  $P[\mathbf{T}(\mathbf{x})] > P(\mathbf{x})$  except if  $\mathbf{x}$  is a critical point of  $P$ . Thus the transformation,  $\mathbf{T}$ , is appropriately called a growth transformation. The conditions under which  $\mathbf{T}$  is a growth transformation were relaxed by Baum and Sell<sup>17</sup> to include all polynomials with positive coefficients. They further proved that  $P$  increases monotonically on the segment from  $\mathbf{x}$  to  $\mathbf{T}(\mathbf{x})$ . Specifically, they showed that  $P[\eta \mathbf{T}(\mathbf{x}) + (1 - \eta)\mathbf{x}] \geq P(\mathbf{x})$  for  $0 \leq \eta \leq 1$ . Other properties of the transformation (34) have been explored by Passman<sup>18</sup> and Stebe.<sup>19</sup> There may be still less restrictive general criteria on  $P$  for  $\mathbf{T}$  to be a growth transformation.

We can give  $\mathbf{T}(\mathbf{x})$  a simple geometric interpretation. For the purposes of this discussion we shall restrict ourselves to  $\mathbf{x} \in \mathbf{R}^N$ ,  $x_i \geq 0$  for  $1 \leq i \leq N$ , and the single constraint  $G(\mathbf{x}) = \sum_{i=1}^N x_i - 1 = 0$ . We do so without loss of generality, since constraints such as those of (24a, b, and c) are disjoint, i.e., no pair of constraints has any common variables. As shown in Fig. 1, given any  $\mathbf{x}$  satisfying  $G(\mathbf{x}) = 0$ ,  $\mathbf{T}(\mathbf{x})$  is the intersection of the vector  $\mathbf{X}$ , or its extension, with the hyperplane  $\sum_{i=1}^N x_i - 1 = 0$ , where  $\mathbf{X}$  has components  $x_i \frac{\partial P}{\partial x_i}$  for  $1 \leq i \leq N$ .

This may be shown by observing that a line in the direction of  $\mathbf{X}$  passing through the origin has the equation  $\mathbf{y} = r\mathbf{X}$ , where  $r$  is a non-

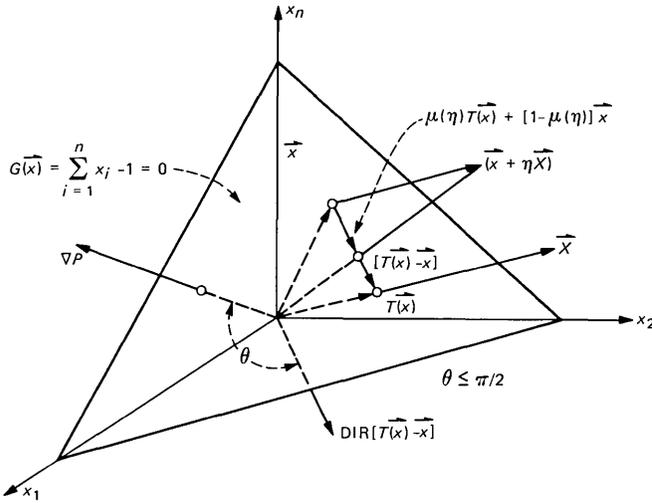


Fig. 1—Geometrical relationship of the quantities involved in the reestimation formulas.

negative scalar. Component-wise this is equivalent to

$$y_i = r x_i \frac{\partial P}{\partial x_i} \quad \text{for } 1 \leq i \leq N. \quad (35)$$

We can find that  $\mathbf{y}$  intersects the hyperplane  $G(\mathbf{x}) = 0$  by summing over  $i$ . Thus

$$\sum_{i=1}^N y_i = r \sum_{i=1}^N x_i \frac{\partial P}{\partial x_i} = 1, \quad (36)$$

since  $\mathbf{y}$  lies on the hyperplane  $G(\mathbf{x}) = 0$ . Rearranging (36) we have

$$r = \frac{1}{\sum_{i=1}^N x_i \frac{\partial P}{\partial x_i}} \quad (37)$$

and

$$y_i = \frac{x_i \frac{\partial P}{\partial x_i}}{\sum_{j=1}^N x_j \frac{\partial P}{\partial x_j}}. \quad (38)$$

Furthermore, as also shown in Fig. 1, the vector  $[\mathbf{T}(\mathbf{x}) - \mathbf{x}]$  is the set of intersections of the vector  $(\mathbf{x} + \eta\mathbf{X})$  with the hyperplane  $G(\mathbf{x}) = 0$  for  $0 \leq \eta \leq +\infty$  with  $\mathbf{T}(\mathbf{x})$  corresponding to  $\eta = +\infty$  and  $\mathbf{x}$  to  $\eta = 0$ .

Finally, in view of the result of Baum and Sell, quoted above, the vector  $\mathbf{T}(\mathbf{x}) - \mathbf{x}$  must also have a positive projection on  $\nabla P$ . This,

too, is easily seen. If  $P$  is a polynomial with positive coefficients, then  $\partial P/\partial x_i \geq 0$  for  $1 \leq i \leq N$ . From the definition of  $\mathbf{T}$  it is clear that

$$\mathbf{T}(\mathbf{x})_i \geq x_i \quad \text{iff} \quad \frac{\partial P}{\partial x_i} \geq \sum_{j=1}^N x_j \frac{\partial P}{\partial x_j} = r, \quad (39)$$

where  $r$  is some constant. Then it must be true that

$$\sum_{i=1}^N [\mathbf{T}(\mathbf{x})_i - x_i] \left( \frac{\partial P}{\partial x_i} - r \right) \geq 0 \quad (40)$$

since both factors in each summand are of the same sign. Rearranging (40) we have

$$\sum_{i=1}^N [\mathbf{T}(\mathbf{x})_i - x_i] \frac{\partial P}{\partial x_i} \geq r \sum_{i=1}^N [\mathbf{T}(\mathbf{x})_i - x_i] = 0. \quad (41)$$

The right-hand side is zero since  $\sum_{i=1}^N \mathbf{T}(\mathbf{x})_i = \sum_{i=1}^N x_i = 1$ . Thus  $[\mathbf{T}(\mathbf{x}) - \mathbf{x}] \cdot \nabla P \geq 0$ , proving that a step of the transformation has a positive projection along the gradient of  $P$ .

This merely guarantees that we can move an infinitesimal amount in the direction of  $[\mathbf{T}(\mathbf{x}) - \mathbf{x}]$  while increasing  $P$ . The theorem of Baum and Eagon, however, guarantees much more, namely that we can take a finite step and be assured of increasing  $P$ . We may, in fact, be able to continue past  $\mathbf{T}(\mathbf{x})$  while still increasing  $P$ . We are unable, at present, to give a geometrical interpretation of this fact.

While the reestimation formulas provide an elegant method for maximizing  $P$ , their success depends critically on the constraint set (24a, b, and c). As we will suggest later, in some cases there may be advantages in using classical optimization methods.

The principle of the classical methods is to search along the projection of  $\nabla P$  on the constraint space,  $G$ , for a local maximum. The method of Rosen,<sup>24</sup> for example, uses only  $\nabla P$  and a crude search strategy. The method of Davidon is one of many quasi-Newton techniques that uses the Fletcher-Powell<sup>25</sup> approximation to the inverse of the Hessian of  $P$  and an exact line search with adaptive step size. There are many collections of general purpose subroutines for constrained optimization that can be used to solve the training problem. We have successfully used a version of the Davidon procedure from the Harwell Subroutine Library.<sup>26</sup> However, for the constraints that  $\pi$ ,  $A$ , and  $B$  be stochastic, the computation can be greatly simplified.

We illustrate this by outlining the gradient search algorithm for the case where  $P$  is a function of the variables  $x_1, \dots, x_N$  subject to the constraints  $x_i \geq 0$  for  $1 \leq i \leq N$  and  $\sum_{i=1}^N x_i - 1 = 0$ . For convenience we will call the last constraint  $G_1$ , and the inequality constraints on  $x_1, \dots, x_N$  as  $G_2, \dots, G_{N+1}$ , respectively.

An initial starting point  $\mathbf{x}$  is chosen and the "active" constraints

identified. For our case  $G_1$  is always active. For  $i > 1$ ,  $G_i$  is active if  $x_{i-1} = 0$ . Let  $G_{n_j}$ ,  $j = 1, \dots, \ell$  be the active constraints (with  $n_1 = 1$ ) at the initial point. Let  $Q = P + \sum_{j=1}^{\ell} \lambda_j G_{n_j}$ . Then according to the Kuhn-Tucker theorem,<sup>27</sup> the Lagrangian multipliers,  $\lambda_j$ , are determined by demanding that  $\nabla Q$  be orthogonal to  $\nabla G_{n_j}$  for  $1 \leq j \leq \ell$ . Now

$$\begin{aligned}\nabla Q &= \nabla P + \sum_{j=1}^{\ell} \lambda_j \nabla G_{n_j} \\ &= \nabla P + \Gamma \lambda,\end{aligned}\tag{42}$$

where  $\Gamma$  is the  $N \times \ell$  matrix with  $\Gamma_{ij} = (\nabla G_{n_j})_i = \partial G_{n_j} / \partial x_i$ , and  $\lambda$  is the vector with components  $\lambda_j$  for  $j = 1, \dots, \ell$ . Thus the Kuhn-Tucker requirement is equivalent to

$$\Gamma' \nabla Q = 0\tag{43}$$

or, from (42),

$$\lambda = -(\Gamma' \Gamma)^{-1} \Gamma' \nabla P.\tag{44}$$

For our special constraints we have

$$\Gamma_{i1} = 1 \quad \text{for } 1 \leq i \leq N\tag{45}$$

and, for  $j \neq 1$

$$\Gamma_{ij} = \begin{cases} 1 & \text{if } i = n_j - 1 \\ 0 & \text{otherwise.} \end{cases}\tag{46}$$

With  $\Gamma$  defined this way

$$\Gamma' \Gamma = \begin{pmatrix} N & 1 & 1 & \dots & 1 \\ 1 & 1 & & & \\ 1 & & 1 & \mathbf{0} & \\ \vdots & \mathbf{0} & & \ddots & \\ 1 & & & & 1 \end{pmatrix}\tag{47}$$

and  $(\Gamma' \Gamma)^{-1}$  may be shown to be

$$(\Gamma' \Gamma)^{-1} = \frac{1}{N - \ell + 1} \begin{pmatrix} 1 & -1 & -1 & \dots & -1 \\ -1 & N - \ell & 1 & \dots & 1 \\ -1 & 1 & N - \ell & 1 & 1 \\ \vdots & & & \ddots & 1 \\ -1 & 1 & 1 & & N - \ell \end{pmatrix}.\tag{48}$$

Substituting (48) into (44) gives  $\lambda$ . When this  $\lambda$  is substituted back into (42), it turns out that the resulting vector  $\nabla Q$  can be computed by the following simple steps:

(i) Compute  $\nabla P$  and let  $S$  be the sum of all components of  $\nabla P$  except  $(\nabla P)_{n_{j-1}}$ ,  $j = 2, \dots, \ell$ .

(ii) Then

$$\begin{aligned}(\nabla Q)_i &= 0 & i = n_j, & \quad j = 2, \dots, \ell \\ &= (\nabla P)_i - \frac{S}{N - \ell + 1} & \text{otherwise.}\end{aligned}$$

Finally, the values of  $P$  are searched along the line

$$\mathbf{x}(\eta) = \mathbf{x} + \eta \frac{\nabla Q}{\|\nabla Q\|} \quad (49)$$

for a maximum with respect to  $\eta$ . The procedure is repeated at this new point.

In applying this technique to the actual training problem, there will be  $2N + 1$  stochasticity constraints analogous to  $G_1$  and a corresponding number of positivity constraints analogous to  $G_2, G_3, \dots, G_{N+1}$ . In this case we have the option of treating all the parameters and their associated constraints together, or we may divide them into disjoint subsets and determine search directions for each subset independently.

Notice that this derivation does not require  $P$  to be of any special form. This may prove to be an advantage since the Baum-Welch algorithm is not applicable to all  $P$ . Furthermore, the constraints may be changed. Although, as we shall see later, the Baum-Welch algorithm can be somewhat generalized in this respect, it does not generalize to work with arbitrary linear constraints.

### III. CONSIDERATIONS FOR IMPLEMENTATION

From the foregoing discussion it might appear that solutions to the problems of hidden Markov modeling can be obtained by straightforward translation of the relevant formulas into computer programs. Unfortunately, for all but the most trivial problems, the naive implementation will not succeed for two principal reasons. First, any of the methods of solution presented here for either the classification or the training problem require evaluation of  $\alpha_t(i)$  and  $\beta_t(i)$  for  $1 \leq t \leq T$  and  $1 \leq i \leq N$ . From the recursive formulas for these quantities, (1) and (2), it is clear that as  $T \rightarrow \infty$ ,  $\alpha_T(i) \rightarrow 0$ , and  $\beta_1(i) \rightarrow 0$  in exponential fashion. In practice, the number of observations necessary to adequately train a model and/or compute its probability will result in underflow on any real computer if (1) and (2) are evaluated directly. Fortunately, there is a method for scaling these computations that not only solves the underflow problem but also greatly simplifies several other calculations.

The second problem is more serious, more subtle, and admits of a less gratifying, though still effective, solution. Baum and Petrie<sup>15</sup> have shown that the maximum likelihood estimates of the parameters of a hidden Markov process are consistent estimates (converge to the true

values as  $T \rightarrow \infty$ ) of the parameters. The practical implication of the theorem is that, in training, one should use as many observations as possible which, as we have noted, make scaling necessary. In reality, of course, the observation sequence will always be finite. Then the following situation can arise. Suppose a given training sequence of length  $T$  results in  $b_{jk} = 0$ . (It is, in fact, possible for a local maximum of  $P$  to lie on a boundary of the parameter manifold.) Suppose further that we are subsequently asked to compute the probability that a new observation sequence was generated by our model. Even if the new sequence was actually generated by the model, it can be such that  $\alpha_{t-1}(i)a_{ij}$  is nonzero for only one value of  $j$  and that  $O_t = v_k$ , whence  $\alpha_t(j) = 0$  and the probability of the observation then becomes zero. This phenomenon is fatal to a classification task; yet, the smaller  $T$  is, the more likely is its occurrence. Jelinek and Mercer<sup>9</sup> have dealt with this problem in a slightly different context. Here, we offer the much simpler solution of constraining the parameter values so that  $x_{ij} \geq \epsilon_{ij} > 0$ .

Finally, in this section we discuss the related problem of model stability. Baum and Eagon<sup>14</sup> note that successive applications of the reestimation formulas converge to a connected component of the local maximum set of  $P$ . In case there are only a finite number of such extrema, the point of convergence is unique to within a renaming of the states. The component of the local maximum set to which the iteration converges as well as which of the  $N!$  labelings of the states is determined by the initial estimates of the parameters. If we wish to average several models resulting from several different starting points to achieve model stability, we must be able to match the states of models whose states are permuted. We have devised a solution to this problem based on a minimum-weight bipartite matching algorithm.<sup>28</sup>

### 3.1 Scaling

The principle on which we base our scaling is to multiply  $\alpha_t(i)$  by some scaling coefficient independent of  $i$  so that it remains within the dynamic range of the computer for  $1 \leq t \leq T$ . We propose to perform a similar operation on  $\beta_t(i)$  and then, at the end of the computation, remove the total effect of the scaling.

We illustrate the procedure for (10), the reestimation formula for the state transition probabilities. Let  $\alpha_t(i)$  be computed according to (1) and then be multiplied by a scaling coefficient,  $c_t$ , where say,

$$c_t = \left[ \sum_{i=1}^N \alpha_t(i) \right]^{-1} \quad (50)$$

so that  $\sum_{i=1}^N c_t \alpha_t(i) = 1$  for  $1 \leq t \leq T$ . Then, as we compute  $\beta_t(i)$  from (2), we form the product  $c_t \beta_t(i)$  for  $T \geq t \geq 1$  and  $1 \leq i \leq N$ . In terms

of the scaled forward and backward probabilities, the right-hand side of (10) becomes

$$\frac{\sum_{i=1}^{T-1} C_t \alpha_t(i) a_{ij} b_j(O_{t+1}) \beta_{t+1}(j) D_{t+1}}{\sum_{i=1}^{T-1} \sum_{\ell=1}^N C_t \alpha_t(i) a_{i\ell} b_\ell(O_{t+1}) \beta_{t+1}(\ell) D_{t+1}}, \quad (51)$$

where

$$C_t = \prod_{\tau=1}^t c_\tau \quad (52)$$

and

$$D_t = \prod_{\tau=t}^T c_\tau.$$

This results from the individual scale factors being multiplied together as we perform the recursions of (1) and (2).

Now note that each summand in both the numerator and the denominator has the coefficient  $C_t D_{t+1} = \prod_{\tau=1}^T c_\tau$ . These coefficients can be factored out and canceled so that (51) has the correct value  $\bar{a}_{ij}$  as specified by (10). The reader can verify that this technique may be equally well applied to the reestimation formulas (11) and (12). It should also be obvious that, in practice, the scaling operation need not be performed at every observation time. One can use any scaling interval for which underflow does not occur. In this case, the scale factors corresponding to values of  $t$  within any interval are set to unity.

While the above described scaling technique leaves the reestimation formulas invariant, (3) and (4) are still useless for computing  $P$ . However,  $\log P$  can be recovered from the scale factors as follows. Assume that we compute  $c_t$  according to (50) for  $t = 1, 2, \dots, T$ . Then

$$C_T \sum_{i=1}^N \alpha_T(i) = 1 \quad (53)$$

and from (53) it is obvious that  $C_T = 1/P$ . Thus, from (52) we have

$$\prod_{i=1}^T c_i = \frac{1}{P}. \quad (54)$$

The product of the individual scale factors cannot be evaluated but we can compute

$$\log P = - \sum_{i=1}^T \log c_i. \quad (55)$$

If one chooses to use the Viterbi algorithm for classification, then  $\log P$  can be computed directly from  $\pi$ ,  $A$ , and  $B$  without regard for

the scale factors. Initially, we let  $\phi_1(i) = \log[\pi_i b_i(O_1)]$  and then modify (7a) so that

$$\phi_t(j) = \max_{1 \leq i \leq N} [\phi_{t-1}(i) + \log a_{ij}] + \log[b_j(O_t)]. \quad (56)$$

In this case  $\log P = \max_{1 \leq i \leq N} [\phi_T(i)]$ .

If the parameters of the model are to be computed by means of classical optimization techniques, we can make the computation better conditioned numerically by maximizing  $\log P$  rather than  $P$ . The scaling method of (50) makes this straightforward.

First note that if we are to maximize  $\log P$ , then we will need the partial derivatives of  $\log P$  with respect to the parameters of the model. So, for example, we will need

$$\frac{\partial}{\partial a_{ij}} (\log P) = \frac{1}{P} \frac{\partial P}{\partial a_{ij}} = C_T \frac{\partial P}{\partial a_{ij}}. \quad (57)$$

Substituting the right-hand side of (29) for  $\partial P / \partial a_{ij}$  in the right-hand side of (57) yields

$$\begin{aligned} \frac{\partial}{\partial a_{ij}} (\log P) &= C_T \sum_{t=1}^{T-1} \alpha_t(i) b_j(O_{t+1}) \beta_{t+1}(j) \\ &= \sum_{t=1}^{T-1} C_t \alpha_t(i) b_j(O_{t+1}) \beta_{t+1}(j) D_{t+1} \\ &= \sum_{t=1}^{T-1} \left( \prod_{\tau=1}^t c_\tau \right) \alpha_t(i) b_j(O_{t+1}) \beta_{t+1}(j) \left( \prod_{\tau=t+1}^T c_\tau \right). \quad (58) \end{aligned}$$

So that if we evaluate (29) formally, using not the true values of the forward and backward probabilities but the scaled values, then we will have the correct value of the partial derivatives of  $\log P$  with respect to the transition probabilities. A similar argument can be made for the other parameters of the model and, thus, the scaling method of (50) provides a means for the direct evaluation of  $\nabla(\log P)$ , which is required for the classical optimization algorithms. Later we shall see that the combination of maximizing  $\log P$  and this scaling technique simplifies the solution of the left-to-right Markov modeling problem as well.

### 3.2 Finite training sets

We now turn our attention to solving the problems created by finite-training-set size. As we noted earlier, the effect of this problem is that observation sequences generated by a putative model will have zero probability conditioned on the model parameters. Since the cause of the difficulty is the assignment of zero to some parameters, usually

one or more symbol probabilities, it is reasonable to try to solve the problem by constraining the parameters to be positive.

We can maximize  $P$  subject to the new constraints  $a_{ij} \geq \epsilon > 0$ ;  $b_{jk} \geq \epsilon > 0$ , most easily using the classical methods. In fact, the algorithm described earlier based on the Kuhn-Tucker theorem is unchanged except that the procedure for determining the active constraints is based on  $\epsilon$  rather than zero.

While the Lagrangian methods are perfectly adequate, it is also possible to build the new constraints into the Baum-Welch algorithm. We can show how this is done by making a slight modification to the proof of the algorithm given earlier (Section 2.1). Recall that the proof of the Baum-Welch algorithm was based on maximization of  $2N + 1$  expressions of the type maximized in Lemma 2, eq. (14). Since these expressions involve disjoint sets of variables chosen from  $A, B, \pi$ , it suffices to consider any one of the maximizations. In fact, it suffices to show how Lemma 2 gets modified. Thus we wish now to maximize

$$F(\mathbf{x}) = \sum_i c_i \ln x_i \quad (59)$$

subject to the constraints

$$\sum_i x_i = 1 \quad (60a)$$

and

$$x_i \geq \epsilon, \quad i = 1, \dots, N. \quad (60b)$$

(From the following discussion it will be obvious that a trivial generalization allows  $\epsilon$  to depend on  $i$ .)

Now without the inequality constraints (60b), Lemma 2 showed that  $F(\mathbf{x})$  attains its unique global maximum when  $x_i = c_i / \sum_i c_i$ . Suppose now that this global maximum occurs outside the region specified by the inequality constraints (60b). Specifically, let

$$\bar{x}_i = \frac{c_i}{\sum_{j=1}^N c_j} \geq \epsilon \quad \text{for } i = 1, \dots, N - \ell \quad (61a)$$

$$< \epsilon \quad \text{for } i = N - \ell + 1, \dots, N. \quad (61b)$$

From the concavity of  $F(\mathbf{x})$  it follows that the maximum, subject to the inequality constraints, must occur somewhere on the boundary specified by the violated constraints (61b). Now it is easily shown that if  $\bar{x}_i$  for some  $i > N - \ell$  is replaced by  $\epsilon$ , then the global maximum over the rest of the variables occurs at values *lower* than those given above. From this we conclude that we must set

$$\bar{x}_i = \epsilon \quad \text{for } i > N - \ell \quad (62)$$

and maximize

$$\tilde{F}(\mathbf{x}) = \sum_{i=1}^{N-\ell} c_i \ln x_i \quad (63)$$

subject to the constraint  $\sum_{i=1}^{N-\ell} x_i = 1 - \ell\epsilon$ . But this, analogously to Lemma 2, occurs when

$$\bar{x}_i = (1 - \ell\epsilon) \frac{c_i}{\sum_{j=1}^{N-\ell} c_j} \quad i \leq N - \ell. \quad (64)$$

If these new values of  $\bar{x}_i$  satisfy the constraints, we are done. If one or more become lower than  $\epsilon$ , they too must be set equal to  $\epsilon$ , and  $\ell$  augmented appropriately.

Thus the modified Baum-Welch algorithm is as follows. Suppose we wish to constrain  $b_{jk} \geq \epsilon$  for  $1 \leq j \leq N$  and  $1 \leq k \leq M$ . We first evaluate  $B$  using the reestimation formulas. Assume that some set of the parameters in the  $j$ th row of  $B$  violates the constraint so that  $b_{jk_i} < \epsilon$  for  $1 \leq i \leq \ell$ . Then set  $\tilde{b}_{jk_i} = \epsilon$  for  $1 \leq i \leq \ell$  and readjust the remaining parameters according to (64) so that

$$\tilde{b}_{jk} = (1 - \ell\epsilon) \frac{b_{jk}}{\sum_{i=1}^{N-\ell} b_{ji}} \quad \forall k \notin \{k_i | 1 \leq i \leq \ell\}. \quad (65)$$

After performing the operation of (65) for each row of  $B$ , the resulting  $\tilde{B}$  is the optimal update with respect to the desired constraints. The method can be extended to include the state transition matrix if so desired. There is no advantage to treating  $\pi$  in the same manner since, for any single observation sequence,  $\bar{\pi}$  will always be a unit vector with exactly one nonzero component. In any case, (65) may be applied at each iteration of the reestimation formulas, or once as a post-processing stage after the Baum-Welch algorithm has converged.

### 3.3 Combining models

The final implementational issue that we shall consider in this section is that of combining models for improved stability. There are several circumstances under which it may be desirable to combine several models into one. In spectral estimation, for example, to compute a long-term average spectrum of a stationary signal, it may be convenient to average a number of spectra computed over shorter intervals. It seems quite natural to apply similar block-processing techniques to the Markov modeling problem if the source is assumed to be ergodic. We may, for example, divide a long sequence of observations into contiguous subsequences, estimate model parameters for each subsequence, and combine the results.

Whether or not the source is ergodic, we may still attempt to increase the robustness of our model by averaging the parameter estimates derived from multiple initial values and/or independent observation sequences.

In any case, the difficulty that will be encountered is that even if there are finitely many isolated local maxima of  $P$ , they are only unique to within a renaming of the states. For two different observation sequences,  $q_i$  and  $q_j$  may be topologically equivalent, but  $i \neq j$ . We might try to avoid this problem by using the final parameter values for one observation sequence as the initial values for the next in hopes that this will restrict the search to a neighborhood of a single local maximum. This method, unfortunately, is not reliable. A better approach is that of finding a renaming of the states that minimizes, in some sense, the difference between two models.

Suppose  $\mathbf{b}_j$  and  $\tilde{\mathbf{b}}_j$ ,  $1 \leq j \leq N$  are, respectively, the rows of two estimates of  $B$ . Let  $p(j)$  be a permutation of the state index,  $j$ , and let  $d(\cdot, \cdot)$  be some distance metric. Then we seek the permutation,  $p$ , of  $(q_1, q_2, \dots, q_N)$  such that

$$D = \sum_{j=1}^N d[\mathbf{b}_j, \tilde{\mathbf{b}}_{p(j)}] \quad (66)$$

is minimized. The naive solution is to try all possible  $N!$  permutations and select the best one in the sense of (66). However, for  $N > 10$  the computation becomes intractable. The problem can be brought within reach, however, by transforming it into a minimum-weight bipartite-graph-matching problem on  $2N$  vertices. In the literature on combinatorial optimization (see, e.g., Ref. 28), several algorithms are available for accomplishing such a match in a number of operations that grow as  $N^3$ . In Appendix B, we describe one such algorithm based on an outline provided to us by R. E. Tarjan.

#### IV. LEFT-TO-RIGHT HIDDEN MARKOV MODELS

For the purposes of isolated word recognition, it is useful to consider a special class of absorbing Markov chains that leads to what we call left-to-right models. These models have the following properties:

(i) The first observation is produced while the Markov chain is in a distinguished state called the starting state, designated  $q_1$ .

(ii) The last observation is generated while the Markov chain is in a distinguished state called the final or absorbing state, designated  $q_N$ .

(iii) Once the Markov chain leaves a state, that state cannot be revisited at a later time.

The simplest form of a left-to-right model is shown in Fig. 2, from which the origin of the term left-to-right becomes clear.

In this section we shall consider two problems associated with these

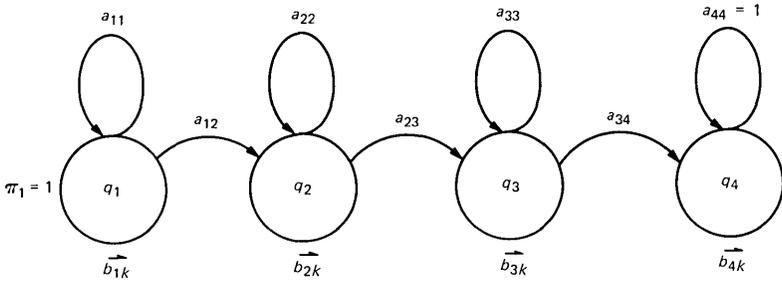


Fig. 2—The simplest form of left-to-right model.

special hidden Markov models. Note that a single, long-observation sequence is useless for training such models, because once the state  $q_N$  is reached, the rest of the sequence provides no further information about earlier states. The appropriate training data for such a model are a set of observation sequences obtained by several starts in state  $q_1$ . In the case of isolated word recognition, for instance, several independent utterances of the same word provide such a set. We wish, therefore to modify the training algorithm to handle such training data. We also wish to compute the probability that a single given observation sequence,  $O_1, O_2 \dots, O_T$ , was produced by the model, with the assumption that  $O_1$  was produced in state  $q_1$  and  $O_T$  in state  $q_N$ . The three conditions mentioned above can be satisfied as follows:

Condition (i) will be satisfied if we set  $\pi = (1, 0, \dots, 0)$  and do not reestimate it. Condition (ii) can be imposed by setting

$$\beta_T(j) = \begin{cases} 1 & \text{for } j = N \\ 0 & \text{otherwise.} \end{cases} \quad (67)$$

Condition (iii) can be guaranteed in the Baum-Welch algorithm by initially setting  $a_{ij} = 0$  for  $j < i$  (and in fact for any other combination of indices that specify transitions to be disallowed). It is clear from (34) that any parameter once set to zero will remain zero. For the gradient methods the appropriate  $a_{ij}$ 's are just set to zero and only the remaining parameters are adjusted.

The modification of the training procedure is as follows: Let us denote by  $\mathbf{O} \equiv [\mathbf{O}^{(1)}, \mathbf{O}^{(2)}, \dots, \mathbf{O}^{(K)}]$  the set of observation sequences, where  $\mathbf{O}^{(k)} = O_1^{(k)} O_2^{(k)} \dots O_{T_k}^{(k)}$  is the  $k$ th sequence. We treat the observation sequences as independent of each other and then we adjust the parameters of the model  $\mathbf{M}$  to maximize

$$\begin{aligned} P &= \prod_{k=1}^K \text{Prob}(\mathbf{O}^{(k)} | \mathbf{M}). \\ &= \prod_{k=1}^K P_k. \end{aligned} \quad (68)$$

Since the Baum-Welch algorithm computes the frequency of occurrence of various events, all we need to do is to compute these frequencies of occurrence in each sequence separately and add them together. Thus the new reestimation formulas may be written as

$$\bar{a}_{ij} = \frac{\sum_{k=1}^K \sum_{t=1}^{T_k-1} \alpha_t^k(i) a_{ij} b_j(O_{t+1}^{(k)}) \beta_{t+1}^k(j)}{\sum_{k=1}^K \sum_{t=1}^{T_k-1} \alpha_t^k(i) \beta_t^k(i)} \quad (69)$$

and

$$\bar{b}_{ij} = \frac{\sum_{k=1, \exists O_{t(k)}=v_j}^K \sum_{t=1}^{T_k} \alpha_t^k(i) \beta_t^k(i)}{\sum_{k=1}^K \sum_{t=1}^{T_k} \alpha_t^k(i) \beta_t^k(i)} \quad (70)$$

As noted above,  $\pi$  is not reestimated.

Scaling these computations requires some care since the scale factors for each individual set of forward and backward probabilities will be different. One way of circumventing the problem is to remove the scale factors from each summand before adding. We can accomplish this by returning the  $1/P$  factor [which appears in (8) and (9) and was cancelled to obtain (10)] to the reestimation formula. Using the reestimation formula for the transition probabilities as an example, (69) becomes

$$\bar{a}_{ij} = \frac{\sum_{k=1}^K \frac{1}{P_k} \sum_{t=1}^{T_k-1} \alpha_t^k(i) a_{ij} b_j(O_{t+1}^{(k)}) \beta_{t+1}^k(j)}{\sum_{k=1}^K \frac{1}{P_k} \sum_{t=1}^{T_k-1} \alpha_t^k(i) \beta_t^k(i)} \quad (71)$$

If the right-hand side of (71) is evaluated using the scaled values of the forward and backward probabilities, then each term in the inner summation will be scaled by  $C_t^k D_{t+1}^k$ , which will then be cancelled by the same factor which multiplies  $P_k$ . Thus, using the scaled values in computing (69) results in an unscaled  $\bar{a}_{ij}$ . The procedure is easily extended to computation of the symbol probabilities. Also note that for the purposes of classification only one subsequence is to be considered so that either (55) or (56) may be used unaltered to compute  $P$ .

To apply Lagrangian techniques to left-to-right models we note that upon taking logarithms of (68) we have

$$\log P = \sum_{k=1}^K \log P_k \quad (72)$$

The derivatives needed to maximize  $\log P$  in (72) can be obtained by evaluating expressions for the derivatives of each individual subse-

quence and summing. For example, for  $a_{ij}$  we have [cf. (57) and (58)]

$$\frac{\partial}{\partial a_{ij}} (\log P) = \sum_{k=1}^K \frac{\partial}{\partial a_{ij}} (\log P) = C_T^k \sum_{t=1}^{T_k-1} \alpha_t^k(i) b_f(O_{t+1}^k) \beta_{t+1}^k(j). \quad (73)$$

As in all previous cases an analogous formula may be derived for the other parameters.

In practice,  $A$  and  $B$  for left-to-right models are especially sparse. Some of the zero values are so by design but others are dependent on  $O$ . Parameters of this type will be found one at a time by standard line search strategies. We have found that the convergence of the Lagrangian techniques can be substantially accelerated by taking large enough steps so that several positivity constraints become binding. The corresponding variables are then clamped and (65) is applied before beginning the next iteration.

## V. NUMERICAL EXAMPLES

In this section, we give some instructive examples of the behavior of several of the algorithms discussed above. The algorithms were all coded in FORTRAN 77 on a Data General MV-8000, which uses a 32-bit floating point word. The data used in the tests came from either a Monte Carlo simulation of a hidden Markov chain or from a portion of a newspaper text that was edited to include only the 26 characters of the English alphabet and a special character denoting an interword space. The simulations have the valuable property that the model is known a priori, so that simple models may be used for checking program correctness while the more complicated ones can elicit some subtle and important numerical and methodological characteristics of the algorithms.

In our experiments we used the following procedure to generate observation sequences by means of a random number generator whose output is uniform on  $[0, 1]$  and specified values of  $\pi$ ,  $A$ ,  $B$ ,  $T$ , and  $V$ :

(i) Partition the unit interval proportionally to the components of  $\pi$ . Generate a random number and select a start state,  $q_i$ , according to the subinterval in which the number falls. Set  $t = 1$ .

(ii) Partition the unit interval proportionally to the components of the  $i$ th row of  $B$ . Generate a random number and select a symbol,  $v_k$ , according to the subinterval in which the number falls. Set  $O_t = v_k$ .

(iii) Partition the unit interval proportionally to the components of the  $i$ th row of  $A$ . Generate a random number and select the next state,  $q_j$ , according to the subinterval in which the number falls.

(iv) Increment  $t$ . If  $t \leq T$  set  $q_i = q_j$  and repeat (ii) through (iv); otherwise stop.

Using this observation generator, several two- and three-state Markov models were simulated. These simulations were used to verify that

the parameter estimation algorithms were working correctly and to study the effects of the scaling interval on the accuracy of the algorithms. In this study we found that all scaling intervals that were sufficiently short to prevent underflow yielded numerically identical results. Thus one can, at one extreme, scale the forward and backward probabilities after each observation or, at the other, wait until a threshold signaling that underflow is imminent is exceeded and only then perform the scaling operation.

We next proceeded to study a pair of four-state ( $N = 4$ ), four-symbol ( $M = 4$ ) models shown below and referred to as SRC44 and SRC45.

SRC44:

$$A = \begin{bmatrix} 0 & 0 & 0.5 & 0.5 \\ 0.5 & 0 & 0 & 0.5 \\ 0.5 & 0.5 & 0 & 0 \\ 0 & 0.5 & 0.5 & 0 \end{bmatrix} \quad B = \begin{bmatrix} 0.5 & 0.5 & 0 & 0 \\ 0 & 0.5 & 0.5 & 0 \\ 0 & 0 & 0.5 & 0.5 \\ 0.5 & 0 & 0 & 0.5 \end{bmatrix}$$

$$\pi = [0.25 \quad 0.25 \quad 0.25 \quad 0.25]$$

and

SRC45:

$$A = \begin{bmatrix} 0 & 0 & 0.25 & 0.75 \\ 0.15 & 0 & 0 & 0.85 \\ 0.2 & 0.8 & 0 & 0 \\ 0 & 0.22 & 0.78 & 0 \end{bmatrix} \quad B = \begin{bmatrix} 0.25 & 0.75 & 0 & 0 \\ 0 & 0.15 & 0.85 & 0 \\ 0 & 0 & 0.1 & 0.9 \\ 0.2 & 0 & 0 & 0.8 \end{bmatrix}$$

$$\pi = [0.25 \quad 0.25 \quad 0.25 \quad 0.25].$$

The state transition diagrams for these models are shown in Fig. 3.

Model SRC44 is a balanced model in the sense that all permissible transitions and symbols are equally likely, whereas model SRC45 is skewed in that it distinctly favors some transitions and symbols over others.

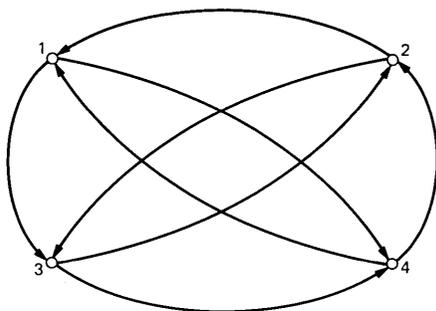


Fig. 3—The four-state model used for testing.

For each of these sources we processed observation sequences ranging in length from 100 to 4000 with the Baum-Welch algorithm. Initial values for  $A$ ,  $B$ , and  $\pi$  were chosen at random and the algorithm was terminated in one of two ways; either when the change in  $\log P$  from one iteration to the next fell below an arbitrary threshold, or when the number of iterations exceeded a specified maximum value. The maximum number of iterations was varied from 100 to 1000. For each estimate,  $\hat{B}$ , of the source matrix  $B$ , a measure of estimation error

$$\|\hat{B} - B\| = \left\{ \frac{1}{M \cdot N} \sum_{j=1}^N \sum_{k=1}^M \left[ \hat{\delta}_{jk} - b_{p(j)k} \right]^2 \right\}^{1/2} \quad (74)$$

was computed, where  $p(j)$  is the state permutation that minimizes the

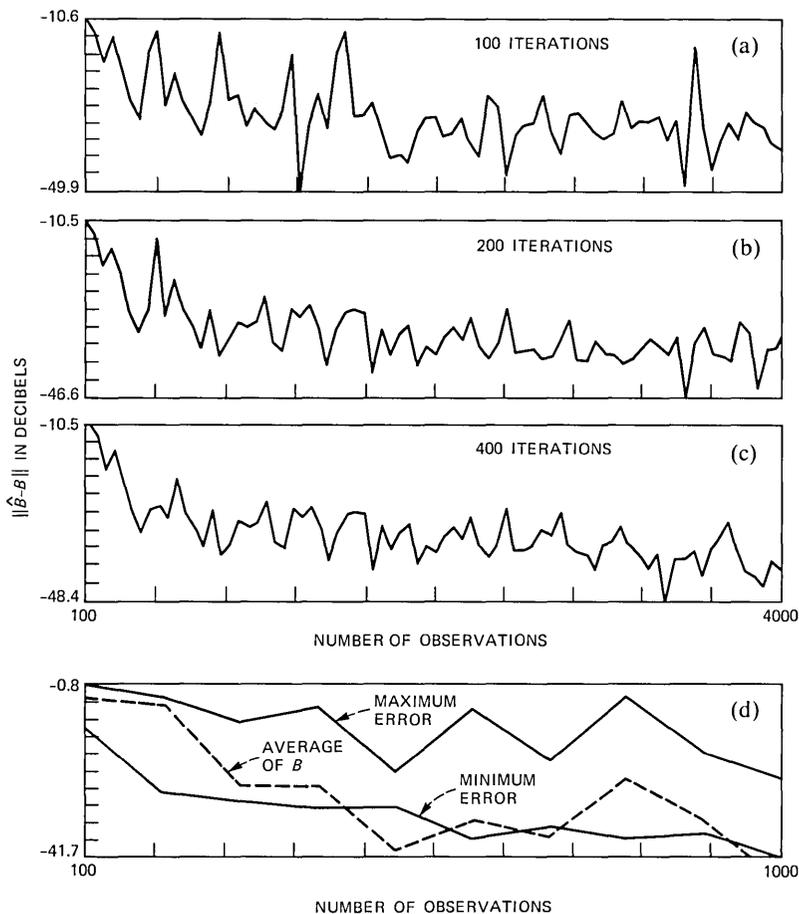


Fig. 4—Estimation error as a function of number of observations for source SRC44 for: (a) 100 iterations maximum, (b) 200 iterations, (c) 400 iterations, and (d) 10 random initial starts with 200 iterations maximum.

estimation error. The technique of minimum bipartite matching (see Appendix B) was used to determine the optimum state permutation.

Plots of the quantity  $\|\hat{B} - B\|$  (on a log scale) versus  $T$ , the number of observations, are given in Fig. 4 for source SRC44. Separate results are shown for 100 iterations maximum (part a) of the  $BW$  reestimation procedure, 200 iterations (part b), and 400 iterations (part c). Also shown in Fig. 4d are the results of using 10 random initial starts with 200 iterations maximum. Shown in Fig. 4d are the maximum and minimum estimation errors for each  $T$  and the estimation error for the average of all 10  $B$  matrices. (The reader should note that  $T$  goes to 4000 in parts a through c, but only to 1000 in part d of Fig. 4.)

The curves given in Fig. 4 show several very interesting properties of the reestimation procedure. First we see that as the number of observations increases, a slow decrease in the average estimation error is obtained. However, we can see that statistical fluctuations (owing to different initial guesses for the model parameters) are often of larger magnitude than the slowly decreasing components of the curve. As the maximum number of iterations increases, the magnitude of the statistical fluctuations decreases, especially for larger values of  $T$ .

The curves of Fig. 4d show that although there is a wide range in the value of the estimation error for multiple starting choices, averaging the  $B$  matrices (after appropriate state alignment) leads to estimation errors comparable with the best single estimates.

Figure 5 shows a similar set of results for the Markov source SRC45. Figure 5a shows a curve of estimation error versus number of observations for a maximum of 400 iterations, Fig. 5b shows the same curve with a maximum of 1000 iterations, Fig. 5c shows the curve when the initial estimates of both  $A$  and  $B$  are set to the source values exactly, and Fig. 5d shows maximum and minimum estimation errors for 10 random starting points.

Although the general trends of the data in Fig. 5 are similar to those of Fig. 4, there are several key differences. From Fig. 5b it can be seen that even for 1000 iterations, the variation in model estimates is enormous (36-dB variations). This result suggests that it is significantly more difficult to estimate parameters of a skewed Markov model than those of a fairly uniform model. The curves of Fig. 5c, in which the initial conditions were set to the source generator values, show that extremely good solutions could be obtained if the reestimation procedure could start in the neighborhood of the "exact" solution. Obviously this situation (i.e., starting near the correct parameter values) is not enforceable for real data.

The curves of Fig. 5d, in which multiple estimates of the Markov model are averaged, show that averaging the individual parameter estimates does not lead to a low error estimate for SRC45. This is

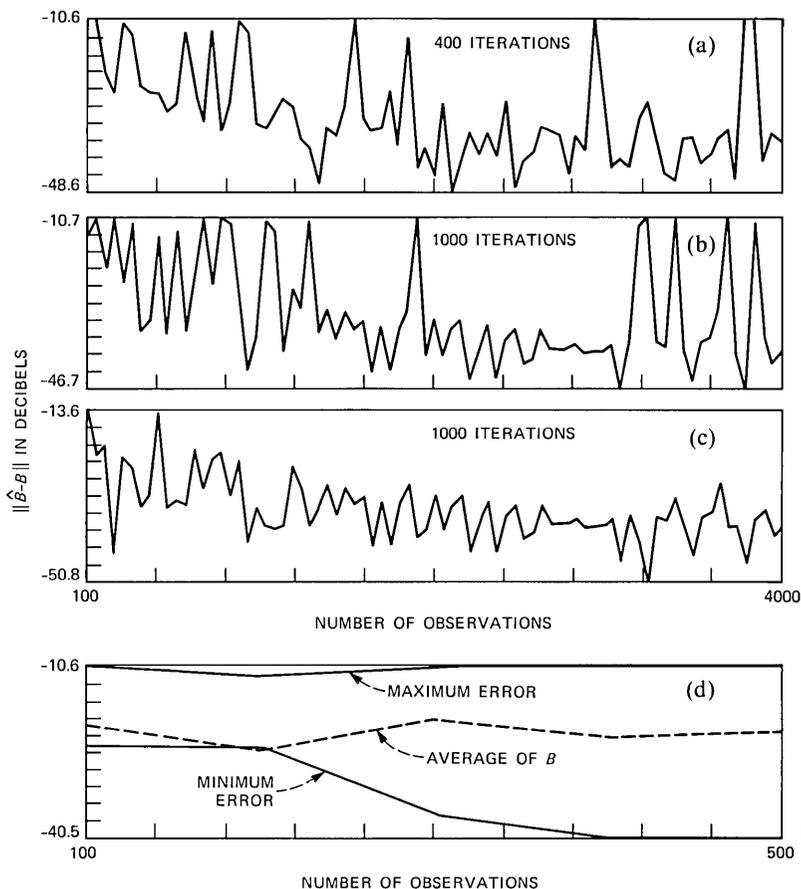


Fig. 5—Estimation error as a function of number of observations for source SRC45 for: (a) 400 iterations maximum, (b) 1000 iterations maximum, (c) initial estimates of  $A$  and  $B$  set to source values, and (d) maximum and minimum estimation errors for 10 random starting points.

undoubtedly because of the parameter estimates with high errors that occur and which have an undue influence on the average.

### 5.1 Left-to-right Markov source estimation

The second series of experiments dealt with the left-to-right Markov models, as would be appropriate for our intended application to isolated word recognition. Figure 6 shows four such models. For each of these models (denoted as SRC195, SRC295, SRC395, and SRC495 in Fig. 6), the specifications were:

$$N = 5, M = 9, \pi = \{1, 0, 0, 0, 0\}$$

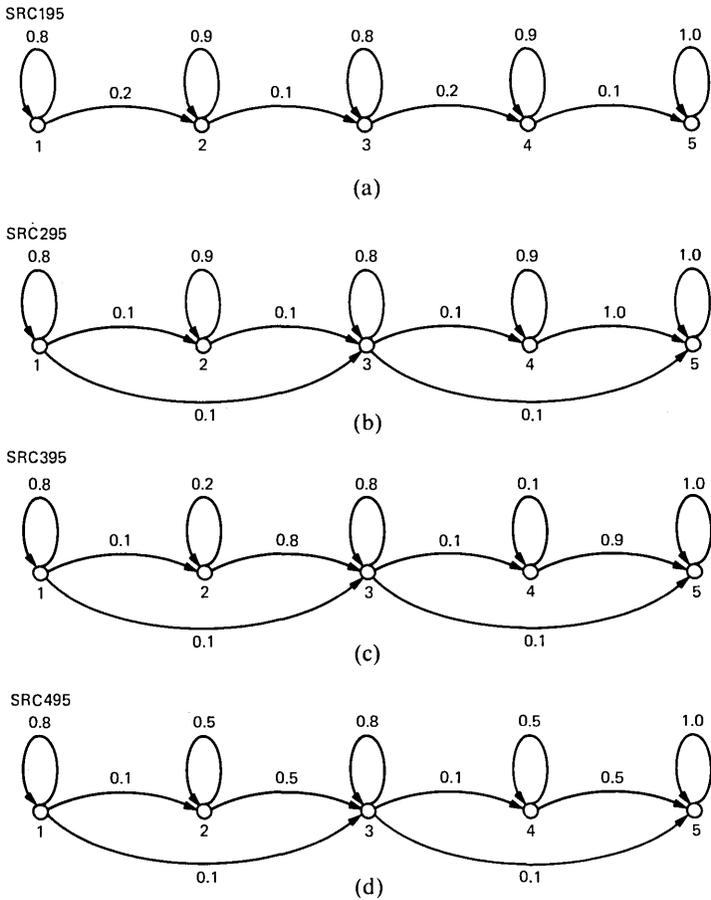


Fig. 6—Left-to-right models used for testing for: (a) SRC195, (b) SRC295, (c) SRC395, and (d) SRC495.

and

$$B = \begin{bmatrix} 0.7 & 0.3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.8 & 0.2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.2 & 0.8 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.3 & 0.7 \end{bmatrix}.$$

The state transition probabilities were those shown in Fig. 6. The SRC195 model is a left-to-right model. The SRC295 model allows a transition between states 1 and 3 and states 3 and 5, as well as transitions between sequentially numbered states. Both the SRC395 and SRC495 models include states whose self-transition probabilities

( $a_{ii}$ ) are very small. We will see below that the average occupancy of such states is only 1 to 2 observations. For these non-ergodic models the concept of occupancy of the transient (i.e., non-absorbing) states is important.

If we denote the probability of a transition from a state to itself as  $p$ , then the probability of a transition out of that state at time  $T + 1$  (assuming the state was entered at time  $t = 0$ ) is

$$\text{Prob}(q_i \text{ at } t = 0, 1, 2, \dots, T \text{ and } q_j \neq q_i \text{ at } T + 1) = p^T(1 - p)$$

for models of the form of SRC195. Hence the average occupancy of a state is given by

$$\begin{aligned} \bar{d} &= \sum_{t=1}^{\infty} (t + 1)p^t(1 - p) \\ &= \frac{1}{1 - p}. \end{aligned} \tag{75}$$

For  $p = 0.9$  we get  $\bar{d} = 10$ , for  $p = 0.8$  we get  $\bar{d} = 5$ , for  $p = 0.5$  we get  $\bar{d} = 2$ , and for  $p = 0.1$  we get  $\bar{d} = 1.1$ . Standard formulas are available for computing average state occupancies for arbitrary transition matrices. We will not consider them here; however, it is clear that states 2 and 4 in models SRC395 and SRC495 are of low occupancy.

To test the reestimation procedure on the Markov sources of Fig. 6, a set of  $K$  sequences were generated for each model, where  $K$  was the set (10, 25, 50, 100). Each sequence was generated using the Markov sequence generator described earlier, modified slightly to ensure that each sequence terminated in state 5 (the final state) and stayed there for 5 observations. The sequences were, however, of variable duration, depending on the exact sequence of state transitions that occurred.

The results showed that for sequences generated from model SRC195, the correct model parameters (to within small estimation errors) were obtained for all values of  $K$  from 10 to 100. For sequences generated from model SRC295 only the 10 observation training sequence yielded grossly incorrect model parameter estimates. All other sequences ( $K = 25, 50, 100$ ) yielded the correct parameter values.

For both source models SRC395 and SRC495, however, no completely correct parameter estimates were obtained. In particular, the states whose expected occupancy was small (i.e., states 2 and 4 in both models) were merged with either the preceding or the following state (or both), while other states whose expected occupancy was larger were often split into 2 states, each with the same set of output symbols. These experiments indicate that it is difficult to reliably estimate parameters of a state, in a left-to-right model, whose average occupancy is very much smaller than that of the states to which it is connected.

## 5.2 Tests on a non-Markov source

In the experiments described above, the observations were in fact generated by a known probabilistic function of a Markov chain. A more difficult test of these techniques was performed in which the observations were 5000 characters from a newspaper article edited to contain only the letters of the alphabet and spaces. We used the observation sequence to train a four-state, 27-symbol model. Of course, the "true" underlying model is not known, as it was in the tests previously described. Nor is it likely that the four-state model is complex enough to model the richness of structure of written English. Even if it were, it is unlikely that 5000 characters is sufficient to capture the structure. Unfortunately, these are exactly the limitations with which the experimenter will be faced in trying to model "real" processes. Our hope was that the text analysis problem would reveal some of the ambiguities that will be encountered in making hidden Markov models of natural phenomena.

The text was first analyzed using the Baum-Welch reestimation formulas with a randomly chosen starting point. The algorithm converged in 310 iterations with  $\log P = -1.317 \times 10^4$ . For purposes of comparison, we analyzed the same data with a quasi-Newton optimization routine, VE01A, from the Harwell Library.<sup>26</sup> It required 125 iterations to obtain a maximum value of  $\log P$  of  $-1.356 \times 10^4$ . In this case some care was required with the parameter values. The finite precision arithmetic occasionally results in a parameter value of  $-10^{-7}$ , which appears to satisfy the positivity constraints. Such a value is fatal to the computation of  $\log P$  since it will result in an attempt to take a logarithm of a negative quantity. Fortunately, this condition is readily detected in the scaling routine and corrected by setting the offending parameter to zero.

Finally, we applied the Lagrangian technique described earlier to the same observation sequence but with a different set of initial parameter values. After 136 iterations, a still different model with  $\log P = -1.327 \times 10^4$  is obtained. These results illustrate some important features of hidden Markov modeling. The computational methods used to obtain the models are roughly equivalent. All of the resulting models capture some of the structure of the data being analyzed. There are many different possible models with very little evidence for selecting the "best" one. For even very simple models, the likelihood function is too complicated to attribute the selection of one model or another by one algorithm or another to its properties. Finally, we note that all of the algorithms tested make large improvements to  $P$  during the early iterations and only slight incremental improvements later. In fact, the last half of the iterations provides no significant change to the model. We have used a convergence criterion

of  $10^{-7}$  on the relative increment between iterations. This may be relaxed substantially, resulting in many fewer iterations with no attendant degradation in the model.

## VI. CONCLUSION

We have presented some of the salient theoretical and practical issues associated with modeling data by probabilistic functions of a Markov chain. In our presentation we have concentrated on three issues: alternatives to the Baum-Welch reestimation algorithm, critical facets of implementation, and behavior of Markov models on certain artificial but realistic signals.

We have observed that, while most of the discussion of parameter estimation for Markov models in the open literature is devoted to the Baum-Welch algorithm, classical optimization techniques are not only a viable alternative but may even be preferable in some cases. In particular, classical techniques are virtually unrestricted by the forms of either the likelihood function or the constraints. The reestimation formulas may be growth transformations for a wider class of functions and constraints than has heretofore been proven; however, it is not likely that a universal reestimation formula exists. For applications to continuous density functions (c.f. Liporace<sup>29</sup>), the classical techniques may have still other advantages.

The open literature has provided only a perfunctory, if any, discussion of some crucial numerical and implementational problems associated with Markov modeling. We have given details of methods of dealing with floating point loss of significance, finite-training-set size, and model stability. Wherever possible we have made our techniques formal and algorithmic.

Finally, we have given several examples of the behavior of Markov modeling techniques on some reasonably realistic data. The most important lesson that can be drawn from these experiments is that even under ideal conditions (i.e., when the data are associated with a known hidden Markov process) and all the more so under realistic conditions, the computed models may contain artifacts and may not faithfully represent the inherent structure of the data. Thus, great caution and empirical validation is required in using these techniques.

Despite this caveat, hidden Markov models may be beneficial in studying many diverse problems. In our companion paper<sup>21</sup> we recount a successful application of this body of theory to a problem in automatic speech recognition.

## REFERENCES

1. J. K. Baker, "The DRAGON System—An Overview," *IEEE Trans. Acoustics, Speech and Signal Processing, ASSP-23* (February 1975), pp. 24-9.

2. J. K. Baker, "Stochastic Modeling for Automatic Speech Understanding," in *Speech Recognition*, R. Reddy, ed., New York: Academic Press, 1975.
3. L. R. Bahl and F. Jelinek, "Decoding for Channels with Insertions, Deletions and Substitutions, with Applications to Speech Recognition," *IEEE Trans. Information Theory*, *IT-21* (July 1975), pp. 404-11.
4. L. R. Bahl, J. K. Baker, P. S. Cohen, A. G. Cole, F. Jelinek, B. L. Lewis, and R. L. Mercer, "Automatic Recognition of Continuously Spoken Sentences from a Finite State Grammar," *Proc. ICASSP* (April 1978), pp. 418-21.
5. L. R. Bahl, J. K. Baker, P. S. Cohen, N. R. Dixon, F. Jelinek, R. L. Mercer, and H. F. Silverman, "Preliminary Results on the Performance of a System for the Automatic Recognition of Continuous Speech," *Proc. ICASSP* (April 1976), pp. 425-9.
6. L. R. Bahl, J. K. Baker, P. S. Cohen, F. Jelinek, B. L. Lewis, and R. L. Mercer, "Recognition of a Continuously Read Natural Corpus," *Proc. ICASSP* (April 1978), pp. 422-4.
7. L. R. Bahl, R. Bakis, P. S. Cohen, A. G. Cole, F. Jelinek, B. L. Lewis, and R. L. Mercer, "Further Results on the Recognition of a Continuously Read Natural Corpus," *Proc. ICASSP* (April 1980), pp. 872-5.
8. L. R. Bahl, R. Bakis, P. S. Cohen, A. G. Cole, F. Jelinek, B. L. Lewis, and R. L. Mercer, "Recognition Results with Several Experimental Acoustic Processors," *Proc. ICASSP* (April 1979), pp. 249-51.
9. F. Jelinek and R. L. Mercer, "Interpolated Estimation of Markov Source Parameters from Sparse Data," in *Pattern Recognition in Practice*, E. S. Gelsema and L. N. Kanal, eds., Amsterdam, The Netherlands: North Holland Publishing Company, 1980.
10. F. Jelinek, "Fast Sequential Decoding Algorithm Using a Stack," *IBM J. Res. Develop.* (1969), pp. 675-85.
11. F. Jelinek, L. R. Bahl, and R. L. Mercer, "Design of a Linguistic Statistical Decoder for the Recognition of Continuous Speech," *IEEE Trans. Information Theory*, *IT-21* (May 1975), pp. 250-6.
12. F. Jelinek, "Continuous Speech Recognition by Statistical Methods," *Proc. IEEE*, *64* (April 1976), pp. 532-56.
13. L. E. Baum, "An Inequality and Associated Maximization Technique in Statistical Estimation for Probabilistic Functions of a Markov Process," *Inequalities*, *3* (1972), pp. 1-8.
14. L. E. Baum and J. A. Eagon, "An Inequality with Applications to Statistical Estimation for Probabilistic Functions of a Markov Process and to a Model for Ecology," *Bull. AMS*, *73* (1967), pp. 360-3.
15. L. E. Baum and T. Petrie, "Statistical Inference for Probabilistic Functions of Finite State Markov Chains," *Ann. Math. Stat.*, *37* (1966), pp. 1559-63.
16. L. E. Baum, T. Petrie, G. Soules, and N. Weiss, "A Maximization Technique Occurring in the Statistical Analysis of Probabilistic Functions of Markov Chains," *Ann. Math. Stat.*, *41* (1970), pp. 164-71.
17. L. E. Baum and G. R. Sell, "Growth Transformations for Functions on Manifolds," *Pac. J. Math.*, *27* (1968), pp. 211-27.
18. D. S. Passman, "The Jacobian of a Growth Transformation," *Pac. J. Math.*, *44* (1973), pp. 281-90.
19. P. Stebe, "Invariant Functions of an Iterative Process for Maximization of a Polynomial," *Pac. J. Math.*, *43* (1972), pp. 765-83.
20. L. R. Rabiner and S. E. Levinson, "Isolated and Connected Word Recognition—Theory and Selected Applications," *IEEE Trans. Commun.*, *COM-29* (May 1981), pp. 621-59.
21. L. R. Rabiner, S. E. Levinson, and M. M. Sondhi, "On the Application of Vector Quantization and Hidden Markov Models to Speaker-Independent Isolated Word Recognition," *B.S.T.J.*, this issue.
22. A. A. Markov, "An Example of Statistical Investigation in the Text of 'Eugene Onyegin' Illustrating Coupling of Tests in Chains," *Proc. Acad. Sci. St. Petersburg*, *7* (1913), pp. 153-62.
23. A. J. Viterbi, "Error Bounds for Convolutional Codes and an Asymptotically Optimal Decoding Algorithm," *IEEE Trans. Information Theory*, *IT-13* (April 1967), pp. 260-9.
24. J. B. Rosen, "The Gradient Projection Method for Nonlinear Programming—Part I: Linear Constraints," *J. Soc. Indust. Appl. Math.*, *8* (1960), pp. 181-217.
25. R. Fletcher and M. J. D. Powell, "A Rapidly Convergent Descent Method for Minimization," *Computer J.*, *6* (1963), pp. 163-8.

26. M. J. Hopper, ed., *Harwell Subroutine Library: A Catalog of Subroutines*, 55, Oxfordshire, England: A.E.R.E. Harwell, 1979.
27. M. R. Hestenes, *Optimization Theory*, New York: Wiley, 1975.
28. E. Lawler, *Combinatorial Optimization*, New York: Holt Rinehart and Winston, 1976.
29. L. R. Liporace, "Maximum Likelihood Estimation for Multivariate Observations of Markov Sources," *IEEE Trans. Information Theory, IT-28* (September 1982), pp. 729-34.

## APPENDIX A

Several of the formulas derived in the text are much more compact in matrix notation. Let ' denote matrix transposition, as usual, and let the column vectors  $\pi$  and  $\mathbf{1}$ , and the matrices  $A, B_t, t = 1, \dots, T$  be defined as in the text. Also let  $\alpha_t$  and  $\beta_t$  be column vectors with components  $\alpha_t(i), i = 1, \dots, N$  and  $\beta_t(i), i = 1, \dots, N$ , respectively. Then the recursion for  $\alpha_t$  is

$$\alpha_{t+1} = B_{t+1}A'\alpha_t, \quad t = 1, \dots, T-1. \quad (76)$$

The recursion for  $\beta_t$  is

$$\beta_t = AB_{t+1}\beta_{t+1} \quad t = T-1, \dots, 1. \quad (77)$$

The starting values are

$$\begin{aligned} \alpha_1 &= B_1\pi \\ \beta_T &= \mathbf{1}. \end{aligned} \quad (78)$$

The probability  $P$  is given by

$$P = \beta_t'\alpha_t \quad \text{for any } t \text{ in } (1, T). \quad (79)$$

The special cases  $t = 1$  and  $t = T$  give

$$P = \pi'B_1\beta_1 \quad (80)$$

and

$$\begin{aligned} P &= \mathbf{1}'\alpha_T \\ &= \mathbf{1}'B_TA'B_{T-1} \cdots A'B_1\pi. \end{aligned} \quad (81)$$

In each of these formulas  $P$  can be regarded as the trace of a  $1 \times 1$  matrix, which [as expanded in (81)], is a product of several matrices. The fact that the trace of a product of matrices is invariant to a cyclic permutation of the matrices can be used to advantage in finding the gradient of  $P$ . Define  $\nabla_A P$  as the matrix whose  $ij$ th component is  $\partial P / \partial a_{ij}$ . Similarly, define  $\nabla_B P$  and  $\nabla_\pi P$ . Then it is straightforward to show that

$$\begin{aligned} \nabla_\pi P &= B_1\beta_1 \\ \nabla_A P &= \sum_{t=1}^{T-1} \alpha_t \beta_{t+1}' B_{t+1} \\ (\nabla_B P)_{jk} &= \sum_{t \ni O_t=k} (A'\alpha_{t-1})_j (\beta_t)_j. \end{aligned} \quad (82)$$

In the last equation, if  $O_1 = v_k$  then the corresponding term in the sum is just  $\pi'\beta_1$ .

## APPENDIX B

As we mentioned in Section 3.3, it is frequently necessary to compare (or average) two different estimates  $B$  and  $\tilde{B}$  of the symbol matrix. The optimization procedures, in general, relabel the states; therefore the rows of  $\tilde{B}$  may, in general, be permuted relative to those of  $B$ . Before comparison, therefore, the optimum permutation must be found. This is defined as one that minimizes the distance  $D$  defined in eq. (66). The problem of finding this permutation can be converted to a network optimization problem called "bipartite weighted matching." To this end define  $w_{ij}$  as the distance of the  $i$ th row of  $B$  from the  $j$ th row of  $\tilde{B}$ . As we have done in Fig. 7, draw two sets  $P$  and  $Q$  of  $N$  vertices each. For  $1 \leq i, j \leq N$ , draw an edge from the  $i$ th vertex in  $P$  to the  $j$ th vertex in  $Q$ , and label this edge with the weight  $w_{ij}$ . The resulting graph is a complete weighted bipartite graph, and the problem is to find an  $N$ -match (i.e., one to one matching with  $N$  edges) such that it has minimum weight.

Suppose that  $Z_k$  is a  $k$ -match ( $k \leq N$ ). With respect to this match make the following definitions:

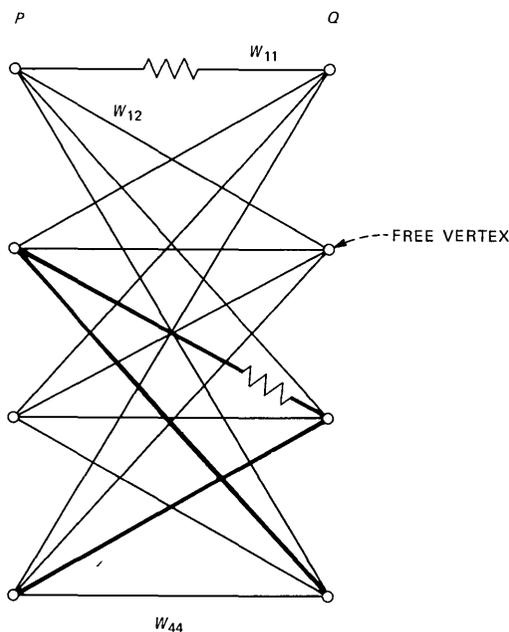


Fig. 7—Complete bipartite weighted graph ( $N = 4$ ) and a 2-match (matched edge  $\rightarrow$ ). Path shown in heavy lines is an alternating path for the 2-match. It also happens to be an augmenting path.

- (i) A matched edge is an edge in  $Z_k$ .
- (ii) A free vertex is one that is not on a matched edge.
- (iii) An alternating path is a path along edges that alternately belong to  $Z_k$  and do not belong to  $Z_k$ . [N.B. the number of matched edges,  $m$ , on an alternating path may have any value  $m \leq k$ . The degenerate case  $m = 0$  is also valid.]
- (iv) An augmenting path is an alternating path between two free vertices. Again note that a single edge connecting two free vertices is a valid augmenting path.

An augmenting path has the structure

$$p_1 U q_1 M p_2 U q_2 \cdots U q_j. \quad (83)$$

Here  $U$  represents an unmatched edge,  $M$  represents a matched edge, and  $p_i$  and  $q_j$  are the only free vertices on the path. (Here  $p_i$  is the  $i$ th  $P$ -vertex along the path. The  $q_i$  are similarly defined.) Note that the number of  $U$ 's on an augmenting path is exactly one more than the number of  $M$ 's. Hence the total number of edges in an augmenting path is always odd.

Suppose we are given a  $k$ -match  $Z_k$  and an augmenting path  $\mathbf{ap}$  of length  $2j + 1$ . Then we can obtain a  $(k + 1)$ -match  $Z_{k+1}$  by a complementary labeling of the edges of  $\mathbf{ap}$  (i.e., by changing every  $U$  to  $M$  and vice versa). If  $w_1, w_2, \dots, w_{2j+1}$  are the weights along  $\mathbf{ap}$ , then the weight  $Z_{k+1}$  exceeds that of  $Z_k$  by the amount  $w_1 - w_2 + w_3 - \dots + w_{2j+1}$ .

This method of obtaining a  $(k + 1)$ -match from a  $k$ -match has the following key property (proof given below): Suppose  $\mathbf{M}_k$  is a minimum-weight  $k$ -match. Let  $\mathbf{apm}$  be an augmenting path for  $\mathbf{M}_k$  with minimum incremental weight. Then the match  $\mathbf{M}_{k+1}$  obtained from  $\mathbf{M}_k$  and  $\mathbf{apm}$  is a minimum weight  $(k + 1)$ -match.

Assuming this property for the moment, a minimum-weight  $N$ -match can be determined by the following  $N$ -step algorithm:

For  $k = 1, 2, \dots, N$  generate  $\mathbf{M}_k$  by finding an optimum augmenting path for  $\mathbf{M}_{k-1}$ . (Note that  $\mathbf{M}_0$  is an empty set.)

We now show that finding an optimum augmenting path is equivalent to a shortest path problem. With reference to Fig. 8 let us generate a directed graph by directing each edge in  $\mathbf{M}_k$  to the left and each unmatched edge to the right. Also let us multiply the weights of all matched edges by  $-1$ . Finally, add two vertices labeled  $s$  and  $t$ . Connect  $s$  to all *free* vertices in  $P$  by right-going edges of zero weight. Similarly, connect all *free* vertices of  $Q$  to  $t$  by right-going edges of zero weight. In this directed graph any path from  $s$  to  $t$  is an augmenting path for the matching  $\mathbf{M}_{k-1}$ . Hence, if we interpret the weights as lengths, it is clear that an optimum augmenting path is a shortest path from  $s$  to  $t$ .

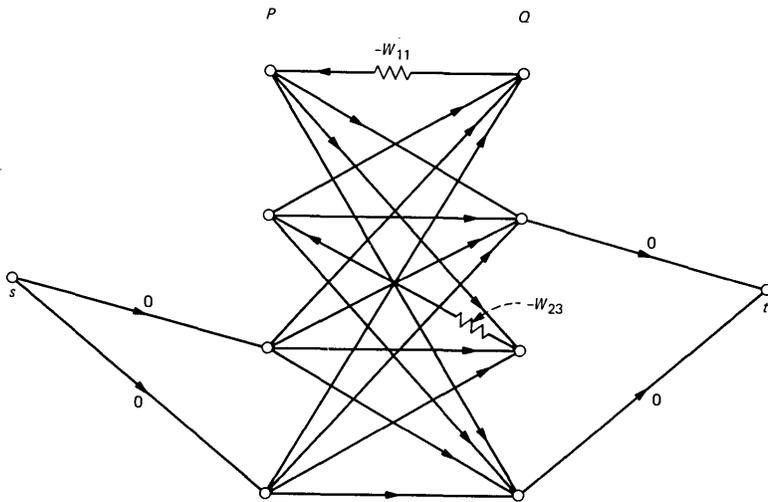


Fig. 8—Directed graph obtained from Fig. 7. Every path from  $s$  to  $t$  is an augmenting path of the 2-match (except for the dummy edges of 0 weight from  $s$  and to  $t$ ). Interpreting weights as lengths, the length of a path from  $s$  to  $t$  is the incremental weight of the corresponding augmenting path.

A shortest path problem can be solved in polynomial time. However, the problem can be solved much more easily and efficiently if the path lengths are all nonnegative. In that case the problem can be solved in  $N^2$  time by Dijkstra's method.<sup>28</sup>

It is possible to avoid negative distances in our problem by the method of assigning a "potential"  $f(v)$  to each vertex. Suppose that at step  $k$  the graph has nonnegative weights, and the edges corresponding to  $\mathbf{M}_{k-1}$  have zero weight. Then use Dijkstra's method to find shortest paths to all vertices (including  $t$ ) from  $s$ . Define  $f(v)$  for the vertex  $v$  as its shortest distance from  $s$ . Next, modify the weight  $w_{ij}$  of the edge from  $i$  to  $j$  to

$$w'_{ij} = w_{ij} + f(i) - f(j). \quad (84)$$

It is easily seen that this procedure leaves all weights nonnegative, does not alter shortest paths, and all shortest paths have weight 0. Reversing a shortest path from  $s$  to  $t$  gives us a matching  $\mathbf{M}_k$ , and the new graph has nonnegative weights and zero weights for the matched edges. Now  $\mathbf{M}_0$  trivially has the postulated properties. Therefore, at every step the graph will have these properties.

An important property of the above procedure is that if a vertex  $p$  (or  $q$ ) is on some matched edge in  $\mathbf{M}_k$ , then it will be on some matched edge in  $\mathbf{M}_{k+1}$  also. In writing the actual computer program, this property simplifies the bookkeeping considerably.

Another important property concerns the vertex potentials. Suppose

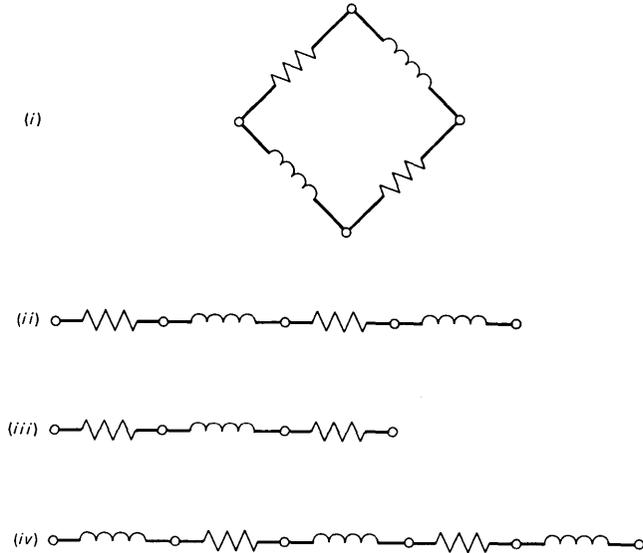


Fig. 9—Types of paths in the symmetric difference between  $M_k$  and  $N_{k+1}$ . There must be exactly one more path of type (iv) than of type (iii).

$F(v)$  is the sum of the potentials assigned to vertex  $v$  at each of the  $N$  steps. Suppose  $w_{ij}$  and  $d_{ij}$  are the original and final weights, respectively, of the edge connecting vertex  $i$  to vertex  $j$ . Then

$$d_{ij} = w_{ij} + F(i) - F(j) \geq 0, \tag{85}$$

and  $d_{ij} = 0$  for every edge in the final  $N$ -match. The numbers  $F(v)$  thus provide a simple proof that the final match is indeed an optimum match.

We turn now to a proof of the key property mentioned above. Let  $M_k$  be an optimum  $k$ -match and let  $N_{k+1}$  be any optimum  $k + 1$ -match. Then we will show that there is a  $k + 1$ -match obtained from an augmenting path of  $M_k$  such that its weight is equal to that of  $N_{k+1}$ . For this purpose define  $S$  as the set of edges in the symmetric difference of  $M_k$  and  $N_{k+1}$ . (Recall that the symmetric difference of two sets,  $A$  and  $B$ , is the set of elements that belong either to  $A$  or to  $B$  but not to both.)

From the geometry of a bipartite graph three properties of the set  $S$  are obvious:

(i) The number of edges in  $S$  which belong to  $N_{k+1}$  must exceed the number that belongs to  $M_k$  by exactly 1.

(ii) The edges on any path in  $S$  must alternate between  $M_k$  and  $N_{k+1}$ .

(iii) A vertex on an edge in  $\mathbf{S}$  cannot be shared with any edge in  $\mathbf{M}_k$  or  $\mathbf{N}_{k+1}$  that does not belong to  $\mathbf{S}$ .

From the first two of these properties it follows that there can be four types of paths in  $\mathbf{S}$  (see Fig. 9).

(i) Circuits, each consisting of an even number of edges;

(ii) Open paths, each with an even number of edges;

(iii) Open paths, each with an odd number of edges beginning with an edge in  $\mathbf{M}_k$ ;

(iv) Open paths, each with an odd number of edges beginning with an edge in  $\mathbf{N}_{k+1}$ . The number of such paths must be exactly one more than the number of paths of type (iii).

It is easily seen that the incremental weight of every path of type (i) or (ii) must be exactly zero. (If it is negative then the weight of  $\mathbf{M}_k$  can be decreased by a complementary labeling of the path; if positive then the weight of  $\mathbf{N}_{k+1}$  can be decreased in the same way. But this contradicts the hypothesis that  $\mathbf{M}_k$  and  $\mathbf{N}_{k+1}$  are optimum matches.)

In view of this and property (3),  $\mathbf{N}_{k+1}$  can be modified by replacing its edges on all paths of types (i) and (ii) by the corresponding edges of  $\mathbf{M}_k$ . The modified  $k + 1$ -match has exactly the same weight as that of  $\mathbf{N}_{k+1}$ ; however, the symmetric difference between  $\mathbf{M}_k$  and the modified  $k + 1$ -match has no paths of type (i) or (ii).

The same argument applies to pairs of paths, one path each of types (iii) and (iv). Thus a final modified  $k + 1$  match is obtained whose symmetric difference with  $\mathbf{M}_k$  is exactly one path of type (iv). The weight of this final modified  $k + 1$ -match is exactly the same as that of the original  $\mathbf{N}_{k+1}$ , and is obtained from an augmenting path of  $\mathbf{M}_k$ .

We have written a subroutine that implements the above procedure. The timing, from a number of test runs, is approximately  $0.063N^3$  ms central processing unit (CPU) time on the MV-8000. Thus for  $N = 40$  about 4 seconds of CPU time is needed.

## On the Application of Vector Quantization and Hidden Markov Models to Speaker-Independent, Isolated Word Recognition

By L. R. RABINER, S. E. LEVINSON, and M. M. SONDHI

(Manuscript received September 10, 1982)

*In this paper we present an approach to speaker-independent, isolated word recognition in which the well-known techniques of vector quantization and hidden Markov modeling are combined with a linear predictive coding analysis front end. This is done in the framework of a standard statistical pattern recognition model. Both the vector quantizer and the hidden Markov models need to be trained for the vocabulary being recognized. Such training results in a distinct hidden Markov model for each word of the vocabulary. Classification consists of computing the probability of generating the test word with each word model and choosing the word model that gives the highest probability. There are several factors, in both the vector quantizer and the hidden Markov modeling, that affect the performance of the overall word recognition system, including the size of the vector quantizer, the structure of the hidden Markov model, the ways of handling insufficient training data, etc. The effects, on recognition accuracy, of many of these factors are discussed in this paper. The entire recognizer (training and testing) has been evaluated on a 10-word digits vocabulary. For training, a set of 100 talkers spoke each of the digits one time. For testing, an independent set of 100 tokens of each of the digits was obtained. The overall recognition accuracy was found to be 96.5 percent for the 100-talker test set. These results are comparable to those obtained in earlier work, using a dynamic time-warping recognition algorithm with multiple templates per digit. It is also shown that the computation and storage requirements of the new recognizer were an order of magnitude less than that required for a conventional pattern recognition system using linear prediction with dynamic time warping.*

## I. INTRODUCTION

There currently exist two standard approaches to isolated word recognition, namely, feature extraction methods and statistical pattern recognition models. A statistical pattern recognition approach has the property of being a nonparametric approach to recognition and therefore is widely used in most commercial and industrial recognizers.<sup>1-6</sup> The feature-based approach to recognition has been primarily used in the (computationally) less expensive systems, and as a basis for recognition of continuous speech (in conjunction with segmentation and labeling algorithms).<sup>4-9</sup>

In the past few years a new approach to speech processing has been proposed, namely, using probabilistic functions of Markov models. This approach has been applied at the Institute for Defense Analyses for speaker recognition,<sup>10</sup> and at Carnegie Mellon University and IBM to solve problems in continuous speech recognition<sup>11,12</sup> with good success. Based on its success in these related areas of speech processing, a question that arises naturally is how well these probabilistic models would work on problems in isolated word recognition.

It is the prime purpose of this paper to provide an answer to the question posed above. Before discussing the approach we have taken to get at the answer, we must first describe the structure of a word recognition system based on (hidden) Markov models (HMM). As in most recognition systems we assume we have a labeled training set of data from which we build a series of Markov models, one for each vocabulary word. Then when we want to recognize an unknown token, we compute a probability score for each word HMM on that token, and choose as the recognized word the one corresponding to the model with the highest probability score (i.e., the most likely word HMM). Techniques for training and scoring such HMMs are discussed both here and in the companion paper.<sup>3</sup>

In a conventional pattern recognition system the unknown test token is time-aligned in turn to each reference pattern via some form of time-warping procedure, typically, dynamic time warping (DTW). By contrast, no such direct alignment is performed in the HMM system; only an indirect time alignment is obtained based on the probabilistic scoring. Thus it is interesting to study the relationship between probabilistic scoring and DTW as applied to isolated word recognition. As we shall see, there is no simple relationship. We will point out several similarities and differences in the two approaches.

The organization of this paper is as follows. In Section II we briefly review the conventional DTW word recognizer based on LPC modeling, since this will be the focus of comparison throughout the paper. In Section III we review the basic ideas behind the use of HMMs for isolated word recognition. It is the purpose of this section to establish

notation and terminology that will define the basic parameters of interest in the HMM system. Section III shows that one inherent feature of the HMM recognizer (as we have implemented it) is that the models need a discrete, finite set of observations (input data) to obtain the best model parameters for each word in the vocabulary. A vector quantizer (VQ) was used to transform the continuous set of linear predictive coefficient (LPC) vectors into a finite observation set. Therefore, in Section IV we describe the key ideas behind vector quantization of LPC sets, and discuss the particular implementation that we used. In Section V we describe the overall structure of the HMM isolated word recognizer. In Section VI we describe a series of experiments used to evaluate the performance of the HMM word recognizer and compare it to the performance of a DTW recognizer on the same vocabulary. The effects, on performance, of several parameter variations in the HMM and VQ are also described in this section. In Section VII we discuss the results of the performance evaluation and comparison experiments. The strengths and weaknesses of the HMM word recognizer are discussed, along with computational and storage comparisons of HMM and DTW word recognizers. We attempt, in this section, to determine the fundamental relationships between the HMM and DTW systems.

## II. REVIEW OF CONVENTIONAL DTW WORD RECOGNIZER BASED ON LPC MODELING

Figure 1 shows a block diagram of the LPC-based isolated word recognizer.<sup>2,3</sup> The input speech signal,  $s(n)$ , recorded over a standard dialed-up telephone line, is bandpass-filtered between 100 and 3200 Hz, and digitized at a 6.67-kHz rate. The first step in the processing is

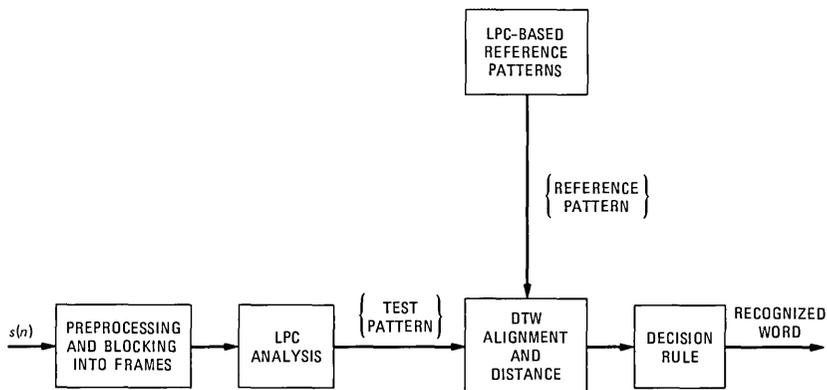


Fig. 1—Block diagram of conventional LPC-based word recognizer using a standard dynamic time-warping algorithm for registering test and reference patterns.

the preprocessing block, a first-order digital network, which provides a high-frequency pre-emphasis to the speech. The pre-emphasized signal is blocked into frames of 45 ms (300 samples) with each consecutive frame spaced 15 ms (100 samples) apart. An 8-pole LPC analysis (autocorrelation method) is performed on each frame of the word (after isolating it with an endpoint detector<sup>14</sup>), thus creating the test pattern. This test pattern is compared with each reference pattern using a DTW alignment algorithm that simultaneously provides a distance score associated with the alignment. The distance scores for all the reference patterns are sent to a decision rule, which provides a classification of the spoken word, and possibly an ordered (by distance) set of the best  $n$  candidates.

The word reference patterns for the recognizer of Fig. 1 are created by a training algorithm. For speaker-trained applications, typically a single reference pattern is created for each word in the vocabulary using a robust training algorithm.<sup>15</sup> For speaker-independent applications, a set of  $Q$  reference patterns is created for each vocabulary word using a clustering procedure.<sup>16,17</sup> Typically, about 12 templates per word are sufficient for recognizing words from a fairly homogeneous adult population of native American talkers.

### III. BASICS OF HMM FOR WORD RECOGNITION

We assume we have a finite sequence,  $\mathbf{O}$ , of observations,

$$\mathbf{O} = O_1 O_2 \cdots O_T, \quad (1)$$

where each observation is a discrete symbol drawn from a finite alphabet of symbols. (For the system we will be describing, the observations are the indices of the LPC vectors obtained from an LPC vector quantizer.) We further assume that the sequence of observations may be modeled as a probabilistic function of an underlying Markov chain whose state transitions are not directly observable; hence the name "Hidden Markov Model." Figure 2 shows such a model,  $\mathbf{M}$ , which is characterized by the following:

(i)  $N =$  the number of states in the model. For the model of Fig. 2,  $N = 5$ .

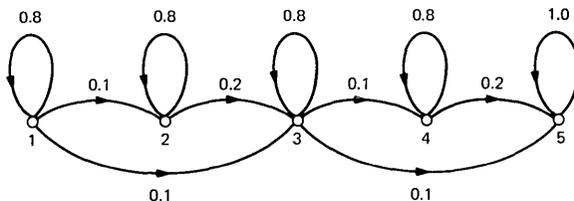


Fig. 2—A typical state diagram for a 5-state Markov model.

(ii)  $M$  = the number of output symbols in the discrete alphabet of the model. For the present example,  $M = 5$ .

(iii)  $A = \{a_{ij}\}$ , the transition matrix of the underlying Markov chain. Here,  $a_{ij}$  is the probability of making a transition to state  $j$ , given that the model is in state  $i$ . For the model of Fig. 2 we have

$$A = \begin{bmatrix} 0.8 & 0.1 & 0.1 & 0 & 0 \\ 0 & 0.8 & 0.2 & 0 & 0 \\ 0 & 0 & 0.8 & 0.1 & 0.1 \\ 0 & 0 & 0 & 0.8 & 0.2 \\ 0 & 0 & 0 & 0 & 1.0 \end{bmatrix}.$$

Note that only 11 of the 25  $a_{ij}$ 's are nonzero.

(iv)  $B = \{B_{jk}\} = \{b_j(k)\}$ , the model output symbol probability matrix, where  $b_j(k)$  is the probability of outputting symbol  $k$ , given that the model is in state  $j$ . For the example chosen,

$$B = \begin{bmatrix} 0.5 & 0.5 & 0 & 0 & 0 \\ 0 & 0.5 & 0.5 & 0 & 0 \\ 0 & 0 & 0.5 & 0 & 0.5 \\ 0.5 & 0 & 0 & 0.5 & 0 \\ 0 & 0 & 0 & 0.5 & 0.5 \end{bmatrix}.$$

(v)  $\pi = \{\pi_i\}$ ,  $i = 1, 2, \dots, N$ , the initial state probability vector. For the left-to-right models of the type shown in Fig. 2, we assume the system always begins in state 1, i.e.,  $\pi_1 = 1$ ,  $\pi_i = 0$ ,  $i \neq 1$ .

Isolated word recognition using HMM consists of two phases, training and recognition (or classification). In the training phase, the training set of observations is used to derive a set of reference models of the above type, one for each word in the vocabulary. In the classification phase, the probability of generating the test observation is computed for each reference model. The test is classified as the word whose model gives the highest probability. The computations in each of these phases are fairly straightforward.

Let us begin with the classification phase. Given the observation sequence,  $\mathbf{O}$ , and a model,  $\mathbf{M}$  (i.e.,  $N, M, A, B$ , and  $\pi$ ), the probability of  $\mathbf{O}$  having been generated by model  $\mathbf{M}$  is

$$P(\mathbf{O}|\mathbf{M}) = \sum_{i_1, i_2, \dots, i_T} \pi_{i_1} b_{i_1}(O_1) a_{i_1 i_2} \cdots a_{i_{T-1} i_T} b_{i_T}(O_T). \quad (2)$$

The summation in eq. (2) is more readily computed by defining a forward partial probability,  $\alpha_t(i)$ , as

$$\alpha_t(i) = P(O_1 O_2 \cdots O_t \text{ and state } i \text{ at time } t | \mathbf{M}). \quad (3)$$

This leads to the recursion

$$\alpha_{t+1}(j) = \left[ \sum_{i=1}^N \alpha_t(i) a_{ij} \right] b_j(O_{t+1}), \quad t = 1, 2, \dots, T-1 \quad (4)$$

by which eq. (2) can be expressed as

$$P(\mathbf{O}|\mathbf{M}) = P = \sum_{j=1}^N \alpha_T(j). \quad (5)$$

In the training phase an initial estimate of the model is made, and  $P$  is computed for the training observation sequence according to eq. (5). Next the model is iteratively adjusted to increase  $P$ . The iterations are stopped when  $P$  stops increasing significantly, or when some other stopping criterion is met (e.g., when the number of iterations exceeds a limit).

One remarkable algorithm for improving a trial model is the Baum-Welch reestimation algorithm.<sup>18</sup> However, maximizing  $P$  can also be looked upon as a constrained optimization problem, for which many algorithms have been proposed. In the companion paper in this issue of the Journal,<sup>13</sup> we discuss the relative merits of these procedures.

We now discuss a number of factors that influence the performance of HMM recognizers.

### 3.1 Initial estimates of $A$ and $B$

One factor of interest for the HMM recognizer is the choice of initial estimates for the elements of the matrices  $A$  and  $B$ . The problem here is that although the training procedure is guaranteed to reach a critical point of  $P$ , the value of  $P$  obtained is typically a local maximum. Hence, alternative starting values of  $A$  and  $B$  could yield models with higher (or lower) values of  $P$ . For our simulations we have chosen to start the training models with essentially random choices for the nonzero elements of both  $A$  and  $B$ , normalized to satisfy the constraints

$$\sum_{j=1}^N a_{ij} = 1 \quad i = 1, 2, \dots, N \quad (6a)$$

$$\sum_{k=1}^M b_j(k) = 1 \quad j = 1, 2, \dots, N. \quad (6b)$$

An alternative starting condition could be

$$a_{ij} = 1/N + \epsilon \quad (7a)$$

$$b_j(k) = 1/M + \epsilon, \quad (7b)$$

where  $\epsilon$  is a uniformly distributed random variable whose peak is much smaller than either  $1/N$  or  $1/M$ . [Again the  $a_{ij}$ 's and  $b_j(k)$ 's of eq. (7) must be normalized using eq. (6) prior to running the optimization.]

### 3.2 HMM structures and the number of states

A second factor affecting the determination of optimum HMMs for each vocabulary word is the model structure and the number of states.

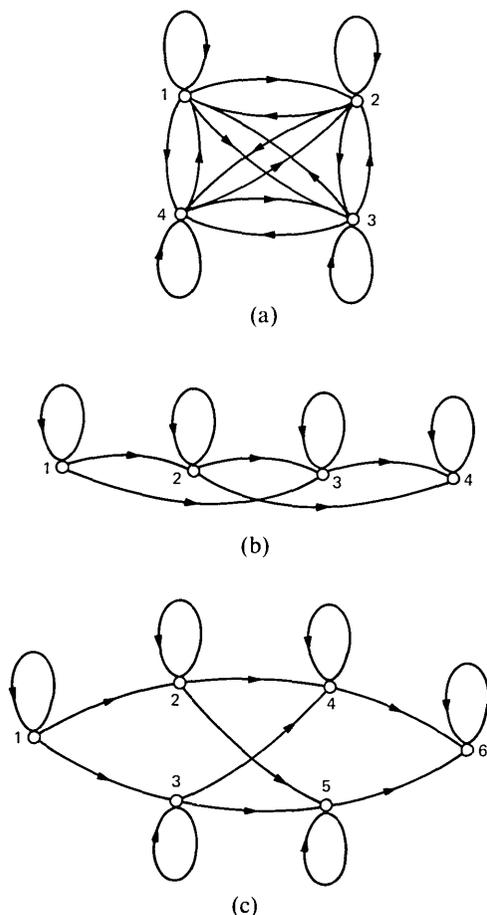


Fig. 3—State diagrams for: (a) unconstrained Markov model with four states, (b) constrained serial Markov model with four states, and (c) constrained parallel Markov model with six states.

We have considered three types of model structures, namely unconstrained, constrained serial, and constrained parallel. Typical examples of each of these models are shown in Fig. 3. In the unconstrained models (shown in Fig. 3a) a transition from any state to any other state can be made—i.e., all  $a_{ij}$ 's are allowed to be nonzero. Both the constrained serial models (shown in Fig. 3b) and the constrained parallel models (shown in Fig. 3c) are left-to-right models, i.e., the state transition matrix  $A$  is upper triangular. The serial models generally proceed sequentially through the states (although individual states can be skipped over), whereas the parallel models allow multiple paths through the model, with each path skipping one or more model states. For example, there are four distinct paths through the model of

Fig. 3c, 1-2-4-6, 1-2-5-6, 1-3-4-6 and 1-3-5-6, each of which traverses four of the six model states.

Each of the model structures of Fig. 3 can be generalized to include an arbitrary number of states. Recall, however, that the number of free parameters of the Markov model is on the order of  $N^2$  (for the  $A$  matrix) plus  $NM$  (for the  $B$  matrix). Hence, if  $N$  gets too large, accurate and reliable determination of the optimum  $A$ 's and  $B$ 's may become difficult for a fixed-size training set. However, within these constraints we have investigated models with as few as two states, and as many as 20 states. There appears to be no good theoretical way to choose the number of states needed for a word model, since the states need not be physically related to any single observable phenomenon.

### 3.3 Multiple observation sequences

A third factor affecting the determination of the optimum HMM for each vocabulary word is the observation sequence used for training. Since we are interested in obtaining speaker-independent models, the observation sequence,  $\mathbf{O}$ , actually consists of several independent sequences  $\mathbf{O}^{(k)}$ ,  $k = 1, 2, \dots, K$ , where  $\mathbf{O}^{(k)}$  is the training sequence for talker  $k$ , and  $K$  is the total number of talkers used for training. Typically, a value of  $K = 100$  has been used in our clustering work for speaker-independent training. The way in which we handle multiple sequences is to calculate  $P(\mathbf{O}^{(k)}|\mathbf{M})$ , using eq. (5), for each sequence, and maximize the product of the probabilities, i.e.,

$$P = \prod_{k=1}^K P(\mathbf{O}^{(k)}|\mathbf{M}). \quad (8)$$

The implementation of the computation of eq. (8) is straightforward for the Baum-Welch reestimation procedure, as well as for the gradient methods.<sup>13</sup> Thus the fact that the training data consist of multiple sequences causes no problem in estimating the optimum HMM parameters.

### 3.4 Constraints on $A$ , $B$ matrices during training

As we show in Fig. 3 we have considered three general HMM structures. For the unconstrained structure the  $A$  and  $B$  matrix elements are allowed to assume any value consistent with the stochasticity constraints. For the constrained serial models we have considered two general constraints, namely:

$$\text{SC1: } a_{ij} = 0 \quad \text{for } j < i \quad \text{and } j \geq i + 3 \quad (\text{double skip allowed}) \quad (9a)$$

$$\text{SC2: } a_{ij} = 0 \quad \text{for } j < i \quad \text{and } j \geq i + 2 \quad (\text{single skip allowed}). \quad (9b)$$

These two cases are illustrated in Fig. 4 for a 5-state model. Constraints SC1 allow single- or double-state jumps when exiting a given state,

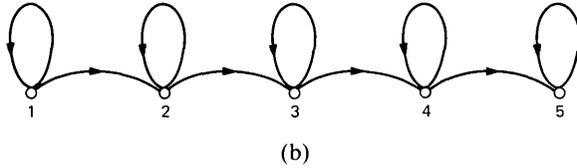
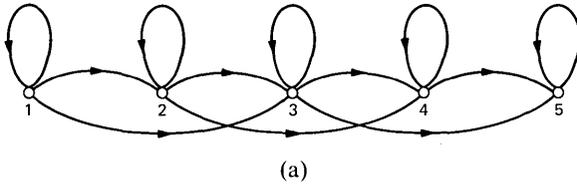


Fig. 4—Markov models for two types of serial constraints: (a) single and double transitions allowed, and (b) only single transitions allowed.

whereas constraints SC2 only allow single jumps when exiting a state. Hence, models of the type shown in Fig. 4a have somewhat more flexibility than those of the type shown in Fig. 4b.

For the constrained parallel models we have only considered transition matrices of the type illustrated in Fig. 3c, i.e., a given state can only exit to either of a pair of states in the next column of the grid.

For the most part we have not constrained the  $B$  matrix. However, one problem arises if the  $B$  matrix is left completely unconstrained. The problem is that a finite training sequence of length  $T$  may result in  $b_j(k) = 0$ . In classification it can then be the case that  $\alpha_{t-1}(i) a_{ij}$  is nonzero for only one value of  $j$ , and  $O_t = k$ , then the probability of that sequence arising from the model with  $b_j(k) = 0$  is  $P = 0$ ; hence a recognition error must occur. This is the so-called missing or inadequate training data problem. We handle this problem (see Ref. 13 for a justification) by using post-estimation constraints on the  $b_j(k)$ 's of the form

$$b_j(k) \geq \epsilon, \quad (10)$$

where  $\epsilon$  is a suitably chosen threshold value. All  $b_j(k)$ 's are compared to the  $\epsilon$  threshold and those that are below  $\epsilon$  are replaced by  $\epsilon_j$  for each  $j$ . After this replacement, each  $b_j(k)$  that was not changed to the  $\epsilon$  value is rescaled by the quantity  $1 - R_j\epsilon$  [where  $R_j$  is the number of  $b_j(k)$ 's changed for a given  $j$ ] to properly normalize the  $b_j(k)$ 's.

It should be clear that all the constrained HMMs are left-right models in that the observations must begin in state 1, must proceed from state to state in a monotonically increasing manner, and must

end in state  $N$ . Thus, temporal information in the observation sequence is coded directly into the left-to-right HMM.

### 3.5 Multiple estimates of $A$ , $B$ , and averaging

As we mentioned earlier, the reestimation and gradient optimization procedures are guaranteed to find a critical point of  $P$  for each HMM. However, in practice, a large number of such points exist in the parameter space. Thus, different initial conditions on  $A$  and  $B$  may lead to different solutions. To understand the variability in model parameters as well as its effects on overall recognition performance, a series of HMMs were obtained for each word by selecting  $R$  random starting sets for  $A$  and  $B$ , and solving for the optimum  $A$  and  $B$  in each case. By scoring each of the  $R$  models individually, we can obtain an indication of the statistical variability in performance score owing to uncertainty in  $A$  and  $B$ .

An alternative procedure to using multiple HMMs for each word, obtained from different random starting values for  $A$  and  $B$ , is to average the  $R$  sets of  $A$  and  $B$  to give an averaged model for each word. The effects of such averaging on word recognition accuracy will be discussed in Section 6.3.

### 3.6 Scoring of observation sequences

One way to score a given observation sequence,  $\mathbf{O}$ , is to use the iterative calculation of eqs. (4) and (5). We call this the Baum-Welch score,  $P_{BW}$ . For left-to-right models, eq. (5) is modified as

$$P_{BW} = \alpha_T(N) \quad (11)$$

because the sequence is constrained to end in state  $N$ .

An alternative scoring procedure for the observation sequence,  $\mathbf{O}$ , given the model,  $\mathbf{M}$ , is the Viterbi algorithm,<sup>20</sup> which may be compactly stated as:

(i) Initialization— $\delta_1(i) = \log[\pi_i b_i(O_1)]$ ,  $i = 1, 2, \dots, N$

(ii) Recursion—for  $2 \leq t \leq T$ ,  $1 \leq j \leq N$

$$\delta_t(j) = \max_{1 \leq i \leq N} \{ \delta_{t-1}(i) + \log[a_{ij} b_j(O_t)] \}$$

(iii) Termination— $P_{VI} = \delta_T(N)$  for left-to-right models

$$= \sum_{j=1}^N \delta_T(j) \text{ for unconstrained models.}$$

The above algorithm is a form of the well-known dynamic programming method and can be shown to have the property of determining the state sequence  $\mathbf{i} = i_1 i_2 \dots i_T$ , which maximizes

$$P(\mathbf{i} | \mathbf{O}, \mathbf{M}).$$

It is easily shown that both the Baum-Welch and Viterbi scoring procedures require roughly the same amount of computation. The major differences are in the interpretation of the resulting solutions.

#### IV. VECTOR QUANTIZATION OF LPC COEFFICIENTS

In Section III we noted that in our implementation of HMMs for isolated word recognition the inputs to the model are assumed to be sequences of discrete symbols chosen from a finite alphabet. We obtained these discrete symbols by using the method of vector quantization<sup>19,21</sup> of the LPC vectors measured as described in Section II. In this section we review the theory of vector quantization and discuss its implementation for isolated word recognition.

##### 4.1 Theory of vector quantization

Assume we have a training set of LPC vectors,  $\mathbf{a}_i$ ,  $i = 1, 2, \dots, I$ , which are a good representation of the types of LPC vectors that occur when the words in the vocabulary are pronounced by a wide range of talkers. The main idea behind vector quantization is to determine the optimum set of codebook LPC vectors,  $\hat{\mathbf{a}}_m$ ,  $m = 1, 2, \dots, M$ , such that for a given  $M$ , the average distortion in replacing each of the training set vectors,  $\mathbf{a}_i$ , by the closest codebook entry,  $\hat{\mathbf{a}}_m$ , is minimum.

More formally stated, if we define  $d(\mathbf{a}_R, \mathbf{a}_T)$  as the distance between two LPC vectors,  $\mathbf{a}_R$  and  $\mathbf{a}_T$ , then the goal of vector quantization is to find the set,  $\hat{\mathbf{a}}_m$ , such that

$$\|D_M\| = \min_{\hat{\mathbf{a}}_m} \left\{ \frac{1}{I} \sum_{i=1}^I \min_{1 \leq m \leq M} [d(\hat{\mathbf{a}}_m, \mathbf{a}_i)] \right\} \quad (12)$$

is satisfied. The quantity  $\|D_M\|$  is the average distortion (distance) of the vector quantizer.

The way in which eq. (12) is solved, for a given value of  $M$ , is due to Juang et al.<sup>21</sup> The algorithm first finds the optimum solution for  $M = 2$  (two codebook entries), then splits each optimum LPC vector into two components, and finds the optimum solution for  $\hat{M} = 2 \cdot M$ . This procedure iterates until  $M$  is as large as desired. A flow diagram of the details of the codebook generation procedure is given in Ref. 21. The local distance used in our system was the likelihood distance,<sup>2</sup>

$$d(\mathbf{a}_R, \mathbf{a}_T) = \frac{\mathbf{a}_R V_T \mathbf{a}'_R}{\mathbf{a}_T V_T \mathbf{a}'_T} - 1, \quad (13)$$

where  $V_T$  is the autocorrelation matrix of the sequence that gave rise to LPC vector  $\mathbf{a}_T$ .

## 4.2 Implementation of the vector quantizer

To train the vector quantizer we used a set of 39708 LPC vectors, obtained by using all the vectors in one complete set of 10 isolated digits uttered by each of 100 talkers (50 male, 50 female). Applying the algorithm of Juang et al.,<sup>21</sup> we generated vector quantizers of size  $M = 2, 4, 8, 16, 32, 64,$  and 128. During the course of running the algorithm, several performance criteria were monitored, including:

- (i) Average distortion,  $\|D_M\|$ , of eq. (12)
- (ii) Sigma ratio (cluster separation) of the resulting codebook entries (clusters), defined as

$$\sigma = \frac{\frac{1}{M} \sum_{i=1}^M \left( \frac{1}{M-1} \right) \sum_{j=1}^M d(\hat{\mathbf{a}}_i, \hat{\mathbf{a}}_j)}{\|D_M\|}, \quad (14)$$

where the numerator is the average intercluster distance, and  $\|D_M\|$  is the average intracluster distance.

(iii) Cluster cardinality,  $N_i$ , defined as the number of tokens in the  $i$ th cluster (i.e., the cluster represented by the  $i$ th codebook entry).

(iv) Cluster distortion,  $\tilde{d}_i$ , defined as the average distortion (distance) for the  $i$ th cluster.

It should be clear that the average distortion,  $\|D_M\|$ , satisfies the relation

$$\|D_M\| = \frac{1}{I} \sum_{i=1}^M \tilde{d}_i \cdot N_i \quad (15)$$

and that the cluster occupancy satisfies the relation

$$\sum_{i=1}^M N_i = I. \quad (16)$$

Results of running the VQ algorithm on the training set of 39708 vectors are given in Figs. 5 through 8. Figure 5 shows plots of  $\|D_M\|$  versus  $M$  (on a log scale) (part a), and the  $\sigma$ -ratio versus  $M$  (part b) for values of  $M$  from 2 to 128. We can see that for values of  $M \geq 32$  the average distortion falls below 0.3, and that for  $M = 64$  the value of  $\|D_M\| \approx 0.2$ . If we use the conventional recognizer of Fig. 1, the average distance between repetitions of a word (after DTW alignment) has been found to be on the order of 0.3 to 0.4;<sup>16</sup> hence, values of  $\|D_M\| < 0.3$  imply smaller error for the VQ than for interreplication variations of words. The  $\sigma$ -ratio plot shows ratios greater than 10 for  $M \geq 32$ ; hence, extremely good cluster separation is achieved in the vector quantizer for these values of  $M$ .

Figures 6 through 8 show a detailed analysis of the statistics of the vector quantizer output for  $M = 128$ . Figure 6a shows the cluster

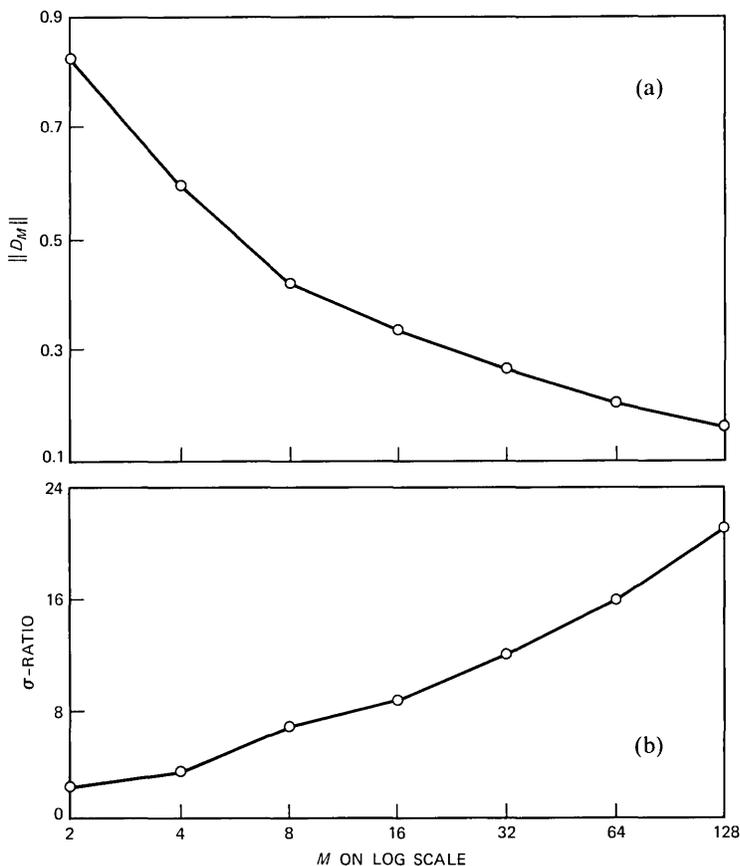


Fig. 5—Plots of vector quantizer performance versus size of codebook for: (a) average distortion, and (b) sigma ratio.

cardinality as a function of the VQ index, and Fig. 6b a histogram of cluster cardinality. The largest cluster has 857 tokens, whereas the smallest cluster has 119 tokens; hence, a spread of over 7 to 1 in cluster occupancy is obtained. The average cluster cardinality, for this case, is 310 tokens, as denoted by the dashed line in Fig. 6a. The histogram of cluster cardinality indicates that the vast majority of clusters have fewer than the average number of tokens.

Figure 7a shows the cluster distortion as a function of the VQ index, and Fig. 7b shows a histogram of cluster distortions. The largest distortion for any cluster is 0.303, whereas the smallest distortion is 0.047; hence, a spread of more than 6 to 1 is observed in cluster distortions. The dashed lines in Fig. 7 denote the average cluster distortion, which in this case is 0.165.

Finally, Fig. 8 shows a plot of the total cluster distortion, defined as

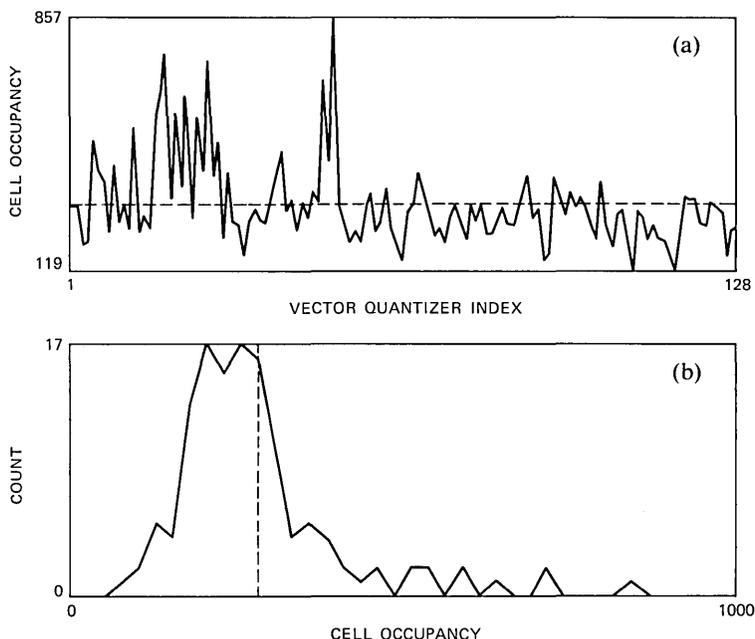


Fig. 6—Plots of (a) cell occupancy versus codebook index, and (b) histogram of cell occupancy for 128-codeword vector quantizer.

$N_i \tilde{d}_i$ , versus VQ index. The range of total cluster distortion is from 25 to 72; hence, a spread of less than 3 to 1 is obtained. It is conjectured that the “ideal” vector quantizer seeks to determine the set of “optimum” codebook vectors such that the total cluster distortion is as close to uniform as possible. Hence, clusters with large average distortions should have low cardinality, whereas clusters with small average distortion should have high cardinality. It can be seen from Figs. 6 through 8 that the total cluster distortion statistics are much closer to uniform than are either the cardinality or the average cluster distortion statistics.

Based on the results shown in Figs. 5 through 8, it was decided to implement the HMM recognizer using a  $M = 64$  VQ, since the small decrease in average distortion from  $M = 64$  to  $M = 128$  did not justify the increased computation owing to the larger codebook.

Figure 9 shows some properties of the LPC vectors in the codebook for  $M = 64$ . Shown in this figure are plots of the first few resonances of the 64 codebook entries (part a), and plots of first versus second resonance (part b), first versus third resonance (part c), and second versus third resonance (part d). As we anticipated, typical vowel resonances for the digits are seen clearly in the plots (e.g., high front

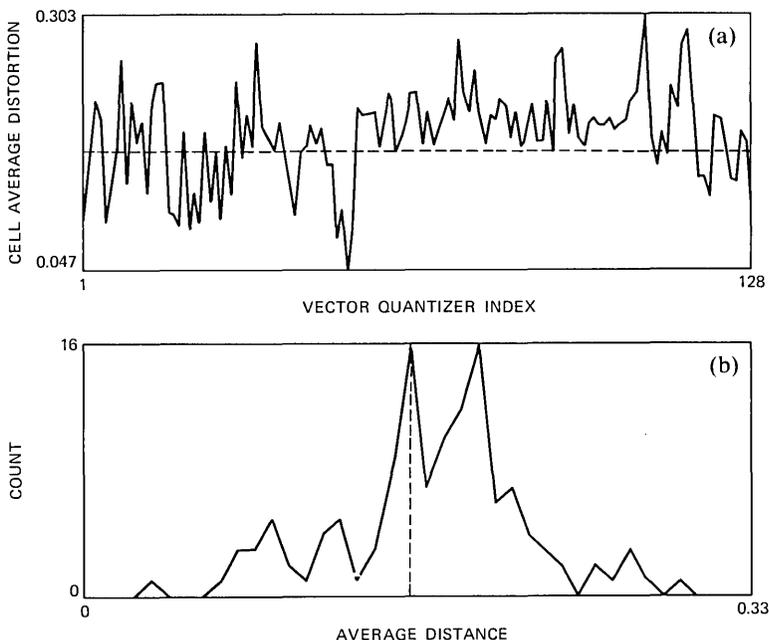


Fig. 7—Plots of (a) cell average distortion versus codebook index, and (b) histogram of cell average distortion for 128-codeword vector quantizer.

vowels, low back vowels, etc), along with characteristic resonances of transient and other nonvoiced sounds. Detailed examination of the spectra of the 64 codebook entries did not provide any further enlightenment as to the VQ properties.

## V. OVERALL HMM/VQ ISOLATED WORD RECOGNIZER

A block diagram of the overall HMM/VQ isolated word recognizer is given in Fig. 10. The recognizer operates as a speaker-independent word recognizer, which runs first in a training mode, to provide the codebook entries of the VQ, and the model coefficients of each word HMM.

In the classification mode the LPC sets of the unknown word are first sent through the vector quantizer (to give a finite set of VQ indices) and then scored on each word HMM (using either the Viterbi scoring or the Baum-Welch scoring) to give a probability score for each word model. The decision rule chooses the word whose model gives the highest probability.

In the next section we describe the results of several tests designed to measure the performance of the HMM/VQ word recognizer and to compare it with that of a conventional LPC/DTW recognizer.

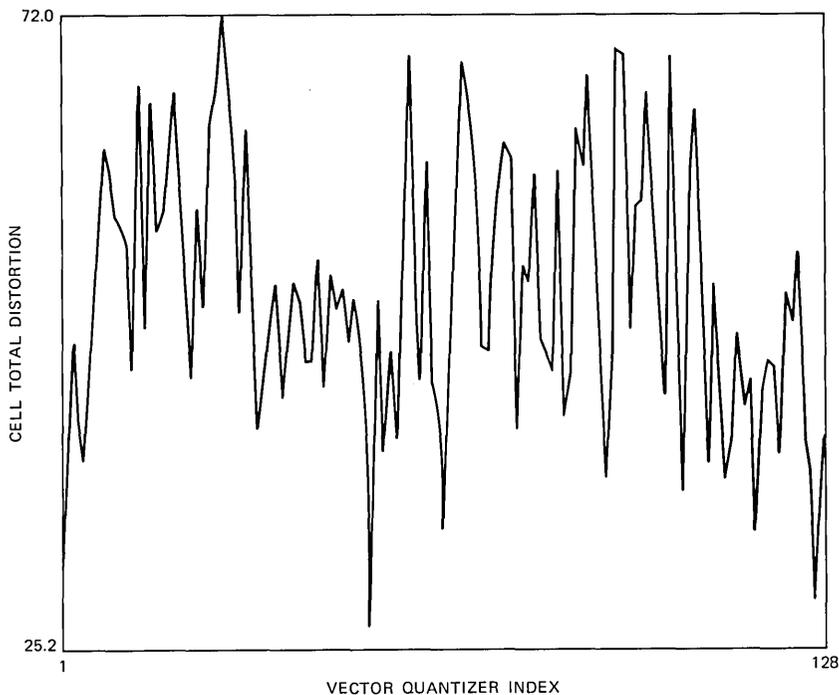


Fig. 8—Plot of cell total distortion versus codebook index for 128-codeword vect quantizer.

## VI. EVALUATION EXPERIMENTS AND RESULTS

Several evaluation tests were performed on the HMM/VQ and LPC/DTW isolated word recognizers. For most conditions a single test set of data, denoted as TS1, was used, consisting of one replication of each of the 10 digits by a set of 100 talkers. These talkers were the same ones used to train the recognizer; however, the test replication was recorded many days after the training replication. A second test set of data, denoted as TS2, was used in a couple of tests. This test set consisted of 20 replications of each of the 10 digits by a set of 10 new talkers (5 male, 5 female). Thus, TS2 contained twice as many test tokens as TS1, but represented only one-tenth the number of talkers; however, none of the talkers was included in the training set for either the VQ or the HMMs.

The results presented in this section are the output of a series of recognition tests in which one or more features of the HMM/VQ recognizer were varied. Following the presentation of the results of each of the individual experiments, we shall endeavor to provide a measure of coherency to the results.

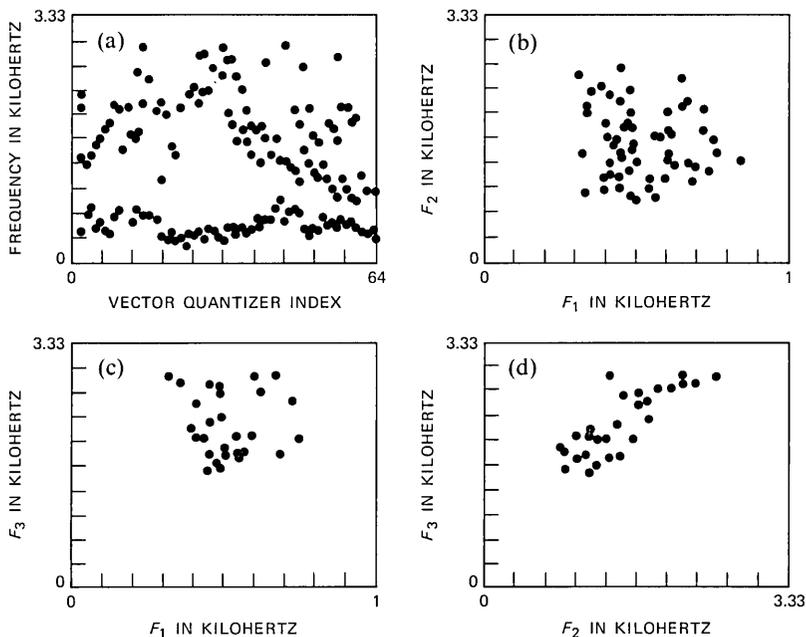


Fig. 9—Plots of locations of first three resonances of codebook vectors: (a) plotted as a function of the vector index, (b) plotted in the  $F_1 - F_2$  plane, (c) plotted in the  $F_1 - F_3$  plane, and (d) plotted in the  $F_2 - F_3$  plane.

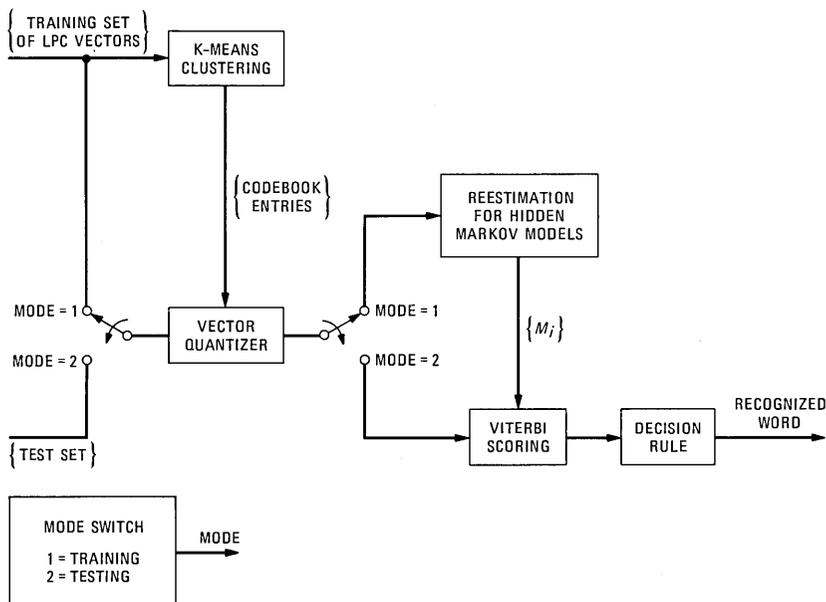


Fig. 10—Overall block diagram of the hidden Markov model—vector-quantizer isolated word recognizer.

### 6.1 Effects of constrained and unconstrained HMMs

The first set of experiments sought to understand the effects of placing constraints on the  $A$  and  $B$  matrices on the performance of the overall recognizer. As such, the HMM was trained for an  $N = 5$  state model with the following constraints:

- UC model: No constraints placed on  $A$ ; epsilon constraints [of the type given in eq. (10)] placed on  $B$ .
- CO model: The constraints of eq. (9a) placed on  $A$ ; epsilon constraints of eq. (10) placed on  $B$ .
- UC.35 model: Same as UC model but all training sequences with VQ distortions greater than 0.35 were eliminated.
- CO.35 model: Same as CO model but all training sequences with VQ distortions greater than 0.35 were eliminated.

For each of the above four models, the 1000-digit sequences of TS1 were used to measure the overall error rate as a function of the  $\epsilon$  constant parameter. The results are given in Fig. 11, which shows plots of error rate versus  $\epsilon$  (on a log scale) for each of the four models. Several trends clearly emerge from these results. First, we can see that a nonzero value of  $\epsilon$  is an absolute *necessity* for obtaining good performance. Whenever a symbol,  $k$  (a VQ index), appears in a test word in a state,  $j$ , where  $b_j(k) = 0$ , the probability for that word model is multiplied by the  $\epsilon$  value. If  $\epsilon = 0$  then the word model is eliminated from consideration and an error occurs. For finite, nonzero values of  $\epsilon$ , however, such errors need not, and generally will not, occur. Hence,

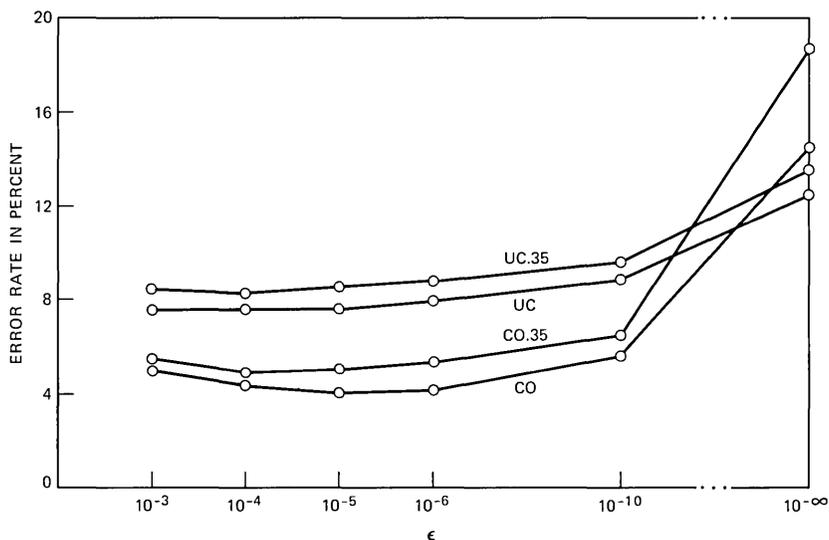


Fig. 11—Plots of average word error rate versus the minimum value of the symbol probability matrix,  $\epsilon$ , for four types of hidden Markov models, for TS1 data.

even for  $\epsilon$  on the order of  $10^{-10}$ , the error rate is substantially smaller than for  $\epsilon = 0$ .

A second clear trend that can be seen in Fig. 11 is that the constrained serial models performed consistently better than the unconstrained models for all nonzero values of  $\epsilon$ . This result implies that the extra freedom of the unconstrained models tends to raise the probability scores for incorrect words more than for correct words. This is somewhat reminiscent of the fact that opening up the search region of a conventional DTW search helps the wrong words much more than it does the correct words.<sup>16</sup>

It can also be seen from Fig. 11 that it is *always* preferable to train with all available sequences. This result suggests that the more training data given to the HMM model estimation algorithm the better the estimates of the HMM parameters, even if some of the data are less than ideal.

The final trend that emerges from the curves of Fig. 11 is that there is a large range of values of  $\epsilon$  for which essentially identical performance results. For example, in the range  $10^{-10} \leq \epsilon \leq 10^{-3}$ , for model CO, the recognition error rate changes by less than 1.6 percent. Thus, so long as  $\epsilon$  is in this broad range, the exact value of  $\epsilon$  is not overly significant.

Based on the results of this first series of experiments, we applied the following restrictions:

- (i) Consider only constrained HMMs
- (ii) Constrain  $B$  matrix entries such that  $b_j(k) \geq \epsilon = 10^{-5}$
- (iii) Use all possible training sequences for the HMMs.

Before we proceed to the next series of experiments, some comments should be made about practical methods of implementing constraints on the  $B$  matrix. In Ref. 13 we show how the constraints on the  $b_j(k)$  coefficients, of the type given in eq. (10), can be incorporated directly into either the gradient or the Baum-Welch reestimation algorithm. We have also tried a post-normalization technique in which no constraints were placed directly on the  $B$  matrix. Following convergence, the  $B$  matrix was examined and all entries whose values were below  $\epsilon$  were reset to the value  $\epsilon$ , and the rows of the matrix were suitably renormalized to sum to 1.0. Our recognition results indicate identical performance for both the direct and the post-normalization constraint methods. Hence, there appears to be no advantage to constraining the  $B$  matrix directly.

## 6.2 Markov model with variable number of states

The second set of experiments consider the effects on recognition accuracy of using the constrained serial HMMs with different numbers of states. In particular we computed the optimum SC1 model (see Fig.

4) for each digit where the number of states varied from 2 to 9. We also computed the optimum SC1 model for a 20-state model for each digit.

To evaluate these different models a recognition test was conducted in which each digit was represented by a single  $N$ -state HMM, where  $N$  took on the values 2 to 9, and 20. The results of this experiment are given in Fig. 12a, which shows word error rate versus number of states in the HMM for the data of TS1. We see a steady but slow decrease in the average word error rate in this curve. We also see a statistical fluctuation in the curve owing to the sampling variability in the  $A$ 's and  $B$ 's for each word HMM. (We will return to this issue later in this section.)

A second recognition test was conducted in which each word was represented by *all* the word models with number of states up to some maximum value,  $NMAX$ . The results of this recognition test (again using TS1 data) are shown in Fig. 12b. A somewhat smoother curve of

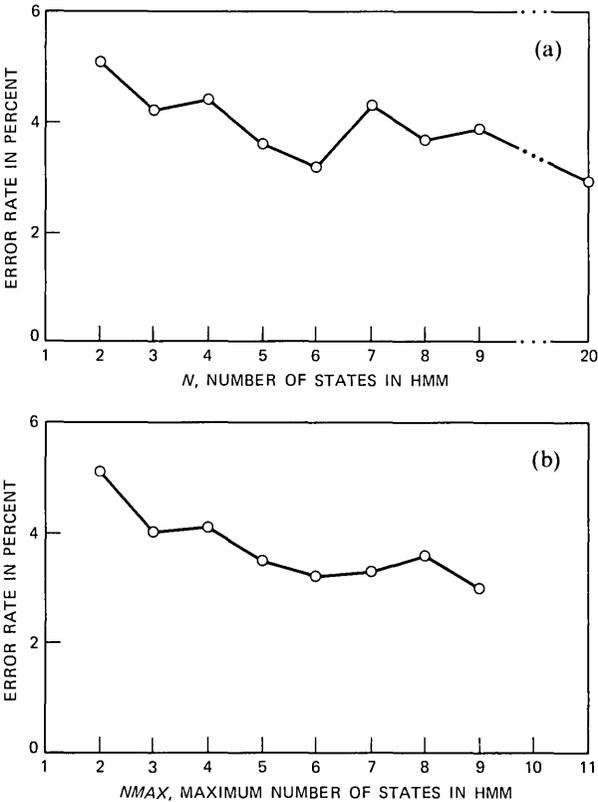


Fig. 12—Plots of (a) error rate versus the number of states in the HMM, and (b) error rate versus the maximum number of states in the HMMs for TS1 data.

errors versus  $NMAX$  is obtained in Fig. 12b than for the individual models of Fig. 12a; however, the general behavior of both curves is similar.

Figure 13 shows a breakdown of the error rates for each digit for the experiment in which a single HMM, with  $N$  states, was used for each digit. There is a highly complex interaction between error rate and model size for all digits. Thus it cannot be argued, for instance, that digits like zero and seven (two syllables) need more states in their models than digits like two or one (monosyllables).

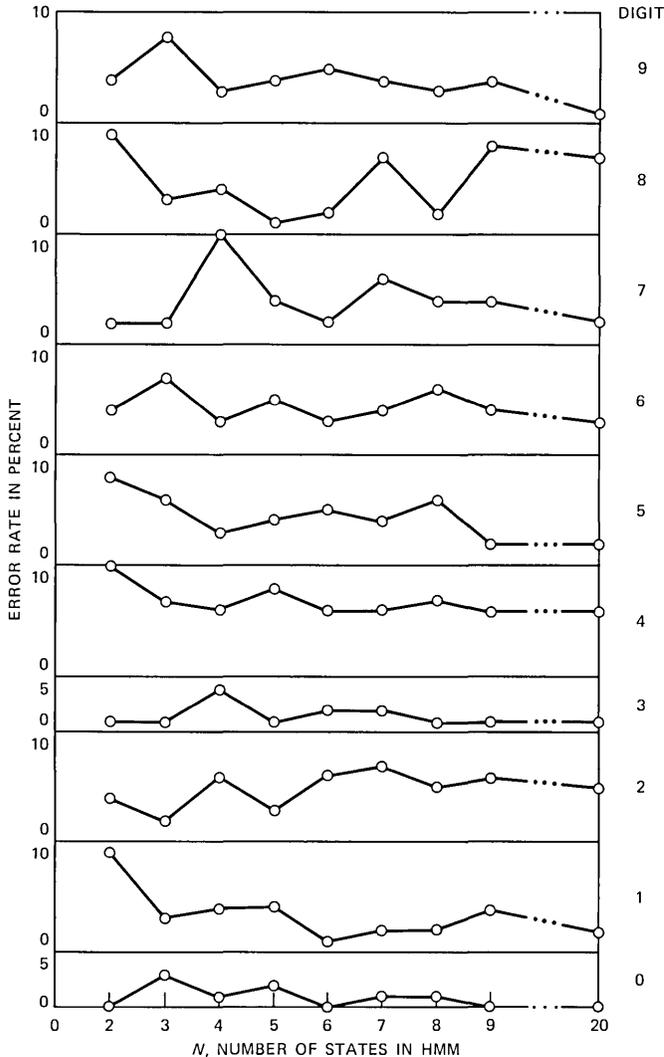


Fig. 13—Individual plots of digit error rates versus the number of states in the HMMs.

From the results shown in Figs. 12 and 13, it was concluded that there is very little gain in using HMMs with more than five or six states when the SC1 structure for each model is being used. It was also concluded that no simple relationship existed between word (digit) accuracy, number of sounds (syllables, etc) in the word, and number of states needed in the word HMM.

### 6.3 Effects of random starting points

The third set of experiments was concerned with the statistical variability in the performance scores resulting from statistical variability in the parameters of the word HMMs because of different random initial estimates of the parameters. To quantify this effect, a 5-state SC1 model was generated for each digit using 10 different random starting sets of model parameters. Thus, for each digit, 10 "equivalent" HMMs were created.

A recognition test was then run, using TS1 data, in which each of the 10 models was tested separately. Also tested was the case in which all 10 models were used for each digit, as well as the case in which a single model was used for each digit, where the model parameters were obtained by averaging the parameter estimates for each of the 10 word models. The results of these recognition tests are given in Fig. 14, which shows word error rate versus the random start number. Also shown, as single isolated values, are the error scores for the average and the combined 10 state runs. The dashed line in Fig. 14 is the average error rate of the 10 individual models.

From the data of Fig. 14, we can see that the 10 individual models all performed identically to within  $\pm 1$  percent; hence, the expected statistical variability in error rate scores, due to random starts, should

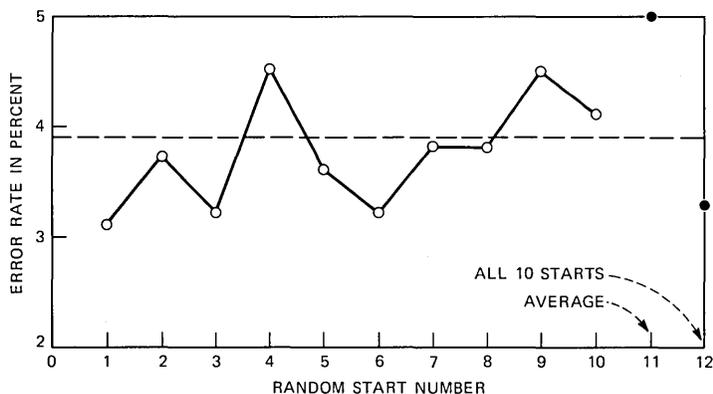


Fig. 14—Plots of error rate for 10 different random starting sets of values of the HMM parameters. Also shown are individual error rates for an averaged model and for combining 10 random start models.

be on the order of  $\pm 1$  percent. We can also see that the performance of the “averaged” model (obtained by averaging HMM parameters for each word) was somewhat poorer than that of any of the 10 individual models. The fact that using all 10 word models for each digit gives an error rate comparable to that of the *best* word models indicates that multiple word models provide a small gain that comes at the cost of greatly increased computation.

Overall, the data of Fig. 14 suggest that a single model per word should be adequate for most purposes, and that the effects of different random starts on the overall error performance are small.

#### **6.4 Parallel constrained HMMs**

The last factor investigated was the effect of using constrained parallel HMMs for each digit. The main idea was that a true parallel structure could model the effects of using a multiplicity of word models in much the same way as multiple templates are used in the conventional LPC/DTW word recognizer.

Figure 15a shows the 5-state constrained serial HMM that was used in previous experiments along with a 7-state, constrained parallel HMM (Fig. 15b), and an 8-state, constrained parallel model (Fig. 15c). The 7-state parallel structure was intended to represent four distinct 5-state word models, in that there were four sets of paths through the model, namely, 1-2-5-7, 1-2-6-7, 1-3-5-7, and 1-3-6-7. The 8-state parallel structure was intended to represent eight distinct 5-state word models in that there were eight sets of paths through this model, namely, 1-2-4-6-8, 1-2-4-7-8, 1-2-5-6-8, 1-2-5-7-8, 1-3-4-6-8, 1-3-4-7-8, 1-3-5-6-8, and 1-3-5-7-8.

Each of the three HMM structures of Fig. 15 was used to generate a word model for each of the digits. The three sets of models were then tested using TS1 data. The results showed each of the three systems obtained the same word error rate (3.5 percent) to within  $\pm 0.1$  percent. These results indicated that there was really no advantage in using the parallel structure.

#### **6.5 Comparison with LPC/DTW recognizers**

To provide some basis of comparison for the performance of the HMM/VQ recognizer with that of more conventional word recognizers, the data of TS1 was tested on the LPC/DTW recognizer of Fig. 1. The reference set consisted of 12 templates per digit, generated from a clustering analysis of the 100 tokens of each digit in the training set (the same training set used to train each word HMM). The decision rule was the nearest neighbor rule ( $KNN = 1$ ).<sup>17</sup>

Tables I and II show average word recognition accuracies for the following three recognizers:

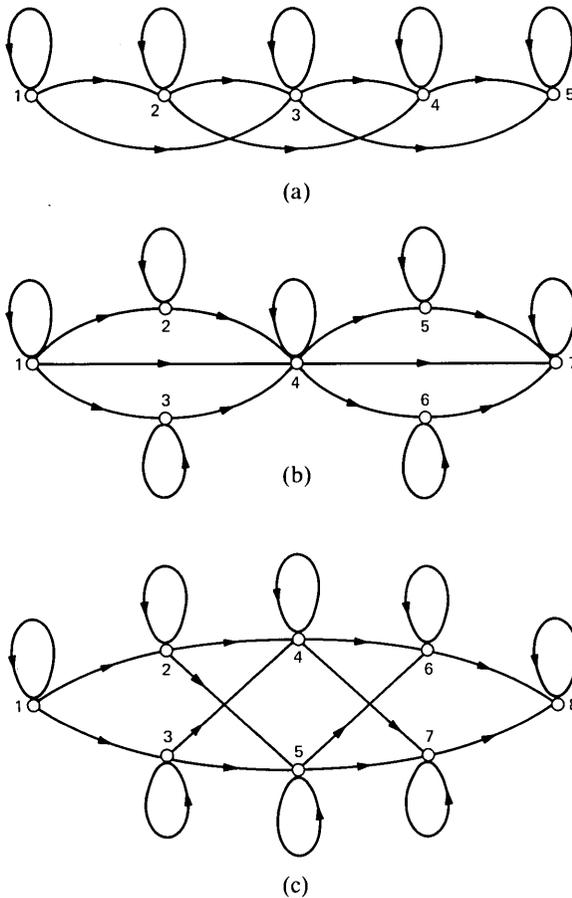


Fig. 15—State diagrams for (a) simple constrained serial 5-state model, (b) constrained parallel 7-state model, and (c) constrained parallel 8-state model.

(i) HMM/VQ using a constrained serial structure with five states for the HMM and the 64-element VQ.

(ii) LPC/DTW, the conventional recognizer.

(iii) LPC/DTW/VQ, the conventional recognizer with both reference and test patterns quantized using the same VQ used in the HMM case.

The results shown in Table I are for the 1000 digits of TS1, whereas those shown in Table II are for the 2000 digits (10 talkers) of TS2. The results given in Table I show that for TS1, both the LPC/DTW and HMM recognizers, when using the VQ, achieved essentially the same digit accuracy; however, the LPC/DTW system without the VQ achieved a 2-percent higher word accuracy. For TS2 in Table II the

Table I—Comparison of results on HMM/VQ and LPC/DTW word recognizers for [average word accuracy (%)] TS1, 100 talkers, 10 digits per talker

Digit	Recognizer		
	HMM/VQ	LPC/DTW	LPC/DTW/VQ
0	98	99	99
1	98	98	99
2	96	100	96
3	99	99	97
4	93	97	96
5	97	96	93
6	96	100	94
7	99	100	94
8	92	98	96
9	95	98	96
Average	96.3	98.5	96.5

Table II—Comparison of results on HMM/VQ and LPC/DTW word recognizers for TS2, 10 talkers, 200 digits per talker

Talker	Recognizer		
	HMM/VQ	LPC/DTW	LPC/DTW/VQ
1	74.5	96	87
2	99.5	100	99.5
3	94	99	97
4	89	99	91
5	95.5	100	100
6	95.5	99	99.5
7	100	100	99.5
8	91.5	96	93
9	91.5	99.5	93.5
10	96.5	98.5	98
Average	92.8	98.7	95.5

results show that the VQ led to a 3.2-percent reduction in accuracy for the LPC/DTW system, and an additional 2.7-percent loss in accuracy for the HMM system. A good deal of the loss in accuracy, however, was contributed by talker 1, whose accuracy was 87 percent for the LPC/DTW/VQ recognizer, and 74.5 percent for the HMM/VQ recognizer. With only 10 talkers, the influence of a single talker on the overall accuracy may be substantial.

An analysis of the actual errors of all three of the recognizers of Tables I and II shows the following:

(i) Of the 37 tokens misclassified by recognizer HMM/VQ in TS1, 31 were correctly identified by the LPC/DTW recognizer, and 27 were correctly identified by the LPC/DTW/VQ recognizer.

(ii) The vast majority (25) of the 37 errors made by recognizer HMM/VQ were cases in which the probability of the correct word was

lower than the probability of the incorrect word by a factor of  $e^6$  or larger.

The results show that when the LPC/DTW recognizer (either with or without the VQ) has incorrectly identified the word, most of the time the HMM recognizer has correctly identified the word. Hence, a side result of this work is that the HMM/VQ model can be combined with an LPC/DTW model to provide word accuracies greater than either individual recognizer could obtain. In particular, for the data of TS1, the combined recognizer could have achieved a 99.4-percent word accuracy by using appropriate decision logic<sup>22</sup> on all cases in which both recognizers did not agree.

The same sort of trends are noted in the errors of TS2. Of the 124 errors made by the HMM/VQ recognizer, 113 were correct in the LPC/DTW or equivalently of the 26 errors made by the LPC/DTW recognizer, 15 were correct in the HMM/VQ system. Hence, again a combined system could potentially achieve an accuracy of about 99.5 percent on TS2 data. Considering that even the talkers in the TS2 data were different from those in the training set, this accuracy appears to be quite remarkable.

In summary, comparisons between the HMM/VQ and LPC/DTW recognizers indicate that without the VQ, the LPC/DTW recognizer achieves from 2- to 6-percent higher accuracy than the HMM/VQ system; with the VQ the differential in accuracy is from 0 to 3 percent.

## VII. DISCUSSION

In this paper we showed that the techniques of vector quantization of LPC vectors and hidden Markov modeling can be combined in a simple, straightforward manner to implement a speaker-independent, isolated word recognizer. With adequate training of the vector quantizer and the Markov model estimation algorithm, a digits vocabulary can be recognized, with accuracies of from 93 to 96 percent across a wide variety of talkers. Direct comparisons with a conventional linear predictive coding recognizer using dynamic time warping for time alignment with multiple templates for each vocabulary word showed that the HMM/VQ recognizer performs only a little worse (0.2 percent in one test, 2.7 percent in another) than the LPC/DTW recognizer when using the VQ. Without the VQ, the LPC/DTW recognizer was about 2 to 3 percent better than when the VQ was used.

Several general conclusions can be drawn from the results. The first is that the HMM/VQ recognizer performed exceedingly well on the difficult task of speaker-independent recognition of isolated digits. The fact that the overall performance of the HMM/VQ recognizer was somewhat poorer than the LPC/DTW/VQ recognizer appears to be primarily because of the insufficiency of the HMM training data.

Although 100 training sequences per word are adequate for a clustering analysis, as used in the conventional recognizer, they appear to be inadequate for obtaining good HMM models for these words. This suggestion is made plausible by considering what is being estimated in the HMM for each word. For an  $N$ -state Markov model, with  $M$  finite symbols per state, a total of  $N^2 + NM$  parameters must be estimated. (Of course with constraints there are somewhat fewer parameters.) For  $N = 5$ ,  $M = 64$ , we need 345 parameters to be estimated from about  $100 \times 40$  frames of VQ indices. The “curse of dimensionality” would imply that this amount of training data is woefully inadequate. [We have seen one consequence of this inadequacy in that we had to use the  $\epsilon$ -constraints of eq. (10) on any  $b_j(k)$  whose value fell below the  $\epsilon$  threshold.] In view of this, the fact that we achieve the results we are getting is rather remarkable.

A second conclusion that can be drawn from the results is that the use of the VQ on the LPC sets leads to a small, but not insignificant, degradation in performance of both the HMM/VQ and LPC/DTW/VQ recognizers as compared to the conventional LPC/DTW system. This suggests the need for using more than 64 vectors in the codebook or resorting to continuous models of the LPC parameters.

The results have shown that the errors made by the HMM/VQ and LPC/DTW recognizers are largely disjoint. Here there exists the potential of using some fairly standard techniques to combine the two recognizers into one whose accuracy is as good as the best of both recognizers on any given word.<sup>22</sup> This topic merits further consideration.

The experimentation with various forms of the Markov models used in the recognizer showed fairly conclusively that:

(i) Constrained models (with constrained transition matrices) performed consistently better than unconstrained models.

(ii) A finite minimum constraint on the state symbol probability matrix was a necessity for good system performance.

(iii) The effects of different random starting values for the HMM parameters were negligible in evaluating overall recognizer performance.

(iv) The required number of states in each word HMM needed to be on the order of 5. More states did not lead to significant improvements in performance.

(v) Parallel HMM structures yielded no real improvements over cascade structures, thereby indicating that an equivalent of multiple HMMs is not readily obtainable by simply changing the model structure.

(vi) The Viterbi scoring and the Baum-Welch scoring of test sequences give essentially identical performance.

### 7.1 Computational considerations of the HMM/VQ recognizer

It is worthwhile estimating the storage required and the computation needed to process an unknown test utterance using the HMM/VQ recognizer, and to compare them to the requirements of the conventional LPC/DTW recognizer. Our intention is to provide only a rough estimate of computational expense. A number of straightforward reductions in computation can be achieved for each recognizer through the judicious use of table storage. Also we ignore overhead owing to index computation, etc.

The first extra step in the HMM/VQ recognizer (after conventional LPC analysis) is vector quantization of the unknown test pattern. If we assume there are  $T$  frames in the test word, and  $M$  codebook entries in the VQ, then we need a total of

$$C_1 = M \cdot T(p + 1) \quad (17)$$

multiplications\* (where  $p$  = LPC order) to perform the  $M \cdot T$  dot products required to get the best codebook entry for each frame.

Evaluation of the word HMM score, using the Viterbi scoring method with the constrained  $A$  matrix, requires approximately

$$C_2 = T \cdot N \cdot 3 \quad (18)$$

multiplications and logarithms per word model, where  $N$  is the number of states in the model, and the factor 3 accounts for the number of valid transitions into a given state. For a vocabulary of  $V$  words a total of

$$C_* = M \cdot T \cdot (p + 1) + V \cdot T \cdot N \cdot 3 \quad (19a)$$

$$C_{\log} = V \cdot T \cdot N \cdot 3 \quad (19b)$$

multiplications and logarithms are required. For  $M = 64$ ,  $T = 40$ ,  $V = 10$ ,  $N = 5$ , and  $p = 8$ , eq. (19) gives  $C_* = 29040$  multiplies and  $C_{\log} = 6000$  logarithms.

For a conventional LPC/DTW recognizer with  $Q$  templates per vocabulary word, the computation for DTW processing is

$$C_3 \cong Q \cdot V \cdot T^2 / 3 \cdot (p + 1), \quad (20)$$

which for  $Q = 12$  and other parameters the same as above gives  $C_3 = 576000$  multiplications. Hence, the HMM/VQ recognizer requires about 17 times less computation (assuming logarithms are equivalent to multiplications) than the LPC/DTW recognizer.

With regard to storage, the HMM/VQ recognizer requires

---

\* In this simplified analysis we neglect additions and comparisons of data and use multiplication count as the measure of computation.

$$S_1 = M(p + 1) \quad (20a)$$

$$S_2 = (M \cdot N + 3N) \cdot V, \quad (20b)$$

where  $S_1$  is the storage (in words\*) for the VQ codebook entries, and  $S_2$  is the storage (in words) for the set of  $V$  word HMMs. For the given values of the parameters, the total storage is  $S = S_1 + S_2 = 4106$  words.

For the conventional LPC/DTW recognizer the storage is

$$S_3 = Q \cdot V \cdot T \cdot (p + 1), \quad (20c)$$

which gives 43,200 words for the assumed parameter values. Again we see a 10 to 1 reduction for the HMM/VQ model over the LPC/DTW model.

It should be noted that in our analysis of computation we have not included the computation of LPC coefficients. This computation must be performed in “real-time” and is independent of the vocabulary size,  $V$ . Hence, for a sufficiently large vocabulary the computation for scoring each word dominates the overall computation, and the rough analysis given above is appropriate. Furthermore, the computation for coding each LPC vector into the nearest codebook entry [eq. (17)] is also independent of vocabulary size and often could be performed in the “real-time” part of the recognizer. For such implementations the gain in computation of the HMM/VQ recognizer, over the conventional LPC/DTW recognizer, is even higher than our simple analysis predicts. Finally, it is straightforward to show that if we compare the computation of the HMM/VQ recognizer with that of the LPC/DTW/VQ recognizer, assuming that the VQ is done in real time and that tabular computation of products and logarithms is used, then by comparing the number of additions, the computational advantage of the HMM/VQ system over the LPC/DTW/VQ system still holds.

### ***7.2 Some comments on the relationships between DTW and HMMs***

Contemporary research on speech recognition has produced two algorithmic procedures for dealing with the nonstationarity of the speech signal: temporal alignment techniques, and Markov modeling. These methods display certain superficial similarities (e.g., both use dynamic programming methods, can be cast in a Bayesian framework, and have a state transition network associated with them), as a result of which it has occasionally been claimed that they are identical. To the best of our knowledge, the experiments reported here represent the first direct comparison of the two methods. From these experiments it is abundantly clear that the methods are not identical. While

---

\* Words of storage refer to unquantized floating point data.

their overall performances are comparable, they appear to make different errors and involve different amounts of computation and different complexities of training.

In all of these respects the two methods reflect the dichotomy between parametric and nonparametric methods of pattern recognition in the sense of Patrick.<sup>23</sup> In Markov modeling we assume that there is a family of models of a particular structure differing only in the values of their parameters. We use a large training set to estimate these parameters and assume that, if correctly done, the parameters will capture the structure of the data. The training procedure is computationally expensive but need be done only once. After training is complete, relatively little computation is required to determine whether an unknown observation was generated by the model.

Temporal alignment methods are opposite in the following ways. We assume that there is an underlying structure to the training data but its form is unknown. We attempt to capture that structure by simply storing one or more samples and measuring their "distance" to an unknown sample with a metric that is sensitive to the distinctive features of the categories that we seek to identify. The metric is monotonically related to the class conditional density functions so that minimum distance corresponds to maximum model probability. In this case training is a computationally simple data collection and storage process. Probability computation, on the other hand, is very costly since we must measure the distance to every prototype in the training set.

All of these characteristics are made manifest by our experiments. What remains to be determined is whether the parity of these methods extends to more difficult problems of speech recognition. We hope to answer that question by further experimentation.

### VIII. SUMMARY

We have described the results of an extensive investigation into the applicability of the techniques of vector quantization and hidden Markov modeling to speaker-independent, isolated word recognition. We have shown that, when properly designed, the resulting recognition system produces highly accurate word recognition on a vocabulary of isolated digits. We have also discussed the effects of variations of model parameters on system performance. Our experiments show that the resulting recognizer requires about 10 times less storage, and about 17 times less computation for classifying a test utterance than does an equivalent recognizer using LPC coding and dynamic time warping. These economies are obtained at the expense of only a slight increase in error rate.

## REFERENCES

1. T. B. Martin, "Practical Applications of Voice Input to Machines," *Proc. IEEE*, 64 (April 1976), pp. 487-501.
2. F. Itakura, "Minimum Prediction Residual Principle Applied to Speech Recognition," *IEEE Trans. on Acoustics, Speech, and Signal Processing, ASSP-23*, No. 1 (February 1975), pp. 67-72.
3. L. R. Rabiner and S. E. Levinson, "Isolated Connected Word Recognition—Theory and Selected Applications," *IEEE Trans. on Communications, COM-29*, No. 5 (May 1981), pp. 621-59.
4. N. R. Dixon and T. B. Martin, Eds., *Automatic Speech and Speaker Recognition*, New York: IEEE Press, 1979.
5. W. Lea, Ed., *Trends in Speech Recognition*, Englewood Cliffs, NJ: Prentice Hall, 1980.
6. D. R. Reddy, Ed., *Speech Recognition*, New York: Academic Press, 1974.
7. A. Newell et al., "Speech Understanding Systems," Final Report of a Study Group, Carnegie Mellon Report, May 1971.
8. M. R. Sambur and L. R. Rabiner, "A Speaker-Independent Digit Recognition System," *B.S.T.J.*, 54, No. 1 (January 1975), pp. 81-102.
9. T. B. Martin, "Acoustic Recognition of a Limited Vocabulary in Continuous Speech," Ph.D. Dissertation, Univ. of Penn., 1970.
10. A. B. Poritz, "Linear Predictive Hidden Markov Models and the Speech Signal," *Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*, 82, Paris, France (May 1982), pp. 1291-4.
11. J. K. Baker, "The Dragon System—An Overview," *IEEE Trans. on Acoustics, Speech, and Signal Processing, ASSP-23*, No. 1 (February 1975), pp. 24-9.
12. F. Jelinek, L. R. Bahl, and R. L. Mercer, "Design of a Linguistic Decoder for the Recognition of Continuous Speech," *IEEE Trans. on Information Theory, IT-21* (May 1975), pp. 250-6.
13. S. E. Levinson, L. R. Rabiner, and M. M. Sondhi, "An Introduction to the Application of the Theory of Probabilistic Functions of a Markov Process to Automatic Speech Recognition," *B.S.T.J.*, this issue.
14. L. F. Lamel, L. R. Rabiner, A. E. Rosenberg, and J. G. Wilpon, "An Improved Endpoint Detector for Isolated Word Recognition," *IEEE Trans. on Acoustics, Speech, and Signal Processing, ASSP-29*, No. 4 (August 1981), pp. 777-85.
15. L. R. Rabiner and J. G. Wilpon, "A Simplified Robust Training Procedure for Speaker Trained, Isolated Word Recognition Systems," *J. Acoust. Soc. Amer.*, 68, No. 5 (November 1980), pp. 1271-6.
16. S. E. Levinson, L. R. Rabiner, A. E. Rosenberg, and J. G. Wilpon, "Interactive Clustering Techniques for Selecting Speaker Independent Reference Templates for Isolated Word Recognition," *IEEE Trans. on Acoustics, Speech, and Signal Processing, ASSP-27*, No. 2 (April 1979), pp. 134-41.
17. L. R. Rabiner, S. E. Levinson, A. E. Rosenberg, and J. G. Wilpon, "Speaker Independent Recognition of Isolated Words Using Clustering Techniques," *IEEE Trans. on Acoustics, Speech, and Signal Processing, ASSP-27*, No. 4 (August 1979), pp. 336-49.
18. L. E. Baum, T. Petrie, G. Soules, and N. Weiss, "A Maximization Technique Occurring in the Statistical Analysis of Probabilistic Functions of Markov Chains," *Annals of Mathematical Statistics*, 41, No. 1 (1970), pp. 164-71.
19. A. Buzo, A. H. Gray, Jr., R. M. Gray, and J. D. Markel, "Speech Coding Based Upon Vector Quantization," *IEEE Trans. on Acoustics, Speech and Signal Processing, ASSP-28*, No. 5 (October 1980), pp. 562-74.
20. A. J. Viterbi, "Error Bounds for Convolutional Codes and an Asymptotically Optimum Decoding Algorithm," *IEEE Trans. on Information Theory, IT-13* (April 1967), pp. 260-9.
21. B. H. Juang, D. Y. Wong, and A. H. Gray, Jr., "Distortion Performance of Vector Quantization for LPC Voice Coding," *IEEE Trans. on Acoustics, Speech, and Signal Processing, ASSP-30*, No. 2 (April 1982), pp. 294-304.
22. M. K. Brown and L. R. Rabiner, "On the Use of Energy in LPC Based Recognition of Isolated Words," *B.S.T.J.*, 61, No. 10, Part 1 (December 1982), pp. 2971-87.
23. E. A. Patrick, *Fundamentals of Pattern Recognition*, Englewood Cliffs, NJ: Prentice Hall Inc., 1972.



## A Lithographic Mask System for MOS Fine-Line Process Development

By J. M. ANDREWS

(Manuscript received September 23, 1982)

*A mask set, incorporating a group of seven test chips, has been designed for fine-line process development and process control. Although six lithographic levels are available, the masks are generally intended to be used only in subsets of two or three levels to minimize the delay encountered in obtaining electrical test results for whichever processes require investigation. The mask levels serve a variety of purposes for special process development experiments. Available structures include: metal-oxide-semiconductor capacitors, p-n junctions, guarded and unguarded Schottky barrier diodes, ohmic contacts, van der Pauw patterns, insulated gate field-effect transistors, gated diodes, resistors for sheet resistance and linewidth variations, and tapped electromigration test strings. It is not anticipated that a process engineer should ever need more than a maximum of four levels to achieve an appropriate experimental structure for process development. It is not the purpose of these masks to establish fine-line design rules. The masks are intended to be used primarily with standard photolithographic processing, and most device structures have been designed to tolerate up to 5  $\mu\text{m}$  in misalignment errors. However, certain selected features have been coded in a diminishing sequence to a minimum of 1.0  $\mu\text{m}$  for special fine-line investigations. A salient feature of this mask system is the option to interleave rapid turnaround photolithographic steps with fine-line X-ray patterning; therefore, some mask levels have been reissued for X-ray lithography.*

### I. INTRODUCTION

In the past, the development of new silicon integrated circuit processes was impeded by the fact that an adequate set of simple test structures usually could not be fabricated without resorting to the full set of six lithographic levels required by the Poon Tester Chips.<sup>1</sup> This set sometimes requires several months to fabricate if X-ray lithography

is used. If device wafers could not be sacrificed, the processing engineer had to resort to simulating device structures, either by using metal dots of fixed areas on unpatterned oxides or deposited films, or by other schemes such as the use of offset circular windows using a pair of photolithographic steps.<sup>2,3</sup>

A set of photolithographic masks has been designed and is now available to fill the process development gap. The goal has been to provide the processing engineer with the means to simulate critical processing steps by introducing a monitor wafer, prepared in advance by one photolithographic step, and usually requiring only one more lithographic step of any type to obtain a structure ready for electrical testing.

The full set of fine-line process development masks consists of six photolithographic levels, but these have been designed to be utilized in subsets of 2, 3, or 4 levels only. Available structures include metal-oxide-semiconductor (MOS)\* capacitors, contact windows, guarded and unguarded Schottky diodes, van der Pauw patterns, insulated gate field-effect transistors (IGFETs), gated diodes, and tapped electromigration test strings. Also available are large areas accessible for direct probing, for evaporation of MOS dots, or for Auger, scanning electron microscope (SEM), and transmission electron microscope (TEM) studies.

Section II contains a complete description of each of the devices that are available in the MOS mask set. The organization of the devices among each of seven test chips, the chip designations, the device assignments, wafer layout, design rules, and the alignment features are discussed in Section III. Section IV consists of a detailed description of each mask level separately, including the primary purpose for which the level is intended and the features available. Section IV also includes specific recommendations regarding which levels can be omitted with respect to the particular devices required or the experimental intent. Specific applications to silicon integrated circuit processing are discussed in Section V, and the MOS mask system is summarized in Section VI.

## II. DEVICE DESCRIPTION

Most of the test structures in the lithographic mask system are MOS capacitors, because such two-terminal devices are easily fabricated. Normally, only two photolithographic operations are needed, and the first could usually be done on a large number of wafers before specific experiments are planned. Furthermore, the same patterns can be used

---

\* Acronyms and abbreviations are defined fully in the Glossary at the end of this paper. Tables and figures are located at the end of the text.

to fabricate p-n junctions, Schottky diodes, or ohmic contacts to the substrate by simply omitting the gate oxidation. All such devices, with dimensions ranging from 1 to 500  $\mu\text{m}$ , are contained on a single chip designated A. Larger devices, with dimensions ranging from 1000 to 4000  $\mu\text{m}$ , are included on chips B and C1 through C4. The aim has been to provide maximum experimental flexibility by including a wide range of available device dimensions, which permits electrically active areas to span more than seven orders of magnitude. A summary of all devices contained in the MOS mask system is presented in Table I in the form of a device key, listing MOS device, nominal dimension, the chip assignment, and the appropriate probing pad number. Detailed descriptions of each device follow in Subsections 2.1 to 2.5. Dimensional data on all device structures are contained in the pad keys shown in Tables II through V.

## 2.1 Sixfold MOS capacitor group (HEXCAP)

The design of the lithographic mask system has evolved from a sixfold set of MOS capacitors (HEXCAP) that can be implemented at almost any stage of device processing to provide electrical characterization of dielectric layers.

### 2.1.1 FOXCAP and GOXCAP

The field oxide capacitor (FOXCAP) and gate oxide capacitor (GOXCAP) are shown in Fig. 1. For this pair of capacitors, the dimension  $L_1$  increases through the sequence 50, 100, 200, 500, 1000, 2000, and 4000  $\mu\text{m}$ . In all cases the dimension  $L_2 = L_1 + 10 \mu\text{m}$ , to conform with relaxed design rules to minimize registration errors (see Section 3.3). All FOXCAP and GOXCAP devices with  $L_1 \leq 500 \mu\text{m}$  are contained on the A chip, with peripheral probing pad locations coinciding with the Poon Tester<sup>1</sup> and the Process Monitor<sup>4</sup> chips. To facilitate MOS characterization of thick oxides or deposited dielectrics, a limited number of HEXCAP devices with  $L_1$  equal to 1000 and 2000  $\mu\text{m}$  were assigned to a larger chip B with area equal to that of ten standard Poon chips. The largest FOXCAP and GOXCAP devices on the C1 chips (see Section 3.1.3) are not square. Rectangular GOXCAP dimensions  $L_3$  and  $L_4$  for the largest devices have been selected so that

$$L_3 L_4 = (4000 \mu\text{m})^2. \quad (1)$$

The rectangular structures of FOXCAP and GOXCAP are shown in Fig. 16. Specific values for  $L_3$  and  $L_4$  were selected so that the overall C1 chip dimensions could be adjusted to accommodate the contact metallization test chip (METEST) (see Sections 3.1.4 and 3.2).

In each HEXCAP group, the portions of the polysilicon (POLY) areas that overlay the gate and source and drain (GASAD) areas are

equal. Thus, the capacitors formed over gate oxide ( $C_{GOX}$ ) are equal among all elements of each HEXCAP group except FOXCAP. Furthermore, the total POLY areas are equal for FOXCAP, GOXCAP, and the devices with comb-shaped electrodes (POLYCOMB) to facilitate use with automatic testing programs. Thus, constant parasitic capacitance contributed by the field oxide ( $C_{FOX}$ ) in parallel with  $C_{GOX}$  has been maintained among GOXCAP, GOXCOMB, and POLYCOMB (see following Section 2.1.2) for convenience in software development. Detailed dimensional data are contained in the pad keys, Tables II, III, and IV.

It should be clear from Fig. 1 that omission of the gate oxide in GOXCAP results in a structure suitable for ohmic contact, Schottky barrier diode, and p-n junction experiments. For this reason the dimension  $L_1$  also decreases through the sequence 20, 15, 10, 7, 5, 4, 3, 2, 1.5, and 1.0  $\mu\text{m}$  on the A chip. It is not expected, however, that the smallest GOXCAP windows would be routinely resolved by standard photolithographic processes. To resolve the smallest features, the GASAD mask level has been reissued for X-ray lithography. Because the X-ray target is nearly a point source, the finite separation between X-ray mask and the silicon substrate results in a slight magnification of the X-ray image. To compensate for the magnified X-ray image, a small demagnification of all features on the X-ray mask was required to make the X-ray lithographic level compatible with previous or subsequent optical lithographic levels. Alignment in the X-ray exposure facility, however, is done optically. Therefore the spacing between alignment features on the X-ray mask must remain the same as on the optical mask. With these modifications, X-ray lithography can be interleaved with more easily performed photolithographic steps.

The use of the GOXCAP structure to form a guarded Schottky diode is illustrated in Fig. 2, in which a guard ring with width  $W$  is diffused into the substrate before GASAD lithography. Guard-ring width options available in the mask set decrease through the sequence 10, 7, 5, 3, 2, and 0  $\mu\text{m}$  (see Section 4.1). In all cases the guard ring has been centered on the GASAD boundary so that half its width extends into the contacting metal and reduces the effective contact area (see Fig. 2), so that:

$$A_{\text{eff}} = (L_1 - W)^2 \quad (\mu\text{m}^2). \quad (2)$$

For the cases in which  $W \geq L_1$ , the structure simply reduces to a p-n junction. Square or rectangular features have also been included in the GUARDRING level under the FOXCAP structures to provide the option of fabricating buried channel MOS capacitors<sup>5-8</sup> or to investigate the profiles of various ion implantations, such as those used to control threshold or punchthrough or for depletion loads.

### 2.1.2 GOXCOMB and POLYCOMB

To study peripheral effects or defects, structures with expanded gate-oxide/field-oxide periphery were included as the third HEXCAP element. This structure, abbreviated GOXCOMB, is shown in Fig. 3. On the A chip, the GOXCOMB structures comprise rectangular elements of width  $d_1 = 25 \mu\text{m}$ , which are spaced  $d_2 = 10 \mu\text{m}$  apart. The total gate oxide areas of the two GOXCOMB structures on the A chip are  $(200 \mu\text{m})^2$  and  $(500 \mu\text{m})^2$ . To further enhance peripheral effects, GOXCOMB structures have been included on the B and C2 chips with  $d_1 = 5 \mu\text{m}$  and  $d_2 = 10 \mu\text{m}$ . Because of the decreased filling factor associated with reducing  $d_1$  to  $5 \mu\text{m}$ , it was not practical to keep the total gate oxide areas equal to the areas of the associated 1000-, 2000-, and 4000- $\mu\text{m}$  GOXCAP structures; the actual areas are listed in Tables III and IV.

The POLYCOMB structure is similar to GOXCOMB except that the increased perimeter or peripheral expansion occurs at the gate-oxide/polysilicon boundary. The structural detail of POLYCOMB, the fourth HEXCAP element, is shown in Fig. 4. The chip assignments for POLYCOMB are the same as for GOXCOMB except for the 4000- $\mu\text{m}$  structure, which is on the C3 chip. The same values for  $d_1$  and  $d_2$  apply to both COMB structures.

### 2.1.3 OVLAP and NOVLAP with FIELD PLATE

In some cases it is desirable to minimize parasitic capacitance in an MOS structure, i.e., the parallel capacitance composed of the area in which the gate electrode overlaps field oxide. The last pair of HEXCAP devices has been designed to minimize parasitic capacitance, consistent with the design rules discussed in Section 3.3. The OVLAP capacitor, shown on the left in Fig. 5, has a  $5\text{-}\mu\text{m}$  overlap of the gate electrode onto the surrounding field oxide. The NOVLAP capacitor, shown on the right, has a gate electrode that has been retracted  $5 \mu\text{m}$  from the GASAD boundary. The portion of the gate electrode that covers gate oxide in the OVLAP capacitor is equal to the area of the gate electrode in the NOVLAP capacitor. Both OVLAP and NOVLAP must be probed directly, because they are completely surrounded by a field plate that can be used to control the surface potential near the edges of each capacitor.

If a metallic silicide is formed in place of gate oxide, the resulting structure allows investigation of the effects of overlying metallization when excessive metallic penetration at contact window edges is suspected.<sup>9</sup> For these investigations it may be useful to include the GUARDRING option prior to silicide formation.

## 2.2 Sheet resistance group (SADSHEET and POLYSHEET)

Two three-terminal structures on the A chip can be used to obtain sheet resistance data from lines 400  $\mu\text{m}$  long. Thus, accurate sheet resistance measurements can be made, even though the linewidths may deviate from the coded values because of unknown degrees of overetching or other process variations.

The left side of Fig. 6 shows the structure for measuring polysilicon sheet resistance (POLYSHEET). It consists of two 400- $\mu\text{m}$  lines in series, but the coded linewidths are different:  $W_1 = 5 \mu\text{m}$  and  $W_2 = 8 \mu\text{m}$ . After processing, the actual linewidths may differ from the coded linewidths by a constant amount,  $\epsilon$ . Assume that a positive value for  $\epsilon$  corresponds to a linewidth loss from the coded value,  $W_i$ . If the resistance of each line is measured, it is possible to solve for both the sheet resistance and a constant linewidth loss,  $\epsilon$ . It can be shown that

$$R_S = \frac{R_1 R_2}{R_1 - R_2} \cdot \frac{W_2 - W_1}{L} = 7.5 \times 10^{-3} \frac{R_1 R_2}{R_1 - R_2} (\Omega/\square) \quad (3)$$

and

$$\epsilon = \frac{R_1 W_1 - R_2 W_2}{R_1 - R_2} = \frac{5R_1 - 8R_2}{R_1 - R_2} (\mu\text{m}). \quad (4)$$

A GUARDRING structure is included beneath some of the POLYSHEET lines. If the GUARDRING option were used, for example, to etch channels of various widths in a field oxide, the resulting POLYSHEET structure would provide information on poly-Si linewidths within oxide channels or straddling oxide steps.

Source-and-drain sheet resistance and linewidth variations can be determined from measurements on the structure shown on the right side of Fig. 6. The coded dimensions of the SADSHEET lines are exactly the same as for POLYSHEET, and eqs. (3) and (4) apply.

When the GUARDRING level precedes GASAD, some of the SADSHEET lines are imbedded into the guard-ring diffusion. Such a structure could be useful in determining the sheet resistance of a metallic silicide line for cases in which a low Schottky barrier height between the silicide and the substrate would interfere with electrical measurements.

## 2.3 van der Pauw group (VANDERPAUW)

The four-terminal symmetric structure shown in Fig. 7 has been provided on the A chip to make accurate determinations of polycrystalline silicon (poly-Si) sheet resistance in a way that is independent of the actual shape of the resistive pattern.<sup>10</sup> The poly-Si lines have been extended to the probing pads so that window and metallization lithography is not necessarily required. However, the option to have

overlying metallization on the lines leading to the van der Pauw pattern, with windows to the underlying poly-Si, is available for unusual circumstances in which polysilicon sheet resistance may be very high.

The other VANDERPAUW structures are on the C4 chip, and consist of GUARDRING and GASAD patterns. The combinations POLY/GUARDRING and POLY/GASAD VANDERPAUW patterns are also included on the C4 chip to enable measurements of inversion layer sheet resistance<sup>11</sup> and to investigate CHANSTOP performance.

#### **2.4 IGFET group**

It is not anticipated that the IGFET group of devices will be utilized as often as the HEXCAP group of MOS capacitors, because the IGFETs require a minimum of four mask levels (GASAD, POLY, WINDOW, and METAL). For this reason, all IGFETs have been relegated to the B chip that has more available terminals than the A chip, although it may be less convenient for automatic probing.

Most of the IGFETs are included in one group with common sources and gates. The structure of the IGFET with  $L = 20 \mu\text{m}$  is shown in Fig. 8. All gates are  $100 \mu\text{m}$  wide, and the gate lengths  $L$  descend through the sequence 20, 15, 10, 8, 6, 5, 4, 3, 2, 1.5, and  $1.0 \mu\text{m}$ . It is not anticipated, however, that the shortest gates will be resolved with ordinary photolithographic processing. Therefore, the POLY mask level may also be reissued for X-ray lithography. Even with wide variations in processing, the range of gate lengths provides a means to determine the true (electrically active) channel length from a plot of  $\beta^{-1}$  versus the coded value for  $L$ , where  $\beta$  is the transconductance of the IGFET.

The GUARDRING option is available on all elements of the IGFET group. The guard ring straddles the GASAD feature on the three sides that are not adjacent to the gate, as shown in Fig. 9. Such a structure may be useful to minimize edge leakage when Schottky barrier sources and drains are investigated.

In some cases it is useful to make C-V (capacitance measured as a function of voltage) measurements of the gate electrode in an active IGFET. But practical IGFETs are generally designed to minimize gate capacitance in order to maximize switching speed, and the true gate capacitance is difficult to separate from parasitic capacitance. Therefore, four large-gate IGFETs have also been included on the B chip with gate dimensions descending through the sequence 500, 300, 200, and  $100 \mu\text{m}$  square.

#### **2.5 Gated diode group (GATODE)**

The measurement of the depleted surface recombination velocity

$s_0$ <sup>12</sup> is especially useful in evaluating the effectiveness of a low-temperature anneal to reduce surface state density<sup>13</sup> and to investigate the effects of radiation damage.<sup>14,15</sup> In the determination of  $s_0$  it is necessary to directly control the surface potential near an MOS capacitor by means of a third electrode. This option has been made available by means of the gated diode group (GATODE), with dimensions decreasing through the sequence 500, 300, 200, and 100  $\mu\text{m}$  square. The structure of the 100- $\mu\text{m}$  gated diode is shown in Fig. 10. Obviously, a nearly equivalent structure could be realized by simply shorting the source to the drain of one of the large area IGFETs. The GATODE structures on the B chip differ from the IGFETs, however, in that the source-and-drain diffusions completely surround the gate electrode except for a 10- $\mu\text{m}$  tab that connects the gate electrode to the probing pad. The GATODE structures are inverted from the usual gate-controlled diode in the sense that the p-n junction surrounds the gate electrode, whereas, the original gate-controlled diode structure consisted of an MOS capacitor in the form of a ring that surrounded a p-n junction.<sup>16,17</sup> The advantage of the GATODE structure is that better control of minority carrier production is possible when primary interest is centered on the properties of deeply depleted MOS capacitors.<sup>18,19</sup>

## **2.6 Contact metallization test chip (METEST)**

Electromigration studies generally require high current densities, of the order of  $10^6$  A/cm<sup>2</sup>, to achieve accelerated aging at a practical rate.<sup>20</sup> In the vicinity of contact windows, electromigration has been difficult or impossible to study, because the only test structure available has been the 100-window arrays on the Poon Tester A and C chips.<sup>1</sup> At the required current density, the sum of the voltage drops accumulated over a 100-window array often exceeds the breakdown voltage of the p-n junction that exists beneath each pair of windows. The contact metallization test chip D (METEST) has been designed to avoid large accumulated voltage drops by means of a tapped string of metal-to-diffusion windows, as shown in Fig. 11. Each tapped string is composed of series combinations of 1, 2, and 4 contact cells. Structural detail of one such contact cell is shown in Fig. 12 for a window size of 7  $\mu\text{m}$ . With the D chip, a reliability engineer can select 2, 4, 6, 8, 10, or 14 windows in series, depending upon the particular breakdown characteristics of the structure. Each tapped string has been reproduced for a variety of contact window sizes, decreasing through the sequence 7, 5, 3, 2, 1.5, and 1.0  $\mu\text{m}$  square. It is not expected, however, that the smallest windows would be routinely resolved by standard photolithographic processes. Therefore, the POLYCON mask level, which contains the contact windows for the METEST structures, has been reissued for X-ray lithography. The X-ray alignment features

have been modified appropriately to make the X-ray lithographic level compatible with prior GASAD and subsequent POLY optical lithographic levels.

There has been a tendency to avoid rectangular contacts with large aspect ratios, i.e.,  $L/W > 3$ . The reason is related to photolithographic exposure problems with very small contact windows, such as  $W \leq 2 \mu\text{m}$ . When additional contact area is required, parallel strings of square contact windows are often preferred to large, rectangular contacts. For this reason, the smallest contact windows on the D chip have been repeated in multiples of 4, 6, and 8 parallel windows for the 2-, 1.5-, and 1.0- $\mu\text{m}$  windows, respectively. Structural detail of a multiple-window contact cell is shown in Fig. 13 for 4-window, 2- $\mu\text{m}$  contacts. Obviously, the current does not divide evenly among the windows in such a multiple-window structure, but the extra contact windows can be regarded as providing an experimental backup when the first window fails. The multiple window strings also tend to increase continuity probability when working close to the limit of lithographic resolution.

### III. ORGANIZATION

The lithographic mask system for fine-line process development has been organized on the wafer so that the simplest structures with the most convenient dimensions are available together on chip A. The included structures are HEXCAP, SADSHEET, POLYSHEET, and VANDERPAUW (see Section II). Perhaps the most unusual feature of the mask organization has stemmed from the enormous range of device sizes that have been made available to maximize experimental flexibility. Thus, structural dimensions ranging from 1  $\mu\text{m}$  to 4000  $\mu\text{m}$  are all present together on the same wafer. Furthermore, the largest areas can be used for direct probing, for evaporated MOS dots, or can be easily cleaved for Auger, SEM, TEM, X-ray, and other analytical investigations. It is the large range of device sizes (4-1/2 orders of magnitude) which has dictated chip designation and wafer layout.

#### 3.1 Test chip designation

##### 3.1.1 Chip A (1600 $\times$ 4096 $\mu\text{m}$ )

Most of the devices with dimensions ranging between 1.0 and 500  $\mu\text{m}$  are included on the A chip. A composite view of the POLY and WINDOW levels of the A chip is shown in Fig. 14. Both the dimensions of this chip and the placement of the 36 probing pads have been selected to coincide with the Poon Tester<sup>1</sup> and the Process Monitor<sup>4</sup> chips to facilitate automated probing with existing probe cards.

##### 3.1.2 Chip B (8000 $\times$ 8192 $\mu\text{m}$ )

The dimensions of the B chip are integral multiples ( $5 \times 2$ ) of the A

chip to facilitate wafer layout (see Section 3.2), and the area is equal to that occupied by ten A chips. A composite view of the POLY, WINDOW, and METAL levels of the B chip is shown in Fig. 15. The size of the B chip has been selected to accommodate HEXCAP groups measuring 1000 and 2000  $\mu\text{m}$  square, in the case of FOXCAP, GOXCAP, OVLAP, and NOVLAP. The GOXCOMB and POLYCOMB capacitors are nearly square, and have been laid out so that the equivalent areas are equal to the areas of the square capacitors (see Table III). Please note that to provide adequate resolution for illustration in Fig. 15, the width and spacing of the tines in POLYCOMB have been magnified 3X, and the number of tines has been accordingly reduced by a factor of 3 so that the overall dimensions remain unchanged. Accurate dimensional data on POLYCOMB can be measured from Fig. 15 by scaling down detail 3X. Areas and perimeters are listed in Table III. Also the gap between the field plate and the OVLAP and NOVLAP capacitors has been widened 3X. The IGFET arrays and gated diodes were assigned to the B chip for two reasons: (i) at least four photolithographic steps are required to realize completed devices, so it is anticipated that these will not be used as frequently as the two-level structures on the A chip; (ii) the IGFET and gated diode arrays require 23 additional probing pads that are not available on the A chip. To provide adequate resolution for illustration, the spacing between gates and sources and drains has been increased 3X in Fig. 15. In the gated diodes the space between gates and junctions has also been increased 3X.

### 3.1.3 Chip C (6400 $\times$ 8192 $\mu\text{m}$ )

The largest MOS capacitors, with areas measuring  $(4000 \mu\text{m})^2$ , had to be allocated to four separate chips. The C1 chip contains FOXCAP and GOXCAP capacitors with areas equal to  $(4000 \mu\text{m})^2$ . A composite view of the POLY and WINDOW levels is shown in Fig. 16. The C2 chip contains a rectangular GOXCOMB structure with area somewhat reduced from the rectangular devices on chip C1; the exact coded areas are listed in Table IV. A composite view of the POLY and WINDOW levels is shown in Fig. 17. The C3 chip contains a rectangular POLYCOMB structure with area somewhat reduced from the rectangular devices on chip C1; the exact coded areas are listed in Table IV. A composite view of the POLY and WINDOW levels is shown in Fig. 18. To provide adequate resolution for illustration in Fig. 18, the width and spacing of the tines have been magnified 3X, and the number of tines has been accordingly reduced by a factor of 3 so that the overall dimensions remain unchanged. Accurate dimensional data can be measured from Fig. 18 by scaling down detail 3X. Areas and perimeters are listed in Table IV.

The C4 chip contains both OVLAP and NOVLAP rectangular capacitors with areas measuring  $(4000 \mu\text{m})^2$  and surrounded by a field plate. A composite view of GUARDRING, GASAD, POLY, and WINDOW levels is shown in Fig. 19. For the purpose of illustration the gap between the field plate and the OVLAP and NOVLAP capacitors has been widened 3X in Fig. 19. All C chips measure  $6400 \times 8192 \mu\text{m}$ . When combined with the D chip (see next section), the overall dimensions of the combination are exactly equal to the dimensions of the B chip or a  $2 \times 5$  array of A chips.

### **3.1.4 Chip D (1600 × 4096 μm) METEST**

A composite view of the POLY and WINDOW levels of the contact metallization test chip D (METEST) is shown in Fig. 20. Both the dimensions and the locations of the 36 probing pads of the METEST chip have been selected to coincide with the Poon Tester<sup>1</sup> and the Process Monitor<sup>4</sup> chips to facilitate automatic probing. It is expected that the tapped strings with 2-, 3-, 5-, and 7- $\mu\text{m}$  windows will be used most extensively. These strings have been terminated on probing pad numbers 3, 4, 5, 6 (3  $\mu\text{m}$ ), 9, 10, 11, 12 (2  $\mu\text{m}$ ), 21, 22, 23, 24 (7  $\mu\text{m}$ ), and 27, 28, 29, 30 (5  $\mu\text{m}$ ) to coincide with existing metallization probing cards.

Detailed dimensional data for the D chip are listed in Table V. Table V differs from Tables II through IV in many respects, because the tapped strings were not intended to provide capacitance data. There are no GUARDRING features. Entries tabulated under GASAD tub refer to features straddling the indicated pad numbers, although all tubs within each string are connected in parallel after metallization with the POLY level. Entries tabulated under POLYCON window refer to the total cross-sectional area of a single tub input or output. However, current density is not expected to be uniform over any given window and especially among multiple windows. Entries tabulated under POLY are intended to aid in estimating string resistances from the sheet resistance of the metallization layer provided by the POLY pattern. Taps and ties are defined in Fig. 11, and the equivalent numbers of squares straddling each pair of pad numbers are indicated. The equivalent number of squares for the contact areas were not included, because these depend upon the sheet resistance of the underlying tubs.

### **3.2 Wafer layout**

The location of each of the test chips described in the preceding section is shown in Fig. 21. The A chip is the most numerous, totaling 130 and arranged in blocks of 10 to form the cross-shaped pattern coded AX in Fig. 21. The number X, following A, denotes the width of

the guard ring when the GUARDRING (N35) option is selected. The symbol A0 denotes unguarded devices. The arrangement of the A chips, which have been laid out to permit automatic probing with existing facilities, is obviously intended to reveal horizontal and vertical parametric trends on test wafers.

There are only twelve B chips, which contain 1000- and 2000- $\mu\text{m}$  devices. When the GUARDRING (N35) option is selected, there are only two chips for each guard-ring width, i.e., 2, 3, 5, 7, and 10  $\mu\text{m}$ . As in the case of the A chip, guard-ring width is represented by X in the notation BX, shown in Fig. 21. The asterisks denote undefined guard-ring diffusions or implantations that cover the entire chip for evaluation of guard-ring performance without a parallel Schottky diode.

There are six C1 and six C4 chips on the test wafer. In each case, three are unguarded and three have 10- $\mu\text{m}$  guard rings, *viz.* C1-0 and C1-10 in Fig. 21. The C2 and C3 chips contain large GOXCOMB and POLYCOMB structures, respectively. Two are unguarded, *viz.* C2-0, and two have guard rings, *viz.* C2-10, in Fig. 21. In the case of GOXCOMB, the guard-ring width does not permit interdigitating the individual COMB elements, so the guard-ring option provides a buried diffused tub beneath the GASAD structure. As in the B chip, the asterisks denote undefined guard-ring diffusions or implantations.

The contact metallization test chip is denoted D in Fig. 21. There are a total of 40, which have been divided equally among the four quadrants of the wafer.

At the top and bottom and left and right are four alignment patterns that have been designed to permit aligning any mask to any other in any order (see Section 3.4). Also associated with each alignment pattern are two TEM test chips that have been specially designed to facilitate transmission electron microscopic (TEM) analysis.<sup>21</sup> (See Section 3.5.)

### 3.3 Design rules

The lithographic mask system has been designed specifically for fine-line process development and process control. Since the masks are generally to be used with standard photolithographic processing, it obviously would be inappropriate to interpret data in terms of fine-line design rules. Consequently, most device structures have been designed to tolerate up to 5  $\mu\text{m}$  in misalignment errors. Figure 22 is an example of such relaxed design rules, showing the source-gate structure of a typical element from the IGFET group (see Section 2.4). The same structure also applies to the junction contacts of the gated diode group (GATODE) (see Section 2.5). In general, all contact windows have a minimum width of 5  $\mu\text{m}$ , and all overlapping regions are a minimum of 5  $\mu\text{m}$ .

A number of exceptions exist, however. Most prevalent are the GOXCAP group (see Section 2.1.1), which has been deliberately continued to a minimum size of 1.0  $\mu\text{m}$  to enable special experiments with nonstandard photolithographic processes or X-ray lithography. A similar philosophy has been applied to the entire contact metallization test chip D (METEST), which has been fully described in Section 2.6.

### 3.4 Alignment features

It is hoped that maximum flexibility has been achieved by the use of a modified version of the standard Perkin-Elmer projection (PEP) alignment features. These modified PEP (MOPEP) features are shown in Figs. 23a through f and are presented in the anticipated “normal” order or suggested sequences, i.e. GUARDRING, GASAD, POLYCON, POLY, WINDOW, and METAL. The upper set of MOPEP features in each of Figs. 23a through f corresponds to the “normal” processing sequence. Unlike alignment procedures for virtually all device codes, each mask in this lithographic system must be aligned to the immediately preceding level, because levels prior to the one immediately preceding introduce overlapping patterns. But the unique feature of the MOPEP alignment features is that any number of levels can be skipped. For example, it has been anticipated that a popular sequence may be GASAD followed by POLY only. The alignment feature remaining on the test wafer after GASAD lithography is shown in Fig. 23b. Alignment of POLY to GASAD corresponds to a “normal” processing sequence, so the second MOPEP feature in the upper half of Fig. 23d would have to be aligned to the second (right-hand) MOPEP feature in the upper half of Fig. 23b. The left-hand MOPEP feature in Fig. 23b is simply ignored, because the GUARDRING level was omitted.

The lower set of MOPEP alignment features in Figs. 23a through f have been included to enable an “inverted” processing sequence. Such an “inverted” processing sequence might be required for some unique or novel structure that was not originally intended or anticipated. For example, POLY features can be defined on the surface of an unpatterned field oxide. After oxidation, or deposition of an intermediate dielectric layer, it might be necessary to define additional conductive features of either poly-Si or metal directly over the original poly-Si features. This capability is available by using a “GASAD” level with reverse tone (see Table VI), consisting of opaque features within a transparent background, to produce conductive patterns in polysilicon or metal. Alignment is carried out by inserting the central MOPEP feature in the lower half of Fig. 23b into the right-hand MOPEP feature on the lower half of Fig. 23d, which would be the pattern left

on the test wafer after POLY lithography. (Tone reversal is not applied to MOPEP features.)

To be consistent with the relaxed 5- $\mu\text{m}$  design rules discussed in the preceding section, the right-hand MOPEP feature in the upper halves of Figs. 23a through f are all 25  $\mu\text{m}$  wide. All of the other MOPEP features in the upper halves of Fig. 23b through e and all MOPEP features in the upper half of Fig. 23f are 20  $\mu\text{m}$  wide. A similar scheme that provides 2.5- $\mu\text{m}$  frames for alignment in inverted order applies to the lower halves of Figs. 23a through f. The 2.5- $\mu\text{m}$  alignment frame has resulted from a compromise that should offset the effects of photolithographic processing variations, but proper alignment does require some judgment on behalf of the alignment operator to optimize registration of sequential mask levels.

### 3.5 TEM test chip

Sample preparation techniques for transmission electron microscopy (TEM) usually produce sections sufficiently thin over a region that may vary between 40 and 100  $\mu\text{m}$ . All morphological features essential for process evaluation can be translated into the area for TEM study by a special test pattern 1 mm wide and approximately 6.7 mm long, and with a structural period of 29.5  $\mu\text{m}$ . The TEM test pattern is shown in Fig. 24, showing gate oxide within regions defined by GASAD, contact windows formed by POLYCON, layers of poly-Si defined by POLY, and subsequent P-glass and metallization levels. Thus all windows, steps, and other peripheral features normally encountered in fine-line process development are reproduced over 200 times within each TEM test chip. Two TEM test chips have been placed symmetrically with respect to each MOPEP alignment feature (see Fig. 21), and no active device areas have been sacrificed. A total of eight TEM test chips have been incorporated into the lithographic mask system, and the feature boundaries of each TEM test chip are oriented orthogonal to a  $\langle 100 \rangle$  cleavage plane so that the cross section shown in Fig. 24 can be readily obtained from widely separated areas of the wafer. The TEM test chip shown in Fig. 24 differs from the one published by Sheng and Marcus,<sup>21</sup> partially because the chip in Fig. 24 was designed for a fine-line process that does not involve selective oxidation.

## IV. MASK LEVELS

The six mask levels that comprise the normal tone portion of the lithographic mask system are listed in the upper part of Table VI and are intended to be used with positive photoresist. The suggested sequence reflects the primary purpose for which each level was intended; a few examples are shown in Table VII. Each mask level

contains four full sets of MOPEP alignment features, corresponding to “normal” and “inverted” processing sequences. Thus, it is possible to align any mask level to any other, thereby permitting novel device structures with processing sequences that have not been anticipated. (See Section 3.4 for alignment details.) The three levels that have been issued in reverse tone are shown in the lower part of Table VI. They have been intended for use with negative photoresist, uniform gold metallization, selective oxidation or other special processes.

#### **4.1 GUARDRING**

The GUARDRING level would generally be omitted for most MOS processing investigations. In a sense, it may be viewed as being analogous to the isolation tub diffusion that occurs in CMOS processing prior to GASAD. The principal purpose of the guard ring is to provide p-n junctions that straddle the boundaries of Schottky barrier diodes to electrically isolate metallization edges that often obscure barrier characterization. The GUARDRING level is unusual because it comprises a subset of six patterns that provide guard-ring widths of 10, 7, 5, 3, 2, and 0  $\mu\text{m}$ . The location of each of these subsets is indicated in Fig. 21 by the final hyphenated integers X, i.e., C1-X. Each guard ring is located such that it frames each GASAD boundary symmetrically. In the case of the smallest GASAD features, the area enclosed by the larger GUARDRING features vanishes, and a p-n junction is formed, which is useful for evaluating guard-ring performance. The asterisks denote undefined diffused or implanted areas that cover the entire chip. These chips can also be used to characterize the guard rings independently from the other features or to evaluate ion implantation profiles (see Section 2.1.1).

#### **4.2 GASAD**

The GASAD level is normally the first level that would be used for MOS process development and monitoring. The principal purpose of the GASAD level is to open up areas in the field oxide in preparation for a possible ion implantation, for threshold control, followed by gate oxidation. If gate oxidation is omitted, however, the GASAD level provides a range of areas for investigations of contact resistance, Schottky barrier diodes, and p-n junctions. Most of the patterns provided by GASAD are square and progressively increase in size through the sequence 1.0, 1.5, 2, 3, 4, 5, 7, 10, 15, 20, 50, 100, 200, 500, 1000, 2000, and 4000  $\mu\text{m}$ . At 200  $\mu\text{m}$  and above, comb-like structures (GOXCOMB) are included in the GASAD level for investigation of peripheral effects and defects. The 200- and 500- $\mu\text{m}$  GOXCOMB structures consist of a series of 25- $\mu\text{m}$  slots separated by 10  $\mu\text{m}$  of field oxide (see Fig. 3). All of the slots are interconnected to guarantee

equalization of surface potential. The choice of the relatively large slots assures that all of the MOS capacitors within HEXCAP will have nearly equal gate areas despite typical variations in linewidth owing to processing variables. Obviously, GOXCOMB can be utilized for aggravating edge effects or defects in device structures. In the largest GOXCOMB structures (1000, 2000, and 4000  $\mu\text{m}$ ) the slots have been reduced to 5  $\mu\text{m}$ , separated by 10  $\mu\text{m}$  of field oxide, to further increase the aggravation caused by peripheral electric fields and edge-related defects.

It is anticipated that in most MOS process monitoring and development applications it would be possible to use wafers that had been previously patterned using the GASAD mask. Thus, the most important device structures would be complete after only one additional photolithographic step (see POLY, Section 4.4).

### 4.3 POLYCON

For most MOS monitoring or process development applications the POLYCON level would be omitted. The main reason for including this level has been to provide windows to the underlying GASAD tub diffusions or ion implantations that are required by the metallization test chip D (METEST). However, when investigation of p-n junctions with large areas or extended peripheries is required, the GASAD mask is used to define the diffused or ion-implanted regions. In this case the POLYCON level provides the required contact windows to junctions formed by the GASAD features. Evaluation of p-n junctions thus requires a minimum of three mask levels, because the POLY level must be used for metallization. If it should be necessary to investigate the effects of high-temperature processing after poly-Si deposition, such as oxidations and/or insulating depositions, additional mask levels WINDOW and METAL might be required.

### 4.4 POLY

It has been anticipated that the combination of GASAD followed by POLY would be the most widely used sequence of mask levels for MOS process development. For this reason, the POLY patterns have been extended to include the probing pads so that the poly-Si can be probed directly. Typical probe-spreading resistance measurements are of the order of 50 $\Omega$  in a poly-Si film with a sheet resistance of 20  $\Omega/\square$ , *providing that the probe tips are sufficiently hard and sharp enough to pierce 50 $\text{\AA}$  of native oxide that typically occurs on the surface of  $n^+$ -poly-Si.* For this purpose tungsten carbide probes are recommended with tip radii of  $5 \times 10^{-4}$  cm or less. The use of palladium probes with planar tips  $5 \times 10^{-3}$  cm in diameter has been found to be unsatisfactory for probing  $n^+$ -poly-Si. Unfortunately, many probe

cards designed for automatic probing cannot be used for probing n<sup>+</sup>-poly-Si directly, because erratic probe contact resistance may range over many orders of magnitude, sometimes exceeding 10 megohms. Regular cleaning and inspection of probe tips are mandatory when probing n<sup>+</sup>-poly-Si. Ion implantation causes an amorphous layer of semi-insulating material to occur near the surface of n<sup>+</sup>-poly-Si. Thus, any n<sup>+</sup>-poly-Si that has been exposed to ion implantation must be annealed at 950°C in N<sub>2</sub> for 30 minutes to avoid excessive probe resistance. In automatic probe stations with existing probe cards, the POLY mask level may be used to pattern aluminum, with or without an intervening poly-Si layer, to reduce probe resistance. Alternatively, the WINDOW and METAL mask levels can be used (see next sections).

At 200 μm and above, comb-like structures (POLYCOMB) are included in the POLY level for investigating edge effects or defects (see Fig. 4). The width and separation of individual elements are the same as for GOXCOMB (see GASAD, Section 4.2).

#### **4.5 WINDOW**

Occasionally, it is necessary to investigate changes in device characteristics resulting from oxidations, insulating depositions, or annealing after poly-Si definition. For this purpose the WINDOW level has been provided, which opens 90-μm square windows over each probing pad. In most cases the poly-Si could then be probed directly at the contact pads without resorting to aluminum deposition and lithography, providing the recommendations contained above in Section 4.4 are followed. In the case of the OVLAP and NOVLAP MOS capacitors surrounded by field plates, the capacitors must be probed directly. For this purpose, the WINDOW level also contains large contacting areas over each capacitor surrounded by a field plate. Occasionally, it may be beneficial to access the van der Pauw pattern with aluminum metallization. Therefore, the WINDOW level also contains four contacts oriented directly over the leads at the edges of the van der Pauw structures to provide conduction to overlying metal lines leading to the probing pads. The WINDOW level is also required for contacts to the sources and drains of IGFETs, the junction terminal of the GATODEs, and to SADSHEET.

#### **4.6 METAL**

As in the case of the GUARDRING and POLYCON levels, it is expected that the METAL level could be omitted in most MOS monitoring or process development applications. The principal exceptions include IGFETs and gated diodes, in which structures the re-

quired electrical continuity could not have been provided by the POLY level alone.

In most cases the metal has been excluded from any areas where poly-Si is in contact with gate oxide. The reason for this exclusion stems from experimental evidence that aluminum, sintered into polycrystalline silicon, sometimes deteriorates the dielectric breakdown strength of underlying gate oxides, especially if the gate oxide is very thin (i.e.,  $\leq 250\text{\AA}$ ). Exception occurs for all of the MOS capacitors surrounded by field plates, which must be probed directly. These OVLAP and NOVLAP capacitors thus provide experimental structures appropriate for cases in which it is necessary to compare the breakdown voltages in MOS capacitors with and without overlying metallization.

## V. APPLICATIONS

Most basic circuit elements utilized by unipolar semiconductor integrated electronics can be easily fabricated with an appropriate subset of the lithographic mask system. These elements fall into five general classifications: MOS capacitors, p-n junctions, contacts, sheet resistors, and IGFETs. The contact class can be further subdivided to include guarded and unguarded Schottky diodes, ohmic contacts, and contact metallization test cells. The sheet resistor class can be subdivided to include polysilicon sheet resistors (POLYSHEET), source and drain sheet resistors (SADSHEET), and van der Pauw patterns in the GUARDRING, GASAD, and POLY levels. The IGFET class also includes gated diodes. These thirteen subdivisions of unipolar device structures are shown in the first column of Table VII. The remaining six columns show the *required* mask levels needed to realize a particular device structure. Other mask levels are generally optional, but some may be required for certain experiments.

At the present time the lithographic mask system is in wide use, and more than fifty experiments have been initiated in the Advanced Large-Scale Integration (LSI) Development Laboratory using one or more levels. References 22 through 26 contain published experiments that have utilized this mask system.

## VI. SUMMARY

Fine-line MOS process characterization, determination of base-line parameters, and new process development can be efficiently carried out using the lithographic mask system. By selecting an appropriate subset of photo- or X-ray lithographic mask levels, most unipolar semiconductor circuit elements can be fabricated in an enormous range of sizes. X-ray and trilevel lithographic processes are used only when

absolutely required, and these can be interleaved with photolithographically defined patterns. Any mask level can be aligned to any other, and any number of mask levels can be skipped. Registration tolerance is  $5\ \mu\text{m}$  for most device structures. Most experimental device structures can be completed and ready for electrical evaluation in a fraction of the time required to fabricate the elements on the Poon Tester chips,<sup>1</sup> which are included within the array of fine-line device chips and require six X-ray lithographic levels to complete.

## VII. ACKNOWLEDGMENTS

The author is indebted to W. T. Lynch for initiating this project and for his continued interest, suggestions, and encouragement. The initial ideas were refined and implemented as a result of numerous discussions with E. N. Fuls, R. A. Kushner, E. I. Povilonis, and N. C. Wittwer. The universal alignment capability, resulting in modified PEP (MO-PEP) features, grew out of discussions with N. C. Wittwer, K. G. Schaffner, and R. Colish. The modification of X-ray alignment features to permit mixing of X-ray and photolithography was done by G. A. Coquin. The design of the metallization test chip (METEST) was carried out in collaboration with A. K. Sinha and S. Vaidya. Discussions with J. R. Brews and G. W. Taylor contributed to the design of the gated diodes and the option of buried channel capacitors. The TEM test chip was contributed by R. B. Marcus. Chip layouts were carried out by A. W. Russo, M. L. Oeser, J. P. Boyle, and M. Wagner of the Engineering Information Department. The wafer array was prepared by S. R. Juice, also of the Engineering Information Department. Mask fabrication was done in the Murray Hill Mask Lab; the cooperation of J. G. Skinner, L. J. Heilos, and other members of the Photomask and Pattern Generation Department is gratefully acknowledged. Final mask inspection was done by D. P. Rosenfeld of the Lithographic Systems Development Department.

Table I—Device key\*

MOS Device	Nominal Dimension	Chip Assignment	Pad No(s).
<b>HEXCAP Group</b>			
<b>1. FOXCAP</b>			
	4000	C1	93
	2000	B	41
	1000	B	3
	500	A	2
	200	A	32
	100	A	26
	50	A	19
<b>2. GOXCAP</b>			
	4000	C1	42
	2000	B	49
	1000	B	7
	500	A	4
	200	A	31
	100	A	29
	50	A	20
	20	A	25
	15	A	1
	10	A	3
	7	A	5
	5	A	7
	4	A	8
	3	A	10
	2	A	11
	1.5	A	12
	1.0	A	14
<b>3. GOXCOMB</b>			
	4000	C2	67
	2000	B	106
	1000	B	34
	500	A	6
	200	A	30
<b>4. POLYCOMB</b>			
	4000	C3	68, 69, 70, 71
	2000	B	95, 96
	1000	B	22
	500	A	9
	200	A	29
<b>5/6. OVLAP and NOVLAP with FIELD PLATE</b>			
	4000	C4	14
	2000	B	75
	1000	B	13
	500	A	13
	200	A	33
	100	A	28
	50	A	18
<b>SHEET Group</b>			
<b>1. SADSHEET</b>			
	80SQ	A	15
	COM	A	16
	50SQ	A	17
<b>2. POLYSHEET</b>			
	80SQ	A	36
	COM	A	35
	50SQ	A	34
<b>VANDERPAUW Group</b>			
<b>1. POLYSI</b>			
	250	A	21
			22
			23
			24
<b>2. GASAD</b>			
	250	C4	101
			102
			103
			104
<b>3. GUARDRING</b>			
	250	C4	96
			97
			98
			99

\* All dimensions are in micrometers. Exact areas and perimeters are listed in Tables II, III, and IV.

Table I-Device key\*(Continued)

MOS Device	Nominal Dimension	Chip Assignment	Pad No(s).			
4. POLYSI and GUARDRING	250	C4	87			
			88			
			89			
			90			
			91			
			92			
			93			
			94			
			5. POLYSI and GASAD	250	C4	75
						76
77						
78						
79						
80						
81						
82						
IGFET Group		Channel				
Terminal	Width	Length				
1. Drain	100	20	B	87		
	100	15	B	86		
	100	10	B	85		
	100	8	B	84		
	100	6	B	83		
	100	5	B	82		
	100	4	B	81		
	100	3	B	80		
	100	2	B	79		
	100	1.5	B	78		
	100	1.0	B	77		
	Common Gate	100		B	88	
	Common Source	100		B	76	
2. Source Gate Drain	500	500	B	53		
			B	54		
			B	55		
3. Source Gate Drain	300	300	B	56		
			B	57		
			B	58		
4. Source Gate Drain	200	200	B	59		
			B	60		
			B	61		
5. Source Gate Drain	100	100	B	62		
			B	63		
			B	64		
GATED DIODE Group		Channel				
Terminal	Width and Length					
1. Junction Gate	500		B	65		
			B	67		
2. Junction Gate	300		B	68		
			B	69		
3. Junction Gate	200		B	70		
			B	71		
4. Junction Gate	100		B	72		
			B	73		
METEST Group		Window Size				
1. Uni-Window	2		D	9-12		
			D	3-6		
			D	27-30		
			D	21-24		
2. Quad-Window	2		D	7, 8, 25, 26		
3. Hex-Window	1.5		D	1, 2, 35, 36		
4. Oct-Window	1		D	31-34		

\* All dimensions are in micrometers. Exact areas and perimeters are listed in Tables II, III, and IV.

Table II—Pad key for chip A\*

Pad No.	MOS Device	Nom. Dim.	GASAD		POLYCON Area	POLY				Window Area	Metal Area	GUARDRING (Width: 2)				
			Area	Peri-meter		Area	Peri-meter	Overlap				Area	Area	Area	Peri-meter	Area
								FOX	GOX							
1	GOXCAP	15	225	60	81	11,750	540	11,525	225	8,100	10,000	120	120	169		
2	FOXCAP	500	0	0	0	271,350	2,490	271,350	0	8,100	10,000	262,144	2,048	0		
3	GOXCAP	10	100	40	16	11,300	530	11,200	100	8,100	10,000	80	80	64		
4	GOXCAP	500	250,000	2,000	100	271,350	2,490	21,350	250,000	8,100	10,000	4,000	4,000	248,004		
5	GOXCAP	7	49	28	0	11,054	524	11,005	49	8,100	10,000	56	56	25		
6	GOXCOMB	500	250,000	20,050	100	361,705	2,818	111,705	250,000	8,100	10,000	40,976	40,100	229,954		
7	GOXCAP	5	25	20	0	10,900	520	10,875	25	8,100	10,000	40	40	9		
8	GOXCAP	4	16	16	0	10,826	518	10,810	16	8,100	10,000	32	32	4		
9	POLYCOMB	500	350,455	2,368	100	271,350	20,702	21,350	250,000	8,100	10,000	4,736	4,736	348,091		
10	GOXCAP	3	9	12	0	10,754	516	10,745	9	8,100	10,000	24	24	1		
11	GOXCAP	2	4	8	0	10,684	514	10,680	4	8,100	10,000	16	16	0		
12	GOXCAP	1.5	2.25	6	0	10,649.75	513	10,647.5	2.25	8,100	10,000	12.25	14	0		
13	FIELD PLATE	500	0	0	0	111,500	8,020	111,500	0	8,100	10,000	0	0	0		
—	OVLAP	500	250,000	2,000	240,100	260,100	2,040	10,100	250,000	240,100	260,100	4,000	4,000	248,004		
—	NOVLAP	500	260,100	2,040	240,100	250,000	2,000	0	250,000	240,100	250,000	4,080	4,080	258,064		
14	GOXCAP	1.0	1.0	4	0	10,616	512	10,615	1.0	8,100	10,000	9	12	0		
15	SADSHEET	80SQ	3,250	990	225	10,000	400	10,000	0	8,325	11,225	1,980	1,980	2,264		
16	SADSHEET	COM			450	10,000	400	10,000	0	8,550	12,450					
17	SADSHEET	50SQ	4,450	984	225	10,000	400	10,000	0	8,325	11,225	1,968	1,968	3,470		
18	FIELD PLATE	50	0	0	0	32,750	1,670	32,750	0	8,100	10,000	0	0	0		
—	OVLAP	50	2,500	200	1,600	3,600	240	1,100	2,500	1,600	3,600	400	400	2,304		
—	NOVLAP	50	3,600	240	1,600	2,500	200	0	2,500	1,600	2,500	480	480	3,364		
19	FOXCAP	50	0	0	0	14,850	690	14,850	0	8,100	10,000	3,844	248	0		
20	GOXCAP	50	2,500	200	100	14,850	690	12,350	2,500	8,100	10,000	400	400	2,304		
21	VANDERPAUW	250	0	0	0	137,675	5,210	137,675	0	8,685	30,912.5	63,504	1,008	0		
22	VANDERPAUW	250	0	0	0	137,675	5,210	137,675	0	8,587.5	15,587.5	63,504	1,008	0		
23	VANDERPAUW	250	0	0	0	137,675	5,210	137,675	0	8,437.5	11,212.5	63,504	1,008	0		
24	VANDERPAUW	250	0	0	0	137,675	5,210	137,675	0	8,602.5	17,462.5	63,504	1,008	0		
25	GOXCAP	20	400	80	100	12,250	550	11,850	400	8,100	10,000	160	160	324		
26	FOXCAP	100	0	0	0	23,350	890	23,350	0	8,100	10,000	12,544	448	0		
27	GOXCAP	100	10,000	400	100	23,350	890	13,350	10,000	8,100	10,000	800	800	9,604		
28	FIELD PLATE	100	0	0	0	37,750	2,520	37,750	0	8,100	10,000	0	0	0		
—	OVLAP	100	10,000	400	8,100	12,100	440	2,100	10,000	8,100	12,100	800	800	9,604		
—	NOVLAP	100	12,100	440	8,100	10,000	400	0	10,000	8,100	10,000	880	880	11,664		
29	POLYCOMB	200	56,350	950	100	55,350	3,782	15,350	40,000	8,100	10,000	1,900	1,900	55,404		
30	GOXCOMB	200	40,000	3,250	100	67,600	1,400	27,600	40,000	8,100	10,000	6,500	6,472	36,754		
31	GOXCAP	200	40,000	800	100	55,350	1,290	15,350	40,000	8,100	10,000	1,600	1,600	39,204		
32	FOXCAP	200	0	0	0	55,350	1,290	55,350	0	8,100	10,000	44,944	848	0		
33	FIELD PLATE	200	0	0	0	56,787.5	4,043	56,787.5	0	8,100	10,000	0	0	0		
—	OVLAP	200	40,000	800	36,100	44,100	840	4,100	40,000	36,100	44,100	1,600	1,600	39,204		
—	NOVLAP	200	44,100	840	36,100	40,000	800	0	40,000	36,100	40,000	1,680	1,680	43,264		
34	POLYSHEET	50SQ	0	0	0				0	8,100	10,000	4,020	814	0		
35	POLYSHEET	COM				35,216	2,778	35,216	0	8,100	10,000					
36	POLYSHEET	80SQ	0	0	0				0	8,100	10,000	2,800	824	0		

\* All dimensions are in micrometers.

Table II—Pad key for chip A\* (Continued)

PAD No.	MOS device	Nom. Dia.	GUARDRING (Width: 3)			GUARDRING (Width: 5)			GUARDRING (Width: 7)			GUARDRING (Width: 10)		
			Area	Peri-meter	Encloused Area	Area	Peri-meter	Encloused Area	Area	Peri-meter	Encloused Area	Area	Peri-meter	Encloused Area
1	GOXCAP	15	180	120	144	300	120	100	420	120	64	600	120	25
2	FOXCAP	500	263,169	2,052	0	265,225	2,060	0	267,289	2,068	0	270,400	2,080	0
3	GOXCAP	10	120	80	49	200	80	25	280	80	9	400	80	0
4	GOXCAP	500	6,000	4,000	247,009	10,000	4,000	245,025	14,000	4,000	243,049	20,000	4,000	240,100
5	GOXCAP	7	84	56	16	140	56	4	196	56	0	289	68	0
6	GOXCOMB	500	60,150	40,100	219,934	100,250	40,100	199,900	140,350	40,100	179,874	200,560	22,404	149,850
7	GOXCAP	5	60	40	4	100	40	0	144	48	0	225	60	0
8	GOXCAP	4	48	32	1	81	36	0	121	44	0	196	56	0
9	POLYCOMB	500	7,104	4,736	346,912	11,840	4,736	344,560	16,576	4,736	342,216	23,680	4,736	338,715
10	GOXCAP	3	36	24	0	64	32	0	100	40	0	169	52	0
11	GOXCAP	2	25	20	0	49	28	0	81	36	0	144	48	0
12	GOXCAP	1.5	20.25	18	0	42.25	26	0	72.25	34	0	132.25	46	0
13	FIELD PLATE	500	0	0	0	0	0	0	0	0	0	0	0	0
—	OVLAP	500	6,000	4,000	247,009	10,000	4,000	245,025	14,000	4,000	243,049	20,000	4,000	240,100
—	NOVLAP	500	6,120	4,080	257,049	10,200	4,080	255,025	14,280	4,080	253,009	20,400	4,080	250,000
14	GOXCAP	1.0	16	16	0	36	24	0	64	32	0	121	44	0
15	SADSHEET	80SQ	2,970	1,980	1,774	4,950	1,090	800	6,130	1,090	648	7,850	1,150	450
16	SADSHEET	COM												
17	SADSHEET	50SQ	2,952	1,968	2,983	4,920	1,968	2,015	6,888	1,968	1,055	9,020	1,024	450
18	FIELD PLATE	50	0	0	0	0	0	0	0	0	0	0	0	0
—	OVLAP	50	600	400	2,209	1,000	400	2,025	1,400	400	1,849	2,000	400	1,600
—	NOVLAP	50	720	480	3,249	1,200	480	3,025	1,680	480	2,809	2,400	480	2,500
19	FOXCAP	50	3,969	252	0	4,225	260	0	4,489	268	0	4,900	280	0
20	GOXCAP	50	600	400	2,209	1,000	400	2,025	1,400	400	1,849	2,000	400	1,600
21	VANDERPAUW	250	64,009	1,012	0	65,025	1,020	0	66,049	1,028	0	67,600	1,040	0
22	VANDERPAUW	250	64,009	1,012	0	65,025	1,020	0	66,049	1,028	0	67,600	1,040	0
23	VANDERPAUW	250	64,009	1,012	0	65,025	1,020	0	66,049	1,028	0	67,600	1,040	0
24	VANDERPAUW	250	64,009	1,012	0	65,025	1,020	0	66,049	1,028	0	67,600	1,040	0
25	GOXCAP	20	240	160	289	400	160	225	560	160	169	800	160	100
26	FOXCAP	100	12,769	452	0	13,225	460	0	13,689	468	0	14,400	480	0
27	GOXCAP	100	1,200	800	9,409	2,000	800	9,025	2,800	800	8,649	4,000	800	8,100
28	FIELD PLATE	100	0	0	0	0	0	0	0	0	0	0	0	0
—	OVLAP	100	1,200	800	9,409	2,000	800	9,025	2,800	800	8,649	4,000	800	8,100
—	NOVLAP	100	1,320	880	11,449	2,200	880	11,025	3,080	880	10,609	4,400	880	10,000
29	POLYCOMB	200	2,850	1,900	54,934	4,750	1,900	54,000	6,650	1,900	53,074	9,500	1,900	51,700
30	GOXCOMB	200	9,360	6,458	35,134	16,250	6,430	31,900	22,750	28,674	6,402	55,200	4,120	23,850
31	GOXCAP	200	2,400	1,600	38,809	4,000	1,600	38,025	5,600	1,600	37,249	8,000	1,600	36,100
32	FOXCAP	200	45,369	852	0	46,225	860	0	47,089	868	0	48,400	880	0
33	FIELD PLATE	200	0	0	0	0	0	0	0	0	0	0	0	0
—	OVLAP	200	2,400	1,600	38,809	4,000	1,600	38,025	5,600	1,600	37,249	8,000	1,600	36,100
—	NOVLAP	200	2,520	1,680	42,849	4,200	1,680	42,025	5,880	1,680	41,209	8,400	1,680	40,000
34	POLYSHEET	50SQ	4,422	826	0	5,226	830	0	6,030	834	0	7,236	840	0
35	POLYSHEET	COM												
36	POLYSHEET	80SQ	3,200	816	0	4,000	820	0	4,800	824	0	6,000	830	0

\* All dimensions are in micrometers.

Table III—Pad key for chip B\*

Pad No.	MOS Device	Nom. Dim.	GASAD		POLYCON Area	POLY				Window Area	Metal Area	GUARDRING (Width: 2)				
			Area	Peri-meter		Area	Peri-meter	Overlap				Area	Area	Area	Peri-meter	Excluded Area
								FOX	GOX							
3	FOXCAP	1,000	0	0	0	1,031,350	4,490	1,031,350	0	8,100	10,000	1,024,144	4,048	0		
7	GOXCAP	1,000	1,000,000	4,000	100	1,031,350	4,490	31,350	1,000,000	8,100	10,000	8,000	8,000	996,004		
13	FIELD PLATE	1,000	0	0	0	209,000	15,170	209,000	0	8,100	10,000	0	0	0		
—	OV LAP	1,000	1,000,000	4,000	980,100	1,020,100	4,040	20,100	1,000,000	980,100	1,020,100	8,000	8,000	996,004		
—	NOVLAP	1,000	1,020,100	4,040	980,100	1,000,000	4,000	0	1,000,000	980,100	1,000,000	8,080	8,080	1,016,064		
22	POLYCOMB	1,000	2,027,970	5,766	90	706,350	270,927	31,325	675,025	8,100	10,000	11,532	11,532	2,022,208		
34	GOXCMB	1,000	675,000	270,020	90	2,038,220	6,256	1,363,220	675,000	8,100	10,000	540,020	540,020	404,994		
41	FOXCAP	2,000	0	0	0	4,051,350	8,490	4,051,350	0	8,100	10,000	4,048,144	8,048	0		
49	GOXCAP	2,000	4,000,000	8,000	100	4,051,350	8,490	51,350	4,000,000	8,100	10,000	16,000	16,000	3,992,004		
53	IGFET: SOURCE		0	0	0	0	0	0	0	10,500	18,250	1,080	1,084	0		
54	IGFET: GATE	500	275,000	2,100	0	265,500	2,500	15,500	250,000	8,100	10,000					
55	IGFET: DRAIN		0	0	0	0	0	0	0	10,500	18,250	1,080	1,084	0		
56	IGFET: SOURCE		0	0	0	0	0	0	0	9,500	18,647.5	680	684	0		
57	IGFET: GATE	300	105,000	1,300	0	103,500	1,700	13,500	90,000	8,100	10,000					
58	IGFET: DRAIN		0	0	0	0	0	0	0	9,500	18,647.5	680	684	0		
59	IGFET: SOURCE		0	0	0	0	0	0	0	9,000	16,487.5	480	484	0		
60	IGFET: GATE	200	50,000	900	0	52,500	1,300	12,500	40,000	8,100	10,000					
61	IGFET: DRAIN		0	0	0	0	0	0	0	9,000	16,487.5	480	484	0		
62	IGFET: SOURCE		0	0	0	0	0	0	0	8,500	14,987.5	280	284	0		
63	IGFET: GATE	100	15,000	500	0	21,500	900	11,500	10,000	8,100	10,000					
64	IGFET: DRAIN		0	0	0	0	0	0	0	8,500	14,987.5	280	284	0		
65	GATOD: JUNC.	500	278,200	2,110	0	0	0	0	0	10,600	18,475	4,220	4,220	276,094		
67	GATOD: GATE		0	0	0	263,475	2,517	13,375	250,100	8,100	10,000					
68	GATOD: JUNC.	300	107,200	1,310	0	0	0	0	0	9,600	15,475	2,620	2,620	105,894		
69	GATOD: GATE		0	0	0	102,350	1,672	12,250	90,100	8,100	10,000					
70	GATOD: JUNC.	200	51,700	910	0	0	0	0	0	9,100	13,975	1,820	1,820	50,794		
71	GATOD: GATE		0	0	0	52,190	1,618	12,090	40,100	8,100	10,000					
72	GATOD: JUNC.	100	16,200	510	0	0	0	0	0	8,100	8,600	1,020	1,020	15,694		
73	GATOD: GATE		0	0	0	21,600	1,100	11,500	10,100	8,100	10,000					
75	FIELD PLATE	2,000	0	0	0	436,500	30,220	436,500	0	8,100	10,000	0	0	0		
—	OV LAP	2,000	4,000,000	8,000	3,960,100	4,040,100	8,040	40,100	4,000,000	3,960,100	4,040,100	16,000	16,000	3,992,004		
—	NOVLAP	2,000	4,040,100	8,040	3,960,100	4,000,000	8,000	0	4,000,000	3,960,100	4,000,000	16,080	16,080	4,032,064		
76	IGFET: COMSOURCE		0	0	0	10,000	400	10,000	0	12,500	61,100	280	284	0		
77	IGFET: DRAIN	1.0	5,100	302	0	10,000	400	10,000	0	8,500	16,935					
78	IGFET: DRAIN	1.5	5,150	303	0	10,000	400	10,000	0	8,500	16,875					
79	IGFET: DRAIN	2	5,200	304	0	10,000	400	10,000	0	8,500	16,870					
80	IGFET: DRAIN	3	5,300	306	0	10,000	400	10,000	0	8,500	16,820					
81	IGFET: DRAIN	4	5,400	308	0	10,000	400	10,000	0	8,500	16,770					
82	IGFET: DRAIN	5	5,500	310	0	10,000	400	10,000	0	8,500	16,720					
83	IGFET: DRAIN	6	5,600	312	0	10,000	400	10,000	0	8,500	16,670					
84	IGFET: DRAIN	8	5,800	316	0	10,000	400	10,000	0	8,500	16,565					
85	IGFET: DRAIN	10	6,000	320	0	10,000	400	10,000	0	8,500	16,465					
86	IGFET: DRAIN	15	6,500	330	0	10,000	400	10,000	0	8,500	16,225					
87	IGFET: DRAIN	20	7,000	340	0	10,000	400	10,000	0	8,500	15,980					
88	IGFET: COMGATE		0	0	0	31,650	5,595	24,100	7,550	8,100	10,000	280	284	0		
95	POLYCOMB	2,000	8,028,720	11,334	90	2,717,840	1,072,598	51,350	2,666,490	8,100	10,000	22,668	22,668	8,017,390		
106	GOXCMB	2,000	2,676,190	1,070,486	90	8,038,970	11,824	5,362,780	2,676,190	8,100	10,000	2,140,972	2,140,972	1,605,708		

\* All dimensions are in micrometers.

Table III—Pad key for chip B\* (Continued)

PAD No.	MOS Device	Nom. Dims.	GUARDRING (Width: 3)			GUARDRING (Width: 5)			GUARDRING (Width: 7)			GUARDRING (Width: 10)		
			Area	Peri-meter	Enclosed Area	Area	Peri-meter	Enclosed Area	Area	Peri-meter	Enclosed Area	Area	Peri-meter	Enclosed Area
3	FOXCAP	1,000	1,026,169	4,052	0	1,030,225	4,060	0	1,034,289	4,068	0	1,040,400	4,080	0
7	GOXCAP	1,000	12,000	8,000	994,009	20,000	8,000	990,025	28,000	8,000	986,049	40,000	8,000	980,100
13	FIELD PLATE	1,000	0	0	0	0	0	0	0	0	0	0	0	0
-	OV LAP	1,000	12,000	8,000	994,009	20,000	8,000	990,025	28,000	8,000	986,049	40,000	8,000	980,100
-	NOVLAP	1,000	12,120	8,080	1,014,049	20,200	8,080	1,010,025	28,200	8,080	1,006,009	40,400	8,080	1,000,000
22	POLYCOMB	1,000	17,298	11,532	2,019,330	28,830	11,532	2,013,580	40,362	11,532	2,007,838	57,660	11,532	1,999,240
34	GOXCOMB	1,000	810,030	540,020	269,994	2,013,580	5,746	0	2,019,330	5,754	0	2,027,970	5,766	0
41	FOXCAP	2,000	4,052,169	8,052	0	4,060,225	8,060	0	4,068,289	8,068	0	4,080,400	8,080	0
49	GOXCAP	2,000	24,000	16,000	3,988,009	40,000	16,000	3,980,025	56,000	16,000	3,972,049	80,000	16,000	3,960,100
53	IGFET: SOURCE		1,620	1,086	0	2,700	1,090	0	3,780	1,094	0	5,400	1,100	0
54	IGFET: GATE	500												
55	IGFET: DRAIN		1,620	1,086	0	2,700	1,090	0	3,780	1,094	0	5,400	1,100	0
56	IGFET: SOURCE		1,020	686	0	1,700	690	0	2,380	694	0	3,400	700	0
57	IGFET: GATE	300												
58	IGFET: DRAIN		1,020	686	0	1,700	690	0	2,380	694	0	3,400	700	0
59	IGFET: SOURCE		720	486	0	1,200	490	0	1,680	494	0	2,400	500	0
60	IGFET: GATE	200												
61	IGFET: DRAIN		720	486	0	1,200	490	0	1,680	494	0	2,400	500	0
62	IGFET: SOURCE		420	286	0	700	290	0	980	294	0	1,400	300	0
63	IGFET: GATE	100												
64	IGFET: DRAIN		420	286	0	700	290	0	980	294	0	1,400	300	0
65	GATOD: JUNC.	500	6,330	4,220	275,044	10,550	4,220	272,950	14,770	4,220	270,864	21,100	4,220	267,750
67	GATOD: GATE													
68	GATOD: JUNC.	300	3,930	2,620	105,244	6,550	2,620	103,950	9,170	2,620	102,664	13,100	1,620	100,750
69	GATOD: GATE													
70	GATOD: JUNC.	200	2,730	1,820	50,344	4,550	1,820	49,450	6,370	1,820	48,564	9,100	1,820	47,250
71	GATOD: GATE													
72	GATOD: JUNC.	100	1,530	1,020	15,444	2,550	1,020	14,950	3,570	1,020	14,464	5,100	1,020	13,750
73	GATOD: GATE													
75	FIELD PLATE	2,000	0	0	0	0	0	0	0	0	0	0	0	0
-	OV LAP	2,000	24,000	16,000	3,988,009	40,000	16,000	3,980,025	56,000	16,000	3,972,049	80,000	16,000	3,960,100
-	NOVLAP	2,000	24,120	16,080	4,028,049	40,200	16,080	4,020,025	56,200	16,080	4,012,009	80,400	16,080	4,000,000
76	IGFET: COMSOURCE		420	286	0	700	290	0	980	294	0	1,400	300	0
77	IGFET: DRAIN	1.0												
78	IGFET: DRAIN	1.5												
79	IGFET: DRAIN	2												
80	IGFET: DRAIN	3												
81	IGFET: DRAIN	4												
82	IGFET: DRAIN	5												
83	IGFET: DRAIN	6												
84	IGFET: DRAIN	8												
85	IGFET: DRAIN	10												
86	IGFET: DRAIN	15												
87	IGFET: DRAIN	20												
88	COMGATE		420	286	0	700	290	0	980	294	0	1,400	300	0
95	POLYCOMB	2,000	34,002	22,668	8,011,728	56,670	22,668	8,000,410	79,338	22,668	7,989,100	113,340	22,668	7,972,150
106	GOXCOMB	2,000	3,211,458	2,140,972	1,070,470	8,000,410	11,314	0	8,011,728	11,322	0	8,028,720	11,334	0

\* All dimensions are in micrometers.

Table IV—Pad key for chips C1 through C4 \*

Pad No.	MOS Device	Chip	Nom. Dim.	GASAD		Polycon Area	POLY				Window Area	Metal Area	GUARDING (Width: 10)		
				Area	Peri-meter		Area	Peri-meter	Overlap				Area	Peri-meter	Enclosed Area
									FOX	GOX					
42	GOXCAP	C1	4,000	16,000,000	16,400	100	16,093,350	16,890	93,350	16,000,000	8,100	10,000	164,000	32,800	15,918,100
43	FOXCAP	C1	4,000	0	0	0	16,093,350	16,890	16,093,350	0	8,100	10,000	16,164,400	16,480	0
67	GOXCOMB	C2	4,000	14,400,000	5,760,010	90	43,221,500	27,020	28,821,500	14,400,000	8,100	10,000	43,211,250	26,530	0
68-71	POLYCOMB	C3	4,000	43,211,250	26,530	90	14,493,350	5,762,167	93,350	14,400,000	32,400	40,000	265,300	53,060	43,078,700
14	FIELD PLATE	C4	4,000	0	0	0	715,500	56,600	715,500	0	8,100	10,000	0	0	0
-	OVLAP	C4	4,000	16,000,000	16,400	15,918,100	16,082,100	16,440	82,100	16,000,000	15,918,100	16,082,100	164,000	32,800	15,918,100
-	NOVLAP	C4	4,000	16,082,100	16,440	15,918,100	16,000,000	16,400	0	16,000,000	15,918,100	16,000,000	164,400	32,880	16,000,000
DOUBLE VANDERPAUW															
75	GASAD	C4	250	67,812.5	1,425	585.0	53,252.5	3,865			8,685.0	43,382.5	0	0	0
76	POLY	C4	260	0	0	0	169,975	7,430	101,875	68,100	8,685.0	34,675.0	0	0	0
77	GASAD	C4	250	67,812.5	1,425	487.5	29,597.5	1,971			8,587.5	19,715.0	0	0	0
78	POLY	C4	260	0	0	0	169,975	7,430	101,875	68,100	8,587.5	13,125.0	0	0	0
79	GASAD	C4	250	67,812.5	1,425	337.5	15,077.5	811			8,437.5	5,202.5	0	0	0
80	POLY	C4	260	0	0	0	169,975	7,430	101,875	68,100	8,437.5	3,062.5	0	0	0
81	GASAD	C4	250	67,812.5	1,425	502.5	16,952.5	961			8,602.5	7,072.5	0	0	0
82	POLY	C4	260	0	0	0	169,975	7,430	101,875	68,100	8,602.5	15,325.0	0	0	0
87	GUARDRING	C4	250	0	0	585.0	55,600	4,048	55,600	0	8,685.0	55,725	67,812.5	1,425	0
88	POLY	C4	260	0	0	0	172,875	7,662	172,875	0	8,685.0	43,875	0	0	0
89	GUARDRING	C4	250	0	0	487.5	29,425	1,954	29,425	0	8,587.5	29,550	67,812.5	1,425	0
90	POLY	C4	260	0	0	0	172,875	7,662	172,875	0	8,587.5	23,550	0	0	0
91	GUARDRING	C4	250	0	0	337.5	14,300	794	14,300	0	8,437.5	14,425	67,812.5	1,425	0
92	POLY	C4	260	0	0	0	172,875	7,662	172,875	0	8,437.5	11,675	0	0	0
93	GUARDRING	C4	250	0	0	502.5	16,800	944	16,800	0	8,602.5	16,925	67,812.5	1,425	0
94	POLY	C4	260	0	0	0	172,875	7,662	172,875	0	8,602.5	26,675	0	0	0
VANDERPAUW															
96	GUARDRING	C4	250	0	0	585.0	31,350	2,108	31,350	0	8,685.0	31,325	67,225	1,378	0
97	GUARDRING	C4	250	0	0	487.5	16,025	882	16,025	0	8,587.5	16,025	67,225	1,378	0
98	GUARDRING	C4	250	0	0	337.5	11,650	532	11,650	0	8,437.5	11,650	67,225	1,378	0
99	GUARDRING	C4	250	0	0	502.5	17,900	1,032	17,900	0	8,602.5	17,900	67,225	1,378	0
101	GASAD	C4	250	67,075	1,366	585.0	31,387.5	2,111	30,087.5	1,300.0	8,685.0	31,387.5	0	0	0
102	GASAD	C4	250	67,075	1,366	487.5	16,062.5	885	14,875.0	1,187.5	8,587.5	16,062.5	0	0	0
103	GASAD	C4	250	67,075	1,366	337.5	11,687.5	535	10,775.0	912.5	8,437.5	11,687.5	0	0	0
104	GASAD	C4	250	67,075	1,366	502.5	17,937.5	1,035	16,762.5	1,175.0	8,602.5	17,937.5	0	0	0

\* All dimensions are in micrometers.

Table V—Pad key for METEST chip D\*

PAD NO.	GASAD			POLYCON			POLY							
	TUB			WINDOW			TAPS			TIES			CONTACTS	
	SIZE	NO.	AREA	SIZE	NO.	AREA*	W <sub>1</sub>	L <sub>1</sub>	SQ <sub>1</sub>	W <sub>2</sub>	L <sub>2</sub>	SQ <sub>2</sub>	W <sub>3</sub>	L <sub>3</sub>
1,2	20×78	4	6,240	1.5	6	13.5	25	187	7.48	5	50	10	11.5	272
3,4	20×36	2	1,440	3	1	9	25	125	5	5	153	30.6	13	52
4,5	20×36	1	720	3	1	9	25	125	5	5	189	37.8	13	26
5,6	20×36	4	2,880	3	1	9	25	125	5	5	81	16.2	13	104
7,26	20×70	4	5,600	2	4	16	25	125	5	5	746.5	149.3	12	240
7,8	20×70	7	9,800	2	4	16	25	125	5	5	1788.0	357.6	12	420
8,25	20×70	2	2,800	2	4	16	25	125	5	5	886.5	177.3	12	120
9,10	20×34	2	1,360	2	1	4	25	125	5	5	157	31.4	12	48
10,11	20×34	1	680	2	1	4	25	125	5	5	191	38.2	12	24
11,12	20×34	4	2,720	2	1	4	25	125	5	5	89	17.8	12	96
13,14	20×33	2	1,320	1.5	1	2.25	25	125	5	5	159	31.8	11.5	46
14,15	20×33	1	660	1.5	1	2.25	25	125	5	5	222	44.4	11.5	23
15,16	20×33	4	2,640	1.5	1	2.25	25	125	5	5	93	18.6	11.5	92
17,18	20×32	2	1,280	1.0	1	1	25	125	5	5	161	32.2	11	44
18,19	20×32	1	640	1.0	1	1	25	125	5	5	223	44.6	11	22
19,20	20×32	4	2,560	1.0	1	1	25	125	5	5	97	19.4	11	88
21,22	20×44	2	1,760	7	1	49	25	125	5	5	137	27.4	17	68
22,23	20×44	1	880	7	1	49	25	125	5	5	181	36.2	17	34
23,24	20×44	4	3,520	7	1	49	25	125	5	5	49	9.8	17	136
25,26	20×70	1	1,400	2	4	16	25	125	5	5	155	31	12	60
27,28	20×40	2	1,600	5	1	25	25	125	5	5	145	29	15	60
28,29	20×40	1	800	5	1	25	25	125	5	5	185	37	15	30
29,30	20×40	4	3,200	5	1	25	25	125	5	5	65	13	15	120
31,32	20×74	2	2,960	1.0	8	8	25	125	5	5	77	15.4	11	128
32,33	20×74	1	1,480	1.0	8	8	25	125	5	5	143.5	28.7	11	64
33,34	20×74	4	5,920	1.0	8	8	25	171	6.84	5	50	10	11	256
35,36	20×78	2	3,120	1.5	6	13.5	25	125	5	5	69	13.8	11.5	136
36,1	20×78	1	1,560	1.5	6	13.5	25	125	5	5	139.5	27.9	11.5	68

\* All dimensions are in micrometers.

† Cross-sectional area of tub input or output.

Table VI—Mask levels

Suggested Sequence	Mask Level	Tone	Note	Features	Background
1	GUARDRING	Normal	1	Clear	Opaque
2	GASAD	Normal	1	Clear	Opaque
3	POLYCON	Normal	1	Clear	Opaque
4	POLY	Normal	1	Opaque	Clear
5	WINDOW	Normal	1	Clear	Opaque
6	METAL	Normal	1	Opaque	Clear
$\frac{5}{2}$	R. T. GASAD	Reverse	2	Opaque	Clear
$\frac{4}{4}$	R. T. POLY	Reverse	2	Clear	Opaque
$\frac{6}{6}$	R. T. METAL	Reverse	2	Clear	Opaque

1. For use with positive photoresist.
2. For use with negative photoresist, uniform gold metallization, selective oxidation, or other special processes.

Table VII—Experimental devices

Device Structures		Required Mask Levels					
		GUARDRING	GASAD	POLYCON	POLY	WINDOW	METAL
MOS Capacitors			*		*		
Schottky Diodes:	Guarded	*	*		*		
	Unguarded		*		*		
Ohmic Contacts			*	*	*		
p-n Junctions			*	*	*		
IGFETs			*		*	*	*
Gated Diodes	GATODE		*		*	*	*
SADSHEET			*			*	*
POLYSHEET					*		
VANDERPAUW:	POLY				*		
	GASAD		*	*	*		
	GUARDRING	*		*	*		
Contact Metallization Test Cells	METEST		*	*	*		

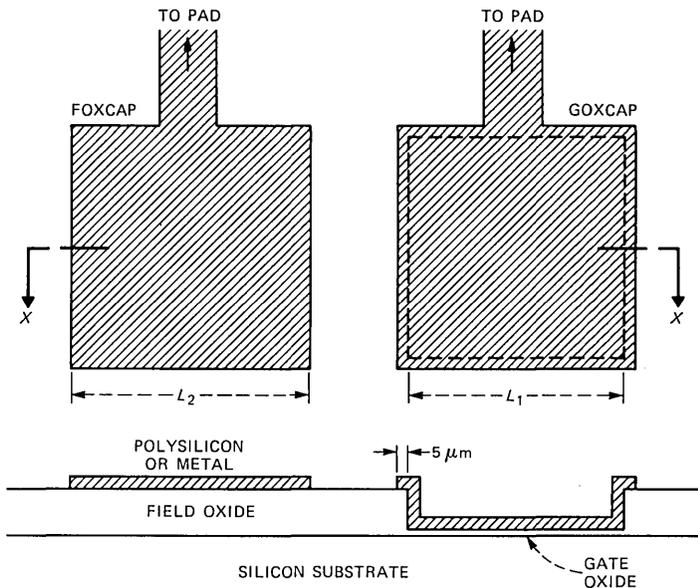


Fig. 1—FOXCAP and GOXCAP MOS capacitors (chips A through C).

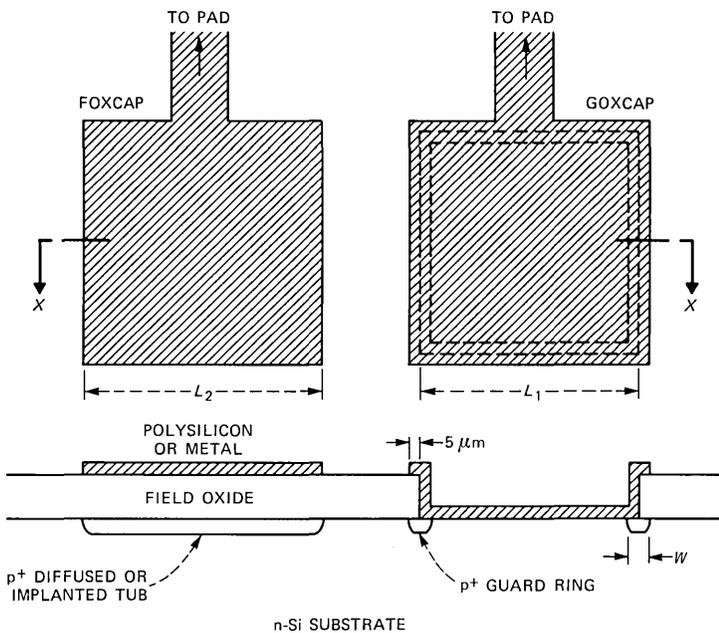


Fig. 2—Guarded Schottky diode and buried channel capacitor formed with FOXCAP and GOXCAP features (chips A through C).

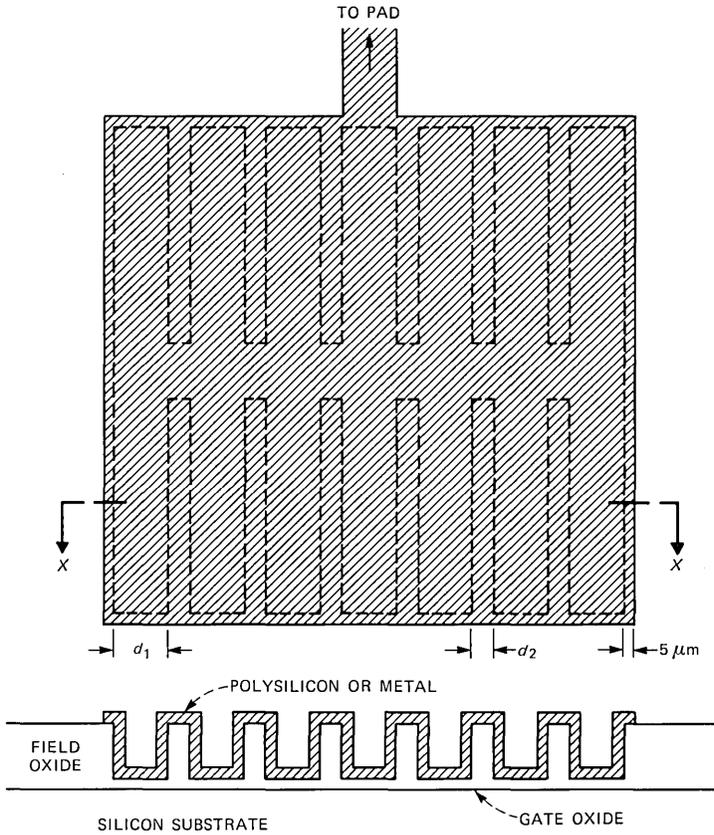


Fig. 3—GOXCOMB MOS capacitor (chips A through C).

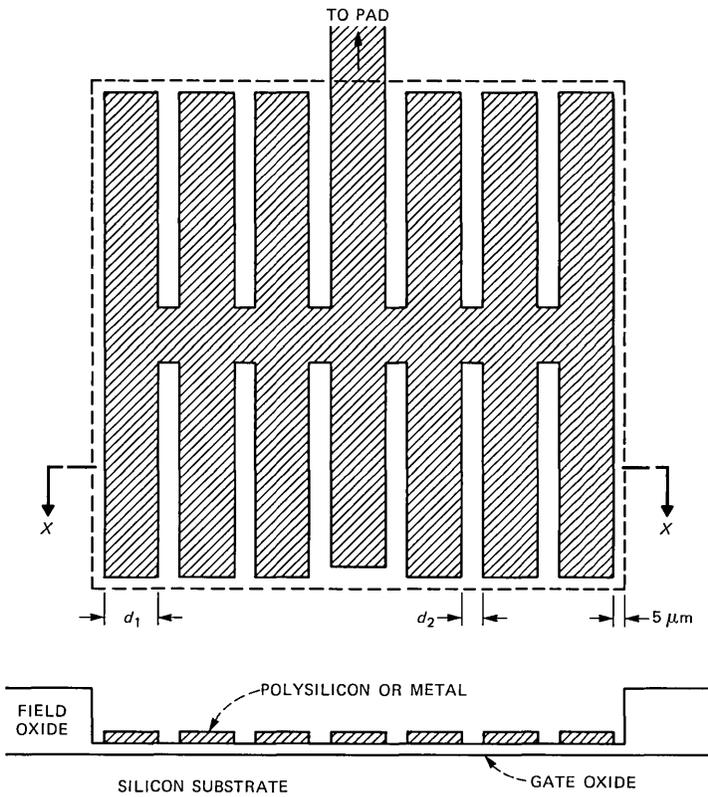


Fig. 4—POLYCOMB MOS capacitor (chips A through C).

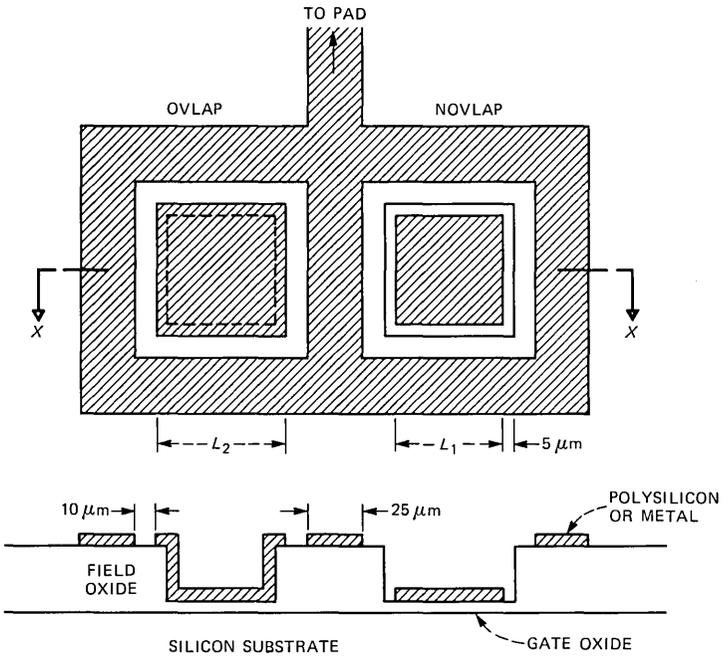


Fig. 5—OVLAP and NOVLAP MOS capacitors with field plate to control surrounding surface potential (chips A through C).

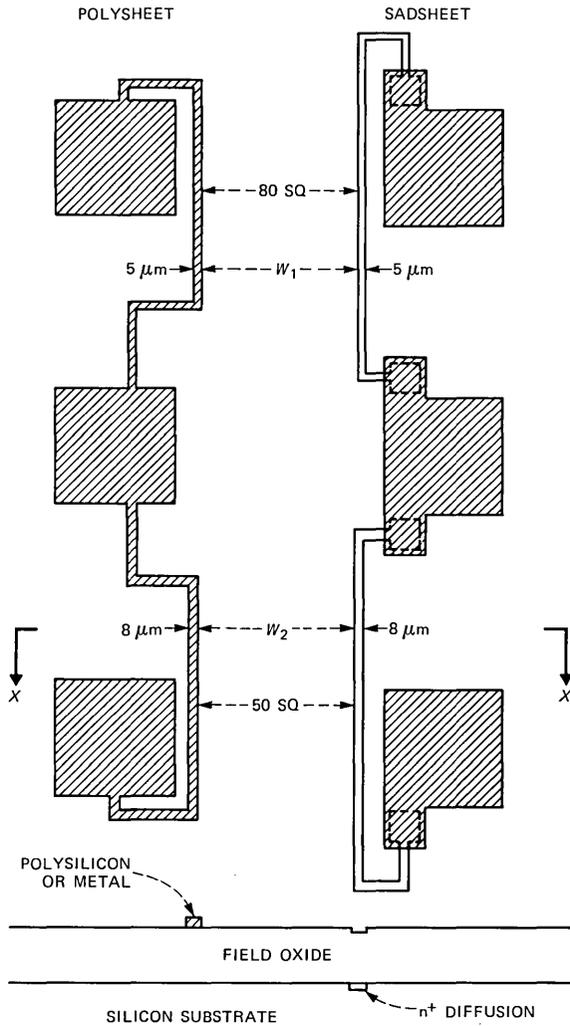


Fig. 6—POLYSHEET and SADSHEET sheet resistance and linewidth features (chip A).

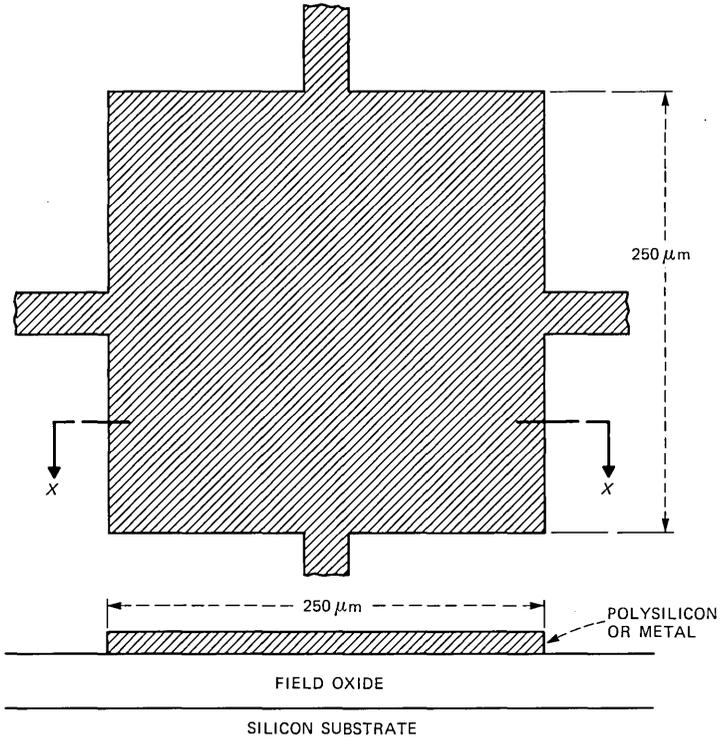


Fig. 7—Van der Pauw pattern (chip A and chip C4).

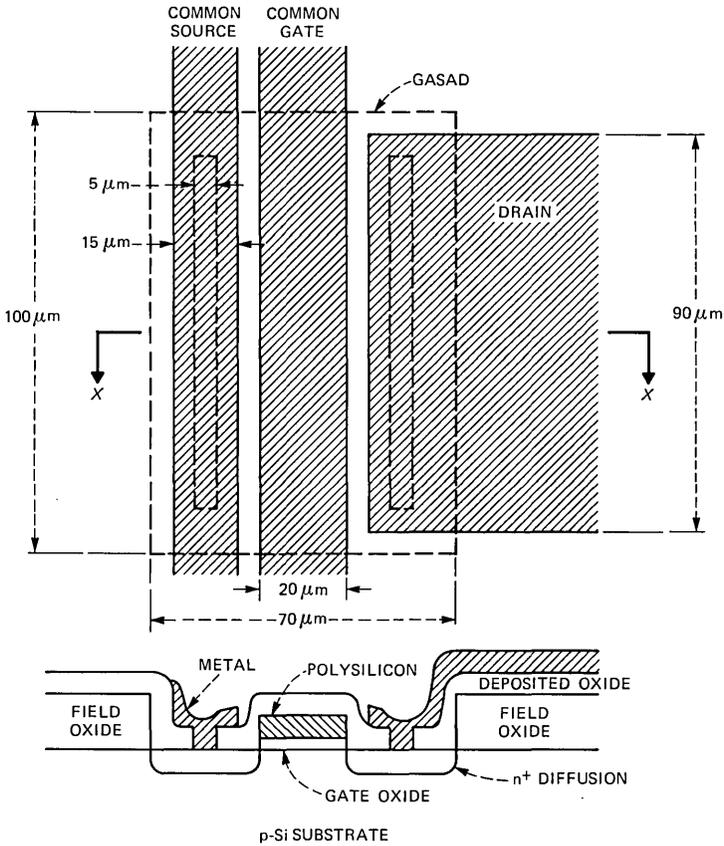


Fig. 8—Standard IGFET with common sources and drains (chip B).

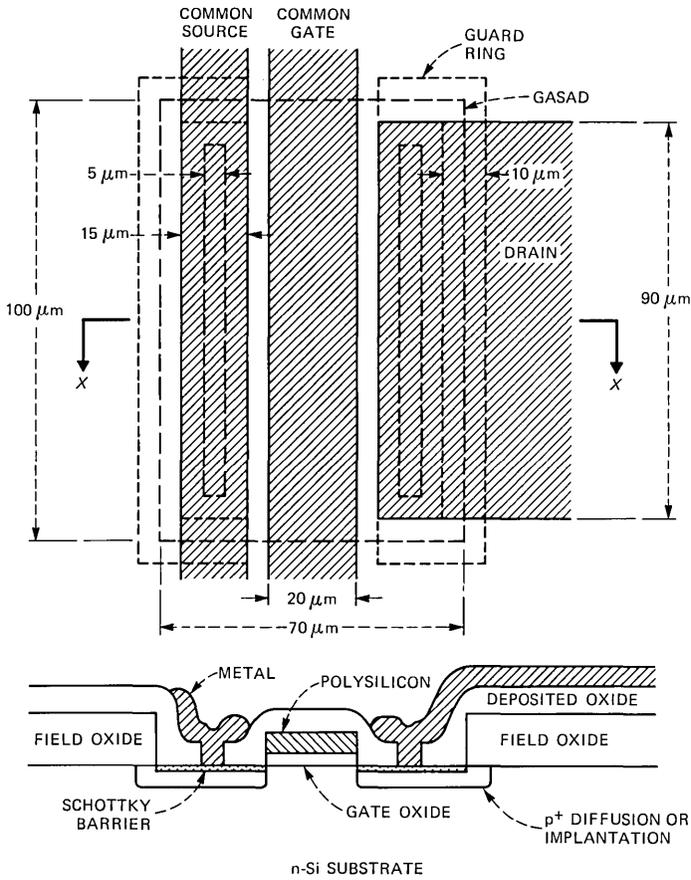


Fig. 9—Guarded IGFET with common sources and drains (chip B).

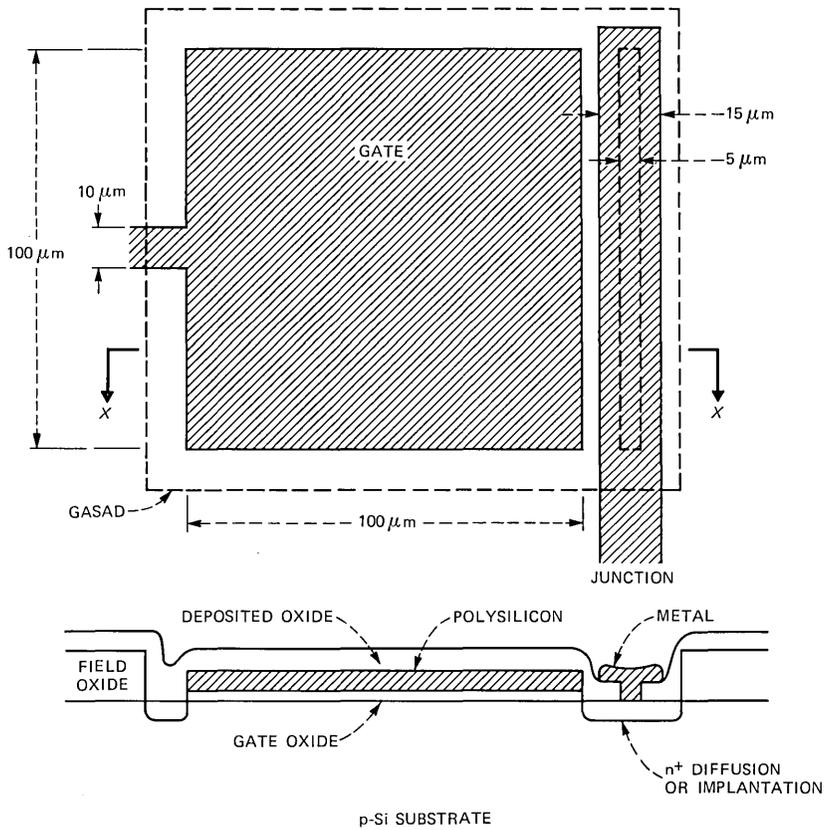


Fig. 10—Gated diode with  $n^+$  diffusion or implantation completely surrounding the gate (GATODE chip B).

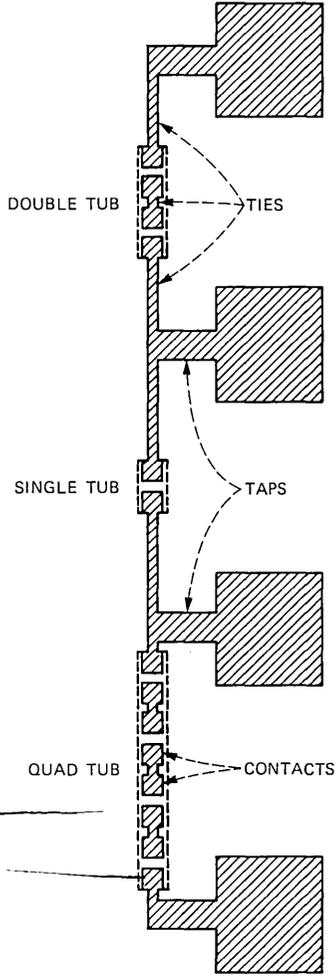


Fig. 11—Tapped string for the metallization test chip (METEST chip D).

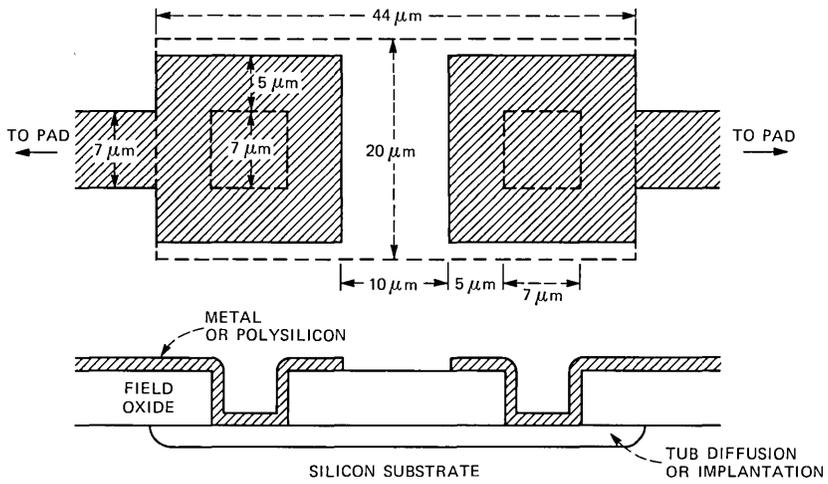


Fig. 12—Dual contact cell for the metallization test chip (METEST chip D).

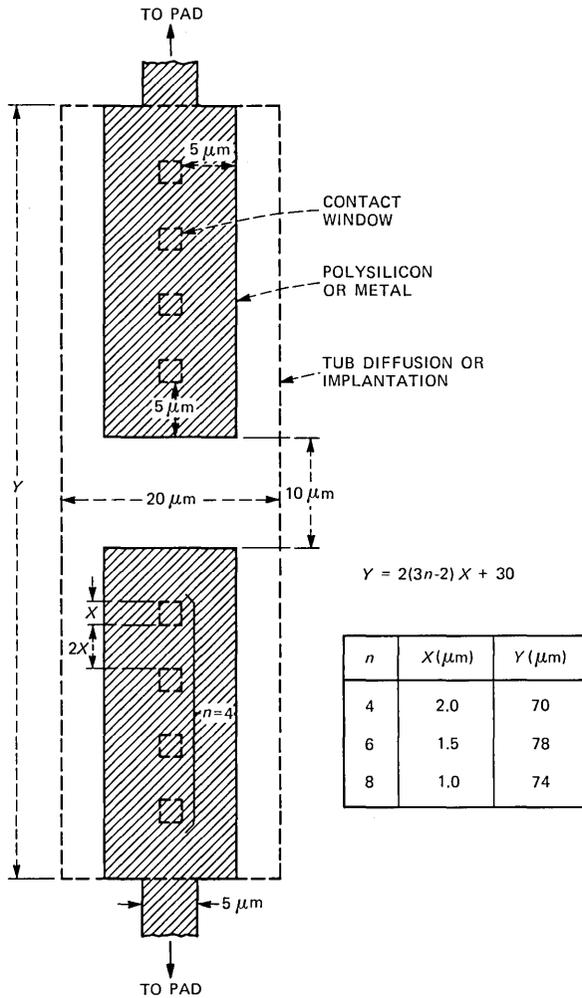


Fig. 13—Multiple contact cell for the metallization test chip (METEST chip D).

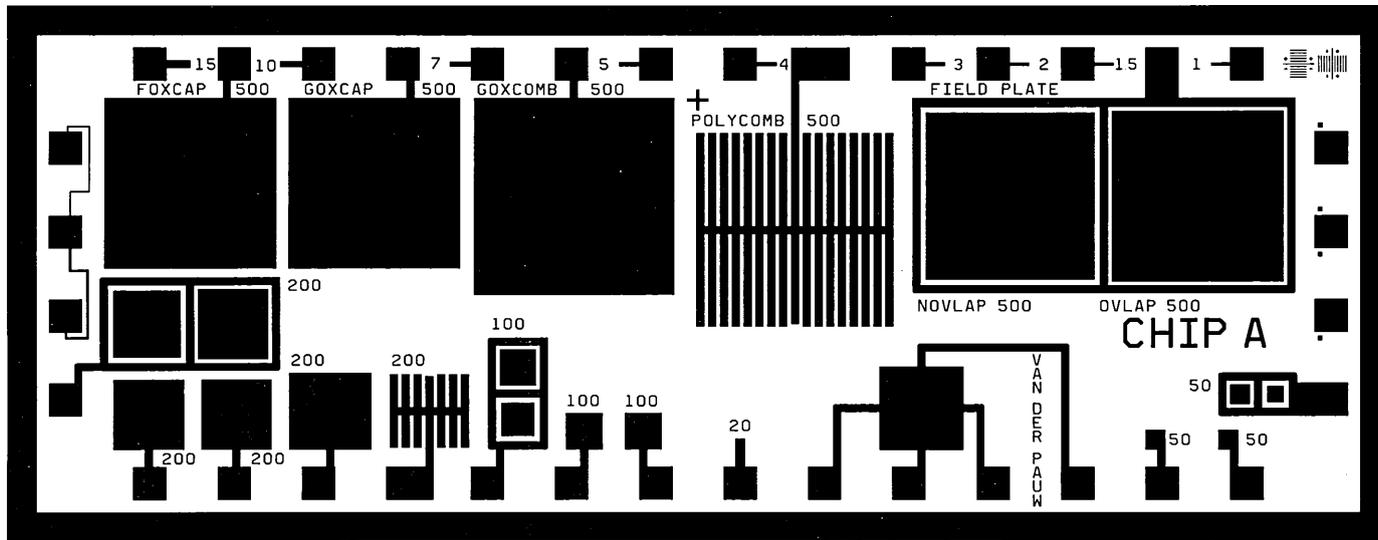


Fig. 14—Composite POLY and WINDOW levels for the A chip.

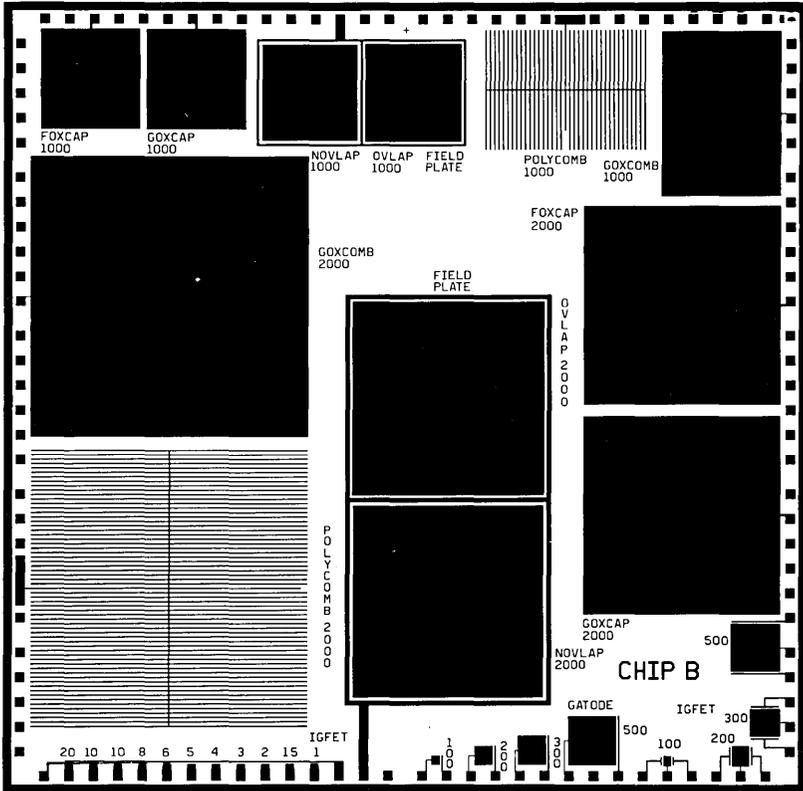


Fig. 15—Composite POLY, WINDOW, and METAL levels for the B chip. Some of the detail has been enlarged 3X to achieve adequate resolution for this illustration.

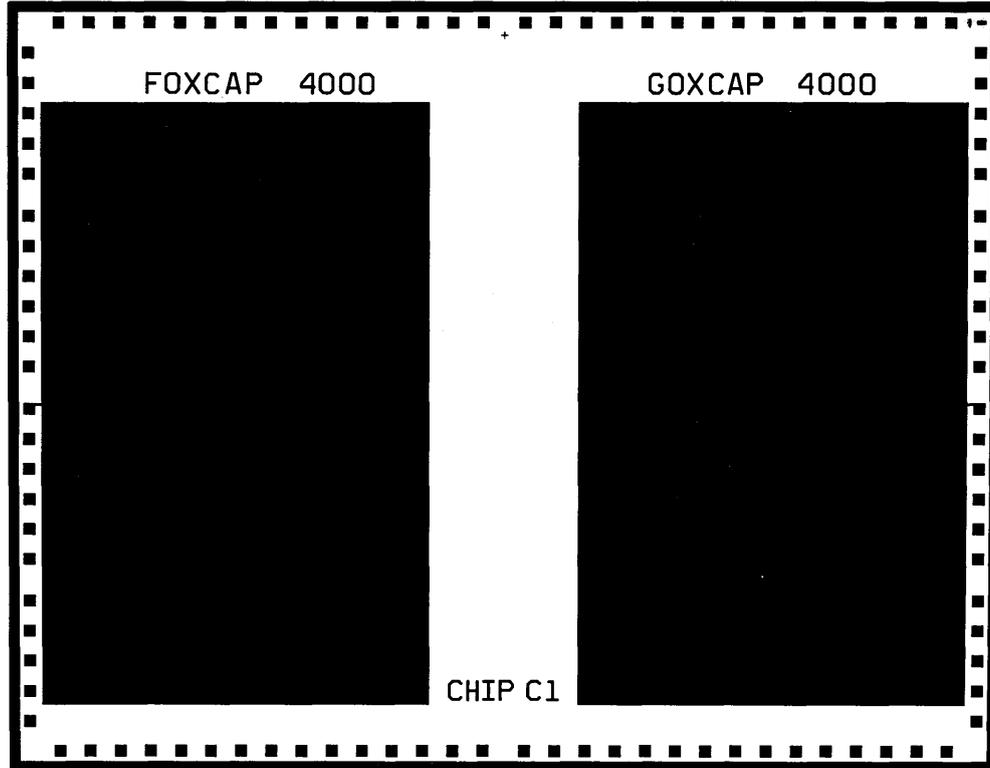


Fig. 16—Composite POLY and WINDOW levels for the C1 chip.

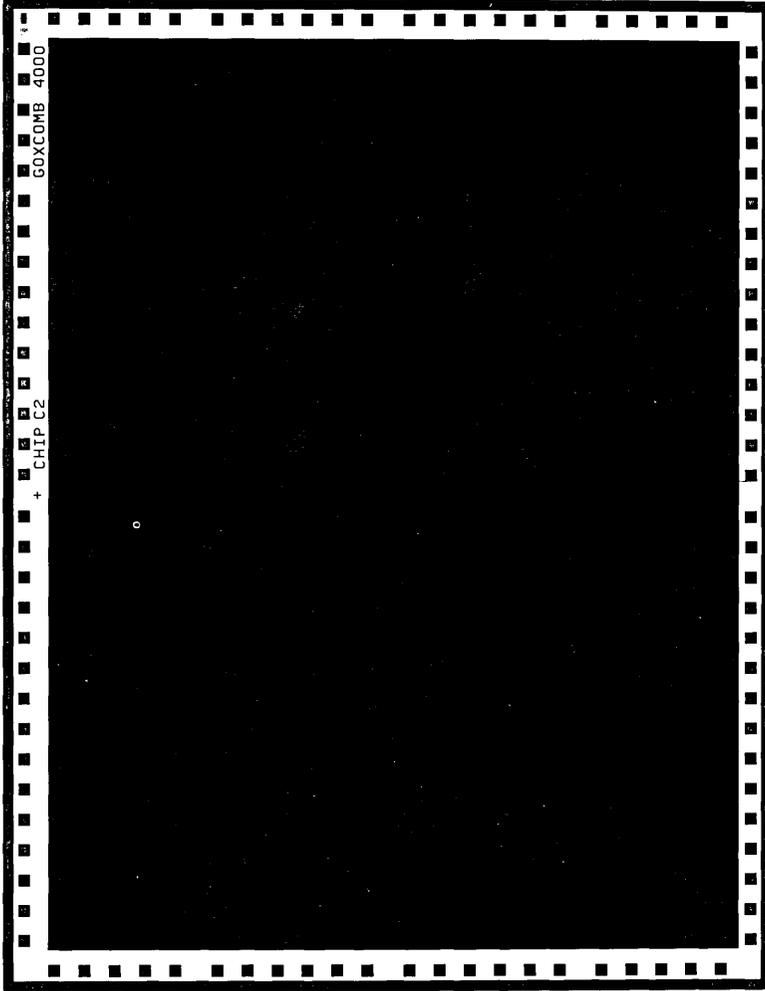


Fig. 17—Composite POLY and WINDOW levels for the C2 chip.

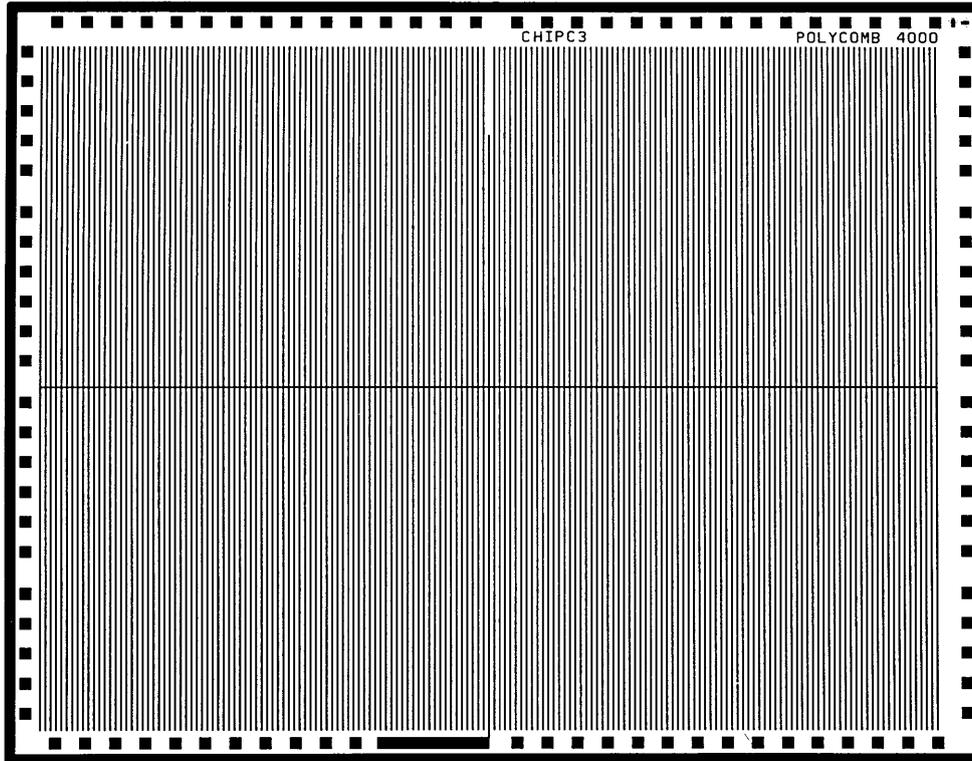


Fig. 18—Composite POLY and WINDOW levels for the C3 chip. Detail has been enlarged 3X to achieve adequate resolution for this illustration.

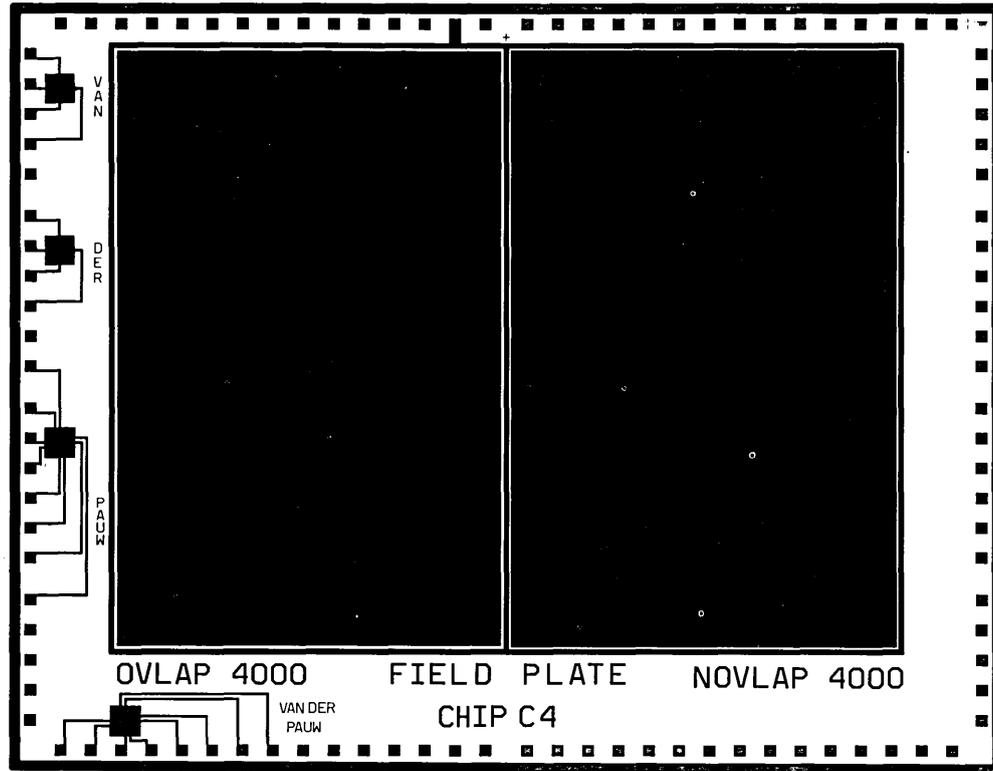


Fig. 19—Composite GUARDRING, GASAD, POLY, and WINDOW levels for the C4 chip. The frames surrounding OVLAP and NOVLAP have been enlarged 3X to achieve adequate resolution for this illustration.

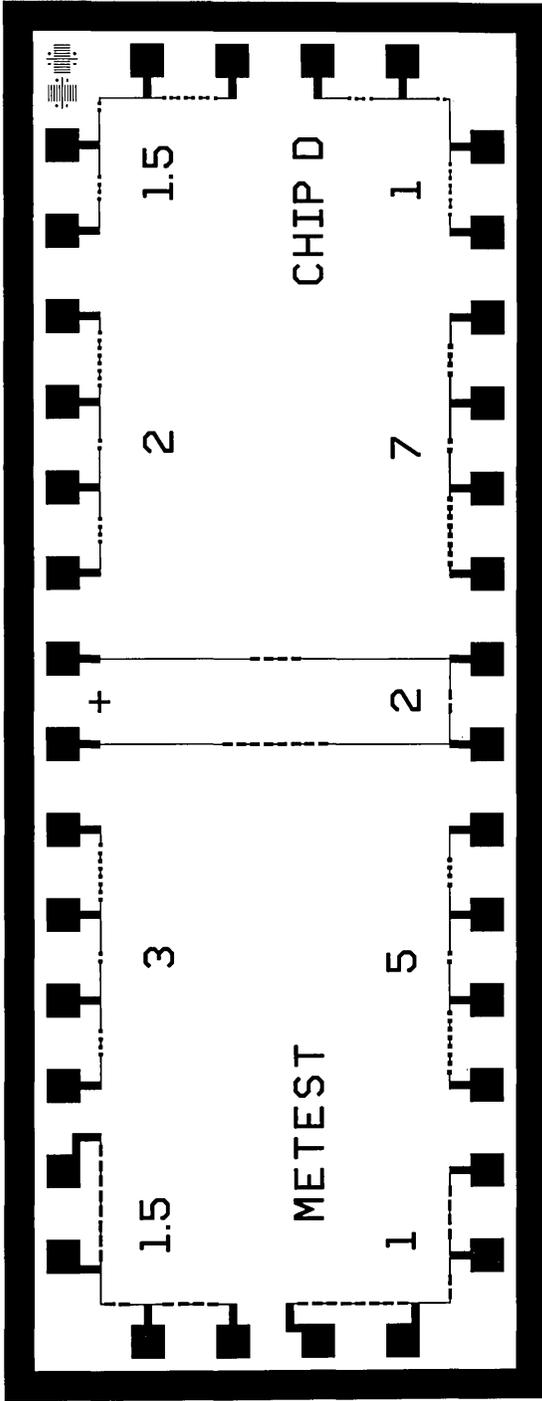


Fig. 20—Composite POLY and WINDOW levels for the D chip.

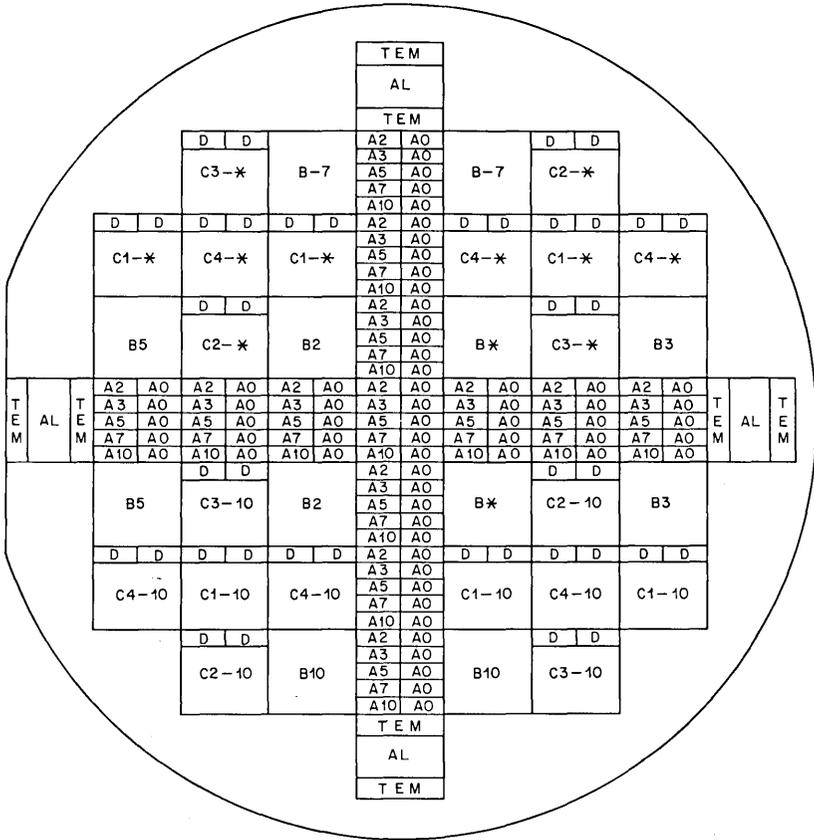


Fig. 21—Chip layout on the fine-line process development wafer.

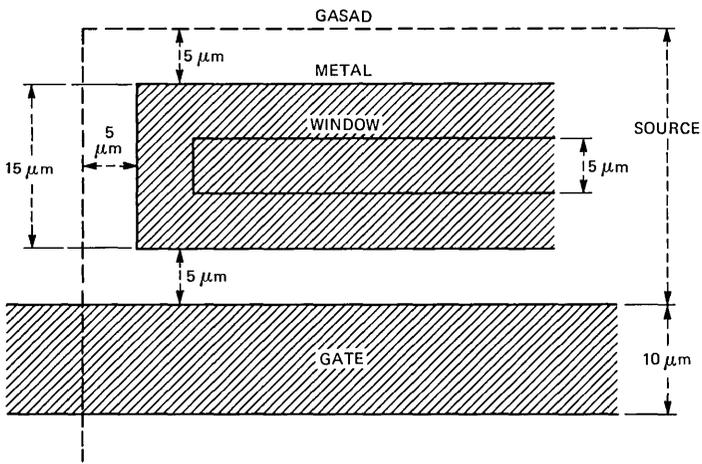


Fig. 22—IGFET design rules for fine-line process development system.

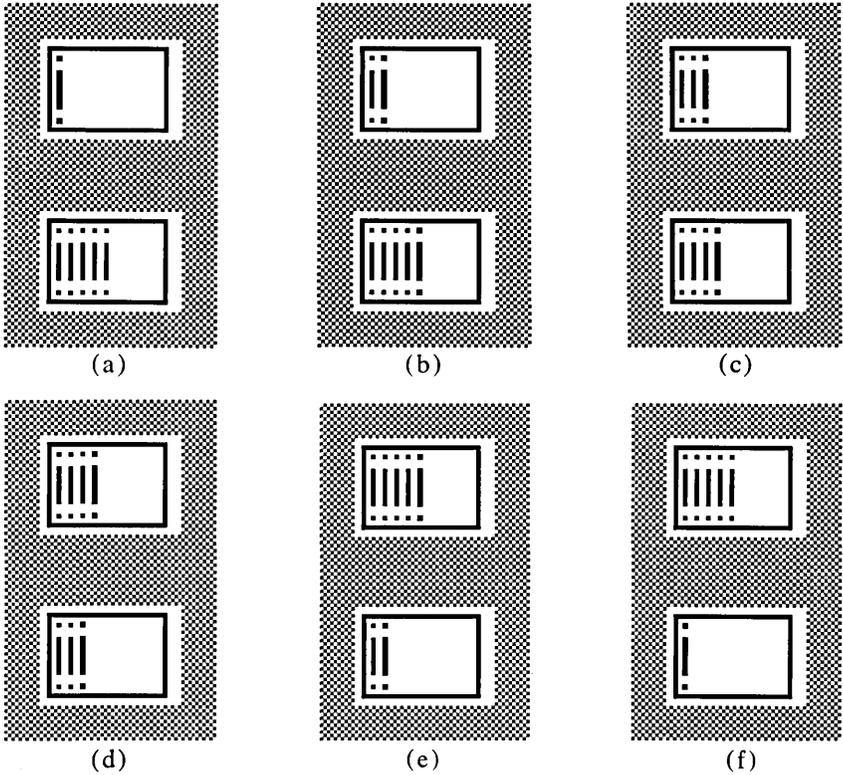


Fig. 23—MOPEP alignment features. (a) GUARDRING level. (b) GASAD level. (c) POLYCON level. (d) POLY level. (e) WINDOW level. (f) METAL level.

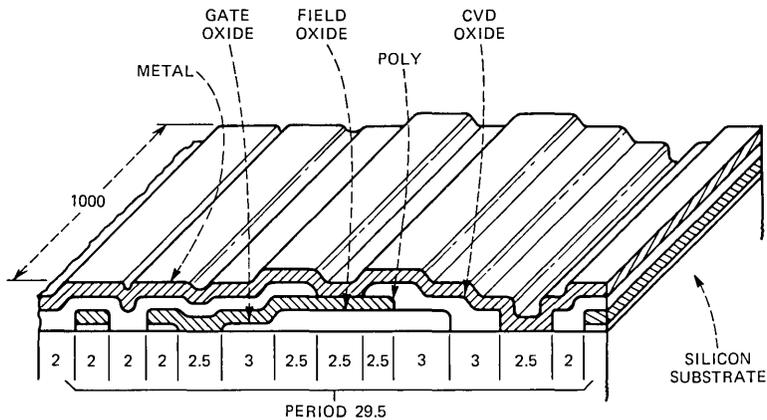


Fig. 24—TEM test chip.

## GLOSSARY

CHANSTOP	channel stopping implantation or diffusion to avoid inversion of the silicon surface at the Si-SiO <sub>2</sub> interface
C-V	capacitance measured as a function of voltage
FOXCAP	field oxide capacitor
GASAD	gate and source and drain feature delineated in the field oxide prior to gate oxidation. Also, the second photolithographic level in the set of fine-line process development masks.
GATODE	gated diode, essentially an IGFET (see below) with common source and drain.
GOXCAP	gate oxide capacitor
GOXCOMB	a gate oxide feature with a comb-shaped structure
GUARDRING	electrically guarded structure, fabricated by ion implantation or diffusion, which straddles and surrounds the boundary of a metallization feature, forming a closed ring.
HEXCAP	six-fold or hexadic capacitor group
IGFET	insulated gate field-effect transistor
LSI	large-scale integration
METAL	metallization pattern, the final photolithographic level in the set of fine-line process development masks.
METEST	metallization test structure consisting of tapped strings with contacts to underlying diffused tubs.
MOPEP	modified Perkin Elmer projection alignment features
MOS	metal-oxide-semiconductor sandwich structure used

	for electrical characterization of device fabrication processes
NOVLAP	conductive pad not overlapping field oxide and forming the top level of a metal-oxide-semiconductor capacitor
OVLAP	conductive pad overlapping field oxide and forming the top level of a metal-oxide-semiconductor capacitor
PEP	Perkin Elmer projection alignment features
POLY	polycrystalline silicon which, when patterned, forms a conductive electrode for electrical tests. Also, an intermediate photolithographic mask level in the set of fine-line process development masks.
POLYCOMB	a polycrystalline silicon feature with a comb-shaped structure
POLYCON	polycrystalline contact to underlying silicon substrate. Also, an intermediate photolithographic mask level in the set of fine-line process development masks.
POLYSHEET	polycrystalline silicon feature for sheet resistance and linewidth loss measurements
SADSHEET	structure formed during gate and source and drain (GASAD) lithography to determine source and drain sheet resistance and linewidth loss
SEM	scanning electron microscope
TEM	transmission electron microscope
VANDER-PAUW	a symmetric structure introduced by L. J. van der Pauw <sup>10</sup> to determine the electrical resistivity of thin conductive layers
WINDOW	next to the last photolithographic mask level in the set of fine-line process development masks to form source and drain contacts to insulated gate field-effect transistors, junction contacts in gated diodes, and access to polycrystalline silicon features when the poly Si is covered by an intermediate dielectric.

## REFERENCES

1. H. C. Poon and R. L. Johnston, unpublished work.
2. J. M. Andrews, *J. Electron. Mater.*, **8** (1979), pp. 227-47.
3. R. A. Kushner, private communication.
4. D. M. Boulin, unpublished.
5. A. M. Mohsen and F. J. Morris, *Solid-St. Electron.*, **18** (1975), pp. 407-16.
6. J. R. Brews and D. Kahng, unpublished work; U.S. Patent No. 4,135,289, applied for August 23, 1977, issued January 23, 1979.
7. B. V. Gokhale and J. Zeidenbergs, *Technical Digest of the 1974 International Electron Devices Meeting*, Abs. No. 23.1, p. 529.
8. G. W. Taylor, *Solid-St. Electron.*, **19** (1976), pp. 495-503.
9. John Andrews, E. F. Labuda, and Paul Norton, *The Electrochemical Society Extended Abstracts, Fall Meeting, Pittsburgh, PA, Oct. 15-20, 1978, Abstract 203*, p. 543.

10. L. J. van der Pauw, *Philips Res. Repts.*, *13* (1958), pp. 1-9.
11. B. R. Chawla, A. P. Gnadinger, and W. Rosenzweig, unpublished work.
12. R. F. Pierret, *Solid-St. Electron.*, *17* (1974), pp. 1257-69.
13. P. L. Castro and B. E. Deal, *J. Electrochem. Soc.*, *118* (1971), pp. 280-86.
14. D. J. Fitzgerald and A. S. Grove, *Proc. IEEE*, *54* (1966), pp. 1601-2.
15. E. H. Snow, A. S. Grove, and D. J. Fitzgerald, *Proc. IEEE*, *55* (1967), pp. 1168-85.
16. A. S. Grove and D. J. Fitzgerald, *Solid-St. Electron.*, *9* (1966), pp. 783-806.
17. D. J. Fitzgerald and A. S. Grove, *Surface Sci.*, *9* (1968), pp. 347-69.
18. G. W. Taylor, private communication.
19. J. R. Brews, private communication.
20. S. Vaidya, D. B. Fraser, and W. S. Lindenberger, *J. Appl. Phys.*, *51* (1980), pp. 4475-82.
21. T. T. Sheng and R. B. Marcus, *J. Electrochem. Soc.*, *127* (1980), pp. 737-43.
22. S. Vaidya, A. K. Sinha, and J. M. Andrews, *J. Electrochem. Soc.*, *130* (1983), pp. 496-501.
23. J. M. Andrews, P. A. Heimann, and R. A. Kushner, *Proc. First Int. Symp. on Very Large Scale Integration Science and Technology*, Detroit, MI, October 17-21, 1982, Abstract 147, pp. 68-75.
24. C. J. Koeneke, S. M. Sze, R. M. Levine, and E. Kinsbron, *IEDM Tech. Digest* (1981), pp. 367-70.
25. P. A. Heimann, P. S. D. Lin, and T. T. Sheng, *Electrochem. Soc. Extended Abstr., Fall Meeting*, Detroit, MI, October 17-21, 1982, Abstract 195, pp. 309-10.
26. P. A. Heimann and J. E. Olsen, *J. Appl. Phys.* *53* (1982), pp. 546-9.

## A Circuit That Changes the Word Rate of Pulse Code Modulated Signals

By J. C. CANDY and O. J. BENJAMIN

(Manuscript received November 2, 1982)

*In this paper we describe a circuit that accepts pulse code modulated signals sampled at about 8 kHz and resamples them at any desired rate up to 512 kHz. When the sampling satisfies Nyquist's criterion, the distortion introduced is at least 35 dB below the signal level. The circuit uses a digital low-pass filter to interpolate sample values, and it may be integrated as about 2500 gates on a 5 mm<sup>2</sup> chip.*

### I. INTRODUCTION

It often is impractical to synchronize all of the clocks of an extensive digital network. Consequently, data will arrive at connections out of synchronism and special circuits are needed to bring them into time with the local clock. For irregular bursts of data, synchronism can be easily obtained using buffer memories, but for continuous streams of data, such methods are useful only for very small discrepancies in clock frequencies. A case of particular interest in telephone networks is the transmission and processing of pulse code modulation (PCM). Changing the word rate of such data can introduce objectionable noise into the signal. We describe a circuit that uses digital filters to contain this noise.

### II. RESAMPLING

We know that when an analog signal,  $x(t)$ , having spectral density  $X(\omega)$  and bandwidth  $\omega_0$  is sampled at regular intervals,  $\tau$ , the spectral density of the sampled signal can be expressed as the sum of images

$$X'(\omega) = \sum_{-\infty}^{\infty} X\left(\omega + \frac{2\pi n}{\tau}\right). \quad (1)$$

When  $\omega_0\tau < \pi$ , the original signal can be recovered by filtering out the images for  $n \neq 0$ .

If we were to change the sampling rate<sup>1</sup> by first holding each sample value constant throughout its period and resampling at the new period,  $\tau_1 < \pi/\omega_0$ , the spectral density of the resampled signal can be expressed as

$$X''(\omega) = \sum_n \sum_k X \left( \omega + \frac{2\pi n}{\tau} + \frac{2\pi k}{\tau_1} \right) H \left( \omega + \frac{2\pi n}{\tau} \right), \quad (2)$$

where

$$H(\omega) = \frac{\sin \left( \frac{\omega\tau}{2} \right)}{\frac{\omega\tau}{2}} = \text{sinc} (f\tau). \quad (3)$$

When we try to recover the original signal from  $X''(\omega)$  by filtering out the components for  $n \neq k \neq 0$ , it is contaminated by cross modulation products for which

$$0 < \left( \frac{n}{\tau} + \frac{k}{\tau_1} \right) < f_0 = \frac{\omega_0}{2\pi}. \quad (4)$$

Notice that when  $\tau$  and  $\tau_1$  are integer multiples of one another, this condition never holds and reflected noise is absent from the baseband.

An obvious means of eliminating in-band cross-modulation products from eq. (2) is to smooth out the high-frequency components of the sampled signal  $x'(t)$  before resampling it. This has been accomplished by replacing the sample and hold represented by  $H(\omega)$  in eq. (3) by a better low-pass filter that interpolates new sample values from the old ones.<sup>1,2</sup> Another implementation of this method is demodulation with a digital/analog (D/A) converter, analog smoothing, and remodulation with an analog/digital (A/D) converter. The method<sup>1</sup> presented here raises the sample rate to a high multiple of the original rate, smooths the sample values with digital filters, holds the smoothed samples, and resamples them at the desired rate.

### III. A CIRCUIT FOR RESAMPLING 8-kHz PCM

We will describe a circuit for resampling a 3.5-kHz signal that has been pulse code modulated at a nominal 8-kHz rate using 16 bits per word. The new sampling can be at any rate up to 512 kHz and even higher rates can be accommodated by minor modification of the circuits. The technique first raises the sampling rate 16 times to 128 kHz using digital interpolating filters to smooth out all unnecessary images of the signal, leaving only those that are adjacent to the new

rate and its harmonics. The high-frequency code is placed in a holding register from which the output is gated at the desired rate. We will see that the filter action of this hold,  $\text{sinc}(f/128)$ , provides more than 30-dB attenuation of the cross-modulation products that fold into base-band.

Because the complexity of digital filters increases sharply with increasing sampling rate, it pays to raise the rate in stages.<sup>1</sup> Figure 1 illustrates the process. Our first stage raises the sampling rate four times from 8 to 32 kHz, employing a low-pass filter that cuts off sharply to attenuate spectral images between 4 and 28 kHz. The second stage raises the rate to 128 kHz by simple linear interpolation, and the third stage is a holding register. Most of the circuits used originally were

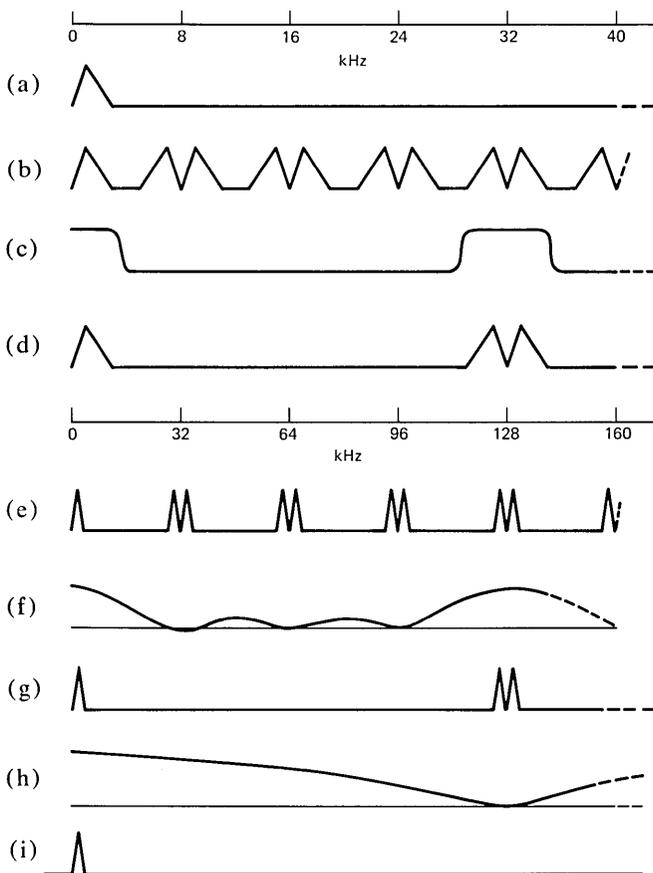


Fig. 1—An illustration of the signal's spectrum at various stages of the conversion: (a) The original signal. (b) The signal sampled at 8 kHz. (c) The response of the low-pass filter. (d) and (e) The output from the low-pass filter. (f) The response of the linear interpolator. (g) The output of the interpolator. (h) The response of the holding circuit. (i) The held signal.

designed to be part of an oversampled coder/decoder (codec).<sup>3</sup> We shall summarize their relevant properties in the following sections.

#### IV. THE LOW-PASS FILTER

The low-pass filter shown in Fig. 2 processes data at 32 kilowords per second. Its  $z$ -transform response is given by

$$F(z) = \frac{1}{32} \frac{1 - z^{-4}}{1 - z^{-1}} \cdot \frac{1 - \frac{5}{4}z^{-1} + z^{-2}}{1 - \frac{19}{16}z^{-1} + \frac{31}{64}z^{-2}} \cdot \frac{1 - \frac{3}{4}z^{-1} + z^{-2}}{1 - \frac{23}{16}z^{-1} + \frac{55}{64}z^{-2}} \quad (5)$$

Each word of its 8-kHz input signal is repeated four times and fed into two second-order sections. This filter attenuates the images of the signal in the range 4 to 28 kHz by more than 34 dB. Its output is a good approximation of pulse code modulation at 32 kHz. The spectral response of the entire resampler is shown in Fig. 3. The zeros at 4.5 and 6 kHz are introduced by the second-order sections, and those at 8 and 16 kHz by the repetition of input words.

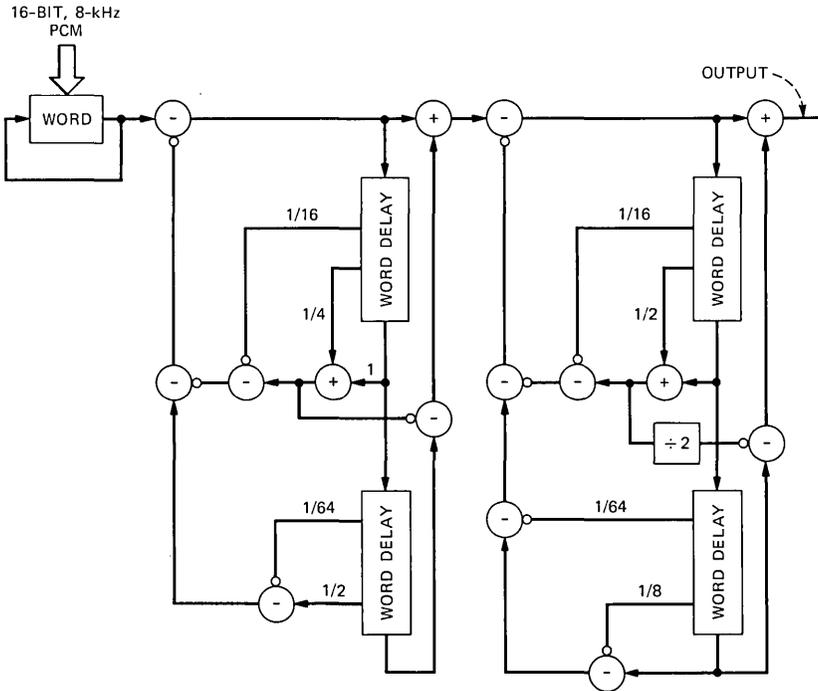


Fig. 2—An outline of the low-pass filter, clocked at 32 kilowords per second, that is used to raise the sampling rate from 8 to 32 kHz.

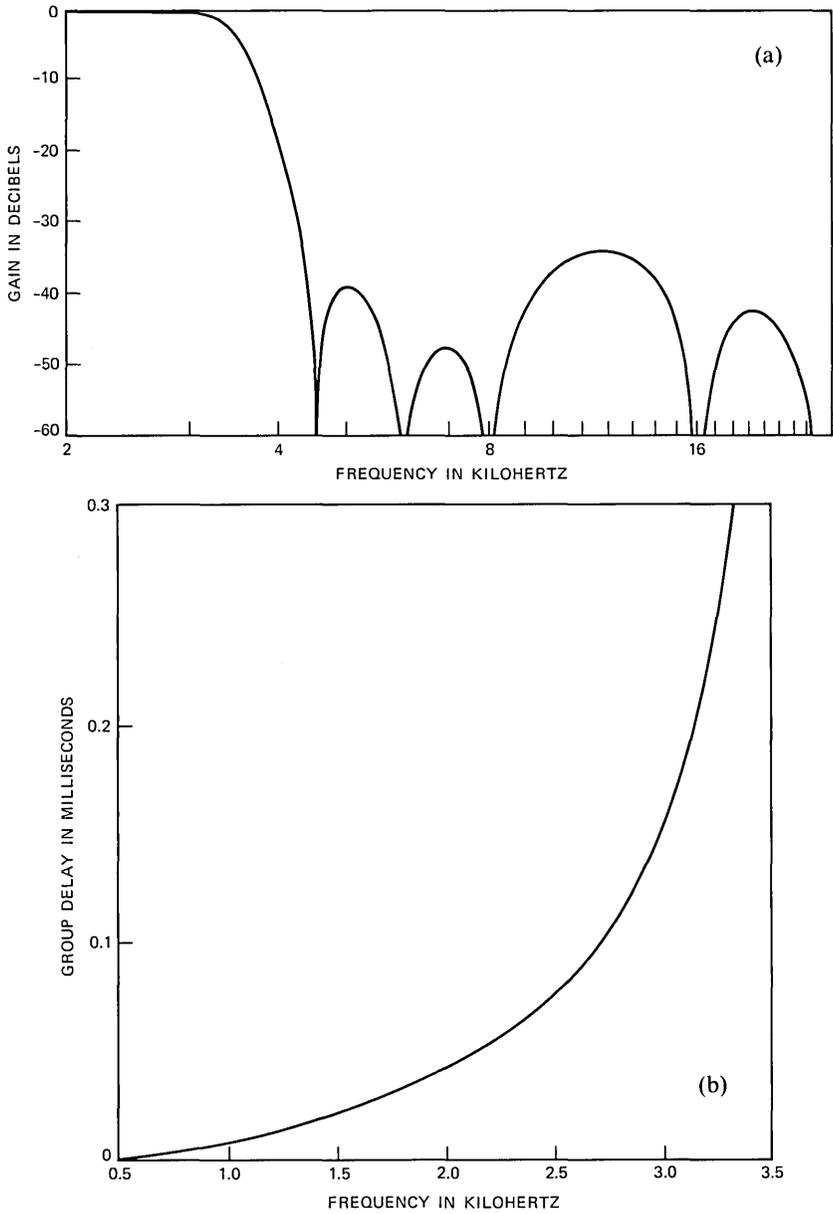


Fig. 3—Spectral response of the resampler. (a) The calculated gain of the cascaded filters used in the resampler. (b) The group delay of the filters.

## V. LINEAR INTERPOLATION

Simple linear interpolation of three sample values increases the sampling rate from 32 to 128 kHz. This process has frequency responses that can be expressed as

$$I(\omega) = \left( \frac{\text{sinc}(f/32)}{\text{sinc}(f/128)} \right)^2 \quad (6)$$

Its attenuation of spectral images in the range 28 to 36, 60 to 68, and 92 to 100 kHz exceeds 40 dB. The small amount of droop introduced into baseband is compensated in the low-pass filter so that the entire circuit has inband gain in the range  $-0.41$  to  $-0.57$  dB. The circuit implementation shown as Fig. 4 is based on the results

$$y(n\tau) = y[(n-1)\tau] + \frac{1}{4} \Delta x(n\tau)$$

$$\Delta x(n\tau) = x(4n\tau) - x[4(n-1)\tau]$$

and

$$y(4n\tau) = x(4n\tau). \quad (7)$$

After each new input sample enters register  $R_1$ , the output, held in register  $R_2$ , increments four times to make its value equal to the input.

## VI. RESAMPLING

Figure 5 shows the circuit that is used to resample the signal at the desired output rate without causing conflict with the internal clock. Here the output from the linear interpolator is loaded into register  $R_3$  in time with the internal clock running at 1 MHz. This loading is inhibited by the presence of the output clock, which after a short delay loads  $R_4$  from  $R_3$ . The frequency response associated with the holding action of this circuit may be expressed as

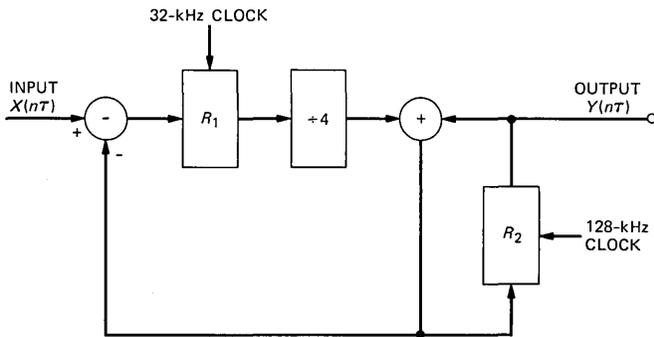


Fig. 4—An outline of the linear interpolator used to raise the sampling rate from 32 to 128 kHz.

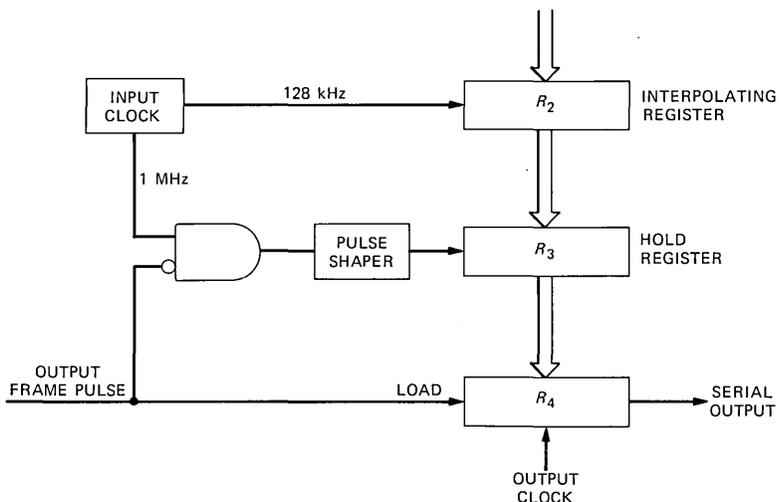


Fig. 5—The circuit used to sample the output. Register  $R_2$  is part of the linear interpolator,  $R_3$  is loaded from  $R_2$  at 1 MHz. An output demand inhibits loading of  $R_3$ , and then loads  $R_4$  from  $R_3$ .

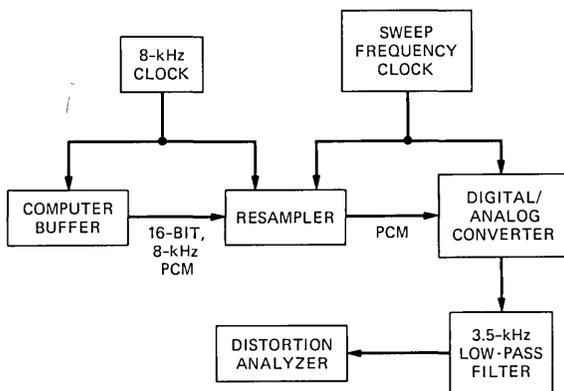


Fig. 6—The test circuit.

$$R(\omega) = \text{sinc}(f/128).$$

Images of the signal in the range 124 to 132 kHz are thereby attenuated by more than 30 dB.

## VII. TESTING THE CIRCUIT

A version of the circuit was built of standard digital circuit components and tested in the setup shown in Fig. 6. The net gain of the circuit and its signal-to-noise ratio were measured as the sampling rate was varied continuously from 0 to 256 kHz. For a 1.02-kHz input signal

sampled at 8 kHz, the gain remained constant within  $\pm 0.05$  dB and the signal-to-noise ratio always exceeded 40 dB. Input amplitudes were varied in the range 0 to  $-60$  dB. At lower amplitudes quantization noise inherent in the 16-bit input word was significant.

#### VIII. CONCLUDING REMARKS

Experience obtained while designing filters for a version of a codec enables us to estimate that the circuits described here can be implemented on about  $5 \text{ mm}^2$  of silicon in a standard technology. The performance of the circuit is good enough that imperfections introduced by resampling would be insignificant compared with those normally obtained from  $\mu$ -255 encoding of the signal.

#### IX. ACKNOWLEDGMENT

We thank H. S. McDonald for suggesting this use for our circuits, and J. L. Henry for providing information on the integrated circuit implementation.

#### REFERENCES

1. D. J. Goodman and M. J. Carey, "Nine Digital Filters for Decimation and Interpolation," *IEEE Trans. Acoust., Speech, Signal Processing, ASSP-25*, 2(April 1977), pp. 121-6.
2. R. E. Crochiere and L. R. Rabiner, "Interpolation and Decimation of Digital Signals—A Tutorial Review," 53 references, *Proc. IEEE*, 69, pp. 300-31.
3. J. C. Candy, B. A. Wooley, and O. J. Benjamin, "A Voiceband Codec with Digital Filtering," *IEEE Trans. Commun., COM-29* (June 1981), pp. 815-30.

## CONTRIBUTORS TO THIS ISSUE

**John Andrews**, B.S., 1958, Houghton College; Ph.D. (Solid State Physics), 1963, Massachusetts Institute of Technology; Research Associate, Massachusetts Institute of Technology, 1963–1966; Bell Laboratories, 1966—. The subject of Mr. Andrews' doctoral thesis was ultrasonic attenuation in superconductors at radio and microwave frequencies. After coming to Bell Laboratories, he worked on laser mixing in nonlinear optical crystals. Next, he investigated ohmic and rectifying metal-semiconductor contacts formed by metallurgical reaction between transition metals and silicon. From 1970 to 1976 he worked with Western Electric engineers at Allentown on silicon integrated circuit manufacture. Since 1976 he has been associated with the Advanced LSI Development Laboratory at Murray Hill, investigating new materials and special processes for fine-line VLSI technology. Senior Member, IEEE; member APS, ECS, and TMS-AIME; associate editor, *Materials Letters*.

**O'Connell J. Benjamin**, B.S.E.E., 1976, Polytechnic Institute of Brooklyn; graduate studies, 1977–1982, Polytechnic Institute of New York; Bell Laboratories, 1973—. Mr. Benjamin has been involved with research in digital MF receivers, voiceband codecs, and, presently, he is working on computer communications.

**G. Samuel Brockway**, B.S.C.E., 1966, University of Miami; M.S.E.M., 1968, Georgia Institute of Technology; Ph.D. (Applied Mechanics), 1972, California Institute of Technology; Bell Laboratories, 1976–1981. Mr. Brockway has worked extensively in such areas of mechanics as elasticity, viscoelasticity, and fracture mechanics, and in the development of plastics for use in fiber optics. He is presently president of Plastics Engineering Consultants.

**James C. Candy**, B.Sc. and Ph.D. (Engineering), University of Wales, Bangor, in 1951 and 1954, respectively; Bell Laboratories, 1961—. From 1954 to 1956 Mr. Candy was with S. Smith and Sons, Guided Weapons Department, Cheltenham, and for the next four years he worked on nuclear instrumentation at the Atomic Energy Research Establishment, Harwell. In 1959 Mr. Candy went to the United States to take up an appointment as a Research Associate at the University of Minnesota, and a year later he joined Bell Laboratories, Holmdel, NJ, where he has investigated digital circuits and the efficient transmission of video signals. He is now concerned with techniques for converting signals between analog and digital formats.

**Stephen E. Levinson**, B.A. (Engineering Sciences), 1966, Harvard; M.S. and Ph.D. (Electrical Engineering), University of Rhode Island, Kingston, Rhode Island, 1972 and 1974, respectively; General Dynamics, 1966-1969; Yale University, 1974-1976; Bell Laboratories, 1976—. From 1966 to 1969, Mr. Levinson was a design engineer at Electric Boat Division of General Dynamics in Groton, Connecticut. From 1974 to 1976, he held a J. Willard Gibbs Instructorship in Computer Science at Yale University. In 1976, he joined the technical staff at Bell Laboratories, where he is pursuing research in the areas of speech recognition and cybernetics. Member, Association for Computing Machinery, Acoustical Society of America, editorial board of *Speech Technology*; associate editor, *IEEE Transactions on Acoustics, Speech and Signal Processing*.

**Lawrence R. Rabiner**, S.B. and S.M., 1964, Ph.D. (Electrical Engineering), The Massachusetts Institute of Technology; Bell Laboratories, 1962—. From 1962 through 1964, Mr. Rabiner participated in the cooperative plan in electrical engineering at Bell Laboratories. He worked on digital circuitry, military communications problems, and problems in binaural hearing. Presently, he is engaged in research on speech communications and digital signal processing techniques. He is coauthor of *Theory and Application of Digital Signal Processing* (Prentice-Hall, 1975) and *Digital Processing of Speech Signals* (Prentice-Hall, 1978). Former President, IEEE, ASSP Society; former Associate Editor, ASSP Transactions; former member, Technical Committee on Speech Communication of the Acoustical Society, ASSP Technical Committee on Speech Communication; Member, IEEE Proceedings Editorial Board, Eta Kappa Nu, Sigma Xi, Tau Beta Pi. Fellow, Acoustical Society of America, IEEE.

**Adel A. M. Saleh**, B.Sc. (Electrical Engineering), 1963, University of Alexandria, Alexandria, Egypt; M.S. and Ph.D. (Electrical Engineering) Massachusetts Institute of Technology, Cambridge, MA, 1967 and 1970, respectively; Bell Laboratories, 1970—. From 1963 to 1965 Mr. Saleh worked as an instructor at the University of Alexandria. At Bell Laboratories he is engaged in research on microwave circuits, components, and communication systems. Senior Member, IEEE; member, Sigma Xi.

**Jack Salz**, B.S.E.E., 1955, M.S.E., 1956, and Ph.D., 1961, University of Florida; Bell Laboratories, 1961—. Mr. Salz first worked on remote line concentrators for the electronic switching system. Since 1968 he has supervised a group engaged in theoretical studies in data commu-

nications and is currently a member of the Communications Methods Research Department. During the academic year 1967-68, he was on leave as Professor of Electrical Engineering at the University of Florida. In Spring 1981, he was a visiting lecturer at Stanford University. Member, Sigma Xi.

**Manuel R. Santana**, B.Sc. (Electrical Engineering), 1970, University of Hartford; M.S., (Electrical Engineering), 1971, Georgia Institute of Technology; Bell Laboratories, 1970—. Mr. Santana has worked on the development of multipair cable and in the design, analysis, and testing of lightguide cable. He is presently Supervisor of the Lightguide Cable Group. Member, IEEE, Kappa Mu.

**Man Mohan Sondhi**, B.Sc. (Physics), Honours degree, 1950, Delhi University, Delhi, India, D.I.I.Sc. (Communications Engineering), 1953, Indian Institute of Science, Bangalore, India; M.S., 1955; Ph.D. (Electrical Engineering), 1957, University of Wisconsin, Madison, Wisconsin; Bell Laboratories, 1962—. Before joining Bell Laboratories, Mr. Sondhi worked for a year at the Central Electronics Engineering Research Institute, Pilani, India and taught for a year at the University of Toronto. At Bell Laboratories his research has included work on speech signal processing, echo cancellation, adaptive filtering, modeling of auditory and visual processes, and acoustical inverse problems. From 1971 to 1972 Mr. Sondhi was a guest scientist at the Royal Institute of Technology, Stockholm, Sweden.



# PAPERS BY BELL LABORATORIES AUTHORS

## COMPUTING/MATHEMATICS

- Bank R. E., Rose D. J., **Analysis of a Multilevel Iterative Method for Non-Linear Finite-Element Equations.** *Math Comput* 39(160):453-465, 1982.
- Benevit C. A., Cassard J. M., Dimmler K. J., Dumbri A. C., Mound M. G., Procyk F. J., Rosenzweig W., Yanof A. W., **A 256K Dynamic Random-Access Memory.** *IEEE J Soli* 17(5):857-862, 1982.
- Cleveland W. S., **Correlations Perceived and Measured.** *Science* 218(4575):919, 1982.
- Conway J. H., Sloane N. J. A., **Laminated Lattices.** *Ann Math* 116(3):593-620, 1982.
- Conway J. H., Sloane N. J. A., **Leech Roots and Vinberg Groups.** *P Roy Soc A* 384(1787):233-258, 1982.
- Daniels L. A. et al., **A High-Density Magnetic-Tape Cartridge Storage-System.** *IEEE Magnet* 18(6):1227-1229, 1982.
- Fishburn P. C., **Moment-Preserving Shifts and Stochastic-Dominance.** *Math Oper R* 7(4):629-634, 1982.
- Flajolet P., Odlyzko A., **The Average Height of Binary-Trees and Other Simple Trees.** *J Comput Sy* 25(2):171-213, 1982.
- Flatto L., **Limit-Theorems for Some Random-Variables Associated with Urn Models.** *Ann Probab* 10(4):927-934, 1982.
- Green M. L., Gyorgy E. M., **Computer Design of Induction Coil Cores.** *IEEE Magnet* 18(6):1556-1558, 1982.
- Hawkins D. T., **Inspec on Dialog.** *Database* 5(4):12-25, 1982.
- Jarvis J. F., **Research Direction in Industrial Machine Vision—A Workshop Summary.** *Computer* 15(12):55-61, 1982.
- Lax M., Agrawal G. P., **Evaluation of Fourier Integrals Using B-Splines.** *Math Comput* 39(160):535-548, 1982.
- Stroustrup, B., **An Experiment With the Interchangeability of Processes and Monitors.** *Software* 12(11):1011-1025, 1982.
- Yannakakis M., Papadimi C. H., **Algebraic Dependencies.** *J Comput Sy* 25(1):2-41, 1982.

## ENGINEERING

- Amitay N., Gans M. J., **Narrow Multibeam Satellite Ground Station Antenna Employing a Linear-Array With a Geosynchronous Arc Coverage of 60-Degrees. 1. Theory.** *IEEE Antenn* 30(6):1063-1067, 1982.
- Bjorkholm J. E. et al., **Optical Bistability Based on Self-Focusing—An Approximate Analysis.** *IEEE J Q El* 18(12):2016-2022, 1982.
- Capasso F., **The Channeling Avalanche Photo Diode—A Novel Ultra-Low-Noise Interdigitated p-n-Junction Detector.** *IEEE Device* 29(9):1388-1395, 1982.
- Cohen L. G., Mammel W. L., Jang S. J., **Low-Loss Quadruple-Clad Single-Mode Lightguides With Dispersion Below 2 ps/km nm Over the 1.28  $\mu\text{m}$ -1.65  $\mu\text{m}$  Wavelength Range.** *Electr Lett* 18(24):1023-1024, 1982.
- Forrest S. R., Smith R. G., Kim O. K., **Performance of  $\text{In}_{0.55}\text{Ga}_{0.47}\text{As}/\text{InP}$  Avalanche Photodiodes.** *IEEE J Q El* 18(12):2040-2048, 1982.
- Jin S., Tiefel T. H., Bennett J. E., Archer W. E., **New Pe-Cr-Mo Glass-Sealable Semihard Magnet Alloys for Remanent Reed Contacts.** *IEEE Magnet* 18(6):1454-1456, 1982.
- Joyce W. B., **Role of the Conductivity of the Confining Layer in DH-Laser Spatial Hole Burning Effects.** *IEEE J Q El* 18(12):2005-2009, 1982.
- Kantor W. M., **Independent Pairs of Good Self-Dual Codes (Letter).** *IEEE Info T* 28(6):966-967, 1982.
- Kaspi A. H., Berger T., **Rate-Distortion for Correlated Sources With Partially Separated Encoders.** *IEEE Info T* 28(6):828-840, 1982.

- Kiehl R. A., Flahive P. G., Wemple S. H., Cox H. M., **Direct-Coupled GaAs Ring Oscillators With Self-Aligned Gates.** *Elec Dev L* 3(11):325-326, 1982.
- Levey L., **Calculation of Ground Fault Currents Using an Equivalent-Circuit and a Simplified Ladder Network.** *IEEE Power* 101(8):2491-2497, 1982.
- Nelson T. J. et al., **Effects of N-H<sub>2</sub> Doses in Ion-Implanted Bubble Memory Test Chips.** *IEEE Magnet* 18(6):1358-1360, 1982.
- Rainal A. J., **Optimal Detection of FM Clicks (Letter).** *IEEE Info T* 28(6):971-973, 1982.
- Rana V. V. S., Lecraw R. C., **High Curie-Temperature Vanadium Containing Films for 8- $\mu$ m Period Magnetic-Bubble Devices.** *IEEE Magnet* 18(6):1292-1294, 1982.
- Stayt J. W., **A Fail-Safe Automatic Vent Control for Pfeiffer Turbomolecular Pumps.** *J Vac Sci T* 21(4):1028-1030, 1982.
- Trumbore F. A., **Citation Classic-Solid Solubilities of Impurity Elements in Germanium and Silicon.** *CC/Eng Tech* 1983(1):16, 1983.

## MANAGEMENT/ECONOMICS

- Allen S. J., Wilson B. A., Tsui D. C., **Cyclotron-Resonance, the Memory Function, and the Random Potential at the Si-SiO<sub>2</sub> Interface.** *Phys Rev B* 26(10):5590-5595, 1982.
- Buchsbaum S. J., **How Much R-and-D is Enough (Editorial).** *Laser Focus* 18(11):6, 1982.

## PHYSICAL SCIENCES

- Bally J., **Energetic Molecular Flows in Star-Forming Cloud Cores.** *Ann NY Acad* 395(Oct):191-196, 1982.
- Banavar J. R. et al., **Dynamic Monte-Carlo Scaling Method for Polymers.** *Chem P Lett* 93(1):35-37, 1982.
- Bishop D. J., Sprenger W. O., Pindak R., Neubert M. E., **Mechanical Measurements of the Melting Transition in Thin Liquid-Crystal Films.** *Phys Rev L* 49(25):1861-1864, 1982.
- Capasso F., Logan P. A., Tsang W. T., Hayes J. R., **Channeling Photodiode—A New Versatile Interdigitated P-N-Junction Photodetector.** *Appl Phys L* 41(10):944-946, 1982.
- Carroll P. J. et al., **The Crystal and Molecular-Structure of Bi(1,2,4,5,6-Pentahydro-1,3-Ditellurapentalen-2-Lidene)(Hexamethylenetetrafulvalene)—The Shortest Observed Intermolecular Te-Te Contact.** *J Chem S Ch* 1982(22):1316-1318, 1982.
- Citrin P. H., Hamann D. R., Mattheiss L. F., Rowe J. E., **Geometry and Electronic-Structure of Cl on the Cu(001) Surface.** *Phys Rev L* 49(23):1712-1715, 1982.
- Cohen R. L., Sherwood R. C., West K. W., **Particle-Size Distribution of Ni Micro-precipitates in LaNi<sub>5</sub> Used for Hydrogen Storage.** *Appl Phys L* 41(10):999-1001, 1982.
- Craighead H. G., Mankiewich P. M., **Ultra-Small Metal-Particle Arrays Produced by High-Resolution Electron-Beam Lithography.** *J Appl Phys* 53(11):7186-7188, 1982.
- Dicenzo S. B., **Relaxed Incommensurate Structure of the Intercalant Layer in Higher-Stage Graphite-Intercalation Compounds.** *Phys Rev B* 26(10):5878-5881, 1982.
- DiCenzo S. B., Wertheim G. K., Buchanan D. N., **XPS Studies of Adatom-Adatom Interactions: I/Ag(111) and I/Cu(111).** *Surf Sci* 121(3):411-420, 1982.
- Donnelly V. M., Flamm D. L., Tu C. W., Ibbotson D. E., **Temperature-Dependence of InP and GaAs Etching in a Chlorine Plasma.** *J Elchem So* 129(11):2533-2537, 1982.
- Douglass D. C. et al., **The Use of NMR Linewidths to Study b-Axis Distributions in Poled and Unpoled PVDF.** *J Chem Phys* 77(11):5826-5835, 1982.

- Draper C. W., Kaufmann E. N., Buene L., **Suppression of Metal-Surface Oxidation During Laser Surface Treatments.** Surf Int An 4(1):8-13, 1982.
- Dutta, N. K., **Calculated Threshold Current of GaAs Quantum Well Lasers.** J Appl Phys 53(11):7211-7214, 1982.
- Egan W. G., Aspnes D. E., **Finite-Wavelength Effects in Composite Media.** Phys Rev B 26(10):5313-5320, 1982.
- Englert T., Tsui D. C., Portal J. C., Beerens J., Gossard A., **Magnetophonon Resonances of the Two-Dimensional Electron-Gas in GaAs/AlGaAs Heterostructures.** Sol St Comm 44(8):1301-1304, 1982.
- Etienne B., Shah J., Leheny R. F., Nahory R. E., **Influence of Hot Carriers on the Temperature-Dependence of Threshold in 1.3- $\mu$ m InGaAsP Lasers.** Appl Phys L 41(11):1018-1020, 1982.
- Everaert J., Rentzepis P. M., **Picosecond Fluorescence Kinetics of N-All-Trans-Retinyldene-N-Butylamine.** Photochem P 36(5): 543-550, 1982.
- Frerking M. A., Langer W. D., Wilson R. W., **The Relationship Between Carbon-Monoxide Abundance and Visual Extinction in Inter-Stellar Clouds.** Astrophys J 262(2):590-605, 1982.
- Goldfarb R. B., Fickett F. R., Rao K. V., Chen H. S., **Spin-Freezing Below the Ferromagnetic Transition Determined by the Imaginary Component of AC Magnetic-Susceptibility.** J Appl Phys 53(11):7687-7689, 1982.
- Gottscho R. A., Burton R. H., Davis G. P., **Radiative Lifetime and Collisional Quenching of Carbon Monochloride (A 2-Delta) in an Alternating Current Glow-Discharge.** J Chem Phys 77(11):5298-5301, 1982.
- Greenside H. S., Ahlers G., Hohenberg P. C., Walden R. W., **A Simple Stochastic-Model for the Onset of Turbulence in Rayleigh-Benard Convection.** Physica D 5(2-3):322-334, 1982.
- Harrison T. R. et al., **Thin-Film CaF<sub>2</sub> Inorganic Electron Resist and Optical-Read Storage Medium.** Appl Phys L 41(11):1102-1104, 1982.
- Helfand E., Skolnick J., **Mechanism and Rates of Conformational Transitions in Heterogeneous Polymers.** J Chem Phys 77(11):5714-5724, 1982.
- Hwang J. C. M. et al., **Application of Molecular-Beam Epitaxy To III-V-Micro-wave and High-Speed Device Fabrication.** Sol St Tech 25(10):166-169, 1982.
- Inoue A., Okamoto S., Masumoto T., Chen H. S., **The Effect of Cold-rolling on the Superconducting Properties of an Amorphous Nb<sub>50</sub>Zr<sub>35</sub>Si<sub>15</sub> Alloy.** Scrip Metal 16(12):1393-1396, 1982.
- Jarvis, J. F., Valides F., Tyson J. A., **Analysis of Synthesized Dim Galaxy Images.** P Soc Photo 331:473-477, 1982.
- Joyce W. B., **Carrier Transport in Double-Heterostructure Active Layers.** J Appl Phys 53(11):7235-7239, 1982.
- Keramidas, V. G., **Metallurgical Interactions at Metallization Compound Semiconductor Interfaces.** Thin Sol Fi 96(4):347-363, 1982.
- Lee P. A., **Density of States and Screening Near the Mobility Edge.** Phys Rev B 26(10):5882-5885, 1982.
- Manheimer M. A., Bhagat S. M., Chen H. S., **Disappearance of Ferromagnetism in Amorphous (Fe<sub>1-x</sub>Mn<sub>x</sub>)<sub>75</sub>P<sub>16</sub>B<sub>6</sub>Al<sub>3</sub> Ribbons.** J Appl Phys 53(11):7737-7739, 1982.
- Marcuse D., **Influence of Curvature on the Losses of Doubly Clad Fibers.** Appl Optics 21(23):4208-4213, 1982.
- McBrierty V. J., Douglass D. C., Wang T. T., **Correlation Between the Piezoelectric Constant  $d_{31}$  and  $b$ -Axis Distribution in Polyvinylidene Fluoride.** Appl Phys L 41(11):1051-1053, 1982.
- Moncton D. E., Pindak R., Davey S. C., Brown G. S., **Melting of Variable-Thickness Liquid-Crystal Thin-Films—A Synchrotron X-Ray Study.** Phys Rev L 49(25):1865-1868, 1982.
- Miller T. A., Bondybey V. E., **The Laser Spectroscopy of Cool, Cold, and Very Cold Molecular-Ions.** Phi T Roy A 307(1500):617-631, 1982.
- Murphy D. W. et al., **Lithium Insertion in Anatase—A New Route to the Spinel LiTi<sub>2</sub>O<sub>4</sub>.** Rev Chim Mi 19(4-5):441-449, 1982.
- Pirronello V., et al., **Formaldehyde Formation in a H<sub>2</sub>O/CO<sub>2</sub> Ice Mixture Under Irradiation by Fast Ions.** Astrophys J 262(2):636-640, 1982.

- Presby H. M., Marcuse D., **Immersionless Single-Mode Preform Index Profiling.** *Appl Optics* 21(23):4253-4259, 1982.
- Pressman L. D., Forrest S. R., Bonner W. A., Van Uitert L. G., **Non-Invasive Analysis of InP Surfaces using Hg-InP Schottky-Barrier Diodes.** *Appl Phys L* 41(10):969-971, 1982.
- Rao K. V., Gerber J., Halley J. W., Chen H. S., **High-Field Magnetization Studies on an Amorphous Alloy with Para, Ferro and Spin-Glass Regions.** *J Appl Phys* 53(11):7731-7733, 1982.
- Rentzepis P. M., **Advances in Picosecond Spectroscopy.** *Science* 218(4578):1183-1189, 1982.
- Rossetti R., Brus L., **Electron-Hole Recombination Emission as a Probe of Surface-Chemistry in Aqueous CdS Colloids (Letter).** *J Phys Chem* 86(23):4470-4472, 1982.
- Shank C. V., Yen R., Fork P. L., Orenstein, J., **Picosecond Dynamics of Photo-Excited GaP States in Polyacetylene.** *Phys Rev L* 49(22):1660-1663, 1982.
- Stillinger F. H., **Structure of the Interface Between Coexisting Fluid Phases.** *Int J Quant* 1982(S16):137-147, 1982.
- Tamari N. et al., **High Quantum Efficiency InGaAsP InP Lasers.** *Appl Phys L* 41(11):1025-1027, 1982.
- Temkin H., Dutt B. V., Bonner W. A., Keramidas V. G., **Deep Radiative Levels in InP.** *J Appl Phys* 53(11):7526-7533, 1982.
- Tolk N. H. et al., **The Creation of Excited Hydrogenic States Near Surfaces.** *Nucl Instru* 202(1-2):247-251, 1982.
- Tsang W. T. et al., **1.3- $\mu$ m Wavelength GaInAsP InP Double Heterostructure Lasers Grown by Molecular-Beam Epitaxy.** *Appl Phys L* 41(11):1094-1096, 1982.
- Valdes F., **Resolution Classifier.** *P Soc Photo* 331:465-472, 1982.
- Vandenbergh J. M. et al., **Structural Study of Alloyed Gold Metallization Contacts on InGaAsP/InP Layers.** *J Appl Phys* 53(11):7385-7389, 1982.
- Vanuitert L. G., **The Effects of Electron Retention and Spin Pair Bonding on Crystal Cohesion in Insulator Oxides and Halides.** *J Appl Phys* 53(11):7140-7143, 1982.
- Walstedt R. E., Walker, L. R., **Numerical-Simulation of Spin-Glass Transition Phenomena.** *J Appl Phys* 53(11):7985-7990, 1982.
- Wilkins C. W., Reichmanis E., Chandross E. A., **Lithographic Evaluation of an Ortho-Nitrobenzyl Ester Based Deep UV Resist System.** *J Elchem So* 129(11):2552-2555, 1982.
- Wilson R. J., Hahn E. L., **Off-Resonance Transient-Response of a Three-Level System.** *Phys Rev A* 26(6):3404-3407, 1982.
- Wood D. L., Nassau K., Chadwick D. L., **Optical-Properties of New Oxide Glasses With Potential for Long-Wavelength Optical Fibers.** *Appl Optics* 21(23):4276-4279, 1982.

## **SOCIAL AND LIFE SCIENCES**

- Friedman J. M. et al., **Transient Raman Study of Hemoglobin—Structural Dependence of the Iron-Histidine Linkage.** *Science* 218(4578):1244-1246, 1982.
- Pashley R. D., Pedersen R. A., **Special Issue on Logic and Memory—Foreword (Editorial).** *IEEE J Soli* 17(5):791-792, 1982.
- Tartter V. C., Fischer S. D., **Perceiving Minimal Distinctions in ASL Under Normal and Point-Light Display Conditions.** *Perc Psych* 32(4):327-334, 1982.

## **SPEECH AND ACOUSTICS**

- Broek H. W., **Temporal and Spatial Fluctuations in Single-Path Underwater Acoustic-Wave Fronts. 1. Transmission From the 1st Convergence Zone at 43-nmi Range.** *J Acoust So* 72(5):1527-1532, 1982.
- Broek H. W., **Temporal and Spatial Fluctuations in Single-Path Underwater**

**Acoustic-Wave Fronts. 2. Transmission over 270-nmi Range.** J Acoust So 72(5):1533-1537, 1982.

**Broek H. W., Temporal and Spatial Fluctuations in Single-Path Underwater Acoustic-Wave Fronts. 3. Transmission Over 540-nmi Range.** J Acoust So 72(5):1538-1543, 1982.

**Mangano D. T., Broek H. W., Correlation of Phase and Amplitude Fluctuations at Hydrophones Spaced 1/2 to 2 Wavelengths Apart.** J Acoust So 72(5):1556-1563, 1982.



# CONTENTS, MAY-JUNE 1983

## Part 1

A Long Wavelength Optical Receiver Using a Short Channel Si-MOSFET

K. Ogawa, B. Owen, and H. J. Boll

Modal Analysis of Loss and Mode Mixing in Multimode Parabolic Index Splices

I. A. White and S. C. Mettler

Experimental Results of 20-Mb/s FSK Digital Transmission on 4-GHz (TD) Radio

Y. Y. Wang

An Experimental Broadband Imaging Feed

T. S. Chu and R. W. England

An M/G/1 Queue With a Hybrid Discipline

B. T. Doshi

Off-Line Quality Control in Integrated Circuit Fabrication Using Experimental Design

M. S. Phadke, R. N. Kacker, D. V. Speeney, and M. J. Grieco

The Effects of Selected Signal Processing Techniques on the Performance of a Filter-Bank-Based Isolated Word Recognizer

B. A. Dautrich, L. R. Rabiner, and T. B. Martin











**THE BELL SYSTEM TECHNICAL JOURNAL** is abstracted or indexed by *Abstract Journal in Earthquake Engineering, Applied Mechanics Review, Applied Science & Technology Index, Chemical Abstracts, Computer Abstracts, Current Contents/Engineering, Technology & Applied Sciences, Current Index to Statistics, Current Papers in Electrical & Electronic Engineering, Current Papers on Computers & Control, Electronics & Communications Abstracts Journal, The Engineering Index, International Aerospace Abstracts, Journal of Current Laser Abstracts, Language and Language Behavior Abstracts, Mathematical Reviews, Science Abstracts (Series A, Physics Abstracts; Series B, Electrical and Electronic Abstracts; and Series C, Computer & Control Abstracts), Science Citation Index, Sociological Abstracts, Social Welfare, Social Planning and Social Development, and Solid State Abstracts Journal*. Reproductions of the Journal by years are available in microform from University Microfilms, 300 N. Zeeb Road, Ann Arbor, Michigan 48106.

