

May-June 1985 Vol. 64 No. 5

AT&T TECHNICAL JOURNAL

A JOURNAL OF THE AT&T COMPANIES

Trellis Coding

Antennas

Digit and Word Recognition

COMPUTING SCIENCE AND SYSTEMS:

An Accurate Light Pen

Sojourn Time Distribution

Queueing Models

PLL Analysis and Design

EDITORIAL BOARD

M. IWAMA, *Board Chairman*¹

S. M. BAUMAN²

W. F. BRINKMAN³

J. CHERNAK¹

M. F. COCCA¹

B. R. DARNALL¹

A. FEINER²

P. A. GANNON⁴

W. R. HARLIN⁵

T. J. HERR⁴

D. M. HILL⁵

D. HIRSCH²

S. HORING¹

J. S. NOWAKI¹

L. C. SEIFERT⁶

W. E. STRICH⁷

J. W. TIMKO¹

V. A. VYSSOTSKY¹

¹AT&T Bell Laboratories ²AT&T Information Systems ³Sandia National Laboratories

⁴AT&T Network Systems ⁵AT&T Technology Systems ⁶AT&T Technologies

⁷AT&T Communications

COMPUTING SCIENCE & SYSTEMS TECHNICAL EDITORIAL COMMITTEE

M. D. MCILROY

Technical Editor

A. V. AHO

D. L. BAYER

W. FIGHTNER

L. E. GALLAHER

R. W. GRAVES¹

M. G. GRISHAM

B. W. KERNIGHAN

S. G. WASILEW

S. J. YUILL¹

¹AT&T Information Systems

EDITORIAL STAFF

B. G. KING, *Editor*

P. WHEELER, *Managing Editor*

L. S. GOLLER, *Assistant Editor*

A. M. SHARTS, *Assistant Editor*

B. VORCHHEIMER, *Circulation*

AT&T TECHNICAL JOURNAL (ISSN 8756-2324) is published ten times each year by AT&T, 550 Madison Avenue, New York, NY 10022; C. L. Brown, Chairman of the Board; T. O. Davis, Secretary. The Computing Science and Systems section and the special issues are included as they become available. Subscriptions: United States—1 year \$35; foreign—1 year \$45.

Back issues of the special, single-subject supplements may be obtained by writing to the AT&T Customer Information Center, P.O. Box 19901, Indianapolis, Indiana 46219, or by calling (800) 432-6600. Back issues of the general, multisubject issues may be obtained from University Microfilms, 300 N. Zeeb Road, Ann Arbor, Michigan 48106.

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 AT&T Technical Journal and sent to AT&T Bell Laboratories, Circulation Dept., Room 1E335, 101 J. F. Kennedy Pky, Short Hills, NJ 07078.

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

Printed in U.S.A. Second-class postage paid at Short Hills, NJ 07078 and additional mailing offices. Postmaster: Send address changes to the AT&T Technical Journal, Room 1E335, 101 J. F. Kennedy Pky, Short Hills, NJ 07078.

Copyright © 1985 AT&T.

AT&T TECHNICAL JOURNAL

VOL. 64

MAY-JUNE 1985

NO. 5

Copyright® 1985 AT&T, Printed in U.S.A.

Four-Dimensional Modulation With an Eight-State Trellis Code	1005
A. R. Calderbank and N. J. A. Sloane	
100-GHz Measurements of Two Astigmatic Launchers	1019
R. A. Semplak	
On the Use of Vector Quantization for Connected-Digit Recognition	1033
S. C. Glinski	
Incorporation of Temporal Structure Into a Vector-Quantization-Based Preprocessor for Speaker-Independent, Isolated-Word Recognition	1047
A. F. Bergh, F. K. Soong, and L. R. Rabiner	

COMPUTING SCIENCE AND SYSTEMS

A New Light Pen With Subpixel Accuracy	1065
M. Hatamian and E. F. Brown	
Sojourn Time Distribution in a Multiprogrammed Computer System	1077
K. M. Rege and B. Sengupta	
Application of Decomposition Principle in M/G/1 Vacation Model to Two Continuum Cyclic Queueing Models—Especially Token-Ring LANs	1091
S. W. Fuhrmann and R. B. Cooper	
PLACE 2.0—An Interactive Program for PLL Analysis and Design	1101
J. H. Saunders	

PAPERS BY AT&T BELL LABORATORIES AUTHORS	1135
CONTENTS, JULY-AUGUST ISSUE	1143
ERRATA	1145

Four-Dimensional Modulation With an Eight-State Trellis Code

By A. R. CALDERBANK and N. J. A. SLOANE*

(Manuscript received January 9, 1985)

A trellis code is a “sliding window” method for encoding a binary data stream $\{a^i\}$, $a^i = 0, 1$, as a sequence of signal points drawn from \mathbf{R}^n . The rule for assigning signal points depends on the state of the encoder. In this paper $n = 4$, and the signal points are 4-tuples of odd integers. We describe an infinite family of eight-state trellis codes. For $k = 3, 4, 5, \dots$ we construct a trellis encoder with a rate of k bits/four-dimensional signal. We propose that the codes with rates $k = 8$ and 12 be considered for use in modems designed to achieve data rates of 9.6 kb/s and 14.4 kb/s, respectively.

I. INTRODUCTION

A trellis code is a “sliding window” method for encoding a binary data stream $\{a^i\}$, $a^i = 0, 1$, as a sequence of signal points $\{x^i\}$ drawn from \mathbf{R}^n . The set of possible signal points is finite, and this set is called the *signal constellation*. The purpose of coding is to gain noise immunity beyond that provided by standard uncoded transmission at the same data rate. In this paper $n = 4$, and the signal points are drawn from $(2\mathbf{Z} + 1)^4$, the lattice of 4-tuples of odd integers. We shall regard transmission of a four-dimensional signal as one use of the channel, and we measure the rate of the code in bits per channel use. The four-dimensional signal space can be realized by using two space-orthogonal electric field polarizations to communicate on the same carrier frequency. It is also possible to regard each four-dimensional symbol as two consecutive two-dimensional symbols.

* Authors are employees of AT&T Bell Laboratories.

Copyright © 1985 AT&T. Photo reproduction for noncommercial use is permitted without payment of royalty provided that each reproduction is done without alteration and that the Journal reference and copyright notice are included on the first page. The title and abstract, but no other portions, of this paper may be copied or distributed royalty free by computer-based and other information-service systems without further permission. Permission to reproduce or republish any other portion of this paper must be obtained from the Editor.

Ungerboeck¹ described a technique called *set partitioning*, which assigns signal points to successive blocks of input data. The rule for assigning signal points depends on the state of the encoder. Ungerboeck constructed simple *trellis codes* providing the same noise immunity as is given by increasing the power of uncoded transmission by factors ranging from 2 to 4 (coding gains ranging from 3 to 6 dB). Calderbank and Mazo² have given a different algebraic description of trellis codes. Trellis codes with a rate of 4 bits/two-dimensional symbol have recently been proposed for use in modems designed to achieve data rates of 9.6 kb/s on dial-up voice telephone lines. These codes use the signal constellation shown in Fig. 1, which was originally described by Campopiano and Glazer,³ and gain 4 dB over uncoded transmission at the same rate. In Section III of this paper we describe the first code in our infinite family. This code has a rate of 8 bits/four-dimensional symbol and promises a gain of 4.7 dB over uncoded transmission. The signal constellation consists of 512 four-dimensional signal points. Transmission of two consecutive two-dimensional signals using one of the proposed trellis codes with a rate of 4 bits/two-dimensional symbol requires $1024 = 32^2$ four-dimensional signal points. Furthermore, the restriction of the 512-point constellation to the first two coordinates, or to the last two coordinates, is the 32-point constellation shown in Fig. 1. The 0.7-dB improvement in performance is derived from reducing the average transmitted power.

In Section IV we briefly describe the second code in the family, which has a rate of 12 bits/four-dimensional symbol and promises a

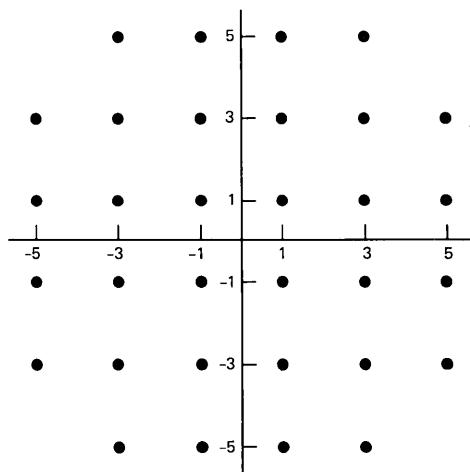


Fig. 1—Signal constellation for proposed trellis codes with rate 4 bits/two-dimensional symbol.

coding gain of 4.9 dB over uncoded transmission. We propose using this code in modems designed to achieve data rates of 14.4 kb/s.

We begin by presenting a rate 3/4 binary convolutional code that is basic to the construction of the new trellis codes. In Section V we describe the general code in the family, with rate k bits/four-dimensional symbol, for $k = 3, 4, 5 \dots$, and we show that in the limit, as $k \rightarrow \infty$, the coding gain is asymptotic to $10 \log_{10} \pi \approx 4.9715$ dB. The difference between this limiting coding gain and that provided by the code with $k = 12$ is very small.

After this paper was submitted, we discovered that Forney et al.⁴ had independently proposed a different rate $k = 8$ code with approximately the same performance. Wilson, Sleeper, and Smith⁵ have described simple trellis codes (with up to four encoder states) that use four-dimensional signal constellations.

II. A RATE 3/4 BINARY CONVOLUTIONAL CODE

We assume that binary data are being encoded at a rate of k bits/signal point and that the data enter the encoder in k parallel sequences, $\{a_1^i\}, \{a_2^i\}, \dots, \{a_k^i\}$. We assume that the output \mathbf{x}^i of the trellis encoder at time i depends not only on the present values $a_1^i, a_2^i, \dots, a_k^i$, of the input sequences, but also on the previous $v_j \geq 0$ bits of the j th sequence. If $v_j = 0$ for some j , then $a_1^j, a_2^j, a_3^j, \dots$ is said to be a sequence of *uncoded bits*. The *constraint length* ν is given by $\nu = \sum_{j=1}^k v_j$. The output \mathbf{x}^i of the encoder is a fixed vector-valued function \mathbf{x} of the $\nu + k$ binary variables $a_1^i, \dots, a_1^{i-\nu_1}; a_2^i, \dots, a_2^{i-\nu_2}; \dots; a_k^i, \dots, a_k^{i-\nu_k}$. That is,

$$\mathbf{x}^i = \mathbf{x}(a_1^i a_1^{i-1} \dots a_1^{i-\nu_1}; \dots; a_k^i \dots a_k^{i-\nu_k}).$$

The ν -tuple $(a_1^{i-1} \dots a_1^{i-\nu_1}; a_2^{i-1} \dots a_2^{i-\nu_2}; \dots; a_k^{i-\nu_k})$ is the *state* of the encoder and there are 2^ν states. Figure 2 shows a state transition diagram for a trellis code with $k = 3, \nu_1 = 0, \nu_2 = 1, \nu_3 = 2$. The average transmitted signal power P is given by

$$P = \frac{1}{2^{k+\nu}} \sum \| \mathbf{x}(a_1^i \dots a_1^{i-\nu_1}; \dots; a_k^i \dots a_k^{i-\nu_k}) \|^2.$$

Basic to the trellis codes constructed below is a certain rate 3/4 binary convolutional code with total memory 3 and free distance 4. The encoder is presented in Fig. 3, which is taken from Ref. 6 (Fig. 10.3, p. 292). The three parallel input sequences determine the output sequence $\{\mathbf{v}^i = (v_1^i, v_2^i, v_3^i, v_4^i)\}$ according to the following rules:

$$v_1^i = a_1^i,$$

$$v_2^i = a_1^i + a_2^i + a_2^{i-1} + a_3^{i-1},$$

$$v_3^i = a_1^i + a_2^{i-1} + a_3^i + a_3^{i-2},$$

$$v_4^i = a_1^i + a_2^i + a_3^i + a_3^{i-2}.$$

The triple $a_2^{i-1}a_3^{i-1}a_3^{i-2}$ is the state of the encoder. The possible transitions between states are shown in Fig. 2. The edge joining state $a_2^{i-1}a_3^{i-1}a_3^{i-2}$ to state $a_2^i a_3^i a_3^{i-1}$ is labeled with the outputs $\mathbf{v}^i = (v_1^i, v_2^i, v_3^i, v_4^i)$ and $\bar{\mathbf{v}}^i = (\bar{v}_1^i, \bar{v}_2^i, \bar{v}_3^i, \bar{v}_4^i)$ corresponding to this transition. Note that

$$\bar{v}_1^i = \bar{a}_1^i,$$

$$\bar{v}_2^i = \bar{a}_1^i + a_2^i + a_2^{i-1} + a_3^{i-1},$$

$$\bar{v}_3^i = \bar{a}_1^i + a_2^{i-1} + a_3^i + a_3^{i-2},$$

$$\bar{v}_4^i = \bar{a}_1^i + a_2^i + a_3^i + a_3^{i-2}.$$

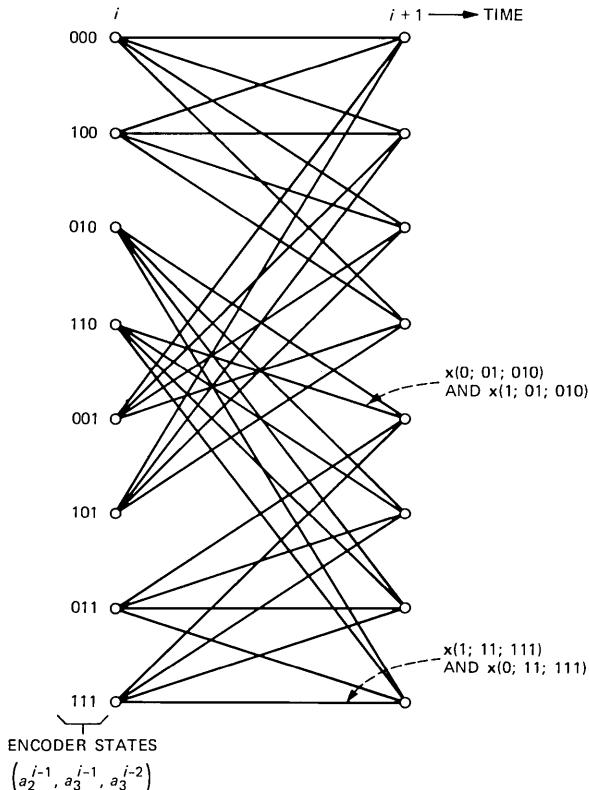


Fig. 2—A state transition diagram for a trellis code with $k = 3$, $\nu_1 = 0$, $\nu_2 = 1$, $\nu_3 = 2$. (Every edge represents two possible transitions.)

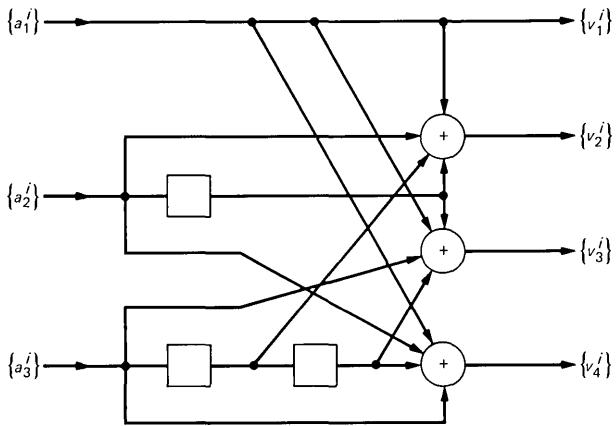


Fig. 3—A rate 3/4 binary convolutional code.

We change from 0, 1 notation to ± 1 notation ($0 \leftrightarrow +1$ and $1 \leftrightarrow -1$). An edge joining two states is now labeled with pairs of vectors $\pm(w_1, w_2, w_3, w_4)$, where $w_i = \pm 1$, $i = 1, 2, 3, 4$. This defines a trellis encoder with a rate of 3 bits/four-dimensional symbol. The minimum squared distance of this trellis code is simply four times the free distance of the original binary convolutional code, namely, 16. This is because 0 opposite 1 contributes 1 to the free distance, whereas 1 opposite -1 contributes 4 to the squared minimum distance.

Transmission at the higher rates of 8 and 12 bits/four-dimensional symbol requires more channel symbols. Indeed, to achieve any coding gain, we have to use more symbols than are required by uncoded transmission at the same rate.

III. A TRELLIS CODE WITH RATE 8 BITS/FOUR-DIMENSIONAL SYMBOL

Uncoded transmission at the rate 4 bits/two-dimensional symbol uses the rectangular signal constellation shown in Fig. 4. To achieve uncoded transmission of a four-dimensional symbol at a rate of 8 bits/symbol, simply take two copies of this scheme. There are 256 possible signals and the average power is $4(1^2 + 3^2)/2 = 20$. Since the minimum squared distance between distinct signals is 4, we have

$$\left(\frac{d_{\min}^2}{P} \right)_{\text{uncoded}} = \frac{4}{20}.$$

For coded transmission we shall use $2 \times 256 = 512$ signal points. Representative signal points are listed in Table I. The remaining points are obtained from these representatives by permuting the

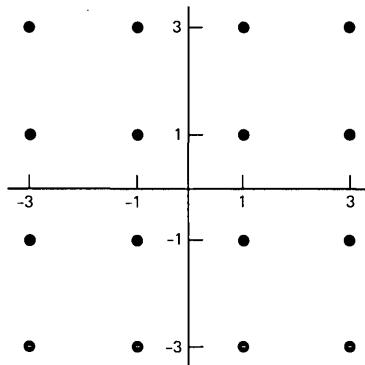


Fig. 4—The rectangular constellation for uncoded transmission at 4 bits/two-dimensional symbol.

Table I—The signal constellation for coded transmission at 8 bits/four-dimensional symbol. All permutations of coordinates and all sign changes are allowed.

Representative	Energy	$1/16 \times$ Number
(1111)	4	1
(3111)	12	4
(3311)	20	6
(5111)	28	4
(3331)	28	4
(5311)	36	12
(3333)	36	1

coordinates and changing signs in all possible ways. For example, $(\bar{3}\bar{1}\bar{3}1)$, $(13\bar{1}\bar{5})$, and $(\bar{5}111)$ are all signal points (where \bar{x} denotes $-x$). For every vector $\mathbf{w} = (w_1, w_2, w_3, w_4)$ with $w_i = \pm 1$, $i = 1, 2, 3, 4$, let $S(\mathbf{w})$ be the set of 32 signal points (x_1, x_2, x_3, x_4) satisfying $x_i \equiv w_i \pmod{4}$, for $i = 1, 2, 3, 4$. The sets $S(\mathbf{w})$ partition the signal constellation into 16 equal parts. The set $S(1111)$ is shown in Table II and the other sets are obtained from $S(1111)$ by changing signs. For example, $S(1\bar{1}\bar{1}\bar{1})$ is obtained from $S(1111)$ by changing the signs of the second and fourth entries. The distance $d(A, B)$ between two sets of vectors A and B is given by

$$d(A, B) = \min_{\mathbf{x} \in A, \mathbf{y} \in B} \{\|\mathbf{x} - \mathbf{y}\|\}.$$

The partition into sets $S(\mathbf{w})$ satisfies the following metric properties:

Table II—The set $S(1111)$. All permutations of coordinates are allowed.

Signal Point	Energy	Number
(1111)	4	1
(3111)	12	4
(3311)	20	6
(5111)	28	4
(3331)	28	4
(5311)	36	12
(3333)	36	1

$$(M1) \text{ if } \mathbf{x}, \mathbf{y} \in S(\mathbf{w}) \text{ then } \|\mathbf{x} - \mathbf{y}\|^2 \geq 16,$$

$$(M2) \text{ if } \mathbf{v} \neq \mathbf{w} \text{ then } d^2(S(\mathbf{v}), S(\mathbf{w})) = \|\mathbf{v} - \mathbf{w}\|^2.$$

To verify (M1) let $\mathbf{x} = (x_1, x_2, x_3, x_4)$ and $\mathbf{y} = (y_1, y_2, y_3, y_4)$. Then $x_i \neq y_i$ for some i . Since $x_i \equiv y_i \pmod{4}$, we have $\|\mathbf{x} - \mathbf{y}\|^2 \geq 16$. To verify (M2) let $\mathbf{x} = (x_1, x_2, x_3, x_4) \in S(\mathbf{v})$ and $\mathbf{y} = (y_1, y_2, y_3, y_4) \in S(\mathbf{w})$. If $x_i \neq y_i \pmod{4}$ then $|x_i - y_i|^2 \geq 4$. Hence $\|\mathbf{x} - \mathbf{y}\|^2 \geq \|\mathbf{v} - \mathbf{w}\|^2$ and equality holds when $\mathbf{x} = \mathbf{v}$ and $\mathbf{y} = \mathbf{w}$.

In Section II we described a trellis code with a rate of 3 bits/four-dimensional symbol and minimum squared distance $d_{\min}^2 = 16$. To achieve the higher transmission rate of 8 bits/four-dimensional symbol, we add 5 uncoded bits. There are now eight parallel input sequences $\{a_1^i\}, \dots, \{a_8^i\}$. The sequences $\{a_2^i\}, \{a_3^i\}$ determine the state $a_2^{i-1}a_3^{i-1}a_3^{i-2}$ of the encoder as in Fig. 3. An edge joining two states that was originally labeled by the pair of vectors $\pm \mathbf{v}$ is now labeled by the 64 vectors in $S(\mathbf{v}) \cup S(-\mathbf{v})$. This is because there are 64 parallel transitions between states $a_2^{i-1}a_3^{i-1}a_3^{i-2}$ and $a_2^i a_3^i a_3^{i-1}$ corresponding to the 64 possible inputs $a_1^i a_4^i \dots a_8^i$. We allow any fixed assignment of channel symbols in $S(\mathbf{v}) \cup S(-\mathbf{v})$ to inputs $a_1^i a_4^i \dots a_8^i$.

Consider the distance properties of the high-rate code. Properties (M1) and (M2) guarantee that the squared distance of any error event of length 1 is at least 16. Consider any error event in the eight-state trellis of length greater than 1. If the squared distance for the low-rate code is

$$\sum_{i=1}^l \|\mathbf{v}^i - \hat{\mathbf{v}}^i\|^2,$$

then the squared distance for the high-rate code is at least

$$\sum_{i=1}^l d^2(S(\mathbf{v}^i), S(\hat{\mathbf{v}}^i)).$$

Property (M2) now implies that the minimum squared distance of the high-rate code is at least 16.

The average signal power P of the 512-point signal constellation is given by

$$P = \frac{16(4 \times 1 + 12 \times 4 + 20 \times 6 + 28 \times 8 + 36 \times 13)}{512} = 27.$$

Thus,

$$\left(\frac{d_{\min}^2}{P}\right)_{\text{coded}} = \frac{16}{27},$$

and the coding gain (in decibels) is

$$10 \log_{10} \left(\frac{(d_{\min}^2/P)_{\text{coded}}}{(d_{\min}^2/P)_{\text{uncoded}}} \right) = 10 \log_{10} \left(\frac{16/27}{4/20} \right) \approx 4.717 \text{ db.}$$

IV. A TRELLIS CODE WITH RATE 12 BITS/FOUR-DIMENSIONAL SYMBOL

Uncoded transmission at the rate of 6 bits/two-dimensional symbol uses the 64-point rectangular constellation shown in Fig. 5. To achieve uncoded transmission of a four-dimensional symbol at a rate of 12 bits/symbol, simply take two copies of this scheme. There are $64^2 = 2^{12}$ possible signals, and the average signal power P is $4(1^2 + 3^2 + 5^2 + 7^2)/4 = 84$. Thus,

$$\left(\frac{d_{\min}^2}{P}\right)_{\text{uncoded}} = \frac{4}{84}.$$

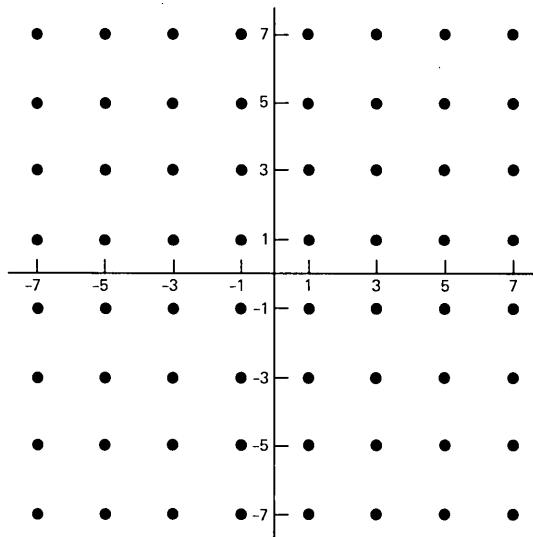


Fig. 5—A rectangular constellation for uncoded transmission at 6 bits/two-dimensional symbol.

Table III—The signal constellation for coded transmission at 12 bits/four-dimensional symbol. All representatives are taken from $S(1111)$.

Representative	Energy	$(1/16) \times \text{Number}$
(1111)	4	1
(3111)	12	4
(3311)	20	6
(5111), ($\overline{3331}$)	28	$4 + 4 = 8$
(5311), ($\overline{3333}$)	36	$12 + 1 = 13$
(5331)	44	12
(7111), ($\overline{5333}$), (5511)	52	$4 + 4 + 6 = 14$
(7311), (5531)	60	$12 + 12 = 24$
(7331), ($\overline{5533}$)	68	$12 + 6 = 18$
(7333), ($\overline{5551}$), ($\overline{7511}$)	76	$4 + 4 + 12 = 20$
(9111), ($\overline{7531}$), (5553)	84	$4 + 24 + 4 = 32$
(9311), ($\overline{7533}$)	92	$12 + 12 = 24$
(9331), ($\overline{7711}$), ($\overline{7551}$), (5555)	100	$12 + 6 + 12 + 1 = 31$
(7553), ($\overline{9333}$), (9511), ($\overline{7731}$)	108	$12 + 4 + 12 + 12 = 40$
(9531), ($\overline{7733}$)	116	$24 + 6 = 30$
(9533), (11111), ($\overline{7751}$), ($\overline{7555}$)	124	$12 + 4 + 12 + 4 = 32$
(11311), (9711), (9551), ($\overline{7753}$)	132	$12 + 12 + 12 + 12 = 48$
(11331), (9731), (9553)	140	$12 + 24 + 12 = 48$
(11333), (11511), (9733), ($\overline{7755}$), ($\overline{7771}$)	148	$4 + 12 + 12 + 6 + 4 = 38$
(11531), (9555), (9751), ($\overline{7773}$)	156	$24 + 4 + 24 + 4 = 56$
(11533), (9911), (9753)	164	only 13

For coded transmission we use $2 \times 2^{12} = 2^{13}$ signal points. As in Section III we partition the signal constellation into 16 sets $S(\mathbf{w})$ according to congruence of the entries modulo 4. Each set $S(\mathbf{w})$ contains 512 signal points. Representative signal points are listed in Table III, where the representatives are all taken from $S(1111)$.

To achieve the transmission rate of 12 bits/four-dimensional symbol, we add 9 uncoded bits to the low-rate trellis code described in Section II. There are now 1024 parallel transitions between states $a_2^{i-1}a_3^{i-1}a_3^{i-2}$ and $a_2^ia_3^ia_3^{i-1}$ in the eight-state trellis. If the edge corresponding to this transition was originally labeled $\pm\mathbf{v}$, it is now labeled with the 1024 vectors in $S(\mathbf{v}) \cup S(-\mathbf{v})$. The metric properties (M1) and (M2) guarantee that the squared minimum distance of the high-rate code is equal to the squared minimum distance of the low-rate code, which is 16. An easy calculation shows that the average signal power P is 108.625, so

$$\left(\frac{d_{\min}^2}{P} \right)_{\text{coded}} = \frac{16}{108.625}.$$

The coding gain is

$$10 \log_{10} \left(\frac{10/108.625}{4/84} \right) \approx 4.904 \text{ dB.}$$

V. ASYMPTOTIC PERFORMANCE OF A FAMILY OF CODES

To achieve coded transmission at the rate of k bits/four-dimensional signal, we add $k - 3$ uncoded bits to the low-rate trellis code described in Section II. There are 2^{k-2} parallel transitions between states $a_2^{i-1}a_3^{i-1}a_3^{i-2}$ and $a_2^ia_3^ia_3^{i-1}$ in the eight-state trellis. Coded transmission requires 2^{k+1} signal points. The points of the lattice $(2\mathbb{Z} + 1)^4$ lie in shells around the origin consisting of 16 vectors of energy 4, 64 vectors of energy 12, and so on (see Table III). The 2^{k+1} signal points are obtained by taking all points of energy 4, 12, 20, ... and just enough points of a final shell to bring the total number up to 2^{k+1} . The signal constellation is partitioned into 16 sets $S(\mathbf{v})$ according to congruence of the entries modulo 4. Each set contains 2^{k-3} signal points. Edges in the eight-state trellis originally labelled $\pm \mathbf{w}$ are now labeled with the 2^{k-2} vectors in $S(\mathbf{v}) \cup S(-\mathbf{v})$. The metric properties (M1) and (M2) guarantee that the minimum squared distance of this trellis code is 16.

Consider the asymptotic performance of this family of codes. For simplicity suppose that the signal constellation of each code in the family is a complete union of energy shells. If \mathbf{x} is a vector in the lattice $(2\mathbb{Z} + 1)^4$, then $\|\mathbf{x}\|^2 \equiv 4 \pmod{8}$, since $\|\mathbf{x}\|^2$ is the sum of four odd squares. A classical result, due to Jacobi and to Legendre, is that every positive integer of the form $8n + 4$ is a sum of four odd squares in $\sigma(2n + 1)$ ways, where $\sigma(m)$ is the sum of divisors of m . The generating function

$$16 \sum_{\substack{m \geq 1 \\ m \text{ odd}}} \sigma(m) q^{4m}$$

expresses the fact that there are 16 $\sigma(m)$ vectors of energy $4m$ in the lattice $(2\mathbb{Z} + 1)^4$. The factor of 16 arises from the 16 possible sign changes.

The following estimates (which are proved in the Appendix) will be used to calculate the average energy of the vectors \mathbf{x} in $(2\mathbb{Z} + 1)^4$ with $\|\mathbf{x}\|^2 \leq 4n$:

$$\sum_{\substack{1 \leq m \leq n \\ m \text{ odd}}} \sigma(m) = \frac{\pi^2 n^2}{32} + O(n \log n), \quad (1)$$

$$\sum_{\substack{1 \leq m \leq n \\ m \text{ odd}}} m\sigma(m) = \frac{\pi^2 n^3}{48} + O(n^2 \log n). \quad (2)$$

The average energy P of the vectors \mathbf{x} in $(2\mathbb{Z} + 1)^4$ with $\|\mathbf{x}\|^2 \leq 4n$ is given by

$$P = \frac{4(\pi^2 n^3/48 + O(n^2 \log n))}{\pi^2 n^2/32 + O(n \log n)} = \frac{8}{3} n + O(\log n).$$

Therefore,

$$\left(\frac{d_{\min}^2}{P} \right)_{\text{coded}} = \frac{16}{(8/3)n + O(\log n)} = \frac{6}{n} + O(\log n)/n^2. \quad (3)$$

The number of points in the signal constellation is

$$16 \sum_{\substack{1 \leq m \leq n \\ m \text{ odd}}} \sigma(m) = \frac{\pi^2 n^2}{2} + O(n \log n).$$

For uncoded transmission we use just half this many points. The signal constellation is the set of all 4-tuples $\mathbf{x} = (x_1, x_2, x_3, x_4)$, where $x_i = \pm 1, \pm 3, \dots, \pm(2a - 1)$. The number of points in this constellation is $16a^4$, so we choose a to make a^4 close to $\pi^2 n^2 / 64$. Now

$$1^2 + 3^2 + 5^2 + \dots + (2a - 1)^2 = \frac{a(4a^2 - 1)}{3},$$

so the average signal power P is given by

$$P = \frac{4a(4a^2 - 1)}{3} a = \frac{4}{3} (4a^2 - 1).$$

Since the minimum squared distance between distinct signals is 4, we have

$$\left(\frac{d_{\min}^2}{P} \right)_{\text{uncoded}} = \frac{4}{4(4a^2 - 1)/3} = \frac{6}{8a^2 - 2}.$$

Now $8a^2 - 2 = \pi n + O(\sqrt{n})$, so

$$\left(\frac{d_{\min}^2}{P} \right)_{\text{uncoded}} = \frac{6}{\pi n} + O(n^{-3/2}).$$

Therefore,

$$\lim_{n \rightarrow \infty} \frac{(d_{\min}^2/P)_{\text{coded}}}{(d_{\min}^2/P)_{\text{uncoded}}} = \pi,$$

and the limiting coding gain is $10 \log_{10} \pi = 4.9714 \dots$ dB.

REFERENCES

1. G. Ungerboeck, "Channel Coding With Multilevel/Phase Signals," IEEE Trans. Inform. Theory, IT-28, No. 1 (January 1982), pp. 55–67.
2. A. R. Calderbank and J. E. Mazo, "A New Description of Trellis Codes," IEEE Trans. Inform. Theory, IT-30 (December 1984), pp. 784–91.
3. C. N. Campopiano and B. G. Glazer, "A Coherent Digital Amplitude and Phase

- Modulation Scheme," IRE Trans. Commun. Syst., CS-10 (1962), pp. 90-5.
4. G. D. Forney, Jr., et al., "Efficient Modulation for Band-Limited Channels," IEEE J. Selected Areas Commun., SAC-2 (August 1984), pp. 632-47.
 5. S. G. Wilson, H. A. Sleeper, and N. K. Smith, "Four-Dimensional Modulation and Coding: An Alternative to Frequency-Reuse," in *Science, Systems and Services for Communications*, P. Dewilde and C. A. May, eds., New York and Amsterdam: IEEE/Elsevier-North Holland, 1984, pp. 919-23.
 6. S. Lin and D. J. Costello, Jr., *Error Control Coding: Fundamentals and Applications*, Englewood Cliffs, N.J.: Prentice-Hall, 1983.
 7. T. M. Apostol, *Introduction to Analytic Number Theory*, New York: Springer-Verlag, 1976.

APPENDIX

Proof of Equations (1) and (2)

To prove that

$$\sum_{\substack{1 \leq m \leq n \\ m \text{ odd}}} \sigma(m) = \frac{\pi^2 n^2}{32} + O(n \log n),$$

we write

$$\begin{aligned} \sum_{\substack{1 \leq m \leq n \\ m \text{ odd}}} \sigma(m) &= \sum_{\substack{1 \leq m \leq n \\ m \text{ odd}}} \sum_{\substack{q \mid m \\ q \text{ odd}}} q \\ &= \sum_{\substack{d \leq n \\ d \text{ odd}}} \sum_{\substack{q \leq n/d \\ q \text{ odd}}} q \\ &= \sum_{\substack{d \leq n \\ d \text{ odd}}} \frac{1}{4} ((n/d)_0 + 1)^2, \end{aligned}$$

where $(n/d)_0$ is the largest odd integer $\leq n/d$. Then $(n/d)_0 + 1 = n/d + \mu$, where $-1 \leq \mu \leq 1$, and

$$\begin{aligned} \sum_{\substack{1 \leq m \leq n \\ m \text{ odd}}} \sigma(m) &= \frac{n^2}{2} \left(\sum_{\substack{d \leq n \\ d \text{ odd}}} \frac{1}{d^2} \right) + O \left(n \sum_{\substack{d \leq n \\ d \text{ odd}}} \frac{1}{d} \right) \\ &= \frac{n^2}{4} \left(\sum_{\substack{d \leq n \\ d \text{ odd}}} \frac{1}{d^2} \right) + O(n \log n) \\ &= \frac{n^2}{4} \left(\left(\sum_{\substack{d \leq n \\ d \text{ even}}} \frac{1}{d^2} \right) - \left(\sum_{\substack{d \leq n \\ d \text{ odd}}} \frac{1}{d^2} \right) \right) + O(n \log n) \\ &= \frac{n^2}{4} \left(\left(\sum_{\substack{d \leq n \\ d \text{ even}}} \frac{1}{d^2} \right) - \frac{1}{4} \left(\sum_{\substack{d \leq n/2 \\ d \text{ odd}}} \frac{1}{d^2} \right) \right) + O(n \log n) \end{aligned}$$

$$\begin{aligned}
&= \frac{n^2}{4} \left(\left(\frac{\pi^2}{6} + O(1/n) \right) - \frac{1}{4} \left(\frac{\pi^2}{6} + O(1/n) \right) \right) \\
&\quad + O(n \log n) \\
&= \frac{n^2 \pi^2}{32} + O(n \log n).
\end{aligned}$$

The estimates for partial sums are obtained using Euler's summation formula (see Ref. 7, p. 54). To prove

$$\sum_{\substack{1 \leq m \leq n \\ m \text{ odd}}} m\sigma(m) = \frac{\pi^2 n^3}{48} + O(n^2 \log n),$$

we write

$$\begin{aligned}
\sum_{\substack{1 \leq m \leq n \\ m \text{ odd}}} m\sigma(m) &= \sum_{\substack{d \leq n \\ d \text{ odd}}} d \sum_{\substack{qd \leq n \\ q \text{ odd}}} dq \\
&= \sum_{t=1}^{(n-1)/2} \{1 + 3 + 5 + \dots + (2t-1)\} \\
&\quad \cdot \sum_{\substack{n/(2t+1) < d \leq n/(2t-1) \\ d \text{ odd}}} d^2 \\
&= \sum_{t=1}^{(n-1)/2} t^2 \sum_{\substack{n/(2t+1) < d \leq n/(2t-1) \\ d \text{ odd}}} d^2 \\
&= \sum_{t=1}^{(n-1)/2} [t^2 - (t-1)^2] \sum_{\substack{d \leq n/(2t-1) \\ d \text{ odd}}} d^2 \\
&= \sum_{t=1}^{(n-1)/2} (2t-1) \\
&\quad \cdot \left[\frac{1}{6} \frac{n}{2t-1} \left(\frac{n}{2t-1} + 1 \right) \left(\frac{n}{2t-1} + 2 \right) \right].
\end{aligned}$$

The coefficient of n^3 is

$$\frac{1}{6} \sum_{\substack{d \leq (n-1)/2 \\ d \text{ odd}}} \frac{1}{d^2} = \frac{\pi^2}{48} + O(1/n),$$

so

$$\sum_{\substack{1 \leq m \leq n \\ m \text{ odd}}} m\sigma(m) = \frac{\pi^2 n^3}{48} + O(n^2 \log n)$$

as required.

AUTHORS

A. Robert Calderbank, B.Sc. (Mathematics), 1975, Warwick University, England; M.Sc. (Mathematics), 1976, Oxford University; Ph.D. (Mathematics), 1980, California Institute of Technology; AT&T Bell Laboratories, 1980—. Mr. Calderbank is presently employed in the Mathematics and Statistics Research Center. His research interests include applications of mathematics to electrical engineering and to computer science, and algebraic methods in combinatorics.

N. J. A. Sloane, B.E.E., 1959, and B.A. (Hons.), 1960, University of Melbourne, Australia; M.S., 1964, and Ph.D., 1967, Cornell University; AT&T Bell Laboratories, 1969—. Mr. Sloane is the author of four books. He was the editor in chief of the IEEE Transactions on Information Theory from 1978 to 1980. In 1979 Mr. Sloane was awarded the Chauvenet Prize by the Mathematical Association of America. Fellow, IEEE.

100-GHz Measurements of Two Astigmatic Launchers

By R. A. SEMPLAK*

(Manuscript received October 3, 1984)

Astigmatic launchers that would permit a single earth station antenna to communicate with all the satellites along the geosynchronous arc have been fabricated and measured at a frequency of 100 GHz. Good agreement between measured data and calculated values has been obtained for astigmatic corrections required by feeds displaced 18 and 29 degrees from the focus.

I. INTRODUCTION

For high-capacity satellite communication systems, communication satellites are placed at different locations along the geosynchronous arc with the usual practice of using a separate earth station antenna to communicate with each satellite in the system. If both the satellites and earth stations are equipped with multiple-beam antennas, these high-capacity communication systems could be achieved by using a single earth station antenna and simultaneously communicating with all the satellites in the system.¹

Measurements and theory have indicated that the geometry of an offset Cassegrainian antenna results in an ideal configuration²⁻⁵ for both earth station and satellite antennas. Since the antenna aperture has no blockage, this significantly reduces the sidelobe levels and, in turn, reduces interference. However, since only one of the multiple beams can be aimed along the axis of the antenna reflector, the remaining beams must be displaced from the focus. The loss in

* AT&T Bell Laboratories.

Copyright © 1985 AT&T. Photo reproduction for noncommercial use is permitted without payment of royalty provided that each reproduction is done without alteration and that the Journal reference and copyright notice are included on the first page. The title and abstract, but no other portions, of this paper may be copied or distributed royalty free by computer-based and other information-service systems without further permission. Permission to reproduce or republish any other portion of this paper must be obtained from the Editor.

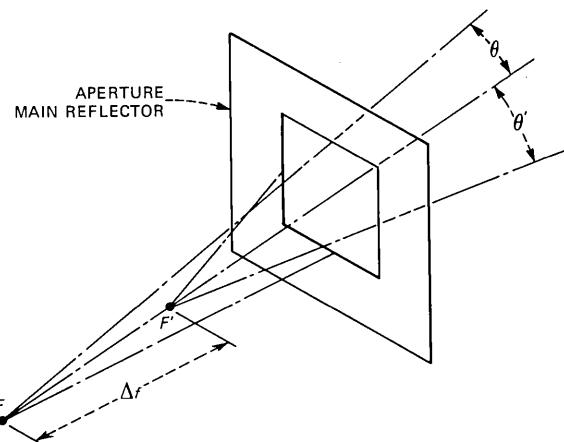


Fig. 1—Astigmatic correction can be obtained by a feed with two different phase centers, F and F' , in the two principal planes of its beam.

efficiency that these displaced beams exhibit is a function of the amount of astigmatism introduced as a result of the displacement from the focus. By using a feed with different phase centers in the two principal planes of its beam, shown in Fig. 1, one can eliminate the astigmatic loss.⁶ For efficient operation over a wide band of frequencies, both the two phase centers (F, F') and the beamwidths in the two principal planes (θ, θ') must be frequency independent.

Earlier work by Dragone⁷ and Chu⁸ shows that frequency-independent astigmatic corrections can be obtained by combining a small horn with two cylindrical reflectors whose focal lengths are such that a magnified image of the feed horn is produced over the main reflector aperture.⁹ However, this feed arrangement is not very suitable for an earth station antenna supporting multiple beams, since the distance between the two phase centers is fixed and cannot be varied after the feed is constructed. If one were to vary this distance, the beamwidths in the two principal planes would change, causing a reduction in aperture efficiency. This is an important restriction, for it implies that a given feed can only be used at certain locations in the vicinity of the focus; at other locations corresponding to other beam displacements, different feed parameters are required, necessitating the design of different feeds for different displacements. In addition, a large feed aperture is required, along with relatively large dimensions for one of the two reflectors.

II. DISCUSSION

Reference 6 describes a single launcher design that overcomes the above difficulties and permits the phase center separation to vary

while maintaining constant beamwidth in the two principal planes. Using the principles of Ref. 6, two launchers (one long and one short) were designed and fabricated for operation at 100 GHz.

The electroformed feed horn used with the launchers is shown in Fig. 2. To permit polarization rotation in the rectangular aperture of the feed horn, the horn was fabricated in two sections—one section tapering down to a square aperture and the second section tapering down to rectangular waveguide. The complete feed horn can be seen mounted together with the mixer on the short astigmatic launcher shown on the right of Fig. 3.

The long astigmatic launcher, shown without feed horn on the left of Fig. 3, has the top parallel plate removed to display the first reflector that would be illuminated by the feed horn. Both the short and long launchers shown here have identical pairs of reflectors; the only difference is the length of the parallel plates.

The cylindrical wave radiated by the feed horn positioned at the focus of the first reflector is guided to the first reflector by the parallel plates. After being reflected, the wave is again guided by the parallel plates in the direction of the second reflector. After some distance the parallel plates are truncated and the aperture illuminated by the reflected cylindrical wave is defined by this truncation. The width of

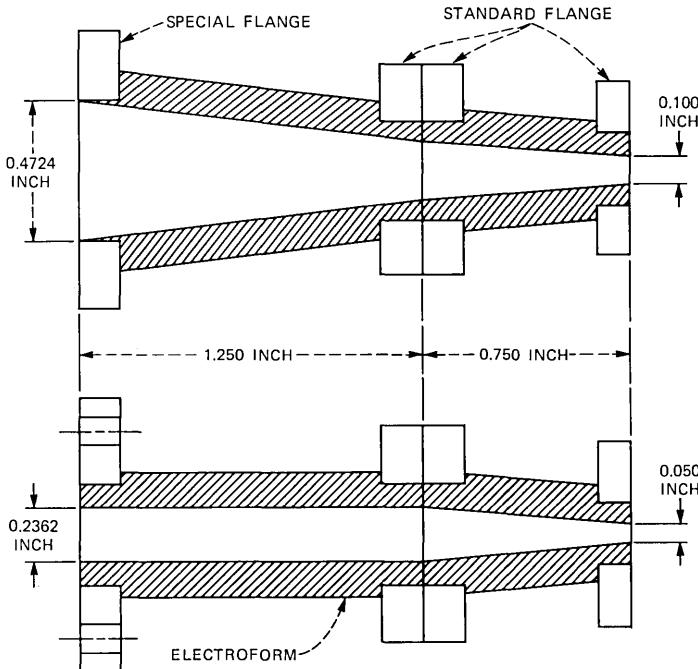


Fig. 2—100-GHz feed horn used with the astigmatic launchers.

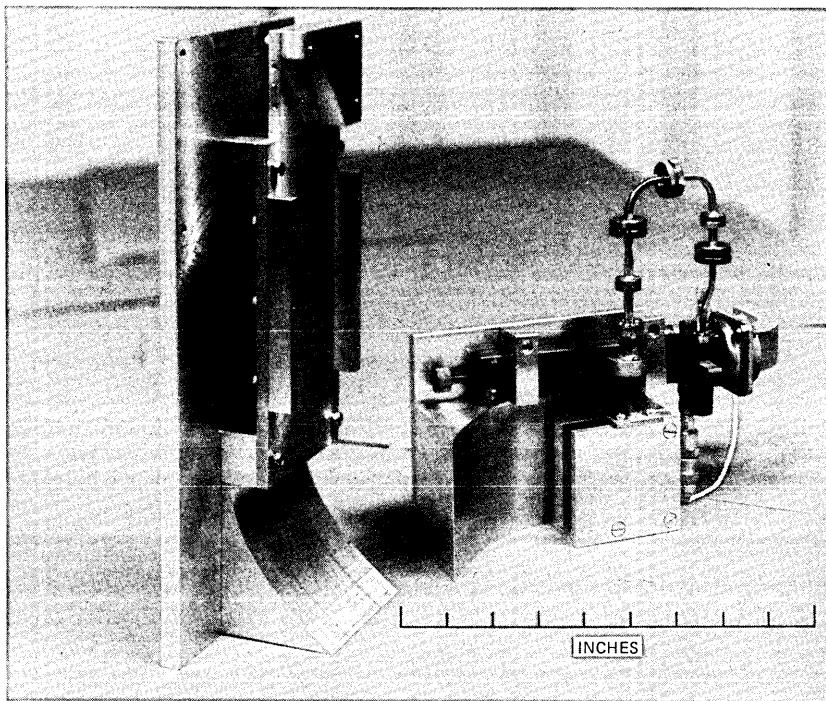


Fig. 3—The short astigmatic launcher (right) and the long astigmatic launcher (left).

the aperture is defined by the spacing between the two parallel plates; the wave radiated by this aperture illuminates the second cylindrical reflector.

To produce an image of the feed horn aperture over the aperture of the main reflector, the distances of the phase centers of the feed horn and the truncated parallel plate aperture must satisfy the optical thin lens equation.⁶

III. ASTIGMATIC LAUNCHER MEASUREMENTS

Using the newly constructed anechoic chamber at the radio range facilities at Holmdel, New Jersey, measurements were made of the radiation characteristics of the two 100-GHz astigmatic launchers depicted in Fig. 3. The measured data,* both amplitude and phase, are presented in Figs. 4 through 7 for both launchers and are shown by the solid curves. The dashed curves are the calculated theoretical values.

* These data were obtained at a distance equivalent to that of the main reflector of an offset Cassegrainian Antenna, i.e, the data represent the actual illumination at the aperture of the main reflector.

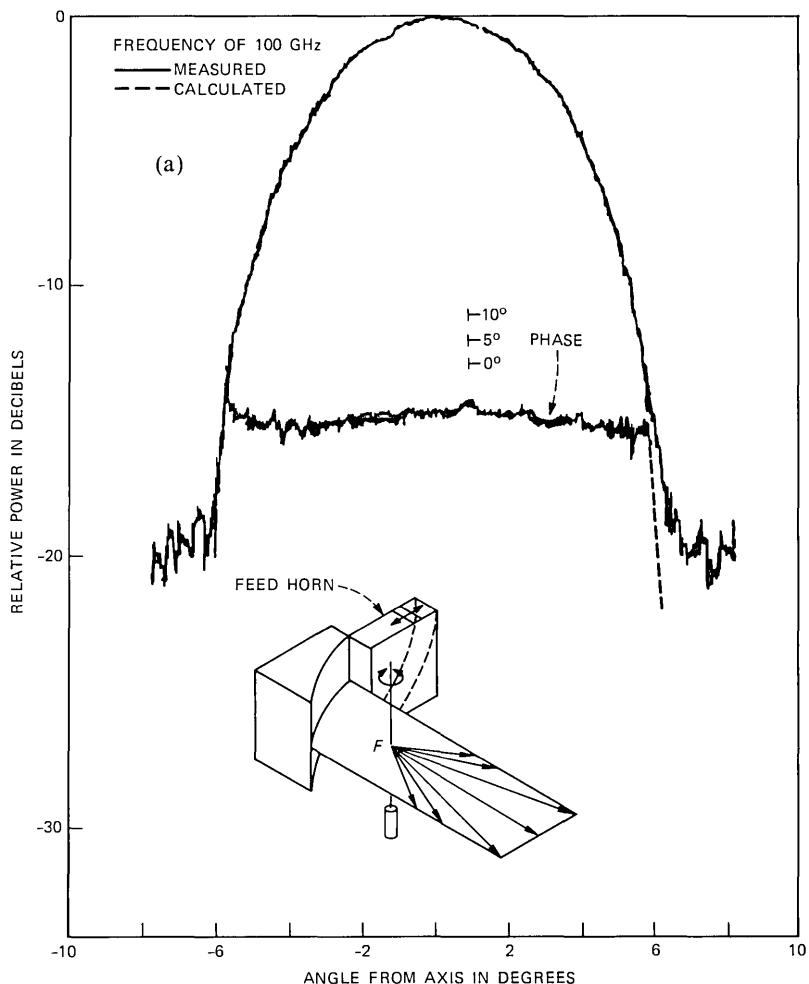


Fig. 4—(a) Measurements for an electric field parallel to the plates of the short launcher, and the theoretical amplitude distribution. The phase center lies in front of the aperture. (Cont.)

To avoid any difficulty in visualizing the polarization of the electric field, the electric field will always be referred to the plane of the parallel plates. Therefore, the electric field will be either parallel to or orthogonal to the plates. Further, the insert in each of these figures shows the position of the launcher with respect to the plane of measurements. The location of the phase center is also shown on the insert.

The amplitude measurements shown by the solid curve of Fig. 4a were obtained with the electric field parallel to the plates. The agree-

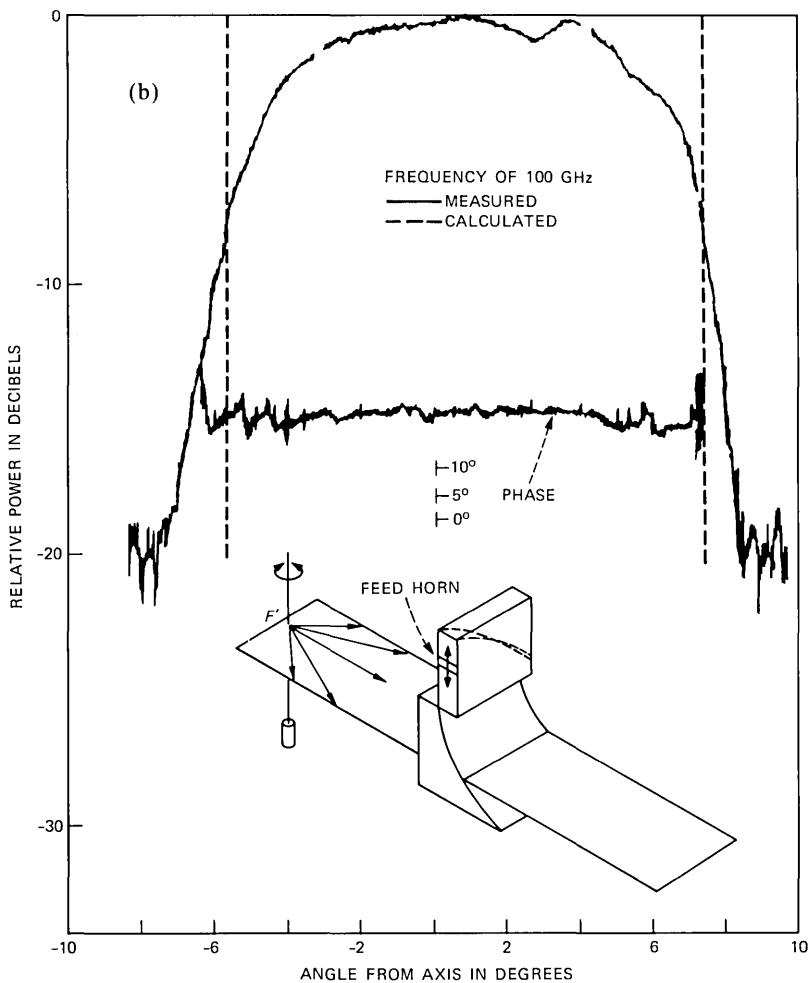


Fig. 4—(b) The electric field remains parallel to the plates, but the launcher is rotated 90 degrees. The phase center lies behind the aperture.

ment with the theoretical calculations is very good. An examination of the phase measurements shows a maximum phase variation of the order 6 degrees. Over much of the aperture, the phase is essentially constant.

For the amplitude data shown by the solid curve of Fig. 4b, the electric field is still parallel to the plates. As shown by the insert on this figure, the launcher is rotated 90 degrees. The dashed curve is that for a uniformly illuminated aperture. The measured data, given by the solid curve, are in good agreement. From the phase data shown here, one sees that the phase change over the aperture is of the order

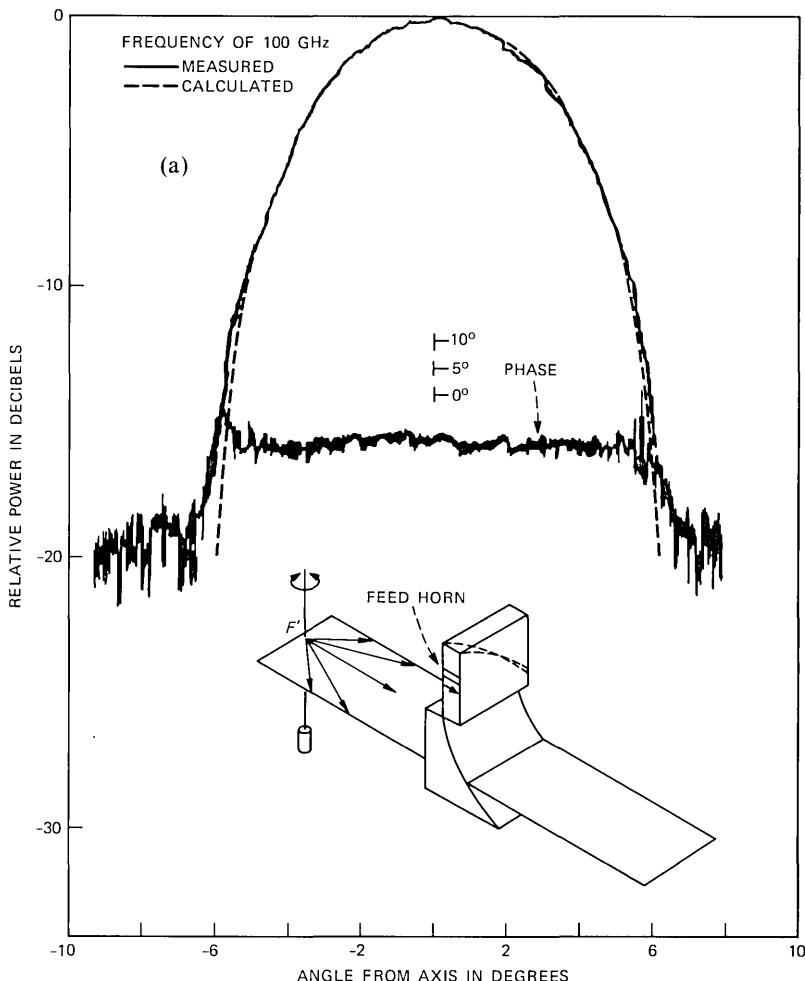


Fig. 5—(a) Measurements for an electric field orthogonal to the plates of the short launcher, and the theoretical amplitude distribution. The phase center lies behind the aperture. (Cont.)

8 degrees. However, over most of the aperture the phase is essentially constant.

Measurements made with the electric field orthogonal to the plates are shown in Figs. 5a and b. As shown here by the dashed curves, the agreement between the calculated values and the measured data is very good. From the phase measurements shown on these figures, one can see that the phase is essentially constant across the aperture. The two pairs of measured data shown by Figs. 4 and 5 are essentially identical.

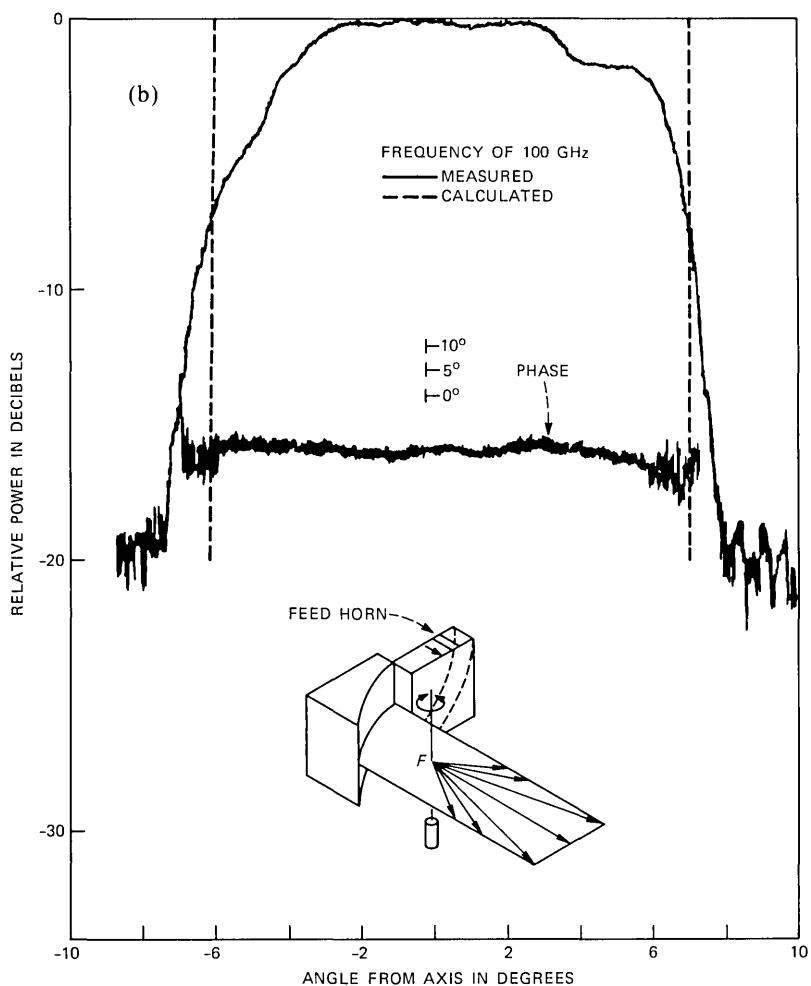


Fig. 5—(b) The electric field remains orthogonal to the plates, but the launcher is rotated 90 degrees. The phase center lies in front of the aperture.

The feed horn and mixer assembly were then transferred to the long astigmatic launcher shown at the left of Fig. 3. The amplitude and phase measurements obtained with the long launcher are shown in Figs. 6 and 7 by the solid curves. Again, the dashed curves are the theoretical calculations.

The data shown in Figs. 6a and b were obtained with the electric field parallel to the plates. As shown by the inserts on these figures, the launcher was rotated 90 degrees to obtain the second set of data. Here again, the measurements agree well with the calculated values. The phase deviations across the aperture are very small.

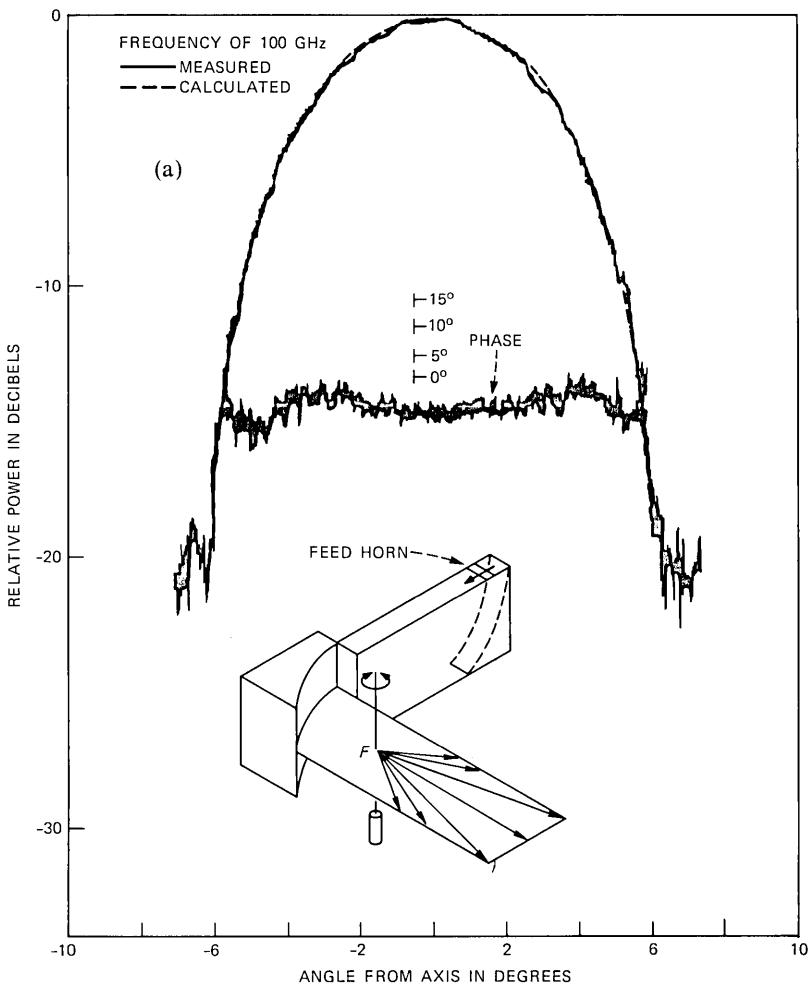


Fig. 6—(a) Measurements for an electric field parallel to the plates of the long launcher, and the theoretical amplitude distribution. The phase center lies in front of the aperture. (Cont.).

The data for the long launcher were completed with the measurements shown in Figs. 7a and b. For these data the electric field is orthogonal to the plates. Again, the agreement between amplitude measurements and theoretical calculations is very good and the phase variations across the aperture are small.

A comparison of the data presented in Figs. 4 through 7 for both the short and long launchers confirms the fact that the phase center separation for this arrangement of launcher can indeed be varied and still maintain constant beamwidth in the two principal planes. The

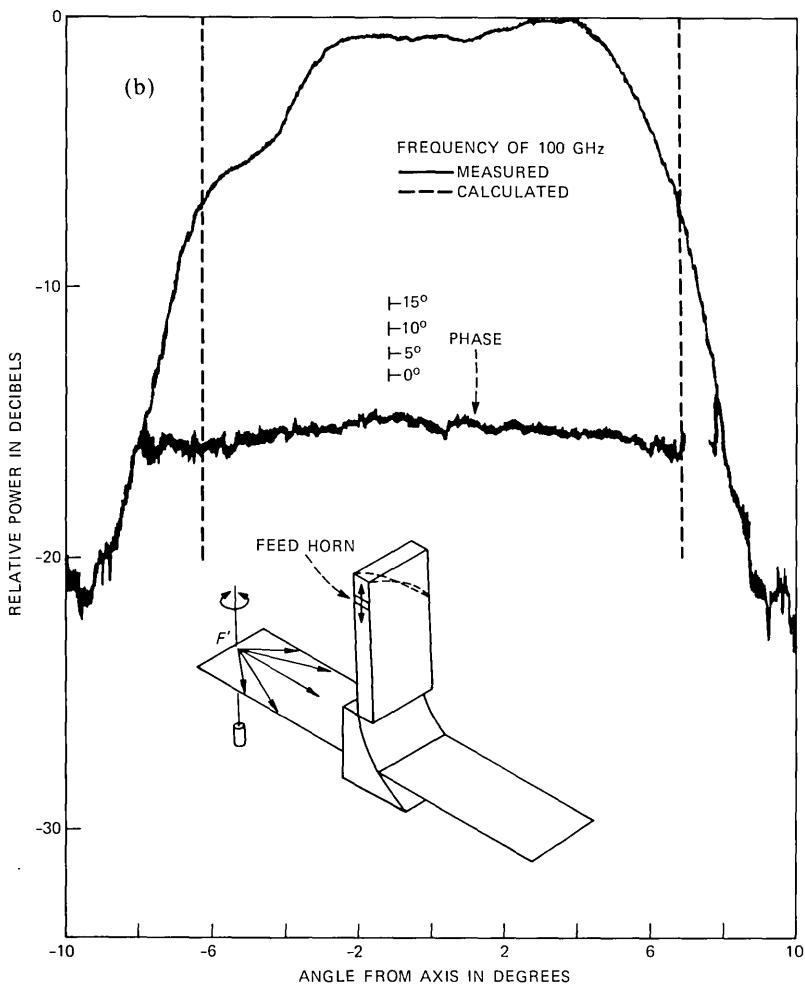


Fig. 6—(b) The electric field remains parallel to the plates, but the launcher is rotated 90 degrees. The phase center lies behind the aperture.

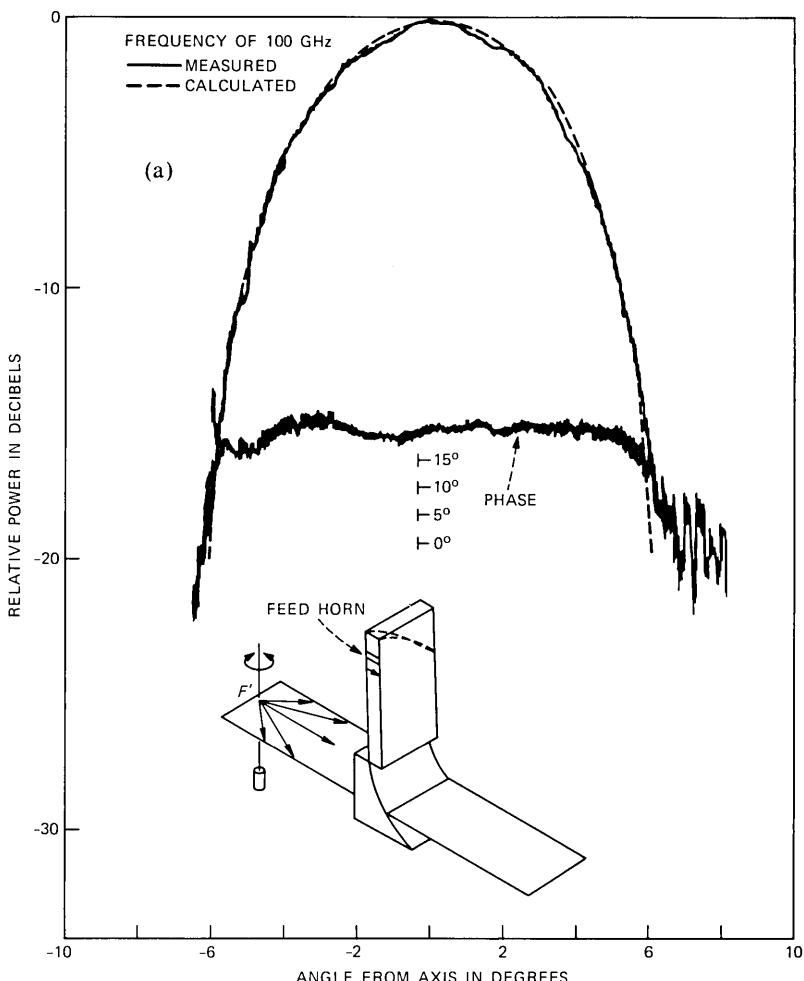


Fig. 7—(a) Measurements for an electric field orthogonal to the plates of the long launcher, and the theoretical amplitude distribution. The phase center lies behind the aperture. (Cont.)

frequency independence of these launchers was checked over a 20-percent band with no discernible change in beamwidth.

Using the methods described in Ref. 10, the measured phase center separation of the short launcher would correct the astigmatism associated with a feed displaced about 18 degrees from the focus, whereas the long launcher would correct the astigmatism associated with a feed displacement of about 29 degrees.

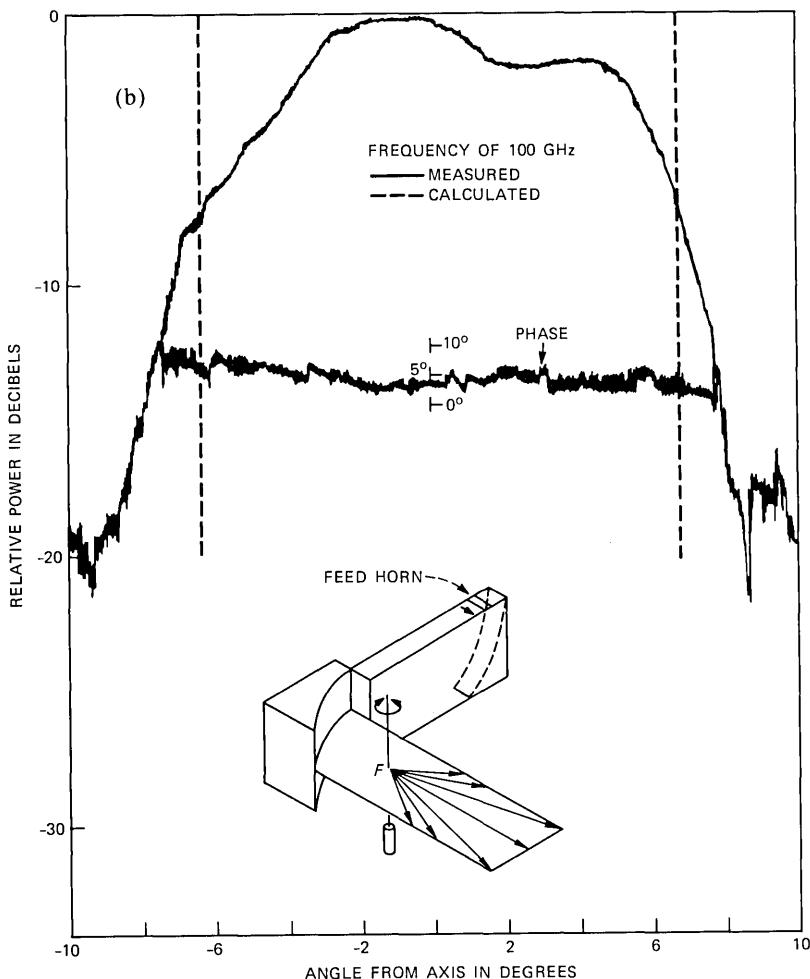


Fig. 7—(b) The electric field remains orthogonal to the plates, but the launcher is rotated 90 degrees. The phase center lies in front of the aperture.

IV. CONCLUSION

Based upon the data presented here, a multiple-beam earth station antenna equipped with launchers of the type described here (where only the separation of the phase centers needs to vary) can indeed communicate with all satellites along the geosynchronous arc.

V. ACKNOWLEDGMENTS

It is with appreciation that I acknowledge G. F. Apgar, for the successful results obtained are due to the attention he gave to the

fabrication of the feed horn and launchers. The design discussions with C. Dragone are acknowledged.

REFERENCES

1. L. C. Tillotson, "A Model of a Domestic Satellite Communication System," B.S.T.J., 47, No. 10 (December 1968), pp. 2111-37.
2. C. Dragone and D. C. Hogg, "The Radiation Pattern and Impedance of Offset and Symmetrical Near-Field Cassegrainian and Gregorian Antennas," IEEE Trans. Ant. Propag., AP-22, No. 3 (May 1974), pp. 472-5.
3. M. J. Gans and R. A. Semplak, "Some Far-Field Studies of an Offset Launcher," B.S.T.J., 54, No. 7 (September 1975), pp. 1319-40.
4. Ta-Shing Chu and R. H. Turrin, "Depolarization Properties of Offset Reflector Antennas," IEEE Trans. Ant. Propag., AP-21, No. 3 (May 1973), pp. 339-45.
5. R. A. Semplak, "100-GHz Measurements on a Multiple-Beam Offset Antenna," B.S.T.J., 56, No. 3 (March 1977), pp. 385-98.
6. C. Dragone and R. A. Semplak, "An Antenna Feed Arrangement for Correcting for Astigmatism," U.S. Patent 4482898, issued November 1984.
7. C. Dragone, "An Improved Antenna for Microwave Radio Systems Consisting of Two Cylindrical Reflectors and a Corrugated Horn," B.S.T.J., 53, No. 7 (September 1974), pp. 1351-77.
8. T. S. Chu, "Broadband Astigmatic Compensation," Dig. 1981 IEEE Ant. Propag. Symp., pp. 131-4.
9. C. Dragone and M. J. Gans, "Imaging Reflector Arrangements to Form a Scanning Beam Using a Small Array," B.S.T.J., 59, No. 3 (March 1980), pp. 449-61.
10. C. Dragone, "First Order Treatment of Aberrations in Cassegrainian and Gregorian Antennas," AP-30, No. 3 (May 1982), pp. 331-9.

AUTHOR

Ralph A. Semplak, B.S. (Physics), 1961, Monmouth College; AT&T Bell Laboratories, 1955-1985. Mr. Semplak's main research interest is in studies of atmospheric effects on micro- and millimeter-wave propagation. He was a member of the Radio Communications Research Department when he retired in 1985. Member, Sigma Xi, Commission F of the International Union of Radio Science (URSI/USNC).

On the Use of Vector Quantization for Connected-Digit Recognition

By S. C. GLINSKI*

(Manuscript received June 6, 1984)

Recent work at AT&T Bell Laboratories has demonstrated the efficacy of vector quantization in greatly reducing both the computational and memory requirements of isolated-word recognition systems. This efficiency is obtained at the expense of a marginal decrease in performance, and thus is an attractive approach. The purpose of this paper is to report on the results of a series of experiments in the application of vector-quantization strategies to a small-vocabulary, connected-word recognition task. Several strategies are investigated, including the use of speaker-trained code books versus universal code books, the use of binary and higher-order tree searches versus full searches of these code books, and the quantization of both test and reference frames versus reference frames only. For various strategies, the effect on error rate of varying the code-book size is also reported. Results indicate that the vector quantization approach is attractive for linear predictive coding-based connected-digit recognition.

I. INTRODUCTION

In the area of speech coding, the technique of Vector Quantization (VQ) has recently been successfully applied.^{1,2} In the standard approach, a speech signal is framed and a feature vector is extracted from each frame. Each element of the feature vector is then separately quantized. In other words, the value of each element is replaced by its closest match from a set of discrete values. The set of values is chosen to minimize some error criterion while reducing the number of bits

* AT&T Bell Laboratories.

Copyright © 1985 AT&T. Photo reproduction for noncommercial use is permitted without payment of royalty provided that each reproduction is done without alteration and that the Journal reference and copyright notice are included on the first page. The title and abstract, but no other portions, of this paper may be copied or distributed royalty free by computer-based and other information-service systems without further permission. Permission to reproduce or republish any other portion of this paper must be obtained from the Editor.

required to identify the element value. This feature vector is typically a set of Linear Predictive Coding (LPC) coefficients and perhaps an energy term. In the VQ approach, a feature vector is extracted as before. The entire vector of features is then quantized by replacing the vector with its closest match from a set or "code book" of feature vectors. Similarly, the entries in the code book are chosen to minimize some distortion measure while reducing the number of bits required to identify each frame of speech. In addition to reducing the number of bits required to represent each feature vector, by using a suitably compact code book in place of a large set of reference vectors, it is possible to greatly reduce the number of comparisons made between the unknown (test) feature vector and the stored-feature vectors. Since this comparison (or distortion measure) is the current bottleneck in many speech recognition systems, the VQ approach is quite effectively used in them. In fact, VQ has been used heavily in several different approaches to speech recognition, including Hidden Markov Modeling (HMM)^{3,4} and Dynamic Time Warping (DTW),³ among others.⁵

The purpose of this paper is to report on the results of a series of experiments in the application of VQ strategies to a small-vocabulary, connected-word recognition task.⁶⁻⁷ These strategies include the use of Speaker-Dependent (SD) versus Speaker-Independent (SI) code books; the use of binary and higher-order tree searches versus full searches of these code books, as suggested in Ref. 2; and the quantization of both test and reference frames versus reference frames only.^{8,9} The effect on error rate of varying the code-book size is also reported. In all experiments, the reference and test data sets are disjoint, and code books are trained with reference data. That is, speaker-dependent reference templates are quantized by vocabulary-dependent code books that are either speaker dependent or speaker independent. All test strings consist of deliberately spoken, connected words.

In Section II some useful terminology is presented. In Section III theory is developed. In Section IV experimental results are presented.

II. TERMINOLOGY

This terminology is based, in part, on Refs. 7 and 10.

p	LPC order
M^*	maximum code-book size in words
M	code-book size in words $2 \leq M \leq M^*$
k	$\log_2 M$ code-book rate
i	code-word (reference) index $0 \leq i \leq M - 1$
$I(k)$	best code-word match at code book k
j	training-frame index $1 \leq j \leq J$
J	number of training frames

B	branching factor
α	$\text{Log}_2 B$
$a_i^k(l)$	LPC reference vector $0 \leq l \leq p$
$r_i^k(l)$	LPC reflection coefficient vector $0 \leq l \leq p$
V_j	autocorrelation matrix of training frame
E_j	LPC residual energy of training frame
$R_j(l)$	autocorrelation vector of training frame
$d(a_i, V_j)$	distortion between reference and training frames
$\rho(l)$	perturbation vector
\bar{D}_J	mean distortion over training frames
ϵ	distortion change threshold (0.01)
δ	perturbation factor (0.01)
$\{T_M(i)\}$	set of training vectors whose best match is code word i
$C_M(i)$	number of training vectors in $\{T_M(i)\}$.

III. VECTOR QUANTIZATION

The VQ procedure consists of two main parts: code-book generation and the classification of test frames. In both cases, a code-book search procedure must be employed to classify the training or test (unknown) frames, respectively. The code-book generation process will be presented first and the search strategy second.

3.1 Code-book generation

The basic generation procedure discussed in Refs. 1, 2, and 10 is employed and is illustrated in Fig. 1. The flowchart is from Ref. 10, but has been generalized to allow B -way splitting of centroids versus the original two-way splitting (B is a power of 2 in this work). The overall goal is to find a set of code words $\{a\}$, such that the mean distortion \bar{D}_J , produced by replacing each of the J training frames by its closest match from the code book $\{a\}$, is minimized. Succinctly stated,

$$\bar{D}_J(M^*) = \min_{\{a\}} \left[\frac{1}{J} \sum_{j=1}^J \min_{1 \leq i \leq M^*} d(a_i, V_j) \right]. \quad (1)$$

The distortion d can be calculated using the likelihood-ratio-distance metric as follows:

$$d(a_i, V_j) = \frac{a_i V_j a_i^t}{a_j V_j a_j^t} - 1, \quad (2)$$

where row m , column n of matrix V_j contains $(1/N)R_j(|m-n|)$. N is the window length and R is the autocorrelation vector. Only the spectral-shape information is used in the quantizer.

In practice, $\{a\}$ is found by first initializing with a small code book

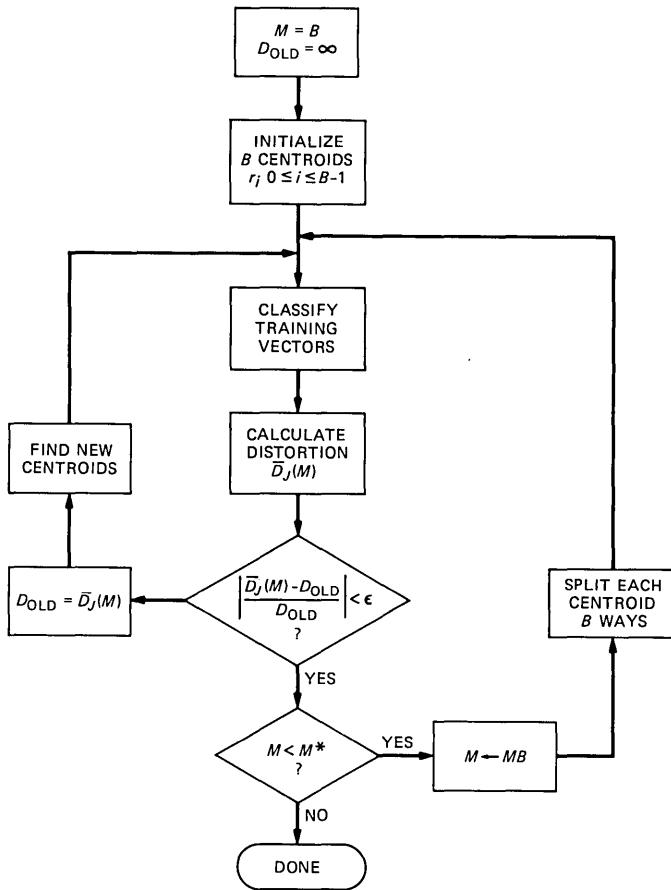


Fig. 1—Code-book generator flowchart.

of size $M = B$ (where B is typically 2), and then successively splitting its code words B ways to obtain larger and larger code books until a code book of the desired size is obtained (see outer loop of Fig. 1). Each successive code book is corrected by iteratively classifying the training set, and adjusting each code word to be the centroid of the subset of training frames which best match that code word (inner loop). The final iteration is determined as that for which the mean distortion changes by less than a threshold ϵ in relation to the previous mean distortion. Typically, $\epsilon = 0.01$.

Initialization of centroids is as follows:

$$\begin{aligned}
 r_i^k(l) &= 0.5(-1)^{l2-i_l} \quad 0 \leq l \leq \log_2 B - 1 \\
 &= 0.0 \quad \log_2 B \leq l \leq p,
 \end{aligned} \tag{3}$$

where $k = \text{Log}_2 B$ and $0 \leq i \leq B - 1$. This splits the LPC reflection coefficient space on the first $\text{Log}_2 B$ coordinate axes. For instance, for $B = 4$,

$$\begin{aligned}r_0^2 &= (0.5, 0.5, 0, \dots, 0) \\r_1^2 &= (-0.5, 0.5, 0, \dots, 0) \\r_2^2 &= (0.5, -0.5, 0, \dots, 0) \\r_3^2 &= (-0.5, -0.5, 0, \dots, 0).\end{aligned}$$

Centroid splitting is done similarly as follows:

$$\begin{aligned}\rho_i(l) &= \delta(-1)^{\lfloor 2^{-l} i \rfloor} \quad 0 \leq l \leq \text{Log}_2 B - 1 \\&= 0.0 \quad \text{Log}_2 B \leq l \leq p \\&\quad 0 \leq i \leq B - 1.\end{aligned}\tag{4}$$

Then

$$\begin{aligned}r_{Bn+i}^{k+\alpha} &= r_n^k \cdot (1 + \rho_i) \quad 0 \leq i \leq B - 1 \\&\quad 0 \leq n \leq 2^k - 1\end{aligned}\tag{5}$$

accomplishes a B -way split of each reflection coefficient vector in code book k . The factor δ is typically set to 0.01. LPC model stability is ensured by requiring that $-1 \leq r \leq 1$.

The new centroid (code-word) computation depicted in Fig. 1 is done by averaging the normalized autocorrelations of those training frames that best match a given code word:

$$R_i(l) = (C_M(i))^{-1} \sum_j E_j^{-1} R_j(l) \quad j \in \{T_M(i)\}.\tag{6}$$

This computes the centroid as a spectral average of training frames.² The total-average-distortion calculation is

$$\bar{D}_J(M) = M^{-1} \sum_{i=1}^M (C_M(i))^{-1} \sum_j d(a_i, V_j) \quad j \in \{T_M(i)\},\tag{7}$$

where the LPC coefficients a_i of eq. (7) are derived from the R_i of eq. (6).

The only program step in Fig. 1 not yet discussed is the classification of training vectors. Since this is common to both the code-book training and ultimate quantization of test vectors, it will be discussed in the next section.

3.2 Classification of frames

In the original development of VQ, a full search of the code book was used to classify training (or test) frames. The best matching code word in a k -bit code book is

$$I(k) = \operatorname{argmin}_i [d(a_i^k, V)] \quad 0 \leq i \leq M^* - 1. \quad (8)$$

For classification of the test, $k = \log_2 M^*$. During the code-book generation, k varies from $\alpha = \log_2 B$ to $\alpha = \log_2 M^*$, depending on how many times the code words have been split.

However, Fig. 2a shows that a binary tree search is also possible. In any given code book (level), only two code words are searched; those that were split from the best matching code word in the previous code book (level). The search is initialized by setting $I(O) = 0$, and then executing

$$I(k) = \operatorname{argmin}_i [d(a_i^k, V)] \quad 1 \leq k \leq \log_2 M^* \\ 2[I(k-1)] \leq i \leq 2[I(k-1)] + 1. \quad (9)$$

The search stops at code book $k = \log_2 M^*$ for test classification, or at an intermediate k during code-book training.

The tree search may be carried out using a different branching factor, B (which in this work is specified to be an integer power of 2). Initialize $I(O) = 0$, and then

$$I(k) = \operatorname{argmin}_i [d(a_i^k, V)] \quad k = \alpha, 2\alpha, \dots, \log_2 M^* \\ B[I(k-\alpha)] \leq i \leq B[I(k-\alpha) + 1] - 1. \quad (10)$$

The stopping criteria are the same as for eq. (9). If $\log_2 M^* \neq \alpha\beta$ for some positive integer β , then the branching factor to the last code book (level) is not $B = 2^\alpha$, but $B = 2^\gamma$, $\gamma < \alpha$, where

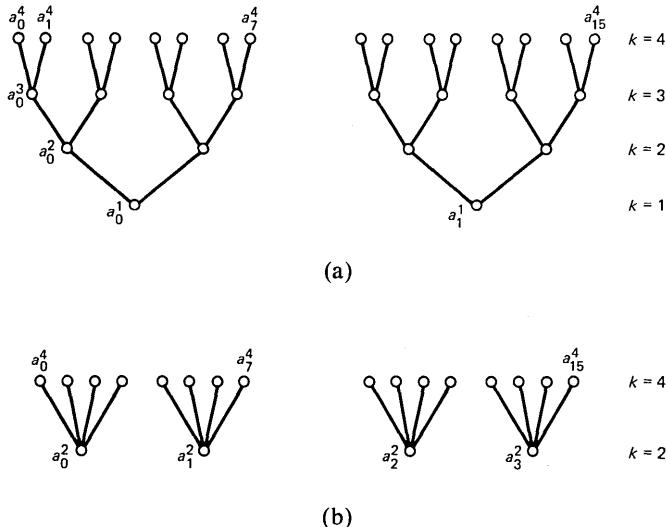
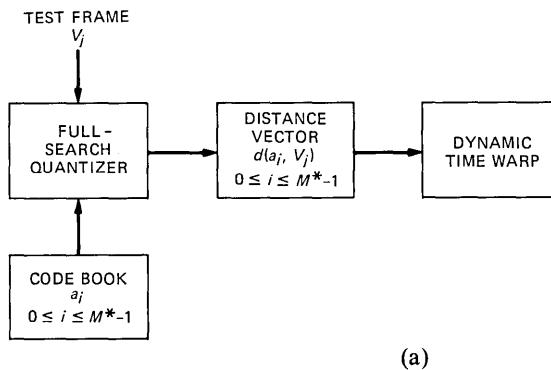


Fig. 2—Code-book tree search: (a) quaternary search; (b) binary search.

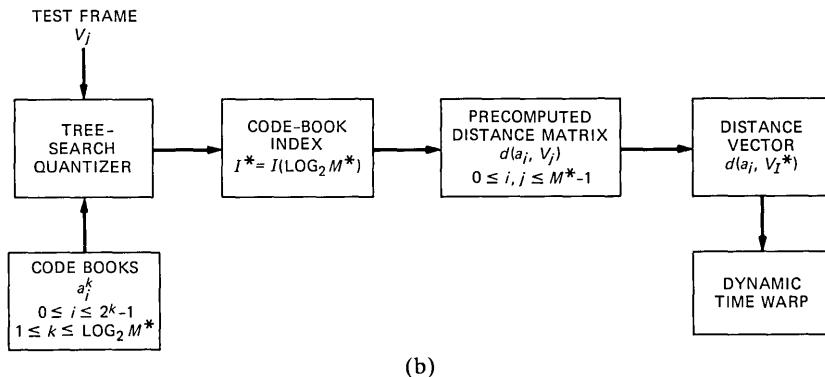
$$\gamma = \left\{ \log_2 M^* - \left\lfloor \frac{\log_2 M^*}{\alpha} \right\rfloor \alpha \right\}. \quad (11)$$

Figure 2b shows an example of the tree-search classification procedure for a branching factor of 4 ($B = 4$).

Note that a full search of a k -bit code book requires $M = 2^k$ distortion computations, while a tree search of the same requires only $B \log_B M$ distortion computations with a branching factor of B . Thus, a great computational savings is incurred by using the tree-search strategy. The two search strategies suggest two approaches to connected-word recognition via DTW, as pictured in Fig. 3. In the following discussion an LPC distortion will be referred to as a distance, which is more common in the jargon of speech recognition. In Fig. 3a (full-search strategy) a vector of distances representing the distances between the current test frame and the entire code book is passed to the DTW procedure.⁸ The vector is sufficient to represent the distance between any test and reference template frames. Since a full search was



(a)



(b)

Fig. 3—Quantizer strategies: (a) full search; (b) tree search.

employed, all these distances are available along with $I(k)$, the best matching code word. In the tree-search strategy (see Fig. 3b) this is not the case. Only the index $I(k)$ and corresponding distance $d(a_{I(k)}, V_j)$ are produced. Thus, it is necessary to precompute and store a cross table of distances between all pairs of code words. This implies that both reference and test frames are quantized to the code-book entries, unlike in Fig. 3a, where solely the reference frames are quantized. Thus, the tree-search strategy will require much less computing power but probably do more poorly than the full-search strategy, because of both the implicit quantization of the test signal and errors introduced by the tree search.

In the HMM approach, however, for the procedure analogous to time warping (Viterbi scoring, for example), the only information necessary to pass from the quantizer is the index $I(k)$ (no distance vector). Thus, there is no need for the precomputed distance matrix in the tree-search strategy, and the only error introduced results from the tree search.

3.3 Empty cluster problem

In practice, during the classification of training frames, it may happen that certain code words may not match any training frame. In other words, the set $\{T_M(i)\}$ may be empty for one or more i , and the count $C_M(i) = 0$ for the same i . This is especially a problem as the code-book size approaches the number of training frames. In this work, the problem is addressed by simply ignoring the empty clusters (which reduces the effective code-book size). During a full-search strategy, the empty clusters are simply skipped. In a tree-search strategy, however, it may happen that as the tree is traversed, a "dead end" is reached before the last level is reached. That is, all branches from a given node lead to code words corresponding to empty clusters. In this case, the test frame is classified as the code word corresponding to the last nonempty cluster encountered as the tree was traversed.

The advantage of this approach is that experiments may be run where the number of training frames is fewer than the specified number of code words. This is especially true for the speaker-dependent code-book experiments in this work.

IV. EXPERIMENTAL RESULTS

To evaluate the various VQ strategies, the connected-word recognizer of Ref. 7 was employed. It can be shown that this recognition algorithm performs identically to the level-building algorithm of Myers and Rabiner¹¹ in the case where no constraint is imposed on the number of words in the test. When this constraint is removed, the level-building algorithm performance is marginally superior.¹² The

distance measure used in the recognizer was the likelihood ratio of eq. (2), clipped at 2.0. The test data consist of 40 random strings, spoken by each of 18 different speakers, 9 male and 9 female. These are equal numbers of strings of length 2, 3, 4, and 5 digits. The strings used in this work are all deliberately spoken (about two words per second).

The reference data consist of 33 reference templates (about 960 reference frames) for each speaker. Three groups of 11 templates each contain the digits 0 through 9, and one repetition of the digit with an unreleased final consonant "t." The first group includes standard isolated-word reference templates; while the second contains embedded, noncoarticulated, deliberately spoken references; and the third contains embedded, coarticulated, normally spoken references. Embedded templates are those extracted from between two other templates in running speech. The three groups are used to compensate for differences in speaking rates and coarticulation effects.

Both the reference and test data were passed through an endpoint detector. All the speech data are identical to those used in previous work.¹³ Note that all experiments entail speaker-dependent, connected-word recognition, and code books are trained with the reference template data. Reference templates and test data sets are disjoint.

For each of the 18 speakers, a speaker-independent code book was generated from the reference templates of the other 17 speakers. These code books are "vocabulary dependent," since the same vocabulary was included in the code-book training data as in reference and test data.

For speaker-dependent code books, the reference database for an individual speaker was used to train the code book for that speaker. This corresponds to an implementation in which reference templates for a given speaker are created first, these templates are then used to train a code book, and then the reference templates are quantized with the same code book. Thus, in this case the code books are both speaker dependent and vocabulary dependent.

Parameters of the code-book generator were chosen as follows:

$$\epsilon = 0.01 \text{ (distortion change threshold)}$$

$$\delta = 0.01 \text{ (perturbation factor).}$$

The baseline string error rate for the recognizer is 5.4 percent. This baseline error rate is for the case where no vector quantization is used.

V. SPEAKER-INDEPENDENT CODE BOOKS

In the first experiment, speaker-independent code books were used to quantize reference frames only (one-sided quantization⁹). Code books of rate 4, 6, 8, and 10 bits were investigated for both binary and

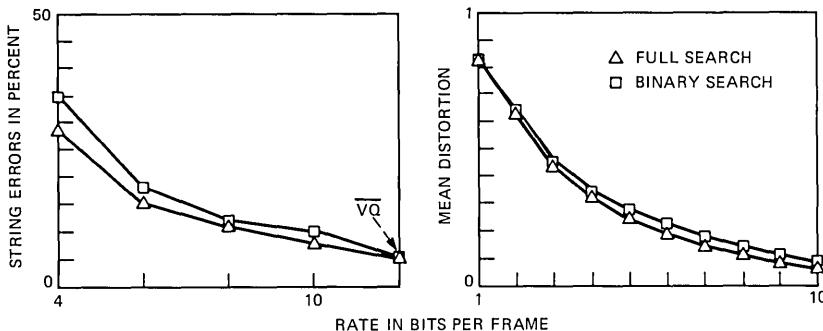


Fig. 4—Recognizer performance and code-book distortion for one-sided VQ and SI code books.

full-search classification during code-book generation and reference quantization. The mean distortion of the binary and full-search code books is shown in Fig. 4. This is the mean distance \bar{D}_j between the training vectors and their best matching code words. The recognizer performance, as a function of code-book rate, is shown in Fig. 4 and listed in Table I. Results show that: (1) Only the use of a 10-bit, full-search code book results in performance approaching the baseline performance. Other rates result in approximately a doubling (or worse) in error rate. (2) About a 1-bit savings in code-book rate is obtained by using a full-search strategy versus a tree-search strategy, while recognizer performance is held constant.

VI. SPEAKER-DEPENDENT CODE BOOKS

The second experiment was identical to the first except that speaker-dependent code books were used in place of speaker-independent code books. The mean distortion of the binary- and full-search code books is shown in Fig. 5. The recognizer performance as a function of code-book rate is shown in Fig. 5, and listed in Table I. Results indicate the following: (1) The recognizer shows little loss in performance at rates of 8 bits and higher. This represents a compression on the reference

Table I—Recognizer performance

Code Book	VQ	Search	Percent String Errors			
			4	6	8	10
SI	1 Side	Full	28.3	15.1	11.0	8.1
SI	1 Side	Binary	34.9	18.2	12.1	10.1
SD	1 Side	Full	15.3	9.3	6.0	5.7
SD	1 Side	Binary	22.1	9.6	6.4	5.6
SD	2 Side	Full	21.0	10.8	7.5	5.8
SD	2 Side	Binary	31.2	14.7	10.7	9.3

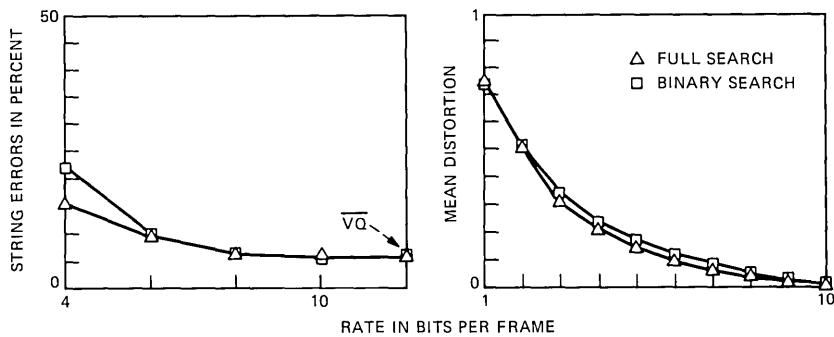


Fig. 5—Recognizer performance and code-book distortion for one-sided VQ and SD code books.

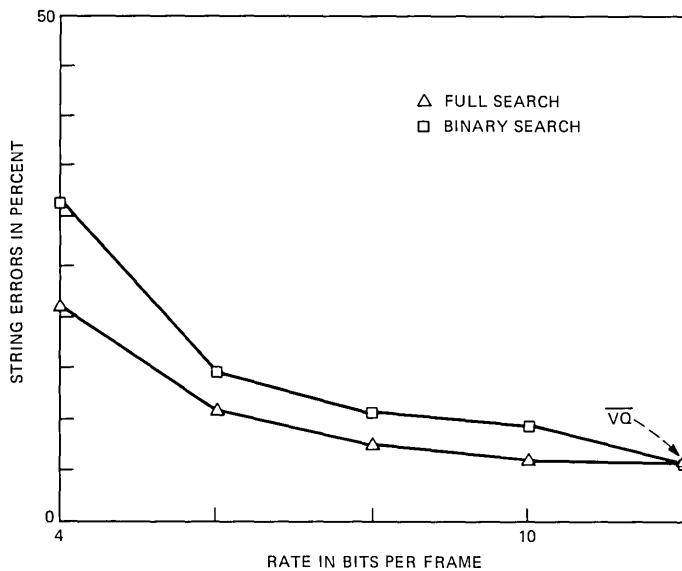


Fig. 6—Recognizer performance for two-sided VQ and SD code books.

database of about 4:1 for each speaker. (2) The binary tree-search strategy rivals the full-search strategy with regard to recognizer performance at rates of 6 bits and up. (3) For a given performance, the SD code book incurred a bit-rate savings from 1 bit (at a low-bit rate) to 3 bits (at a high-bit rate) over the SI code book.

VII. REFERENCE AND TEST QUANTIZATION

The third experiment was identical to the second, except that both reference and test frames were quantized (two-sided quantization⁹). Recognizer performance is plotted in Fig. 6 and listed in Table I. The

Table II—Recognizer performance using tree-search strategies

Branching Factor					
Full search	2	4	8	16	% String errors
6.0	6.4	6.5	6.9	7.4	

results show the following: (1) Two-sided quantization in the full-search case rivals one sided for bit rates of 6 and up. (2) The binary-search case does not compete as favorably with the full-search approach as in experiment 2.

VIII. TREE-SEARCH EXPERIMENT

In experiment 4, experiment 2 was repeated for the 8-bit code-book rate, while the branching factor B in the tree-search approach was varied between 2 and 16. Only reference frames were quantized (SD code book). Performance data listed in Table II and plotted in Fig. 7 indicate that performance drops off only slightly with an increasing branching factor.

IX. CONCLUSIONS

Four experiments were run to determine the efficacy of VQ as applied to a small-vocabulary, connected-word recognition task. Re-

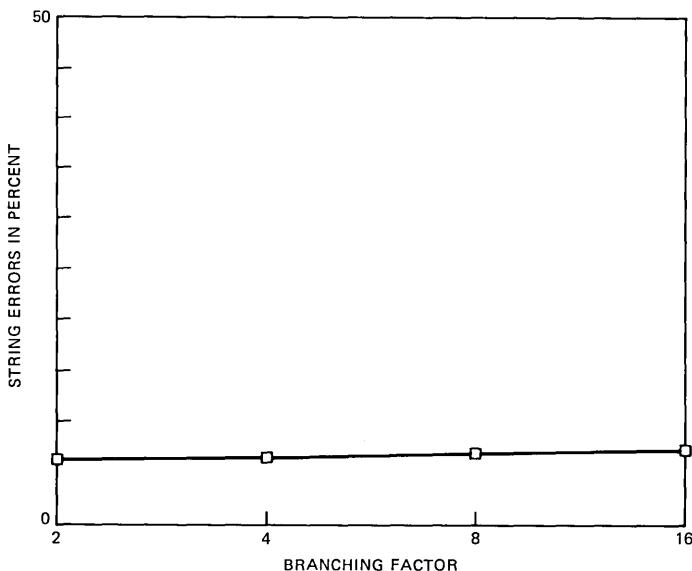


Fig. 7—Recognizer performance for tree-search strategies using one-sided VQ and SD code books at 8 bits per frame.

sults show that both full- and binary-search, one-sided VQ, at code-book rates of 6 bits and higher, offers an attractive cost-performance trade-off in an LPC-based connected-digit recognizer. Full-search two-sided VQ is also a competitive approach. For tree-search strategies, recognizer error rate is relatively insensitive to branching factor.

REFERENCES

1. Y. Linde, A. Buzo, and R. Gray, "An Algorithm for Vector Quantizer Design," IEEE Trans. Commun., COM-28, No. 1 (January 1980), pp. 84-5.
2. A. Buzo et al., "Speech Coding Based Upon Vector Quantization," IEEE Trans. Acoust., Speech, and Signal Processing, ASSP-28, No. 5 (October 1980), pp. 562-74.
3. 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., 62, No. 4 (April 1983), pp. 1075-105.
4. 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 in Automatic Speech Recognition," B.S.T.J., 62, No. 4 (April 1983), pp. 1035-74.
5. J. Shore and D. Burton, "Discrete Utterance Speech Recognition Without Time Alignment," IEEE Trans. Information Theory, IT-29, No. 4 (July 1983), pp. 473-91.
6. T. K. Vintsyuk, "Element-wise Recognition of Continuous Speech Consisting of Words of a Given Vocabulary," Kibernetika (Cybernetics), 7, No. 2 (1971), pp. 361-72.
7. J. S. Bridle, M. D. Brown, and R. M. Chamberlain, "An Algorithm for Connected Word Recognition," Proc. 1982 ICASSP, pp. 899-902.
8. H. Sakoe, "Device for Recognizing and Input Pattern With Approximate Patterns Used for Reference Patterns on Mapping," U.S. Patent 4,256,924, March 17, 1981.
9. K. Shikano, "Spoken Word Recognition Based Upon Vector Quantization of Input Speech," Trans. Committee on Speech Research (December 1982), pp. 473-80.
10. L. R. Rabiner, M. M. Sondhi, and S. E. Levinson, "Note on the Properties of a Vector Quantizer for LPC Coefficients," B.S.T.J., 62, No. 8 (October 1983), pp. 2603-16.
11. C. S. Myers and L. R. Rabiner, "A Level Building Dynamic Time Warp Algorithm for Connected Word Recognition," IEEE Trans. Acoust., Speech, and Signal Processing, ASSP-29, No. 2 (April 1981), pp. 284-97.
12. S. Glinski, "A Comparison of Dynamic Time Warp Algorithms for Connected Word Recognition," paper presented at 106th meeting of Acoustical Society of America, San Diego, Calif., November 1983.
13. L. R. Rabiner, A. Bergh, and J. G. Wilpon, "An Embedded Word Training Procedure for Connected Digit Recognition," Proc. 1982 ICASSP, pp. 1621-4.

AUTHOR

Stephen C. Glinski, B.S. (summa cum laude, Electrical Engineering), 1977, Lowell Technological Institute, Mass.; M.S., 1978, Ph.D., 1981 (Electrical Engineering), University of Illinois at Urbana-Champaign; AT&T Bell Laboratories, 1982—. While at the University of Illinois, Mr. Glinski was a research assistant at the Computer Education Research Laboratory involved in the area of speech synthesis. At AT&T Bell Laboratories, he has been with the Speech Recognition Group engaged in the development of efficient algorithms and architectures for connected-word speech recognition. Member, IEEE, Eta Kappa Nu.

Incorporation of Temporal Structure Into a Vector-Quantization-Based Preprocessor for Speaker-Independent, Isolated-Word Recognition

By A. F. BERGH, F. K. SOONG, and L. R. RABINER*

(Manuscript received November 15, 1984)

Recently a new structure for isolated-word recognition was proposed in which a separate Vector Quantization (VQ) code book was designed for each word in the vocabulary. The word-based VQs were used as a front-end preprocessor to eliminate word candidates whose distortion scores were large; a dynamic time-warping processor then resolved the choice among the remaining word candidates. The above scheme worked very well for small vocabularies; however, the major flaw was the lack of temporal information in the word-based VQ processor. As such, as the vocabulary grew in size and complexity, the ability of the VQ processor to resolve among similar sounding words decreased dramatically, and the effectiveness of the proposed recognition structure similarly decreased. To alleviate this difficulty a technique for incorporating temporal structure into the preprocessor is proposed. In particular, the probability density function of the time of occurrence for each vector in the code book is estimated from a training sequence. In the recognizer, the spectral distance score of the VQ is combined with a temporal distance score, for each frame in the word. An evaluation of the modified recognizer showed slightly improved performance on the digits vocabulary and greatly improved performance on a vocabulary of 129 airlines terms.

I. INTRODUCTION

There has been a great deal of interest recently in isolated-word recognition techniques that maintain high performance, but do so at

* Authors are employees of AT&T Bell Laboratories.

Copyright © 1985 AT&T. Photo reproduction for noncommercial use is permitted without payment of royalty provided that each reproduction is done without alteration and that the Journal reference and copyright notice are included on the first page. The title and abstract, but no other portions, of this paper may be copied or distributed royalty free by computer-based and other information-service systems without further permission. Permission to reproduce or republish any other portion of this paper must be obtained from the Editor.

low computational cost.¹⁻⁵ The reason for this renewed interest in "low-cost" recognizers is the desire to implement such systems on conventional microprocessors, where the computational power is nowhere near as great as needed for the "higher-cost" recognition systems.

One of the most promising of the low-cost recognizers is the Vector-Quantization (VQ)-based recognizer, originally proposed by Shore and Burton,² and modified by Burton et al.⁴ and Pan et al.⁵ The basic idea in this recognition system is to design a separate VQ code book for each word in the vocabulary, based on a training sequence of several tokens of each word by one or more talkers. In the original Shore and Burton implementation,² the recognizer chose the word in the vocabulary whose average quantization distortion (according to its particular code book) was minimum. In the implementation of Pan et al.,⁵ the word-based VQs were used as a front-end preprocessor to eliminate word candidates whose distortion scores were large; a Dynamic Time Warping (DTW) processor then resolved the choice among the remaining word candidates.

Both of the above implementations of the word-based VQ recognizer worked very well for small vocabularies; however, as the vocabulary size and/or complexity grew, the ability of the VQ processor to resolve among similar sounding words decreased dramatically, and the effectiveness of the recognizer similarly decreased.

The major problem with the word-based VQ processor, for large vocabularies, was its inability to use temporal information, i.e., to integrate information about the times of occurrence of the speech sounds with the fact that the sounds occurred within the word. One simple method for incorporating this type of temporal information was proposed by Buzo et al.,⁶ and developed by Burton et al.⁴ In this approach, gross temporal information was incorporated into the recognizer by subdividing each input word into R nonoverlapping regions, using a separate code book for each region. In this manner each word was characterized by R code books, obtained from a training procedure in which a similar subdivision of each training word was made. Burton et al. reported good success with this method.⁴

An alternative procedure for incorporating temporal information into the VQ-based preprocessor is proposed in this paper. In particular, for each vector in each word-based code book, the probability density function of the time of occurrence (on a normalized time scale) is estimated from the same set of training sequences used to derive the code-book vectors. In the recognizer, the spectral distance score of the VQ preprocessor is combined with a (scaled) temporal distance score, for each frame in the word. We use the structure of a preprocessor to screen out unlikely word candidates (based on the combined spectral

and temporal distance), and resolve the fine word distinctions with a DTW processor.

An evaluation of the modified recognizer structure, described above, was performed using both a small vocabulary (the 10 digits), and a moderate-size vocabulary (129 airline terms). Both vocabularies were tested in a speaker-independent mode, i.e., code books and probability histograms were generated from speaker-independent training sets. Results showed recognition error performance on both vocabularies was comparable to that of the best recognizers; however, computational costs were comparable to those of a "low-cost" recognizer.

The organization of this paper is as follows. In Section II we discuss the proposed recognition algorithm, which combines temporal information along with the spectral information of the word-based VQ preprocessor. In Section III we describe an experimental evaluation of the new recognition structure. In Section IV we review the results of the evaluation, and discuss potential ways of lowering the cost of the recognizer even further. Finally, in Section V, we summarize our findings.

II. THE PROPOSED RECOGNITION ALGORITHM

A block diagram of the proposed recognizer is given in Fig. 1. The input speech signal is digitized at a 6.67-kHz rate, the word endpoints (beginning and ending frames) are detected, and a Linear Predictive Coding (LPC) analysis is performed on all frames within the word. The LPC analysis is an eighth-order analysis of 45-ms frames (300 samples), spaced every 15 ms (100 samples) along the word. Each overlapping 45 ms section of speech is windowed using a Hamming window, and an eighth-order autocorrelation analysis is performed (giving nine autocorrelation values per frame). The results of the LPC analysis are the set of frame log energies (suitably normalized to the peak log energy of the word), E_i , $1 \leq i \leq I$, and the LPC vectors a_i , $1 \leq i \leq I$, where I denotes the number of frames in the word.

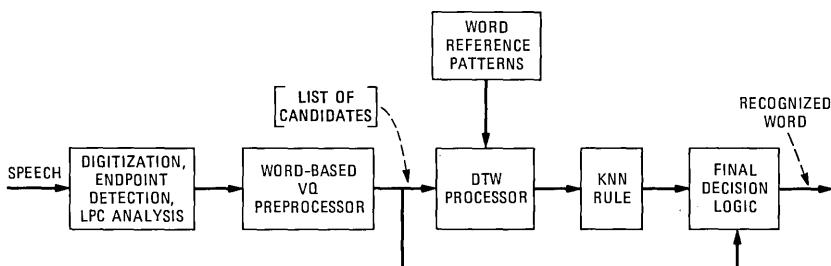


Fig. 1—Block diagram of isolated-word recognizer that incorporates a word-based vector quantization preprocessor and a dynamic time-warping processor.

The word-based LPC preprocessor uses the analysis results (i.e., the frame log energies and the LPC vectors) to eliminate all unlikely candidates from further analysis. Thus the output of the preprocessor is a list of candidates for the unknown word. A DTW processor then decides among the words in the candidate list using a conventional dynamic time-warping alignment of the unknown test word against a set of stored word reference patterns. A K Nearest Neighbor (KNN) decision rule chooses the word whose average DTW distance of the K -best word patterns is smallest. In cases where the list of candidates from the preprocessor contains only a single choice, the DTW processor is bypassed and a final decision is made by the preprocessor.

2.1 The word-based VQ preprocessor

A block diagram of the word-based VQ preprocessor is given in Fig. 2. Each word in the vocabulary is characterized, in the preprocessor, by a code book, B , and by a temporal probability table, P . The code book consists of a set of LPC vectors (supplemented by a log energy scalar), \mathbf{b}_k , $1 \leq k \leq L$, which characterize the LPC vectors of a training set of multiple occurrences of the word. The code-book vectors are chosen by a VQ design algorithm, which minimizes the average distortion between the training vectors and the code-book vectors.⁷⁻⁹ Typically, for word recognition applications, values of L (the total number of vectors in each word code book) range from 4 to 32.

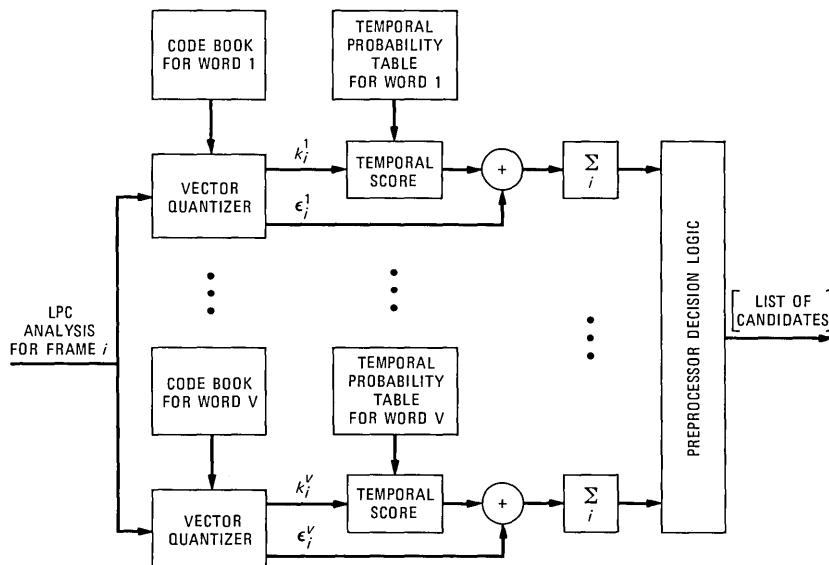


Fig. 2—Block diagram of the word-based vector quantization preprocessor, which combines spectral and temporal distance scores.

The temporal probability table, P , is derived from both the code book, B , and the word training data in the following way. The elements of P are the values $p_k(t)$, defined as:

$$p_k(t) = \text{probability that the code-book vector, } k, \text{ occurs at normalized time } t = i/I \text{ within the word.}$$

Thus the values $p_k(t)$ (where suitably quantized values of t are used in practice) constitute a temporal probability table for the code-book vectors. The way in which values of $p_k(t)$ are obtained, from the training set, is as follows:

1. Each training sequence is linearly warped to a fixed length, $\hat{I} = 40$ frames. (Thus values of $p_k(t)$ are obtained for $t = 1/40, 2/40, \dots, 40/40$.)
2. Each vector of each linearly warped training sequence is vector quantized, using code book B .
3. At each time t , all code-book vectors whose spectral distortion distance score is within a fixed threshold, Δ , of the minimum distortion score for the frame are considered to have occurred.
4. The value used for $p_k(t)$ is the ratio between the number of times code-book vector k occurred at time t (as defined in step 3 above), and the number of times any code-book vector occurred at time t , over the entire training set for the word. In this manner $\sum_{k=1}^L p_k(t) = 1$ for all t .

To illustrate the results of the above procedure, Fig. 3 shows the resulting $p_k(t)$ temporal probability tables for an $L = 8$ vector code book for the word six with a training set of 150 tokens of the word derived from 150 different talkers (75 male, 75 female). A value of $\Delta = 0.25$ was used in computing $p_k(t)$. Experimentation with Δ showed the resulting temporal probability tables were insensitive to Δ over a broad range; this was because with $L = 8$ (or 16) vectors, generally there was only a small fraction of the code-book vectors whose distortion scores were low. Given a large enough training set, the exact value of Δ (as long as it was relatively small) is almost irrelevant.

The temporal probability tables of Fig. 3, for the word six, show that a smooth probability density was obtained for all vectors. Further, we see that for some vectors a unimodal distribution resulted; for other vectors distinct multimodal distributions are found. In this example, the code-book vectors whose sounds represent the vowel /i/ have a unimodal distribution, since this sound occurs only at a single place in the word six. Code-book vectors whose sounds represent the fricative /s/ have a distinct two-mode distribution, since /s/ occurs at both the beginning and end of the word six. Finally, code-book vectors whose sounds represent silence have three modes, since silence can be

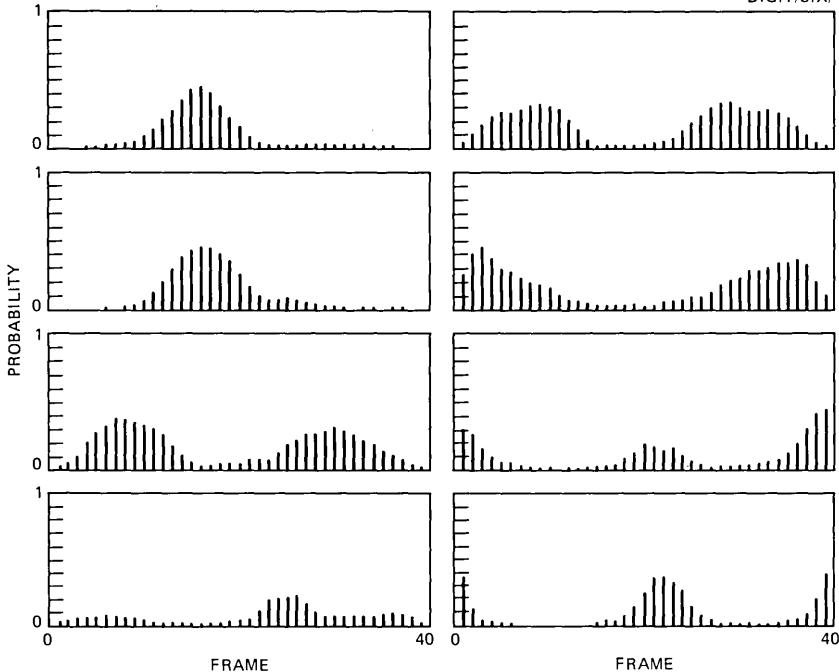


Fig. 3—Estimates of temporal probability density functions for the eight code-book vectors of the digit six. (Forty-frame normalization of the word duration is assumed.)

found at the beginning, end, and in the stop gap of the word six. All three types of distributions are clearly seen in the data of Fig. 3.

For convenience, and to reduce computation, the temporal probability tables were stored as

$$\hat{p}_k(t) = -\gamma \log[p_k(t)], \quad (1)$$

i.e., as negative log probabilities, so they could be combined readily with the LPC distances. The multiplier, γ , was chosen so that, averaged over the entire training set, the average value of $\hat{p}_k(t)$ was the same as the average LPC distance. Typically, the value of γ was about 0.45 for $L = 8$ vector code books, and about 0.22 for $L = 16$ vector code books. Also, values of $p_k(t)$ were clipped at a level of 10^{-4} ; hence no temporal probability score was 0.

2.2 Combining LPC distance and temporal probability score

After a great deal of investigation into ways of combining LPC distance and temporal probability scores, the resulting distance score that was used was

$$d(\mathbf{a}_i, E_i, B, P) = (1 - \alpha)d_{SP}(\mathbf{a}_i, E_i, B) + \alpha d_{TP}(k_i, P), \quad (2)$$

where d_{SP} was the spectral (LPC combined with energy) distance and d_{TP} was the temporal probability distance. The scaling value α was chosen by optimization and determined the mix of spectral and temporal "distances." A value of $\alpha = 0$ represents pure spectral distance; similarly, a value of $\alpha = 1.0$ represents pure "temporal distance."

The spectral distance, which combined the LPC distance with the energy distance, had the form

$$d_{SP}(\mathbf{a}_i, E_i, B) = \min_{1 \leq k \leq L} [d_{LPC}(\mathbf{a}_i, \mathbf{b}_k) + cf(d_E(E_i, \hat{E}_k))], \quad (3)$$

where

$$d_{LPC}(\mathbf{a}_i, \mathbf{b}_k) = \left(\frac{\mathbf{b}'_k V_{\mathbf{a}_i} \mathbf{b}_k}{\mathbf{a}'_i V_{\mathbf{a}_i} \mathbf{a}_i} - 1 \right), \quad (4)$$

with $V_{\mathbf{a}_i}$ being the autocorrelation matrix of the input frame, E_i being the normalized log energy of the input frame, and \hat{E}_k being the normalized log energy of the k th code-book vector. We then have

$$d_E(E_i, \hat{E}_k) = |\hat{E}_k - E_i|, \quad (5)$$

with

$$f(E) = \begin{cases} 0 & 0 \leq E \leq E_{LO} \\ E - E_{LO} & E_{LO} < E \leq E_{HI} \\ E_{HI} - E_{LO} & E_{HI} < E, \end{cases} \quad (6)$$

where c , E_{LO} , E_{HI} , and E_{OF} were suitably chosen constants. (We used $c = 0.1$, $E_{LO} = 6$ dB, and $E_{HI} = 20$ dB.)

The temporal distance of eq. (2) was of the form

$$d_{TP}(k_i, P) = \hat{p}_k([i | I]), \quad (7)$$

where $[i | I]$ is the rounded value of i/I to the nearest 1/40.

The following sequence of steps was required to generate a combined distance score in the preprocessor:

1. Vector quantize the input frame (by each word-based code book), at time $t = i/I$, consisting of LPC vector \mathbf{a}_i and normalized energy E_i , and determine the minimum spectral distance, d_{SP} , and the index of the best code-book vector, k_i .

2. Access the temporal distance as $\hat{p}_{k_i}(t)$, where t is quantized to the nearest 1/40 (since tables with 40 entries were used).

3. Combine d_{SP} and d_{TP} according to eq. (2).

The above procedure is performed at each frame for each word in the vocabulary, and the resulting distance scores are accumulated for each word, as shown in Fig. 2.

The preprocessor decision logic is essentially the same as used by Pan et al.,⁵ namely:

1. Find all word candidates v , such that the average distortion, D^v ,

$$D^v = \frac{1}{I} \sum_{i=1}^I d(\mathbf{a}_i, E_i, B^v, P^v) \quad (8)$$

is within a fixed threshold, δ , of the minimum average distortion across all words.

2. If only a single word candidate exists, then the recognition is over—i.e., no DTW processing is required.

3. If more than one word candidate exists, then use the DTW processor to make the final recognition decision among the word candidates.

We now describe the results of a series of experiments designed to evaluate the performance of the overall recognizer of Figs. 1 and 2.

III. EXPERIMENTAL EVALUATION

Two databases were used to evaluate the performance of the recognizer. All recordings were made over a standard, local, dialed-up telephone line. The first database was a digits set consisting of four sets of 1000 digits each (100 talkers·10 digits/talker). We call the digits sets DIG1, DIG2, DIG3, and DIG4. Their characteristics are as follows:

DIG1—100 talkers (50 male, 50 female), 1 replication of each digit by each talker.¹⁰ These recordings have been used as a training set in a wide variety of evaluations of isolated-word recognizers.

DIG2—Same 100 talkers and recording conditions as DIG1; recordings made several weeks later than those of DIG1.

DIG3—100 new talkers (50 male, 50 female), 1 averaged occurrence of each digit by each talker obtained from averaging a pair of robust tokens of the digit.^{11,12} The transmission conditions (i.e., analog front end, filter cutoff frequencies, etc.) differed slightly from those used in recording the DIG1 and DIG2 databases.

DIG4—A second group of 100 new talkers (50 male, 50 female), 20 recordings of each digit by each talker.¹³ A random sampling of 1 of the recordings of each digit by each talker was used. The transmission conditions differed substantially from those used in recording the other databases.

The templates (12 per word, speaker independent) for the DTW processing were created from the data of set DIG1. The training data for the word-based VQ preprocessor (to get the code books, B^v , and the temporal probability tables, P^v) were derived from a randomly chosen set of 150 tokens of each word from sets DIG1, DIG3, and DIG4. (Of course these same training data could have been used to create the speaker-independent reference templates for DTW process-

ing; however, a conveniently available template set was used.) For testing the recognizer, all four digit sets were used.

The second database was a vocabulary of 129 words used in an airlines information and reservation system.¹⁴ Two sets of data, called AIR1 and AIR2, were used. Their characteristics were:

AIR1—100 talkers (50 male, 50 female), 1 averaged occurrence of each word by each talker obtained from averaging a pair of robust tokens of the word.¹²

AIR2—20 new talkers (10 male, 10 female), 1 replication of each word by each talker. The data of set AIR1 were used to create both the word reference templates (speaker independent, 12 per word), and to give the word code books and word temporal probability tables. The data of set AIR2 were used to test the recognizer.

3.1 Results on the digits vocabulary

For each of the digit test sets, a preliminary test run was performed in which the preprocessor was used by itself to make the final recognition decision based on the word with the lowest combined spectral plus temporal distance score. (Equivalently, δ , in the decision logic, was set to 0.) The distance combining parameter, α , in eq. (2) was then varied from 0 to 1 (in steps of 0.1) and a curve of the preprocessor recognition accuracy versus α was computed. A typical such curve for the test set DIG1 is given in Fig. 4. The behavior of the recognition rate, shown in this figure, is typical for all the digit test sets. It can be seen that for $\alpha = 0$ (only spectral distance) and for $\alpha = 1.0$ (only temporal distance), the recognition rate of the preprocessor (91.4 percent for $\alpha = 0$, 91.2 percent for $\alpha = 1.0$) is significantly lower than its value at the peak of the curve (97.5 percent for $\alpha = 0.7$). This result strongly points out the value of combining spectral and temporal

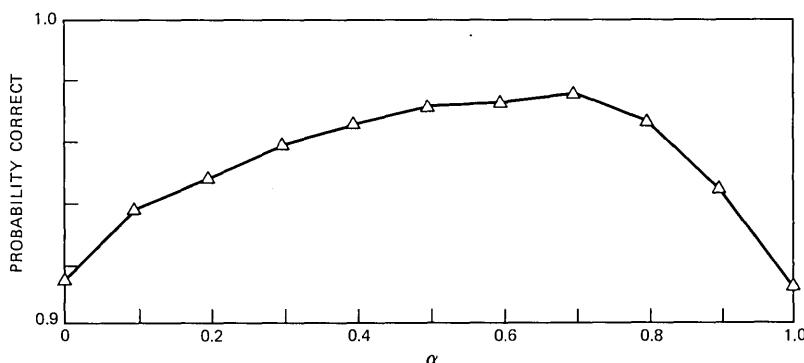


Fig. 4—Curve of average digit recognition rate versus the combining multiplier, α , for the data of test set DIG1. (Note that $\alpha = 0$ corresponds to pure spectral distance and $\alpha = 1.0$ corresponds to pure temporal distance.)

Table I—Average error rates for digits vocabulary

Code- Book Size	Average Digit Error Rate (%)				
	DIG1	DIG2	DIG3	DIG4	Overall
(a) Processor Alone					
8	2.5	3.1	3.1	4.3	3.3
16	2.5	2.6	2.5	3.7	2.8
(b) Complete Recognizer					
8	2.9	2.5	2.2	2.9	2.6
16	1.3	2.3	2.2	2.8	2.2

distances in the preprocessor. It also can be seen that in the vicinity of the peak (near $\alpha = 0.7$), the recognition rate is fairly constant (its value at $\alpha = 0.5$ is 97.1 percent); hence, a fairly broad region of choices for α is possible. Across the four digit test sets, the optimum value of α varied from 0.4 to 0.7. If we used the value $\alpha = 0.5$ for all digit sets, the preprocessor recognition rate changed less than 0.2 percent, on average.

A complete set of performance results on the digits test sets is given in Table I. Table Ia gives average digit error rates for the preprocessor working without the DTW processor, for the four test sets (and an overall average), for code books with 8 and 16 vectors per word. The average digit error rate is 3.3 percent for 8 vector code books, and 2.8 percent for 16 vector code books. Table Ib gives average digit error rates for the complete recognizer, as a function of code-book size. The threshold, δ , in the preprocessor was set so that, on average, about 83 percent of the time no DTW was required (i.e., the preprocessor made the final decision), and about 17 percent of the time, the average number of word candidates passed on to the DTW processor was 2.25. No quantization of the reference templates in the DTW processor was used; previous experience with this data set indicates that no degradation need occur if the reference template quantization is done correctly.⁵

From Table Ib it can be seen that the entire recognizer achieved an average digit error rate of 2.6 percent for $L = 8$ vector code books, and 2.2 percent for $L = 16$ vector code books. These results represent improvements of 0.6 to 0.7 percent in word accuracy; for 4000 test digits, such a result is statistically significant.

3.2 Results on the airline vocabulary

For the airline vocabulary, a curve of preprocessor average performance versus the combining multiplier α was again run, and the results are given in Fig. 5. Although the form of the curve is similar to that of the digits case (Fig. 4), performance improves significantly when

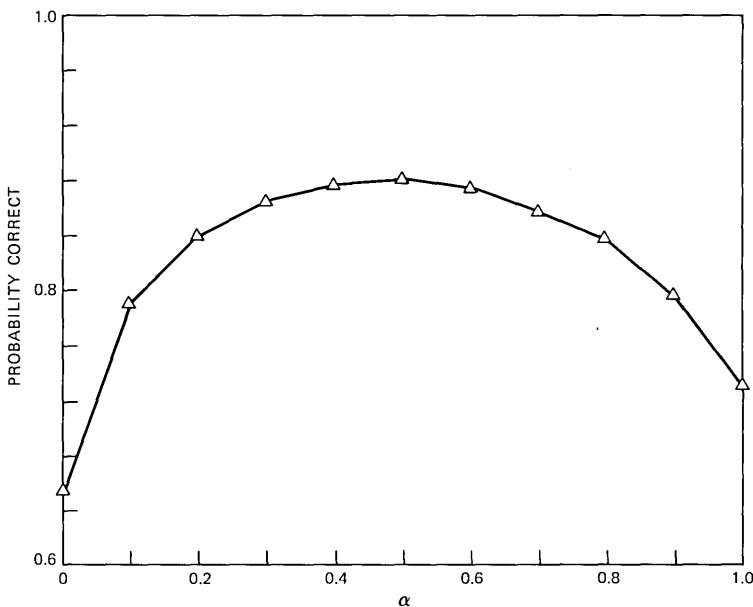


Fig. 5—Curve of average word recognition rate versus the combining multiplier, α , for the airline test data.

Table II—Average word error rates for the airlines vocabulary

Code- Book Size	Average Word Error Rate (%)	
	Preprocessor Alone	Total Recognizer
8	14.8	11.7
16	11.9	8.9

using both spectral and temporal distance, as opposed to either spectral or temporal distance alone. We see from Fig. 5 that for $\alpha = 0$ (spectral distance only), the preprocessor achieves a 65.4-percent accuracy; for $\alpha = 1.0$ (temporal distance only), the accuracy is 73.2 percent (it is better than the result for $\alpha = 0$). However, for $\alpha = 0.5$, the combined distance yields a performance of 88.1-percent word accuracy, an improvement in accuracy of from 15.5 to 22.7 percent over the individual distances.

The overall recognizer performance on the airline vocabulary is given in Table II. The 8-vector-per-word system has a preprocessor error rate of 14.8 percent, whereas the 16-vector-per-word system has

a preprocessor error rate of 11.9 percent. By setting the preprocessor decision threshold so that a unique decision was made by the preprocessor on 76 percent of the trials, and on 24 percent of the trials, an average of 2.5 candidates (out of 129 possible) were passed on to the DTW processor, the overall word error rates fell to 11.7 percent for the 8-vector code books, and to 8.9 percent for the 16-vector code books.

3.3 Typical recognition example

To illustrate how the addition of temporal information aids the preprocessor, Fig. 6 shows a recognition case in which a word (the

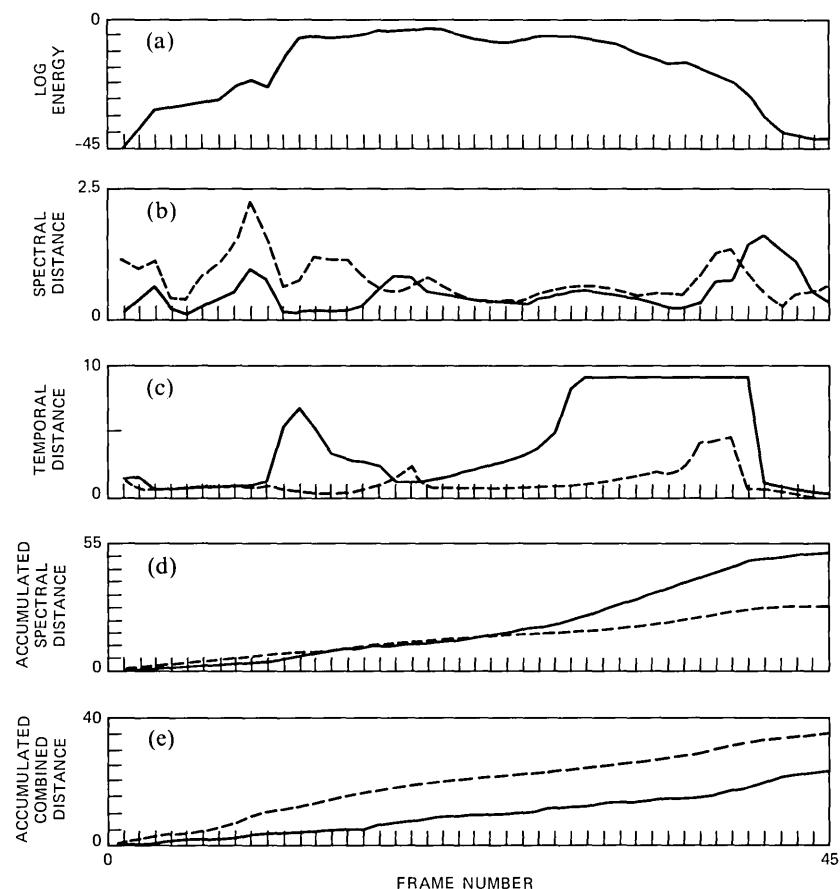


Fig. 6—The enhanced recognition performance obtained by combining temporal and spectral distances in the preprocessor: (a) the test word (zero) log energy contour; (b) spectral and (c) temporal distances on a frame-by-frame basis (solid curve is for the word three, dashed curve is for the correct word zero); (d) accumulated spectral and (e) accumulated combined distance scores.

digit zero) would have been misrecognized (as the digit three) based on VQ spectral distances alone, but is correctly recognized based on the combination of spectral and temporal distance. Shown in this figure are the log energy contour of the test word zero (Fig. 6a), the VQ spectral distance (frame by frame) for both the word zero (dashed line) and the word three (solid line, Fig. 6b), the log probability distances for both words (Fig. 6c), the accumulated spectral distance scores for both words (Fig. 6d), and the combined, accumulated total distance scores for both words (Fig. 6e). On the basis of VQ spectral matches, the preprocessor would have made a hard error since the distance for zero was not close enough to the distance for three; however, using the combined distance the correct word zero was uniquely recognized. The reason that the temporal distance helped so much, in this case, was the large temporal distance during the /O/ vowel in zero for the word three. Thus, although there is a code-book vector that matches the /O/ spectrum well in three, the probability of it occurring at the end of the word is very small. Although this example is an extreme case, it does illustrate well why the addition of temporal information to the preprocessor can help the performance to improve.

IV. DISCUSSION

The results presented in the previous section clearly show that the addition of temporal information to a word-based VQ preprocessor increases the accuracy of the recognizer and makes it more robust to vocabulary size and complexity.

To gain perspective on how the current system performance compares with previous recognizers, Table III gives digit recognition error rates for the current system, for the best DTW recognizer,¹⁵ and for a previous recognizer using a word-based VQ preprocessor.⁵ Similarly, Table IV gives word recognition error rates, for the airline vocabulary, for the current system, for the best DTW recognizer, and for a previous recognizer using a word-based VQ preprocessor.⁵

For the digits, the DTW system performs slightly better, on average, than the current system. However, the best performance is on test sets DIG1 and DIG2, from which the word reference templates were derived. On the test sets DIG3 and DIG4, the current system performed

Table III—Average digit error rates for three recognition systems

Recognizer	Average Digit Error Rate (%)				
	DIG1	DIG2	DIG3	DIG4	Overall
Current system	1.3	2.3	2.2	2.8	2.2
DTW alone	0.0	0.6	2.7	3.9	1.8
Previous recognizer	—	2.0	—	—	—

Table IV—Average word error rates for three recognition systems for the airline vocabulary

Recognizer	Average Word Error Rate (%)
Current system	8.9
DTW alone	10.2
Previous recognizer	12.6

slightly better than the DTW recognizer. On the test set DIG2 (which was the only common one between the current system and the previous recognizer with the preprocessor), the system performances were essentially the same.

For the airline vocabulary we see that the error rate of the current system is 1.3 percent lower than that of the DTW recognizer alone, and 3.7 percent lower than the previous recognizer based on the VQ preprocessor. For this vocabulary a real performance improvement has been achieved.

4.1 Computational considerations

It remains for us to show that this increase in system performance is achieved at essentially no increase in system cost (i.e., computational complexity). To do this we define the following system variables:

L = Code-book size

V = Vocabulary size

I = Average number of frames in a word

Q = Number of templates per word in DTW

p = LPC order

γ = Average fraction of words that are resolved in the preprocessor

β = Average fraction of words passed on to DTW processor, when more than a single word candidate exists.

The computation of the preprocessor can be expressed as:

$$C_{\text{PRE}} = V \cdot I \cdot L \cdot (p + 1) \cdot *, +$$

and the computation of the DTW postprocessor is

$$C_{\text{POST}} = (1 - \gamma)\beta C_{\text{DTW}},$$

where

$$C_{\text{DTW}} = V \cdot Q \cdot \frac{I^2}{3} (p + 1) \cdot *, +.$$

The overall computation of the recognizer is

$$\begin{aligned} C_R &= C_{\text{PRE}} + C_{\text{POST}} \\ &= V \cdot I \cdot (p + 1) \left(L + Q (1 - \gamma) \beta \frac{I}{3} \right). \end{aligned}$$

The ratio between the full DTW computation (without a preprocessor) and the current recognizer computation is then

$$R = \frac{C_{\text{DTW}}}{C_R} = \frac{Q(I/3)}{L + Q(1 - \gamma)\beta \left(\frac{I}{3}\right)}.$$

Substituting typical values of $Q = 12$, $I = 40$, $p = 8$, $L = 8$ (or 16), $(1 - \gamma) = 0.25$, $\beta = 0.02$, we get

$$\begin{aligned} R &\cong 20 & (L = 8) \\ &\cong 10 & (L = 16). \end{aligned}$$

Thus, a computational reduction (over a standard DTW recognizer) of from 10 to 20 times is achieved by the proposed recognizer.

4.2 Further computational reduction via universal code book

Although the performance of the proposed recognizer is impressive, it is possible to reduce its computational complexity even further. If we analyze the computation above, the major computation is in the preprocessor, where a total of $V \cdot L$ dot product distances need to be computed for each test frame. In the case where V is large (e.g., the 129-word airline vocabulary), the total number of code-book vectors becomes large. In such a case it would be less expensive to use a universal code book (word and talker independent) of say 1024 vectors, and to choose the word-based code books from the universal code book. In this manner the number of distance computations per frame is fixed, and does not grow with the vocabulary size V . Of course, it must be shown that performance will not degrade, but it seems reasonable that for a sufficiently large code book, this will indeed be the case.

V. SUMMARY

In this paper we have shown how the addition of temporal information into the preprocessor of an isolated-word recognizer can improve the system performance and make the overall recognizer more robust to vocabulary size and complexity. The way in which the temporal information was added was straightforward; namely, we defined and measured from a training set a probability density function

on the time of occurrence of the code-book vectors in the word-based VQ preprocessor. A temporal distance was defined as the scaled, negative log probability of the probability of occurrence of the vector chosen by the vector quantizer. A combined measure in which the spectral distance (from the VQ) was added to the temporal distance was used in the recognizer and shown to improve performance for both a digits and moderate-size airline vocabulary. Finally, it was shown that, on average, the computational complexity of the resulting recognizer was less than that required for a conventional dynamic time-warping implementation by at least a factor of ten, whereas the recognition performances of the two systems were comparable.

REFERENCES

1. K. Shikano, "Spoken Word Recognition Based Upon Vector Quantization of Input Speech," *Trans. Comm. Speech Res.* (December 1982), pp. 473-80.
2. J. E. Shore and D. K. Burton, "Discrete Utterance Speech Recognition Without Time Alignment," *IEEE Trans. Inform. Theory, IT-29*, No. 4 (July 1983), pp. 473-91.
3. 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.*, 62, No. 4 (April 1983), pp. 1075-105.
4. D. K. Burton, J. T. Buck, and J. E. Shore, "Parameter Selection for Isolated Word Recognition Using Vector Quantization," *Proc. ICASSP 84*, San Diego, Calif. (March 1984), pp. 9.4.1-4.
5. K. C. Pan, F. K. Soong, and L. R. Rabiner, "A Vector Quantization Based Preprocessor for Speaker Independent Isolated Word Recognition," *IEEE Trans. Acoust., Speech, and Signal Processing, ASSP-33*, No. 3 (June 1985).
6. A. Buzo, H. G. Martinez, and C. Riviera, "Discrete Utterance Recognition Based Upon Source Coding Techniques," *Proc. ICASSP 82*, Paris, France (May 1982), pp. 539-42.
7. Y. Linde, A. Buzo, and R. M. Gray, "An Algorithm for Vector Quantization," *IEEE Trans. Commun., COM-28*, No. 1 (January 1980), pp. 84-95.
8. B. Juang, D. Wong, and A. H. Gray, Jr., "Distortion Performance of Vector Quantization for LPC Voice Coding," *IEEE Trans. Acoust., Speech, and Signal Processing, ASSP-30*, No. 2 (April 1982), pp. 294-303.
9. L. R. Rabiner, M. M. Sondhi, and S. E. Levinson, "A Vector Quantizer Combining Energy and LPC Parameters and Its Application to Isolated Word Recognition," *AT&T Bell Lab. Tech. J.*, 63, No. 5 (May-June 1984), pp. 721-35.
10. L. R. Rabiner, S. E. Levinson, A. E. Rosenberg, and J. G. Wilpon, "Speaker Independent Recognition of Isolated Words Using Clustering Techniques," *IEEE Trans. Acoust., Speech, and Signal Processing, ASSP-27*, No. 4 (August 1979), pp. 336-49.
11. 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. 1271-5.
12. J. G. Wilpon, L. R. Rabiner, and A. Bergh, "Speaker Independent Isolated Word Recognition Using a 129-Word Airline Vocabulary," *J. Acoust. Soc. Amer.*, 72, No. 2 (August 1982), pp. 390-6.
13. A. E. Rosenberg, K. L. Shipley, and D. E. Bock, "A Speech Data Base Facility Using a Computer Controlled Cassette Tape Deck," *J. Acoust. Soc. Amer., Suppl. 1*, 72, (Fall 1982), p. 580.
14. S. E. Levinson, and K. L. Shipley, "A Conversational Mode Airline Information and Reservation System Using Speech Input and Output," *B.S.T.J.*, 59, No. 1 (January 1980), pp. 119-37.
15. J. G. Wilpon and L. R. Rabiner, "A Modified K-Means Clustering Algorithm for Use in Speaker Independent Isolated Word Recognition," *IEEE Trans. Acoust., Speech, and Signal Processing, ASSP-33*, No. 3 (June 1985).

AUTHORS

Arthur F. Bergh, currently enrolled at Swarthmore College; AT&T Bell Laboratories, 1980—. Mr. Bergh has been engaged in speech-recognition research in the Acoustics Research Department.

Lawrence R. Rabiner, S.B. and S.M., 1964, Ph.D., 1967 (Electrical Engineering), The Massachusetts Institute of Technology; AT&T Bell Laboratories, 1962—. From 1962 through 1964 Mr. Rabiner participated in the cooperative plan in electrical engineering at AT&T Bell Laboratories, in Whippany and Murray Hill, New Jersey. He worked on digital circuitry, military communications problems, and problems in binaural hearing. Presently, Mr. Rabiner 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), *Digital Processing of Speech Signals* (Prentice-Hall, 1978), and *Multirate Digital Signal Processing* (Prentice-Hall, 1983). Member, National Academy of Engineering, Eta Kappa Nu, Sigma Xi, Tau Beta Pi. Fellow, Acoustical Society of America, IEEE.

Frank K. Soong, B.S., 1973, National Taiwan University, M.S., 1977, University of Rhode Island, Ph.D., 1983, Stanford University, all in Electrical Engineering; AT&T Bell Laboratories, 1982—. From 1973 to 1975 Mr. Soong served as a teacher at the Chinese Naval Engineering School at Tsoying, Taiwan. In 1982 he joined the technical staff at AT&T Bell Laboratories, where he engaged in research in speech coding and recognition. Member, IEEE.

A New Light Pen With Subpixel Accuracy

By M. HATAMIAN* and E. F. BROWN†

(Manuscript received September 13, 1984)

A new light pen system using a real-time algorithm for computing the centroid of the intensity pattern seen by a photosensor is described. This new light pen achieves an accuracy of better than one-quarter of a pixel on a cathode-ray-tube screen with $1K \times 1K$ resolution. This corresponds to a capability of resolving 0.004 inch on a 15-inch screen, which is an improvement by a factor of at least 50 over the conventional light pens available today. The transistor-transistor logic hardware implementation of the algorithm is described in detail. The central part of the hardware is a real-time, moment-generating circuit that implements an efficient moment calculation algorithm reported earlier. Issues such as complexity of the hardware compared with the conventional techniques, and the possibility of implementing the algorithm in a single complementary metal-oxide semiconductor chip are addressed. Some line and curve drawing results sketched by the new light pen are presented and compared to similar drawings obtained by a conventional light pen.

I. INTRODUCTION

The light pen, an input device for graphics displays and work stations, has been around for quite some time. Among the many input devices for graphics systems, the light pen is probably the most natural one to use and, other than the touch-sensitive screen, the only one that directly interacts with the screen. In the world of interactive computer graphics, the light pen has been used mostly as a selection device for pointing to objects and characters on a Cathode-Ray-Tube

* AT&T Bell Laboratories. † AT&T Bell Laboratories; present affiliation Bell Communications Research, Inc.

Copyright © 1985 AT&T. Photo reproduction for noncommercial use is permitted without payment of royalty provided that each reproduction is done without alteration and that the Journal reference and copyright notice are included on the first page. The title and abstract, but no other portions, of this paper may be copied or distributed royalty free by computer-based and other information-service systems without further permission. Permission to reproduce or republish any other portion of this paper must be obtained from the Editor.

(CRT) screen. Although very simple and inexpensive, the light pen has not been very popular among the users of graphics systems. Some have called it a clumsy, fragile device and predicted that it will become less and less popular as raster graphics grows.¹ One of the main reasons for the light pen's poor popularity is its limited resolution, which makes it unsuitable for accurate pointing, or writing and sketching on a CRT screen. Provided with improved resolution capability, the light pen is useful in a variety of applications, such as telewriting, accurate pointing and selecting, facsimile, brushing, and graphics generation. An example of the light pen used in a telewriting system is described in Ref. 2.

The poor resolution of the light pen is a combined result of its relatively large field of view, the signal-to-noise ratio performance of the pen's photosensor, limited image sharpness,³ and, most important, the simple thresholding technique used to detect the position of the pen. We recently proposed a new approach to estimating the position of the pen by processing the analog signal generated by the light pen's photosensor and computing the centroid of the two-dimensional image received by the sensor. Using this new technique we can achieve subpixel accuracies on a $1K \times 1K$ screen. The algorithm is extensively treated in Ref. 4.

This paper describes the real-time hardware implementation of the algorithm. First, we briefly describe the operation of the currently used light pen devices. In Section III we describe the new algorithm. Section IV discusses the hardware implementation and Section V presents some results.

II. CONVENTIONAL APPROACH

Conventional light pens used in today's computer graphics systems employ a very simple circuit for detecting the position of the pen on the display screen. This is illustrated in Fig. 1. A photosensor is placed in the tip of a penlike housing. Whenever the scanning electron beam falls in the photosensor's field of view, it generates an analog signal whose amplitude is a function of the light intensity received by the sensor. This analog signal is compared with a predefined threshold, and a pulse is generated indicating a hit. This pulse latches the values of two counters (the X-Y counters in Fig. 1) that track the horizontal and vertical positions of the beam. The two counters are reset at the beginning of each scan, namely, when the beam is at the top-most left corner of the screen. The X counter tracks the position of the beam on each line and is reset at the end of the line; the Y counter indicates the number of the line currently being scanned by the beam.

Obviously, when the hit pulse is generated, the values of the X and Y counters indicate the coordinates of the hit point. These coordinates

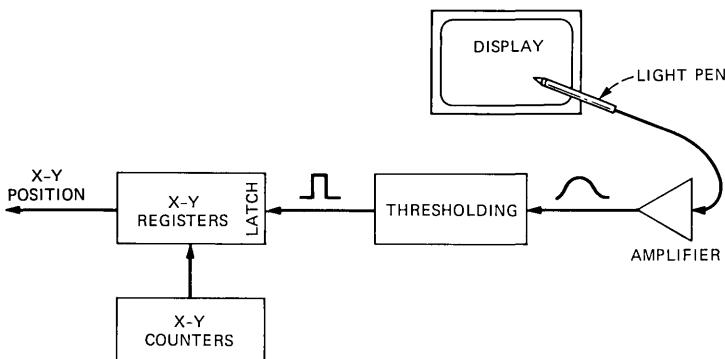


Fig. 1—Conventional light pen system.

are latched into two registers to be used by a host processor (e.g., the graphics station's display processor). Using this simple thresholding technique, the accuracy in estimating the position of the light pen is a function of the photosensor's signal-to-noise ratio and the aperture of the photosensor (size of the field of view). The latter is more important. Many attempts have been made to improve the accuracy of the light pen by controlling the field of view of the pen and using photosensors with high sensitivity.

Designers have employed various mechanical and optical arrangements in the tip of the light pen to improve the accuracy. All of these attempts, although successful in improving the performance, have not achieved the ultimate desired accuracy. To the best of our knowledge, the best of today's available light pens do not have a repeatable accuracy of better than ± 5 pixels (in both directions) on a screen with a resolution of 500×500 pixels. Besides, even this poor accuracy is only achievable when the pen is held perpendicular to the screen and not at an angle.

Such poor accuracy in estimating the position of the light pen is mainly due to the simple thresholding technique used in detecting the hit point. In our approach, a different technique is used. Rather than thresholding the analog signal generated by the photosensor, we take advantage of the information contained in that signal to achieve an accurate and highly robust estimate of the position of the light pen on the display screen. As we will see in the following sections, using the right kind of processing can convert an ordinary light pen into one with an accuracy of one-quarter of a pixel on a display with 1024×1024 resolution.

III. MOMENT COMPUTATION APPROACH

If the analog signal generated by the light pen's photosensor is displayed on a monitor, one sees a cometlike pattern similar to what

is shown in Fig. 2. The tail of the pattern is caused by the persistence of the phosphor used in the CRT display at which the light pen is pointed. A long persistence phosphor will cause a longer tail. In our new light pen, the centroid of this intensity pattern is used as the estimate of the position of the light pen. As shown in Ref. 4, this approach produces a highly accurate and robust estimate of the position.

The x and y coordinates of the centroid are calculated in real time (60 fields/s) by computing various moments of the intensity pattern. The scheme is illustrated in Fig. 3. After proper amplification, the

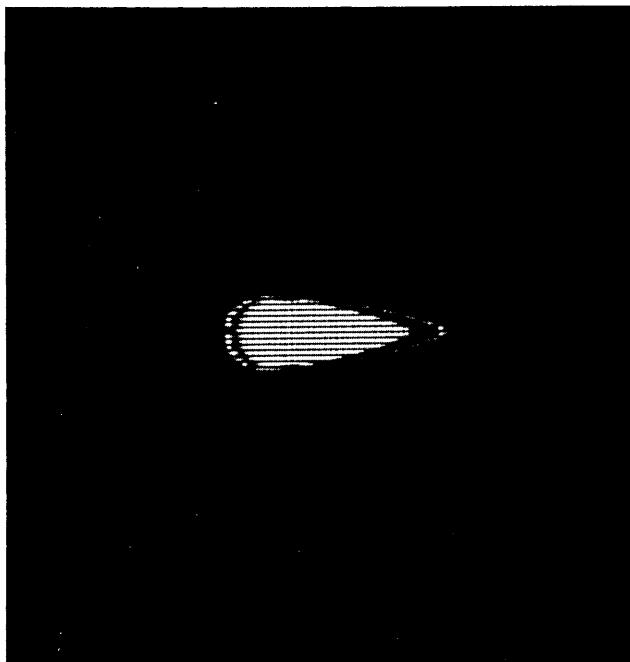


Fig. 2—Intensity pattern generated by the light pen's photosensor.

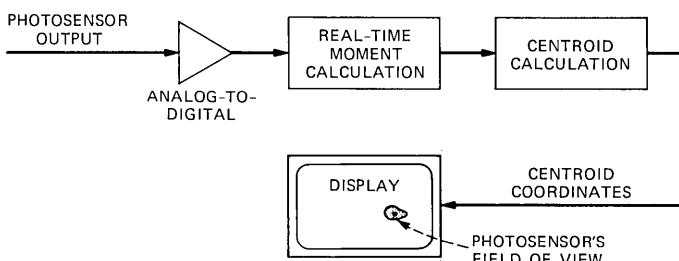


Fig. 3—Light pen system using the moment computation approach.

photosensor's output is digitized and fed to a two-dimensional digital filter, which can compute the intensity moments of the two-dimensional signal represented by the photosensor's output. The moments m^{pq} about the point (N, M) are defined as

$$m^{pq} = \sum_{i=0}^N \sum_{j=0}^M x(i, j)(i - N)^p(j - M)^q, \quad (1)$$

where $x(i, j)$ represent the digitized samples of the sensor's output, and N and M are the horizontal and vertical resolution of the display system. We recently proposed a very efficient algorithm for real-time computation of (1). This algorithm is described and analyzed in Ref. 5. It uses a set of identical single-pole digital filters and has a highly regular and expandable structure. In Ref. 5 we also present a Very Large-Scale-Integrated (VLSI) design for single-chip implementation of this algorithm in Complementary Metal-Oxide Semiconductor (CMOS) technology. The chip is capable of simultaneously computing 16 moments $m^{p,q}$ ($p = 0, 1, 2, 3$; $q = 0, 1, 2, 3$) of a 512×512 , 8 b/pixel image in real time (i.e., at conventional video rate).

In this application for estimating the position of the light pen, only the three moments $m^{0,0}$, $m^{0,1}$, and $m^{1,0}$ are used to compute the centroid coordinates as

$$\begin{aligned} x_c &= \frac{m^{1,0}}{m^{0,0}} \\ y_c &= \frac{m^{0,1}}{m^{0,0}}. \end{aligned} \quad (2)$$

The subpixel accuracy in our estimation stems from the fact that the above division operations can be carried out in floating point mode. This generates results accurate to almost one digit after the decimal point, as we will see in Section V.

IV. HARDWARE IMPLEMENTATION

The light pen algorithm described above has been implemented as part of a larger graphics test bed described in Ref. 6. Figure 4 shows the light pen system used in the test bed. Except for the moment generator circuit, the rest of the blocks in this figure are part of the graphics test bed. As described earlier, the signal generated by the light pen's photosensor is digitized and fed to the moment generator circuit, which generates a set of raw moments at the rate of 60 times a second (video field rate). These moments are read by the test bed's control processor (a 68000 microprocessor), which calculates the centroid coordinates and feeds them to the display processor for proper action (e.g., writing, erasing, and brushing).

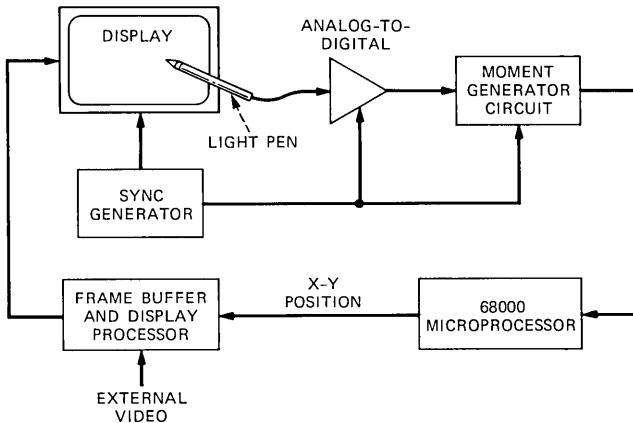


Fig. 4—Light pen system using the moment approach implemented on a graphics test bed.

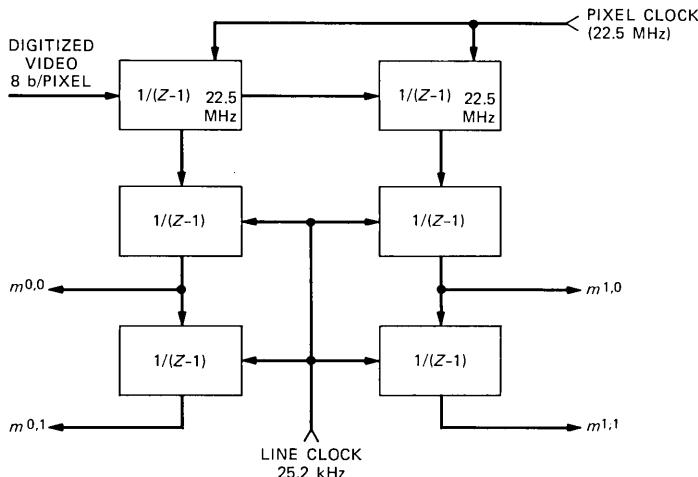


Fig. 5—Digital filter representation of the moment computation algorithm.

The moment generator circuit is basically a Transistor-Transistor Logic (TTL) implementation of the algorithm we described in Ref. 4. It consists of a number of single-pole digital filters, with transfer function $1/(Z - 1)$, interconnected as shown in Fig. 5. This is only part of a larger filter network described in Ref. 5, mainly because higher-order moments are not required in this application. Each filter block is simply implemented by an accumulator, which is basically an adder with an output register and a feedback path. This is shown in Fig. 6 for the two blocks in the first row of Fig. 5, referred to as a row filter in Ref. 5. These two blocks operate at the pixel clock rate. Figure

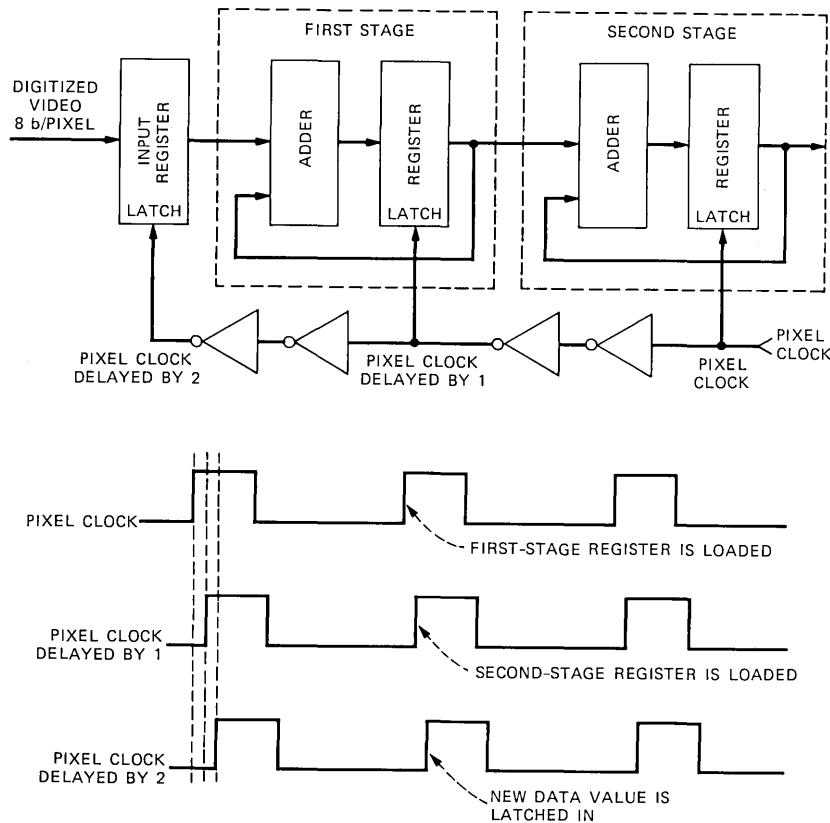


Fig. 6—Circuit and timing diagram of the row filter section of the moment generator.

6 illustrates the operation of this simple arrangement of adders and registers. The rest of the blocks in Fig. 5, referred to as column filters, are implemented in exactly the same way, except that they run at the display's scan-line clock rate rather than the much faster pixel clock rate. This feature relaxes the speed requirement on the adders in the column filter to the point where they can be implemented serially. This results in considerable savings in space and power consumption. This feature is extremely important in the VLSI implementation of the moment generator circuit, as discussed in Ref. 5.

V. RESULTS

Results obtained from the hardware implementation of our new light pen algorithm indicate that the moment computation approach can indeed achieve subpixel accuracy in estimating the position of the light pen. All the line and curve drawings presented in this section

were drawn by the light pen on a 1000-line raster scan black and white monitor that uses a short persistence phosphor (2 to 3 μ s). For a display monitor with a long persistence phosphor, the output of the photosensor's amplifier should be differentiated and rectified before processing. See Ref. 4 for more details.

Figure 7 shows a number of curves and lines drawn by the light pen using the moment computation approach. Figure 8 shows similar drawings using the conventional method in determining the position

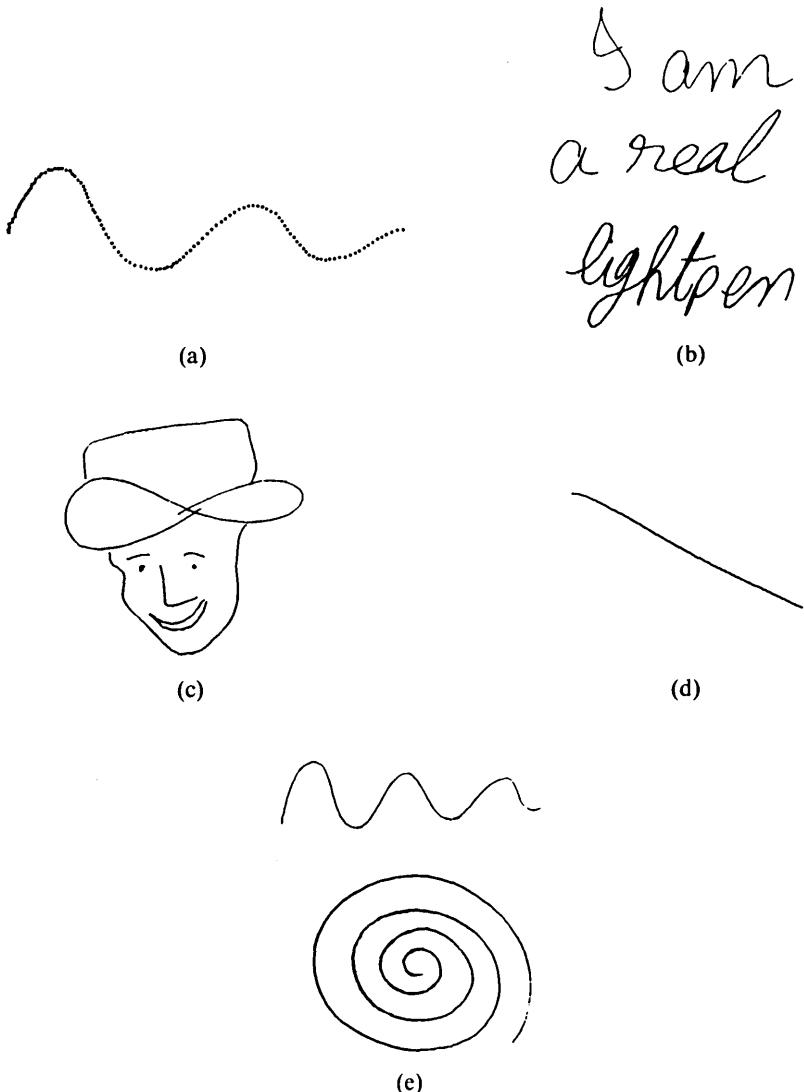


Fig. 7—Light pen drawings using the moment computation approach.

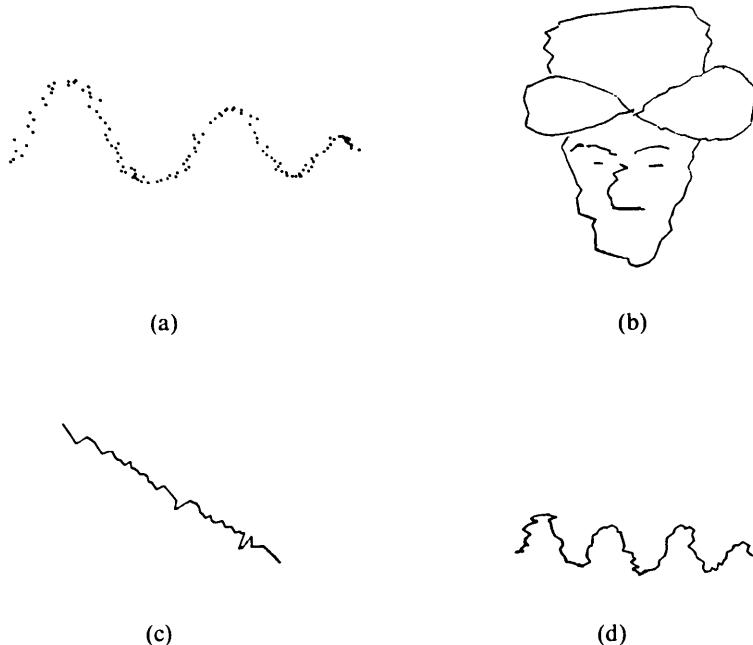


Fig. 8—Light pen drawings using the conventional technique.

of the light pen (same photosensor was used). Except for Figs. 7a and 8a, a linear interpolation has been used for connecting the points. A comparison of the drawings in Figs. 7 and 8 clearly shows the degree of improvement achieved by the moment computation approach. Figure 7b is of special interest. It shows that, even with a simple linear interpolation, light pens using this new algorithm can generate very high-quality writing on CRT displays.

To obtain some quantitative measure of the accuracy, the following experiment was performed. The light pen was attached to a micropositioning device facing the display screen. The micropositioner was moved horizontally in steps of 0.001 inch, and the floating point x and y coordinates generated by the moment processor were recorded. The results are plotted in Fig. 9. From these results we can observe an accuracy of one-quarter of a pixel on a $1K \times 1K$ resolution monitor. The pen can resolve 0.004 of an inch on a 15-inch screen. It should be noted that, although the resolution of our display system is $1K \times 1K$, the signal fed to the moment generator circuit is subsampled by a factor of 2, resulting in an effective input resolution of about 500×500 . It should also be noted that part of the error in the recorded data for the x and y coordinates is due to the jitter in the mechanical setup in the above experiment.

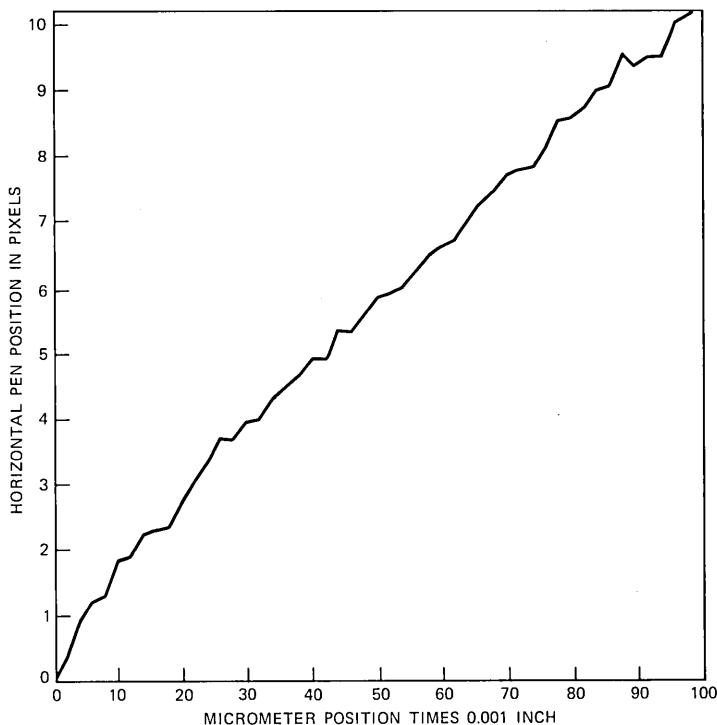


Fig. 9—Position computed by the moment algorithm in pixels as a function of the movement of the pen by a micropositioner.

To be of practical use in today's graphics systems, the cost of this new light pen must be drastically reduced. We are currently studying the possibility of implementing the algorithm in a single-hybrid CMOS chip. This would include the Analog-to-Digital (A/D), moment generator circuit, and postprocessing. A VLSI design for the moment generator part of this chip has already been prepared.⁵ In applications where an accuracy of one pixel is sufficient, the requirement on the size of the A/D and the adder circuits in the moment generator can be considerably relaxed. Preliminary investigations indicate that the moments of the area of the pattern in Fig. 2, rather than the intensity moments, are sufficient for one-pixel accuracy. Using the area moment, if possible, reduces the A/D to a simple comparator (1 b/pixel), and the moment generator requires much less silicon area. All these options are being investigated and will be reported in the future.

VI. CONCLUSION

A new light pen system based on computing the centroid of an intensity pattern generated by a photosensor was presented. The

hardware implementation of this system was described in detail. The central part of the hardware is a TTL implementation of an efficient, real-time, moment-generating algorithm reported earlier. The moments are used to compute the x and y coordinates of the centroid as an estimate of the position of the light pen on the screen. It was shown that this new technique can achieve an accuracy of at least one-quarter of a pixel on a display screen with a resolution of $1K \times 1K$. The pen can thus resolve 0.004 inch on a 15-inch screen. This is an improvement by a factor of at least 50 over the conventional techniques used in today's light pens. Such an improvement is gained at the cost of increasing the complexity of the hardware by a considerable factor. However, due to the simplicity of our moment-generating algorithm and the regularity of its structure, the required hardware can be implemented in a single CMOS chip. Such a chip can have many potential applications in the field of computer graphics. Work on this single-chip implementation of the algorithm is in progress.

REFERENCES

1. J. D. Foley and A. Van Dan, *Fundamentals of Interactive Computer Graphics*, Reading, Mass.: Addison-Wesley, 1982.
2. P. Zorkoczy, *Information Technology—An Introduction*, Knowledge Industry Publications, 1983.
3. J. E. Scott, *Introduction to Interactive Computer Graphics*, New York: Wiley, 1982.
4. Z. L. Budrikis and A. N. Netravali, private communication.
5. Z. L. Budrikis and M. Hatamian, "Moment Calculation by Digital Filters," AT&T Bell Lab. Tech. J., 63, No. 2 (February 1984), pp. 217-29.
6. E. F. Brown et al., "An Interactive Graphics Conferencing Test-Bed," ICC-84 Conf., Amsterdam.

AUTHORS

Earl F. Brown, 1955, RCA Institute; AT&T Bell Laboratories, 1955-1984. Present affiliation Bell Communications Research, Inc. At AT&T Bell Laboratories Mr. Brown was engaged in image processing bandwidth compression, image quality studies, video teleconferencing, and audio teleconferencing. At Bell Communications Research he is a member of the Wideband Data and Video Services Division.

Mehdi Hatamian, B.S.E.E., 1977, Ary-Mehr University of Technology, Tehran; M.S. and Ph.D. (Electrical Engineering), University of Michigan, Ann Arbor, in 1978 and 1982, respectively; AT&T Bell Laboratories, 1982—. From 1979 to 1982 Mr. Hatamian was a Research Assistant on a NASA project working on the design and development of hardware and software for one of the space shuttle experiments. His research interests are in real-time signal processing, circuit design, image processing, and VLSI design. Member, IEEE, Sigma Xi.

Sojourn Time Distribution in a Multiprogrammed Computer System

By K. M. REGE and B. SENGUPTA*

(Manuscript received July 5, 1984)

We present a method for calculating the moments and the distribution of sojourn time in a multiprogrammed computer system. We assume that the CPU and I/O subsystem can be represented by a general state-dependent server who works according to the processor sharing discipline. Further, at most m jobs may be simultaneously receiving service. Thus, m is the multiprogramming level of the system. The arrival of jobs occurs according to a Poisson process, and the arrivals must wait in a waiting area if m jobs are already receiving service. The method presented may be useful in designing the multiprogramming level needed to meet certain objectives on the characteristics of the sojourn time.

I. INTRODUCTION

This paper is concerned with finding the moments and the distribution of sojourn time in a multiprogrammed computer system. We assume that jobs arrive into a computer system according to a Poisson process at a rate of λ . The computer system is divided into two areas, a waiting area and a service area. The service area can hold at most m jobs, where m is the multiprogramming level of the system. The jobs in the service area receive service from a server whose service rate is assumed to be state dependent. The server operates at a rate μ_j [using the Processor Sharing (PS) discipline] whenever there are j jobs in the service area with $\mu_m = \mu$. We assume that each job's service

* Authors are employees of AT&T Bell Laboratories.

Copyright © 1985 AT&T. Photo reproduction for noncommercial use is permitted without payment of royalty provided that each reproduction is done without alteration and that the Journal reference and copyright notice are included on the first page. The title and abstract, but no other portions, of this paper may be copied or distributed royalty free by computer-based and other information-service systems without further permission. Permission to reproduce or republish any other portion of this paper must be obtained from the Editor.

time is exponentially distributed, so that on every service completion, each customer in the service area is equally likely to leave the system. An arrival into the system goes directly into the service area if it is not full. An arriving job goes to the waiting area of unlimited size if there are already m jobs in the service area. The customers are drawn from the waiting area into the service area according to the First-In First-Out (FIFO) discipline. Figure 1 depicts this situation schematically. The model may be useful in determining the multiprogramming level needed to meet a certain response-time characteristic.

The model is an obvious generalization of the M/M/1 First-Come First-Served (FCFS) queue ($m = 1$) and of the M/M/1 PS queue ($\mu_j = \mu$ and $m = \infty$). A little thought should convince the reader that the M/M/m FCFS queue is also a special case of this model with $\mu_i = i\sigma$ for $i = 1, \dots, m$, where σ is the service rate of each server. Avi-Itzhak and Heyman had proposed the use of a state-dependent server to approximate the CPU and I/O subsystem.¹ We depict a multiple CPU and disk subsystem in Fig. 2, which is approximated by a state-dependent server in our model. The service rate μ_i of our model is obtained by solving for the throughput in the closed queueing network of Fig. 2 with a population size of i . Reiser and Lavenberg describe a

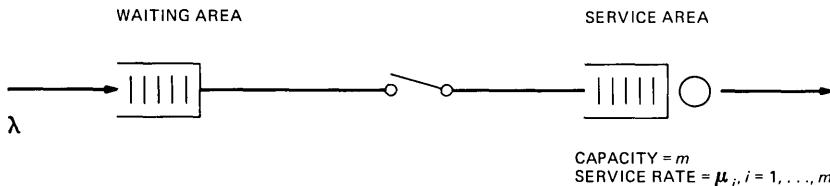
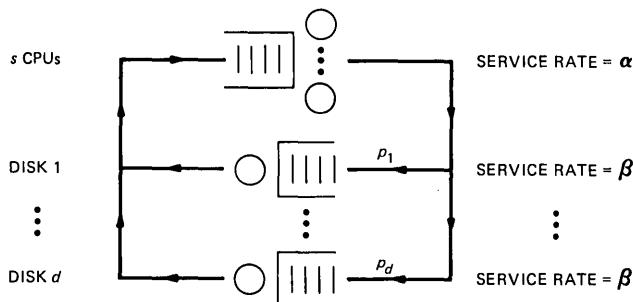


Fig. 1—Model of a multiprogrammed computer.



NOTES: 1. POPULATION SIZE = *i*

2. FCFS DISCIPLINE

3. THROUGHPUT = μ_i , $i = 1, \dots, m$

Fig. 2—Model of a multiple CPU and disk subsystem.

method of solving the throughput of such a closed queueing network.² Fredericks obtains approximations for mean delays in a multiprogrammed computer system and discusses the question of accuracy of the state-dependent server model.³ Konheim and Reiser examine a computer system with one disk and one CPU, subject to a bound on the number of jobs present.⁴ Mitra obtains the waiting time distribution for a computer system fed by jobs from a finite number of terminals.⁵ Salza and Lavenberg investigate hierarchical decomposition methods for approximating response-time distributions in certain closed queueing network models of computer performance.⁶ Coffman, Muntz, and Trotter obtain the sojourn time distribution of the M/M/1 queue with processor sharing discipline.⁷

In Section II of this paper, we discuss some preliminary results. Sections III and IV are concerned with finding the moments of sojourn time. In Section V, we provide a method of obtaining the distribution of sojourn time. In Section VI, we present some numerical examples.

II. DEFINITIONS AND PRELIMINARIES

Let us define the following quantities:

N = Number in system seen by an arrival, excluding itself.

$P_n = P(N = n)$.

I = Number in system seen by a customer entering the service area, including itself.

$q_i = P(I = i)$.

U = Time spent in the waiting area by an arrival.

$F(u) = P(U \leq u)$.

$z_n = E(U^n)$.

T = Time spent in the service area by a tagged customer.

$B_i(s) =$ Laplace-Stieltjes transform of the conditional distribution of T given $I = i$, i.e.,

$$B_i(s) = \int_0^\infty e^{-st} dP(T \leq t | I = i).$$

$$x_{in} = E(T^n | I = i) = (-1)^n \left. \frac{d^n}{ds^n} B_i(s) \right|_{s=0}.$$

V = sojourn time, i.e., $V = U + T$.

By solving the equation of a birth and death process, one obtains the following results directly.

Let

$$w_n = \prod_{j=1}^n \frac{\lambda_j}{\mu_j} \quad \text{for } n = 1, \dots, m.$$

Then

$$P_o = \left[1 + \sum_{n=1}^{m-1} w_n + \frac{w_m}{1 - \rho} \right]^{-1},$$

where

$$\rho = \lambda/\mu_m = \lambda/\mu$$

and

$$P_n = \begin{cases} w_n P_o & \text{if } n = 1, \dots, m \\ \rho^{n-m} P_m & \text{if } n \geq m+1. \end{cases}$$

Let L be the mean number in the system. Then

$$L = P_o \sum_{n=1}^{m-1} nw_n + \frac{P_m \rho}{(1 - \rho)^2} + \frac{m P_m}{1 - \rho}.$$

Let W be the mean time spent in the system. Then, from Little's law,

$$W = L/\lambda.$$

III. CHARACTERIZATION OF U , I , AND V

In this section, we will derive the distribution and moments of U , and the distribution of I , and we will characterize the moments of V . It is clear that

$$U = \begin{cases} 0 & \text{if } N \leq m-1 \\ \text{sum of } (N-m+1) \text{ independent exponentials each} \\ & \text{of rate } \mu \text{ if } N \geq m. \end{cases}$$

Then,

$$F(u) = \sum_{i=0}^{m-1} P_i + \sum_{k=m}^{\infty} P_k \int_0^u \frac{\mu e^{-\mu t} (\mu t)^{k-m}}{(k-m)!} dt \quad \text{for } u \geq 0$$

and

$$z_n = EU^n = \int_0^{\infty} u^n dF(u) = \sum_{k=m}^{\infty} P_k \int_0^{\infty} u^n \frac{\mu e^{-\mu u} (\mu u)^{k-m}}{(k-m)!} du.$$

This can be shown to be equal to

$$\frac{P_m n!}{(1 - \rho)(\mu - \lambda)^n}.$$

This result could also have been obtained as follows:

$$z_n = EU^n = E(U^n | N \geq m)P(N \geq m) + E(U^n | N < m)P(N < m).$$

Given that a customer waits (i.e., $N \geq m$), the waiting room behaves

like an M/M/1 FCFS queue. Thus, the wait time corresponds to the sojourn time of an M/M/1 queue and is exponential with parameter $(\mu - \lambda)$. So,

$$z_n = \frac{n!}{(\mu - \lambda)^n} \cdot \frac{P_m}{1 - \rho}.$$

We will now derive the distribution of I . Consider first the case where $I > m$:

$$\begin{aligned} q_i &= \int_{0+}^{\infty} P(I = i | U = u) dF(u) \\ &= \int_0^{\infty} \sum_{k=m}^{\infty} \frac{e^{-\lambda u} (\lambda u)^{i-m}}{(i-m)!} \cdot P_k \cdot \frac{\mu e^{-\mu u} (\mu u)^{k-m}}{(k-m)!} du. \end{aligned}$$

After some algebra one can show that

$$q_i = P_i \quad \text{for } i > m.$$

This can be explained by the fact that an arrival to and a departure from the waiting area see the same distribution of the number in the waiting area and that the number in the service area is constant, given that $N > m$. For $I < m$, the number in the system including the arrival is one more than the number seen by the arrival. So

$$q_i = P_{i-1} \quad \text{for } i = 1, \dots, m-1.$$

For $I = m$, it is possible for an arrival into the service area to see m customers including itself in one of two possible ways. First, there were $m-1$ customers in the system and an arrival occurred. Second, the arrival saw at least m customers and no one arrived during its wait. So,

$$q_m = P_{m-1} + \int_{0+}^{\infty} P(I = m | U = u) dF(u) = P_{m-1} + P_m$$

by the arguments used above. Thus,

$$q_i = \begin{cases} P_{i-1} & \text{if } i = 1, \dots, m-1 \\ P_{m-1} + P_m & \text{if } i = m \\ P_i & \text{if } i > m. \end{cases}$$

We now characterize the moments of the sojourn time in terms of x_{in} . In the next section, we will show how to calculate x_{in} .

$$EV^n = \sum_{j=0}^n \frac{n!}{j!(n-j)!} E(T^j U^{n-j})$$

and

$$\begin{aligned} E(T^j U^{n-j}) &= \int_{0^-}^{\infty} u^{n-j} E(T^j | U = u) dF(u) \\ &= \sum_{i=1}^{\infty} \int_{0^-}^{\infty} u^{n-j} E(T^j | I = i) P(I = i | U = u) dF(u). \end{aligned}$$

The last step follows from the fact that T is independent of U given I . We now consider three cases:

1. For $j = n$,

$$ET^n = \sum_{i=1}^{\infty} x_{in} q_i.$$

2. For $0 < j < n$, the expression for $E(T^j U^{n-j})$ is

$$\sum_{i=m}^{\infty} \int_{0^-}^{\infty} u^{n-j} x_{ij} \frac{e^{-\lambda u} (\lambda u)^{i-m}}{(i-m)!} dF(u),$$

which, after some algebra, reduces to

$$P_m \sum_{i=m}^{\infty} \frac{(i+n-j-m)!}{(i-m)!} \frac{x_{ij}}{\lambda^{n-j}} \rho^{i+n-j-m}.$$

3. For $j = 0$, but $n \neq 0$, the expression is simply

$$EU^n = z_n = \frac{P_m n!}{(1-\rho)(\mu-\lambda)^n}.$$

So,

$$E(T^j U^{n-j}) = \begin{cases} \sum_{i=1}^{\infty} x_{in} q_i & \text{if } j = n \\ \frac{P_m}{\lambda^{n-j}} \sum_{i=m}^{\infty} \frac{(i+n-j-m)!}{(i-m)!} x_{ij} \rho^{i+n-j-m} & \text{if } 0 < j < n \\ \frac{P_m n!}{(1-\rho)(\mu-\lambda)^n} & \text{if } j = 0, \text{ but } n \neq 0. \end{cases}$$

IV. CALCULATION OF x_{in}

In this section, we show how to calculate x_{in} . The transforms $B_i(s)$ satisfy the following equations:

$$\begin{aligned} B_{i+1}(s) &= \left[\frac{\lambda + \mu_{i+1}}{\lambda + \mu_{i+1} + s} \right] \\ &\cdot \left[\frac{\lambda}{\lambda + \mu_{i+1}} B_{i+2}(s) + \frac{\mu_{i+1}}{\lambda + \mu_{i+1}} (c_{i+1} B_i(s) + (1 - c_{i+1})) \right] \quad (1) \end{aligned}$$

for $i = 1, 2, \dots$ and

$$B_1(s) = \left[\frac{\lambda + \mu_1}{\lambda + \mu_1 + s} \right] \left[\frac{\lambda}{\lambda + \mu_1} B_2(s) + \frac{\mu_1}{\lambda + \mu_1} \right], \quad (2)$$

where

$$c_i = \begin{cases} \frac{i-1}{i} & \text{if } i < m \\ \frac{m-1}{m} & \text{if } i \geq m \end{cases} \quad \text{and} \quad \mu_i = \begin{cases} \mu & \text{if } i < m \\ \mu & \text{if } i \geq m. \end{cases}$$

These equations are obtained by assuming that the system is in state $i+1$ in steady state and conditioning on the time of the next event. An event is defined to be a service completion or an arrival, whichever occurs first. By rearrangement,

$$B_{i+2}(s) - \frac{\lambda + \mu_{i+1} + s}{\lambda} B_{i+1}(s) + \frac{\mu_{i+1} c_{i+1}}{\lambda} B_i(s) = -\frac{\mu_{i+1}}{\lambda} (1 - c_{i+1}).$$

Taking the n th derivative, multiplying by $(-1)^n$, and setting $s = 0$, we get

$$x_{i+2,n} - \frac{\lambda + \mu_{i+1}}{\lambda} x_{i+1,n} + \frac{\mu_{i+1} c_{i+1}}{\lambda} x_{i,n} = -\frac{n}{\lambda} x_{i+1,n-1} \quad (3)$$

for $i \geq 1, n = 1, 2, \dots$, and

$$x_{2,n} - \frac{\lambda + \mu_1}{\lambda} x_{1,n} = -\frac{n}{\lambda} x_{1,n-1}, \quad (4)$$

where

$$x_{i,0} = 1, \quad i \geq 1. \quad (5)$$

Equation (3) is a second-order partial difference equation with variable coefficients. However, we note that for $i \geq m$, the coefficients do not vary with i . So we will first show a method of solving an equivalent system with coefficients that do not vary with i and then present a procedure for the solution of eq. (3). The interested reader is referred to Boole⁸ or Jagerman⁹ for details of the techniques used to solve this difference equation. Consider the equation

$$y_{i+2,n} - \frac{\lambda + \mu}{\lambda} y_{i+1,n} + \frac{\mu c}{\lambda} y_{in} = -\frac{n}{\lambda} y_{i+1,n-1} \quad (6)$$

for positive integer valued i and n , where $c = c_m$. For the homogeneous version of this equation, the solution is

$$y_{in} = A_n \sigma_1^i + B_n \sigma_2^i,$$

where σ_1 and σ_2 are the roots of

$$\sigma^2 - \frac{\lambda + \mu}{\lambda} \sigma + \frac{\mu c}{\lambda} = 0,$$

i.e.,

$$\sigma_1 = \frac{[\lambda + \mu + \sqrt{\lambda^2 + \mu^2 + 2\lambda\mu(1 - 2c)}]}{2\lambda}$$

and

$$\sigma_2 = \frac{[\lambda + \mu - \sqrt{\lambda^2 + \mu^2 + 2\lambda\mu(1 - 2c)}]}{2\lambda}.$$

It can be shown that $\sigma_1 > 1 > \sigma_2$. The constants A_n and B_n will be evaluated later. To find a particular solution to (6), we define the translation operator E and the difference operator Δ such that

$$Eu_i = u_{i+1}$$

and

$$\Delta u_i = u_{i+1} - u_i = (E - 1)u_i.$$

We can then rewrite (6) as

$$(E - \sigma_1)(E - \sigma_2)y_{in} = -\frac{n}{\lambda} y_{i+1,n-1}.$$

The solution is

$$y_{in} = \frac{1}{(\sigma_1 - \sigma_2)} \left(\frac{1}{E - \sigma_1} - \frac{1}{E - \sigma_2} \right) \left(-\frac{n}{\lambda} y_{i+1,n-1} \right).$$

In order to solve this, we must evaluate $(E - \sigma)^{-1} \left(-\frac{n}{\lambda} y_{i+1,n-1} \right)$.

Now,

$$\begin{aligned} (E - \sigma)^{-1} \left(-\frac{n}{\lambda} y_{i+1,n-1} \right) &= (E - \sigma)^{-1} \sigma^{i+1} \sigma^{-(i+1)} \left(-\frac{n}{\lambda} y_{i+1,n-1} \right) \\ &= \sigma^{i+1} (\sigma E - \sigma)^{-1} \sigma^{-(i+1)} \left(-\frac{n}{\lambda} y_{i+1,n-1} \right). \end{aligned}$$

The last step is obtained from the shift formula (see footnote on page 73 of Boole⁸):

$$f(E)(a^i u_i) = a^i f(aE) u_i.$$

The expression now is

$$\begin{aligned}
& \sigma^{i+1} [\sigma(E - 1)]^{-1} \sigma^{-(i+1)} \left(-\frac{n}{\lambda} y_{i+1,n-1} \right) \\
&= \sigma^i \Delta^{-1} \left[\sigma^{-(i+1)} \left(-\frac{n}{\lambda} y_{i+1,n-1} \right) \right] \\
&= -\frac{n}{\lambda} \sum_{j=1}^i \sigma^{i-j} y_{j,n-1}.
\end{aligned}$$

So a particular solution is

$$y_{in} = - \left(\frac{n}{\lambda(\sigma_1 - \sigma_2)} \right) \sum_{j=1}^i (\sigma_1^{i-j} - \sigma_2^{i-j}) y_{j,n-1}.$$

The general solution is

$$y_{in} = - \left(\frac{n}{\lambda(\sigma_1 - \sigma_2)} \right) \sum_{j=1}^i (\sigma_1^{i-j} - \sigma_2^{i-j}) y_{j,n-1} + A_n \sigma_1^i + B_n \sigma_2^i \quad (7)$$

for positive integer valued i and n . It is easy to verify that this satisfies eq. (6).

4.1 Calculation of A_n

We first state that y_{in} has a probabilistic interpretation. We replace the original system by a new one in which the service rate of the server is μ , regardless of the number of customers in the service area. Further, consider a tagged customer whose probability of leaving the system is $1/m$ at each service completion, whenever there are at least two customers in the service area. Then eq. (6) describes the behavior of the conditional moments of the time spent in the service area by the tagged customer in this new system for $i \geq 2$. Clearly, as i tends to infinity, the conditional moments of the original and new systems are identical. Further, as i tends to infinity, the server is going to operate at rate μ for a long time in the original system. If we now tag a customer in the service area, it will leave the system with probability $1/m$ and, therefore, T is exponential with rate μ/m as i tends to infinity in the original system. So

$$\lim_{i \rightarrow \infty} y_{in} = \lim_{i \rightarrow \infty} x_{in} = n! (m/\mu)^n.$$

If we divide eq. (7) by σ_1^i and let i tend to infinity, we have

$$A_n = \frac{n}{\lambda(\sigma_1 - \sigma_2)} \sum_{j=1}^{\infty} \sigma_1^{-j} y_{j,n-1}.$$

Thus,

$$y_{in} = \frac{n}{\lambda(\sigma_1 - \sigma_2)} \left(\sum_{j=i+1}^{\infty} \sigma_1^{i-j} y_{j,n-1} + \sum_{j=1}^i \sigma_2^{i-j} y_{j,n-1} \right) + B_n \sigma_2^i. \quad (8)$$

At this point, it is possible to verify that this result indeed yields the familiar answer for the M/M/1 FCFS queue. To do this, one has to set $m = 1$. This yields $c = 0$ and $\sigma_2 = 0$. After some algebra, one obtains $y_{in} = n!/\mu^n$ for $i \geq 1$ as expected.

4.2 Calculation of B_n

We have so far been able to find y_{in} up to a constant, and a comparison of (3) and (6) should convince the reader that y_{in} obtained from (8) satisfies (3) for $i \geq m$. Thus, after finding x_{in} for $i \geq m$, we can recursively compute $x_{m-1,n}, \dots, x_{1,n}$ by using (3). It is clear that each of these variables is a linear function of the unknown constant B_n . Finally, we will find B_n to satisfy (4). This is done by choosing two trial values of B_n , evaluating two sets of x_{in} , and using linear interpolation to satisfy (4). The formal procedure to do this is as follows:

1. Set $n = 1$. Set $x_{i0} = y_{i0} = 1$ for all $i \geq 1$.
2. Select two trial values of B_n , say B_n^1 and B_n^2 .
3. For each B_n^k ($k = 1, 2$), use (8) to obtain y_{in}^k for $i \geq 1$ and $k = 1, 2$.
2. In this step, the terms $y_{j,n-1}$ on the right-hand side of (8) should not be indexed by k .
4. Set $x_{in}^k = y_{in}^k$ for $i \geq m$ and $k = 1, 2$.
5. Use (3) recursively to compute $x_{m-1,n}^k, \dots, x_{1,n}^k$ for $k = 1, 2$. In this step, the terms $x_{i+1,n-1}$ on the right-hand side of (3) should not be indexed by k .
6. From x_{in}^k ($k = 1, 2$) evaluate the left-hand side of (4). Let L_n^k be the left-hand side corresponding to B_n^k for $k = 1, 2$.
7. Since x_{in} is a linear function of B_n for all i , we have

$$B_n = \gamma B_n^1 + (1 - \gamma) B_n^2,$$

$$x_{in} = \gamma x_{in}^1 + (1 - \gamma) x_{in}^2,$$

and

$$Y_{in} = \gamma Y_{in}^1 + (1 - \gamma) Y_{in}^2 \quad \text{for all positive } i,$$

where

$$\gamma = \left(-\frac{n}{\lambda} x_{1,n-1} - L_n^2 \right) / (L_n^1 - L_n^2).$$

Note that γ does not have to be between zero and one.

8. Set $n = n + 1$ and go to step 2.

V. THE DISTRIBUTION OF SOJOURN TIME

In this section, we provide a direct method of obtaining the distribution of sojourn time. First, we observe that U and T are independent random variables conditional on I . Thus,

$$P(V \leq t) = \sum_{i=1}^{\infty} [P(U \leq t | I = i)^* P(T \leq t | I = i)] q_i, \quad (9)$$

where $*$ is the convolution operator. It is easy to see that

$$P(U \leq t | I = i)$$

$$= \frac{\left[P(I = i | U = 0)P(U = 0) + \int_{0+}^t P(I = i | U = u)dF(u) \right]}{q_i},$$

from which we obtain, after some routine algebra (for $t \geq 0$),

$$P(U \leq t | I = i) = \begin{cases} 1 & \text{if } i \leq m - 1 \\ 1 - P_m e^{-\mu t} / q_m & \text{if } i = m \\ \int_0^t \frac{(\mu u)^{i-m} \mu e^{-\mu u}}{(i-m)!} du & \text{if } i > m. \end{cases} \quad (10)$$

If we let $V(s)$ and $U_i(s)$ be the Laplace-Stieltjes transforms of $P(V \leq t)$ and $P(U \leq t | I = i)$, respectively, then from (9) we have

$$V(s) = \sum_{i=1}^{\infty} B_i(s) U_i(s) q_i. \quad (11)$$

It is possible to use the method of Section IV to solve eq. (1) for $B_i(s)$ directly. In fact, for $i \geq m$, $B_i(s)$ has the form

$$B_i(s) = \frac{\mu/m}{\mu/(m+s)} + B(s) \sigma_2^i(s), \quad (12)$$

where

$$\sigma_2(s) = \frac{[\lambda + \mu + s - \sqrt{(\lambda + \mu + s)^2 - 4\lambda\mu c}]}{2\lambda}.$$

After finding $B_i(s)$ for $i \geq m$, we can recursively compute $B_{m-1}(s), \dots, B_1(s)$ by using (1). It is clear that all of these quantities depend on the unknown function $B(s)$. Finally, one can find $B(s)$ to satisfy (2). This would have to be done by choosing two trial values of $B(s)$, evaluating two sets of $B_i(s)$, and using linear interpolation to satisfy (2). It is possible to show that $V(s)$ simplifies [by using eqs. (10), (11), and (12)] to

$$\sum_{i=1}^{m-1} B_i(s) q_i + B_m(s) \frac{(\mu q_m + s P_{m-1})}{(\mu + s)} \\ + \frac{\mu^2}{\mu + s} \left(\frac{\mu}{s(\mu + sm)} + \frac{B(s) \sigma_2^{m+1}(s)}{s + \mu[1 - \sigma_2(s)]} \right).$$

To find $P(V \leq t)$, one would have to invert this transform numerically.

In the remainder of this section, we will show a method of calculating $P(T \leq t | I = i)$ directly. Let us assume that at all times, events occur in this system according to a Poisson process at a rate of $\lambda + \text{Max}(\mu_1, \mu_2, \dots, \mu_m)$. Let α be the maximum value of the arguments. Whenever the system is in state i , arrivals occur with probability $\lambda/(\lambda + \alpha)$, departures occur with probability $\mu_i/(\lambda + \alpha)$, and with probability $(\alpha - \mu_i)/(\lambda + \alpha)$ the system does not change its state. We assume that $\nu_0 = 0$. Let $\nu_{i,k}$ be the probability that a tagged customer departs on the k th event given that $I = i$. Then, $\nu_{i,k}$ can be recursively computed from

$$\nu_{i,k} = \frac{\lambda}{\lambda + \alpha} \nu_{i+1,k-1} + \frac{c_i \mu_i}{\lambda + \alpha} \nu_{i-1,k-1} + \frac{\alpha - \mu_i}{\lambda + \alpha} \nu_{i,k-1} \quad \text{for } i \geq 1, k \geq 2 \\ \nu_{i,1} = \frac{(1 - c_i)\mu_i}{(\lambda + \alpha)} \quad \text{for } i \geq 1$$

and

$$\nu_{0,k} = 0 \quad \text{for } k \geq 1.$$

Since events are occurring at a Poisson rate of $\lambda + \alpha$, departure of the tagged customer at the k th event means that its time in the service area is a Gamma random variable of order k and parameter $\lambda + \alpha$. Thus,

$$P(T \leq t | I = i) = \sum_{k=1}^{\infty} \nu_{i,k} \int_0^t \frac{(\lambda + \alpha)e^{-(\lambda+\alpha)x} [(\lambda + \alpha)x]^{k-1}}{(k-1)!} dx. \quad (13)$$

It is now possible to use (10), (11), and (13) to show that $V(s)$ is given by

$$\sum_{k=1}^{\infty} \left(\frac{\lambda + \alpha}{\lambda + \alpha + s} \right)^k \\ \cdot \left[\sum_{i=1}^{m-1} \nu_{i,k} q_i + \nu_{m,k} \frac{(\mu q_m + s P_{m-1})}{(\mu + s)} + \sum_{i=m+1}^{\infty} \nu_{i,k} \left(\frac{\mu}{\mu + s} \right)^{i-m+1} \right].$$

VI. NUMERICAL RESULTS

In Table I we present numerical results for four examples, obtained by using the methods described earlier. In all examples, ρ is 0.9 or 0.95 and μ is 1. The first example is for the M/M/1 FCFS queue and the

Table I—Numerical results

Type of Queue	m	$\mu_i (i = 1, \dots, m)$	λ	W	EV	EV^2	EV^3	EV^4
M/M/1	1	1	0.90	10.0	10.0	200.0	6,000	240,000
FCFS			0.95	20.0	20.0	800.0	47,995	3,838,494
M/M/5	5	i/m	0.90	12.625	12.625	278.75	8,756	358,121
FCFS			0.95	22.556	22.556	927.71	56,023	4,483,284
M/M/1	50	1	0.90	10.0	10.0	358.25	26,817	3,252,699
PS			0.95	20.0	20.0	1322.0	157,901	27,094,857
General	10	$1 - [(m - i)/m]^2$	0.90	14.388	14.388	380.17	14,260	686,744
			0.95	24.356	24.356	1073.82	67,654	5,538,847

second for the M/M/5 FCFS queue. We approximate the M/M/1 PS queue in the third example by choosing a high value (50) for m . The fourth example is likely to be typical for a multiprogrammed computer system where μ_i is increasing in i but at a diminishing rate. We get a good correspondence between W (obtained from Little's formula) and EV in all cases. It is possible to verify that the higher moments in the first two examples are very accurate.

The computational procedure described earlier works well for up to fairly high values of m (about 100). However, for very large values of m (say 200), the method is prone to numerical difficulties. We feel that this happens because m determines the number of recursions and this, in turn, determines the extent to which errors are compounded. However, in real-life applications, one may not encounter values of m greater than 40, which makes our method suitable for these applications.

VII. ACKNOWLEDGMENTS

The authors are grateful to Dave Jagerman for his help in solving the difference equation of Section IV.

REFERENCES

1. B. Avi-Itzhak and D. P. Heyman, "Approximate Queueing Models for Multiprogramming Computer Systems," *Oper. Res.*, 21, No. 6 (November 1973), pp. 1212-30.
2. M. Reiser and S. S. Lavenberg, "Mean Value Analysis of Closed Multichain Queueing Networks," IBM Research Report RC7023, 1978.
3. A. A. Fredericks, "Approximations for Customer Viewed Delays in Multiprogrammed, Transaction Oriented Computer Systems," *B.S.T.J.*, 59, No. 9 (November 1980), pp. 1559-74.
4. A. G. Konheim and M. Reiser, "Finite Capacity Queueing Systems With Applications in Computer Modeling," *SIAM J. Computing*, 7, No. 2 (May 1978), pp. 210-29.
5. D. Mitra, *Waiting Time Distributions From Closed Queueing Network Models of*

- Shared Processor Systems*, Performance 81, F. J. Kylstra (Ed.), New York: North Holland, 1981, pp. 113-31.
- 6. S. Salza and S. S. Lavenberg, *Approximating Response Time Distributions in Closed Queueing Network Models of Computer Performance*, Performance 81, F. J. Kylstra (Ed.), New York: North Holland, 1981, pp. 133-45.
 - 7. E. G. Coffman, R. R. Muntz, and H. Trotter, "Waiting Time Distributions for Processor Sharing Systems," *JACM*, 17, No. 1 (January 1970), pp. 123-30.
 - 8. G. Boole, *A Treatise on the Calculus of Finite Differences*, New York: G. E. Stechert, 1931.
 - 9. D. L. Jagerman, unpublished work.

AUTHORS

Kiran M. Rege, B. Tech. (Electrical Engineering), 1977, I.I.T., Bombay; Ph.D. (Electrical Engineering), 1981, University of Hawaii; AT&T Bell Laboratories, 1982—. Mr. Rege spent 1984 on leave of absence, teaching at I.I.T. Bombay in the Department of Electrical Engineering. His technical work at AT&T Bell Laboratories includes modeling and analysis of switching, computer, and communication systems. His research interests include communication theory, queueing theory, and performance analysis of computer and communication systems.

Bhaskar Sengupta, B. Tech. (Electrical Engineering), 1965, I.I.T. Kharagpur, Eng. Sc.D. (Operations Research), 1976, Columbia University; AT&T Bell Laboratories, 1981—. Mr. Sengupta has worked at IBM and the Service Bureau Company and was an Assistant Professor at the State University of New York at Stony Brook. He was also a consultant to Turner Construction Company in New York. At AT&T Bell Laboratories he works on performance problems for communication, computer, and manufacturing systems.

Application of Decomposition Principle in M/G/1 Vacation Model to Two Continuum Cyclic Queueing Models—Especially Token-Ring LANs

By S. W. FUHRMANN* and R. B. COOPER†

(Manuscript received November 15, 1984)

We apply a recent decomposition result of Fuhrmann and Cooper for the M/G/1 queue with server vacations to obtain mean waiting times for the following two cyclic queueing models: The server scans at a constant velocity (1) serving work as it is encountered, or (2) collecting work that it serves at the end of each cycle. Model 1 describes token-ring polling in certain computer-communication networks; Model 2 has been used to describe mail pickup and delivery systems.

I. INTRODUCTION AND SUMMARY

Cyclic queueing models, in which a single server switches back and forth among a (large) number n of queues, have been studied by many authors. These studies were motivated largely by the need to describe the performance of electronic telephone-switching systems. Recent technological developments in computer-communication networks (local area networks, or LANs) have generated renewed interest in these models. In the present paper we consider two *continuum cyclic queueing models*, i.e., models where $n \rightarrow \infty$ while the total arrival rate remains fixed. Model 1 describes the behavior of certain token-ring LANs, while Model 2 has been used to describe mail pickup and delivery systems.

* AT&T Bell Laboratories. † Florida Atlantic University, Boca Raton, Florida.

Copyright © 1985 AT&T. Photo reproduction for noncommercial use is permitted without payment of royalty provided that each reproduction is done without alteration and that the Journal reference and copyright notice are included on the first page. The title and abstract, but no other portions, of this paper may be copied or distributed royalty free by computer-based and other information-service systems without further permission. Permission to reproduce or republish any other portion of this paper must be obtained from the Editor.

Exact analytic models for finite n tend to be very complicated (see, for example, Refs. 1 through 8). One of the earliest techniques for the analysis of these complicated multiqueue models was to define the ordinary single-server *vacation model*, in which the server periodically leaves the queue and takes a "vacation"; this vacation model is then "connected" to the model of n queues served in cyclic order by interpreting the vacation as the time interval from when the server leaves a particular queue until its return to that queue after cycling through the other $n - 1$ queues.²

The basis for the analysis in the present paper is to relate the server vacations to the cyclic queueing model in an entirely different manner, and then to apply a new stochastic decomposition result of Fuhrmann and Cooper⁹ for the M/G/1 vacation model. (As noted, in the present paper we are interested in models where n is infinite. Another paper¹⁰ uses a related method to analyze certain cyclic queueing models where n is finite.)

In both of our continuum models, the server scans (or polls) at a constant velocity along a closed path. Customers arrive according to a Poisson process (with rate λ) in time, and are uniformly and independently distributed in space along the scanning path. Service times have distribution function $H(\cdot)$, with mean τ and variance σ^2 , and are independent of the arrival process and each other. In Model 1, the server stops scanning and serves customers as they are encountered along the scanning path. In Model 2, the server *collects* customers (there is no time expended in collecting a customer) as they are encountered along the scanning path; when the server reaches a unique point on the path called the *origin*, the server stops and serves all the customers it has collected since last leaving the origin. Model 2 is closely related to a model studied by Nahmias and Rothkopf,¹¹ which we shall refer to as Model 3, in which customers are served at the origin and are then randomly (uniformly) redistributed (delivered) over the scanning path on the server's next cycle. This is in contrast to Models 1 and 2, where each customer departs from the system as soon as his service is completed.

Model 1 provides a good description of a large, symmetric, token-ring LAN. In such a network, a number n of terminals (devices, work stations) are interconnected in either a physical or logical ring structure. The terminals' access to the transmission medium is controlled by a "token" (a signal) that circulates around the ring. A terminal gains access to the medium by seizing the circulating token as it goes by. It retains the token while it is transmitting, thereby preventing other terminals from simultaneously accessing the medium; it then releases the token to circulate around the ring, enabling another terminal to gain access to the transmission medium. (For a general

description of LANs and token-passing protocols, see, for example, Refs. 12 through 14.)

One identifies the scan time c as the time required for the server to poll all the terminals once (equivalently, the time required for the token to cycle once around the LAN ring when no terminals are waiting to transmit). If the number n of terminals is large, and the terminals submit statistically identical loads (i.e., each terminal is characterized by the same arrival rate and distribution of service times), then there is a good correspondence between Model 1 and the LAN.

Model 3 has been studied by Nahmias and Rothkopf,¹¹ who used it to describe a delivery system in which a clerk (the server) traverses (scans) at a constant velocity a route along which letters (customers) are generated (arrive) randomly in space and time. As the clerk travels along the route, he picks up the letters that have been generated since his last traversal, and he delivers the letters (to locations distributed uniformly along the route) that were previously picked up and sorted. When the clerk reaches the end of the route he sorts (serves) the letters he has just picked up; then he again traverses the route, delivering the letters that have just been sorted, and picking up the new letters that have been generated since his last traversal of the route. This process is repeated indefinitely.

For each model, the equilibrium *cycle time* T_j , defined as the time (during equilibrium) between successive visits in Model j by the server to any given point along the scanning path, has the same mean value $\bar{T} = E(T_j)$, given by

$$\bar{T} = \frac{c}{1 - \rho} \quad (\rho < 1), \quad (1)$$

where c is the (constant) length of time the server spends scanning during each cycle (which is the time to complete a scan cycle when there is no work to be done), and $\rho (= \lambda\tau)$ is the server utilization. [Equation (1) is easily derived by the following argument, given by Kuehn,⁷ for a very general model of n queues served in cyclic order by a single server: The mean cycle time \bar{T} is the sum of the (constant) time c spent scanning and the mean time s spent serving per cycle; that is, $\bar{T} = c + s$. Clearly, $s = \rho\bar{T}$, and (1) follows. Note that \bar{T} does not depend on the form of the service-time distribution function, but only on its mean value; also, the parameter n does not appear explicitly.]

Our main results are these: Let W_j ($j = 1, 2$) be the equilibrium *waiting time* (time from request for service until start of service) in Model j , and let W_0 be the equilibrium waiting time in the corresponding M/G/1 queue (Model 0). Then,

$$E(W_1) = \frac{1}{2} \bar{T} + E(W_0), \quad (2)$$

and

$$E(W_2) = \bar{T} + E(W_0), \quad (3)$$

where \bar{T} is given by (1), and $E(W_0)$ is given by the celebrated Pollaczek-Khintchine formula [see, for example, Ref. 15, p. 217, eq. (8.39)]:

$$E(W_0) = \frac{\rho\tau}{2(1-\rho)} \left(1 + \frac{\sigma^2}{\tau^2} \right). \quad (4)$$

The simplicity of (2) and (3) and their similarity are quite remarkable.

We also define S_j ($j = 0, 1, 2$) to be the equilibrium *sojourn time* (waiting time plus service time) in Model j . Since

$$E(S_j) = E(W_j) + \tau \quad (j = 0, 1, 2),$$

eqs. (2) and (3) are equivalent to the following two equations:

$$E(S_1) = \frac{1}{2} \bar{T} + E(S_0) \quad (5)$$

and

$$E(S_2) = \bar{T} + E(S_0). \quad (6)$$

For Model 3, we define the equilibrium *delivery time* D_3 as the elapsed time between the generation of a letter and its delivery to its destination. We will show that $E(D_3)$ is given by the following formula, again remarkable in its simplicity:

$$E(D_3) = \tau + \frac{3}{2} \bar{T} + \frac{1+2\rho}{1+\rho} E(W_0). \quad (7)$$

The special case of (7) when $\sigma^2 = 0$ (i.e., when the time required to sort a letter is constant) was found (in a different form, by a more complicated argument) by Nahmias and Rothkopf.¹¹ The general result (7) was found also by Shanthikumar,¹⁶ using level-crossing analysis.

The well-known textbook by Tanenbaum¹² discusses a model of a token-ring LAN with an arbitrary number n of terminals that, for n infinite, coincides with our Model 1. Tanenbaum gives eq. (1) and then states (p. 310) that the mean waiting time is “about half” the mean cycle time. It is interesting to note that, for the case of n infinite, eq. (2) shows that Tanenbaum’s approximation (i.e., $E(W_1) = \bar{T}/2$) underestimates the correct value by exactly $E(W_0)$, an amount that can be considerable, being essentially proportional to σ^2 and inversely proportional to $1 - \rho$. [The reason that $\bar{T}/2$ underesti-

mates the correct value is a manifestation of the phenomenon of length biasing, i.e., the cycle to which an arbitrary customer (the *test customer*) arrives is stochastically longer than an arbitrary cycle. In particular, if T_1^* is the length of the cycle during which the test customer arrives, then $E(T_1^*) = E(T_1) + V(T_1)/E(T_1)$, where $V(T_1)$ is the variance of the cycle times in Model 1 (see, for example, Ref. 15, pp. 200–6). Since $E(W_1) = E(T_1^*)/2$, it follows from this observation and eq. (2) that $V(T_1) = 2\bar{T}E(W_0)$. A practical implication of this observation is that the mean waiting time $E(W_1)$ can be estimated using measurements of cycle times only.]

Several authors (see Refs. 4, 17, 18, and 19), using arguments more complicated than ours, have obtained results for related models with different queue disciplines (e.g., exhaustive service or gated service) and a finite number n of terminals; our result (2) can be obtained from their results when $n \rightarrow \infty$. Coffman and Gilbert²⁰ have analyzed Model 1 for the case of constant service times and, for this special case, derived a number of explicit distributional results, such as the distribution of waiting times.

In Section II we state the M/G/1 decomposition result alluded to earlier. In Section III we apply this decomposition result to obtain eqs. (2), (3), (5), and (6). For completeness, in Section IV we quickly derive eq. (7) by directly comparing the mean delays in Models 2 and 3.

II. A STOCHASTIC DECOMPOSITION RESULT

At all times the server is either scanning or is serving customers. The basis for the analysis of this paper is to interpret the time intervals when the server is scanning to be vacations, and then to invoke Proposition 3 of Fuhrmann and Cooper.⁹ We define

- $\psi_j(\cdot)$ = the p.g.f. (probability generating function) of the equilibrium distribution of the number of the customers present in Model j ($j = 1, 2$) at an arbitrary point in time;
- $\chi_j(\cdot)$ = the p.g.f. of the equilibrium distribution of the number of customers present in Model j ($j = 1, 2$) at an arbitrary point in time, given that the server is scanning (on vacation); and
- $\pi(\cdot)$ = the p.g.f. of the equilibrium distribution of the number of customers present in the corresponding M/G/1 queue at an arbitrary point in time (or, equivalently, just after a service completion epoch).

Thus, $\pi(\cdot)$ is given by a well-known formula [see, for example, eq. (8.12), p. 210, Ref. 15]. It follows directly from Proposition 3 of Fuhrmann and Cooper⁹ that

$$\psi_j(z) = \chi_j(z)\pi(z) \quad (j = 1, 2). \quad (8)$$

In terms of mean values,

$$\psi'_j(1) = \chi'_j(1) + \pi'(1) \quad (j = 1, 2). \quad (9)$$

In Section III we show that it is a simple matter to find $\chi'_j(1)$ for $j = 1, 2$. Since (by Little's theorem) $\pi'(1) = \lambda E(S_0)$ and $\psi'_j(1) = \lambda E(S_j)$, eq. (9) yields eqs. (5) and (6) or, equivalently, (2) and (3).

III. MEAN WAITING TIMES: MODELS 1 AND 2

In this section we derive formulas (2) and (3) for the mean waiting times for Models 1 and 2. To do this, first note that (for either model) during each customer's service time, k new customers arrive to the system with probability

$$p_k = \int_0^\infty \frac{(\lambda t)^k}{k!} e^{-\lambda t} dH(t) \quad (k = 0, 1, 2, \dots). \quad (10)$$

We now define two auxiliary models, Auxiliary Model 1 and Auxiliary Model 2. Auxiliary Model 1 is defined in exactly the same way as Model 1 except for the following aspect: Now, whenever the server encounters a customer, the customer is served in zero time and departs from the system; coincidental with his departure, however, a batch of k new customers joins the system (distributed along the scanning path in a uniform and independent manner) with probability p_k , given by (10). Thus, while the lengths of all service times have been collapsed to zero, the number of customers in the system just after a service completion epoch is stochastically the same for both Model 1 and Auxiliary Model 1. (This is true in a distributional sense. Or, if we go to the trouble to define Model 1 and Auxiliary Model 1 on the same sample space, we can make this statement true on every sample path.) This leads to the following conclusion: If we define A_1 to be the mean number of customers present in Auxiliary Model 1, then

$$\chi'_1(1) = A_1. \quad (11)$$

To calculate A_1 , let λ_1^* and S_1^* be the arrival rate and sojourn time in Auxiliary Model 1. Then, by Little's theorem,

$$A_1 = \lambda_1^* E(S_1^*), \quad (12)$$

where, clearly,

$$E(S_1^*) = \frac{c}{2}. \quad (13)$$

To calculate λ_1^* , let N be the total number of customers (including himself) generated by a *Poisson* arrival in Auxiliary Model 1; then $\lambda_1^* = \lambda E(N)$. Now observe that the average number of new customers

generated when a customer is served is $\lambda\tau = \rho$; and each of these customers will generate, on average, $E(N)$ additional customers. Hence, $E(N) = 1 + \rho E(N)$; that is, $E(N) = (1 - \rho)^{-1}$, and therefore

$$\lambda_i^* = \frac{\lambda}{1 - \rho}. \quad (14)$$

[Note that $E(N)$ is precisely the mean number of customers served during an M/G/1 busy period. Observe also that (14) follows immediately from the requirement that the mean number of arrivals per cycle be the same in Auxiliary Model 1 as in the original Model 1: $\lambda_i^*c = \lambda\bar{T}$.] Equations (11) through (14) yield $\chi'_i(1) = \lambda c / 2(1 - \rho)$; in light of (1), we have

$$\chi'_i(1) = \frac{\lambda\bar{T}}{2}. \quad (15)$$

This completes the calculation of (9) for Model 1, from which the main result (2) follows.

We now define Auxiliary Model 2 in a completely analogous manner, that is, in exactly the same way as Model 2, except that now when a customer is served (at the origin), he is served in zero time and is instantaneously replaced by a batch of k customers with probability p_k , given by (10). We define A_2 to be the mean number of customers present in Auxiliary Model 2. By the same argument used earlier,

$$\chi'_2(1) = A_2; \quad (16)$$

and the same argument that was used to derive eq. (15) for Model 1 applies. Hence, combining the equations that are analogous to (12) and (14), we have

$$A_2 = \frac{\lambda}{1 - \rho} E(S_2^*). \quad (17)$$

But, in contrast with (13), the mean sojourn time of a customer in Auxiliary Model 2 is exactly the cycle time c ,

$$E(S_2^*) = c. \quad (18)$$

Therefore, the analogue of (15) is

$$\chi'_2(1) = \lambda\bar{T}, \quad (19)$$

and the main result (3) follows.

IV. MEAN DELIVERY TIME: MODEL 3

For completeness, we now derive eq. (7). This is accomplished by directly comparing the mean delays in Models 2 and 3. Recall that in

these models, all customers are served at the origin. For either model, consider an arbitrary customer (the test customer) and define

- y = the mean number of customers in the test customer's batch;
- y_b = the mean number of customers in the test customer's batch that are served before the test customer; and
- y_a = the mean number of customers in the test customer's batch that are served after the test customer.

Then

$$y = y_b + y_a + 1 \quad (20)$$

and, by symmetry,

$$y_b = y_a. \quad (21)$$

Now let L_2 denote the number of customers present in Model 2, excluding the test customer, when the test customer enters service. Then

$$E(L_2) = \frac{\lambda c}{2} + y_b\rho + y_a. \quad (22)$$

The term $\lambda c/2$ equals the mean number of customers who arrived during the last scan, but behind the server. (These customers will be collected on the server's next cycle.) The term $y_b\rho (= \lambda y_b\tau)$ equals the mean number of customers who arrived during the service times of the customers (in the test customer's batch) who were served before the test customer. Finally, the term y_a equals the mean number of customers that have not yet been served.

Now observe, on the other hand, that L_2 has the same distribution as the number of customers present in Model 2 at an arbitrary point in time, excluding the customer being served (if any). [This is true because departures see the same distribution of customers that arrivals see (see Ref. 15, p. 187) and the arrivals see time averages (see Ref. 21).] Hence, by Little's theorem,

$$E(L_2) = \lambda E(W_2). \quad (23)$$

Combining (21), (22), and (23) yields

$$\lambda E(W_2) = \frac{\lambda c}{2} + y_a(1 + \rho). \quad (24)$$

Equations (3) and (24) determine y_a . Now observe that, clearly,

$$E(D_3) = E(W_2) + \tau + y_a\tau + \frac{c}{2}. \quad (25)$$

Equations (3), (24), and (25) now yield eq. (7) after some straightforward algebra.

REFERENCES

1. R. B. Cooper and G. Murray, "Queues Served in Cyclic Order," *B.S.T.J.*, 48, No. 3 (March 1969), pp. 675-89.
2. R. B. Cooper, "Queues Served in Cyclic Order: Waiting Times," *B.S.T.J.*, 49, No. 3 (March 1970), pp. 399-413.
3. M. Eisenberg, "Queues With Periodic Service and Changeover Times," *Oper. Res.*, 20, No. 2 (March-April 1972), pp. 440-51.
4. O. Hashida, "Analysis of Multiqueue," *Review of the Electrical Communication Laboratories*, N. T. T. Public Corp., 20, Nos. 3-4 (March-April 1972), pp. 189-99.
5. A. G. Konheim and B. Meister, "Waiting Lines and Times in a System With Polling," *J. Ass. Comput. Mach.*, 21, No. 3 (July 1974), pp. 470-90.
6. S. Halfin, "An Approximate Method for Calculating Delays for a Family of Cyclic Type Queues," *B.S.T.J.*, 54, No. 10 (December 1975), pp. 1733-54.
7. P. J. Kuehn, "Multiqueue Systems With Nonexhaustive Cyclic Service," *B.S.T.J.*, 58, No. 3 (March 1979), pp. 671-98.
8. G. B. Swartz, "Polling in a Loop System," *J. Ass. Comput. Mach.*, 27, No. 1 (January 1980), pp. 42-59.
9. S. W. Fuhrmann and R. B. Cooper, "Stochastic Decompositions in the M/G/1 Queue With Generalized Vacations," *Oper. Res.*, 33, 1985.
10. S. W. Fuhrmann, private communication.
11. S. Nahmias and M. H. Rothkopf, "Stochastic Models of Internal Mail Delivery Systems," *Manage. Sci.*, 30, No. 9 (September 1984), pp. 1113-20.
12. A. S. Tanenbaum, *Computer Networks*, Englewood Cliffs: Prentice-Hall, 1981.
13. W. Bux, "Performance Issues in Local-Area Networks," Research Report RZ 1268 (45715), IBM Zurich Research Laboratory, November 1983.
14. H. Takagi and L. Kleinrock, "Analysis of Polling Systems," Technical Report TR87-0002, IBM Japan Science Institute, Chiyoda-ku, Tokyo, 1985.
15. R. B. Cooper, *Introduction to Queueing Theory*, second edition, New York: North-Holland (Elsevier), 1981.
16. J. G. Shanthikumar, private communication.
17. W. Bux, "Local-Area Subnetworks: A Performance Comparison," *IEEE Trans. Commun.*, COM-29, No. 10 (October 1981), pp. 1465-73.
18. L. F. M. de Moraes and I. Rubin, "Analysis and Comparison of Message Queueing Delays in Token-Rings and Token-Buses Local Area Networks," *Proc. IEEE Int. Conf. Commun.*, Amsterdam, May 14-17, 1984, pp. 130-5.
19. H. Takagi, "Mean Message Waiting Time in N Symmetric Queues With Nonexhaustive Cyclic Service Discipline," IBM Japan Science Institute, Chiyoda-ku, Tokyo, 1984.
20. E. G. Coffman and E. N. Gilbert, private communication.
21. R. W. Wolff, "Poisson Arrivals See Time Averages," *Oper. Res.*, 30, No. 2 (March-April 1982), pp. 223-31.

AUTHORS

Robert B. Cooper, B.S., 1961, Stevens Institute of Technology; M.S. (Systems Engineering and Operations Research), 1962, and Ph.D. (Electrical Engineering), 1968, University of Pennsylvania; AT&T Bell Laboratories, 1961-1969; Georgia Institute of Technology, 1969-1976; University of Michigan, 1975; New Mexico Institute of Mining and Technology, 1976-1978; Florida Atlantic University, 1978-. Mr. Cooper's primary interest is queueing theory and its application in telecommunications and computer science.

Steve W. Fuhrmann, B.A. (Mathematics), 1970, University of Montana; Ph.D. (Statistics), 1975, Purdue University; Rutgers University, 1975-1977; AT&T Bell Laboratories, 1977-. Mr. Fuhrmann's interests and work focus on stochastic processes and stochastic models for the performance evaluation of computer-communication systems.

PLACE 2.0—An Interactive Program for PLL Analysis and Design

By J. H. SAUNDERS*

(Manuscript received January 14, 1985)

Phase Locked Loop Analysis and Circuit Emulation (PLACE) is an interactive program to assist in the design of Phase Locked Loop (PLL) systems. Written in C language, PLACE computes the loop filter components (up to active third-order filters); performs a stability analysis (finding the phase margin and damping ratio); and calculates the lockup time, hold range, and capture range. PLACE computes the PLL output jitter response due to incoming reference signal jitter, output jitter due to reference leakage through the phase detector, and output jitter response due to phase noise of the PLL components. The open and closed loop gain and phase, as well as jitter response, is plotted. Additional features found in PLACE 2.0 are frequency and magnitude of jitter peaking; a sensitivity analysis, which computes changes in loop performance as a function of component variation; and an interactive routine to help the designer optimize PLL performance. Currently residing on Digital Equipment Corporation's VAX-11/780, AT&T 3B20, and IBM System/370 processors, PLACE is presently available at most AT&T Bell Laboratories locations. This paper illustrates the capabilities of PLACE, shows several examples, and discusses the required calculations.

I. INTRODUCTION

Phase Locked Loops (PLLs) are used extensively in communication systems. Yet there has been no generally available computer-aided design program specifically targeted to assist the PLL designer in evaluating the performance of his system *before* building it in the

* AT&T Bell Laboratories.

Copyright © 1985 AT&T. Photo reproduction for noncommercial use is permitted without payment of royalty provided that each reproduction is done without alteration and that the Journal reference and copyright notice are included on the first page. The title and abstract, but no other portions, of this paper may be copied or distributed royalty free by computer-based and other information-service systems without further permission. Permission to reproduce or republish any other portion of this paper must be obtained from the Editor.

laboratory. Phase Locked Loop Analysis and Circuit Emulation (PLACE) is an interactive program that allows the designer to optimize his PLL, trading off one parameter (e.g., lockup time) for another (e.g., jitter response). PLACE performs the following functions:

1. Determines if the PLL is stable by computing the phase margin, damping ratio, and undamped natural frequency. PLACE accounts for any parasitic Voltage Controlled Oscillator (VCO) or operational amplifier poles, and phase shift due to the feedback counter.
2. Finds the appropriate loop filter components for a given damping ratio or 3-dB frequency. PLACE notifies the user if desired loop parameters result in unrealistic component values.
3. Computes the PLL system bandwidth, noise bandwidth, and loop filter bandwidth.
4. Approximates the hold range, capture range, and lockup time. (The hold range is normalized for the active loop filter case.)
5. Determines the output jitter response due to the internal phase noise of the PLL components and the output jitter response due to jitter on the incoming reference signal.
6. Determines the sensitivity of PLL performance to component variation.
7. Plots open and closed loop gains, and phase noise response.

Section II of this paper discusses PLACE input/output, Section III shows several examples, and Section IV explains the calculations that PLACE performs.

II. PLACE 2.0 INPUT/OUTPUT

2.1 PLACE 2.0 user input

PLACE consists of a nongraphics module followed by a graphics module. The two modules are independent, and thus a user can invoke PLACE on a nongraphics terminal. PLACE recognizes the PLL shown in Fig. 1, where we have defined the following constants:^{*}

K_p = phase detector gain constant (V/rad)

K_V = VCO gain constant (Hz/V)

N_{FB} = feedback counter divisor

N_{FF} = feed-forward counter divisor.

The program will ask the user for the values of these constants. Typical values for K_p are 1.4 V/rad for an exclusive-or gate, 0.4 V/rad for the RCA 4046 phase comparator II, 0.11 V/rad for the Motorola 4044 phase detector, and 0.16 V/rad for the Motorola 12040.

For best results, the VCO gain constant should be measured because the PLL stability is highly dependent on K_V . The value of K_V will

* Variables used in this paper are defined in Appendix A.

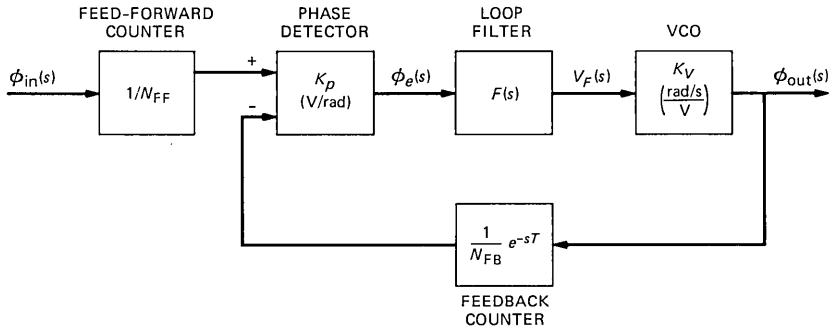


Fig. 1—PLL model depicting the required input parameters for PLACE.

most likely vary over the VCO's frequency range; the sensitivity analysis will compute the effect of this variation. If the results are unacceptable, the user will need to build a linearizer in order to maintain a constant K_V . (See Ref. 1 for an excellent example, or Ref. 2.) If a passive attenuator is used ahead of the VCO, the entered value of K_V must be reduced by the attenuation factor. For frequency synthesizer applications, the feedback divisor N_{FB} is often a variable, and running PLACE twice with the maximum and minimum values will show the change in loop performance.

Next, PLACE asks the user if it is desired to account for any parasitic VCO pole; if a Voltage Controlled Crystal Oscillator (VCXO) is used, it is strongly recommended that this pole be entered since it is often the dominant pole of the PLL. (See Section 4.1.4 on how to measure the VCO pole.) For the active loop filter case, if it is desired to account for the op amp pole, the designer may lump this pole into the VCO pole and enter the value at this time.

Next, PLACE asks for the reference frequency, i.e., the frequency entering the phase detector. The reference frequency is the frequency at which the phase comparison is performed. This information is required for the jitter analysis and implicitly defines the output frequency of the PLL.

PLACE will then ask the user for the type of loop filter desired. Five types of loop filters are recognized (see Fig. 2):

1. No loop filter, where

$$F(s) = 1.$$

2. Resistor Capacitor (RC) loop filter, where

$$F(s) = \frac{1}{1 + \tau s},$$

where $\tau = RC$.

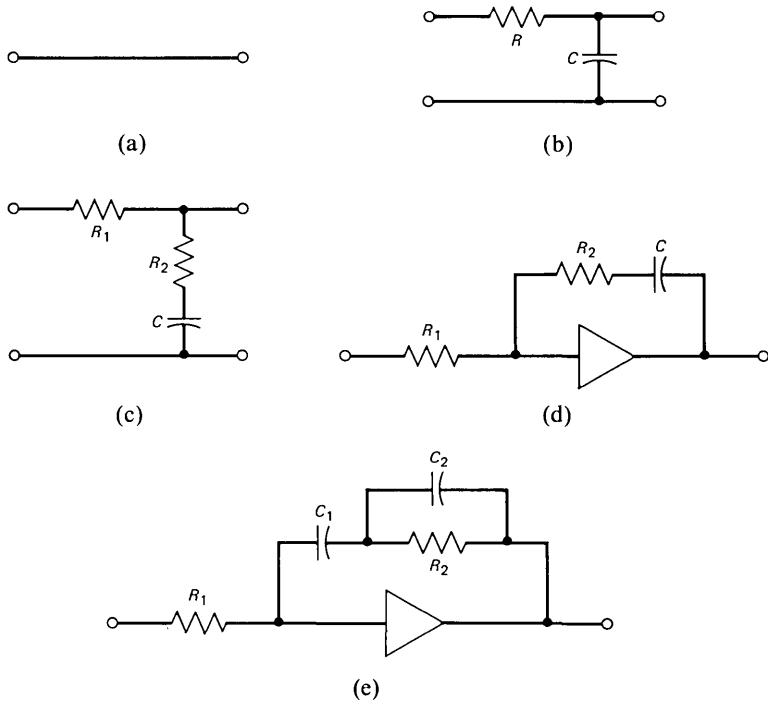


Fig. 2—Loop filter topologies: (a) No loop filter, (b) RC loop filter, (c) lag-lead loop filter, (d) second-order active filter, and (e) third-order active filter.

3. Lag-lead loop filter, where

$$F(s) = \frac{1 + \tau_2 s}{1 + \tau_1 s},$$

where $\tau_1 = (R_1 + R_2)C$, $\tau_2 = R_2C$.

4. Second-order active filter, where

$$F(s) = \frac{1 + \tau_2 s}{\tau_1 s},$$

where $\tau_1 = R_1C$, $\tau_2 = R_2C$.

5. Third-order active filter, where

$$F(s) = \frac{1 + \tau_2 s}{\tau_1 s(1 + \tau_3 s)},$$

where $\tau_1 = R_1C_1$, $\tau_2 = R_2(C_1 + C_2)$, $\tau_3 = R_2C_2$.

We now list several guidelines for selecting a loop filter. The choice between an active or passive loop filter must be made first. If the

designer chooses a passive loop filter, the lag lead is preferred over the RC loop filter for the following reasons:

1. For the lag-lead loop filter, the designer can specify the loop natural frequency ω_n and damping ζ independently of each other, and thus it is possible to have a narrowband loop with a substantial damping. However, for the RC loop filter, a narrowband PLL (low ω_n) requires a high value of τ (loop filter time constant), but a high τ results in low damping and possible instability (see Section IV).

2. The VCO parasitic pole is less likely to render a lag-lead PLL unstable compared with the RC loop filter PLL. (This is because the open loop phase for the RC loop filter PLL approaches -180 degrees at high frequencies, whereas the phase approaches -90 degrees for the lag-lead case.)

3. The lag-lead PLL exhibits improved transient behavior. For a given PLL damping value, the phase error transient of the lag-lead PLL reaches its steady-state value much sooner than that of the RC loop filter.

4. All of these improvements are obtained for the cost of one resistor.

An active loop filter is required when a higher loop gain is required (to increase the hold range, for example). The advantages of the third-order active filter over the second-order active filter are improved response to phase-step changes, zero steady-state phase error due to frequency ramp inputs, and better reduction of VCO noise.

After selecting a loop filter type, the user has two options: the desired damping and natural frequency may be specified, or the loop filter time constants may be specified. The first option is usually used when designing a new PLL, while the second option is normally used when analyzing an existing PLL.

2.2 PLACE 2.0 output

The user can request a stability analysis, a loop filter analysis, a tracking analysis, a jitter analysis, a sensitivity analysis, and an interactive optimization routine.

The *stability* analysis yields (1) second-order undamped natural frequency ω_n in hertz, (2) phase margin in degrees, (3) phase margin degradation due to the VCO (and/or op amp) pole, (4) phase margin degradation due to any phase delay of the feedback counter, (5) damping, and (6) PLL system bandwidth in hertz.

The *loop filter* analysis yields (1) the loop filter time constants in seconds, (2) values for the loop filter resistors and capacitors, and (3) the loop filter 3-dB frequency.

The *tracking* analysis yields (1) hold range in hertz (for active loop filters, the hold range is normalized, i.e., assuming $F(0) = 1$); (2)

approximate capture range in hertz; (3) approximate pull-in range in hertz; and (4) approximate lockup time in seconds.

The *jitter* analysis yields (1) output jitter due to incoming reference signal jitter: jitter bandwidth in hertz, frequency of jitter peaking in hertz, and noise bandwidth in hertz; (2) output jitter due to reference leakage through the phase detector: frequency of first sideband in hertz, magnitude of first sideband relative to the carrier in decibels, peak phase jitter in degrees, peak frequency deviation in hertz, and loop filter's attenuation of reference frequency in decibels; and (3) output jitter due to VCO phase noise: VCO phase noise reduction 3-dB frequency.

The *sensitivity* analysis computes upper and lower bounds on the hold range, capture range, pull-in range, and lockup time. The user input is the tolerance of the gain constants and loop filter components.

The *optimization* routine asks if the user wants to design for any one of the following: (1) larger hold range, (2) larger capture and pull-in range, (3) faster lockup time, or (4) less output jitter.

PLACE then automatically changes the PLL parameters to achieve the desired goal; then, the user can rerun the loop filter analysis to observe the new component values and use the tracking analysis to observe the new lockup time, etc.

The graphics portion of PLACE plots (on a TEK 4014 or similar device) the open and closed loop gains and phase, and the PLL response to VCO phase noise. For the graphics portion, the S package³ must be installed on the system.

III. EXAMPLE

This example demonstrates the lag-lead loop filter case; it also demonstrates that the parasitic VCO pole may add sufficient phase shift to cause possible instability.

The user input to PLACE is as follows: The PLL for this example phase locks a 3.088-MHz crystal oscillator to an incoming 1.544 MHz signal. The VCXO has a measured gain constant of 800 Hz/V around the center frequency of 3.088 MHz; it also has a parasitic pole at 10 Hz. The phase detector is a Complementary Metal-Oxide Semiconductor (CMOS) exclusive-or gate measured at 1.4 V/rad; this was derived by measuring the logic high/low levels: $(4.6 - 0.2)/\pi = 1.4$. The phase comparison is done at 4 kHz; thus, the values for the frequency dividers are $N_{FF} = 386 = 1.544 \text{ MHz}/4 \text{ kHz}$ and $N_{FB} = 772 = 3.088 \text{ MHz}/4 \text{ kHz}$. The loop filter resistors and capacitors were measured accurately, yielding time constants of $\tau_1 = 57.4513 \text{ ms}$ and $\tau_2 = 4.00336 \text{ ms}$.

The results from the stability, loop filter, tracking, and jitter analysis

Table I—Comparing PLACE output with measured results

Parameter	Calculated	Measured
Hold range	+/-568 Hz	+496, -588 Hz
Pull-in range	+/-568 Hz	+493, -585 Hz
Lockup time	324 ms	400 ms
Frequency of jitter peaking	1.6 Hz	1.5 Hz
Magnitude of jitter peaking	1.3 dB	3 dB
Jitter bandwidth	3 Hz	4 Hz
Frequency of first sideband	8 kHz	8 kHz
Magnitude of first sideband	-52 dBc	-50 dBc
Peak base jitter	0.279 degrees	0.343 degrees
Peak frequency deviation	20 Hz	25 Hz
Reference frequency attenuation	23 dB	23 dB

appear in Appendix B. Table I summarizes the calculated results and also shows the measured values. The stability analysis indicates a damping of 0.7 and natural frequency of 2 Hz. The phase margin is 60 degrees: the VCO pole reduced the phase margin by 7.4 degrees, while the feedback counter caused negligible phase margin degradation of 0.02 degree. The PLL bandwidth is 1.2 Hz; this low PLL bandwidth is due to the very low K_V of 800 Hz/V. The previous results can also be found from the PLACE graphic display of open loop gain, shown in Fig. 3. Unity gain occurs at 1.2 Hz, where the phase is -120 degrees.

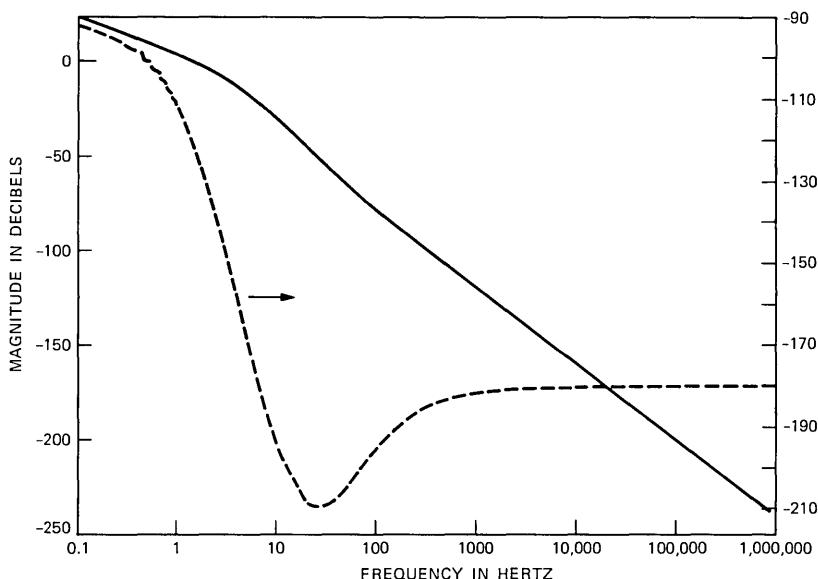


Fig. 3—PLACE graphic display of open loop gain for the lag-lead loop filter example. The magnitude is indicated by the solid line, and the phase is indicated by the dotted line (right scale).

Note that the parasitic VCO pole has brought the total phase shift to -180 degrees (instead of 0 degrees) for high frequencies. The tracking analysis indicates a hold range of +/-568 Hz. The measured value is +496 Hz, -588 Hz. The hold range is asymmetrical due to the larger than 50-percent duty cycle phase detector output during the lock condition. The pull-in range is +493, -582 Hz; it is common for low gain loops to have pull-in ranges close to the hold range.

The calculated lockup time is 324 ms, while the measured time is 400 ms; this may be found from Fig. 4, a display of tuning voltage during the lock-in process.

The jitter analysis indicates 1.3 dB of jitter peaking at 1.6 Hz, and a jitter bandwidth of 3 Hz. The closed loop gain Bode plot is shown in Fig. 5. The peaking is difficult to observe given the scale of the plot. The measured values may be found from Fig. 6, a spectrum analyzer display of closed loop gain versus jitter frequency from an HP 8557A. There is 3 dB of jitter peaking at 1.5 Hz (picture only displays up to 50 Hz). (The measured jitter peaking is higher than calculated due to additional poles in the VCXO.) The gain is down 3 dB (jitter bandwidth) at 4 Hz. The test equipment configuration for measuring the closed loop gain transfer function of Fig. 6 is shown in Fig. 7.

The calculated magnitude of the 8-kHz first sideband off the PLL output carrier is -52 dB; the measured level is -50 dB. This is shown in Fig. 8, a spectrum display from an HP 8557A. The peak phase jitter is 0.279 degree; the measured value from an HP 8901A modulation analyzer is 0.343 degree.

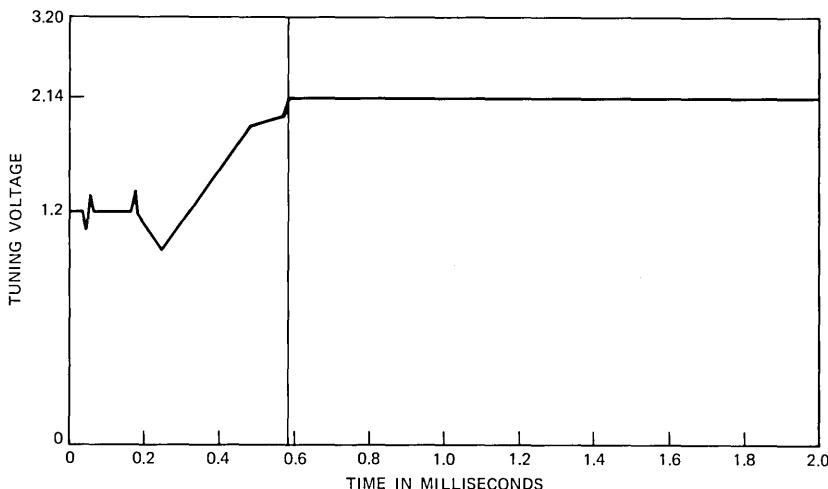


Fig. 4—Tuning voltage versus time during acquisition measured on a Paratronics 5000 logic analyzer.

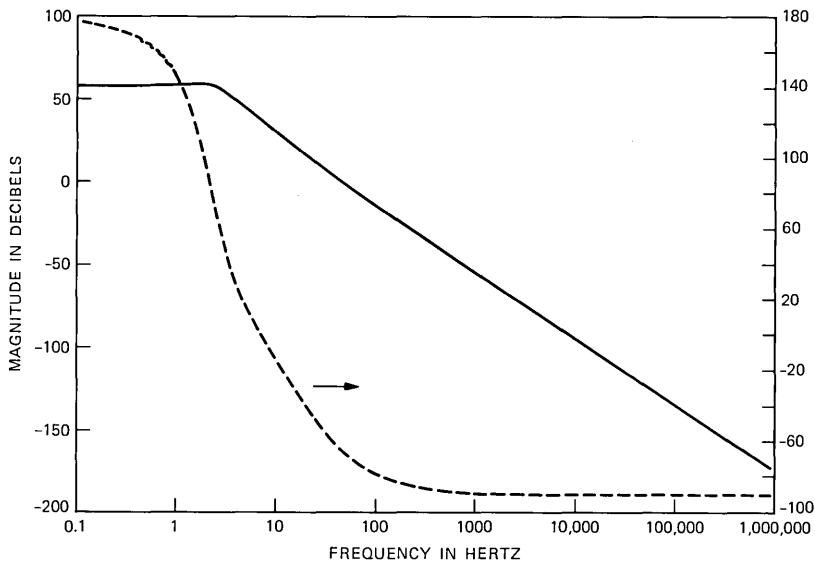


Fig. 5—PLACE graphic display of closed loop gain. The magnitude is indicated by the solid line, and the phase is indicated by the dotted line (right scale).

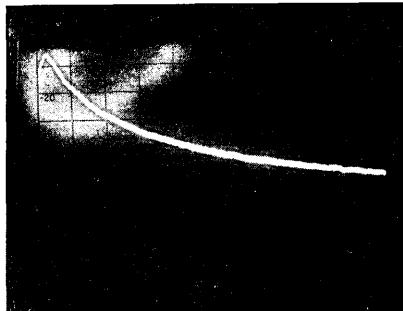


Fig. 6—Measured closed loop gain versus frequency showing 3 dB of jitter peaking. Vertical scale is 10 dB/division, horizontal scale is 5 Hz/division, and resolution bandwidth is 1 Hz.

The sensitivity analysis is also found in Appendix B. The upper and lower bounds found there reflect a 5-percent tolerance on the entered VCO gain constant, and 2-percent tolerance on the loop filter resistors and capacitors.

The remainder of this paper explains the calculations that PLACE performs.

IV. ANALYSIS

4.1 Stability analysis

The stability analysis requires computation of the PLL transfer

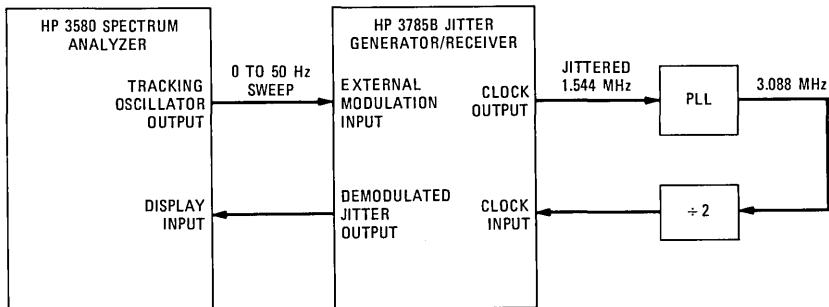


Fig. 7—Test equipment configuration used to obtain the closed loop gain transfer function shown in Fig. 6.

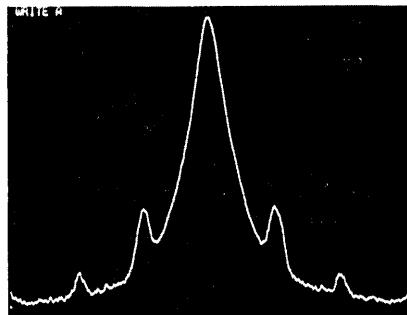


Fig. 8—Spectrum of VCO output carrier showing 8-kHz sidebands down 50 dB. Vertical scale is 10 dB/division, horizontal scale is 5 kHz/division, and resolution bandwidth is 1 kHz.

function. The components of a PLL are shown in Fig. 1; constants are defined in Section 2.1.

For the following calculations, we define $K_{Vr} = 2\pi K_V$ = VCO gain constant in rad/s/V. We also define the lumped gain constant:

$$K = \frac{K_p K_{Vr}}{N_{FB}}.$$

The equivalent frequency domain model of the PLL is also shown in Fig. 1, where $F(s)$ and $V_F(s)$ are the Laplace transforms of the loop filter transfer function and loop filter output voltage, respectively. The VCO model of K_{Vr}/s is derived as follows:

$$\begin{aligned} f_{out} &= \frac{d\phi_{out}}{dt} = K_{Vr} v_F \\ L\left(\frac{d\phi_{out}}{dt}\right) &= s\phi_{out}(s) = K_{Vr} V_F(s) \\ \phi_{out}(s) &= \frac{K_{Vr} V_F(s)}{s}. \end{aligned}$$

4.1.1 Loop gains

The transfer function for the PLL may be easily derived if we define the feed-forward gain as

$$G(s) = \frac{\phi_{\text{out}}(s)}{\phi_e(s)} = K_p K_{Vr} F(s)/s \quad (1)$$

and the feedback gain as

$$\beta(s) = 1/N_{FB}. \quad (2)$$

Then, following conventional control theory analysis, the open loop gain is the product of the feed-forward and feedback gains:

$$\beta(s)G(s) = \frac{K_p K_{Vr} F(s)}{s N_{FB}} = \frac{K F(s)}{s}. \quad (3)$$

The closed loop gain is

$$\begin{aligned} H(s) &= \frac{\phi_{\text{out}}(s)}{\phi_{\text{in}}(s)} = \frac{G(s)}{1 + \beta(s)G(s)} \\ &= \frac{K_p K_{Vr} F(s)/s}{1 + K_p K_{Vr} F(s)/s N_{FB}} = \frac{K_p K_{Vr} F(s)}{s + K_p K_{Vr} F(s)/N_{FB}} = \frac{N_{FB} K}{\frac{s}{F(s)} + K}. \end{aligned} \quad (4)$$

The error transfer function $E(s) = (\phi_e(s))/(\phi_{\text{in}}(s))$ is easily found as

$$E(s) = \frac{\phi_e(s)}{\phi_{\text{in}}(s)} = \frac{1}{1 + \beta(s)G(s)}. \quad (5)$$

Note that if there is no feedback counter, $\beta(s) = 1$, and then $E(s) = 1 - H(s)$.

The loop type is specified by the number of poles at the origin of $\beta(s)G(s)$. The loop order is specified by the total number of poles in $\beta(s)G(s)$. We have ignored the phase delay through the feedback counter in the above analysis; this phase delay is treated in Section 4.1.5.

4.1.2 Damping, second-order undamped natural frequency ω_n

From the above equations we see that the PLL response is highly dependent on the form of loop filter $F(s)$. By substituting a particular loop filter function $F(s)$ into eq. (4), we can (except for the no loop filter case) get the denominator into the classic second-order control system form of $s^2 + 2\xi\omega_n s + \omega_n^2$, where ω_n is the undamped natural frequency and ξ is the damping. This is shown in Appendix C.

4.1.3 Phase margin

PLACE calculates the phase of $\beta(s)G(s)$ at unity open loop gain; the difference from -180 degrees is the phase margin. This definition is identical to the stability analysis of feedback amplifiers.

4.1.4 Phase margin degradation due to the VCO (and/or op amp) pole

A parasitic pole causes the open loop gain to fall off more quickly and thus affect the phase margin. The tuning element of a VCO (typically a varactor) cannot respond to rapidly changing tuning voltages. We represent this by assigning a pole to the VCO. It has been the author's experience that this VCO pole may be the dominant pole of the PLL (especially if VCXOs are used). PLACE determines the phase margin degradation due to the parasitic VCO pole by multiplying the open loop gain by $1/(1 + \tau_v s)$, where $\tau_v = 1/(2\pi f_v)$, where f_v is the VCO pole. The VCO pole may be found by impressing an ac modulating signal on the dc voltage to the varactor, and determining the highest modulating frequency for which the VCO output will follow the input. (A typical value for a VCXO is $f_v = 10$ Hz.)

If the user wants to account for the operational amplifier pole for the active loop filter case, he may lump this pole into the VCO parasitic pole. The designer should try to keep the operational amplifier bandwidth much larger than the PLL system bandwidth (defined in Section 4.1.6).

4.1.5 Phase margin degradation due to the feedback counter

The effect of the feedback counter is to add a phase delay of f/f_{ref} to the open loop gain and thus degrade the phase margin by f_u/f_{ref} rad, where f_u is the frequency for unity open loop gain and f_{ref} is the frequency entering the phase detector.^{4,5} This is derived as follows.

Since any change in the VCO frequency can be observed by the phase detector only after the feedback counter overflows T seconds later (worst case), the effect of the counter is to produce a maximum delay of up to T seconds, where the delay time T is given by

$$T = N_{FB}(1/f_{VCO}) = N_{FB} \frac{1}{N_{FB}f_{ref}} = 1/f_{ref}.$$

Thus, in the frequency domain we represent the feedback counter by

$$\frac{1}{N_{FB}} e^{-sT} = \frac{1}{N_{FB}} e^{-j2\pi(f/f_{ref})}.$$

PLACE calculates this phase shift and notifies the user of the resulting phase margin degradation.

4.1.6 PLL system bandwidth in hertz

The PLL system bandwidth is the frequency for PLL unity open loop gain; it is found by solving $|\beta(j\omega)G(j\omega)| = 1$ for ω . We have the following:

No loop filter:

$$F(s) = 1,$$

and

$$\text{PLL}_{\text{BW}} = \frac{1}{2\pi} [K] \text{ Hz.}$$

RC loop filter:

$$F(s) = \frac{1}{(1 + \tau s)},$$

and

$$\begin{aligned} \text{PLL}_{\text{BW}} &= \left[\frac{1}{2\pi} \right] \frac{1}{\sqrt{2\tau}} \{ \sqrt{1 + 4(\tau K)^2} - 1 \}^{1/2} \\ &= \left[\frac{1}{2\pi} \right] \sqrt{2} \zeta \omega_n \{ \sqrt{1 + 1/(4\zeta^4)} - 1 \}^{1/2} \text{ Hz.} \end{aligned}$$

Lag-lead loop filter:

$$F(s) = \frac{1 + \tau_2 s}{1 + \tau_1 s},$$

and

$$\text{PLL}_{\text{BW}} = \left[\frac{1}{2\pi} \right] \frac{1}{\sqrt{2\tau_1}} \{ \sqrt{a^2 + 4(\tau_1 K)^2} + a \}^{1/2} \text{ Hz,}$$

where $a = \tau_2^2 - 1$.

Second-order active loop filter:

$$F(s) = (1 + \tau_2 s)/(\tau_1 s),$$

and

$$\text{PLL}_{\text{BW}} = \left[\frac{1}{2\pi} \right] \omega_n \{ 2\zeta^2 + \sqrt{4\zeta^4 + 1} \}^{1/2} \text{ Hz.}$$

For the third-order active loop filter, the user may specify the desired PLL bandwidth, or, alternatively, the value defaults to $f_{\text{ref}}/50$. PLACE uses this value to optimize the loop for best phase noise performance (see Appendix C).

4.2 Loop filter analysis

4.2.1 Loop filter time constants

The loop filter time constants are calculated from the undamped natural frequency and the damping by using the equations derived in Appendix C. Alternatively, the user may directly specify the loop filter time constants.

4.2.2 Loop filter resistors and capacitors

PLACE assumes 0.1- μ f capacitors and calculates the resistors from the definition of the time constants. Note that the user can linearly scale the resistors and capacitors to any desired value (e.g., if the user desires 0.01- μ f capacitors, multiply the calculated resistor values by 10).

4.2.3 Constraints on lag-lead loop filter

To maintain real values of resistors and capacitors, certain constraints must be met for the lag-lead loop filter case. PLACE first asks for the desired value of ω_n . Then, when it asks for the desired value of damping, the entered value for damping must satisfy two conditions:

1. Since $\tau_2 = 2\xi/\omega_n - 1/K$, to ensure a positive τ_2 we must ensure

$$\xi > \frac{\omega_n}{2K}.$$

2. Since $R_1 = (\tau_1 - \tau_2)/C$, to ensure a positive R_1 we must ensure

$$\tau_1 > \tau_2$$

$$\frac{K}{\omega_n^2} > \frac{2\xi}{\omega_n} - \frac{1}{K}$$

$$\xi < \frac{K^2 + \omega_n^2}{2\omega_n K}.$$

Hence, after the user specifies the desired value for ω_n , PLACE asks the user to enter a value for damping that satisfies

$$\frac{\omega_n}{2K} < \xi < \frac{K^2 + \omega_n^2}{2\omega_n K}.$$

4.2.4 Loop filter 3-dB frequency or zero frequency

The frequency at which $F(s)$ is down 3 dB is found by solving $|F(j\omega)| = 0.707$ for ω . We have the following (iff $\tau_1 > \sqrt{2}\tau_2$, else $F(s)$ is never down 3 dB):

RC loop filter:

$$F(s) = \frac{1}{1 + \tau s},$$

and

$$F_{\text{BW}} = \frac{1}{2\pi} \left[\frac{1}{\tau} \right] \text{ Hz.}$$

Lag-lead loop filter:

$$F(s) = \frac{1 + \tau_2 s}{1 + \tau_1 s},$$

and

$$F_{\text{BW}} = \frac{1}{2\pi} \left[\frac{1}{\tau_1^2 - 2\tau_2^2} \right]^{1/2} \text{ Hz.}$$

For the second and third active loop filter cases, F_{BW} is undefined because $F(s)$ has a pole at the origin. For these cases PLACE calculates the zero frequency.

4.3 Tracking analysis

4.3.1 Hold range

The hold range is defined as the maximum input frequency range that the PLL will track once it is in lock. It may be found experimentally by slowly varying the input frequency to a PLL that is in lock and noting when lock is lost. (The word “slowly” is emphasized because if the input frequency has a step change, the transient behavior of the loop may cause the PLL to lose lock, even though the step is within the hold range.)

The hold range is independent of the type of loop filter $F(s)$ and is given by numerous authors (e.g., see Ref. 6) as $K_p K_{Vr} F(0)/N_{\text{FB}}$, where $F(0)$ is the dc gain of the loop filter. (Note that the error voltage is a constant dc value when the loop is in lock.) For passive loop filters $F(0) = 1$, and accounting for the feed-forward counter, PLACE computes the *hold range* as

$$\pm \frac{1}{2\pi} N_{\text{FF}} \frac{K_p K_{Vr}}{N_{\text{FB}}} \text{ Hz.} \quad (6)$$

For active loop filters $F(0)$, and hence the hold range, can be made arbitrarily large (until the VCO or operational amplitude saturates); thus PLACE computes the normalized hold range (it assumes $F(0) = 1$).

When recovering timing from Pulse Code Modulated (PCM) data lines, the hold (and capture) range may be asymmetrical due to the density of ones.⁷

Nonsinusoidal phase detectors theoretically extend the hold and capture ranges. For example, triangular phase detectors have a so-called "extension factor" of $\pi/2$, while sawtooth phase detectors have a factor of π (see Ref. 6). PLACE leaves it to the user to modify the phase detector gain constant K_p (if the designer is so inclined) when PLACE requests the value.

4.3.2 Capture and pull-in range

The *capture* range is defined as the maximum input frequency range for which a PLL will acquire phase lock without skipping cycles (i.e., the phase detector voltage monotonically drives the VCO toward lock, without any beating). (Some authors call this the lock-in range.) The *pull-in* range is defined as the maximum input frequency range for which a PLL will acquire lock (typically with skipping cycles), even if it takes minutes or hours to lock up. (In this case the phase detector output voltage may be a beat note swinging the VCO above and below f_{in} .)

Whereas eq. (6) gives an accurate value for the hold range, the determination of the capture and pull-in range is extremely difficult because the loop starts (by definition) out of lock, and thus a nonlinear analysis is required. The solution cannot be expressed in closed form (except for the $F(s) = 1$ case), and only a graphical phase-plane trajectory procedure^{6,8} will yield true results. PLACE approximates the capture and pull-in ranges by using the approximating equations found in the literature and modifying them to account for the feedback and feed-forward counters:

No loop filter:

$$f_{capture} = f_{hold} = f_{pull} = \pm \frac{1}{2\pi} N_{FF} \frac{K_p K_{Vr}}{N_{FB}} \text{ Hz}$$

(see Refs. 8 and 9).

RC loop filter: If $K\tau_1 < 0.25$,

$$f_{capture} = f_{hold} = f_{pull} = \pm \frac{1}{2\pi} N_{FF} \frac{K_p K_{Vr}}{N_{FB}} \text{ Hz}$$

(see Ref. 10).

Else,

$$f_{capture} = \pm \frac{1}{2\pi} N_{FF} 2\zeta \omega_n \text{ Hz},$$

and

$$f_{\text{pull}} = \pm \frac{1}{2\pi} N_{\text{FF}} 1.25 \omega_n \text{ Hz}$$

(see Ref. 8).

Lag-lead loop filter:

$$f_{\text{capture}} = \pm \frac{1}{2\pi} N_{\text{FF}} K \frac{\tau_2}{\tau_1} = \frac{1}{2\pi} N_{\text{FF}} \tau_2 \omega_n^2 \text{ Hz},$$

(see Refs. 8 and 10), and

$$f_{\text{pull}} = \frac{1}{2\pi} N_{\text{FF}} 2 \left\{ K \zeta \omega_n + \frac{K}{2\tau_1} \right\}^{1/2} \text{ Hz}$$

(see Refs. 10 and 11).

Second-order active filter:

$$f_{\text{capture}} = \pm \frac{1}{2\pi} N_{\text{FF}} 2 \zeta \omega_n = \frac{1}{2\pi} N_{\text{FF}} \tau_2 \omega_n^2 \text{ Hz}$$

(see Ref. 8).

For the second- and third-order active loop filter cases, the hold and pull-in ranges will be as large as the maximum frequency range of the VCO, assuming the operational amplifier output does not saturate at the supply rail, and assuming negligible loop delay.

4.3.3 Lockup time

The lockup time is defined for PLACE as the time it takes the PLL to acquire phase and frequency lock from an initial frequency offset equal to the pull-in range. For the no loop filter case and low gain ($K\tau_1 < 0.25$) passive loop filter case, PLACE calculates the lockup time as $T_{\text{lock}} = 1/K$ seconds.⁸

For second-order active loop filters and high-gain ($K\tau_1 > 0.25$) passive loop filters, the lockup time is approximately given by $T_{\text{lock}} = \Delta\omega^2/2\zeta\omega_n^3$ (see Refs. 6 and 8). PLACE lets $\Delta\omega = ([f_{\text{capture}} + f_{\text{pull}}]/2N_{\text{FF}})2\pi$, which is where the approximation is most accurate; for the second-order loop filter, PLACE lets $\Delta\omega = 2\pi(2f_{\text{capture}})$. PLACE does not calculate the lockup time for the third-order loop filter.

4.4 Jitter analysis

In the discussion that follows, the term “reference signal” is the signal entering the phase detector; thus, if $N_{\text{FF}} = 1$, the reference signal is the incoming signal itself. Also, the terms jitter and phase noise are equivalent.

The output jitter of a PLL may be conveniently separated into two main components: jitter generation and jitter propagation. Jitter propagation refers to the output jitter due to jitter on the incoming signal (or the reference signal for frequency synthesizer applications). Jitter generation may be subdivided into five components: output jitter due to leakage of the reference signal through the phase detector; output jitter due to VCO phase noise; and output jitter due to phase noise of the phase detector, frequency divider, and loop filter (if an active loop filter is employed). PLACE analyzes each component of jitter separately.

The most important result of this section may be summarized (without mathematics) as follows: A PLL acts as a high-pass filter to VCO phase noise, and acts as a low-pass filter to incoming reference signal jitter.

4.4.1 Output jitter due to jitter on the incoming reference signal

4.4.1.1 Jitter bandwidth. It may easily be shown that the PLL's response to jitter on the incoming reference signal is given by the closed loop gain transfer function $H(s)$, eq. (4). Merely repeat the analysis of Section 4.1.1 with ϕ_i now representing an instantaneous phase deviation of the incoming reference frequency. Equation (4) is repeated here for convenience:

$$H(s) = \frac{\phi_{\text{out}}(s)}{\phi_{\text{in}}(s)} = \frac{G(s)}{1 + \beta(s)G(s)} = \frac{K_p K_{Vr} F(s)/s}{1 + K_p K_{Vr} F(s)/s N_{FB}}$$

$$= \frac{K_p K_{Vr} F(s)}{s + K_p K_{Vr} F(s)/N_{FB}} = \frac{N_{FB} K}{(s/F(s)) + K}.$$

Thus, above the frequency for unity closed loop gain $H(s)$, the input signal jitter will be attenuated. Thus a PLL acts as a low-pass filter to jitter on the incoming reference signal. Also, notice from eq. (4) that within the PLL system bandwidth, a PLL multiplies the reference noise by the feedback counter divisor N_{FB} . That is, the feedback counter effectively adds $20 \log N_{FB}$ dB/Hz phase noise to the reference signal.

The jitter bandwidth is the frequency at which $H(s)$ is down 3 dB from its dc value. If we neglect any parasitic VCO or operational amplifier poles, we have the following:

No loop filter:

$$F(s) = 1,$$

and

$$\text{jitter}_{\text{BW}} = \frac{1}{2\pi} [K] \text{ Hz.}$$

RC loop filter:

$$F(s) = \frac{1}{1 + \tau s},$$

and

$$\text{jitter}_{\text{BW}} = \left[\frac{1}{2\pi} \right] \omega_n \{ (2\zeta^2 + 1) + \sqrt{(2\zeta^2 - 1)^2 + 1} \}^{1/2} \text{ Hz}.$$

Lag-lead loop filter:

$$F(s) = \frac{1 + \tau_2 s}{1 + \tau_1 s},$$

and

$$\text{jitter}_{\text{BW}} = \left[\frac{1}{2\pi} \right] \omega_n \{ a + \sqrt{a^2 + 1} \}^{1/2} \text{ Hz},$$

where

$$a = 2\zeta^2 + 1 - \frac{4\zeta\omega_n}{K} + \frac{\omega_n^2}{K}.$$

Second-order active loop filter:

$$F(s) = \frac{1 + \tau_2 s}{\tau_1 s},$$

and

$$\text{jitter}_{\text{BW}} = \left[\frac{1}{2\pi} \right] \omega_n \{ 2\zeta^2 + 1 + \sqrt{(2\zeta^2 + 1)^2 + 1} \}^{1/2} \text{ Hz}.$$

When there are parasitic poles (which often reduce the bandwidth values calculated above), PLACE uses an iterative search to find the jitter bandwidth.

4.4.1.2 Frequency and magnitude of jitter peaking. The peak jitter gain $|H(j\omega_{\text{peak}})|$ is equivalent to the peak magnitude of $H(s)$. *Jitter peaking* is the difference (in decibels) between the closed loop gain peak magnitude $|H(j\omega_{\text{peak}})|$ and the dc value of $H(s)$. To find it we first find the frequency of peak jitter gain ω_{peak} by setting the derivative of $|H(j\omega)|^2$ to zero and solving for ω_{peak} . Then we substitute ω_{peak} back into $H(j\omega)$. Neglecting any parasitic VCO or operational amplifier poles, we get the following results:

No loop filter:

$$F(s) = 1,$$

and

$$\omega_{\text{peak}} = 0$$

and

$$|H(j\omega_{\text{peak}})| = 0.$$

RC loop filter:

$$F(s) = \frac{1}{1 + \tau s},$$

$$\omega_{\text{peak}} = \omega_n \sqrt{1 - 2\zeta^2},$$

and

$$|H(j\omega_{\text{peak}})| = \frac{1}{2\zeta\sqrt{1 - \zeta^2}}.$$

Second-order active loop filter:

$$F(s) = \frac{1 + \tau_2 s}{\tau_1 s},$$

$$\omega_{\text{peak}} = \frac{\omega_n}{2\zeta} \left\{ \sqrt{1 + 8\zeta^2} - 1 \right\}^{1/2},$$

and

$$|H(j\omega_{\text{peak}})| = (4\zeta^2) \left\{ \frac{1}{[4\zeta^2 - 1]^2 - 3 + 2\sqrt{(1 + 8\zeta^2)}} \right\}^{1/2}.$$

For the lag-lead and third-order active loop filters, PLACE uses an iterative search to find the jitter peaking. The above equations do not reveal the jitter peaking due to parasitic poles. PLACE uses an iterative search to find the jitter peaking in such cases.

4.4.1.3 Noise bandwidth. The noise bandwidth of a PLL is the bandwidth of an equivalent rectangular filter that would yield the same output noise power (variance) as the PLL, if they both have white noise inputs of equal density. PLACE calculates the one-sided noise bandwidth, where $N_{\text{BW}} = \int_0^\infty |H(j\omega)|^2 d\omega$. Various authors have evaluated the integrals, and we have the following:

No loop filter:

$$F(s) = 1,$$

and

$$N_{\text{BW}} = \frac{K}{4} \text{ Hz.}$$

RC loop filter:

$$F(s) = \frac{1}{1 + \tau s},$$

and

$$N_{\text{BW}} = \frac{K}{4} = \frac{\omega_n}{8\zeta} \text{ Hz.}$$

Lag-lead loop filter:

$$F(s) = \frac{1 + \tau_2 s}{1 + \tau_1 s},$$

and

$$N_{\text{BW}} = \frac{K}{4} \left[\frac{K + 1/\tau_2}{K + 1/\tau_1} \right] \text{ Hz.}$$

Second-order active loop filter:

$$F(s) = \frac{1 + \tau_2 s}{\tau_1 s},$$

and

$$N_{\text{BW}} = \frac{\omega_n}{2} \left(\zeta + \frac{1}{4\zeta} \right) \text{ Hz.}$$

For the third-order active loop filter, the noise bandwidth is approximately given by the PLL system bandwidth defined earlier.

The output noise of the PLL, given a white noise input with power P_{in} is given by $P_{\text{out}} = P_{\text{in}} N_{\text{BW}}$. Similarly, the output signal-to-noise ratio of the PLL is inversely proportional to the noise bandwidth.

4.4.2 Output jitter due to leakage of the reference through the phase detector

4.4.2.1 Frequency and magnitude of first sideband. Any feedthrough of the reference frequency through the loop filter phase modulates the VCO, producing sidebands at the VCO output. The frequency of the first sideband is the frequency leaving the phase detector: twice the reference frequency for exclusive-or gate phase detectors, the reference frequency for other phase detectors. The magnitude of the first sideband is found as follows: the modulating frequency is $f_m = f_{\text{ref}}$, and the Modulation Index (MI) is $\text{MI} = \Delta f/f_m = \Delta f/f_{\text{ref}}$. Using conventional frequency modulation/phase modulation analysis, we can express the resulting VCO output voltage $v(t)$ by using Bessel functions as

$$V_0\{J_0(\text{MI})\cos(\omega_0 t) \pm J_1(\text{MI})\cos(\omega_0 \pm \omega_m)t\},$$

where we have neglected the higher-order sidebands. Now, if the modulation index is small ($\text{MI} < 0.3$), we have $(J_1(\text{MI}))/ (J_0(\text{MI})) = \text{MI}/2$. For example, $(J_1(0.2))/(J_0(0.2)) = 0.0995/0.9900 = 0.1 = \text{MI}/2$. Hence,

$$\frac{\text{First sideband amplitude}}{\text{VCO carrier amplitude}} = \frac{J_1(\text{MI})}{J_0(\text{MI})} = \frac{\text{MI}}{2} = \frac{\Delta f}{2f_m} = \frac{\Delta f}{2f_{\text{ref}}}. \quad (7)$$

When we realize that most of the signal power is in the carrier for small modulation indices, we see that the above equation is the definition of the $\mathcal{L}(f)$ (see Refs. 12 and 13) used in specifying oscillator phase noise.

Now, Δf is the frequency deviation of the VCO:

$$\Delta f = \Delta V_F K_V = V_{\text{pk}} |F(j\omega)| K_V,$$

where V_{pk} is the peak amplitude of f_{ref} at the phase detector output. (For exclusive-or gate phase detectors, the phase detector output frequency is twice f_{ref} .)

Thus, the magnitude of the first sideband (relative to the carrier) at the PLL output is given by

$$20 \log V_{\text{pk}} \left[\frac{K_V |F(j\omega)|}{2f_{\text{ref}}} \right], \quad (8)$$

where $F(j\omega)$ is evaluated at ω_{ref} . (This result agrees with Ref. 14.) PLACE asks the user for the value of V_{pk} , the peak voltage output from the phase detector.

4.4.2.2 Peak frequency deviation and phase jitter. It can be shown that the peak phase deviation is related to $\mathcal{L}(f)$ by the following relation:¹⁵

$$\mathcal{L}(f) = \frac{\Delta\phi_{\text{peak}}}{2}.$$

The magnitude of $\mathcal{L}(f)$ is given by eq. (7); PLACE 2.0 solves the above equation for the peak phase deviation in radians. Then, the peak phase jitter in degrees is 57.3 times as large. The peak frequency deviation is found by using $\Delta f_{\text{peak}} = f_m \Delta\phi_{\text{peak}}$.

As long as the peak phase jitter is less than approximately 28 degrees, the PLL will generally stay in lock (see Ref. 16).

4.4.2.3 Loop filter's attenuation of reference frequency. PLACE calculates the magnitude of $F(s)$ at the reference frequency and informs the user of the result.

4.4.3 Output jitter due to PLL components

4.4.3.1 Output jitter due to VCO phase noise.

PLL's relative response to VCO phase noise (i.e., it calculates the PLL's attenuation of VCO phase noise); PLACE does not calculate the absolute value of the VCO-generated phase noise itself. (References 9, 12, and 17 explain how to measure the absolute VCO phase noise.)

A VCO may be modeled as a pure signal source corrupted by an instantaneous phase fluctuation $\phi_{\text{vco}}(t)$. We modify the PLL model by adding this noise source to the VCO of Fig. 1. The total PLL output phase is now $\phi_{\text{out}}(t) = \phi_{\text{loop}}(t) + \phi_{\text{vco}}(t)$, where $\phi_{\text{loop}}(t)$ is the output phase for a PLL with an ideal VCO. We now have the following:

$$\phi_e(s) = \phi_{\text{in}}(s) - \phi_{\text{out}} \frac{(s)}{N_{\text{FB}}},$$

but

$$\phi_{\text{out}}(s) = \phi_e(s)F(s)K_pK_V/s + \phi_{\text{vco}}(s),$$

so that

$$\phi_e = \frac{\phi_{\text{out}} - \phi_{\text{vco}}}{F(s)K_pK_V/s}.$$

Algebraic manipulations yield

$$\begin{aligned}\phi_{\text{out}} &= \phi_{\text{in}} \frac{N_{\text{FB}}KF(s)/s}{1 + KF(s)/s} + \phi_{\text{vco}} \frac{1}{1 + KF(s)/s} \\ \phi_{\text{out}} &= \phi_{\text{in}} \frac{G(s)}{1 + \beta(s)G(s)} + \phi_{\text{vco}} \frac{1}{1 + \beta(s)G(s)}. \end{aligned} \quad (9)$$

Thus, the PLL response to VCO phase noise is given by

$$\frac{1}{1 + \beta(s)G(s)}. \quad (10)$$

(This result is confirmed in Refs. 5, 8, and 13.) The above equation tells us that the -3 dB point of the PLL response to VCO phase noise occurs when $|\beta(s)G(s)| = 1$; but this occurs at the PLL system bandwidth (defined in Section 4.1.6). Hence, the PLL will pass through the VCO phase noise unattenuated above the PLL system bandwidth. Below the PLL system bandwidth, the output jitter due to VCO phase noise is reduced by the loop gain $\beta(s)G(s)$.

In short, a PLL acts as a high-pass filter to VCO phase noise.

4.4.3.2 Output jitter due to frequency divider phase noise. In Section 4.4.1.1 it was shown that the feedback counter effectively adds $20 \log N_{\text{FB}}$ dB/Hz phase noise to the reference signal; this section discusses the jitter generated in the frequency divider itself.

Phase noise generated in digital frequency dividers is due to the

internal active devices. Typical noise floors for Transistor-Transistor Logic (TTL), Emitter-Coupled Logic (ECL), and Metal Oxide Semiconductor (MOS) dividers are -120 to -140 dB. TTL dividers have the lowest noise, followed by ECL, and then MOS. (Typical values may be found in Ref. 9.)

As in the case for VCO phase noise, PLACE does not need to know the absolute value of the divider noise. The PLL response to this noise is the same as for jitter on the incoming signal, because the PLL cannot tell whether the noise comes from the divider or signal inputs to the phase detector. Thus, the PLL response to divider noise is identical to the low-pass filter shape computed for incoming line jitter. [The PLL response to noise from an active loop filter is also given by eq. (4).]

In timing recovery circuits, the divider noise is negligible compared with the input signal jitter. In frequency synthesizers, however, the two may be equal, and then the resultant noise equals their rms sum.

4.4.3.3 Output jitter due to phase detector phase noise. Typical noise floors for phase detectors are -150 to -160 dB/Hz (see Ref. 15). The PLL response to this phase noise is again given by eq. (4). The phase detector noise floor improves 3 dB per octave of reference frequency reduction.

4.5 Sensitivity analysis

PLACE asks the user for the tolerance of the VCO and phase detector gain constants, and the tolerance on the loop filter resistors and capacitors. It then performs a Taylor series expansion and retains the first-order terms. Then it calculates upper and lower bounds for the hold range, capture and pull-in ranges, and lockup time. This allows the designer to observe the effect of component variation on the PLL.

4.6 Optimization routine

PLACE 2.0 features an interactive routine that allows the designer to optimize PLL performance. The user specifies either a larger hold range, larger capture and pull-in range, faster lockup time, or less output jitter. After the user specifies a desired item to optimize, PLACE automatically adjusts the PLL damping and natural frequency to achieve the desired goal.

V. CONCLUSION

An interactive program to aid in the development of PLLs has been written. PLACE has been successfully used in the design of PLLs for digital channel banks, muldem, and radio transmission systems. In

addition, routine use of PLACE on several existing PLL designs revealed problems that were subsequently corrected before the product was placed in the field. The program has saved many PLL designers from tedious calculations, allowing a better understanding of PLL performance.

VI. ACKNOWLEDGMENT

The author wishes to thank M. Luniewicz for his encouragement and inspiration.

REFERENCES

1. C. Corsetto, "RF Synthesizers," *Ham Radio*, 16, No. 10 (October 1983), pp. 17-26.
2. W. Egan, *Frequency Synthesis by Phase Lock*, New York: Wiley, 1981, pp. 29-98.
3. R. Becker and J. Chambers, unpublished work.
4. A. Przedpelski, "PLL Primer," *RF Des.*, 6, No. 2 (July/August 1983), pp. 48-53.
5. J. Gorski-Popiel, *Frequency Synthesis: Techniques and Applications*, IEEE Press, 1975, pp. 70-105.
6. F. Gardner, *Phaselock Techniques*, 2nd ed., New York: Wiley, 1979, pp. 55-79.
7. C. Crue and A. Ansell, unpublished work.
8. V. Kroupa, *Frequency Synthesis*, New York: Wiley, 1973, pp. 176-8.
9. V. Manassewitsch, *Frequency Synthesizers, Theory and Design*, New York: Wiley, 1980, pp. 254, 574.
10. A. Blanchard, *Phase-Locked Loops*, New York: Wiley, 1972, pp. 260-6.
11. J. Klapper and J. Frankle, *Phase-Locked and Frequency-Feedback Systems*, New York: Academic, 1972, p. 102.
12. A. Lance et al., "Automated Phase Noise Measurements," *Microw. J.*, 20, No. 6 (June 1977), pp. 87-96.
13. L. Martin, "Noise Property Analysis Enhances PLL Designs," *Elec. Des. News*, 26, No. 18 (September 1981), pp. 91-8.
14. "Application Note for MC4044," MECL Integrated Circuits, Motorola, Inc., 1983, pp. 7-33.
15. W. Robins, *Phase Noise in Signal Sources*, London: Peregrinus Ltd., 1982, pp. 18-46.
16. R. Best, *Phase-Locked Loops*, New York: McGraw-Hill, 1984, p. 60.
17. J. Payne, "Measure and Interpret Short-Term Stability," *Microwaves*, 15, No. 7 (July 1976), pp. 34-45.
18. A. Przedpelski, "Optimize PLL's to Meet Your Needs," *Electron. Des.*, 26, No. 23 (September 1978), pp. 134-7.

APPENDIX A

List of Symbols

$E(s)$	Error transfer function
$F(s)$	Loop filter transfer function
F_{BW}	Loop filter 3-dB frequency
f_m	Frequency of modulation
f_{ref}	Reference frequency
f_u	Frequency for unity open loop gain
f_v	Frequency of VCO pole
f_{VCO}	Frequency of VCO
$f_{capture}$	Capture range
f_{hold}	Hold range

f_{pull}	Pull-in range
$G(s)$	Feed-forward gain
$H(s)$	Closed loop gain
$ H(j\omega_{\text{peak}}) $	Magnitude of jitter peaking
K	Lumped gain constant, $K = (K_p K_{V_r})/N_{\text{FB}}$
K_p	Phase detector gain constant in V/rad
K_V	VCO gain constant in Hz/V
K_{V_r}	VCO gain constant in rad/s/V
N_{BW}	Noise bandwidth in hertz
N_{FB}	Feedback counter divisor
N_{FF}	Feed-forward counter divisor
PLL_{BW}	PLL bandwidth
T_{lock}	Lockup time
v_F	Loop filter output voltage
$V_F(s)$	Loop filter output voltage
$\beta(s)$	Feedback gain
Δf	Frequency deviation
Δf_{peak}	Peak frequency deviation
$\Delta \phi_{\text{peak}}$	Peak phase deviation
ζ	Damping
$\phi_{\text{in}}(s)$	Input phase
$\phi_{\text{out}}(s)$	Output phase
$\phi_e(s)$	Phase error
ω_n	Natural frequency
ω_{peak}	Frequency of jitter peaking

APPENDIX B

PLACE 2.0 Output for Lag-Lead Loop Filter Example

Stability Analysis:

2nd-Order Undamped Natural Frequency (Hz)	Phase Margin (degrees)	PLL Bandwidth (Open Loop Gain = 1) (Hz)	Damping
2.0	59.3	1.2	0.7

Your VCO pole at 10.0 Hz reduced the phase margin by 7.41 degrees, and reduced the PLL bandwidth by 0.1 Hz.

Your feedback counter reduces the phase margin by 0.02 degree.

Loop Filter Analysis:

Assuming 0.1 uf capacitor, our loop filter components are:

					Loop Filters
T1=(R1+R2) C	T2=R2 (C)	R1(ohms)	R2(ohms)	C(uf)	3-dB Freq.(Hz)
0.05745130	0.00400336	534479	40034	0.100000	2.8

Tracking Analysis:

Hold Range (+/- Hz)	Capture Range (+/- Hz)	Pull-In Range (+/- Hz)	Lockup Time (sec)
568.0	39.6	568.0	0.32447433

Jitter Analysis:

1. Output Jitter Due to Incoming Reference Signal Jitter:

Frequency of Jitter Peaking (Hz)	Magnitude of Jitter Peaking (dB)	Noise Bandwidth (Hz)	Jitter Bandwidth (Hz)
1.6	1.272	22.5	3

2. Output Jitter Due to Reference Leakage Through the Phase Detector:

Frequency of 1st Sideband (Hz)	Magnitude of 1st Sideband (dBc)	Peak Phase Jitter (degrees)	Peak Frequency Deviation (Hz)	Ref. Freq. Attenuation (dB)
8000	-52	0.278	19.4	23

3. Output Jitter Due to VCO Phase Noise:

Your PLL attenuates VCO phase noise BELOW 1.2 Hz.

Sensitivity Analysis:

	Hold Range (+/- Hz)	Capture Range (+/- Hz)	Pull-In Range (+/- Hz)	Lockup Time (sec)
Lower Bound:	539.0	36.1	539.0	0.30792615
Nominal:	568.0	39.6	568.0	0.32447433
Upper Bound:	597.0	43.1	597.0	0.34102252

APPENDIX C

Loop Gain Derivation

C.1 Case 1. No loop filter: $F(s) = 1$

From eq. (4), the closed loop gain is

$$H(s) = \frac{N_{FB}K}{s/F(s) + K} = \frac{N_{FB}}{1 + s/K},$$

the closed loop gain magnitude is

$$|H(j\omega)| = N_{FB} \left[\frac{1}{1 + \left(\frac{\omega}{K}\right)^2} \right]^{1/2},$$

and the closed loop gain phase is

$$\phi(j\omega) = -\arctan(\omega/K).$$

From eq. (3), the open loop gain is

$$\beta(s)G(s) = K/s,$$

the open loop gain magnitude is

$$|\beta(j\omega)G(j\omega)| = K/\omega,$$

and the open loop gain phase is

$$\theta(j\omega) = -90 \text{ degrees (due to the VCO)}.$$

Note that the phase margin is 90 degrees and the loop is theoretically unconditionally stable; but, if we account for any parasitic VCO pole (or any other parasitic poles lumped into the VCO pole), PLACE shows that the PLL can be unstable for very high loop gain K . No loop filter results in a first-order type-1 PLL.

C.2 Case 2. RC loop filter: $F(s) = 1/(1 + \tau s)$

From eq. (4), the closed loop gain is

$$H(s) = \frac{N_{FB}K}{s/F(s) + K} = \frac{N_{FB}K}{s^2\tau + s + K} = \frac{N_{FB}K/\tau}{s^2 + s/\tau + K/\tau}.$$

We can get the denominator into the classical second-order control system form of $s^2 + 2\xi\omega_n s + \omega_n^2$ if we let $\omega_n^2 = K/\tau$ and $2\xi\omega_n = 1/\tau$. Then,

$$H(s) = \frac{N_{FB}\omega_n^2}{s^2 + 2\xi\omega_n s + \omega_n^2} = \frac{N_{FB}}{\left(\frac{s}{\omega_n}\right)^2 + 2\xi\frac{s}{\omega_n} + 1},$$

where the *undamped natural frequency* is

$$\omega_n = \sqrt{K/\tau},$$

and the *damping* is

$$\xi = \frac{1}{2\omega_n\tau} = \frac{1}{2\sqrt{K\tau}}.$$

Thus, the closed loop gain magnitude is

$$|H(j\omega)| = N_{FB} \left[\frac{1}{\left[1 - \left(\frac{\omega}{\omega_n} \right)^2 \right]^2 + 4\xi^2 \left(\frac{\omega}{\omega_n} \right)^2} \right]^{1/2},$$

and the closed loop gain phase is

$$\phi(j\omega) = -\arctan \frac{2\xi \frac{\omega}{\omega_n}}{1 - \left(\frac{\omega}{\omega_n} \right)^2}.$$

From eq. (3), the open loop gain is

$$\beta(s)G(s) = \frac{K}{s(1 + \tau s)},$$

the open loop gain magnitude is

$$|\beta(j\omega)G(j\omega)| = \frac{K}{\omega} \frac{1}{\sqrt{1 + (\omega\tau)^2}},$$

and the open loop gain phase is $\theta(j\omega) = -90^\circ - \arctan \omega\tau$.

Note that the phase starts off at -90° degrees and approaches -180° degrees at high frequencies. The larger the value of τ , the lower the damping and the smaller the phase margin. The RC loop filter results in a second-order type-1 PLL.

C.3 Case 3. Lag-lead loop filter: $F(s) = (1 + \tau_2 s)/(1 + \tau_1 s)$

From eq. (4), the closed loop gain is

$$\begin{aligned} H(s) &= \frac{N_{FB}K}{s/F(s) + K} = \frac{N_{FB}K(1 + \tau_2 s)}{s(1 + \tau_1 s) + K(1 + \tau_2 s)} \\ &= \frac{N_{FB}K(1 + \tau_2 s)/\tau_1}{s^2 + s \frac{1 + K\tau_2}{\tau_1} + \frac{K}{\tau_1}}. \end{aligned}$$

We can get the denominator into the classical second-order control system form of $s^2 + 2\xi\omega_n s + \omega_n^2$ if we let $\omega_n^2 = K/\tau_1$ and $2\xi\omega_n = (1 + K\tau_2)/\tau_1$. Then,

$$H(s) = N_{FB}\omega_n^2 \frac{1 + s(2\xi\omega_n - 1/K)}{s^2 + 2\xi\omega_n s + \omega_n^2} = N_{FB} \frac{1 + s(2\xi\omega_n - 1/K)}{\left(\frac{s}{\omega_n}\right)^2 + 2\xi \frac{s}{\omega_n} + 1},$$

where the undamped natural frequency is

$$\omega_n = \sqrt{K/\tau_1},$$

and the damping is

$$\xi = \frac{1}{2\omega_n} \left(\frac{1}{\tau_1} + \frac{\tau_2 K}{\tau_1} \right) = \frac{\omega_n}{2} (\tau_2 + 1/K).$$

Thus, the closed loop gain magnitude is

$$|H(j\omega)| = N_{FB} \left\{ \frac{1 + (2\xi\omega/\omega_n - \omega/K)^2}{\left[1 - \left(\frac{\omega}{\omega_n} \right)^2 \right]^2 + 4\xi^2 \left(\frac{\omega}{\omega_n} \right)^2} \right\}^{1/2},$$

and the closed loop gain phase is

$$\phi(j\omega) = \arctan \left(2\xi \frac{\omega}{\omega_n} - \frac{\omega}{K} \right) - \arctan \frac{2\xi \frac{\omega}{\omega_n}}{1 - \left(\frac{\omega}{\omega_n} \right)^2}.$$

From eq. (3), the open loop gain is

$$\beta(s)G(s) = \frac{K(1 + \tau_2 s)}{s(1 + \tau_1 s)},$$

the open loop gain magnitude is

$$|\beta(j\omega)G(j\omega)| = \frac{K}{\omega} \sqrt{\frac{1 + (\omega\tau_2)^2}{1 + (\omega\tau_1)^2}},$$

and the open loop gain phase is

$$\theta(j\omega) = -90 + \arctan(\omega\tau_2) - \arctan(\omega\tau_1).$$

Note that the phase starts off at -90 degrees, approaches -135 degrees midway between $1/\tau_1$ and $1/\tau_2$, and then approaches -90 degrees again for high frequencies.

The lag-head loop filter results in a second-order type-1 PLL. (For large gain K , eq. (7) reduces to eq. (8), i.e., the lag-lead loop filter is an approximation of the second-order active loop filter.)

C.4 Case 4. Second-order active loop filter: $F(s) = (1 + \tau_2 s)/(\tau_1 s)$.

If it is desired to account for the operational amplitude pole, the designer may lump this pole into the VCO parasitic pole. Try to keep the operational amplitude bandwidth much larger than the PLL bandwidth (defined in Section 4.2).

From eq. (4), the closed loop gain is

$$H(s) = \frac{N_{FB}K}{s/F(s) + K} = \frac{N_{FB}K(1 + \tau_2 s)}{s^2\tau_1 + K(1 + \tau_2 s)} = \frac{N_{FB}K(1 + \tau_2 s)/\tau_1}{s^2 + sK\tau_2/\tau_1 + K/\tau_1}.$$

We can get the denominator into the classical second-order control system form of $s^2 + 2\xi\omega_n s + \omega_n^2$ if we let $\omega_n^2 = K/\tau_1$ and $2\xi\omega_n = K\tau_2/\tau_1$. Then,

$$H(s) = \frac{N_{FB}\omega_n^2 \left(1 + \frac{2\xi}{\omega_n} s\right)}{s^2 + 2\xi\omega_n s + \omega_n^2} = N_{FB} \frac{1 + s(2\xi/\omega_n)}{\left(\frac{s}{\omega_n}\right)^2 + 2\xi \frac{s}{\omega_n} + 1},$$

where the undamped natural frequency is

$$\omega_n = \sqrt{K/\tau_1},$$

and the damping is

$$\xi = \frac{1}{2\omega_n} \frac{K\tau_2}{\tau_1} = \frac{\omega_n\tau_2}{2}.$$

Thus, the closed loop gain magnitude is

$$|H(j\omega)| = N_{FB} \left\{ \frac{1 + 4\zeta^2 \left(\frac{\omega}{\omega_n} \right)^2}{\left[1 - \left(\frac{\omega}{\omega_n} \right)^2 \right]^2 + 4\zeta^2 \left(\frac{\omega}{\omega_n} \right)^2} \right\}^{1/2},$$

and the closed loop gain phase is

$$\phi(j\omega) = \arctan[2\zeta(\omega/\omega_n)] - \arctan \frac{2\zeta \frac{\omega}{\omega_n}}{1 - \left(\frac{\omega}{\omega_n} \right)^2}.$$

From eq. (3), the open loop gain is

$$\beta(s)G(s) = \frac{K(1 + \tau_2 s)}{\tau_1 s^2},$$

the open loop gain magnitude is

$$|\beta(j\omega)G(j\omega)| = \frac{K}{\tau_1 \omega^2} \sqrt{1 + (\omega \tau_2)^2},$$

and the open loop gain phase is

$$\theta(j\omega) = -180 + \arctan(\omega \tau_2).$$

Note that the phase starts off at -180 degrees and approaches -90 degrees at high frequencies. Thus, the second-order PLL exhibits a large phase margin, even with parasitic (VCO) poles. Ensure $\tau_1 < \tau_2$ for good stability.

This loop filter results in a second-order type-2 PLL.

C.5 Case 5. Third-order active loop filter: $F(s) = (1 + \tau_2 s)/[\tau_1 s(1 + \tau_3 s)]$

From eq. (4), the closed loop gain is

$$\begin{aligned} H(s) &= \frac{N_{FB} K}{s/F(s) + K} = \frac{N_{FB} K(1 + \tau_2 s)}{s^2 \tau_1 (1 + \tau_3 s) + K(1 + \tau_2 s)} \\ &= \frac{N_{FB} K(1 + \tau_2 s)}{s^3 \tau_1 \tau_3 + s^2 \tau_1 + s K \tau_2 + K}. \end{aligned}$$

Thus, the closed loop gain magnitude is

$$|H(j\omega)| = N_{FB} K \left\{ \frac{1 + (\omega \tau_2)^2}{[K - \omega^2 \tau_1]^2 + [\omega K \tau_2 - \omega^3 \tau_1 \tau_3]^2} \right\}^{1/2},$$

and the closed loop gain phase is

$$\phi(j\omega) = \arctan(\omega\tau_2) - \arctan \frac{(\omega K\tau_2 - \omega^3\tau_1\tau_3)}{(K - \omega^2\tau_1)}.$$

From eq. (3), the open loop gain is

$$\beta(s)G(s) = \frac{K(1 + \tau_2 s)}{\tau_1 s^2(1 + \tau_3 s)},$$

the open loop gain magnitude is

$$|\beta(j\omega)G(j\omega)| = \frac{K}{\tau_1 \omega^2} \left\{ \frac{1 + (\omega\tau_2)^2}{1 + (\omega\tau_3)^2} \right\}^{1/2},$$

and the open loop gain phase is

$$\theta(j\omega) = -180 + \arctan(\omega\tau_2) - \arctan(\omega\tau_3).$$

Note that the phase starts off at -180 degrees, approaches -135 degrees midway between τ_2 and τ_3 , and then approaches -180 degrees again for high frequencies.

For the third-order active filter case, it is possible to define an equivalent damping and natural frequency if we let $1/\tau_3$ be much higher than $1/\tau_2$. Rather than making this approximation, PLACE solves directly for the phase margin using an algorithm (see Ref. 18) that optimizes phase noise performance. The point of minimum phase shift (i.e., the inflection point of the open loop gain phase) is placed at exactly the frequency ω_u for unity open loop gain. (The user can specify ω_u , or else PLACE defaults the value to $f_u = f_{ref}/50$.) This will occur when the loop filter time constants obey the following relations:

$$\tau_3 = \frac{\sec(pm) - \tan(pm)}{\omega_u}$$

$$\tau_2 = \frac{\tau_3}{\omega_u^2}$$

$$\tau_1 = \frac{K}{\omega_u^2} \left\{ \frac{1 + (\omega_u\tau_2)^2}{1 + (\omega_u\tau_3)^2} \right\}^{1/2}.$$

PLACE asks the user for the desired phase margin and computes the above values for τ_1 , τ_2 , and τ_3 . Later the user has the opportunity to enter his own values for these variables.

AUTHOR

Jeffrey H. Saunders, B.S.E.E., 1979, The University of Massachusetts at Amherst; M. Engineering (Electrical), 1980, Cornell University; AT&T Bell

Laboratories, 1979—. Mr. Saunders' earliest involvement at AT&T Bell Laboratories was with the integrated circuit design of CMOS microprocessors. He is currently involved in the design of software-controlled digital channel banks and modems. Mr. Saunders' major responsibilities have been system synchronization, carrier recovery, and jitter characterization. Member, Eta Kappa Nu, Tau Beta Pi, Phi Kappa Phi, Phi Eta Sigma, IEEE.

PAPERS BY AT&T BELL LABORATORIES AUTHORS

COMPUTING/MATHEMATICS

- Bentley J., **A Little Program, A Lot of Fun.** Comm ACM 27(12):1179–1182, Dec 1984.
- Billera L. J., Munson B. S., **Triangulations of Oriented Matroids and Convex Polytopes.** SIAM J Alg 5(4):515–525, Dec 1984.
- Calderbank A. R., Goethals J. M., **Three-Weight Codes and Association Schemes.** Phil J Res 39(4–5):143–152, 1984.
- Chung F. R. K., Garey M. R., **Diameter Bounds for Altered Graphs.** J Graph Th 8(4):511–534, Win 1984.
- Cleveland W. S., **Graphical Methods for Data Presentation—Full Scale Breaks, Dot Charts, and Multibased Logging.** Am Statistn 38(4):270–280, Nov 1984.
- Cleveland W. S., **Graphs in Scientific Publications.** Am Statistn 38(4):261–269, Nov 1984.
- Cleveland W. S., McGill R., **The Many Faces of a Scatterplot.** J Am Stat A 79(388):807–822, Dec 1984.
- Coffman E. G., Gilbert E. N., **Dynamic, First-Fit Packings in Two or More Dimensions.** Inf Contr 61(1):1–14, Apr 1984.
- Doyle J. K. et al., **A Portable PDP-11 Simulator.** Software 14(11):1047–1059, Nov 1984.
- Fishburn P. C., **Multiattribute Nonlinear Utility Theory.** Manag Sci 30(11):1301–1310, Nov 1984.
- Flatto L., Hahn S., **Two Parallel Queues Created by Arrivals With Two Demands I.** SIAM J A Ma 44(5):1041–1053, Oct 1984.
- Haley A., Zweber S., **Development and Application of a White Box Approach to Integration Testing.** J Syst Soft 4(4):309–315, Nov 1984.
- Hecht M. S., Gabbe J. D., **Shadowed Management of Free Disk Pages With a Linked List.** ACM T Datab 8(4): 503–514, Dec 1983.
- Iannino A. et al., **Criteria for Software-Reliability Model Comparisons.** IEEE Soft E 10(6):687–691, Nov 1984.
- Jerome J. W., **Fully Discrete Stability and Invariant Rectangular Regions for Reaction-Diffusion Systems.** SIAM J Num 21(6):1054–1065, Dec 1984.
- Kackar R. N., Harville D. A., **Approximations for Standard Errors of Estimators of Fixed and Random Effects in Mixed Linear Models.** J Am Stat A 79(388):853–862, Dec 1984.
- Kaufman L., **Banded Eigenvalue Solvers on Vector Machines.** ACM T Math 10(1):73–86, Mar 1984.
- Landau H. J., Logan B. F., Shepp L. A., Bauman N., **Diffusion, Cell Mobility, and Bandlimited Functions.** SIAM J A Ma 44(6):1232–1245, Dec 1984.
- Lloyd S. P., **Ordered Prime Divisors of a Random Integer.** Ann Probab 12(4):1205–1212, Nov 1984.
- Lohse J. B., Zweber S. H., **Experimental Evaluation of Software-Design Principles—An Investigation Into the Effect of Module Coupling on System Modifiability.** J Syst Soft 4(4):301–308, Nov 1984.
- Lucky R. W., **The Social Impact of the Computer.** Ann NY Acad 426(Nov):1–10, Nov 1 1984.
- Mehravari N., Berger T., **Poisson Multiple-Access Contention With Binary Feedback.** IEEE Info T 30(5):745–751, Sep 1984.
- Musa J. D., Okumoto K., **A Comparison of Time Domains for Software-Reliability Models.** J Syst Soft 4(4):277–287, Nov 1984.
- Nair V. N., **On the Behavior of Some Estimators From Probability Plots.** J Am Stat A 79(388):823–831, Dec 1984.
- Rosenberg E., **Exact Penalty Functions and Stability in Locally Lipschitz Programming.** Math Progr 30(3):340–356, Dec 1984.
- Udwadia F. E., Sharma D. K., **On the Identification of Continuous Vibrating**

Systems Modeled by Hyperbolic Partial-Differential Equations. Q Appl Math 42(4):411-424, Jan 1985.

Whitt W., **Approximations for Departure Processes and Queues in Series.** Nav Res Log 31(4):499-521, Dec 1984.

Whitt W., **Departures From a Queue With Many Busy Servers.** Math Oper R 9(4):534-544, Nov 1984.

Whitted T., **Computer Image Synthesis—Rendering Techniques.** Ann NY Acad 426(Nov):62-75, Nov 1 1984.

ENGINEERING

Acampora A. S., Hluchyj M. G., Tsao C. D., **A Centralized Bus Architecture for Local Area Networks.** J Telecomm 3(2):89-102, Sum 1984.

Agrawal G. P., **Fast-Fourier-Transform-Based Beam-Propagation Model for Stripe-Geometry Semiconductor-Lasers—Inclusion of Axial Effects.** J Appl Phys 56(11):3100-3109, Dec 1 1984.

Agrawal G. P., **Generalized Rate Equations and Modulation Characteristics of External-Cavity Semiconductor Lasers.** J Appl Phys 56(11):3110-3115, Dec 1 1984.

Bridges T. J., Strnad A. R., Wood O. R., Patel C. K. N., Karlin D. B., **Interaction of Carbon-Dioxide Laser Radiation With Ocular Tissue.** IEEE J Q El 20(12):1449-1458, Dec 1984.

Caine E. J., Subbanna S., Kroemer H., Merz J. L., Cho A. Y., **Staggered-Lineup Heterojunctions as Sources of Tunable Below-Gap Radiation—Experimental Verification.** Appl Phys L 45(10):1123-1125, Nov 15 1984.

Chen C. Y., Chi G. C., **Low-Noise Ga_{0.47}In_{0.53} As Photoconductive Detectors Using Fe Compensation.** Appl Phys L 45(10):1083-1085, Nov 15 1984.

Chraplyvy A. R., Stone J., **Measurement of Crossphase Modulation in Coherent Wavelength-Division Multiplexing Using Injection Lasers.** Electr Lett 20(24):996-997, Nov 22 1984.

Chu S. N. G., Sheng T. T., **TEM Cross-Section Sample Preparation Technique for III-V Compound Semiconductor-Device Materials by Chemical Thinning.** J Elchem So 131(11):2663-2667, Nov 1984.

Cohen R. L., West K. W., **Spot Weld Strength Determined From Simple Electrical Measurements.** Welding J 63(12):17-23, Dec 1984.

Cook C. F., Huggins P. A., **Effect of Radio-Frequency Interference on Common Industrial-Hygiene Monitoring Instruments.** Am Ind Hyg 45(11):740-744, Nov 1984.

Cook J. M., Hannon J. J., **Gaseous Cleaning Beneath Surface Mounted Components—Evaluation Using a Beam Lead Test Chip.** IEEE Compon 7(4):328-335, Dec 1984.

Courcoubetis C., Varaiya P. P., **Serving Process With Least Thinking Time Maximizes Resource Utilization.** IEEE Auto C 29(11):1005-1008, Nov 1984.

Downey P. M., Tell B., **Picosecond Photoconductivity Studies of Light-Ion-Bombarded InP.** J Appl Phys 56(10):2672-2674, Nov 15 1984.

Eisenstein G., Koren U., Tucker R. S., Kasper B. L., Gnauck A. H., Tien P. K., **High-Speed Analog and Digital Modulation of 1.51-μm Wavelength, Three-Channel Buried Crescent InGaAsP Lasers.** Appl Phys L 45(4):311-313, Aug 15 1984.

Feuer M. D., **Two-Layer Model for Source Resistance in Selectively Doped Heterojunction Transistors.** IEEE Device 32(1):7-11, Jan 1985.

Forrest S. R., **Photoconductor Receiver Sensitivity.** IEEE Elec D 5(12):536-539, Dec 1984.

Graham R. L., Winkler P. M., **Isometric Embeddings of Graphs.** P Nas Phys 81(22):7259-7260, Nov 1984.

Hall P. M., **Forces, Moments, and Displacements During Thermal Chamber Cycling of Leadless Ceramic Chip Carriers Soldered to Printed Boards.** IEEE Compon 7(4):314-327, Dec 1984.

Hegde S. S., Prewitt J. M. S., **A Layered Approach to PACS Network Architecture.** P Soc Photo 515:442-451, 1984.

- Huang A., **Impact of New Technological Advances and Architectural Insights on the Design of Optical Computers.** P Soc Photo 456:25-29, 1984.
- Huggins R. G. et al., **Nonionizing Radiation Aspects of Optical Fiber Manufacturing.** Am Ind Hyg 45(12):796-801, Dec 1984.
- Jackel L. D., Swartz R. G., Howard R. E., Ko P. K., Grabbe P., **CASFET—A MOSFET-JFET Cascode Device With Ultralow Gate Capacitance.** IEEE Device 31(12):1752-1758, Dec 1984.
- Johannes V. I., **Improving on Bit Error Rate.** IEEE Comm M 22(12):18-20, Dec 1984.
- Johnson R., **Network Reliability and Acyclic Orientations.** Networks 14(4):489-505, Win 1984.
- Joyce W. B., Deloach B. C., **Alignment of Gaussian Beams.** Appl Optics 23(23):4187-4196, Dec 1 1984.
- Knausenberger W. H., Schaper L. W., **Interconnection Costs of Various Substrates—The Myth of Cheap Wire.** IEEE Compon 7(3):261-263, Sep 1984.
- Knox W. H., Downer M. C., Fork R. L., Shank C. V., **Amplified Femtosecond Optical Pulses and Continuum Generation at 5-kHz Repetition Rate.** Optics Lett 9(12):552-554, Dec 1984.
- Koch T. L., Bowers J. E., **Nature of Wavelength Chirping in Directly Modulated Semiconductor Lasers.** Electr Lett 20(25-2):1038-1040, Dec 6 1984.
- Koch T. L., Coldren L. A., Bridges T. J., Burkhardt E. G., Corvini P. J., Wilt D. P., **1.5- μ m Monolithic Shallow-Groove Coupled-Cavity Vapor-Phase Transported Buried Heterostructure Lasers.** Electr Lett 20(24):1001-1002, Nov 22 1984.
- Kyung C. M., **Temperature Profile of a Silicon-on-Insulator Multilayer Structure in Silicon Recrystallization With Incoherent-Light Source.** IEEE Device 31(12):1845-1851, Dec 1984.
- Lee T. P., Burrus C. A., Liou K. Y., Olsson N. A., Logan R. A., Wilt D. P., **Measured Spectral Linewidth of Single-Frequency 1.3- μ m and 1.5- μ m Injection Lasers.** Electr Lett 20(24):1011-1012, Nov 22 1984.
- Louv W. C., **Adaptive Filtering.** Technomet 26(4):399-409, Nov 1984.
- Manchanda L., **Inversion Layer Mobility of MOSFETs Fabricated With NMOS Submicrometer Technology.** IEEE Elec D 5(11):470-472, Nov 1984.
- McDonald K. L., **Humidity Control in Accelerated Life Tests.** IEEE Compon 7(3):268-275, Sep 1984.
- Mehravari N., **TDMA in a Random-Access Environment—An Overview.** IEEE Comm M 22(11):54-59, Nov 1984.
- Miller D. A. B., Chemla D. S., Damen T. C., Wood T. H., Burrus C. A., Gossard A. C., Wiegmann W., **Optical-Level Shifter and Self-Linearized Optical Modulator Using a Quantum-Well Self-Electro-Optic Effect Device.** Optics Lett 9(12):567-569, Dec 1984.
- Prewitt J. M. S., Selfridge P. G., Anderson A. C., **Name-Value Pair Specification for Image Data Headers and Logical Standards for Image Data Exchange.** P Soc Photo 515:452-458, 1984.
- Ross I. M., **Technology for Tomorrow.** J Telecomm 3(2): 63-68, Sum 1984.
- Sandberg I. W., **Citation Classic—Some Results on the Theory of Physical Systems Governed by Nonlinear Functional Equations.** CC/Eng Tech (49):16, Dec 3 1984.
- Schiavone L. M., Bosch M. A., Haller H. R., Hubbard W. E., Good E., Shay J. L., **Optical Recording in Amorphous Hydrogenated Tellurium.** Appl Optics 23(22):3954-3955, Nov 15 1984.
- Suh S. Y., Anderson D. L., **Latent-Heat Effects of Pulsed Laser-Beam Induced Temperature Profiles in Optical-Recording Thin Films.** Appl Optics 23(22):3965-3971, Nov 15 1984.
- Tewksbury S. K., Biazzo M. R., Lindstrom T. L., **Strong Carrier Freezeout Above 77K in Tellurium-Doped Buried-Channel MOS Transistors.** IEEE Device 32(1):67-69, Jan 1985.
- Thomas D. G., **Optical Communications.** Res Dev 26(6):198-204, Jun 1984.
- Vodhanel R. S., Ko J. S., **Reflection Induced Frequency Shifts in Single-Mode Laser Diodes Coupled to Optical Fibers.** Electr Lett 20(23):973-974, Nov 8 1984.

Yan M. F., Rhodes W. W., Efficient Process for Volume Production of TiO₂ Varistor Powder. Am Ceram S 63(12):1484+, Dec 1984.

PHYSICAL SCIENCES

- Aeppli G., Bruinsma R., Hexatic Order and Liquid Density Fluctuations. Phys Rev L 53(22):2133-2136, Nov 26 1984.
- Batlogg B., Superconductivity in Ba (Pb, Bi)O₃. Physica B&C 126(1-3):275-279, Nov 1984.
- Baumgart H., Philipp F., Celler G. K., Defect Structure of Epitaxial Films Grown on Porous Silicon. Inst Phys C (67):223-228, 1983.
- Bean J. C., The Application of Silicon Molecular-Beam Epitaxy to VLSI. AIP Conf Pr (122):198-204, 1984.
- Besomi P., Degani J., Dutta N. K., Wagner W. R., Nelson R. J., High-Quality Indium Gallium-Arsenide Phosphide Double Heterostructure Material Grown by the Near Equilibrium Liquid-Phase-Epitaxy Technique. J Appl Phys 56(10):2879-2882, Nov 15 1984.
- Bhatt R. N. et al., Insulating Phase of a Disordered System—Fermi Glass Versus Electron Glass (Letter). J Phys C 17(24):L639-L643, Aug 30 1984.
- Bloomfield L. A., Freeman R. R., Cooke W. E., Bokor J., Angular-Momentum Dependence of Autoionization Rates in Doubly Excited Rydberg States of Ba. Phys Rev L 53(23):2234-2237, Dec 3 1984.
- Bondybey V. E., Laser-Induced Fluorescence and Bonding of Metal Dimers. Science 227(4683):125-131, Jan 11 1985.
- Bondybey V. E., Haddon R. C., Rentzepis P. M., Spectroscopy and Dynamics of 9-Hydroxyphenalenone and of Its 5-Methyl Derivative in Solid Neon: Effect of Methyl Group Upon Vibrational Relaxation. J Am Chem S 106(20):5969-5973, Oct 3 1984.
- Brasen D., Vaidya S., Kammlott G. W., Chin B. H., Electrothermomigration and Filament Growth in P-InP With Au Contacts. J Appl Phys 56(12):3399-3403, Dec 15 1984.
- Brown B. L., Leventhal M., Mills A. P., Gidley D. W., Positron Annihilation in a Simulated Low-Density Galactic Environment. Phys Rev L 53(24):2347-2350, Dec 10 1984.
- Budai J., Pindak R., Davey S. C., Goodby J. W., A Structural Investigation of the Liquid-Crystal Phases of 4-(2'-Methylbutyl)Phenyl 4'-Normal-Octylbiphenyl-4-Carboxylate. J Phys Lett 45(21):1053-1062, Nov 1 1984.
- Cais R. E., Kometani J. M., Polymerization of Vinylidene-D₂ Fluoride. Minimal Regiosequence and Branch Defects and Assignment of Preferred Chain-Growth Direction From the Deuterium Isotope Effect (Letter). Macromolec 17(9):1887-1889, Sep 1984.
- Campisano S. U., Jacobson D. C., Poate J. M., Cullis A. G., Chew N. G., Impurity and Interfacial Effects on the Formation of Amorphous Si From the Melt. Appl Phys L 45(11):1216-1218, Dec 1 1984.
- Capasso F., Cox H. M., Hutchinson A. L., Olsson N. A., Hummel S. G., Pseudo-Quaternary GaInAsP Semiconductors—A New Ga_{0.47}In_{0.53}As/InP Graded GaP Superlattice and Its Applications to Avalanche Photodiodes. Appl Phys L 45(11):1193-1195, Dec 1 1984.
- Carroll P. J., Patterson G. D., Rayleigh-Brillouin Spectroscopy of Simple Viscoelastic Liquids. J Chem Phys 81(4):1666-1675, Aug 15 1984.
- Cerdeira F., Pinczuk A., Bean J. C., Batlogg B., Wilson B. A., Raman Scattering From Ge_xSi_{1-x}/Si Strained-Layer Superlattices. Appl Phys L 45(10):1138-1140, Nov 15 1984.
- Chance B. et al., X-Ray Absorption Studies of Intermediates in Peroxidase Activity. Arch Bioch 235(2):596-611, Dec 1984.
- Cheng C. L., Liao A. S. H., Chang T. Y., Caridi E. A., Coldren L. A., Lalevic B., Silicon-Oxide Enhanced Schottky Gate In_{0.53}Ga_{0.47}As FETs With a Self-Aligned Recessed Gate Structure. IEEE Elec D 5(12):511-514, Dec 1984.
- Chi G. C., Nakahara S., The Microstructure of Thin Aluminum Copper (4.5 Percent) Films Deposited by Sputtering Techniques. Mater Lett 2(5A):380-385, Jun 1984.

- Cholli A. L., Dumais J. J., Engel A. K., Jelinski L. W., **Aromatic Ring Flips in a Semicrystalline Polymer.** *Macromolec* 17(11):2399-2404, Nov 1984.
- Cliff G., Maher D. M., Joy D. C., **Determination of UTW K_xSi Factors for Seven Elements from Oxygen to Iron.** *J Microsc O* 136(Nov):219-225, Nov 1984.
- Cohen U., Walton K. R., Sard R., **Development of Silver-Palladium Alloy-Plating for Electrical Contact Applications.** *J Elchem So* 131(11):2489-2495, Nov 1984.
- Dautremont-Smith W. C., Barnes P. A., Stayt J. W., **A Nonalloyed, Low Specific Resistance Ohmic Contact to N-InP.** *J Vac Sci B* 2(4):620-625, Oct-Dec 1984.
- Davey S. C., Budai J., Goodby J. W., Pindak R., Moncton D. E., **X-Ray Study of the Hexatic-B to Smectic-A Phase Transition in Liquid-Crystal Films.** *Phys Rev L* 53(22):2129-2132, Nov 26 1984.
- Dupuis R. D., Miller R. C., Petroff P. M., **Growth and Characterization of High-Quality MOCVD AlGaAs/GaAs Single Quantum Wells.** *J Cryst Gr* 68(1):398-405, Sep 1984.
- Dutta N. K., Olsson N. A., Shen T. M., **Temperature Behavior of Optical Absorption in InGaAsP Lasers.** *Appl Phys L* 45(10):1023-1025, Nov 15 1984.
- Dymek C. J., Williams J. L., Groeger D. J., Auborn J. J., **An Aluminum Acid-Base Concentration Cell Using Room-Temperature Chloroaluminate Ionic Liquids.** *J Elchem So* 131(12):2887-2892, Dec 1984.
- Dynes R. C., Garno J. P., Hertel G. B., Orlando T. P., **Tunneling Study of Superconductivity Near the Metal-Insulator Transition.** *Phys Rev L* 53(25):2437-2440, Dec 17 1984.
- Erman M., Theeten J. B., Chambon P., Kelso S. M., Aspnes D. E., **Optical-Properties and Damage Analysis of GaAs Single Crystals Partly Amorphized by Ion Implantation.** *J Appl Phys* 56(10):2664-2671, Nov 15 1984.
- Fisher D. S., **Sliding Charge-Density Waves As a Dynamic Critical Phenomenon.** *Physica B&C* 126(1-3):409-413, Nov 1984.
- Fleming J. W., **Dispersion in GeO_2-SiO_2 Glasses.** *Appl Optics* 23(24):4486-4493, Dec 15 1984.
- Fletcher M. A., Young M. S. S., Geyling F. T., Nakahara S., Parsey J. M., **Bismuth Precipitation in Monocrystalline InBi.** *Metall T-A* 15(11):1963-1967, Nov 1984.
- Forrest S. R., Schmidt P. H., Wilson R. B., Kaplan M. L., **Relationship Between the Conduction-Band Discontinuities and Band-Gap Differences of InGaAsP/InP Heterojunctions.** *Appl Phys L* 45(11):1199-1201, Dec 1 1984.
- Gordon J. P., **The Motion of Neutral Atoms in a Radiative Trap.** *Proc Q Elec* 8(3-4):177-180, 1984.
- Gossard A. C., **Quantum Well Structures and Superlattices and Their Applications Potential.** *Inst Phys C* (69):1-14, 1984.
- Gossmann H. J., Feldman L. C., Gibson W. M., **Ge Deposition on Si(111)-7 × 7 and Si(100)-2 × 1—Effects on Si Surface Structure.** *J Vac Sci B* 2(3):407-408, Jul-Sep 1984.
- Graedel T. E., **The Perceived Intensity of Natural Odors.** *J Chem Educ* 61(8):681-686, Aug 1984.
- Griffiths J. E., Malyj M., Espinosa G. P., Remeika J. P., **Crystalline $SiSe_2$ and Si_xSe_{1-x} Glasses—Syntheses, Glass Formation, Structure, Phase Separation, and Raman Spectra.** *Phys Rev B* 30(12):6978-6990, Dec 15 1984.
- Gurvitch M., **Target Erosion and Deposition Rates in Planar Magnetron Sputtering.** *J Vac Sci A* 2(4):1550-1551, Oct-Dec 1984.
- Hauser J. J., Waszczak J. V., **Spin-Glass Transition in MnO.** *Phys Rev B* 30(9):5167-5171, Nov 1 1984.
- Hilinski E. F., Straub K. D., Rentzepis P. M., **Solvent Effects on the Relaxation Mechanism of Cu(II) Protoporphyrin.** *Chem P Lett* 111(4-5):333-339, Nov 9 1984.
- Ibbotson D. E., Mucha J. A., Flamm D. L., Cook J. M., **Plasmaless Dry Etching of Silicon With Fluorine-Containing Compounds.** *J Appl Phys* 56(10):2939-2942, Nov 15 1984.
- Ihringer J., Abrahams S. C., **Soft Modes and Elastic Strain at the Tetragonal-to-Monoclinic Phase-Transition in Antifluorite and Related Structure Types.** *Phys Rev B* 30(11):6540-6548, Dec 1 1984.
- Imhof W. L. et al., **A Coordinated Satellite and Ground-Based Study of an**

- Intense Electron-Precipitation Spike Over the Southern Polar Cap.** J Geo R-S P 89(NA12):837-846, Dec 1 1984.
- Inoue A., Chen H. S., Krause J. T., Masumoto T., **Young Modulus Sound-Velocity and Young Modulus of Ti-Based, Zr-Based and Hf-Based Amorphous Alloys.** J Non-Cryst 68(1):63-73, Oct 1984.
- Iye Y. et al., **Effect of Impurities on the Electronic Phase Transition in Graphite in the Magnetic Quantum Limit.** Phys Rev B 30(12):7009-7015, Dec 15 1984.
- Iye Y., Berglund P. M., McNeil L. E., **The Magnetic-Field Dependence of the Critical-Temperature for the Electronic Phase Transition in Graphite in the Quantum Limit.** Sol St Comm 52(12):975-980, Dec 1984.
- Jacobs L., Jose J. V., Novotny M. A., **First-Order Reentrant Transition in Granular Superconducting Films.** Phys Rev L 53(22):2177-2180, Nov 26 1984.
- Jayaraman A., Swaminathan V., Batlogg B., **Pressure and Volume Dependence of the LO-TO Phonons in InAs.** Pramana 23(3):405-410, Sep 1984.
- Joy D. C., **Beam Interactions, Contrast and Resolution in the SEM.** J Microsc O 136(Nov):241-258, Nov 1984.
- Joy D. C., **High-Resolution Lithography and the Role of Secondary Electrons.** Inst Phys C (67):445-450, 1983.
- Karlicek R., Long J. A., Donnelly V. M., **Thermal-Decomposition of Metalorganic Compounds Used in the MOCVD of InP.** J Cryst Gr 68(1):123-127, Sep 1984.
- Katz H. E., Starnes W. H., **Utility of Lewis-Bases in Alkytin Trithiolate Stabilizer Systems for Polyvinyl-Chloride.** Macromolec 17(11):2241-2243, Nov 1984.
- Kauzlarich S. M., Averill B. A., Teo B. K., **Structural Studies of FeOC₁ Intercalated With Tetrathiafulvalene and Related Materials.** Molec Cryst 107(1-2): 65-73, 1984.
- Kleiman R. N., Bishop D. J., Pindak R., Taborek P., **Shear Modulus and Specific Heat of the Liquid-Crystal Blue Phases.** Phys Rev L 53(22):2137-2140, Nov 26 1984.
- Kramer L., Hohenberg P. C., **Effects of Boundary Conditions on Spatially Periodic States.** Physica D, 13(3):357-369, Nov 1984.
- Kuk Y., Feldman L. C., **Oscillatory Relaxation of the Ag(110) Surface.** Phys Rev B 30(10):5811-5816, Nov 15 1984.
- Levine B. F., **Near Room-Temperature 1.3-μm Single Photon Counting With an InGaAs Avalanche Photodiode—Reply (Letter).** Electr Lett 20(23):962, Nov 8 1984.
- Lundberg H., Macklin J. J., Silfvast W. T., Wood O. R., **High-Gain Soft-X-Ray-Pumped Photoionization Laser in Zinc Vapor.** Appl Phys L 45(4):335-337, Aug 15 1984.
- Mahajan S., **The Sources of Defects in InP InGaAsP Emitters.** Inst Phys C (67):259-272, 1983.
- Mahajan S., Dutt B. V., Temkin H., Cava R. J., Bonner W. A., **Spinodal Decomposition in InGaAsP Epitaxial Layers.** J Cryst Gr 68(2):589-595, Sep 1984.
- Manzione L. T., **Predictive Parameters in Reactive Processing of Polyurethane.** Polym Prepr 25(2):289-290, Aug 1984.
- Marcus M. A., **The Blue Phase of Cholesteryl Propionate.** Molec Cryst 102 (6-7):207-210, 1984.
- Marcus M., Tsai C.-L., **EXAFS Measurements of Lattice Vibrations of Ti and Cu in Ti.** Sol St Comm 52(5):511-513, Nov 1984.
- Miller D. A. B., **Optical Bistability and Differential Gain Resulting From Absorption Increasing With Excitation.** J Opt Soc B 1(6):857-864, Dec 1984.
- Miller D. A. B., Chemla D. S., Damen T. C., Gossard A. C., Wiegmann W., Wood T. H., Burrus C. A., **Band-Edge Electroabsorption in Quantum Well Structures—The Quantum-Confinement Stark Effect.** Phys Rev L 53(22):2173-2176, Nov 26 1984.
- Mills A. P., Crane W. S., **Emission of Band-Gap-Energy Positrons From Surfaces of LiF, NaF, and Other Ionic Crystals.** Phys Rev L 53(22):2165-2168, Nov 26 1984.
- Miraglia S., Hodeau J. L., Marezio M., Ott H. R., Remeika J. P., **Relationship Between the Crystal Structure and the Reentrant Superconducting Properties of (Sn_{1-x}Er_x)Er₄Rh₆Sn₁₈.** Sol St Comm 52(2):135-137, Oct 1984.
- Mitchell J. W., **Selective Substoichiometric Separation of Fluoride by Solvent-**

- Extraction Into Chloroform With Triphenyl-Tin Chloride.** Mikroch Act 2(3-4):243-250, 1984.
- Murarka S. P., **Interactions in Metallization Systems for Integrated-Circuits.** J Vac Sci B 2(4):693-706, Oct-Dec 1984.
- Murarka S. P., Vaidya S., **Cospattered Cobalt Silicides on Silicon, Polycrystalline Silicon, and Silicon Dioxide.** J Appl Phys 56(12):3404-3412, Dec 15 1984.
- Naqui A., Kumar C., Ching Y. C., Powers L., Chance B., **Structure and Reactivity of Multiple Forms of Cytochrome Oxidase as Evaluated by X-Ray Absorption Spectroscopy and Kinetics of Cyanide Binding.** Biochem 23(25):6222-6227, Dec 4 1984.
- Odagaki T., Lax M., Day R. S., **Dispersive Hopping Conduction in Quasi-One-Dimensional Systems.** Phys Rev B 30(12):6911-6916, Dec 15 1984.
- Ondrias M. R. et al., **A Resonance Raman Study of the Temperature Dependence of Ligand Photolysis and Recombination in Hemoglobins.** Chem P Lett 112(4):351-355, Dec 14 1984.
- Ota Y., Clapper R. A., **Vacuum Evaporation System for Depositing Thick Polycrystalline Silicon.** J Vac Sci B 2(3):320-326, Jul-Sep 1984.
- Patel J. R., Golovchenko J. A., **Determination of Atom Locations on Surfaces With X-Ray Standing Waves.** Inst Phys C (67):349-357, 1983.
- People R., Bean J. C., Lang D. V., Sergent A. M., Stormer H. L., Wecht K. W., Lynch R.T., Baldwin K., **Modulation Doping in Ge_xSi_{1-x}/Si Strained Layer Heterostructures.** Appl Phys L 45(11):1231-1233, Dec 1 1984.
- Platzman P. M., Tzoar N., **Impulse Corrections to the Impulse Approximation for High-Momentum-Transfer Neutron-Scattering.** Phys Rev B 30(11):6397-6401, Dec 1 1984.
- Raghavachari K. **Theoretical Study of Substituent Effects on CH Stretching Frequencies.** J Chem Phys 81(6):2717-2722, Sep 15 1984.
- Rasanen M., Bondybey V. E., **Gauche-N-Butane and Trans-N-Butane and Their Ir-Induced Interconversion in Solid Neon.** Chem P Lett 111(6):515-520, Nov 16 1984.
- Reitman E. A., **Electronic Properties of the Amylose-Iodine Complex.** J Mat Sci L 3(12):1043-1045, Dec 1984.
- Reynolds C. L., Swaminathan V., Tamargo M. C., **Photoluminescence of (Al, Ga)As Double Heterostructure Laser Material Containing a Buffer Layer.** Mater Lett 2(6A-B):512-514, Sep 1984.
- Rietschel R. L., Huggins R., Levy N., Pruitt P. M., **Invivo and Invitro Testing of Gloves for Protection Against UV-Curable Acrylate Resin Systems.** Contact Der 11(5):279-282, Nov 1984.
- Sadler D. M., Gilmer G. H., **A Model for Chain Folding in Polymer Crystals—Rough Growth Faces Are Consistent With the Observed Growth Rates.** Polymer 25(10):1446-1452, Oct 1984.
- Sammon M. J., **A Newly Observed Dislocation Network in Blue Phase-II of Cholesteric Liquid Crystals.** Molec Cryst 102(6-7):161-166, 1984.
- Sandberg I. W., **Citation Classic—Some Results on the Theory of Physical Systems Governed by Nonlinear Functional Equations.** CC/Phy Chem (49):16, Dec 3 1984.
- Schneemeyer L. F., Spengler S. E., Di Salvo F. J., Waszczak J. V., Rice C. E., **Electrochemical Crystal Growth in the Cesium Molybdate Molybdenum Trioxide System.** J Sol St Ch 55(2): 158-164, Nov 15 1984.
- Smith T. P., Phillips J. M., Augustyniak W. M., Stiles P. J., **Fabrication of Metal Epitaxial Insulator-Semiconductor Field-Effect Transistors Using Molecular-Beam Epitaxy of CaF_2 on Si.** Appl Phys L 45(8):907-909, Oct 15 1984.
- Sooryakumar R., Chemla D. S., Pinczuk A., Grossard A., Weigmann W., Sham L. J., **Direct Observation of Band Mixing in $GeAs-(Al_xGa_{1-x})As$ Quantum Heterostructures.** J Vac Sci B 2(3):349-350, Jul-Sep 1984.
- Starnes W. H. et al., **Poly(Vinyl-Chloride) Structural Segments Derived From Azobis(Isobutyronitrile).** Polym Prepr 25(2):75-76, Aug 1984.
- Stillinger F. H., Weber T. A., **Point Defects in BCC Crystals—Structures, Transition Kinetics, and Melting Implications.** J Chem Phys 81(11):5095-5103, Dec 1 1984.

- Stormer H. L., **The Fractional Quantum Hall Effect (Experiment).** Physica B&C 126(1-3):250-253, Nov 1984.
- Suh S. Y., Snyder D. A., Anderson D. L., Craighead H. G., **Dynamic Signal-to-Noise Measurements of Textured Optical Storage Media.** Appl Optics 23(22):3956-3964, Nov 15 1984.
- Temkin H., Dolan G. J., Olsson N. A., Henry C. H., Logan R. A., Kazarinov R. F., Johnson L. F., **1.55- μ m InGaAsP Ridge Wave-Guide Distributed Feedback Laser.** Appl Phys L 45(11):1178-1180, Dec 1 1984.
- Temkin H., Mahajan S., Logan R. A., **Optically Induced Catastrophic Degradation in InGaAsP Laser Structures.** Inst Phys C (67):279-284, 1983.
- Thomas G. A., **Asymmetry in the Metal—Ammonia Phase Diagram.** J Phys Chem 88(17):3749-3751, Aug 16 1984.
- Thomsen C., Strait J., Vardeny Z., Maris H. J., Tauc J., **Picosecond Optical Generation and Detection of Phonon Waves in A-As₂Te₃.** AIP Conf Pr (120):102-109, 1984.
- Thurston R. N., **Distribution of Torque in a Dual-Frequency Liquid Crystal.** Molec Cryst 108(1-2):61-70, 1984.
- Tsang W. T., **Chemical Beam Epitaxy of InP and GaAs.** Appl Phys L 45(11):1234-1236, Dec 1 1984.
- Tsang W. T., Logan R. A., **A New High-Power, Narrow-Beam Transverse-Mode Stabilized Semiconductor Laser at 1.5 μ m—The Heteroepitaxial Ridge Overgrown Laser.** Appl Phys L 45(10):1025-1027, Nov 15 1984.
- Tung R. T., **Schottky-Barrier Heights of Single-Crystal Silicides on Si(111).** J Vac Sci B 2(3):465-470, Jul-Sep 1984.
- Vandenberg J. M., Draper C. W., **An X-Ray-Diffraction Study on the Microstructure of Laser Surface Melted Cu-Al-Fe Alloys.** Mater Lett 2(5A):386-392, Jun 1984.
- Venkatesan T. et al., **Ion Channeling Studies of Regrowth Kinetics of Disordered Surface Layers on Graphite.** J Appl Phys 56(11):3232-3240, Dec 1 1984.
- Venkatesan T. et al., **Structural and Chemical Analysis of Ion-Beam Produced Conductive Regions on Highly Resistive Organic Films.** J Appl Phys 56(10):2778-2787, Nov 15 1984.
- Weber T. A., Stillinger F. H., **Inherent Structures and Distribution Functions for Liquids That Freeze Into BCC Crystals.** J Chem Phys 81(11):5089-5094, Dec 1 1984.
- Wilson B. A., Sergent A. M., Harbison J. P., **The Role of Dangling Bonds in Radiative and Nonradiative Processes in A-Si-H.** AIP Conf Pr (120):149-156, 1984.
- Yurke B., **Conservative Model for the Damped Harmonic Oscillator.** Am J Phys 52(12):1099-1102, Dec 1984.

SOCIAL AND LIFE SCIENCES

- Poller M. F., Garter S. K., **The Effects of Modes on Text Editing by Experienced Editor Users.** Human Fact 26(4):449-462, Aug 1984.
- Yager D., Kramer P., Shaw M., Graham N., **Detection and Identification of Spatial Frequency—Models and Data.** Vision Res 24(9):1021-1035, 1984.

SPEECH/ACOUSTICS

- Buschvishniac I. J., West J. E., Wallace R. L., **A New Approach to Transducer Design Applied to a Foil Electret Acoustic Antenna.** J Acoust So 76(6):1609-1616, Dec 1984.
- Samuel A. G., Kat D., Tartter V. C., **Which Syllable Does an Intervocalic Stop Belong to—A Selective Adaptation Study.** J Acoust So 76(6):1652-1663, Dec 1984.

CONTENTS, JULY-AUGUST 1985

Part 1

Digital Transmission Over Cross-Coupled Linear Channels

J. Salz

Tone Location by Cyclotomic Filters

D. Hertz, R. P. Kurshan, D. Malah, and J. T. Peoples

Spread Spectrum for Indoor Wireless Communications

M. Kavehrad

Recognition of Isolated Digits Using Hidden Markov Models With Continuous Mixture Densities

L. R. Rabiner, B.-H. Juang, S. E. Levinson, and M. M. Sondhi

Maximum-Likelihood Estimation for Mixture Multivariate Stochastic Observations of Markov Chains

B.-H. Juang

Some Properties of Continuous Hidden Markov Model Representations

L. R. Rabiner, B.-H. Juang, S. E. Levinson, and M. M. Sondhi

Variable-Length Packetization of μ -Law PCM Speech

R. Steele and F. Benjamin

Part 2

THE 5ESS SWITCH HARDWARE

Introduction

K. E. Martersteck and A. E. Spencer, Jr.

Applications Planning

W. R. Byrne and G. P. O'Reilly

Architectural Overview

D. L. Carney, J. I. Cochrane, L. J. Gitten, E. M. Prell, and
R. Staehler

Operational Software

J. P. Delatore, R. J. Frank, H. Oehring, and L. C. Stecher

Maintenance Capabilities

G. Haugk, F. M. Lax, R. D. Royer, and J. R. Williams

Hardware Design

J. C. Borum, J. Janik, Jr., R. F. Metz, J. Russell, D. P. Smith,
and E. J. Theriot

Physical Design/Hardware

F. N. Graff, Jr., C. E. Jeschke, C. R. Komp, B. E. Nevis, and
D. W. Zdan

Operations, Administration, and Maintenance Capabilities

P. T. Fuhrer, L. J. Gitten, B. A. Newman, and B. E. Snyder

System Test, First Office Application, and Early Field Experience

H. A. Bauer, L. M. Croxall, and E. A. Davis

System Development Environment

R. G. Basinger, J. A. Herndon, B. Kaskey, J. A. Lindner, and
J. M. Milner

Factory System Testing

J. P. Delatore, M. P. Tull, and D. Van Haften

ERRATA

In the March 1985 issue (Vol. 64, No. 3) on *Assuring High Reliability of Lasers and Photodetectors for Submarine Lightwave Cable Systems*, the list of editorial staff on the inside front cover should have included the following: "R. L. Hartman, *Coordinating Editor of the submarine lightwave cable reliability issue.*"

In the paper by Z. L. Budrikis and M. Hatamian, "Moment Calculations by Digital Filters," *AT&T Bell Laboratories Technical Journal*, Vol. 63, No. 2 (February 1984), the following corrections are noted:

Page 219: equation (3): The upper limit on the summation terms should be "n" instead of "N".

Page 220, Table I, third column, first row:

$$\frac{1}{z - 1} \text{ should be } \frac{z}{z - 1}.$$

Page 220, Table I, fifth column, first row:
 $u(n)$ should be $u(n-1)$.

Page 220, the equation in the middle of the page between eq. (5) and eq. (6):
 $u(n)$ should be $u(n-1)$.

In the paper by M. Hatamian and E. G. Bowen, "Homenet: A Broadband Voice/Data/Video Network on CATV," *AT&T Technical Journal*, Vol. 64, No. 2, Part 1 (February 1985), the following corrections are noted:

Page 348, second paragraph, second line:
Ref. 1 should be Ref. 2.

AT&T 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.

