THE BELL SYSTEM TECHNICAL JOURNAL

DEVOTED TO THE SCIENTIFIC AND ENGINEERING
ASPECTS OF ELECTRICAL COMMUNICATION

Volume 52 December 1973 Number 10

Information Management System:

Interactive Information Management Systems

By D. T. CHAI and J. M. WIER

(Manuscript received October 5, 1972)

This paper and the following three describe computer systems to store, retrieve, and manipulate information. These have all utilized time-shared computer systems. All have evolved toward a system constructed of modular component parts and having a high degree of user interaction. Considerable attention has been given to implementation in a form suitable for simple transfer to systems of adequate capability with minimal programming effort. The data bases involved are all hierarchical in organization. The major parts are a language facility, a data base manager, a processing package, and numerous coordinated administration functions. The parts are currently assembled into a package which can be applied to an arbitrary hierarchically structured data base with little user effort. The component parts are also available for integration into more tailored systems for special applications.

I. INTRODUCTION

This paper and the three that follow it discuss various aspects of the problem of using computers to store, retrieve, and manipulate information. In particular they describe computer systems for carrying out important parts of such work. These parts have been integrated into a system for handling information. The system described in these papers has been designed so that a user and the computer system can interact heavily in reaching the solution to a problem posed by the user.

Systems generically related to the ones described here have appeared in great numbers in the past decade.[1-6] In general they all use a computer to store, process, and provide results from information contained in a "data base" controlled by the computer. However, this deceptively simple description hides the many differences between the systems which make them less generally applicable than would seem immediately evident. No attempt will be made in the following to be complete in categorizing such systems. However, enough information will be given to place the present work in perspective with respect to important requirements placed on such systems in various applications.

To circumscribe the work reported here and its potential field of application, let us characterize information systems according to the properties indicated in Table I.

The systems which have been implemented using the tools reported here generally are most useful in applications corresponding to the earlier-given of the choices in the various categories. The amount of information contained in the data bases served is generally less than 50,000,000 characters. The information is heavily structured into a hierarchical format. The users are typically not highly skilled in the use of computers. A typical request placed on the system will require fewer than ten seconds of processing. Finally, the user will always expect an answer in less than ten minutes, often in less than one minute, and occasionally in less than ten seconds.

These figures are dictated by the uses to which the systems are usually put, tempered by economic and computer limitations. Relatively small packets of information are supplied to the system in any one transaction. Further, requests to provide information and processing are simple since the user employs on-line composition of requests and interpretation of the results delivered.

The properties implied by this method of interaction cause the resulting system to be somewhat specialized in order to carry out such operations to the satisfaction of the potential users. The following are a few cases where the decision to handle processing in the manner indicated may adversely affect the applicability of the system to other uses.

In order that response time to a given request be short, the system tailors its operations to deal with a spectrum of requests assumed known at the time of system origination. Thus, requests for large

TABLE I—CHARACTERIZATION OF INFORMATION SYSTEMS

Amount of Information:
- Up to 100,000 characters
- 100,000 characters to 50,000,000 characters
- 50,000,000 characters and up

Structure of Information:
- Hierarchical
- Network
- List

Users:
- Non-computer skilled
- Computer skilled

Size of Transaction:
- Less than 10 seconds of processing
- Greater than 10 seconds of processing

Time Scale:
- Less than 1 minute
- 1 minute to 10 minutes
- Greater than 10 minutes

amounts of output, complex or lengthy processing, or data stored in some order much different from that assumed may result in poor service. Specifically, mass business data processing is frequently not well handled in this way.

Since the system is designed to serve an interactive user as well as is feasible, the data base may be more difficult to update or set up in the first place than one specifically designed to be processed as a whole. In the same vein, restart procedures are generally more difficult to incorporate as such operations take time and thus cause poorer time response.

The decision to utilize a hierarchically structured data base means that other organizations will be unavailable, except as they can be mapped onto a hierarchy.

The concentration on serving users who are perhaps not skilled in the use of computers limits the complexity of potential operations.

The exact degree of difficulty for other applications caused by each of these choices varies. The positive benefits obtained have been adjudged sufficient rewards in the thriving areas where the system to be described is used.

II. SYSTEM DESCRIPTION

All of the elements of the total system to be described in these papers have been implemented on a time-shared computer system. The computer system thus takes care of many of the details involved in serving many users. Some of the more obvious and important of these are:

(*i*) Provision of an interface to a communication facility.
(*ii*) Provision for separating users into categories and keeping them apart.
(*iii*) Provision of a flexible charging structure.
(*iv*) Provision for physical storage allocation.

The parts of the information management system are assembled in a modular fashion. Between each of them is a well-defined interface for exchanging information. The components are put together as shown in Fig. 1.

In this figure the users are shown impinging on the system at the left. This contact takes place via the switched telephone network. One or more users can be connected to the information system described at any time. Each user interfaces with the Natural Dialogue System (NDS). The Natural Dialogue System is described more fully by Puerling and Roberto.[7] It provides the ability to carry on a relatively simple interactive pseudo-English conversation with the user in order to ascertain his needs.

When an adequate amount of information is available to define the user service request, the Natural Dialogue System passes information sufficient to define the request to a processor. The processor chosen is determined by the user-NDS dialogue. The processor then uses its input data to make calls on Master Links (ML) to provide specified information from the associated data base or send some to it. Master Links, using facilities described by Gibson and Stockhausen,[8] carries out the operations required on the data base and returns the data needed. The chosen processor then formats the response and sends it to the user. The whole sequence may be reinstituted by the user by placing a new request before the system or the user may actively (by signing off) or passively (by hanging up) abandon his quest.

In this system the processors are one of two types:

(*i*) Job-specific ones that have been specially programmed for an application.
(*ii*) General-purpose ones that have been found to be useful in numerous applications and that thus are provided to all users.

In addition to these elements there exist a number of auxiliary capabilities which are necessary to the smooth and complete operation of such systems. These capabilities are provided by numerous programming packages. They, among other tasks, take care of loading

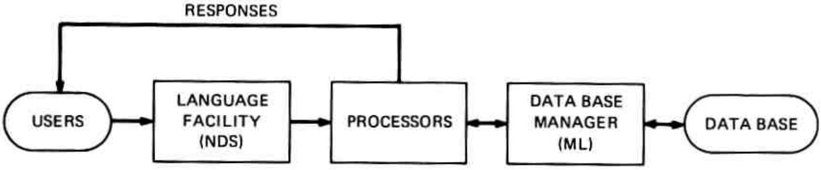


Fig. 1—Components of an information management system.

bulk data, checking its validity, auditing the system for efficiency and completeness, rearranging the data into a different order, and taking statistics on usage. These will not be described.

III. INTERACTIVE INFORMATION PROCESSING

As was mentioned in the introduction, the systems described in this series of papers concentrate on the provision of a highly interactive contact between the user and the information management system. The importance of this type of interaction was dictated by the applications which led to the design. This section discusses some of the considerations leading to the specific design decisions made.

In a very practical sense, an understanding of the system is not possible without examining the environment in which it works. The job of solving problems involving a data base contained in an interactive information system is jointly shared by the user and the system. Each does what "he" can do best.

The information system takes care of data storage, data processing, and information display in addition to a number of housekeeping chores. The user brings in the problem, formulates the solution in the form of a sequence of requests placed before the system, and guides the work of the information system as it progresses.

These operations would appear to be identical with those carried out in classically programmed data processing. The user (a programmer) translates a problem into a sequence of data processing steps which the computer is given to carry out. There is, however, an important difference which makes the interactive process much better for some applications.

That difference stems from the fact that it is not possible to program a computer to provide some solution unless an algorithm exists for doing it. When working with complex data bases, it is frequently necessary to find out a great deal about the data just to be able to write a suitable program. This process of "finding out about the data" is frequently best done by going into the data base on some exploratory trips. It is here that interactive data base processing is very useful.

The user not only can collect the data, but he can also exclude vast areas where it is not worth taking computer time to look. This is possible because he gets a "feel" for the data, the limits of its range, its empty spots, its peculiarities. These allow him to reduce search times, to try simplified models that are "apparent" from looking, and to avoid wasting time and effort. All of these are simple for the user to employ while provided with immediate response from the data base. They are frequently difficult to program. Recognition of patterns is one of man's strong points. Generation of all possible patterns to be explored is not.

In order to provide this interactive capability it is necessary to smooth the communication between the user and the interactive computer system. This process is not a simple one. Basically, it involves a smooth translation from a form which is "natural" and unambiguous to the user to one which the computer can use on input. On output the process is reversed.

Numerous studies have been made of the use of English as a communication medium for talking with a computer.[3,5,6,9–11] Unconstrained English serves this purpose poorly, not only because of implementation difficulties, but because of the heavy use of context and alogical constructions. Even the REL[5] system which has progressed a long way toward natural language usage requires a rather disciplined approach to construction and meaning. Montgomery[11] has collected numerous telling examples which clearly illustrate the difficulties. These problems have led the designers of this system to adopt a pseudo-English language based on independent phrases, each of which begins with a specified keyword. The use of keywords greatly reduces the ambiguities of the user request and, at the same time, reduces the parsing or analyzing time by the computer. The paper by Puerling and Roberto[7] describes the keyword style of languages that is available through the use of the Natural Dialogue System. The paper by Heindel and Roberto[12] describes one implementation of a keyword language for general-purpose retrievals.

The choice of accepting independent phrases in a request also materially simplifies another computer-user interaction process. Economics and the state of technology strongly recommend a keyboard input mechanism (other choices cost too much or are not well developed technically). Unfortunately, typing, particularly facile typing, is not a universally available skill. Thus input, for many potential users, is clumsy and is often a source of errors. The time delays and annoyances in this process often put off potential users and reduce the value of a system. The use of a phrase-type grammar provides some help in the system described by reducing retyping on errors to the level of the phrase rather than the sentence. In actual use the quantity of typing is further reduced by providing editing facilities which preserve common material already placed in the system from interaction to interaction.

A second communications barrier which can exist in an interactive system is that of response time. If the user is employing the system in an interactive way in the pursuit of a solution to his problem, he finds that excessive delays in delivering replies to his requests create gaps in the continuity of his thoughts on the solution. They distract him and, more seriously, they affect his ability to note patterns in the output. They thus reduce his effectiveness in solving the problem. They also bore him and waste his time, both of which reduce the probability that a proper and prompt solution will be forthcoming.

Because of the effect on user acceptance and user effectiveness, the systems to be described have been implemented with response time a major criterion of merit. This criterion has shaped the system in at least the following ways:

(*i*) The complexity of a request is reduced by making simple requests easier to formulate than complex ones.

(*ii*) The Master Links data base management system provides numerous tools for tailoring a data base to the requirements of its potential users.

(*iii*) The languages are designed to reduce search time in the data base by simplifying the specification of data base delimiters.

(*iv*) Monitors have been provided for noting the state of the data base and the usage by the system clientele.

(*v*) Dialogue is retained from request to request to reduce the typing burden.

(*vi*) Numerous detectors of errors are employed and extensive helpful (not critical) diagnostics are provided.

IV. SOME COMMENTS ON PERFORMANCE

As has been mentioned, the systems described have been designed to deliver prompt response in an interactive environment. In addition to pursuing the goals just mentioned, the software has been designed to perform well in an absolute sense as well. To measure the performance actually achieved, extensive unit testing has been employed. In traffic situations, simulations have been run on the performance in the presence of various levels of load. Overall tests of system performance have been designed and run. A model for evaluating system performance as a function of the processing to be done in the data base has been developed. Such tests and models have been most helpful in comparing different system implementations and algorithms. The knowledge so gained has also been used in updating designs and optimizing system use.

The systems described have been used in various applications with data bases containing up to a few tens of millions of characters of data. These have all been hierarchical in organization and generally did not employ more than ten levels in the hierarchy. By using the various tuning facilities, the time to return answers to typical requests can often be reduced below ten seconds. More complex ones occasionally run to a few tens of seconds, but these employ less commonly used facilities. In general, requests requiring extensive data searches are more time-consuming than those requiring less information.

The key to good performance lies in matching the information management system to the needs of the application. In most applications the system can be tailored to provide adequately prompt service for the spectrum of common requests, sometimes at the expense of less important functions. These latter can usually be handled, less expeditiously, without creating an operational problem as they occur less frequently. In the current state of the art, no economic solution has been found which does not require this compromise for the larger and structurally more complex data bases. In all of the latter it is always possible to find pathological interactions with the data base which force data base searches in a very poor order.

V. SUMMARY

The work done in designing, testing, and applying the systems described has indicated the following:

(*i*) Interactive information management systems of acceptable performance are feasible and economically attractive in the current state of the art.

(*ii*) The hierarchical data base organization has been no handicap in providing information management in most applications tested.

(*iii*) It is desirable to match an information management system to the application in order to get prompt responses from it.

REFERENCES

1. Cuadra, C. A. (editor), *Annual Review of Information Science and Technology*, Interscience Publishers, 1966, 1967; Encyclopaedia Brittanica Co., 1968, 1969, 1971.
2. Senko, M. E., "Information Storage and Retrieval," in *Advances in Information Systems Science*, 2 (ed. by J. T. Tou), Plenum Press, 1969, pp. 229–281.
3. Salton, G., and Lesk, M. E., "The SMART Automatic Document Retrieval System—an Illustration," Comm. ACM, *8*, No. 6 (June 1965), pp. 391–398.
4. Sinowitz, N. R., "DATAPLUS—A Language for Real Time Information Retrieval from Hierarchical Data Bases," Proc. AFIPS, *32* (SJCC 1968), pp. 395–401.
5. Dostert, B. H., "REL—An Information System for a Dynamic Environment," REL Report 3, California Institute of Technology, December 1971.
6. Chai, D. T., "An Information Retrieval System Using Keyword Dialog," Information Storage and Retrieval, *9*, No. 7 (July 1973), pp. 373–387.
7. Puerling, B. W., and Roberto, J. T., "The Natural Dialogue System," B.S.T.J., this issue, pp. 1725–1741.
8. Gibson, T. A., and Stockhausen, P. F., "MASTER LINKS—A Hierarchical Data System," B.S.T.J., this issue, pp. 1691–1724.
9. Woods, W. A., "Procedural Semantics for Question Answering," Proc. AFIPS, *33* (FJCC 1968), pp. 457–471.
10. Kellogg, C. H., "A Natural Language Compiler for On-Line Data Management," Proc. AFIPS, *33* (FJCC 1968), pp. 473–492.
11. Montgomery, C. A., "Is Natural Language an Unnatural Query Language?" Proc. ACM, *25* (August 1972), pp. 1075–1078.
12. Heindel, L. E., and Roberto, J. T., "The Off-The-Shelf System—A Packaged Information Management System," B.S.T.J., this issue, pp. 1743–1763.

Copyright © 1973 American Telephone and Telegraph Company
THE BELL SYSTEM TECHNICAL JOURNAL
Vol. 52, No. 10, December, 1973
Printed in U.S.A.

Information Management System:

MASTER LINKS—A Hierarchical Data System

By T. A. GIBSON and P. F. STOCKHAUSEN

(Manuscript received October 5, 1972)

MASTER LINKS is a software system used to build, administer, and access hierarchical data bases. It is designed to operate in a time-sharing environment, and, in particular, it allows multiple concurrent updates and retrievals on the same data base.

A BUILD module is used to specify the hierarchical configuration of a data base and an initial "storage mapping" of the elements of the hierarchy into a particular file layout. A set of administrative routines is provided for altering the mapping and other such maintenance purposes. The access routines have three levels of interface, from primitive and flexible to sophisticated and functional. The interfaces are all defined in terms of the hierarchical structure and independent of the storage mapping. Thus, an alteration of the storage mapping for a data base does not require changing any programs that access data using these interfaces.

The lowest-level interface enables the calling program to add to the data base, update a value, or retrieve a value, in terms of a hierarchy position. The second-level interface facilitates traversal of a hierarchy by enabling the calling program to specify portions of the hierarchy over which a process is to operate. Such a specification, called an "access tree," consists of data which can be generated at execution time by the calling routine. As in the first level, data are transferred one at a time. The third-level interface is a function evaluation mechanism which computes values from data base values and other computed values according to function definitions passed to it at execution time. Like an access tree, a function definition is itself data which can be constructed at execution time by the client process.

I. MASTER LINKS OBJECTIVES

The Master Links data system is a collection of software that accesses and manipulates data stored in a hierarchical structure on a computer's secondary storage devices. It services requests from "client" programs to store and retrieve data, and to create and release space in the data structure.

The Master Links project was designed with the following goals:

(*i*) Provide a basic "low level" set of access mechanisms to retrieve and store data items, and to create and delete branches of the hierarchy. Client programs using these mechanisms work entirely in terms of the hierarchical structure.

(*ii*) Provide "high level" access mechanisms that simplify the programming task for complex retrieval requests.

(*iii*) Support many concurrent users on a data base, doing both retrievals and updates.

(*iv*) Operate well in a time-sharing environment.

(*v*) Enhance portability of the system by basing its design on machine-independent concepts.

Other goals are presented in the text of this paper.

This report begins with a definition of the elements of hierarchical data structures, and a description of the basic access mechanisms, in Section II. Section III examines the requirements of typical client processes. Then high-level access mechanisms are described in Sections IV and V. Thus, these four sections describe the system as viewed by its users. Section VI delves into the system design and shows how the structures are arranged to provide these capabilities in portable form with high performance. The final section discusses the experience acquired with current implementations, and presents an outline of current and future developments of Master Links.

II. ELEMENTS OF HIERARCHICAL DATA STRUCTURE

The elements of a hierarchical data structure are entities, groups, and fields. Groups and fields are the permanent elements of a data base. They are established by a process called "building" the data base. Entities are the dynamic elements. They are added and deleted at any time by client programs using the basic access mechanisms of Master Links. Client programs also use the basic access mechanisms to transmit data for a field of an existing entity.

2.1 *Entities, Groups, and Fields*

A *field* is a set of data all identified by the same *field name.* There are several types of fields: numerical, character, logical, and date.

An *entity* is an element which holds one value for each of a given list of fields. We will draw an entity as a rectangle, with the field names to one side and the values inside, thus:

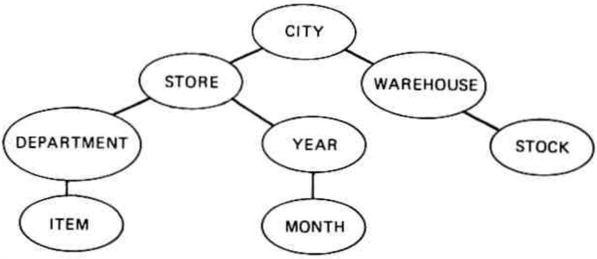
STORE NAME	PLAZA
EARNINGS	10325

A *group* is a set of entities with the same fields. A group has a name which indicates the nature of its entities. The name of a group will be written in an ellipse:

STORE

STORE NAME	PLAZA	MAIN ST	RT 46	PLAZA
EARNINGS	10325	69238	21420	96823

A data base is composed of a set of groups which are hierarchically related. One group is the top group. All others descend in a tree fashion:



This is a STORE and WAREHOUSE data base. It is subdivided by CITY at the top. Each city of the chain of stores is represented by one entity in the CITY group. There are several stores per city, several departments per store, and several items per department. In addition, certain data are kept on an annual and a monthly basis for each store. Each city also has zero, one, or more warehouses, and there are several items of STOCK per warehouse.

Although called a tree, the structure is always drawn "upside down." This is not in fact unusual. Corporation organization charts are frequently drawn this way, as are part lists, inventory lists, etc. It

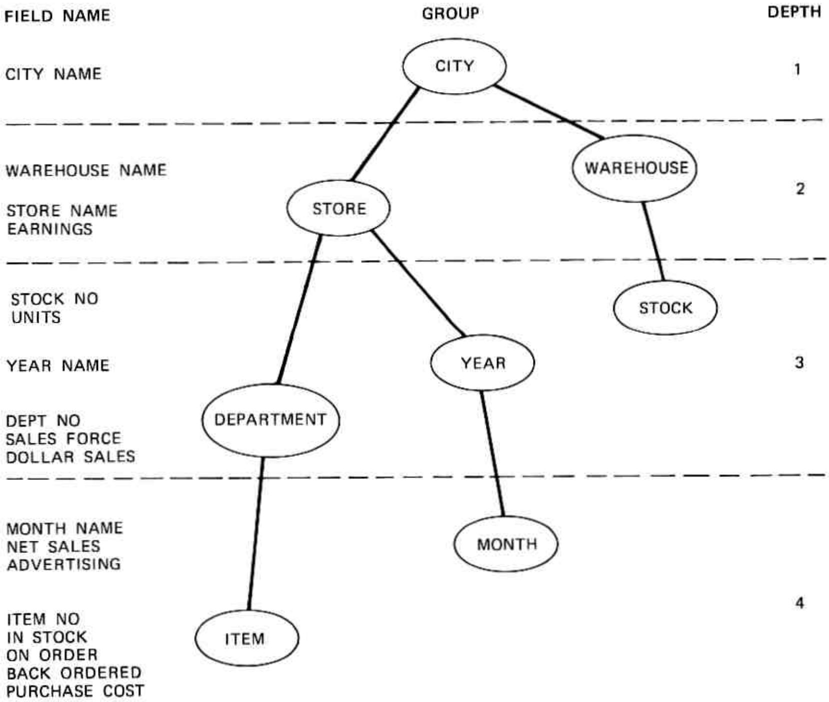
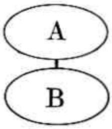


Fig. 1—A group tree with fields.

places the major components at the top and the detailed ones lower down.

Figure 1 shows this data base with fields assigned to each group. Figure 2 shows a blowup of the entities. The ellipsis (· · ·) in a group indicates several entities not shown.

The parent of a group is the group immediately above it. The top group has no parent. All other groups can have only one parent. The parent of an entity is the entity immediately above it. Entities in the top group have no parent and all others have one parent. The parental relation of entities must parallel those of the groups. Thus if group B has group A for its parent:



then all entities of B must have their parent entities in A.

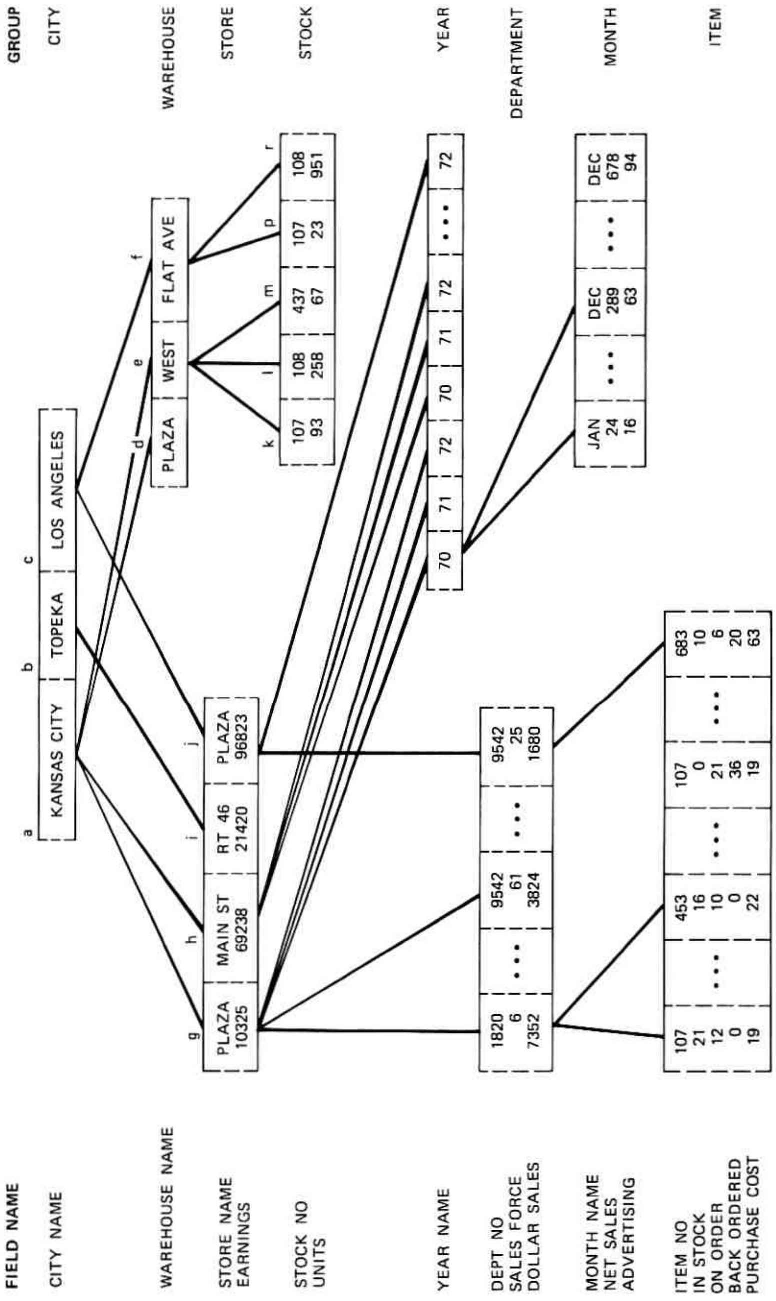


Fig. 2—A blowup of the entities.

In Fig. 2 the entities are labeled a, b, · · ·, r. These labels are not a part of the data base, but are used only as references in this paper.

We have made use of the terms "parent of an entity" and "parent of a group." This suggests the use of other genealogical terms. The kth ancestor of an entity is the entity k steps above it. (Hence the first ancestor is the parent.) The offspring of an entity are all the entities immediately (one step) below it. The descendants are all the entities below it.

For each entity, all its offspring in one group form *a family*. Entities a, b, and c are a family; d and e another family; g and h another family. Notice that entity a has two families under it, one in STORE and one in WAREHOUSE. If two entities are in the same family, such as d and e, they are *siblings* to another. If two entities have the same parent, but are in different families, such as d and g, they are *step-siblings*.

2.2 *Building the Data Base and Entity Dynamics*

A particular data base is established by defining the group tree and the fields of each group. This process is called *building* the data base. The language for describing the data base is called the *build language*. Using this language a data base designer describes the permanent attributes of his data base and submits the description to a utility program called BUILD. After BUILD has processed the description, the data base has no entities, and no data, but only a "skeleton" structure.

Entities are the dynamic components of a data base. They may be added or deleted online, even while other users are working on the data base. Thus the actual data base grows and acquires data, but always in accordance with the structure defined by BUILD.

2.3 *Basic Access Mechanisms*

There are five basic operations which programs can perform on the data structure:

(*i*) Select a top entity or an entity whose parent has been previously selected.

(*ii*) Add a new offspring to a selected entity or add a new entity to the top group.

(*iii*) Delete a selected entity.

(*iv*) Select a field.

(*v*) Transmit data to or from a selected entity for the selected field.

These five basic operations make possible any manipulation of the data structure except modification of the permanent attributes of the data base established by BUILD.

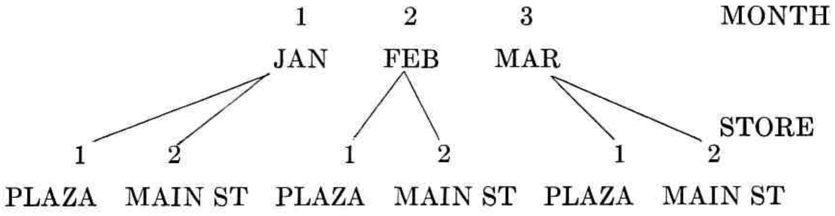
2.4 *Identifications*

A user (or a program) accessing the data base must be able to uniquely identify each element. Users identify elements by *names*, such as 'KANSAS CITY,' 'WAREHOUSE,' or 'EARNINGS.' Names are also called *external identifiers*, because they are used (by people) external to the software. The Master Links software uses *internal identifiers*, which are integers such as group 2, entity 7, field 13. The term *identifier* refers to both internal and external identifiers.

Fields and groups are given unique identifiers. Figure 2 shows group and field names. These names are selected at the time the data base is built and then do not change. Their internal identifiers are positive integers assigned by BUILD.

The identification of entities must be done somewhat differently, since they are not established by BUILD. The internal identifier of an entity is a positive integer called the *entity index*. The first entity of a family has index 1, the second has index 2, etc. Thus, the internal identifier is unique only within a family.

This method of identifying entities allows implicit associations to be established among the entities of a group. The most common use of this is to assign the same index to all the entities which have some attribute in common. In this case the external identifier names that attribute. For example, a data base with just MONTH and STORE groups is shown with internal and external identifiers.



All store entities with index 1 have the common attribute of storing data for the PLAZA store. One can request processing of all data for the PLAZA store, and this will cause all STORE entities with index 1 to be processed. To uniquely identify a single entity in the store group requires specifying both a month and a store identifier.

Another implicit association possible is the ordering of entities. Again using months as an example, JAN through DEC can be assigned indexes 1 through 12 respectively.

III. THE CLIENT PROCESS

Processes that access data bases have a strong tendency to access either many fields from a few entities, or few fields from many entities. An example of the first type is

ENTER NEW SHIPMENT DATA FOR WAREHOUSE______.

Values for many fields are to be put into one warehouse entity. For this type of request the basic access mechanisms are quite convenient. An example of the second type of request is

> FOR ALL STORES IN CITIES ____, ____, AND ____, PRINT STORE NAME, AND 1972 NET SALES PER STORE DIVIDED BY EARNINGS.

Only a few fields (STORE NAME, NET SALES, and EARNINGS) are required, but a large number of specific city, store, year, and month entities must be accessed to fulfill this request. Further, the values of NET SALES and EARNINGS that are retrieved must be functionally combined into the values of "1972 NET SALES per store divided by EARNINGS." It is possible to do these tasks by using the basic access mechanisms, but the programming is tedious and lengthy. Master Links provides a set of higher-level access mechanisms that makes programming of the above PRINT request as simple and straightforward as this:

(*i*) Declare which entities are to be processed.
(*ii*) Step to each of these entities in turn, and retrieve and print a value for the requested function.

The entities to be processed are declared with an *access tree.* The access tree provides directions to the *generator* which steps to each of the entities in turn. Finally, the retrieval is performed by the *function evaluator* which does all the work of evaluating functions of data stored in a data base. These tools for client programmers are described in the next two sections.

IV. ACCESS TREES AND THE GENERATOR

An access tree describes a subtree of the entities of the data base. Thus any entity on the access tree has all its ancestors on the access tree. It can also be visualized as a "pruned" entity tree: when an entity is removed, so are all its descendants. Several concepts underlie the mechanism for building an access tree:

(*i*) The generated group
(*ii*) The refined inclusion of an entity
(*iii*) The refined set of entities
(*iv*) Independently refined sets
(*v*) Whole-family inclusion.

These are described in turn. The data base of Fig. 2 is used for all examples.

4.1 *The Generated Group*

Some groups of the data base will contain data needed by the process, and some will not. Those that contain needed data, and all their ancestors, are the generated groups. The rest of the groups have no entities on the access tree and therefore will not be generated.

The client process may specify what groups have needed data. It therefore specifies by implication the generated groups and the groups to be pruned from the access tree. The generated groups all have entities on the access tree. They are there either by refined inclusion or by whole inclusion.

4.2 *Refined Inclusion*

Refined inclusion means an entity has been put on the access tree by explicitly giving its group identity and entity identity within its family. In Fig. 3, KANSAS CITY has been explicitly named to the access tree, and therefore is a case of refined inclusion. In writing programs, the internal identities are used: the group number and the entity index within its family. In our examples in this section, we will use external identities, as has been done in Fig. 1. It is confusing to wade through a lot of numerical codes in examples when trying to learn about concepts.

The year entity whose name is 70 is not unique. There are several such entities, one for each STORE entity. They have the same external identity, 70, the same internal identity, index 1, and therefore

REFINEMENTS

GROUPS	REFINE LIST 1	REFINE LIST 2	REFINE LIST 3	REFINE LIST 4
CITY	KANSAS CITY	KANSAS CITY	TOPEKA	KANSAS CITY
STORE	PLAZA	MAIN ST	RT 46	
WAREHOUSE				WEST

INDEPENDENT REFINEMENTS

GROUPS	REFINE LIST 1	REFINE LIST 2
YEAR	71	72
MONTH	DEC	JAN

GENERATED GROUPS

CITY, STORE, YEAR, MONTH, WAREHOUSE, STOCK

RESULTING ACCESS TREE

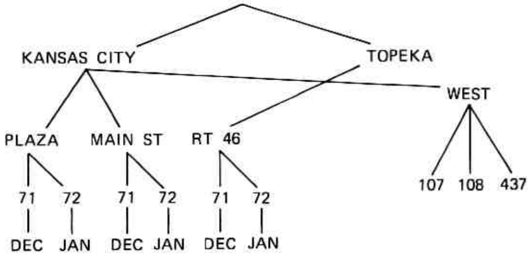


Fig. 3—Building an access tree.

have an association from family to family by identity. Wherever this condition exists, a single refinement can describe many entities in the data base. This is called *multiple refinement.* A refinement to

GROUP	*REFINE LIST*
YEAR	70

denotes every 70 entity of Fig. 2.

Refinements can depend on specific ancestors. This happens when a refine list has two or more entries. Thus:

GROUP	*REFINE LIST*
CITY	KANSAS CITY
STORE	PLAZA

identifies the PLAZA store only in KANSAS CITY, not the one in LOS ANGELES. PLAZA is called a *dependent refinement.*

A refinement can be both multiple and dependent, hence is called a *multiple-dependent refinement.* An example is

GROUP	*REFINE LIST*
YEAR	70
MONTH	JAN

which specifies a set of 70 entities, and the JAN entities under those 70 entities.

A refinement is not restricted to immediately adjacent levels of the data base. The following refinement is acceptable:

GROUP	*REFINE LIST*
CITY	KANSAS CITY
MONTH	JAN

The groups of a refine list must proceed down the data base from ancestors to descendant. However, groups may be skipped in the list.

4.3 *The Refine Set*

A *refine set* is a set of refine lists on particular groups. The groups of the refine set may be any groups, but the first group must be an ancestor of all the others. Figure 3 shows a refine set on the groups CITY, STORE, and WAREHOUSE, and another refine set on YEAR and MONTH.

A group can only be involved in one refine set. Every refine list of a set must start with an entity from the set's first group. Hence, to be a legal refine list, it must proceed to give entities ancestor to descendant down one path of the group tree, and from groups in the refine set.

A refine set allows a multiplicity of dependent refinements. The English clause

FOR PLAZA AND MAIN ST STORES AND WEST WAREHOUSE IN KANSAS CITY AND RT 46 STORE IN TOPEKA

is easily represented as a refine set. In fact, Fig. 3 gives the refine lists to do this.

4.4 *Independent Refine Sets*

Refine sets alone cannot be used to represent independent refinements. The clause

FOR CITIES ___, ___, ___, ___, and ___, IN THE YEARS ___, ___, ___, ___, and ___

represents five cities, and five years in each city. The list of cities is a refine set on one group, CITY. The list of years is a refine set on another group, YEARS. What the entire clause specifies is the independent combination of these two refine sets. This leads to the concept of independent refine sets.

Refine sets are independent if their groups are mutually exclusive, i.e., any group can be in only one of the refine sets. Figure 3 shows a pair of independent refine sets and the resulting members of the access tree. Any number of independent refinements can be put on an access tree.

4.5 *Whole Inclusion*

Figure 3 has all stock items of the WEST warehouse on the access tree. Any generated group not in a refine set is included on the access tree on a *whole inclusion* basis. This means that whole families of the group are either included or excluded from the access tree, depending on whether their parent is included or excluded respectively.

Using Fig. 2, other examples of whole inclusion are:

(*i*) CITY group refined to KANSAS CITY, STORE group whole, all other groups pruned from the access tree. This puts entities a, g, and h on the access tree.

(*ii*) CITY and STORE groups whole. All other groups pruned. This puts all cities and all stores on the access tree.

(*iii*) CITY refined to KANSAS CITY and TOPEKA. YEAR independently refined to 71 and 72. STORE and MONTH whole. All other groups pruned. This puts entities a, b, g, h, and i; all 71 and 72 entities under g, h, and i; and all month entities under those year entities onto the access tree.

4.6 *Generating Entities on the Access Tree*

The generator accepts an access tree as an input. It generates only entities on the access tree. For brevity in this section we will say "offspring," "sibling," etc., but always mean "offspring on the access tree," "sibling on the access tree," etc.

On each call the generator takes one of the following actions:

(A) Takes a step to the "next" entity of the access tree, opening that entity for data accesses. The entity reached is said to be generated. The client program is informed of the group of the entity and the entity's identity among its siblings.

(B) Notifies the client process that the previous entity generated was the last of a family on the access tree. A new entity is not generated on this call. This action gives the client process an opportunity to perform summary processing on families.

(C) Notifies the client process that the previous entity generated was the last on the access tree. This action gives the client process an opportunity to perform final summary processing, and to exit from the processing loop.

On the first call, the entity generated is the leftmost entity of the top group. This becomes the current entity. On subsequent calls, the generator tries to step from the current entity to another entity in the following order:

1. Leftmost offspring of the current entity.
2. Sibling of the current entity.
3. Step-sibling of the current entity.

The first of these that succeeds becomes the new current entity. If all fail, the current entity is redefined as the parent of the current entity, and the above process resumed at step 2. The effect is to continue the list with

4. Sibling of the first ancestor of the current entity.
5. Step-sibling of the first ancestor of the current entity.
6. Sibling of the second ancestor of the current entity.

⋮

etc.

This process defines the meaning of "next" entity for action A.

Actions B and C allow the client program many opportunities to perform processing on individual entities, summaries after families of entities are generated, and a summary at the end of the tree. Process loops are generally organized with the generator at the top of the loop. Following this is a section of code that tests which action was effected by the generator, and at what group. If an entity is generated in a group where retrievals are to be made, control is passed to a section of code that makes the retrievals from that entity and processes the data.

If a "done with family" action is signaled on a group for which family summaries are being made, control is passed to a section of code that effects the summaries for that group. After each of these code sections is complete, control is returned to the generator to take the next step. Eventually the "done with tree" action is signaled. The process then exits from the processing loop and executes terminal processing.

Only the generator looks at the access tree data structure, and it confines the process to entities on the tree. Entities not on the access tree simply do not exist, as far as the process loop is concerned.

The process can direct the generator to break from its normal sequence of "next" entity steps. Thus further screening by data dependent "match" conditions of the entities to be processed can be done. When an entity of a particular group is reached, the client process can retrieve data from it and test for a match condition. If the match condition is satisfied, other data are retrieved from the entity and entered into the process. Then the generator is told to generate the next entity, usually an offspring of the current entity.

But if the match condition is not satisfied, the client program goes directly to the generator, calling it with a skip option that causes all descendants of the current entity to be skipped. Normally the next entity generated in this case is a sibling of the current entity. Other skip options are available. Thus the process has final control over the entities entering the process, within the confines of the access tree.

4.7 *Summary of Access Trees and Generators*

The generator and access trees provide a mechanism for efficiently accessing in a subtree of a data base those entities which may supply the data needed to process a particular request. Access trees have a natural derivation from English clauses that delimit the scope of a request. The generator can directly access the entities specified by an access tree. Thus, together, they constitute a very significant bridge between natural-language query and efficient retrieval algorithms.

V. FUNCTION EVALUATOR

An application program interacts with a data base at each entity generated. For retrieval processes, the values to be displayed are often combinations or functions of the stored data. The function "NET SALES per store divided by EARNINGS" has one value per entity of the store group. The values for this function could have been given a name at build time, and established as a field of the data base. In theory, any possible function that has one value per entity of some group could be a stored field of that group. In practice, only those of sufficiently high usage are stored, and the others are computed on request. Thus there must be some mechanism which can deliver upon request the value of a stored field or of a function of stored fields. This mechanism is called the *function evaluator.*

This section presents a definition of several classes of fields whose values are not stored but can be derived from the stored data and from the hierarchical structure itself.

5.1 *Summarizing a Field*

A hierarchy provides a structure for efficiently summarizing data. For example, a user of the sample data base may require the total DOLLAR SALES for each store. To obtain such a total for a given store, the values of "DOLLAR SALES" must be summed over each department in that store. Repeating this summation for each store produces a set of values for the derived field "total DOLLAR SALES per store," defined at the store group. This type of function is called a *level raise* because it raises the level of definition of a field from one group to a higher group.

The set of entities used to evaluate a level-raise function for the store group consists of one entity of the store group and a collection of descendants of that entity. A *subtree under group G* is defined as an access tree containing at most one entity of group G. Hence, in the descendants of G a subtree under group G may branch out, but from G up to the root there is only one entity path. Let G be an ancestor of G′ and f′ a stored or derived field of G′. A level raise produces a value for a field f of group G by summarizing a field f′ of group G′ across the G′ entities in a subtree under group G. Values entering into the level raise are those of f′ for entities of the subtree. The set of values it produces for all entities of G defines a field f of group G.

In the level-raise function "total DOLLAR SALES per store," "total" is an instance of a level-raise operation, "store" is G, and "DOLLAR SALES" is f′, where G′ is the department group. In order to construct an efficient computation algorithm, level-raise operations are restricted to those which can operate sequentially on a set of values for the field f′ to produce a single value of f. Examples of level-raise operations are total, average, minimum, maximum, and standard deviation for numeric-valued fields; any, all, and none for logic-valued fields; and concatenation for character-string fields.

5.2 *Retrieval Within an Entity*

A derived field measuring average sales might be defined as "DOLLAR SALES divided by SALES FORCE." Since both component fields are defined for each department entity, their quotient is also defined for each department entity, and hence describes a derived field of the department group. Any function of fields is called a field function. The operands of a field function may be level-raised, as in "maximum DOLLAR SALES per store divided by EARNINGS." This is a function of two store fields, "maximum DOLLAR SALES per store" and "EARNINGS." A field function can be defined in terms of fields of different groups, as in "DOLLAR SALES divided by EARNINGS." The numerator is a department field, the denominator a store field. The expression has one value for each department, and hence defines a field of the department group.

A field function for group G is defined as any function of constants, fields of G, and fields of ancestors of G. These fields may be stored or derived. A field function produces a new field of G. It is applied at a single entity and produces a value defined for that entity. The class of field functions contains such operations as the standard arithmetic, Boolean, and trigonometric operations; logarithms; and IF–THEN–ELSE assignments.

Arbitrary nesting of level-raise and field functions is well defined since a function of either class generates a field. An example of such nesting is "maximum per store of (DOLLAR SALES minus total per department of (PURCHASE COST times the sum of ON ORDER and BACK ORDERED))." This expression is equivalent to the following statements:

x = PURCHASE COST times the sum of ON ORDER and BACK ORDERED
y = total x per department
z = DOLLAR SALES minus y
f = maximum z per store.

x, y, z, and f are derived fields. x is an item field, y and z are department fields, f is a store field.

5.3 *Entity Specification and Qualification*

The functions considered thus far generate new fields. The next discussion treats functions which modify the set of entities over which a field is evaluated. An *entity-specification* function describes a process which, given a subtree under group G, selects another subtree under group G using only the intrinsic order of the entities in each group or a constant entity designation. In the sample data base, the years and months are ordered within their respective families. Therefore, the request "ADVERTISING divided by previous year ADVERTISING" is defined for each month. Given a particular month the numerator is obtained directly, whereas the denominator is retrieved for an entity whose location in the tree structure is determined relative to the given month by the operation "previous year." This is called a *relative entity specification.*

Constant entity specification denotes a fixed subtree under group G which overrides the given subtree. The ratio of NET SALES to January NET SALES describes a constant entity specification.

Hierarchical structures make entity specification an efficient process for selecting the entities over which to evaluate an expression. A more general but less efficient selection is that of entity qualification, as in the "with" phrase of "average per year of (BACK ORDERED with PURCHASE COST greater than 500)." Entity qualification is independent of the order of entities in a group. All entities must be examined according to a criterion, such as "PURCHASE COST greater than 500." Each entity is assigned the value "accept" or "reject." When an entity is rejected, all of its descendants are rejected as well. The descendants of an accepted entity are likewise accepted as far as that criterion is concerned. The qualification process is inefficient because data must be retrieved for all candidate entities; in entity specification no test data are retrieved from any entities. Hence, the earlier example with a January specification might be equivalently phrased "NET SALES divided by total NET SALES with MONTH NAME = JAN." In the denominator, each month entity must be examined to determine whether or not its name is January. Although constant entity specification can be contorted into entity qualification if entity identifiers are stored values, the relative entity-specification functions, such as "previous," cannot be expressed at all with entity-qualification, unless the family-order relations are also stored as data values.

In summary, level-raise and field functions can be computed for all entities of a group. A function of either type produces one value for each entity of a group, and hence defines a nonstored field of the group. Entity specification and qualification functions produce a subtree at each entity of a group. A function evaluator enables the user of an interactive data system to dynamically define and redefine derived fields and retrieve values for these fields, all in an interactive communication in real time. A field is either a data-base field or a function of fields, such that it has one value for each entity of some group. A function evaluator enables a client program to retrieve values of a field without having to distinguish whether the field is stored or derived. To accomplish this, a function evaluator must be able to accept function definitions during the dialogue, rather than have them compiled into machine-executable code. In addition, it must be capable of evaluating arbitrarily nested functions if the user can truly ignore distinctions between stored and derived fields. Otherwise the user would be constrained to use only specific types of fields in each class of functions.

5.4 *The Retrieval Process*

This section presents an algorithm for an evaluator capable of computing values for derived fields over a hierarchical data base. The algorithm is a recursive procedure.

Input: A four-tuple: (f, G, t, S).

Arguments:

f: A field defined at group G.
G: A group.
t: An entity-selection function for group G.
S: A subtree under group G.

The argument f can be a stored field, v; a field function, *p*; or a level-raise function, *l*.

p: A field function whose *n*th argument is the triple (f_n, G_n, t_n):
f_n: A field for group G_n.
G_n: A group, either G or an ancestor of G.
t_n: An entity-selection function for group G_n.
l: A level-raise function with three arguments:
f′: A field for group G′.
G′: A descendant of G.
t′: An entity-selection function for group G′.

The argument t can be an entity-specification function, *s*, or an entity-qualification function, *m*.

s: An entity-specification function with one argument:
t_1: An entity specification for group G.
m: An entity-qualification function with three arguments, having the same definition as a triple in *p*, above.

The algorithm for this evaluation is as follows.

1. If t has the form $s(t_1)$, perform the specification function s on S at G to produce a new subtree S_1; then evaluate (f, G, t_1, S_1). Return.
2. If t has the form $m(f_1, G_1, t_1)$, set t′ to null, evaluate (f_1, G_1, t', S), and perform the qualification function m on the result to produce S_1. If S_1 contains an entity of G evaluate (f, G, t_1, S_1) and return; otherwise return a null value.
3. If f is a stored field, v, retrieve its value for the entity of group G on S. Return.
4. If f has the form $p\{(f_1, G_1, t_1), (f_2, G_2, t_2), \cdots\}$, do step 4a for each component (f_n, G_n, t_n); apply p to the resulting values and return.

   4a. If $G_n = G$, evaluate (f_n, G, t_n, S); otherwise construct S_n as the subtree under group G_n containing the entity of G_n present on S and containing all of that entity's descendants, and evaluate (f_n, G_n, t_n, S_n).
5. If f has the form $l(f', G', t')$, select S′ as a subtree of S under group G′, containing the first G′ entity of S. Evaluate (f', G', t', S'). For each succeeding entity of G′ on S, do step 5a. Return.

   5a. Construct a subtree S′ for group G′ using the G′ entity specified in step 5, and evaluate (f', G', t', S'); then apply l to the previous result and the new value.

This algorithm is summarized by the following production.

$$\begin{aligned}(f, G, t, S) \rightarrow\ & (f, G, s(t_1), S)\ |\\ & (f, G, m(f_1, G_1, t_1), S)\ |\\ & (v, G, t, S)\ |\\ & (p\{(f_1, G_1, t_1), (f_2, G_2, t_2), \cdots\}, t, S)\ |\\ & (l(f', G', t'), G, t, S)\end{aligned}$$

5.5 *Unavailable Data*

Some of the fields in a data base may not have values at some time. For example, a new stock item, X, may be ordered although its selling price has not yet been determined. Now someone designing a new product using parts X, Y, and Z needs to determine the total selling price of the components, that is, a summation level raise restricting items to X, Y, and Z. Clearly, if the value of "selling price" is unavailable for X, then the value of the sum is also unavailable. Should an unavailable unit of data be assigned the value zero, the level raise would produce 0 plus Y plus Z as the material cost of the product—an alarming situation at best. Similarly, the field function "selling price times IN STOCK" must yield a result of unavailable if the value of either operand is unavailable. Notice that this situation is not one in which the user has entered a value of null, but rather one in which the data are not available or have not been entered. NA (*n*ot *a*vailable) indicates that no significant data are present.

Unavailable values occur as well in logical-valued fields, particularly when level-raising with operators of "any" and "all." "Any X" has the value "true" if the field X has the value "true" for any descendant of the current entity. "All X" is true only if X is true for all descendants. If X has a value of NA (not available) it could be true or false, but we do not know which. Representing TRUE, FALSE, and NA numerically such that TRUE < NA < FALSE, an investigation of each possible situation will verify the following:

$$\begin{aligned} \text{any X} &= \text{min (X)} \\ \text{all X} &= \text{max (X)} \\ \text{any not X} &= \text{not all X} \\ \text{all not X} &= \text{not any X} \\ \text{where: not NA} &= \text{NA,} \\ \text{not TRUE} &= \text{FALSE,} \\ \text{not FALSE} &= \text{TRUE.} \end{aligned}$$

In criteria evaluation, such as testing if X is less than Y, the result must be NA if the value of either X or Y is NA. If either value is unknown, the criterion may or may not be satisfied; the result is unavailable.

NA is a value which describes the absence of a value. Entity-qualification functions produce an accept or reject status. "Reject" describes the absence of an entity. In "average (ON ORDER with PURCHASE COST greater than 500) per department" the qualifier rejects entities in the averaging. "Reject" is needed as a value of stored and derived fields as well. It enables IF–THEN–ELSE statements to express entity qualification. Moreover, suppose that before April of a certain year the entire sales of a department was recorded in the NET SALES field, while after that time the sales were broken down into NET SALES and SALES TAX. Now if SALES TAX is given the value zero in the first months, the expression "average tax per year" for any department will produce a peculiar result because of the zeroes averaged in. NA is unacceptable as the value of SALES TAX in the early months since it would cause a field such as "average (NET SALES plus SALES TAX) per year" to return a value of NA, although the true value is well defined. Instead the stored value "reject" is used. Operationally "reject" is the identity for PLUS, MINUS, AND, and OR. For other operations, if any operand has the value "reject" the result is also "reject." Hence, when adding 5 to "reject" the result is 5, and when testing whether or not 5 is less than "reject" the result is "reject."

VI. POINTER AND DATA STRUCTURE

The previous sections have defined Master Links from the user's point of view. To implement the features described, and to achieve the other stated goals (high performance in a time-sharing environment, portability, and multiple concurrent users), require a new approach to the layout of the data base elements onto the host systems files. In the classical approach to data-base design, records are used for many purposes. One purpose is to associate data values; another is retrieval efficiency: data values used together are stored together in a record. Update interlocking is a third use: exclusive control of a record or set of records is granted to a process so that it may make a series of changes to their contents without interference from other processes.

Master Links provides three distinct tools to achieve these three results, without having to rely on physical-storage records:

(*i*) Association of data items is accomplished by the pointer structure described in Section 6.2.1.

(*ii*) Retrieval efficiency is achieved by a parametrized layout of the data values into a data block, Section 6.2.2.

(*iii*) Multiple concurrent updates by many users is made possible by the concept of a lock unit, described in Section 6.1.

With Master Links, programs (and people) work with the logical structure of a data base, unhampered by its physical layout on the direct-access files. The details of record and file boundaries are invisible at the logical level. The basic concepts of Master Links, as well as all or most of the detail logic that implements the concepts, are independent of any machine or host system.

The mechanism used to achieve this freedom is the stream.

6.1 *Streams*

A *word* is an arbitrary unit of storage, the meaning of which is determined by the host system. A *stream* is a series of words. A particular word in a stream is identified by its position in the series. This ordinal number is called WIS (*w*ord *i*n *s*tream). The size of a stream in words may be increased or decreased to accomodate changes in data-base size. A data base is built from several streams. A stream therefore needs an identifier, which is designated S. A particular word of a data base is completely determined by S and WIS. A stream is made up of a series of records, where a record is defined as a set of words transmitted between primary and secondary storage by the host system as a single unit. A pointer into a stream is always in terms of WIS, never in terms of record number, or word in record, or any other host-system concept.

Exclusive use of a segment of a stream, called a lock unit, is required for updates. In fact, the lock unit may be a record, a set of records, a file, etc., depending on the capabilities of the host file-management system. The interlocking of multiple concurrent updates anywhere in the data base occurs correctly regardless of the boundaries of the lock units. A lock unit is, from the traffic point of view, a resource. It is important that as few lock units as possible be locked for the shortest time possible in servicing an update, and that a lock unit cover the smallest possible area.

One can plan efficient use of streams in terms of S and WIS alone. The probability that two words, WIS and WIS + K, of the same stream are in the same record is 1 for $K = 0$ and linearly decreases to zero with the magnitude of K. That is, words close together in the stream are likely to be in the same record. The same is true of two words and a lock unit. Thus, by adopting a probabilistic viewpoint, efficient use of streams can be planned without a detailed knowledge of file and record boundaries.

Streams are implemented in the Master Links software using direct-access files. Catalogued, direct-access files with a fixed number of unformatted, fixed-length records are used because such files are generally available and operate efficiently on existing time-sharing systems. This is the simplest and most commonly available type of direct-access file available today. A *file set* is a (possibly null) series of direct access files. Each file of the series has the same dimensions: RPF records per file, and WPR words per record. A file of a file set is identified by its ordinal position in the series; this number is called FIFS (*f*ile *i*n *f*ile *s*et). A file set forms a series of computer words when the files are viewed as logically concatenated in the order of their FIFS numbers, with the records of each file being logically concatenated in the order of their record numbers. Thus, graphically, a file set can be pictured as follows:

1	2	···	RPF	1	2	···	RPF	···	1	2	···	RPF
← FIFS = 1 →				← FIFS = 2 →				···	← FIFS = K →			

Each box represents a record; its record number is shown inside the box. This construction does not imply that the files, or even the records of a file, be physically concatenated on secondary storage. The actual allocation of files upon direct-access devices is a responsibility of the host file-management system. A file set can grow, and the unit of growth is a file. A new file is assigned the next ordinal number available for the file set to which it is assigned. A file set is, therefore, a finite but extensible series of computer words, and hence is an implementation of a stream. Several file sets are used to implement the several streams of a data base.

To access word WIS in stream S, the file set for S is determined from S. Then the dimension RPF and WPR are determined. By integer division and modulo arithmetic on WIS, the FIFS, the record number, and the word in record are calculated. Thus the word is described in terms of files and records, and can be accessed.

6.2 *Pointer Structure*

This section describes the pointer structure of Master Links. The design derives from the following goals:

(*i*) The data structure must be designed for auxiliary storage.

(*ii*) Data may be updated and elements added to and deleted from the hierarchy by simple, efficient algorithms.

(*iii*) These operations serve multiple concurrent users.

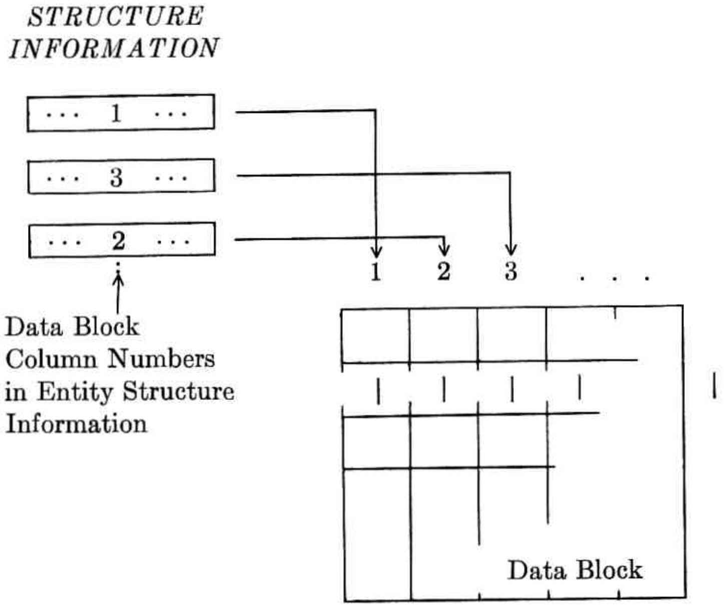
(*iv*) The integrity of the data structure must be maintained in the event of a machine failure.

(*v*) A single set of algorithms must access any hierarchical data base.

(*vi*) The storage of the hierarchy must provide efficient hierarchical traversal; that is, at any position in a hierarchy, the accessing routines must be able to directly address any subordinate or sibling.

6.2.1 *Development of the Pointer Structure*

In Section II, entities were described as having data and structure. Structure connects an entity to its relatives. In order to attain efficient traversal from an entity to any of its siblings or offspring, regardless of the number of offspring, the structure part of the entities in a family must be stored contiguously. Otherwise, a sequence of reads would be needed to follow the chain of sibling pointers through auxiliary storage. For families to grow in real time and still have their members contiguous, the growth process requires copying the old family description to some available space and then appending new entities. If data of an entity were stored with the structure, this copy process would become expensive and would leave large amounts of space vacant. Rather, the data are stored separately in a matrix, or data block. The columns of the matrix correspond to entities, the rows to fields. Therefore, the structure information for an entity must include a reference to the data-block column number assigned to that entity.



Column numbers, rather than absolute storage locations, are used to reference the data block, allowing separation of structure from data.

The hierarchical structure is completely divorced from the data storage structure. Whichever way the matrix is stored—by row, by column, by submatrix—has no bearing on the hierarchical structure information. The Master Links data-block storage arrangement is discussed in Section 6.3.

The offspring of an entity are linked to the entity by means of pointers which specify their storage locations in a stream. Every entity has one offspring pointer for each family of offspring. The collection of the data-block column number and the offspring pointers for one entity is called an *entity pointer set.*

data-block column number (one word)	pointer to an offspring family (one word)	pointer to another offspring family (one word)	. . .

An entity pointer set (EPS) contains the structure information for an entity in a contiguous stream of words.

Since each entity in a group has the same number of offspring families, the entity pointer sets for all entities in one group are the same size. Therefore, since siblings are adjacent, a pointer to the beginning of a family provides direct access to each member of the family. If the siblings were chained together, a null pointer in the chain would indicate the end of a family. However, with the siblings contiguous, the size of each family must be stored instead. It is most convenient to store the family size just before its first entity pointer set.

Family Size (n)	EPS_1	EPS_2	. . .	EPS_n

A family of n entities is described by a one-word family size, n, followed by n entity pointer sets (EPS), one for each entity in the family.

The entity index is the ordinal position of the entity pointer set in the physical pointer structure for one family. In deleting an entity from a family, it is important not to change the entity index of other entities in the family. Therefore, a special flag is encoded in the entity pointer set to show that the entity is deleted. Entities which are physically present in a stream but which have been flagged as deleted have *reserved* status. They are not considered present in the hierarchy. If the delete flag of a reserved entity is later turned off, the entity becomes *active* and is then treated as part of the hierarchy.

All the family pointer sets of a group are stored in a stream. This stream is called the *master link* of the group.

In summary, the description of the entities in a group is stored in a stream of words. A group is made up of families. A family is described by a family size, followed by a set of pointers for each entity in the family. An entity pointer set consists of a data pointer and a collection of offspring pointers. The data pointer is a column number of the data block for the group. The offspring pointers specify word positions of the appropriate streams. Entities allocated in the pointer structure are either active or reserved; the latter do not appear in the logical structure of the data base.

6.2.2 *Pointer Structure Algorithms*

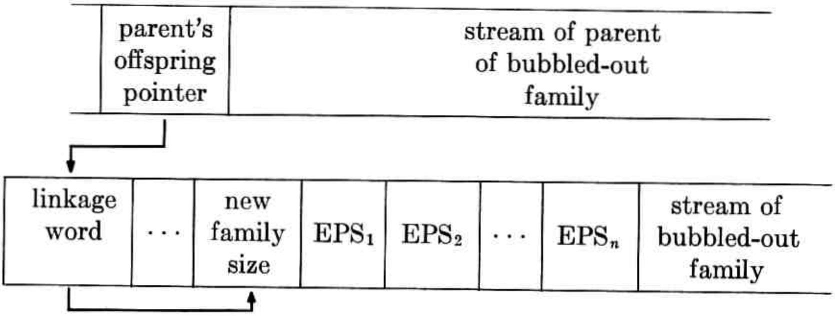
The algorithms which modify the data-base structure must be safe over abnormal termination of the process. A process can be abnormally terminated in many ways, such as by a user interrupt, a hardware or software failure in the host environment, or an intentional stop by the host system for exceeding some resource allocation. The key to making a transaction safe over unexpected terminations is to first allocate any new space needed, then fill out the new space, and finally link the new space with the old by a single pointer. If the write of that pointer succeeds, the new information is secure. If it fails, the area remains disconnected and wasted, but the data structure remains intact.

The algorithms must also work correctly when several concurrent users are trying to execute them. This is assured by locking a word to be updated (the lock unit must cover the entire physical record containing that word), reading the record to obtain a fresh copy, updating the value and any other values in the same record, re-writing the record, and then relinquishing the lock.

There are three functions which modify the pointer structure. An entity's status can be reversed (from active to reserved or back again); new entities can be added to an existing family; and a family can be created and linked to its parent.

To reverse an entity's status, the stream location of the first word of the entity pointer set is computed. The process then requests of the host environment exclusive control of the lock unit containing this word. When exclusive control is authorized, the record is read, the required word is updated, the record is written back to auxiliary storage, and the exclusive control is relinquished. If the process is terminated before the write, it can be re-executed because nothing has been altered. If it is terminated after the write, a restart procedure can read the record to determine that the update was successful, and skip re-doing it.

Extra entities are added to a family by first locking the record which contains the word pointed to by the parent's offspring pointer. This word, called the *linkage word* of the family, is the single word to be made a pointer to the new space. The first time that this algorithm is applied to a given family the linkage word contains the family size. Next, a sequence of contiguous words in the stream is allocated and locked. The existing entity pointer sets of the family are copied into the new space; new entity pointer sets are constructed by generating parent pointers, new data column numbers, and null offspring pointers; any of the new entities that are to be reserved for future assignment are marked reserved; and, finally, the new family size is placed into the first word of the new space. After this new area has been written and unlocked, the linkage word is updated to point to the new area. The record containing the linkage word is then written and unlocked. This update of a single word links the new family pointer set to the existing hierarchy and disconnects the old family description from it. The whole process is called a *bubble-out*; the wasted space containing the old family description is called a *bubble.*


parent's offspring pointer
stream of parent of bubbled-out family
linkage word
. . .
new family size
EPS$_1$
EPS$_2$
. . .
EPS$_n$
stream of bubbled-out family

The linkage word is a single word connecting the new family description to the previously existing structure.

Notice that the linkage word had to be locked from the start to the finish to keep other users from adding to the same family at the same time. Such interference could cause the family update to be lost entirely.

Creating a new family adds one complication to the bubble-out algorithm. Here, there is no linkage word, so the parent's offspring pointer must be locked and re-read instead. If the parent's offspring pointer to this group is still null after locking and re-reading, it is updated to point to the new family. If non-null, another user has in the meantime attached a family to the parent, so either the status-change or the bubble-out algorithm is entered.

6.2.3 *Further Considerations*

There are several considerations which affect the performance of the bubble-out algorithm. Among these are the handling of available space, the disposition of bubbles, and the use of an entity reservation factor for assigning sets of entities at one time. These considerations involve important balances between execution speed and auxiliary-storage space.

For each master link, the WIS for the next available word of the stream is stored as a header to the master link. The bubble-out algorithm first locks the header, then the linkage word. The new family size is calculated and the header is updated, written, and unlocked. Then each record to be written is locked, written, and unlocked before the next one is locked. Since the linkage word is already in use it must lie somewhere between the header and the available space. Hence, the locations locked are within one stream, and in numerically increasing order of WIS. This precludes any chance of a deadlock since a stream is stored in an ordered set of records. As for the bubbles, a utility can be run in the background from time to time to remove them. The bubble-out algorithm assigns column numbers to the new entities, so it must update an available-data-block-column word at the same time it updates the stream available-space word. Hence, the appropriate place to store this available-column number is the master link header.

A parameter of the bubble-out process allows the reservation of extra (inactive) entities. Assigning the extra entities causes only one bubble-out for all of the entities created, releasing only one bubble. The status-change algorithm is considerably more efficient than the bubble-out algorithm, so the average entity creation cost is reduced. The bubble-out algorithm assigns data-column numbers to the new entities, so data for the entities of a family are stored in sets of contiguous columns. As explained in the next section, this usually makes data accessing more efficient than if the columns of a family were scattered throughout the block. The cost of reservations is the cost of carrying the extra entities in storage before they are activated. The reservation factor can specify a constant increase or a growth factor as a percentage of the current family size.

6.3 *Data Blocks*

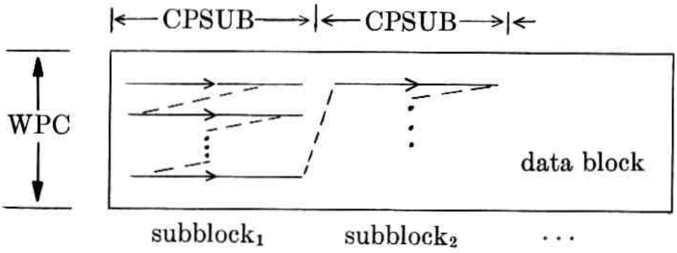
Data-base processes have a strong tendency to access either values of many fields from a few entities, or of a few fields from many entities. In the latter case the entities tend to be requested in an order determined by the hierarchy of the data base. A given data base will have a mix of these two types. If the first type predominates, it is efficient to order the values column-wise. If the second is more common, efficiency is gained by arranging the values row-wise, and by assuring that entities processed together occupy, with high probability, adjacent columns.

For Master Links, the mechanism for storing values is the *data block*. A data block is a matrix of values with one column for each entity, and one row for each field. An element of this matrix is one value, which takes up one or more words. The number of words needed to store one value is a parameter of the field, called its *size*. Thus all the values of one row are of the same size, but the values in two different rows may have different sizes. For each group of a data base there is one data block. The arrangement of a block into records is controlled by several block parameters which are attributes of the corresponding group. These parameters provide a variety of possible structures, of which the column-wise and row-wise layouts are special cases. Using the block parameters as inputs, a single algorithm can access any block arrangement.

6.3.1 *Layout of a Data Block*

A data block is stored in one stream. The block has two parameters, words per column, WPC, and columns per subblock, CPSUB, which are used to divide the block into subblocks. WPC is an integer equal to the sum of the sizes of the fields. This is the vertical dimension, in words, of the block, and also of the subblocks. The parameter CPSUB defines the horizontal dimension of a subblock. The first subblock consists of columns 1, · · ·, CPSUB; the second is columns CPSUB + 1, · · ·, 2·CPSUB; etc. A block is then a horizontal concatenation of subblocks.

The ordering of words in a block is established by keeping the words of a multiword value together, and arranging the values in row-wise order within a subblock, and then concatenating the subblocks from left to right. This is the ordering used to store a block in a stream. The order of values in a block is illustrated below. Each solid arrow indicates contiguous storage of CPSUB values.



It should be noted that the subblock plays no further role except to establish an ordering. It does not correspond to a file, a record, or any other host-system concept. In fact, the mapping of block words onto stream words is performed without concern for record or file boundaries.

6.3.2 *Example of Block to Stream Mapping*

Consider a block with three rows and five columns, where SIZE (words per value) is set to 1, 2, and 1 for rows 1, 2, and 3, respectively. Suppose a numeric value takes one word, and character value takes one word for every four characters. Then a sample of the block looks like:

1.00	2.00	3.00
AAAAAAAA	BBBBBBBB	CCCCCCCC
10	20	30

4.00	5.00
DDDDDDDD	EEEEEEEE
40	50

Words per column (WPC) is four. The mapping of this block onto a stream is shown below for three different values of CPSUB.

CPSUB = 1:

1.00	AAAA	AAAA	10	2.00	BBBB	BBBB	20	3.00

CCCC	CCCC	30	4.00	DDDD	DDDD	40	5.00

EEEE	EEEE	50

CPSUB = 5:

1.00	2.00	3.00	4.00	5.00	AAAA	AAAA	BBBB	BBBB

CCCC	CCCC	DDDD	DDDD	EEEE	EEEE	10

20	30	40	50

CPSUB = 3:

1.00	2.00	3.00	AAAA	AAAA	BBBB	BBBB	CCCC

CCCC	10	20	30	4.00	5.00		DDDD	DDDD

EEEE	EEEE			40	50	

6.3.3 *Accessing a Value from a Data Block*

For the *n*th row of a block, SIZE_n is the number of words for one value in the row, and DIC_n is the *d*isplacement *i*n *c*olumn of the row which is defined as one plus the sum of the SIZE's of the first $n-1$ rows. Thus for the example in Section 6.3.2.:

n	*SIZE* (words)	*DIC* (words)
1	1	1
2	2	2 $(= 1 + \text{SIZE}_1)$
3	1	4 $(= 1 + \text{SIZE}_1 + \text{SIZE}_2)$

The inputs to the algorithm for accessing a value of a data block are the identifier of a field, and a column number, COLNO. DIC, SIZE, and the group of the field can be determined since they are attributes of the field. WPC, CPSUB, and the stream identifier, S, are then determined since they are attributes of the group. From these the following calculations are made using integer arithmetic:

$$\begin{aligned}
\text{SUBBLKS} &= \text{COLNO/CPSUB}\\
\text{WPSUB} &= \text{WPC}\cdot\text{CPSUB}\\
\text{WORDSABOVE} &= \text{CPSUB}\cdot(\text{DIC-1})\\
\text{WORDSLEFT} &= ((\text{COLNO-1})\text{modulo CPSUB})\cdot\text{SIZE}\\
\text{WIS} &= 1 + \text{SUBBLKS}\cdot\text{WPSUB} + \text{WORDSABOVE}\\
&\quad + \text{WORDSLEFT}
\end{aligned}$$

SUBBLKS is the number of subblocks previous to the subblock containing the sought value. WPSUB is the words per subblock. WORDSABOVE is the number of words above the sought value in its subblock. WORDSLEFT is the number of words to the left of the sought value in its row of the subblock. WIS is the word in stream of the first word of the sought value. Hence S and WIS and SIZE are known. These are the inputs needed to access a stream, as described in Section 6.1.

6.3.4 *Row-wise, Column-wise, and Intermediate Layout*

Note that when CPSUB equals 1 the order of storage is column-wise. When CPSUB equals the words per record, storage is row-wise. An intermediate setting of CPSUB between 1 and WPR will for certain usage patterns achieve performance superior to either column-wise or row-wise organizations. This is illustrated in the following example. Suppose that a block has 100 rows and 100 columns. Suppose that process R uses all the data in one row, and that process C uses all the data in one column, and that these processes are run equally often. Suppose also that WPR = 100. Then if CPSUB = 1, C must read one record, R must read 100 records. The average records per process run is 50.5. If CPSUB = 100, C must read 100 records whereas R reads only one, for the same average. If CPSUB = 10 the block is divided into a 10 × 10 checkerboard of records. Each process must read 10 records for an average of 10 records per process. This is the optimum CPSUB for this example.

A utility called CONVERT can be used to change a block from one value of CPSUB to another. Modifying CPSUB adjusts the data base to reflect a changed or unpredicted pattern of usage. It also makes possible periodic changing of the data layout to conform to a cyclic pattern of usage. All programs accessing a data block do so in terms of column numbers and fields. The assignment of a value to a block, row, or column is unchanged by CONVERT, and hence no program is invalidated.

6.4 *Review of the Advantages of Streams*

The process of design involves constructing transformations to achieve a desired structure using available structures as media. The desired structures for Master Links are a hierarchy and data blocks. The transformation is carried out in two steps, from direct-access files into streams and from streams into blocks and pointer sets. The structures and their attributes are summarized in this table:

STRUCTURE	*ATTRIBUTES*
Catalogued Direct-Access Files	Internal Identity (FSNO, FIFS) NAME Records Per File (RPF) Words Per Record (WPR)
Streams	Stream Identity (S) Word In Stream (WIS)
Blocks	Block Identity (B) Words Per Column (WPC) Columns Per Subblock (CPSUB) ROW *n* SIZE for ROW *n* Displacement In Column (DIC) For ROW *n* Column Number (COLNO)

It is no accident that streams, the intermediate structure, are so simple. They amount to an idealized direct access media. The advantage of using this intermediate structure is that it crystalizes the separation of the Master Links structures from the physical-storage media. The programs that implement the desired structure are coded independent of the actual direct-access media. In particular, the parametrized layout of a block would be very cumbersome to implement directly in terms of files, records, and word in record. It is very straightforward in terms of word in stream.

Since the Master Links structure is separated from the physical media, media management utilities such as CONVERT can be run without altering any Master Links programs. The separation of structure from media also makes possible the implementation of alternative media. Streams might be implemented as arrays in primary storage for small data bases, or implemented in an entirely different manner upon direct access files, such as with all streams in one extensible file. Finally, this separation enhances the portability of Master Links allowing most of the logic of the system to be based on a machine-independent direct-access structure.

VII. EXPERIENCE AND FUTURE EXTENSIONS

An experimental version of Master Links was operational in 1970. It was based on the concepts and supported all the features reported in this article, except portability and certain utilities. A production version was completed in May 1972. It supports all features, including portability, all utilities, two different stream implementations, plus improved performance. These versions have been used for a variety of different types of projects: inventory, financial, budget and resource allocation, and construction program administration data bases. Together with the Natural Dialogue System[1] it forms the basis of the Off-The-Shelf-System.[2]

Several efforts are under way to extend and improve the system:

(*i*) Networks—allowing a group to have more than one parent.

(*ii*) Field length data—allowing strings of data, such as a time series of values or a paragraph of text, to be stored efficiently as a single value.

(*iii*) Function evaluation—computing in parallel all requested level raises that are defined over a common subtree. Hence, in "total IN STOCK divided by total ON ORDER," the numerator and denominator totals will be taken simultaneously in a single pass over the item entities of a department.

(*iv*) Access tree generator—allowing execution-time determination of the hierarchy. Suppose, for instance, that a new item field, item class, describes the level of supervision required to approve acquisition of the item. Then "total IN STOCK by item class" is a meaningful function, but the hierarchy formed by partitioning item entities according to their values of item class must be computed at execution time, if the "by" field is allowed to be arbitrarily specified by the user.

(*v*) Report generator—accepting a description of the content and layout of a report and on request producing an instance of the report.

REFERENCES

1. Puerling, B. W., and Roberto, J. T., "The Natural Dialogue System," B.S.T.J., this issue, pp. 1725–1741.
2. Heindel, L. E., and Roberto, J. T., "The Off-The-Shelf-System—A Packaged Information Management System," B.S.T.J., this issue, pp. 1743–1763.

Copyright © 1973 American Telephone and Telegraph Company
THE BELL SYSTEM TECHNICAL JOURNAL
Vol. 52, No. 10, December, 1973
Printed in U.S.A.

Information Management System:

The Natural Dialogue System

By B. W. PUERLING and J. T. ROBERTO

(Manuscript received October 5, 1972)

The Natural Dialogue System (NDS) is a software system designed to permit the easy implementation of time-shared computer programs which employ sophisticated forms of man-machine dialogue to converse with members of a nonprogrammer user audience. The heart of the system is a syntax-directed translator which recognizes user input messages and translates them into an internal text of integers for use by the program. NDS allows the language designer to specify the syntax of the statements in his language, the form of their translations, methods for diagnosing errors in user's input, diagnostic messages to be generated, and the style of dialogue which will exist between the programs and their users. This is accomplished through a dialogue description and a language description consisting of syntactic specification elements with semantic procedures embedded within them. Use of NDS allows the language designer to produce an interactive language which is tailor-made for both his users and his programs. NDS relieves the language designer of the necessity of writing a complex message analyzer, thereby substantially reducing the effort required to produce systems that offer these forms of man-machine dialogue. Furthermore, use of NDS allows such systems to be implemented by less sophisticated programming talent than would otherwise be necessary.

I. INTRODUCTION

The Natural Dialogue System (NDS) is a tool to aid programmers in the implementation of time-sharing-based computer systems which employ keyword-oriented languages and a variety of styles of man-machine dialogue to converse with members of a nonprogrammer user audience. By keyword-oriented languages we mean languages of the type illustrated by Sinowitz[1] and suggested as an alternative to natural language for communicating with information management systems by Chai.[2] NDS provides the designer of such a language with the ability to define the syntax of the statements in the language, the forms of their translations, methods for detecting errors made by users of the language, and diagnostic messages to be sent to users when errors are detected. In addition, NDS provides facilities for the language designer to specify the style of dialogue which will exist between the system and its users.

NDS has been operational on an experimental basis since 1970. It has been implemented under five different host operating systems (including one batch system). Its primary use has been in the area of interactive query languages for information management systems, including inventory management systems, a budget control system,[3] a work force administration system, information retrieval systems based on surveys of financial and equipment data, and a general-purpose hierarchical data base management system.[4] Other uses have included a data checking specification language, a report generator composition language, and bulk data input/output format specification languages. NDS has also led to further work in the area of tools for interactive language design presented by Heindel and Roberto.[5]

In order to get a feel for what can be done using NDS, some concepts concerning styles of man-machine dialogue are presented, followed by a description of the styles of dialogue obtainable using NDS. NDS itself is then described, including an overview of the modules of the system and certain details concerning the specification of systems to be implemented using it.

II. DIALOGUE CONCEPTS

In making conversational software systems available to non-programming audiences for purposes of information retrieval and problem solving in general, a broad spectrum of conversational styles has evolved. At the extreme ends of this spectrum we have machine-initiated dialogue and user-initiated dialogue. In the machine-initiated style, the user is asked questions by the computer. These questions are designed to find out, in an orderly way, what the user wishes the system to do for him. In the user-initiated style, the user presents his problem to the system and directs its action. The two styles are best illustrated by example.

(*i*) Machine-initiated dialogue.

Computer: WHAT IS THE VALUE OF EXPENSE?
User: 50000

Computer: DO YOU WANT THE VALUE OF PROFIT?
User: YES
Computer: PROFIT IS 40000

(*ii*) User-initiated dialogue.
User: EXPENSE IS 50000. WHAT IS PROFIT?
Computer: PROFIT IS 40000

Coupled with these different styles of dialogue is the problem of conversational dynamics where at some point in time during the conversation the subservient participant wishes to seize the initiative. For example:

(*i*) User seizes initiative from computer.
Computer: WHAT IS EXPENSE?
User: IGNORE EXPENSE. REVENUE IS 80000.

(*ii*) Computer seizes initiative from user.
User: TAX IS 5%. WHAT IS PROFIT?
Computer: PROFIT CANNOT BE COMPUTED YET, WHAT IS EXPENSE?
User: 50000

With either of these styles, a user is apt to input information which is syntactically or semantically incorrect. Input of this nature should not cause the conversational program to abort. On detecting invalid input a conversational program may output terse messages such as "WHAT?" or "SYNTAX ERROR" and then invite re-entry of the test in question. Alternatively, programs may output lengthy explanations of valid replies and again ask the user to continue his input. The nature of handling invalid input depends primarily on the experience level of the end-users as well as the experience level of the person implementing the conversational software. In general, a language designer should have the tools at his disposal to tailor his language, including handling of invalid input, to correspond precisely to the environment in which it will be used.

In general, a transaction between man and machine can be viewed as a consulting effort between two "experts":

(*i*) the machine, which is an expert in delivering facts or computing results based on input data, and

(*ii*) the person, who is an expert in the problem to be solved, the environment in which the problem arose, and certain subjective considerations of the possible solutions.

In this situation if the person is burdened by certain conversational constraints, the creative, exploratory environment may suffer. In other words, the conversation between man and machine should be made as natural as possible for the person. With this in mind, NDS offers a variety of dialogue styles which encompass most of the initiative spectrum with the emphasis on the user-initiated end.

III. STYLE OF NDS DIALOGUE

A user's message or request to a system using NDS consists of a series of statements separated by colons. Each statement in the language consists of a unique keyword followed by a sequence of characters called the clause of the statement. A message is ended by the statement GO:. A message to an information management system might be:

> PRINT ITEM NUMBER, SELLING PRICE/PURCHASE COST, REORDER DATE: WHEN AMOUNT IN STOCK > 1000: IN ALL DEPARTMENTS: GO:

This message consists of three statements with the keywords PRINT, WHEN, and IN respectively. The PRINT statement fills the same role as a verb in the English language since it directs the information system to print the information specified in its clause. In general, a user's message must contain a single verb statement. The WHEN and IN statements act as modifiers (adverbial or prepositional) of the PRINT verb. In general, a user's message contains zero, one, or more modifier statements. The statements in a message can be given in any order since NDS does not consider a message to be complete until the GO statement is encountered.

An important feature which NDS offers is the automatic edit mode. NDS remembers the state of the dialogue from message to message. Once a statement is correctly given by a user, that statement remains as part of the "current" message until the user deletes it, or replaces it. Therefore, after the system acts on the above request, the user may continue the dialogue by typing:

> WHEN AMOUNT IN STOCK > 2000: GO:

The PRINT and IN statements which were given as part of the first message are carried over as part of the second message. Therefore, to NDS the second message becomes:

> PRINT ITEM NUMBER, SELLING PRICE/PURCHASE COST, REORDER DATE: WHEN AMOUNT IN STOCK > 2000: IN ALL DEPARTMENTS: GO:

Once a verb statement is correctly given by a user, that statement remains as part of the current message until the user deletes it, replaces it, or enters the statement for a different verb in the language. Thus, following action on the second message, the user may continue his dialogue by typing:

DISTRIBUTE ITEMS BY SELLING PRICE: GO:

The IN statement from the first message and the WHEN statement given in the second message are carried over as part of the third message. If the user continues his dialogue by inputting a statement whose clause has an invalid construct according to the definition of the clause given by the language designer, the system will print a diagnostic message (possibly language designer defined) and remove the statement from the current state of the dialogue.

In addition to verbs and modifiers, a language may contain one or more special statements termed dialogue service statements. These statements usually take the form of aids to the user of a language or debugging tools for the language designer. Services may be included which provide the user with explanations of terms used in the language, news of recent changes to the application, instructions on the use of the language, the ability to change the initiative of the dialogue, or any other facilities which the language designer deems appropriate. For himself, the language designer may include statements which provide dumps or activate traces or timings within his programs.

Through the semantic facilities provided by NDS, a language designer is capable of detecting syntactic or semantic errors in a user's input, informing him of the error, and then allowing him to correct just that part of the text in question. Using this approach a typical interaction might be:

```
    User: PRINT STORE NAME, EARMINGS: WHEN
      DEPRECIATION > 40%: GO:
Computer: 'EARMINGS' IS A MISPELLED NAME, REENTER.
    User: EARNINGS
Computer: DEPRECIATION CANNOT EXCEED 25%,
      REENTER.
    User: 20%
              ⋮
```

Note that the user need correct only that part of the text which is incorrect.

For certain applications or for certain user experience levels, computer-initiated dialogue is a meaningful style of man-machine interaction. A language designer may implement this style of dialogue using the semantic facilities of NDS. The user must initiate the change in the style of the dialogue through a statement in the language. From that point in time, the machine may have the initiative and may interact with the user in a question and answer style illustrated by the following example:

User: HELP:
Computer: WHAT DO YOU WANT TO DO? (PRINT, RANK,
PLOT)
User: PRINT
Computer: WHAT INFORMATION DO YOU WANT
PRINTED?
User: EARNINGS
Computer: FOR WHICH STORES?
User: BUFFALO, SYRACUSE
⋮

Thus, a wide variety of dialogue styles is obtainable using NDS. The system itself is now described.

IV. AN OVERVIEW OF NDS

As illustrated in Fig. 1, NDS consists of two phases, a setup phase and an execution phase. These two phases interface with two different audiences, a language designer and the set of end-users of the language designer's system.

The language designer prepares a description of his language and dialogue style (details of which will be described later) to be presented to the setup phase of NDS. The setup phase translates these descriptions from a form suitable to the programmer to a form suitable to the execution phase of NDS. These translations are written by the setup phase onto a set of language analysis and dialogue monitor driving files for later access by the execution phase. The language designer also prepares a set of program modules containing programs to perform the tasks corresponding to the verbs and dialogue service statements in the language. At appropriate times during the dialogue, the execution phase of NDS will pass control to these program modules to perform the appropriate tasks.

Users communicate with the system through the execution phase of NDS. The execution phase consists of a dialogue monitor, a language analysis module, and a set of "built-in" dialogue service functions which are accessible to all languages. The dialogue monitor accepts

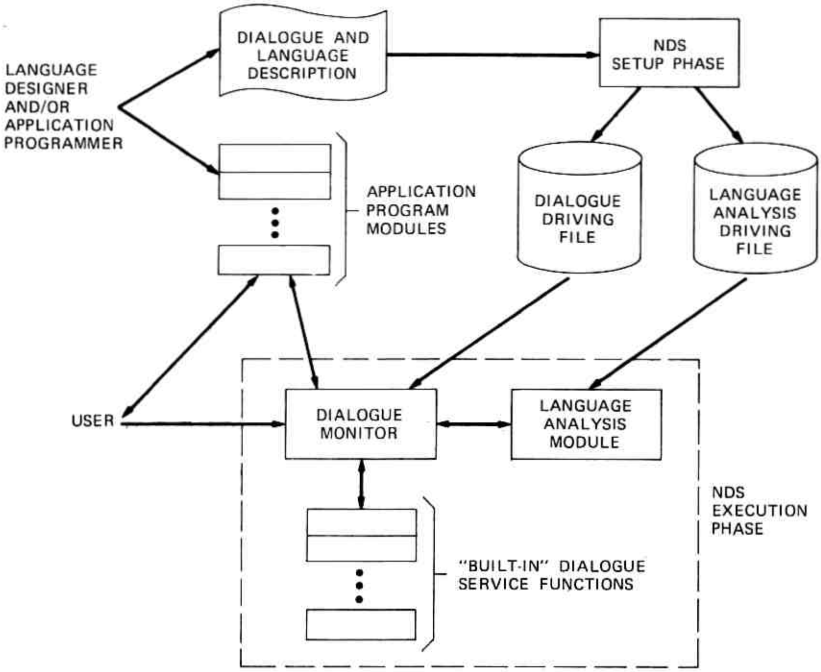

DIALOGUE AND LANGUAGE DESCRIPTION
NDS SETUP PHASE
LANGUAGE DESIGNER AND/OR APPLICATION PROGRAMMER
APPLICATION PROGRAM MODULES
DIALOGUE DRIVING FILE
LANGUAGE ANALYSIS DRIVING FILE
USER
DIALOGUE MONITOR
LANGUAGE ANALYSIS MODULE
NDS EXECUTION PHASE
"BUILT-IN" DIALOGUE SERVICE FUNCTIONS

Fig. 1—An overview of NDS.

input from the user of the system in the form of a series of statements, each beginning with a keyword and ending with the character colon (:). The dialogue monitor breaks out the clause of each statement and passes it to the language analysis module together with the clause description as specified by the language designer. The language analysis module attempts to parse and translate the clause into an internal text of integers as specified by the clause description. The algorithm employed by the language analysis module is an extension of the top-down left-to-right algorithm given by Cheatham and Sattley.[6] Successful translations returned by the language analysis module to the dialogue monitor are placed in translation space for later access and a record is kept regarding which statements are currently active in the dialogue.

The GO statement is used to indicate that the user's message to the system is complete. When it is encountered, NDS makes a series of checks, called GO-analysis, which insure that any interstatement relationships declared by the language designer have been fulfilled. There are really two kinds of GO-analysis. One occurs when the current verb is different from the last verb which was successfully executed. The system checks that a current verb exists, that all required modifiers of the current verb are active, and that no modifier (required or optional) of the current verb is inactive because it was incorrectly given since the last time the GO statement was encountered. The other type of GO-analysis occurs when the current verb is the same as the last verb which was successfully executed. The system makes the same checks described above, but also checks that at least one modifier of the current verb (required or optional) has been correctly given since the verb was last executed. If other interstatement relationships exist, facilities are provided for the language designer to specify additional checks to be executed as part of GO-analysis. If GO-analysis is successful, NDS passes control to the program module corresponding to the current active verb. When execution of the module is complete, control is returned to the dialogue monitor and the dialogue with the user is resumed.

When the dialogue monitor recognizes a dialogue service statement in a user's message, control is passed to the appropriate program module immediately. When execution of the module is complete, control is returned to NDS and the dialogue continues.

NDS provides a set of "built-in" dialogue service statements to provide services common to all languages. These include:

STOP	disconnect the user from the system
RETURN	return control to the host operating system
DELETE	remove rather than replace a currently active statement
CLEAR	remove all currently active statements
RECAP	print out to the user all currently active statements
DETAIL	cause automatic recapping of the current message when the GO statement is recognized
VOCABULARY	print out to the user a list of keywords and their synonyms for all statements in the language
INPUT	direct NDS to take its input from a previously prepared character sequential file
DUMP	print out the translation of a currently active statement or all currently active statements

The language designer may redefine any of these "built-in" dialogue service statements by including in his language a dialogue service statement of the same name and providing a corresponding program to perform the function he desires. The corresponding "built-in" statement is then not accessible to users of the language.

V. SYSTEM SPECIFICATION

In order to create a system using NDS, the language designer must supply a dialogue description and a language description to the NDS setup phase and prepare a set of programs to be called by the NDS dialogue monitor to perform whatever tasks may be requested by his users. The general forms of these specifications are now described.

5.1 *Dialogue Description*

The dialogue description of a language written using NDS consists of a descriptor for each statement in the language. Each descriptor consists of a series of attributes which are to be assigned to the statement. These attributes define certain properties of the statement and may define certain relationships between the statement being described and other statements in the language. In general, the attributes of a statement are the statement identifier, statement keyword(s), clause syntactic type, translation allocator, verb indicator, required and optional modifier specifications, additional GO-analysis checks, dialogue service indicator, program control information, and Polish indicator.

The statement identifier is a unique positive integer which can be thought of as the internal identifier of the statement. This number is used as a key to locate, store, and interrogate the translation of the statement in translation space. The statement keyword, a unique character string containing no blank characters, is the external identifier of the statement. NDS allows a statement to have an arbitrary number of keyword synonyms which again must be unique for the entire language.

If a statement in a language is to have a clause following its keyword, then the language designer must specify a clause syntactic type as part of the descriptor for that statement. The clause syntactic type is the link between the dialogue descriptor of the statement and that part of the language description which defines the syntax and semantics of its clause. A statement having a clause syntactic type as an attribute must also have a translation allocator. The translation allocator is used to specify the length of the largest possible translation of the clause of the statement. If the user inputs a statement whose translation size exceeds that indicated by its translation allocator, the dialogue monitor will output a standard message indicating that the statement is too long.

If a statement in the language is to be recognized as a verb, then a verb indicator must be specified as one of the attributes of the statement. A verb statement may require other statements to be present in the current message when the user types the GO statement. These are called required modifiers. If a verb requires other statements to be present, the language designer specifies these statements according to their respective statement identifiers as part of the verb's descriptor. Statements which are not required in the same message with a verb, but which somehow change the meaning or action of the verb, and may therefore be thought of as optional modifiers, are specified in an identical fashion. If other, more complex relationships are to exist between a verb and other statements in the language, facilities are available for the language designer to include, in the descriptor of the verb, checks of these relationships to be performed as part of GO-analysis.

If a statement is to be recognized as a dialogue service statement, a dialogue service indicator must be part of its descriptor. A dialogue service statement may have a clause, in which case a clause syntactic type and translation allocator must be given as part of its descriptor. For both verbs and dialogue service statements, program control information must be specified as an attribute in the descriptor of the statement. This information identifies the program module which contains the program corresponding to the verb or dialogue service statement being described.

The Polish indicator is used to specify that the clause of a statement consists of a function containing operators and operands and that the language designer has followed certain rules in defining the clause syntactic type of the statement. When a successful translation of a statement with the Polish indicator is returned to the dialogue monitor, it will be converted to early-operator Polish postfix notation[7] before being stored in translation space.

5.2 *Language Description*

The clause descriptions for the statements in a language are written in a language specification meta-language. This meta-language is really a combination of two distinct languages: a descriptive language which is used to describe the syntax or structure of a clause and, embedded in it, a procedural language, called the Natural Dialogue Semantic Programming Language (NDSPL), which is used to specify context-dependent syntax checks, modifications to the normal clause translations, diagnostic messages, error correction methods, and changes in the initiative of the dialogue.

In the descriptive meta-language, a syntactic type is indicated by a name surrounded by $\langle$ and $\rangle$. A syntactic-type definition consists of a syntactic type followed by an equal sign (read as "is defined as") followed by a sequence of language specification elements which define the syntactic type. The language specification elements are members of the following set:

$\langle \cdots \rangle$ syntactic type

$|$ exclusive or (alternation indicator)

& and (conjunction indicator) used to indicate that a portion of a clause is to be parsed and translated in more than one way

$[\cdots](i, j)$ a repeating group of specification elements to be repeated at least i times but not more than j times, $j \geqq i \geqq 0$, $j > 0$, $i = j = 1$ if the parenthesized pair is omitted

$'\ldots'$ a semantic procedure type consisting of one or more NDSPL statements enclosed in primes

"$\ldots$"(t, p) a non-null terminal character string enclosed in quotes, called a literal, with its translation number t and, if the literal is to act as an operator, its precedence p, null translation if the parenthesized pair is omitted

S any member of the terminal class string, consisting of all non-null character strings

N any member of the terminal class number, consisting of all numbers, signed or unsigned, with or without a decimal point

V any member of a language-designer-defined terminal class

The complete specification of a language consists of a syntactic-type definition for each of the clause syntactic types associated with a statement in the dialogue description part of the system description. This specification provides instructions to the language analysis module of the dialogue monitor to parse and translate user input statements. Once the dialogue monitor has recognized a keyword in a user's input statement, it passes the clause following that keyword along with the clause syntactic type associated with the statement to the language analysis module for parsing and translation. The parser applies the clause syntactic-type definition on the input clause from left to right. If the parser encounters an element representing a terminal class (V, N, S, or literal), it must match an initial substring of the remaining input clause as a member of that terminal class and add to the translation of the clause appropriately. If the parser encounters a syntactic-type definition, it must apply it left to right on the remaining input clause. If the parser encounters a semantic procedure type, it must execute it. When the parser simultaneously encounters the end of the clause syntactic-type definition and the end of the user's input clause, the parse is successful and the completed translation of the clause is returned to the dialogue monitor to be placed in translation space.

Of the language specification elements available to a language designer using NDS, two deserve more detailed discussion: the terminal class V and the semantic procedure type. The terminal class V consists of a set of character strings defined by the language designer. Each member of the class is assigned a set of integer attributes. The occurrence of a V in a syntactic-type definition instructs the parser to match an initial substring of the remaining input clause with a member of this set of character strings, to append the value of its first attribute to the translation, and to make the values of its other attributes available for examination by succeeding semantic procedure types. The use of the terminal class V allows the language designer to specify the skeleton of a language where certain terminal class members must be chosen from a particular set. The composition of this set may then be changed without affecting the language specification.

The semantic procedure type is the means by which the procedural part of the meta-language is embedded within the descriptive part. It consists of one or more NDSPL statements surrounded by primes and can succeed or fail just as all other language specification elements can succeed or fail. The statements available in NDSPL are the following:

SET	arithmetic assignment statement
IF	control for conditional execution (similar to the logical IF statement in FORTRAN IV)
FOR, NEXT	iteration control statements
GOTO	unconditional transfer
STASH	add to the current translation
UNSTASH	remove from the current translation

PRINT print a message to the user

FAIL cause unconditional failure of a semantic procedure type

S&F arithmetic assignment and cause unconditional failure of a semantic procedure type

P&F print a message to the user and cause unconditional failure of a semantic procedure type

TEST cause conditional failure of a semantic procedure type

T&P cause conditional printing of a message to the user and failure of a semantic procedure type

READ cause a recursive call on the parser to ask the user for additional input and to parse it according to some syntactic-type definition

CALL cause control to be passed to a language-designer-provided own-code semantic procedure which may succeed or fail

The data which are available for manipulation in NDSPL include numeric constants; a set of language-designer-declared variables; the current translation; the attributes of the most recently matched members of V, N, and S; a set of variables provided by NDS which give a picture of the current state of the parse; and the current state of the dialogue. Messages printed to the user through NDSPL may include any of the above data plus constant character strings; the most recently matched members of V, N, and S; and the character string which the parser most recently attempted to match as a member of V, N, or S.

The semantic procedure type and the facilities of NDSPL give the language designer a powerful tool for creating an interactive language and style of dialogue which are tailored to both his end-user's needs and the needs of his programs. First of all, he has the ability to do context-dependent syntax checks by setting flags or saving the attributes of terminal class members at one point in the parse for later examination to determine what course the parse has taken or should take. He also has the ability to add to, delete from, or modify the normal clause translations using arithmetic functions of the available data. Thus, the translations of the user's input can be tailored to the needs of the application programs. The output facilities of NDSPL provide him with the means to supply his users with timely, relevant error diagnostics when errors are detected in their input statements. Moreover, the READ statement gives him the ability to seize the initiative in the dialogue. This can be used to ask for error corrections in mid-parse or to change the style of dialogue from user-initiated to computer-initiated.

5.3 *Programs*

Except for any own-code semantic procedures needed by the language, the only programs which must be supplied by the language designer are the programs for each verb and each dialogue service statement in the language. When a user issues a dialogue service statement or executes a verb using the GO statement, the NDS dialogue monitor uses the program control information given in the dialogue description to pass control to the proper program module. The program has access to translation space and to a set of NDS-provided variables which give a picture of the current state of the dialogue. An application program can be as simple or as complex as is necessary to perform the desired task. When execution of the module is complete, control is returned to NDS and the dialogue with the user is resumed.

5.4 *A Simple Illustrative Example*

Suppose that one wishes to implement an information retrieval language which allows users to do scatter plots of one variable versus another. The values for these variables are to come from a data base containing data for the years 1961 to 1973. The plot process is to be implemented as a verb, specifying the variables to be plotted, and one required modifier specifying a range of years for which data values are to be included in the plot. The specified variables must, of course, have numeric values. The user will be allowed to make requests such as

PLOT EMPLOYEES BY REVENUES: FOR 1965–1971: GO:

and

PLOT REVENUES BY EXPENSES: FOR 1961 THRU 1970: GO:

The computer program which has been written to carry out the plot request requires as input the internal numeric identifiers of the two variables and two numbers from one to thirteen, in increasing order, which specify the span of years to be included. The problem is to design a language to translate a user's request into the necessary computer program inputs insuring that a valid request has been made.

Suppose that the terminal class V contains, in part:

Symbol	*Attributes*	
	Varno	*Type*
YEAR	1	1
COMPANY	2	3
EMPLOYEES	3	1
REVENUES	4	2
EXPENSES	5	2

where the symbols are variable names and the attributes of a variable are its internal numeric identifier and its type (1 for integer, 2 for floating point, and 3 for character).

The specification of one possible language for communicating with the plot processor is given in the appendix. This language specification could result in a dialogue similar to the following (user input shown in lower case):

REQUEST⟩ plot employees by revenues: for 1965–1970: go:

The PLOT processor would produce a plot of variables 3 and 4 at level 2 for years 5–10.

REQUEST⟩ plot employees by company:

COMPANY IS A NON–NUMERIC VARIABLE ILLEGAL FOR PLOT

REQUEST⟩ plot revenues by expenses: go:

The PLOT processor would produce a plot of variables 4 and 5 at level 2 for years 5–10.

REQUEST⟩ for 1970 thru 1975:

1975 MUST BE A YEAR BETWEEN 1961 AND 1973 INCLUSIVE

REQUEST⟩ for 1968 thru 1962: go:

The PLOT processor would produce a plot of variables 4 and 5 at level 2 for years 2–8.

REQUEST⟩

⋮

VI. CONCLUSION

NDS provides a means for the easy implementation of time-sharing-based systems which employ a keyword style of man-machine dialogue.

Facilities are available to specify the syntax of the keyword clauses; the forms of their translations; timely, relevant error diagnostics; and a spectrum of dialogue styles, ranging from computer-initiated dialogue to user-initiated dialogue.

NDS has been used to produce a variety of application systems primarily in the area of interactive query languages for information management systems. Use of NDS substantially reduces the programming effort required to implement such systems; and moreover, the implementation may be done utilizing less sophisticated programming talent than would otherwise be necessary.

APPENDIX

```
ATTRIBUTES VARNO, TYPE
SCRATCH CELLS TEMP

NOTVAR = T "IS NOT VARIABLE NAME"
NONNUM = V "IS A NON-NUMERIC VARIABLE ILLEGAL
  FOR PLOT"
BADYR = T "MUST BE A YEAR BETWEEN 1961 AND 1973
  INCLUSIVE"

⟨PLT⟩ = ⟨NUMVAR⟩ "BY" ⟨NUMVAR⟩
*
* IF SYMBOL NOT IN TERMINAL CLASS V, PRINT NOTVAR
* IF SYMBOL IS A NON–NUMERIC VARIABLE, PRINT
*    NONUM
*
⟨NUMVAR⟩ = [V | 'P&F NOTVAR'] 'IF (TYPE = 3) P&F
    NONUM'
*
* IF YEARS NOT GIVEN IN INCREASING ORDER, REVERSE
*    THEM
*
⟨FOR⟩ = ⟨YR⟩ ["THRU" | "–"] ⟨YR⟩
         'IF (TRANS (1) ⟨= TRANS (2)) GOTO X; SET TEMP
         = TRANS (1); SET TRANS (1) = TRANS (2); SET
         TRANS (2) = TEMP; X'
*
* IF YEAR SPECIFIED IS NON–NUMERIC OR IF YEAR OUT
*    OF RANGE,
* PRINT THE ERROR MESSAGE BADYR
*
* IF VALID YEAR MATCHED AS N, REMOVE IT FROM
*     TRANSLATION AND
* SUBSTITUTE YEAR–1960 WHICH IS A NUMBER BETWEEN
*     1 AND 13
*
⟨YR⟩ = [N | 'P&F BADYR'] 'T&P VAL⟩ = 1960 ·AND· VAL
        ⟨= 1973, BADYR; UNSTASH 1; STASH VAL–1960'

STATEMENT 1: "PLOT": "P": VERB: REQ MOD 2: SYNTAX
  ⟨PLT⟩: MAX TRANS 3: PROGRAM GPLOTR:
STATEMENT 2: "FOR": "F": SYNTAX ⟨FOR⟩: MAX TRANS 2:
```

REFERENCES

1. Sinowitz, N. R., "DATAPLUS—A Language for Real-time Information Retrieval from a Hierarchical Data Base," AFIPS Conf. Proc,. *32*, 1968.
2. Chai, D. T., "Language Considerations for Information Management Systems," unpublished work.
3. Chai, D. T., "An Information Retrieval System Using Keyword Dialog," Inform. Stor. Retr., *9*, July 1973, pp. 373–387.
4. Heindel, L. E., and Roberto, J. T., "The Off-The-Shelf System—A Packaged Information Management System," B.S.T.J., this issue, pp. 1743–1763.
5. Heindel, L. E., and Roberto, J. T., "LANG-PAK—An Interactive Incremental Compiler-Compiler," Proc. ONLINE 72, Int. Conf. Interactive Computing, *1*.
6. Cheatham, T. E., and Sattley, K., "Syntax Directed Compiling," Proc. Eastern Joint Computer Conf. AFIPS, *25*, 1964, pp. 31–57.
7. Hamblin, C. L., "Translation to and from Polish Notation," Comput. J., *5*, October 1962, pp. 210–213.

Copyright © 1973 American Telephone and Telegraph Company
THE BELL SYSTEM TECHNICAL JOURNAL
Vol. 52, No. 10, December, 1973
Printed in U.S.A.

Information Management System:

The Off-The-Shelf System—A Packaged Information Management System

By L. E. HEINDEL and J. T. ROBERTO

(Manuscript received October 5, 1972)

The Off-The-Shelf System (OTSS) is a packaged information management system for hierarchical data bases. OTSS provides, without computer programming, processes to enter and alter data in such a data base, do complex retrievals of data from the data base, and specify various security mechanisms to limit access to, or alteration of, a data base. OTSS also provides a mechanism for extending the available processes on a project-by-project basis. OTSS has been implemented using MASTER LINKS and the NATURAL DIALOGUE SYSTEM.

I. INTRODUCTION

The Off-The-Shelf System (OTSS) is a packaged information management system for hierarchical data bases. Earlier work done by Sinowitz[1] was aimed at providing information retrieval capabilities for a specific hierarchical data base. OTSS was designed to operate on any hierarchical data base regardless of its structure and regardless of the data fields stored in the data base.

Retrieval processes are available to print, alter, rank, plot, distribute, compute statistics, and perform regression analysis of data. These processes are specified to OTSS in a key-word English-like language in an interactive dialogue environment. OTSS allows for simple alteration of the retrieval process from request to request by selective replacement, deletion, or addition of statements to the dialogue description of the process to be performed.

As part of the package, OTSS provides a data-base-independent security mechanism. This mechanism allows a data base administrator to restrict access or alteration of a data base and use of certain process and language facilities in OTSS on a user-by-user basis. A user can be restricted to a logical hierarchical subsection of the data base and to only certain data items stored in that subsection. The user can be further restricted to using only certain processes of OTSS such as printing, ranking, or altering processes, but not plotting or distributing.

In addition, OTSS provides the ability to extend the processing capabilities of the system by allowing a programmer to add new processes to the system on a project-by-project basis. The processes so installed are available to the project installing them, and do not become a permanent part of OTSS.

OTSS was implemented using the MASTER LINKS[2] data base management system and the NATURAL DIALOGUE SYSTEM,[3] a system for designing and implementing interactive computing languages in a dialogue environment.

II. SYSTEM DESIGN CRITERIA

As a packaged information management system, OTSS was developed to satisfy certain basic design criteria. The two primary design criteria are hierarchical independence and field independence, i.e., the structure and specific content of the data base. In establishing these as design criteria the retrieval processors, security mechanisms, and associated language specifications are written to operate on any hierarchical data base regardless of its logical structure and regardless of the data types of the fields of the data base. The system provides these capabilities by making use of the information contained in the driving tables of the data base system, MASTER LINKS, used to implement OTSS. These driving tables describe the logical structure and data fields of a given data base.

Another design criterion of OTSS was to provide the user with the ability to specify a generalized retrieval function in the sense of Ref. 2, and a comprehensive set of data base processors to operate on such functions. In general, a retrieval function is defined as a combination of data base fields, constants, and previously defined retrieval functions using the standard arithmetic, relational, and logical operators. Through a keyword-oriented language, a user of OTSS can specify an arbitrary retrieval function, and, for example, request the system to print its values, rank its values, or plot its values by the values of another retrieval function. The user can also specify a series of logical retrieval functions which are to be used to selectively delimit the search of the data base during the retrieval process.

Directed output of any retrieval process is a fourth basic design criterion. Through the retrieval language, a user can direct the output of any retrieval process to any external device including line printer, card punch, magnetic tape, disk, or console.

As a final design objective, OTSS offers the programming audience the ability to extend the processing capabilities of the system on a project-by-project basis. A programmer can write a project-dependent processor and "install" such a processor into the OTSS environment. Once installed, this processor is available only to the project installing it, and does not become a permanent part of OTSS.

In the following sections we shall describe in more detail what we mean by a hierarchical data base, examine typical uses of retrieval functions, and present the definition of the load and retrieval phases of OTSS.

III. HIERARCHICAL DATA BASES

As discussed in greater detail in Ref. 2, a *hierarchical data base* is a directed tree which is rooted at one entity (node). At each entity in the tree is stored a set of fields. Two entities belong to the same *group* if the set of fields stored at one entity is identical to the set of fields stored at the other. Two entities belonging to the same group are at the same depth in the tree (i.e., connected to the root entity of the tree by paths of the same length). Also the ancestor groups of one entity belonging to a group must be identical to the ancestor groups of any other entity belonging to the group.

Using the concept of groups, it is possible to represent the structure of a hierarchical data base as a rooted tree whose entities are the groups of the hierarchical data base placed in the tree analogously to the entities in the data base with respect to depth and connectivity. At each entity in the group tree are listed the fields stored at that group. In reference to the group tree, we say that a set of groups forms a chain if and only if for any G_i and G_j belonging to the set of groups, G_i is an ancestor or descendant of G_j. A group chain is complete if all the ancestors of every group on the chain are on the chain. Entity chains and complete entity chains are defined in an analogous way.

3.1 *Structure of a Sample Hierarchical Data Base*

As an example of the structure of a hierarchical data base, consider the group tree presented in Fig. 1. This hierarchical data base is rooted at the COMPANY group as there is only one company. We shall return to this sample data base in Section IX when we discuss examples of using the OTSS retrieval language.

IV. SPECIFYING THE STRUCTURE OF A DATA BASE

OTSS is independent of the hierarchical structure of the data base and the particular fields stored in the data base. OTSS obtains all its required information about the data base from a series of driving tables. These driving tables are produced using the BUILD facility of MASTER LINKS.[2]

BUILD allows the data base designer to completely specify the logical structure of the data base. The driving tables produced by BUILD are used by OTSS to determine the correctness of data loading and retrieval requests and by MASTER LINKS to be able to enter and access data in the data base.

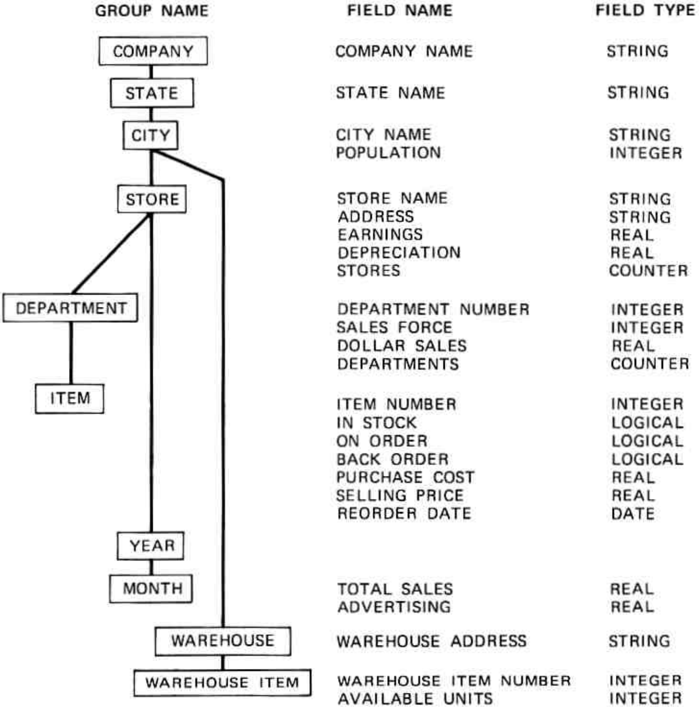
V. LOADING DATA INTO A DATA BASE

Once BUILD has been used by the data base designer to specify the logical structure of a hierarchical data base, it is ready to have data loaded into it. OTSS provides a facility, called LOAD, for bulk loading of data into the data base.

LOAD allows files of data to be sequentially loaded into a data base, i.e., LOAD does not provide for multiple concurrent updates of a data base. LOAD does provide a simple mechanism to restart a data load which was terminated abnormally due to machine failure or human error.

The file of data to be loaded using LOAD is logically divided into sections. Each section is identified by an integer number, called the *card type*, which must appear in columns 1 through 3 of each record of the section. More than one section in the data file may have the same card type. The data within a given card type section must be organized according to a specific card type definition, and it must be organized in the same manner for every section having the same card type.

The definitions of the various card types are defined once by the data base designer using the definition phase of LOAD. In a card-type definition, the designer indicates where, on the records of the specified section, the data values for particular fields can be found and the name of the entity where the data is to be loaded. Along with the field name, the user indicates which record of the section and which field (set of contiguous columns) of that record contains the value of



GROUP NAME	FIELD NAME	FIELD TYPE
COMPANY	COMPANY NAME	STRING
STATE	STATE NAME	STRING
CITY	CITY NAME	STRING
	POPULATION	INTEGER
STORE	STORE NAME	STRING
	ADDRESS	STRING
	EARNINGS	REAL
	DEPRECIATION	REAL
	STORES	COUNTER
DEPARTMENT	DEPARTMENT NUMBER	INTEGER
	SALES FORCE	INTEGER
	DOLLAR SALES	REAL
	DEPARTMENTS	COUNTER
ITEM	ITEM NUMBER	INTEGER
	IN STOCK	LOGICAL
	ON ORDER	LOGICAL
	BACK ORDER	LOGICAL
	PURCHASE COST	REAL
	SELLING PRICE	REAL
	REORDER DATE	DATE
YEAR		
MONTH	TOTAL SALES	REAL
	ADVERTISING	REAL
WAREHOUSE	WAREHOUSE ADDRESS	STRING
WAREHOUSE ITEM	WAREHOUSE ITEM NUMBER	INTEGER
	AVAILABLE UNITS	INTEGER

Fig. 1.—Structure of a sample hierarchical data base.

the field. Every section having the same card type must have the value for a specified field in the same field of the same record of the section as described in the corresponding card-type definition.

Thus the card-type definition specifies what data are contained in a section of a file of input data and where the data are to be loaded into the data base. When LOAD processes a file of data, it looks at columns 1 through 3 of the records to determine what card-type definition should be used for the section and applies the appropriate card-type definition to direct its data loading process.

VI. RETRIEVALS FROM THE SAMPLE DATA BASE

A simple form of a retrieval process on the sample data base is to extract the value of a single field at all its occurrences in the data base. For instance, one might wish to extract all the values of DOLLAR SALES in the data base. A more interesting case is to extract, along with the values of DOLLAR SALES, the corresponding values of DEPARTMENT NUMBER. If the department numbers were only unique within a store, one might wish to also extract the corresponding values of STORE NAME. Notice that due to the structure of our sample hierarchical data base, there is only one value of STORE NAME for each value of DEPARTMENT NUMBER and DOLLAR SALES, but there are several values of DOLLAR SALES and DEPARTMENT NUMBER for each value of STORE NAME.

Suppose now that one were interested only in extracting the value of DOLLAR SALES for a given department within a given store. This is similar to the first example of a retrieval process, except that the retrieval process would first delimit the search of the data base to a subtree of the data base consisting of the particular STORE entity and DEPARTMENT entity. To do this there is a directory into the data base which is based on the name of an entity or a chain of entity names. Having delimited the search of the data base to the particular store and department, the retrieval process can extract the one value of DOLLAR SALES contained in the delimited data base.

Some retrieval processes combine extraction based on known entity names and the values of fields stored in the data base. One might wish to extract the value of ITEM NUMBER for those items in one particular store which have SELLING PRICE greater than $9.00. In this case, the retrieval process would first delimit the search of the data base to the entity in the STORE group with the appropriate name and to all entities which are descendants of it. The retrieval process would then search through all entities in the ITEM group in the delimited part of the data base and extract the ITEM NUMBER for those items having SELLING PRICE greater than $9.00.

So far only examples of extracting the values of simple fields stored in the data base have been discussed. It is also possible to evaluate more complex retrieval functions as part of the retrieval process. A simple example would be to extract the value of SELLING PRICE/PURCHASE COST. This retrieval process creates a new pseudo-field at the ITEM group which is then extracted. A more complex example is the summing of all the values of DOLLAR SALES within a store. To evaluate this function, the retrieval process would have to extract the value of DOLLAR SALES for every DEPARTMENT entity under each STORE entity and then add them together. This process produces a new pseudo-field at the STORE group which is then extracted. An operation which raises the level, in the hierarchy, of definition of a field or expression is called a level-raising operation.

Summing is not the only level-raising operation which can be performed. Others are minimum, maximum, and average on numerical data and any, all, and none for logical data. For instance, one might wish to extract the DEPARTMENT NAME of all departments that have their minimum ratio of SELLING PRICE to PURCHASE COST less than 1. One might also wish to extract the DEPARTMENT NAMES as above, but only including in the level-raising operation items which have a selling price greater than $9.00.

We have now seen several examples of retrieval processes. All these retrieval processes are examples of a complete retrieval process which delimits the search of a data base by entity names, accepts or rejects entities by logical conditions, and evaluates complex retrieval functions including level-raising. We have only referred to extracting data from a data base and have not said anything about what should be done with the data once extracted. This was done intentionally to divorce the retrieval process from the displaying of the extracted data.

The retrieval language provides processes to print, alter, rank, plot, distribute, compute statistics, and perform regression analysis of extracted retrieval functions by using the appropriate keyword.

VII. SPECIFICATION OF RETRIEVAL FUNCTIONS

Retrieval functions are specified by combining data base fields, constants, and previously defined retrieval functions using the standard arithmetic, relational, and logical operators. The arithmetic operators defined on numeric data and their symbols are: addition (+), subtraction (−), multiplication (*), division (/), and exponentiation (↑). The relational binary operators defined for numeric data and date data and their symbols are: equal to (=), not equal to (− =), greater than (>), greater than or equal to (> =), less than (<), and less than or equal to (< =). Relational binary operators defined for string data and their symbols are: equal to (=) and not equal to (− =). Logical binary operators defined for logical data and their symbols are: logical and (AND) and logical or (OR).

The unary operators available in the OTSS retrieval language are of two types: those which operate on a single value and those which operate on a set of values. Unary operators operating on a single value are: unary plus (+), unary minus (−), logarithm to the base 10 (LOG10), logarithm to the base e (LOGE), e raised to a power (EXPF), absolute value (ABSF), sine (SINF), and cosine (COSF) for numeric data and not (NOT) for logical data.

Unary operators which work on a set of values are the level-raising operators. Level-raising operators are of the following two forms:

lr field PER gn

or

GLOBAL lr field PER gn,

where lr is any level-raising operator and gn is any group name. The field must be defined at a group which is a descendant of the group following the PER. The set of values that the level-raising operator operates on are the values of the field in entities which are descendants of an entity in the group following PER. The level-raising operator takes the set of values and computes a single value which depends on the level-raising operator. The level-raising operators for numeric values are: sum (SUM), minimum (MIN), maximum (MAX), and average (AVG). The level-raising operators for logical values are: logical any (ANY), logical all (ALL), and logical none (NO).

If the level-raising operator is not preceded by the literal GLOBAL, any entity restrictions that have been applied to all groups between the group following PER down to and including the group of the field are evaluated and entities and their descendants are rejected for which the entity restriction is not satisfied. An entity restriction is a retrieval function whose type is logical. An entity in the group of the retrieval function is accepted or rejected if the logical function evaluates to TRUE or FALSE respectively. The remaining set of entities at the group of the field are then combined using the level-raising operator. If the literal GLOBAL is present, all entity restrictions below the group following PER are ignored.

Unary operators can be nested without parentheses and are evaluated from right to left. Parentheses can be used to cause evaluation of a retrieval function to occur in other than the normal order of evaluation.

The groups of all fields in a retrieval function must form a group chain. A retrieval function has a definition group associated with it. The definition group of a retrieval function is the group of maximum depth of the groups of fields not operated upon by a level-raising operator and the groups given in level-raising operators. In this manner, retrieval functions are made to be single-valued for each entity in the definition group.

VIII. THE RETRIEVAL PROCESS

The retrieval process used by OTSS can be described by considering the steps involved to evaluate any one retrieval function for all entities at which it is defined in a subset of the data base. To describe the retrieval process, let us assume we wish to evaluate a retrieval function, f, defined at some group, G, at every entity of G in a subset of the data base. The steps of the retrieval process are as follows:

Step 1: Entity Selection Based on Entity Chains

(a) Delimit the search of the data base by constructing an access tree based on any specified entity chains.

Step 2: Entity Selection Based on Entity Restrictions

(a) Start at the root of the access tree constructed in Step 1.
(b) Select the next entity in a depth-first, left-to-right manner. If all entities have been selected, the retrieval process is finished.
(c) If an entity restriction has been placed on entities in the group of the entity obtained by Step 2b, apply it. If the result is "reject," reject the entity and all its descendants and go back to Step 2b. If f is defined at the group of the entity, evaluate it using Step 3. Upon completion of the evaluation or if f is not defined at the entity, go to Step 2b.

Step 3: Function Evaluation

(a) If f is a field, retrieve its value; or if f is a constant, use its value.
(b) If f is a level-raised field without the GLOBAL prefix, apply all entity restrictions to entities in all groups from G down to and including the group of the level-raised field and ignore all entities and their descendants for which the entity restriction is not satisfied. Perform Step 3 on each remaining entity of the group of the field and combine the results according to the appropriate level-raising operator.
(c) If f is a level-raised field with the GLOBAL prefix, perform Step 3 on all descendant entities at the group of the field and combine the results according to the appropriate level-raising operator.
(d) Combine values obtained in Steps 3a, 3b, and 3c using appropriate operators.

The above described retrieval process can be expanded to evaluate several different retrieval functions during one pass through the data base. It can be seen that all the retrieval processes in Section VI can be formulated in terms of the general retrieval process given above.

Hence the user of OTSS need only learn how to specify the retrieval process and the form of output desired. Detailed algorithms for implementing the retrieval process are described in Appendix A.

IX. LANGUAGE EXAMPLES

Having presented the descriptions of retrieval functions and the retrieval process, let us proceed to examine several examples of the OTSS retrieval language. The OTSS retrieval language is a keyword-oriented, English-like language which provides the necessary input to the retrieval process and means of specifying the output format. The language contains FOR and IN statements which are used to specify subtree delimiting of the search of the data base; WHEN statements for specifying logical entity restrictions; LET statements for specifying retrieval functions; and various output specification statements such as PRINT, RANK, PLOT, etc. A complete description of these and other auxiliary statements are described in Appendix B.

To begin our examples, let us print the names of all the departments in the store at 19 Fifth Ave., New York City, New York. The following statements accomplish this:

PRINT DEPARTMENT NAME: FOR 19 FIFTH AVE, NEW YORK CITY, NEW YORK: GO:

Now to print only those departments with dollar sales greater than $5000 we need only enter the following statements:

WHEN DEPARTMENT HAS DOLLAR SALES > 5000: GO:

as OTSS remembers the last occurrence of each keyword statement entered.

To print the department which has the highest dollar sales in the store at 19 Fifth Ave., we would enter the statements:

WHEN DEPARTMENT HAS DOLLAR SALES = GLOBAL MAX DOLLAR SALES PER STORE: GO:

Suppose we now wished to print the ratio of dollar sales of those departments having dollar sales less than $5000 to the dollar sales of the entire store. We would enter the statements:

PRINT DOLLAR SALES PER STORE/GLOBAL DOLLAR SALES PER STORE: WHEN DEPARTMENT HAS DOLLAR SALES < 5000: GO:

If we wished to do the above request for all stores, not just the one at 19 Fifth Ave., we would enter:

```
DELETE FOR: GO:
```

And suppose finally we wanted to do the same request for only those stores whose dollar sales are greater than $1,000,000 but less than $5,000,000. We would enter:

```
WHEN STORE HAS GLOBAL DOLLAR SALES PER STORE
> 1000000 AND GLOBAL DOLLAR SALES PER STORE
< 5000000: GO:
```

To save a small amount of typing, we could have entered the following:

```
LET X = GLOBAL DOLLAR SALES PER STORE: WHEN
X > 1000000 AND X < 5000000: GO:
```

X. REPORT FACILITY

The REPORT statement is the general interface to extend the processes available in the OTSS retrieval language on a project-by-project basis. One can write a process in FORTRAN which can be installed into OTSS to be referenced by some report name using the REPORT statement. The syntax of the REPORT statement is the keyword "REPORT" followed by a report name optionally followed by a list of retrieval functions separated by commas. If the process requires retrieval functions to be passed into it as parameters, they follow the report name in a manner analogous to the retrieval function list in the PRINT statement. The process so installed is available to the project installing it, and does not become a permanent part of OTSS.

XI. SYSTEM SECURITY

The SECURITY statement is a command in the language which enables a data base administrator to define, interrogate, and remove security information for his user audience. The types of security which are available to a data base administrator are environmental, language, and data base. These security mechanisms may be used by the administrator to restrict access or alteration of his data base and use of facilities in the retrieval language on a user-by-user basis.

11.1 *Environmental Security*

The first type of security which may be specified is the environmental security. Environmental security is used by the administrator to specify legal users of the system. This security mechanism uses two pieces of information: a sign-on key passed into the system and password information input by the user. The data base administrator specifies valid combinations of sign-on keys and optional password information for each potential user of the system. When a user initially enters the retrieval environment, the system will interrogate the sign-on key typed in by the user to see if it is valid. If the sign-on key is not in the list of valid sign-on keys provided for by the administrator, the system will terminate the session. A valid sign-on key will cause the system to prompt the user for the password information (if this sign-on key requires a password). If the password is incorrect, the session is terminated.

11.2 *Language Security*

The data base administrator has the ability to limit use of certain facilities in the retrieval language. This type of security is called language security. In order to specify language security the administrator defines one or more statement restriction classes. A statement restriction class is a set of statements in the retrieval language which is *not* available to a set of users of a data base. The administrator would then indicate, on a user-by-user basis, which statement restriction class pertains to each user. If a user attempts to use a statement in the retrieval language which is a member of his statement restriction class, the system will output a message indicating that the statement in question is not available for use by him.

11.3 *Data Base Security*

The final type of security provided for by the system is data base security. Data base security can be subdivided into two parts: field security and access tree security. Field security is specified in a manner similar to the language security specification. The data base administrator defines one or more field restriction classes. A field restriction class is a set of fields in the data base which is *not* accessible to one or more users of the system. A field restriction class may be restricted on a read/write basis or on a write basis only. After defining the sets of field restriction classes, the administrator indicates, on a user-by-user basis, which restriction class pertains to each user. If a user attempts to retrieve or modify the value of a field which is a member of his field restriction class, the system will output a message indicating that the field in question is not available for access or alteration.

Access tree security is used to restrict users to a logical hierarchical subsection of a data base. For each potential user of his system, the data base administrator may specify a corresponding USER statement. The USER statement has the same form as the FOR and IN statements described in Appendix B, and is used to delimit the search of the data base.

When a user initially enters the retrieval environment, after passing the environmental security phase, the USER statement corresponding to that user will be processed to delimit the search of the data base. If the user tries to access a portion of the data base outside of the logical hierarchical subsection specified in the USER statement, the system will output a message indicating this as an illegal action. Note that the USER statement cannot be modified or deleted by a user.

XII. CONCLUSION

OTSS is an information management system designed to be independent of the structure or content of any specific hierarchical data base. OTSS provides a simple keyword-oriented, English-like language for specifying the retrieval of values of complex retrieval functions and the alteration of data in a data base. In addition, OTSS provides a means of loading data into a data base and specifying various forms of security on the data base and the use of statements in the language.

APPENDIX A

The retrieval process (RP) has as its inputs a set of complete access lists, a set of retrieval functions, and a set of entity restriction functions. These inputs completely specify the semantics of the retrieval process.

RP applies the algorithm TB (Tree Building) to construct a subtree of the data base over which the retrieval process will be performed. RP applies the algorithm GTP (Group Tree Pruning) to make up a list, R, whose entry for each group, g, is "referenced" if g is the definition group of a retrieval function or an ancestor of the definition group of a retrieval function.

RP uses the algorithm ACTION to create a list, A, of selector directions for each group in the hierarchy. The entry in A for group, g, is "down" if it is an ancestor of the definition group of a retrieval function and is "right" otherwise.

After having applied TB, GTP, and ACTION, RP proceeds to select the entities of the subtree (using the algorithm GEN) in a left-to-right, depth-first manner. The algorithm GEN only returns entities to RP for which a retrieval function may have to be evaluated or for which an entity restriction function may need to be evaluated and which is the group of a retrieval function or an ancestor of a group of a retrieval function. Hence, in this manner, no extraneous entities are selected.

Whenever an entity is returned to RP by GEN, RP determines if there is an entity restriction function to be evaluated at this entity. If there is, it is evaluated. If the entity restriction function evaluates to false, the action input to GEN is set to "right" and GEN is applied to select the next entity.

Should there be no entity restriction function to be evaluated, or should it evaluate to true, RP examines the list of retrieval functions to be evaluated and evaluates those defined at the group of the current entity. RP then iterates the whole procedure until all the entities on the subtree have been generated.

More formally the following algorithms define the functions of RP, TB, GTP, ACTION, and GEN.

Retrieval Process (*RP*)

Input: Complete entity access lists: $e_1, e_2, \cdots, e_n$.
Retrieval functions: $f_1^{g_1}, f_2^{g_2}, \cdots, f_n^{g_n}$.
Entity restriction functions: $b_1^{g_1}, b_2^{g_2}, \cdots, b_n^{g_n}$.
Output: Values of retrieval functions: $f_1^{g_1}, f_2^{g_2}, \cdots, f_n^{g_n}$.

Step 1: $T \leftarrow TB(e_1, e_2, \cdots, e_n)$ which builds a tree, T, the subtree of the data base that the retrieval process is to be applied to.

Step 2: $R \leftarrow GTP(f_1^{g_1}, f_2^{g_2}, \cdots, f_n^{g_n})$ which builds a list, *R*, containing one entry for each group in the hierarchy. The entry for a group is marked "referenced" if the group is one of the g or one of its ancestors and is marked "unreferenced" otherwise.

Step 3: $A \leftarrow ACTION(f_1^{g_1}, f_2^{g_2}, \cdots, f_n^{g_n})$ which builds a list, A, containing one entry for each group in the hierarchy. The entry for a group is marked "down" if the tree traversal action to be performed is "down" and is "right" otherwise.

Step 4: CE ← root entity of the retrieval tree, T.

Step 5: Examine the list of entity restriction functions to see if any $b_i^{g_i}$ is defined at the group of the current entity, CE. If none, go to Step 7.

Step 6: Evaluate $b_i^{g_i}$. If the value is FALSE, go to Step 11.

Step 7: Any more $f_i^{g_i}$ to be evaluated? If none, go to Step 9.

```
Step  8: Evaluate $f_i^{g_i}$, output result and go to Step 7.
Step  9: If there are no more entities on T to be generated, then exit.
Step 10: CE ← GEN(T, R, A($g_i$), CE) which generates the next
         entity on T, then go to Step 5.
Step 11: If there are no more entities on T to be generated, then exit.
Step 12: CE ← GEN(T, R, "right," CE), then go to Step 5.
```

Tree Building (TB)

Input: Complete entity access lists: $e_1, e_2, \cdots, e_n$.
Output: The retrieval tree, T_1.

```
Step  1: $T_1$ ← null tree.
Step  2: If there are no more $e_i$, go to Step 5.
Step  3: Construct tree, $T_2$, consisting of the entities on the access
         list, $e_i$.
Step  4: $T_1$ ← $T_1$ union $T_2$, go to Step 2.
Step  5: Make $T_2$ a copy of $T_1$.
Step  6: If there are no more unexamined entities on $T_2$, exit.
Step  7: e ← next unexamined entity on $T_2$.
Step  8: Examine each group which is a descendent of the group of e
         to see if there exists an entity on $T_2$ which is a descendent
         of e. For each group in which this is not true, put all entities
         in the data base on $T_1$ which are descendents of e. Go to
         Step 6.
```

Group Tree Pruning (GTP)

Input: Retrieval functions: $f_1^{g_1}, f_2^{g_2}, \cdots, f_n^{g_n}$.
Output: A list, R, containing one entry for each group in the hierarchy. The entry for a group is marked "referenced" if the group is one of the g_i or an ancestor of one of the g_i and is marked "unreferenced" otherwise.

```
Step  1: Initialize the entries in R for each group in the hierarchy to
         "unreferenced."
Step  2: If there are no more $f_i^{g_i}$, exit.
Step  3: g ← $g_i$.
Step  4: If the entry in R for group g is "referenced," go to Step 2.
Step  5: Set the entry in R for group g to "referenced."
Step  6: If g is the root group, go to Step 2.
Step  7: g ← father of g; go to Step 4.
```

Generating Action (ACTION)

Input: Retrieval function: $f_1^{g_1}, f_2^{g_2}, \cdots, f_n^{g_n}$.

Output: A list, A, containing one entry for each group in the hierarchy. The entry for a group is marked "down" if the tree traversal is down, and "right" otherwise.

```
Step 1: Initialize the entries in A for each group in the hierarchy to "right."
Step 2: If there are no more $f_i^{g_i}$, then exit.
Step 3: $g \leftarrow g_i$.
Step 4: If the entry in A for group g is "down," go to Step 2.
Step 5: If g is the root group, go to Step 2.
Step 6: $g \leftarrow$ father of g.
Step 7: If the entry in A for group g is "down," go to Step 2.
Step 8: Set entry in A for group g to "down," go to Step 5.
```

Tree Generation (GEN)

Input: Retrieval tree, T; the list R of referenced and unreferenced groups; and the action, A, either "right" or "down," and the current entity, CE.

Output: CE the next entity to be processed by the retrieval process.

```
Step 1: If the action, A, is "right," go to Step 4.
Step 2: Find the leftmost entity on T, LME, which is a descendent of CE and for which the entry for the group of CE in list R is marked "referenced." If there are none, go to Step 4.
Step 3: CE $\leftarrow$ LME, then exit.
Step 4: If CE has no brother entity to the right, go to Step 6.
Step 5: CE $\leftarrow$ next brother of CE to the right, then exit.
Step 6: Find the leftmost group, g, which is on the same level as the group of CE for which the entry in the list R is marked "referenced" and for which there exists an entity on T which has not previously been processed. If none exists, go to Step 8.
Step 7: CE $\leftarrow$ leftmost entity of group, g, which has not yet been generated; then exit.
Step 8: CE $\leftarrow$ father of CE; go to Step 4.
```

APPENDIX B

OTSS Retrieval Language Statements

To make the description of the OTSS retrieval language statements more readable, the following notations are used:

(a) Capitals and special symbols are literals in the language.

(b) Lower case include:

f —any retrieval function
str —any non-null string of alphanumeric characters
num—any number
gn —the name of a group
null —a null character string
stmt—the name of a statement in the retrieval language.

(c) Square brackets imply that the constructs within the brackets are alternatives starting from the top line down. One item from the vertical list of alternatives must be selected.

$$\text{LET str} = \text{f}:$$

The LET statement is used to create additional fields which are not stored in the data base. The field so created may be used in any other statement or retrieval function.

$$\begin{bmatrix}\text{FOR}\\ \text{IN}\end{bmatrix} \text{e-list}_1;\ \text{e-list}_2;\ \cdots\ \text{e-list}_n:$$

The notation, e-list, indicates a list of entity names separated by commas. Each e-list represents an entity chain. The combination of all entity chains specifies an access tree. This access tree is used to select the subset of the data base over which the retrieval search will take place.

$$\text{WHEN} \begin{bmatrix}\text{gn HAS}\\ \text{null}\end{bmatrix} \text{f}:$$

The WHEN statement specifies an entity restriction function defined at the group, gn, which delimits the search of the data base during the retrieval process. If a WHEN condition is defined for a group, retrieval will take place from a entity within that group only if the entity restriction function evaluates to TRUE. If the entity restriction evaluates to FALSE, the entity (and all its descendents) will be ignored during the retrieval process.

$$\text{PRINT f}_1,\ \text{f}_2,\ \cdots\ \text{f}_n:$$

The PRINT statement specifies a tabular printout of the values of the individual retrieval function.

$$\text{DISTRIBUTE f}_1 \text{ BY f}_2:$$

The DISTRIBUTE statement specifies a tabular histogram with f_1

as the ordinate and f_2 as the abscissa.

$$\text{BETWEEN num}_1 \begin{bmatrix}\text{TO}\\ \text{AND}\end{bmatrix} \text{num}_2 \begin{bmatrix}\text{IN STEPS OF num}_3\\ \text{ISO num}_3\\ \text{null}\end{bmatrix} :$$

The BETWEEN statement specifies the range and cell intervals to be used by the DISTRIBUTE process.

CUMULATIVELY:

The CUMULATIVELY statement may be used with the DISTRIBUTE process to alter the distribution to produce cumulative values in each of the defined cells.

CHART:

The CHART statement may be used with the DISTRIBUTE process to specify bar chart output.

RANK f AT gn:

The RANK statement specifies to rank in descending order (largest to smallest) the individual values of f within each entity of the group gn, and displays the results in tabular form.

INVERSELY:

The INVERSELY statement specifies to the RANK process to invert the order of the RANK output.

$$\text{KEEPING} \begin{bmatrix}\text{THE}\\ \text{null}\end{bmatrix} \begin{bmatrix}\text{LARGEST}\\ \text{HIGHEST}\\ \text{SMALLEST}\\ \text{LOWEST}\end{bmatrix} \text{num}:$$

The KEEPING statement is used to specify to the RANK process to rank the "num" largest or smallest values of the rank function at each entity in the rank group.

CARRYING ALONG $f_1, f_2, \cdots f_n$:

The CARRYING statement may be used to specify to the RANK process to "carry along" the values of other retrieval functions and have them displayed as part of the RANK output.

PLOT $f_1, f_2, \cdots, f_{n-1}$ BY f_n:

The PLOT statement specifies an X-Y point plot with the values of

f_1 through f_{n-1} as the ordinate and f_n as the abscissa.

$$\begin{bmatrix}\text{X-AXIS}\\ \text{Y-AXIS}\end{bmatrix}\begin{bmatrix}\text{BETWEEN}\\ \text{FROM}\\ \text{null}\end{bmatrix}\text{num}_1\begin{bmatrix}\text{AND}\\ \text{TO}\end{bmatrix}\text{num}_2\text{:}$$

The X-AXIS and Y-AXIS statements must be used to specify to the PLOT process to specify the origins and ranges of the independent and dependent fields.

$$\text{STATISTICS } f_1,\ f_2,\ \cdots f_n\text{:}$$

The STATISTICS statement requests a set of standard statistics to be produced for each function and the results piinted in tabular form.

$$\text{REGRESS } f_1 \text{ BY } f_2,\ f_3,\ \cdots f_n\text{:}$$

The REGRESS statement specifies to perform a multiple linear regression analysis of the function f_1 (dependent field) by the functions f_2 through f_n (independent fields).

$$\text{ALTER field TO } f\text{:}$$

The ALTER statement is used to permanently change the value of the field to the value of f.

$$\text{INTERACTION}\begin{bmatrix}\text{BRIEF}\\ \text{DETAIL}\\ \text{VERIFY}\end{bmatrix}\text{:}$$

The INTERACTION statement specifies to the ALTER process a level of verification required by the user when altering a datum value.

$$\text{REPORT str}\begin{bmatrix}, f_1,\ f_2,\ \cdots f_n\\ \text{null}\end{bmatrix}\text{:}$$

The REPORT statement specifies to pass control to an application dependent process identified by the string, str.

$$\text{TITLE str}_1\text{! str}_2\text{ !}\cdots\text{! str}_n\text{:}$$

The TITLE statement specifies to any process to print lines of text centered at the top of the output of the process.

$$\text{PLACES num:}$$

The PLACES statement is used to control the number of decimal places displayed for real-valued functions.

$$\text{OUTPUT TO str:}$$

The OUTPUT statement specifies to any process to direct its output to a specific output device.

$$\text{DELETE}\begin{bmatrix}\text{WHEN FOR gn}_1,\ \text{gn}_2,\ \cdots\text{gn}_n\\ \text{stmt}\\ \text{ALL}\end{bmatrix}:$$

The DELETE statement specifies to remove a statement or set of statements which are currently active.

$$\text{DEFINE var}_1,\ \text{var}_2,\ \cdots\text{var}_n:$$

where var_1 through var_n are the names of created fields previously defined through the use of the LET statement. The DEFINE statement will *permanently* save the name and definition of each of the created fields mentioned in the DEFINE list.

$$\text{UNDEFINE field}_1,\ \text{field}_2,\ \cdots\text{field}_n:$$

where field_1 through field_n are the names of fields which were permanently created through the use of the DEFINE command. The UNDEFINE statement specifies to remove the names of the permanently created fields from the list of all possible fields accessible through the retrieval language.

DETAIL:

The DETAIL statement specifies to automatically recap the current state of the dialogue when a process is executed.

RECAP:

The RECAP statement specifies to display the current state of the dialogue.

INPUT FROM str:

The INPUT statement causes OTSS to accept input from a previously prepared file identified by str.

$$\text{SAVE}\begin{bmatrix}\text{IN}\\ \text{null}\end{bmatrix}\text{str}\begin{bmatrix}\text{WITH GO}\\ \text{null}\end{bmatrix}:$$

When the SAVE statement is given by the user, the system writes the current state of the dialogue on to the file identified by str.

DATABASE str:

where str is the name of another data base. The DATABASE statement allows the user to switch from one data base to another from within OTSS.

ERASE str:

The ERASE statement causes the disk file identified by str to be erased.

VOCABULARY:

The VOCABULARY statement specifies to print out the entire list of keywords and their associated synonyms available in the OTSS retrieval language.

RETURN:

The RETURN statement is used to return control to the operating system level.

STOP:

The STOP statement will disconnect the user from the time-sharing system.

$$\text{SECURITY}\left\{\begin{array}{ll}\text{CREATE} & \left[\begin{array}{l}\text{str}\\ \text{null}\end{array}\right]\\ \text{INTERROGATE} & \left[\begin{array}{l}\text{ALL}\\ \text{user}_1,\ \text{user}_2,\ \cdots\text{user}_n\end{array}\right]\\ \text{REMOVE} & \left[\begin{array}{l}\text{ALL}\\ \text{user}_1,\ \text{user}_2,\ \cdots\text{user}_n\end{array}\right]\end{array}\right\}:$$

The SECURITY statement is used by a data base administrator to define, interrogate, and remove security information for his user audience.

GO:

The GO statement causes the last-mentioned process to be executed.

REFERENCES

1. Sinowitz, N. R., "DATAPLUS—a Language for Real-time Information Retrieval from a Hierarchical Data Base," AFIPS Conf. Proc., *32*, 1968.
2. Gibson, T. A., and Stockhausen, P. F., "MASTER LINKS—A Hierarchical Data System," B.S.T.J., this issue, pp. 1691–1724.
3. Puerling, B. W., and Roberto, J. T., "The Natural Dialogue System," B.S.T.J., this issue, pp. 1725–1741.

THE BELL SYSTEM TECHNICAL JOURNAL
Vol. 52, No. 10, December, 1973
Printed in U.S.A.

The Potential in a Charge-Coupled Device With No Mobile Minority Carriers

By J. McKENNA and N. L. SCHRYER

(Manuscript received May 17, 1973)

The potentials and fields in a two-dimensional model of a charge-coupled device (CCD) are studied. We assume no mobile minority carriers have been injected into the CCD and that the electrode voltages do not vary with time. The nonlinear equations describing the devices are first linearized using the depletion layer approximation. The linearized equations are then solved approximately by a fitting technique. Both surface and buried channel CCD's are considered. The accuracy and cost of obtaining the solution is discussed. This work is a continuation of a study initiated in an earlier paper.[1]

I. INTRODUCTION AND SUMMARY

In this paper we study the electrostatic potential distribution and fields in a two-dimensional model of a charge-coupled device[2,3] (CCD). This work is a continuation of a study initiated in an earlier paper[1] hereafter referred to as I. In I we considered a static, two-dimensional model with no mobile charge, and with electrodes so close together that they could be assumed to touch. We showed there that the depletion layer approximation[4] could be used to linearize the potential equations, and the linearized equations were then solved analytically. The numerical evaluation of these solutions was shown to be very accurate and cheap.

We extend the model of I to allow for gaps between the plates. Our purpose here is twofold. We want to examine the dependence of the potentials and fields in a CCD on various design parameters. As we show, our model allows considerable flexibility in describing various electrode configurations. In addition, however, we want to demonstrate a method of numerically solving the potential equations which we believe is of considerable interest in itself.

Both surface[2] and buried channel[5,6] CCD's are considered. However, as in I, only the analysis for buried channel CCD's is given. The results for a surface CCD can be obtained as special cases of the results for buried channel CCD's. We refer the reader to I for a more detailed derivation of the equations and for a discussion of their linearization by the depletion layer approximation.

As we show by examples, fairly complicated models of CCD's can be analyzed at moderate cost by the methods of this paper. Nevertheless, the cost of using the methods of I to analyze a CCD with zero separation between the electrodes is typically an order of magnitude less than the cost of using the methods of this paper to analyze a CCD with nonzero electrode separation. This suggests that, in any complicated design problem, the methods of I should be used to rough out a solution, and then the solution should be "fine tuned" by using the methods of this paper. In addition, it is shown that, when the gaps between the electrodes are of the order of 1 μm, the potentials of interest are approximated well by the potentials in the same CCD with the electrode separation set to zero.

The nonlinear equations and boundary conditions defining the boundary value problem are introduced in Section II. The linearized equations are also introduced there. In Section III we discuss in some detail how we obtain approximate solutions to the linearized problem. The reader uninterested in the mathematical details should skip Section III and proceed directly to Section IV, which is devoted to examining some of the solutions with emphasis on how they are affected by changes in the design parameters. We examine a number of different design parameters, particularly for buried channel devices. The accuracy of the solutions and the cost of obtaining them is considered in Section V. Finally, some mathematical details are contained in two appendixes.

II. THE POTENTIAL EQUATIONS

We consider CCD's in which the minority carriers are holes and the underlying substrate is n-type silicon. The analysis can be modified in an obvious way to describe the case where the minority carriers are electrons and the substrate is p-type silicon.

A buried channel CCD consists of a substrate of n-type silicon on top of which there is a layer of p-type silicon. The p-type layer is covered with a layer of SiO_2, and closely spaced electrodes are placed on top of the oxide layer. A schematic diagram of such a device is shown in Fig. 1 with some typical dimensions indicated. A surface CCD is the same, except that the p-type layer is missing.

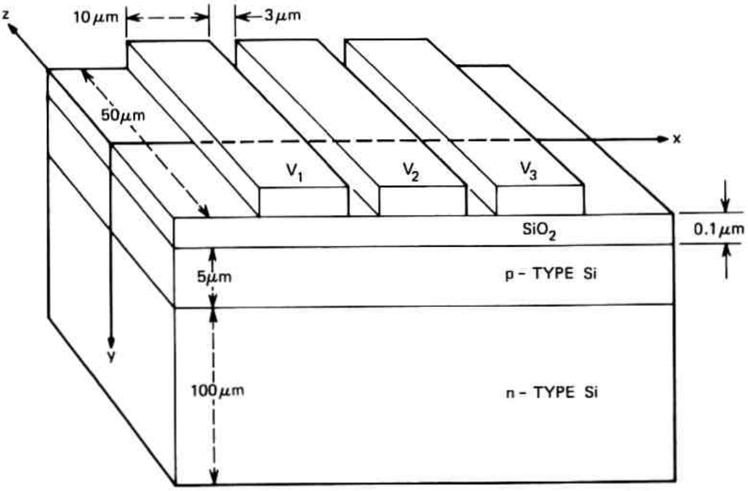


Fig. 1—A schematic diagram of a buried channel CCD.

We study here the static potential and fields in either a surface or buried channel CCD in the absence of mobile charge. Since the length in the z-direction of each plate is much greater than its width in the x-direction, near the center of the plates ($z=0$) the field is essentially two-dimensional; therefore, we treat the problem as two-dimensional.

We assume the bottom (n-type) substrate is infinitely thick. The field can penetrate into the substrate little beyond a depletion depth and, since for typical voltages the depletion depth ranges from 7 to 20 μm and the thickness of a typical device is 100 μm, this is a very good approximation.

It is assumed that there are gaps between the electrodes. We also make the approximation that the electrodes have zero thickness. Although this is a rather drastic simplification, we feel the essential effects of the gaps between the electrodes are still properly described. It will be seen later that electrodes of rectangular cross section could be studied, although at much greater cost. We further assume that the medium surrounding the electrodes has the same dielectric constant as the SiO_2. This is very reasonable, since in practice a CCD is covered with a dielectric coating. Two basic types of metalization are studied, single level and double level. In double-level metalization, two layers of electrodes are separated by an oxide layer. This is illustrated schematically in Fig. 2. We simulate this situation by assuming that the potential distribution in the gaps between the electrodes in the

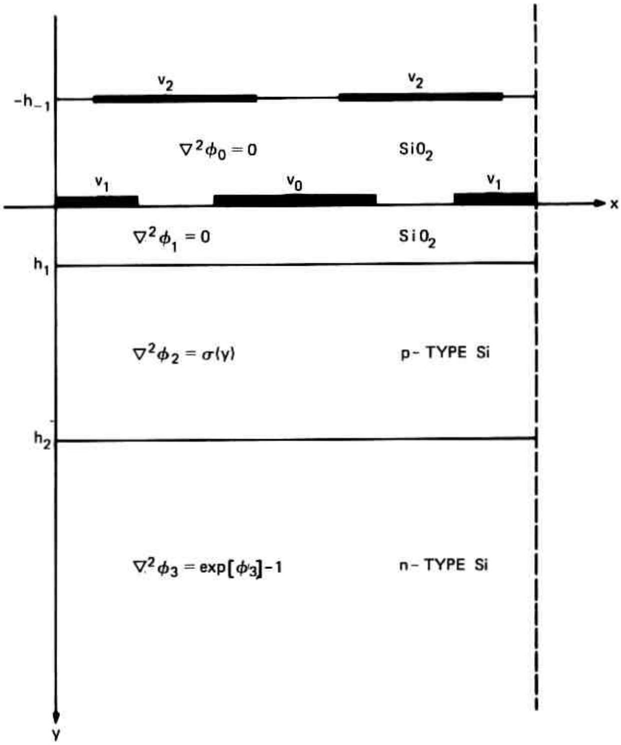


Fig. 2—A schematic diagram of one cell of a buried channel CCD with double-level metalization.

upper layer of electrodes is a known function. (Typically, the potential is assumed to vary piecewise linearly.) In single-level metalization, the upper level of electrodes is missing. Here we assume the dielectric coating over the electrodes is infinitely thick. Again, this is a reasonable assumption, since typically the field will have died out before reaching the surface of the dielectric coating.

Finally, we assume the structure to be periodic in the x-direction, which in the usual mode of operation is an excellent approximation.

The boundary value problem corresponding to our model of a buried channel CCD can be described by a system of partial differential equations which we wish to write in terms of dimensionless quantities. All dimensional quantities (measured in rationalized MKS units) will be starred, with the exception of a few obvious physical parameters. Corresponding unstarred quantities will be dimensionless. The physical parameters of the problem are ϵ_1 and ϵ_2, the permittivity of the oxide and silicon, respectively; $-e$, the charge of an electron; Boltzman's constant k; the absolute temperature T; the length of a unit cell in the device L^*; the separation of the two layers of metalization h^*_{-1}; and h^*_1 and h^*_2, the thickness of the oxide layer and the p-layer, respectively. The donor number density of the n-type substrate is N^*_D, which in the usual method of fabricating a CCD is a constant. However, we assume the acceptor number density in the p-layer is given by the expression[7]

$$N^*_A(y^*) = C^*_s \exp\left\{-\left(\frac{y^* - h^*_1}{h^*_2 - h^*_1}\right)^2 \ell n \frac{C^*_s}{N^*_D}\right\} - N^*_D, \tag{1}$$

where C^*_s is the number density of acceptor ions at the upper surface of the Si.

Now define the (dimensional) Debye length λ_D,

$$\lambda = (\epsilon_2 \mathrm{k}T/e^2 N^*_D)^{\frac{1}{2}}. \tag{2}$$

Then the dimensionless lengths are defined as

$$x = x^*/\lambda_D, \quad y = y^*/\lambda_D, \quad L = L^*/\lambda_D, \quad h_\alpha = h^*_\alpha/\lambda_D, \quad (\alpha = \pm 1, 2). \tag{3}$$

The dimensionless potential is related to the dimensional potential by

$$\varphi(x, y) = e\varphi^*(x^*, y^*)/\mathrm{k}T. \tag{4}$$

If we set

$$C_s = C^*_s/N^*_D, \tag{5}$$

then the dimensionless p-layer acceptor density, $\sigma(y)$, is

$$\sigma(y) = C_s \exp\left\{-\left(\frac{y - h_1}{h_2 - h_1}\right)^2 \ell n\, C_s\right\} - 1. \tag{6}$$

In the strip $0 \leqq x \leqq L$, let φ_0 denote the electrostatic potential above the oxide layer, $-\infty < y \leqq 0$ in the case of single-level metalization and $-h_{-1} \leqq y \leqq 0$ in the case of double-layer metalization. Further, let φ_1 denote the potential in the oxide layer, $0 \leqq y \leqq h_1$; φ_2 the potential in the p-type layer, $h_1 \leqq y \leqq h_2$; and φ_3 the potential in the n-type substrate (see Fig. 2). Then in the dimensionless form, the potential equations are

$$\nabla^2 \varphi_0 = 0, \qquad y \leqq 0, \tag{7}$$

$$\nabla^2 \varphi_1 = 0, \qquad 0 \leqq y \leqq h_1, \tag{8}$$

$$\nabla^2 \varphi_2 = \sigma(y), \qquad h_1 \leqq y \leqq h_2, \tag{9}$$

$$\nabla^2 \varphi_3 = \exp(\varphi_3) - 1, \qquad h_2 \leqq y < \infty, \tag{10}$$

where ∇^2 is the two-dimensional Laplace operator. The standard electromagnetic boundary conditions are as follows:

$$|\varphi_0(x, -\infty)| < \infty \text{ (single-level metalization), and} \tag{11}$$

$$\varphi_0(x, -h_{-1}) = U(x) \text{ (double-level metalization),} \tag{12}$$

where $U(x)$ is a given periodic function of period L, assuming on each electrode of the second level of metalization the constant voltage of the electrode and a specified potential between the electrodes. In reality, of course, the true potential in the gaps of the second level of metalization is unknown *a priori*. However, as indicated in Fig. 2, typically the semiconductor cannot "see" these gaps since they are shielded by the electrodes of the first level of metalization. Thus we simulate the exact boundary conditions in the gaps, most often by assuming the potential varies linearly from one electrode to another. We feel this is a good approximation, since we have performed calculations of the potential in the semiconductor with several different assumptions about the variation of the potential in the gaps, and the results were essentially identical. Further,

$$\varphi_0(x, 0) = V_j = \varphi_1(x, 0), \qquad (j = 1, 2, \cdots, p), \tag{13}$$

$$\varphi_0(x, 0) = \varphi_1(x, 0), \qquad \frac{\partial \varphi_0}{\partial y}(x, 0) = \frac{\partial \varphi_1}{\partial y}(x, 0) + \rho_g(x), \tag{14}$$

where V_j is the constant voltage of the jth electrode in the first level of metalization, eq. (13) holds on each of the p electrodes, and (14) holds in the gaps between the electrodes. Typically, $\rho_g(x) \equiv 0$, but in some cases it may describe a deliberately implanted surface charge in the gaps. In any event, $\rho_g(x)$ is a known function of x in the gaps, and $\rho_g(x) \equiv 0$ on the electrodes. Finally,

$$\varphi_1(x, h_1) = \varphi_2(x, h_1), \qquad \eta \frac{\partial \varphi_1}{\partial y}(x, h_1) = \frac{\partial \varphi_2}{\partial y}(x, h_1) + Q(x), \tag{15}$$

$$\varphi_2(x, h_2) = \varphi_3(x, h_2), \qquad \frac{\partial \varphi_2}{\partial y}(x, h_2) = \frac{\partial \varphi_3}{\partial y}(x, h_2), \tag{16}$$

$$\varphi_3(x, \infty) = 0, \tag{17}$$

and for all y

$$\varphi(0, y) = \varphi(L, y), \qquad \frac{\partial \varphi}{\partial x}(0, y) = \frac{\partial \varphi}{\partial x}(L, y). \tag{18}$$

In (15), $Q(x)$ is a known, periodic surface charge density, which may include deliberately implanted charges,[8] and

$$\eta = \epsilon_1/\epsilon_2. \tag{19}$$

All the boundary conditions (11) to (12) and (14) to (17) hold for $0 \leqq x \leqq L$.

The equations for the potential in a surface CCD are essentially the same, except the p-type layer is eliminated. We only give the analysis for the buried channel CCD. The results for the surface CCD can be obtained from those for the buried channel CCD by setting $\sigma(y) \equiv 0$, $h_1 = h_2$, and $\varphi_2 = \varphi_3$. In either case, the fields are obtained from the potential by

$$\mathbf{E} = - \nabla \varphi. \tag{20}$$

The results of I show that the system of eqs. (7) to (18) can be accurately solved by the method of finite differences only at great expense for even the simplest of devices. However, it was shown in I that, for the simpler problem studied there, the nonlinear boundary value problem could be replaced by a linear boundary value problem. This linear problem was solved analytically. It was then shown that under appropriate conditions the solution of the linear problem was an excellent approximation to the solution of the nonlinear problem in the p-type layer for a buried channel CCD and near the oxide-semiconductor interface for a surface CCD. The condition for the approximation to be a good one is, basically, that the potential along the line $y = h_2$ be large and negative. That condition holds in this problem, as we show later by example. Although the linear problem is much more complicated in this case because of the gaps between the electrodes and we have been unable to solve it analytically, we have been able to obtain good approximate solutions of it. For these reasons we now formulate the linear problem.

The linear equations are based on the depletion layer approximation, and we refer the reader to I and Ref. 4 for a detailed discussion. The linearization consists of replacing the single region $h_2 \leqq y < \infty$ by two regions, $h_2 \leqq y \leqq h_3 = h_2 + \hat{R}$ (the depletion layer) and $h_3 \leqq y < \infty$, and replacing the single nonlinear equation (10) by a different linear equation in each of these subregions (see Fig. 3).

$$\nabla^2\psi_0(x, y) = 0, \qquad y \leqq 0, \tag{21}$$

$$\nabla^2\psi_1(x, y) = 0, \qquad 0 \leqq y \leqq h_1, \tag{22}$$

$$\nabla^2\psi_2(x, y) = \sigma(y), \qquad h_1 \leqq y \leqq h_2, \tag{23}$$

$$\nabla^2\psi_3(x, y) = -1, \qquad h_2 \leqq y \leqq h_3 = h_2 + \hat{R}, \tag{24}$$

$$\nabla^2\psi_4(x, y) = \psi_4(x, y), \qquad h_3 \leqq y < \infty. \tag{25}$$

In addition to ψ_0, ψ_1, ψ_2, and ψ_3 satisfying boundary conditions (11) to

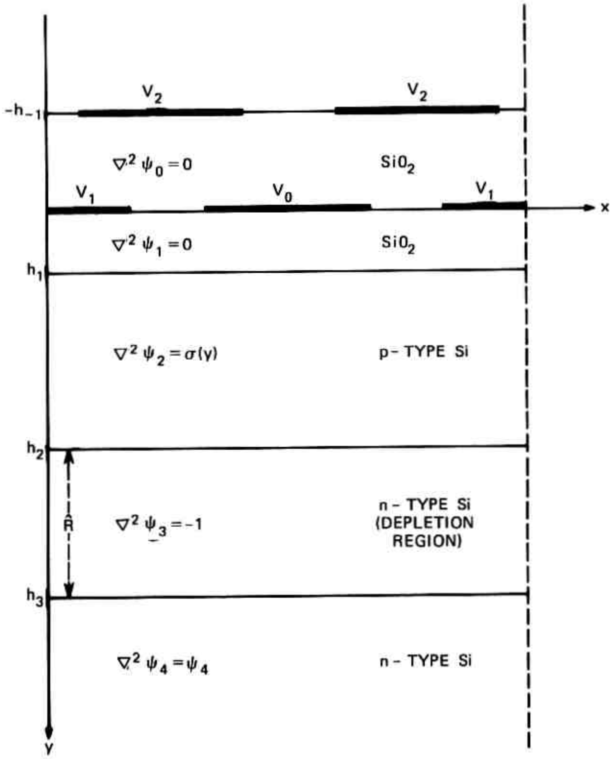


Fig. 3—A schematic diagram of the depletion layer approximation for one cell of a buried channel CCD with double-level metalization.

(16), we have the boundary conditions for $0 \leqq x \leqq L$

$$\psi_3(x,\, h_3) = \psi_4(x,\, h_3), \qquad \frac{\partial \psi_3}{\partial y}(x,\, h_3) = \frac{\partial \psi_4}{\partial y}(x,\, h_3), \tag{26}$$

$$\psi_4(x,\, \infty) = 0, \tag{27}$$

and the ψ_α, ($\alpha = 0,\, 1,\, 2,\, 3,\, 4$) all satisfy (18). The pseudodepletion depth $\hat{R}$ is best given by

$$\hat{R} = -\left(1 + h_2 - h_1 + \frac{h_1}{\eta}\right) + \Bigg[\left(h_2 - h_1 + \frac{h_1}{\eta}\right)^2 - 1 - 2V - \frac{2h_1}{\eta} Q_{ss} + 2\int_{h_1}^{h_2} \left(\xi - h_1 + \frac{h_1}{\eta}\right) \sigma(\xi) d\xi\Bigg]^{\frac{1}{2}}. \tag{28}$$

In (28),

$$Q_{ss} = \frac{1}{L} \int_0^L Q(x) dx$$

and V is the average electrode voltage. As shown in I and Ref. 4, ψ_0, ψ_1, and ψ_2 are quite insensitive to the choice of $\hat{R}$, and an optimal choice of $\hat{R}$ tends to minimize $\sup_{0 \leqq x \leqq L} |\psi_4(x, h_3) + 1|$.

III. SOLUTION OF THE LINEARIZED POTENTIAL EQUATIONS

The method we shall use to solve the linear system of elliptic eqs. (21) to (27) has much in common with previous work[9] on the classical problem of a single linear elliptic boundary value problem on a simply connected domain in the plane. The technique used in Ref. 9 is quite simple: Construct a family of particular solutions of the partial differential equation and, using a finite linear combination of these particular solutions, obtain a Chebyshev fit to the boundary conditions at a finite number of points on the boundary of the domain. It was shown that, as more and more particular solutions are taken in the linear combination and more and more points are chosen in the fit on the boundary, the linear combination converges to the true solution.

In this paper we construct a family of particular solutions of (21) to (27). These solutions depend linearly on a finite number N of parameters and satisfy *all* the boundary and interface conditions *except* that they do not assume the correct voltages on the electrodes at $y = 0$. We then complete the analogy with Ref. 9 by picking M points on the plates, x_i, $1 \leqq i \leqq M$, where $M \geqq N$, and force the potential to take on the correct value at the points x_i in the least-squares sense.

We obtain the family of particular solutions in the form of Fourier series. We assume as given the Fourier series expansions of $U(x)$ and $Q(x)$:

$$U(x) = \tfrac{1}{2}\alpha_0 + \sum_{n=1}^{\infty} G_n(x), \tag{29}$$

$$Q(x) = \tfrac{1}{2}\zeta_0 + \sum_{n=1}^{\infty} \Phi_n(x), \tag{30}$$

where

$$G_n(x) = \alpha_n \cos \lambda_n x + \beta_n \sin \lambda_n x, \tag{31}$$

$$\Phi_n(x) = \zeta_n \cos \lambda_n x + \xi_n \sin \lambda_n x, \tag{32}$$

and

$$\lambda_n = (2n\pi)/L. \tag{33}$$

Further, from I, we can write down formal expressions for ψ_1, ψ_2, ψ_3, and ψ_4 which satisfy (22) to (25) and boundary conditions (15), (16), (18), (26), and (27). Let

$$F_n(x) = a_n \cos \lambda_n x + b_n \sin \lambda_n x, \tag{34}$$

$$E_\pm = 1 \pm 1/\eta, \tag{35}$$

$$\Lambda_n^\pm = 1 \pm (1 + \lambda_n^{-2})^{\frac{1}{2}}, \tag{36}$$

$$M_n(y) = \{E_+\Lambda_n^+ + E_-\Lambda_n^- e^{-2\lambda_n(h_3-h_1)}\}e^{-\lambda_n y} + \{E_-\Lambda_n^+ e^{-2\lambda_n h_1} + E_+\Lambda_n^- e^{-2\lambda_n h_3}\}e^{\lambda_n y}, \tag{37}$$

$$L_n(y) = 2\{\Lambda_n^+ e^{-\lambda_n y} + \Lambda_n^- e^{-\lambda_n(2h_3-y)}\}, \tag{38}$$

where η is given by (19), λ_n by (33), and a_0, a_n, b_n $(n = 1, 2, \cdots)$ are unknown constants. Then these expressions are

$$\psi_1(x, y) = (A\bar{a}_0 + B) + (C\bar{a}_0 + D)y + \sum_{n=1}^{\infty} \left\{ F_n(x) \frac{M_n(y)}{M_n(0)} + \Phi_n(x) \frac{L_n(h_1)}{M_n(0)} \frac{\sinh \lambda_n y}{\eta\lambda_n} \right\}, \tag{39}$$

$$\psi_2(x, y) = \frac{1}{2}\left[\bar{a}_0(1 + h_3 - h_2) - (h_3 - h_2)^2 - (\bar{a}_0 + 2h_2 - 2h_3)(y - h_2) + 2\int_{h_2}^{y} (y - \xi)\sigma(\xi)d\xi \right] + \sum_{n=1}^{\infty} \left\{ F_n(x) + \Phi_n(x) \frac{\sinh \lambda_n h_1}{\eta\lambda_n} \right\} \frac{L_n(y)}{M_n(0)}, \tag{40}$$

$$\psi_3(x, y) = \tfrac{1}{2}[\bar{a}_0(1 + h_3 - y) - (y - h_3)^2] + \sum_{n=1}^{\infty} \left\{ F_n(x) + \Phi_n(x) \frac{\sinh \lambda_n h_1}{\eta\lambda_n} \right\} \frac{L_n(y)}{M_n(0)}, \tag{41}$$

$$\psi_4(x, y) = \tfrac{1}{2}\bar{a}_0 e^{-(y-h_3)} + 4\sum_{n=1}^{\infty} \left\{ F_n(x) + \Phi_n(x) \frac{\sinh \lambda_n h_1}{\eta\lambda_n} \right\} \cdot \frac{\exp[-\sqrt{1 + \lambda_n^2}(y - h_3) - \lambda_n h_3]}{M_n(0)}, \tag{42}$$

where

$$A = \frac{1}{2}\left(1 + h_3 - h_1 + \frac{h_1}{\eta}\right), \tag{43}$$

$$B = \int_{h_1}^{h_2} \left(\xi - h_1 + \frac{h_1}{\eta}\right)\sigma(\xi)d\xi - \frac{1}{2}\left[(h_3 - h_2)(h_3 + h_2 - 2h_1) + \frac{h_1}{\eta}(\zeta_0 + 2h_3 - 2h_2)\right], \tag{44}$$

$$C = -1/(2\eta), \tag{45}$$

$$D = \left[\zeta_0 + 2(h_3 - h_2) - 2\int_{h_1}^{h_2} \sigma(\xi)d\xi\right] \Big/ (2\eta), \tag{46}$$

and

$$\bar{a}_0 = (\tfrac{1}{2}a_0 - B)/A. \tag{47}$$

It should be noted that

$$\psi_1(x, 0) = \tfrac{1}{2}a_0 + \sum_{n=1}^{\infty} F_n(x). \tag{48}$$

In the case of single-level metalization we can write down an expression for a solution of (21) in $y \leqq 0$ which satisfies boundary conditions (11):

$$\psi_0(x) = \tfrac{1}{2}a_0 + \sum_{n=1}^{\infty} F_n(x)e^{\lambda_n y}. \tag{49}$$

In the case of double-level metalization, a solution of (21) in $-h_{-1} \leqq y \leqq 0$ satisfying boundary condition (12) is

$$\psi_0(x, y) = \frac{1}{2}\left(1 + \frac{y}{h_{-1}}\right)a_0 + \sum_{n=1}^{\infty} F_n(x)\frac{\sinh \lambda_n(y + h_{-1})}{\sinh \lambda_n h_{-1}} - \frac{\alpha_0}{2}\left(\frac{y}{h_{-1}}\right) - \sum_{n=1}^{\infty} G_n(x)\frac{\sinh \lambda_n y}{\sinh \lambda_n h_{-1}}. \tag{50}$$

Note that these solutions have been constructed so that from (48) and either (49) or (50), for $0 \leqq x \leqq L$,

$$\psi_0(x, 0) = \psi_1(x, 0). \tag{51}$$

[Note also that, term by term, (49) is the limit as $h_{-1} \to \infty$ of (50).]

Equations (34) to (50) contain expressions for ψ_α, $0 \leqq \alpha \leqq 4$, which satisfy the differential equations (21)–(25) and all the required boundary and interface conditions except condition (13) on the plates and the normal derivative condition of (14) in the gaps. These particular solutions contain the unknown parameters a_0, a_n, b_n, $(n = 1, 2, \cdots)$, which remain to be determined. At this point it might be assumed that the series should be truncated at some $n = N$ and the $2N + 1$ coefficients a_0, a_n, b_n, $n = 1, 2, \cdots, N$, be determined directly by making a least-squares fit to the remaining boundary conditions. However, it can be shown[10] that if $x_{j-1} < x_j$ are the end points of an electrode on $y = 0$ then for $x_{j-1} < x < x_j$, and x near x_j, say, $\partial\psi_0/\partial y(x, 0)$ will behave like $(x_j - x)^{-\frac{1}{2}}$ plus a power series in $(x_j - x)^{\frac{1}{2}}$. This implies that the Fourier series for $\psi_0(x, 0)$ converges very slowly. In fact, we have found that it is often necessary to take up to 2000 terms in the series to represent $\psi_0(x, 0)$ adequately. This makes it impractical to use the Fourier coefficients themselves as the parameters to be determined directly by a least-squares process.

Instead, we used the following technique. We approximated the charge density on $y = 0$ by a finite sum of known functions

$$\rho(x) = \rho_g(x) + \sum_{j=1}^{N} \rho_j p_j(x). \tag{52}$$

The functions $p_j(x)$ are zero except on the electrodes and are of several types, as shown in Fig. 4. The function $p_0(x)$ is a periodic, triangular spline for the ends of the device, as shown in Fig. 4a. Corresponding to the edge of each plate, there is a discontinuous triangular spline, Figs. 4b and 4c, and singular splines of the form $|x - x_k|^{-\frac{1}{2}}$, Figs. 4d and 4e. The remaining $p_j(x)$, whose supports lie wholly interior to the electrodes, are triangular splines as shown in Fig. 4f.

Now each unknown parameter a_0, a_n, b_n $(n = 1, 2, \cdots)$ is determined as a linear sum of the N parameters p_j, $1 \leqq j \leqq N$, by equating $\rho(x)$, given in (52) with $\partial\psi_0/\partial y(x, 0) - \partial\psi_1/\partial y(x, 0)$. In the case of single-level metalization, from (39) and (49),

$$\frac{\partial\psi_0}{\partial y}(x, 0) - \frac{\partial\psi_1}{\partial y}(x, 0) = -(C\bar{a}_0 + D) - \sum_{n=1}^{\infty} F_n(x)\lambda_n E_n - \sum_{n=1}^{\infty} \Phi_n(x) \frac{L_n(h_1)}{\eta M_n(0)}, \tag{53}$$

where

$$E_n = \frac{M_n'(0)}{\lambda_n M_n(0)} - 1. \tag{54}$$

It should be noted from (33), (35), (37), and (38) that, as $n \to \infty$,

$$E_n \approx -2, \qquad \frac{L_n(h_1)}{\eta M_n(0)} \approx \frac{2}{1 + \eta} e^{-\lambda_n h_1}. \tag{55}$$

In the case of double-level metalization, from (39) and (50),

$$\frac{\partial\psi_0}{\partial y}(x, 0) - \frac{\partial\psi_1}{\partial y}(x, 0) = \frac{a_0 - \alpha_0}{2h_{-1}} - (C\bar{a}_0 + D) - \sum_{n=1}^{\infty} F_n(x)\lambda_n H_n - \sum_{n=1}^{\infty} G_n(x) \frac{\lambda_n}{\sinh \lambda_n h_{-1}} - \sum_{n=1}^{\infty} \Phi_n(x) \frac{L_n(h_1)}{\eta M_n(0)}, \tag{56}$$

where

$$H_n = \frac{M_n'(0)}{\lambda_n M_n(0)} - \operatorname{ctnh}(\lambda_n h_{-1}). \tag{57}$$

It again follows from (33) and (37) that, as $n \to \infty$,

$$H_n \approx -2. \tag{58}$$

For a periodic function $f(x)$, of period L, we denote by $c_n[f]$ and

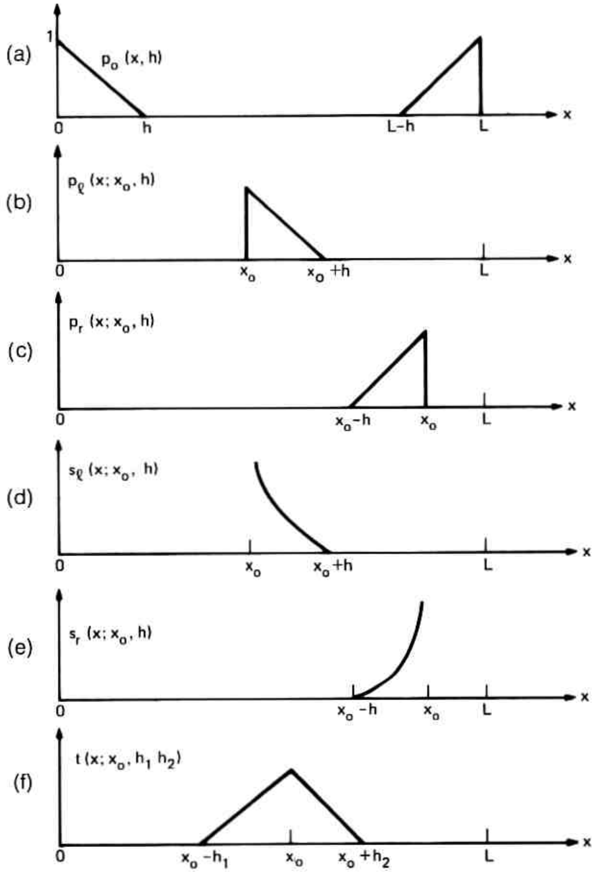


Fig. 4—The splines used to represent the charge density on the electrodes.

$s_n[f]$ the cosine and sine Fourier coefficients of f:

$$c_n[f] = \frac{2}{L}\int_0^L f(x)\cos(\lambda_n x)dx,\ s_n[f] = \frac{2}{L}\int_0^L f(x)\sin(\lambda_n x)dx. \tag{59}$$

Then, from (52), the Fourier series for $\rho(x)$ is

$$\begin{aligned}\rho(x) = \frac{1}{2}\left\{c_0[\rho_g] + \sum_{j=1}^{N}\rho_j c_0[p_j]\right\} \\ + \sum_{n=1}^{\infty}\left\{c_n[\rho_g] + \sum_{j=1}^{N}\rho_j c_n[p_j]\right\}\cos\lambda_n x \\ + \left\{s_n[\rho_g] + \sum_{j=1}^{N}p_j s_n[p_j]\right\}\sin\lambda_n x.\end{aligned} \tag{60}$$

In Appendix A we give $c_n[p_j]$ and $s_n[p_j]$ for the various functions $p_j(x)$. If we now equate (53) with (60), we can obtain the a_n and b_n as linear functions of the ρ_j in the case of single-level metalization:

$$a_0 = \frac{2(BC - AD)}{C} - \frac{A}{C}\left\{c_0[\rho_g] + \sum_{j=1}^{N} \rho_j c_0[p_j]\right\}, \tag{61}$$

$$\begin{aligned} a_n = &- [\zeta_n L_n(h_1)]/[\eta\lambda_n E_n M_n(0)] \\ &- \left\{c_n[\rho_g] + \sum_{j=1}^{N} \rho_j c_n[p_j]\right\} \Big/ (\lambda_n E_n), \end{aligned} \tag{62}$$

$$\begin{aligned} b_n = &- [\xi_n L_n(h_1)]/[\eta\lambda_n E_n M_n(0)] \\ &- \left\{s_n[\rho_g] + \sum_{j=1}^{N} \rho_j s_n[p_j]\right\} \Big/ (\lambda_n E_n). \end{aligned} \tag{63}$$

Similarly, if (56) is equated with (60), we obtain the a_n and b_n in the case of double-level metalization:

$$\begin{aligned} a_0 = &\frac{2h_{-1}(BC - AD) - \alpha_0 A}{(h_{-1})C - A} - \frac{Ah_{-1}}{(h_{-1})C - A} \\ &\cdot \left\{c_0[\rho_g] + \sum_{j=1}^{N} \rho_j c_0[p_j]\right\}, \end{aligned} \tag{64}$$

$$\begin{aligned} a_n = &- \frac{\alpha_n}{H_n \sinh (h_{-1}\lambda_n)} - \frac{\zeta_n L_n(h_1)}{\eta\lambda_n H_n M_n(0)} \\ &- \left\{c_n[\rho_g] + \sum_{j=1}^{N} p_j c_n[p_j]\right\} \Big/ (\lambda_n H_n), \end{aligned} \tag{65}$$

$$\begin{aligned} b_n = &- \frac{\beta_n}{H_n \sinh (h_{-1}\lambda_n)} - \frac{\xi_n L_n(h_1)}{\eta\lambda_n H_n M_n(0)} \\ &- \left\{s_n[\rho_g] + \sum_{j=1}^{N} \rho_j s_n[p_j]\right\} \Big/ (\lambda_n H_n). \end{aligned} \tag{66}$$

Equations (39) to (42) and either (49) or (50), with the a_n and b_n defined by eqs. (61) to (63) or (64) to (66), respectively, define solutions to eqs. (21) to (25) which satisfy boundary conditions (11) or (12), (14) to (16), (26), and (27). They do not, however, assume the correct values on the electrodes at $y = 0$; i.e., (13) is not satisfied. These solutions depend linearly on the N unknown constants ρ_j, $1 \leqq j \leqq N$, and of course on the choice of functions $p_j(x)$ used to describe the charge density on the plates. Having picked the $p_j(x)$ described earlier, M_j points are chosen on the jth electrode, $x_i^{(j)}$, $1 \leqq i \leqq M_j$, with

$$M = \sum_{j=1}^{p} M_j \geqq N.$$

Then the expression

$$\sum_{j=1}^{p} \sum_{i=1}^{M_j} [\psi_0(x_i^{(j)}, 0) - V_j]^2 \tag{67}$$

is minimized with respect to the ρ_α.

It has been observed that near the edge of an electrode, where the potential has square-root behavior, the fitting points $x_i^{(j)}$ should be spaced quadratically closer together as the edge of the electrode is approached. If $x = e$ is an edge of the jth plate, then we distributed the points near this edge by picking $b \neq e$ and an integer $m < M_j/2$ and setting

$$z_j = b + (e - b) \cos \frac{(j-1)\pi}{2(m-1)}, \qquad (1 \leqq j \leqq m). \tag{68}$$

The points z_j were then used as the fitting points x_i near the edge of the plate. Away from the edges of the plate, the fitting points were uniformly distributed.

We have assumed the existence of a bounded solution $\psi(x, y)$ to the linearized problem. In Appendix B we show that if $\psi(x, y)$ is the true solution of the linearized problem and $\psi^a(x, y)$ is one of our approximate solutions, then the error $\psi(x, y) - \psi^a(x, y)$ is bounded at every point by the maximum error on the electrodes. Since the true solution is known on the electrodes, this provides us with *a posteriori* error bounds. We will make use of this important point later in evaluating the quality of our approximate solutions.

The technique described in this section can be formulated in a rather general setting and, we believe, can be applied to many problems of interest in physics and engineering. It has been used by Morrison et al. in a study of microwave scattering by deformed raindrops.[11] Assume that a problem can be separated into two parts: Input data and a governing system of partial differential equations (PDE's), with possible interface conditions, which determine the solution when given the input data. Further, assume that linear families of particular solutions to the PDE's can be found. For example, these may be constructed by separation of variables, Fourier series, Green's Theorems, etc. Finally, assume that by linearly parameterizing some unknowns of the solution (for our problem, the charge distribution on the plates) we can obtain particular solutions to both the PDE's and the interface conditions. Then one could use some fitting procedure, a discrete least-squares fit, for example, to force the linear family of particular solutions to the governing system to have approximately the same input data as the desired solution. This generates a solution of exactly the same governing system, but with input data which differs by a known, and hopefully small, amount from the desired input data. For all practical purposes, this gives an effective bound on the error in the computed approximate solution. For example, if the desired input is only known to 1 percent, because of experimental error in measuring it, then any solution generated by the above procedure which corresponds to the desired input data perturbed by at most 1 percent cannot, on the basis of comparing inputs, be distinguished from the true solution of the problem.

Also, in many cases, one can use the Maximum Principle, conservation of energy, or some other basic principle to give sharp, rigorous bounds on the error in such an approximate solution in terms of how well it satisfies the given input data. We do this for our problem in Appendix B. This is a very great improvement over the standard discretization methods for solving such problems. Those methods generally give an approximate solution to an approximate system of equations, but with exactly the given input data, with the result that it is very difficult to estimate reliably what the true error is for a given approximate solution.

IV. THE POTENTIALS AND FIELDS IN SOME SPECIFIC CCD'S

Using the method described in Section III, we have evaluated approximately the solutions of eqs. (21) to (25) for a number of different plate configurations and design parameters, and we present some of these results graphically in this section.

We have assumed in each case that the n-type substrate doping is $N_D^* = 10^{14}$ cm^{-3}, that $\epsilon_2/\epsilon_0 = 12$, where ϵ_0 is the permittivity of free space, that $\epsilon_1/\epsilon_2 = \frac{1}{3}$, and that $Q(x) \equiv 0$, i.e., there is no trapped or implanted charge at the oxide-semiconductor interface. Then at $T = 300°$K, the Debye length is $\lambda_0 = 0.415$ μm. In addition, we have used the factor $(\mathrm{k}T/e) = 0.025$ V to convert dimensionless potentials to volts, and the factor $(\mathrm{k}T/e\lambda_D) = 600$ V/cm to convert dimensionless fields to volts per centimeter. In each example involving a buried channel CCD, we assume that the acceptor number density in the p-layer is given by (1), [(6)] with $C_s^* = 4.6 \times 10^{15}$ cm^{-3} [$C_s = 46$]. In each such case, this corresponds to an average number density of acceptor atoms of 2×10^{15} cm^{-3} [see eq. (2.5) of I].

In I we investigated the effects of changing the p-layer doping and thickness, and so here we concentrate mainly on the effects of gap width, plate potential, and the separation of the levels of metalization.

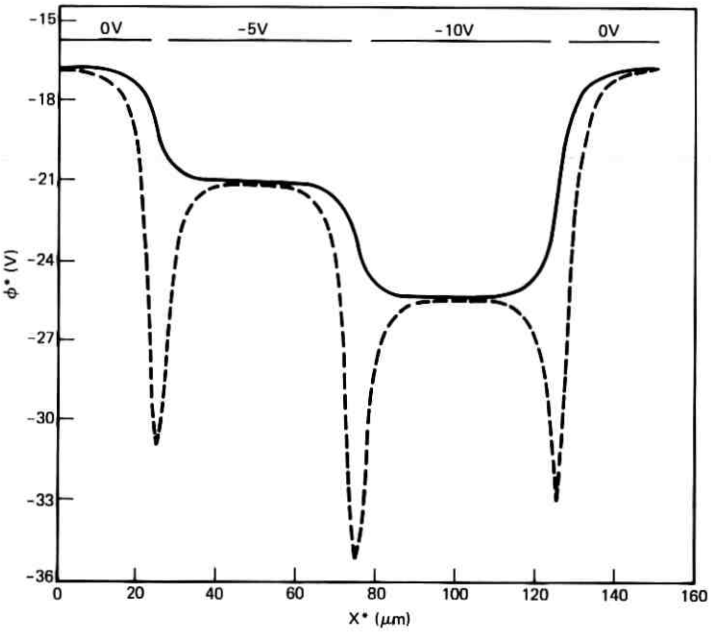


Fig. 5—The channel potential φ^* plotted as a function of x^* for a three-phase buried channel CCD. The 45-μm-wide electrodes are at 0, −5, and −10 V; the gaps are 5 μm wide; $h_1^* = 0.1$ μm; $h_2^* - h_1^* = 5$ μm; and $C_s^* = 4.6 \times 10^{15}$ cm^{-3}. The dashed curve is for no surface charge implanted in the gaps; the solid curve is for $\rho_g^*/e = 0.8 \times 10^{12}$ cm^{-2} implanted in the gaps.

In Figs. 5 and 6 we show some properties of a three-phase buried channel CCD with single-level metalization. The electrodes are 45 μm wide, and the gaps between them are 5 μm wide. The p-layer is 5 μm thick ($h_2^* - h_1^* = 5$ μm), and the oxide layer is 0.1 μm thick ($h_1^* = 0.1$ μm). The region $y < 0$ is assumed to be filled with SiO$_2$. The potentials on the electrodes are 0, −5, and −10 V, as shown. The dashed curve in Fig. 5 shows the channel potential φ^* (that is, the value of the potential at the potential minimum in the p-layer) as a function of x^* when there is no implanted surface charge in the gaps between electrodes $[\rho_g^*(x^*) \equiv 0]$. This curve illustrates one early difficulty encountered in the design of buried channel CCD's, namely the large potential well under the gap between the plates. A CCD with almost these same parameters was constructed[5] and did not work because of the variable amounts of charge trapped in these wells. In the remainder of this section we discuss a number of possible methods of eliminating this potential well in the gaps between the plates.

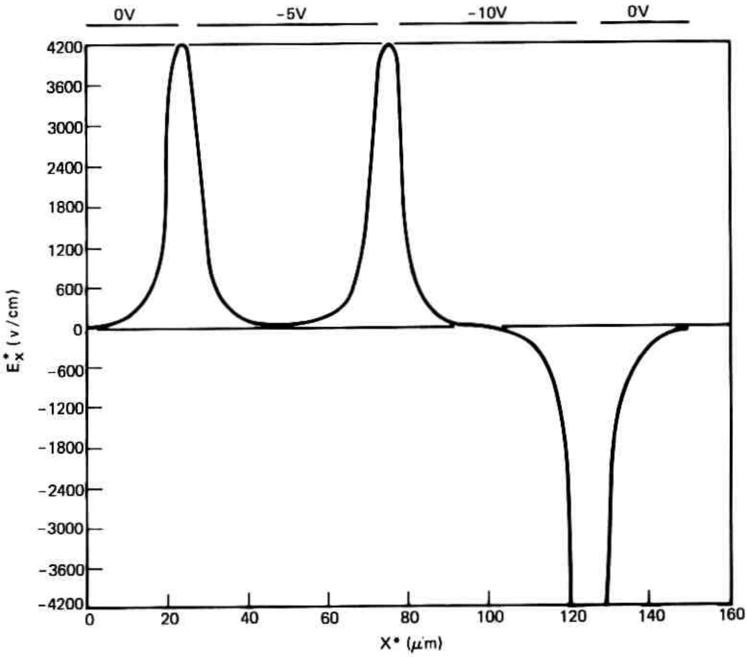


Fig. 6—The channel field $-\partial\varphi^*/\partial x^*$ plotted as a function of x^* for the CCD of Fig. 5 with $\rho_g^*/e = 0.8 \times 10^{12}$ cm^{-2} implanted in the gaps.

An operating buried channel CCD has been reported[12] in which the gaps between the electrodes have been filled with a resistive material so that the potential drop between the electrodes is essentially linear. This CCD was also discussed in I, and it was shown there that the potential wells are eliminated. Another technique for eliminating the potential wells is to implant a layer of positive surface charge in the gap between the electrodes. (Other schemes for eliminating this problem are discussed in the literature.[13]) The solid curve in Fig. 5 shows the channel potential in the same three-phase CCD after a uniform surface charge density, $\rho_g^*(x^*)/e = 0.8 \times 10^{12}$ cm^{-2} $[\rho_g(x) = 578]$, has been implanted in the gaps between the electrodes. Note that

$$\int_{h_1^*}^{h_2^*} N_A^*(y^*)dy^* = 10^{12} \text{ cm}^{-2}.$$

This technique should also eliminate the potential wells under the gaps. In Fig. 6 we plot the channel field $E_x^* = -\partial\varphi^*/\partial x^*$ (that is, the field at the potential minimum in the p-layer) as a function of x^*.

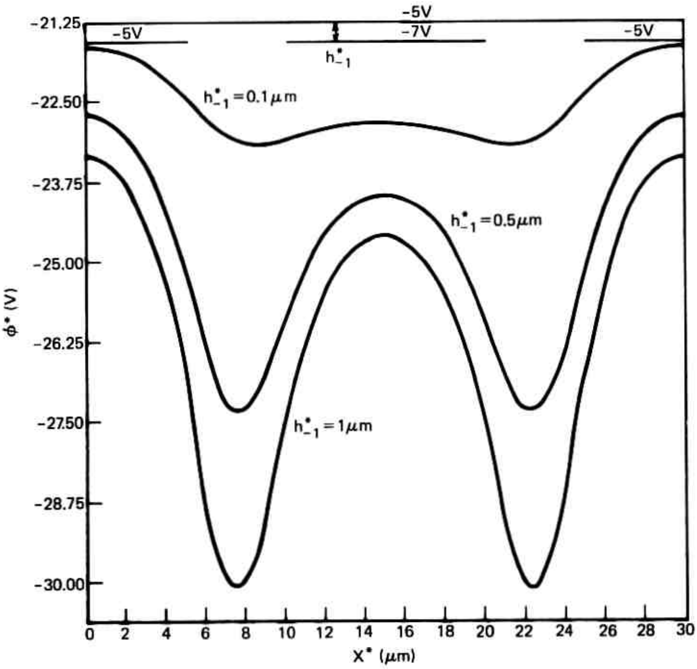


Fig. 7—The channel potential in a buried channel CCD with double-level metalization. The lower level electrodes are 10 μm wide with 5-μm gaps and are at potentials of −5 and −7 V. The upper level is a single electrode at −5 V; $h_1^* = 0.1$ μm; $h_2^* - h_1^* = 5$ μm; $C_s^* = 4.6 \times 10^{15}$ cm^{-3}; and the separation of the metalization levels is $h_{-1}^* = 0.1$, 0.5, and 1.0 μm.

This shows that there are substantial fields in the gap between the −5 and −7 V electrodes, but the field penetration under the electrodes is not too good.

In Figs. 7 to 9 we investigate the possibility of eliminating the potential wells by the use of double-level metalization, in which the upper level of metal is a single continuous piece covering the entire channel and has a dc potential applied to it. The presence or absence of the potential wells is mainly a local phenomenon and can be studied by considering just two adjacent electrodes in the lower level of metalization. Thus, in the interests of economy we consider a model CCD in which alternate electrodes of the lower level are at the same voltage. These plates are 10 μm wide and the gaps between them are 5 μm wide. The oxide layer between the first level of electrodes and the p-layer is 0.1 μm thick ($h_1^* = 0.1$ μm) and the p-layer is 5 μm thick ($h_2^* - h_1^* = 5$ μm). For the CCD of Fig. 7, the electrodes on the lower

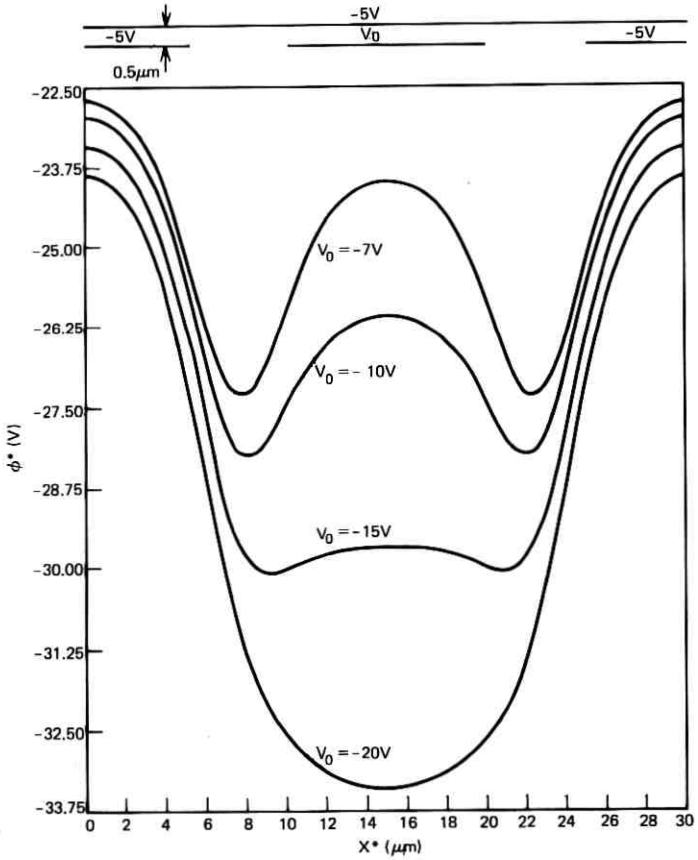


Fig. 8—The channel potential in the CCD of Fig. 7 with the separation of the metalization levels held fixed at $h^*_{-1} = 0.5$ μm and the potential of the center, lower-level electrode taking the values −7, −10, −15, and −20 V.

level are at a potential of −5 and −7 V, and the upper level consists of a single electrode, covering the whole device, which is at a potential of −5 V. In Fig. 7 we plot the channel potential for three different separations of the levels of metalization, $h^*_{-1} = 0.1, 0.5$, and 1 μm. Even with $h^*_{-1} = 0.1$ μm, there is still a very slight well in the gaps.

We next see what happens if we hold the separation of the levels of metalization at $h^*_{-1} = 0.5$ μ and change the voltage, v_0, on the middle electrode in the lower level. In Fig. 8 we plot the resulting channel potential, φ^*, as a function of x^* for $v_0 = -7, -10, -15$, and -20 V. From these graphs we see that a potential difference of 15 V between

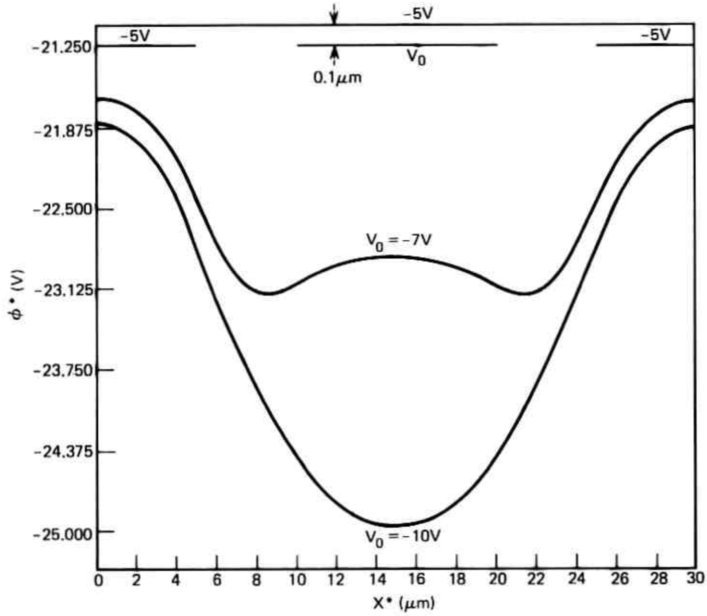


Fig. 9—The channel potential in the CCD of Fig. 7 with the separation of the metalization levels held fixed at $h^*_{-1} = 0.1$ μm and the potential of the center, lower-level electrode taking the values −7 and −10 V.

neighboring electrodes on the lower level will insure the absence of potential wells in the gaps.

In Fig. 9 we plot the channel potential for the same device but with $h^*_{-1} = 0.1$ μm and for $v_0 = -7$ and -10 V. For this small separation of the levels of metalization, a 5-V potential difference between neighboring electrodes eliminates potential wells in the gaps.

In Figs. 10 and 11 we show some effects of gap width. We consider first a buried channel CCD with double-level metalization. The p-layer is 5 μm thick ($h^*_2 - h^*_1 = 5$ μm), the oxide layer between the first level of electrodes and the p-layer is 0.1 μm thick ($h^*_1 = 0.1$ μm), and the layer between the two levels of electrodes is 0.5 μm ($h^*_{-1} = 0.5$ μm). The upper level of electrodes consists of a single electrode at a potential of −5 V. The lower level consists of electrodes 10 μm wide and, as shown in Fig. 10, they are alternatively at potentials of −5 and −7 V. Curves of the channel potentials are plotted for three different gap widths between plates: 5, 1, and 0 μm. (The 0-gap curve was calculated by the methods of I.) The x^* scale for the three curves are different, but are chosen so the centers of the gaps coincide. With 5-μm-wide gaps,

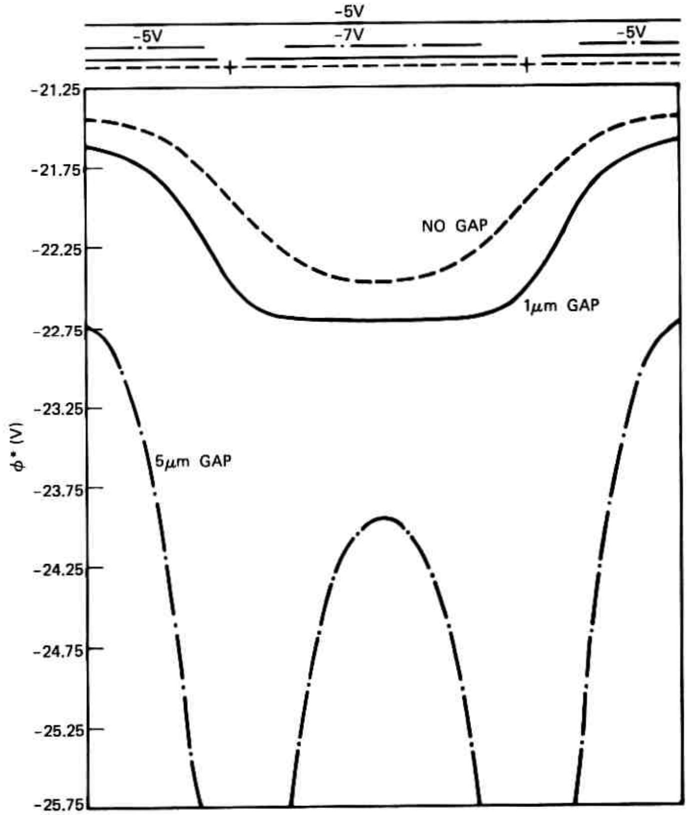


Fig. 10—The channel potential in the CCD of Fig. 7 with the separation of the metalization levels held fixed at $h^*_{-1} = 0.5$ μm, and the gap width taking the values 5, 1, and 0 μm.

there are large potential wells, as we saw in Figs. 7 and 8. However, by reducing the gap width to 1 μm, the potential wells are essentially eliminated. The curve for zero electrode separation is included to show that it is a good approximation to the channel potential in cases of small electrode separation.

Finally, in Fig. 11, we plot the potential along the oxide-semiconductor interface ($y^* = h^*_1$) for two surface CCD's. In each case, the oxide layer is 0.1 μm thick ($h^*_1 = 0.1$ μm) and the region $y^* < 0$ is assumed to be filled with SiO_2. Also, in both cases, the electrodes are 10 μm wide and are held at alternate potentials of −5 and −7 V. In one case the gap between electrodes is 1 μm, while in the other case there are no gaps between electrodes. (The zero gap curve was calcu-

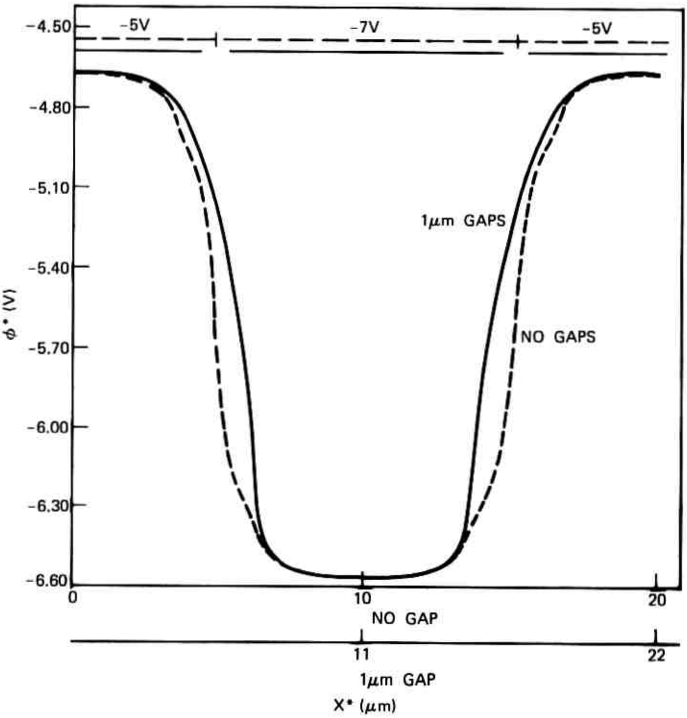


Fig. 11—The potential $\varphi^*(x^*, h_1^*)$ plotted as a function of x^* for a surface CCD with single-level metalization; $h_1^* = 0.1$ μm; 10-μm-wide electrodes held at −5 and −7 V; and with gaps between electrodes of 1 and 0 μm.

lated by the method of I.) Again the x^* scales differ, but the centers of the gaps coincide. Except in the region between the plates, the two curves coincide closely.

V. COMMENTS ON ACCURACY AND COST

We considered in some detail in I how well the solutions of the linearized equations (21) to (25) approximate the solutions of the nonlinear equations (7) to (10). It was shown there and in Ref. 4 that as long as $\max_{0 \leqq x \leqq L} \psi_1(x, h_1) \leqq -160$, and $|\psi_4(x, h_3) + 1| < 10, 0 \leqq x \leqq L$, $[|\psi_3(x, h_2) + 1| < 10$ for surface devices$]$ then the solution of the linearized problem approximates the solution of the nonlinear problem to within several percent in the p-layer for buried channel devices (near the oxide-semiconductor interface for surface devices). In addition, we are interested in examining the accuracy of the approximate solutions of the linearized equations. As we stated in Section III (and prove in Appendix B), the difference between the exact solution and the approximate solution, $\delta^*(x^*, y^*) = \psi^*(x^*, y^*) - \psi^{a*}(x^*, y^*)$, is bounded everywhere by its maximum value on the electrodes.

As typical examples, consider the curves in Figs. 10 and 11. For all three curves in Fig. 10, we found that $\psi_1(x, h_1) \leqq -345$ and $|\psi_4(x, h_2) + 1| < 2.75$ for $0 \leqq x \leqq L$. For the two curves of Fig. 11, we found that $\psi_1(x, h_1) < -186$ and $|\psi_3(x, h_2) + 1| \leqq 1$ for $0 \leqq x \leqq L$. Furthermore, for the curves of Fig. 10, a search of the electrodes showed that, for 5-μm gaps between electrodes, $\max |\delta^*(x^*, y^*)| \leqq 0.29$ V and, for 1-μm gaps, $\max |\delta^*(x^*, y^*)| \leqq 0.18$ V. (The no-gaps curve was calculated by the methods of I.) These correspond to maximum errors of the channel potential of 1.3 and 0.83 percent respectively. For the curve for the surface CCD with 1-μm gaps between the electrodes a search of the electrodes showed that $\max |\delta^*(x^*, y^*)| \leqq 0.4$ V. (The no-gaps curve was again calculated by the methods of I.) This corresponds to a maximum percentage error of 0.86 percent.

For the curves of the buried channel CCD of Fig. 10, we have undoubtedly overestimated the error in the channel for the following reasons. It can be shown that the error in each coefficient a_n and b_n appearing in (34) can be expressed as an integral over the electrodes of the error $\delta(x, y)$ times a weight function. The sign of $\delta(x, y)$ oscillates on the electrodes, and so one would expect the error in the lower-order coefficients to be quite small. Furthermore, an examination of (34) to (38) and (40) shows that, for the parameters involved, only the first ten terms in (40) contribute significantly to the channel potential.

To present some idea of the cost of running these programs, the calculation of the solution for the case of 5-μm gaps in the curves of Fig. 10 took 253 seconds and used 40 K of core, the case of 1-μm gaps took 258 seconds and 40 K of core. Calculation of the solution for the 1-μm gap case of Fig. 11 took 263 seconds and used 40 K of core. By comparison, the two corresponding no-gap solutions, obtained by the methods of I, took 16 seconds and 32 seconds, respectively, and both solutions required 40 K of core.

VI. ACKNOWLEDGMENTS

The authors take pleasure in thanking G. E. Smith for originally suggesting this research and for many subsequent conversations. They also benefitted from many conversations on the subject of CCD's with R. H. Krambeck, R. J. Strain, and R. H. Walden.

APPENDIX A

This appendix lists the coefficients of the Fourier series of the various splines used in approximating the charge density on the electrodes. We begin by defining the function

$$u(x, h) = \begin{cases} \left(1 - \frac{x}{h}\right), & 0 \leqq x \leqq h, \\ 0, & h \leqq x \leqq L. \end{cases} \tag{69}$$

Outside the interval $0 \leqq x \leqq L$, this function is defined by periodicity, $u(x, h) = u(x + L, h)$. The Fourier coefficients of $u(x, h)$ are

$$\begin{aligned} a_n(h) &= c_n[u] = \frac{2}{Lh\lambda_n^2}(1 - \cos \lambda_n h), \\ b_n(h) &= s_n[u] = \frac{2}{Lh\lambda_n^2}(\lambda_n h - \sin \lambda_n h), \end{aligned} \tag{70}$$

where the notation $c_n[u]$ and $s_n[u]$ is defined in (58), $\lambda_n = 2n\pi/L$, and $n = 0, 1, 2, \cdots$. Note that

$$a_0(h) = \frac{h}{L}, \qquad b_0(h) = 0. \tag{71}$$

The triangular splines can all be expressed in terms of $u(x, h)$, and their Fourier coefficients are simple linear functions of the coefficients $a_n(h)$ and $b_n(h)$ defined in eq. (70). Thus (see Fig. 4a):

$$p_0(x, h) = u(x, h) + u(L - x, h), \tag{72}$$

and for $n = 0, 1, 2, \cdots$,

$$c_n[p_0] = 2a_n(h), \qquad s_n[p_0] = 0. \tag{73}$$

Similarly (see Figs. 4b and 4c), we have the end splines

$$p_l(x; x_0, h) = u(x - x_0, h), \qquad p_r(x; x_0, h) = u(x_0 - x, h), \tag{74}$$

and for $n = 0, 1, 2, \cdots$

$$\begin{aligned} c_n[p_l] &= (\cos \lambda_n x_0)a_n(h) - (\sin \lambda_n x_0)b_n(h), \\ s_n[p_l] &= (\sin \lambda_n x_0)a_n(h) + (\cos \lambda_n x_0)b_n(h), \end{aligned} \tag{75}$$

and

$$\begin{aligned} c_n[p_r] &= (\cos \lambda_n x_0)a_n(h) + (\sin \lambda_n x_0)b_n(h), \\ s_n[p_r] &= (\sin \lambda_n x_0)a_n(h) - (\cos \lambda_n x_0)b_n(h). \end{aligned} \tag{76}$$

Finally (see Fig. 4f),

$$\begin{aligned} t(x; x_0, h_1, h_2) &= u(x_0 - x, h_1) + u(x - x_0, h_2), \quad x \neq x_0(\mathrm{mod}\ L) \\ &= 1, \quad x = x_0(\mathrm{mod}\ L) \end{aligned} \tag{77}$$

and

$$\begin{aligned} c_n[t] &= \cos \lambda_n x_0 \{a_n(h_1) + a_n(h_2)\} + \sin \lambda_n x_0 \{b_n(h_1) - b_n(h_2)\}, \\ s_n[t] &= \sin \lambda_n x_0 \{a_n(h_1) + a_n(h_2)\} - \cos \lambda_n x_0 \{b_n(h_1) - b_n(h_2)\}. \end{aligned} \tag{78}$$

The coefficients of the Fourier series of the singular edge splines were calculated as follows. We define (see Figs. 4d and 4e)

$$s_\ell(x; x_0, h) = \begin{cases} = (x - x_0)^{-\frac{1}{2}} - h^{-\frac{1}{2}}, & x_0 < x \leqq x_0 + h, \\ = 0, \quad 0 \leqq x \leqq x_0, & x_0 + h < x \leqq L, \end{cases} \tag{79}$$

and

$$s_r(x; x_0, h) = s_\ell(2x_0 - x; x_0, h). \tag{80}$$

Then after an integration by parts

$$\begin{aligned} c_n[s_\ell] &= \\ &= \frac{4}{L}\left[h^{\frac{1}{2}} \cos \lambda_n(x_0 + h) - \frac{\sin \lambda_n(x_0 + h) - \sin \lambda_n x_0}{2\lambda_n h^{\frac{1}{2}}} \right. \\ &\qquad \left. + \lambda_n \int_{x_0}^{x_0+h} (x - x_0)^{\frac{1}{2}} \sin \lambda_n x dx \right], \\ s_n[s_\ell] &= \\ &= \frac{4}{L}\left[h^{\frac{1}{2}} \sin \lambda_n(x_0 + h) + \frac{\cos \lambda_n(x_0 + h) - \cos \lambda_n x_0}{2\lambda_n h^{\frac{1}{2}}} \right. \\ &\qquad \left. - \lambda_n \int_{x_0}^{x_0+h} (x - x_0)^{\frac{1}{2}} \cos \lambda_n x dx \right]. \end{aligned} \tag{81}$$

The integrals on the right of (81) were evaluated by quadratures using Filon's method.[14] Similarly,

$$\begin{aligned} c_n[s_r] &= \\ &= \frac{4}{L}\left[h^{\frac{1}{2}} \cos \lambda_n(x_0 - h) - \frac{\sin \lambda_n x_0 - \sin \lambda_n(x_0 - h)}{2\lambda_n h^{\frac{1}{2}}} \right. \\ &\qquad \left. - \lambda_n \int_{x_0-h}^{x_0} (x_0 - x)^{\frac{1}{2}} \sin \lambda_n x dx \right], \\ s_n[s_r] &= \\ &= \frac{4}{L}\left[h^{\frac{1}{2}} \sin \lambda_n(x_0 - h) + \frac{\cos \lambda_n x_0 - \cos \lambda_n(x_0 - h)}{2\lambda_n h^{\frac{1}{2}}} \right. \\ &\qquad \left. + \lambda_n \int_{x_0-h}^{x_0} (x_0 - x)^{\frac{1}{2}} \cos \lambda_n x dx \right]. \end{aligned} \tag{82}$$

APPENDIX B

We assume that a solution of equations (21) to (25) satisfying boundary and interface conditions (11) or (12) and (13) to (16), (18), (26), and (27) exists and is bounded. As before, we denote this solution by $\psi(x, y)$, we denote our approximate solution by $\psi^a(x, y)$, and we define

$$\xi(x, y) = \psi(x, y) - \psi^a(x, y). \tag{83}$$

By ψ_α, ψ_α^a, and ξ_α, $(0 \leqq \alpha \leqq 4)$, we mean ψ, ψ^a, and ξ restricted to the various subdomains.

From their construction, the approximate solutions satisfy the same equations and boundary and interface conditions as the exact solution, except they do not assume the correct values on the electrodes. Consequently

$$\nabla^2\xi_\alpha(x, y) = 0, \quad (0 \leqq \alpha \leqq 3), \quad \nabla^2\xi_4(x, y) - \xi_4(x, y) = 0. \tag{84}$$

Also $\xi_\alpha(x, y)$ satisfies either boundary condition (11) in the case of single-level metalization or

$$\xi_0(x, -h_{-1}) = 0 \tag{85}$$

in the case of double-level metalization. In addition, $\xi_0(x, 0) = \xi_1(x, 0)$, $0 \leqq x \leqq L$, and $\xi_\alpha(x, y)$ satisfies (14) with $\rho_g \equiv 0$, (15) with $Q(x) \equiv 0$, (16), (18), (26), and (27).

We now outline a proof that if $M = \sup\limits_{x \in E} |\xi_0(x, 0)|$, where E denotes the electrodes, then $|\xi(x, y)| \leqq M$ for all (x, y). The plan of the proof is to show that $\xi_0(x, y)$ is bounded both above and below by its maximum and minimum on $(0 \leqq x \leqq L,\ y = 0)$; that $\xi_\alpha(x, y)$ is bounded above and below by its maximum and minimum on either $(0 \leqq x \leqq L,$ $y = h_{\alpha-1})$ or $(0 \leqq x \leqq L,\ y = h_\alpha)$, $(\alpha = 1, 2, 3)$, where we define $h_0 = 0$; and that $\xi_4(x, y)$ is bounded above and below by its maximum and minimum on $(0 \leqq x \leqq L,\ y = h_3)$. Then we show that the global maximum and minimum must occur on the electrodes.

First consider $\xi_0(x, y)$ in the case of single-level metalization. Then $\xi_0(x, y)$ is harmonic in the strip $S_0 = (0 \leqq x \leqq L, -\infty < y \leqq 0)$, and, by the Phragmen-Lindelöf theorem [Ref. 15, corollary to theorem 19, Chapter 2, with $w(x, y, = 1 - y]$, $\xi_0(x, y)$ is bounded in S_0, both above and below, by its values on the lines $(x = 0, -\infty < y \leqq 0)$, $(y \leqq x \leqq L,$ $y = 0)$, and $(x = L, -\infty < y \leqq 0)$. Let $m_0 = \inf\limits_{0 \leqq x \leqq L} \xi_0(x, 0)$ and $M_0 = \sup\limits_{0 \leqq x \leqq L} \xi_0(x, 0)$. [Note that since $\xi(x, 0)$ is continuous, there exist points $0 \leqq x_m$, $x_M \leqq L$ such that $m_0 = \xi_0(x_m, 0)$, $M_0 = \xi_0(x_M, 0)$.]

Further, we must have

$$\xi_0(x, y) = \tfrac{1}{2}\beta_0 + \sum_{n=1}^{\infty} (\beta_n \cos \lambda_n x + \gamma_n \sin \lambda_n x)e^{\lambda_n y}, \tag{86}$$

and hence

$$\lim_{y \to -\infty} \xi_0(0, y) = \lim_{y \to -\infty} \xi_0(L, y) = \frac{1}{2}\beta_0 = \frac{1}{L}\int_0^L \xi_0(z, 0)dz. \tag{87}$$

From (87) we can conclude that

$$m_0 \leqq \lim_{y \to -\infty} \xi_0(0, y) = \lim_{y \to -\infty} \xi_0(L, y) \leqq M_0. \tag{88}$$

Now if M_0 is not the maximum value of $\xi_0(x, y)$, from what we have just shown, and from the periodicity of $\xi(x, y)$ in x, this maximum value must be assumed at two points $(0, y_0)$ and (L, y_0), with $-\infty < y_0 < 0$. Further, the outward directed normal derivative at these points must be positive (Ref. 15, theorem 8, Chapter 2); that is, $-(\partial\xi_0/\partial x)(0, y_0) > 0$, $(\partial\xi_0/\partial x)(L, y_0) > 0$. However, from periodicity, $(\partial\xi_0/\partial x)(0, y_0) = (\partial\xi_0/\partial x)(L, y_0)$, which is a contradiction. Hence M_0 is the maximum value of $\xi_0(x, y)$ in S_0. The same reasoning applied to $-\xi_0(x, y)$ shows that m_0 is the minimum value in S_0.

In the case of double-level metalization, the maximum principle for harmonic functions (Ref. 15, theorem 2, Chapter 2), plus the boundary condition $\xi_0(x, -h_{-1}) = 0$, implies that $\xi_0(x, y)$ is bounded everywhere in $(0 \leqq x \leqq L) \times (-h_{-1} \leqq y \leqq 0)$, both above and below, by its values on the sides $(x = 0,\ -h_{-1} \leqq y \leqq 0)$, $(0 \leqq x \leqq L,\ 0)$, and $(x = L,\ -h_{-1} \leqq y \leqq 0)$. Then the same reasoning as in single-level metalization shows that $\xi_0(x, y)$ achieves its maximum and minimum on $(0 \leqq x \leqq L, y = 0)$.

Essentially the same arguments used in the double-level metalization case can be used to show that $\xi_\alpha(x, y)$ $(\alpha = 1, 2, 3)$ must achieve both its maximum and minimum either on the line $(0 \leqq x \leqq L, y = h_{\alpha-1})$ or $(0 \leqq x \leqq L, y = h_\alpha)$, where we define $h_0 = 0$.

The Phragmen-Lindelöf theorem can be applied to $\xi_4(x, y)$ in $S_4 = (0 \leqq x \leqq L) \times (h_3 \leqq y < \infty)$ to show that $\xi_4(x, y)$ is bounded everywhere in S_4 both above and below by its values on $(x = 0, h_3 \leqq y < \infty)$, $(0 \leqq x \leqq L, y = h_3)$, and $(x = L, h_3 \leqq y < \infty)$. Making use of the boundary condition $\xi_4(x, \infty) = 0$, the same arguments used in the case of single-level metalization for $\xi_0(x, y)$ show that $\xi_4(x, y)$ achieves its maximum and minimum values on $(0 \leqq x \leqq L, y = h_3)$.

We have shown that $\xi(x, y)$ must assume its maximum and minimum values at points on the lines $(0 \leqq x \leqq L, y = h_\alpha)$, $0 \leqq \alpha \leqq 3$. Suppose that $\xi(x, y)$ is a global maximum at the point $P = (z, h_\alpha)$. Then clearly $\psi_\alpha(x, y)$ and $\psi_{\alpha+1}(x, y)$ take on their maximum values at P. Consequently, their exterior normal derivatives at P must be positive (Ref. 15, theorem 8, Chapter 2), i.e., $(\partial\psi_\alpha/\partial y)(P) > 0$, $-(\partial\psi_{\alpha+1}/\partial y)(P) > 0$. However, if P is not a point on an electrode, it follows from interface conditions (14), (15) with $Q(x) = 0$, (16), or (26) that either $\eta(\partial\psi_\alpha/\partial y)(P) = (\partial\psi_{\alpha+1}/\partial y(P)$ or $(\partial\psi_\alpha/\partial y)(P) = (\partial\psi_{\alpha+1}/\partial y)(P)$, which is a contradiction. Consequently $\xi(x, y)$ achieves its maximum M_0 on an electrode. The same argument applied to $-\xi(x, y)$ shows that $\xi(x, y)$ also achieves its minimum m_0 on an electrode. If we set $M = \max(|m_0|, |M_0|)$, then we have shown that $|\xi(x, y)| \leqq M$.

REFERENCES

1. McKenna, J., and Schryer, N. L., "The Potential in a Charge-Coupled Device with No Mobile Minority Carriers and Zero Plate Separation," B.S.T.J., *52*, No. 5 (May-June 1973), pp. 669–696.
2. Boyle, W. S., and Smith, G. E., "Charge-Coupled Semi-Conductor Devices," B.S.T.J., *49*, No. 4 (April 1970) pp. 587–593.
3. Amelio, G. F., Tompsett, M. F., and Smith, G. E., "Experimental Verification of the Charge-Coupled Device Concept," B.S.T.J., *49*, No. 4 (April 1970), pp. 593–600.
4. McKenna, J., and Schryer, N. L., "On the Accuracy of the Depletion Layer Approximation for Charge-Coupled Devices," B.S.T.J., *51*, No. 7 (September 1972), pp. 1471–1485.
5. Walden, R. H., Krambeck, R. H., Strain, R. J., McKenna, J., Schryer, N. L., and Smith, G. E., "The Buried Channel Charge-Coupled Device," B.S.T.J., *51*, No. 7 (September 1972), pp. 1635–1640.
6. Boyle, W. S., and Smith, G. E., "Charge-Coupled Devices—A New Approach to MIS Device Structures," IEEE Spectrum, *8*, No. 7 (July 1971), pp. 18–27.
7. Grove, A. S., *Physics and Technology of Semiconductor Devices*, New York: John Wiley & Sons, 1967, pp. 49–50.
8. Krambeck, R. H., Walden, R. H., and Pickar, K. A., "A Doped Surface Two-Phase CCD," B.S.T.J., *51*, No. 8 (October 1972), pp. 1849–1866.
9. Schryer, N. L., "Constructive Approximation of Solutions To Linear Elliptic Boundary Value Problems," SIAM J. Num. Anal., *9*, No. 4 (December 1972), pp. 546–572. See references in this paper to further work along these lines.
10. Lehman, R. S., "Developments at an Analytic Corner of Solutions of Elliptic Partial Differential Equations," J. Math. Mech., *8*, No. 5 (September 1959), pp. 727–760.
11. Morrison, J. A., Cross, M. J., and Chu, T. S., "Rain-Induced Differential Attenuation and Differential Phase Shift at Microwave Frequencies," B.S.T.J., *52*, No. 4 (April 1973), pp. 599–604.
12. Elson, B. M., "Charge-Coupled Concept Studied for Photo-Sensors," Aviation Week and Space Technology, *96*, No. 21 (May 22, 1972), pp. 73–75.
13. Krambeck, R. H., "Zero Loss Transfer Across Gaps in a CCD," B.S.T.J., *50*, No. 10 (December 1971), pp. 3169–3175.
14. Filon, L. N. G., "On a Quadrature Formula For Trigonometric Integrals," Proc. Roy. Soc. Edin., *49* (1928–29), pp. 38–47.
15. Protter, M. H., and Weinberger, H. F., *Maximum Principles in Differential Equations*, Englewood Cliffs, N. J.: Prentice-Hall, 1967.

Copyright © 1973 American Telephone and Telegraph Company
THE BELL SYSTEM TECHNICAL JOURNAL
Vol. 52, No. 10, December, 1973
Printed in U.S.A.

Error Rates of Digital Signals in Charge Transfer Devices

By K. K. THORNBER

(Manuscript received June 21, 1973)

We calculate the probability of error in detecting digital signals transferred through a charge transfer device in the presence of incomplete charge transfer, random noise in the device, and detection uncertainty in the detector. The coefficient of incomplete charge transfer is assumed to be independent of charge-packet size, and both the device noise and detector noise are assumed to be Gaussian. Error probabilities for two-level and four-level codes are computed for the cases of both simple static and optimum dynamic detection. For rms detection voltage level fluctuations V_d *of the order of tenths of volts (much larger than the random noise fluctuations in the device), a very rapid increase in error probability (from* $\approx 10^{-20}$ *to* $\approx 10^{-5}$*) is found to occur for a very small (20 percent) change in* V_d*. This indicates that detection level fluctuations will have to be held down to a few hundred millivolts at most. To achieve equal error rates with an error probability of about* 10^{-14}*,* V_d *for the detection of four-level codes will have to be about 3.5 times* smaller *than for two-level codes. Comparison of error probabilities under static and dynamic detection shows that in CTD's improved detection has a greater potential for reducing error rates than improved coding.*

I. INTRODUCTION

As a packet of charge is transferred through a charge transfer device (CTD), the size of the packet is altered owing to effects of incomplete transfer[1,2] and noise.[3–9] At the output the size of each packet is measured and, depending on its size, a decision is made as to the initial size of the packet. Usually the decision will be correct. However, occasionally the cumulative effects of incomplete transfer and noise will result in a sufficiently distorted charge packet that an error will be made. It is the purpose of this paper to calculate the probability of making such a detection error. When this probability is multiplied by the rate of detection (the clock frequency), we obtain the error rate for a single device. Multiplying by the number of devices of interest in the storage unit or in the processing unit, we obtain the total error rate, a very useful quantity for evaluating digital systems. (By "detection," we include regeneration; by a "single device," we mean a single unregenerated line of transfer elements.)

To calculate the probability of detection error we assume that the effect of incomplete transfer on the signal can be treated in terms of the usual small signal analysis.[10,11] (The coefficient of incomplete charge transfer, α, is assumed to be constant, independent of the size of the signal.) Charge gain or loss because of leakage current is assumed to be sufficiently small that it can be ignored. The random noise which introduces fluctuations into the size of the charge packets is assumed to be Gaussian.[8] This is reasonable by the law of large numbers, since the size of a charge packet is typically 10^6 elementary charges. In the numerical calculations, only shot noise at the input and thermal noise induced during charge transfer are considered, as these are the most important sources of noise in good devices.[8] In addition, the detection levels are assumed to fluctuate with Gaussian statistics. This simulates (*i*) the fluctuation in detection levels from device to device, (*ii*) the uncertainty in the location of the boundary between two decision regions, (*iii*) the uncertainty introduced from nonideal regeneration, and (*iv*) the fluctuations induced by the coupling of the clock lines to the output. In a future paper, we plan to treat several of these effects more carefully.

For our numerical work we take the position that probabilities of error of about 10^{-14} are of greatest interest. Values much higher would necessitate more-often-than-daily correction of a multimegabit store. Attention is focused on how large a fluctuation can be tolerated in the detection levels, so that the probability of error is in this region for the cases of two-level and four-level digital codes. In addition, the error probability is also examined as a function of the number of charge transfers. Similar calculations are made for the theoretically minimum possible error rate, which can be obtained using a dynamic detection scheme.[12] Comparison of this absolutely minimum error probability with the error probability obtained using conventional (static) detection suggests that a substantial improvement in error rate is possible using dynamic rather than static detection levels.[12]

II. PROBABILITY OF ERROR

In previous work,[10,12] it has been shown that in the absence of noise, after $(n + 1)$ transfers, each characterized by a coefficient of incomplete transfer α, a charge packet of some initial size Q_i has at the output a size $Q(i)$ given by

$$Q(i) = (1 - \alpha)^{n+1}Q_i + Q_B, \tag{1}$$

where

$$Q_B = (1 - \alpha)^{n+1} \sum_{N=1}^{\infty} \binom{N+n}{N} \alpha^N Q_N, \tag{2}$$

and where Q_N is the initial size of the Nth packet preceding Q_i. In the presence of noise, the probability $P(Q - Q(i))dQ$ that the observed size Q of the packet is $Q(i)$ to within dQ is given by

$$P[Q - Q(i)]dQ = \exp\{ - [Q - Q(i)]^2/(2\Delta Q^2)\}/(2\pi\Delta Q^2)^{\frac{1}{2}}dQ, \tag{3}$$

where ΔQ^2 is the mean-square fluctuation in the size of the charge packet at the output resulting from noise (see Appendix A). If the range of Q over which the packet will be detected as Q_i is given by $Q_i^- < Q < Q_i^+$, then P_i, the probability of error in detecting a specific $Q(i)$ packet, is

$$P_i = \int_{-\infty}^{Q_i^-} P[Q - Q(i)]dQ + \int_{Q_i^+}^{\infty} P[Q - Q(i)]dQ. \tag{4}$$

To determine error probability, P_i must be averaged over all possible $Q(i)$ for each i.

The quantities Q_i^- and Q_i^+ can be readily determined by rewriting (1) and (2) in the form

$$Q(i) = \bar{Q} + (Q_i - \bar{Q})(1 - \alpha)^{n+1} + Q_B', \tag{5}$$

where

$$Q_B' = (1 - \alpha)^{n+1} \sum_{N=1}^{\infty} \binom{N+n}{N} \alpha^N (Q_N - \bar{Q}). \tag{6}$$

In (5) and (6), $\bar{Q}$ is the (time) average size of a charge packet. (For example, if two packet sizes, Q_1 and Q_0, are used equally frequently in a two-level digital code, then $\bar{Q} = (Q_1 + Q_0)/2$.) If now we average eq. (5) over all possible preceding sequences of packets, then we obtain for $\langle Q(i)\rangle$, the average size of $Q(i)$,

$$\langle Q(i)\rangle = \bar{Q} + (Q_i - \bar{Q})(1 - \alpha)^{n+1}, \tag{7}$$

since the average $\bar{Q}_B'$ of Q_B' is zero. (Note that $\bar{Q}_N = \bar{Q}$ by definition.) The deviation of $Q(i)$ from $\langle Q(i)\rangle$ is simply Q_B', independent of i. $[Q(i) - \langle Q(i)\rangle = Q_B'.]$ By extending the results of a previous treatment[12] of two-level coding to the multilevel coding considered here, it follows at once (see Appendix B) that the theoretically minimum possible error rates will be achieved for Q_i^- and Q_i^+ given by

$$Q_i^- = \frac{\langle Q(i)\rangle + \langle Q(i-1)\rangle}{2} + Q_B' \tag{8}$$

and

$$Q_i^+ = \frac{\langle Q(i)\rangle + \langle Q(i+1)\rangle}{2} + Q_B'. \tag{9}$$

(Note: $Q_{i+1}^- = Q_i^+$. For an M level system, $i = 0, 1, \cdots, M-1$. For completeness we define $Q(-1) \equiv -\infty$ and $Q(M) \equiv +\infty$.) These results apply even if the coding levels are not equally spaced. However, it should be obvious that if each size packet is used equally frequently, then equally spaced levels will result in the least probability of error.

In previous work[12] we have referred to the detection scheme which utilizes detection levels determined by Q_B' (that is, by the preceding signal) as a dynamic detection scheme. In other words, by subtracting out the incompletely transferred portion from the preceding signal prior to *each* detection (achievable under noiseless conditions), we can select detection regions which null out the scatter in the signal-charge size induced by incomplete transfer. Since random noise cannot be nulled out, a lower limit is placed on the error probability.

Using (8) and (9), we now compute the minimum error probability $P_{\min}$ of a single detection and average this over all possible preceding signals to obtain the minimum error probability $\langle P_{\min}\rangle$. Let p_i be the relative average frequency with which charge packets of initial size Q_i are used in the code. Then using (4) we may write

$$\begin{aligned} P_{\min} &= \sum_{i=0}^{M-1} p_i P_i \\ &= \sum_{i=0}^{M-1} p_i \left(\int_{-\infty}^{Q_i^- - Q(i)} P(Q)dQ + \int_{Q_i^+ - Q(i)}^{\infty} P(Q)dQ \right). \end{aligned} \tag{10}$$

If we note that $[Q_i^- - Q(i)] = -[\langle Q(i)\rangle - \langle Q(i-1)\rangle]/2$ and that $[Q_i^+ - Q(i)] = +[\langle Q(i+1)\rangle - \langle Q(i)\rangle]/2$, then $P_{\min}$ becomes

$$\begin{aligned} P_{\min} = \sum_{i=0}^{M-1} p_i \Bigg(& \int_{-\infty}^{-[\langle Q(i)\rangle - \langle Q(i-1)\rangle]/2} P(Q)dQ \\ & + \int_{+[\langle Q(i+1)\rangle - \langle Q(i)\rangle]/2}^{+\infty} P(Q)dQ \Bigg). \end{aligned} \tag{11}$$

As mentioned in the preceding paragraph, $P_{\min}$ is independent of the foregoing charge packets. Thus $\langle P_{\min}\rangle = P_{\min}$. As $\langle P_{\min}\rangle$ is the minimal, or optimal, error probability, we will use it as a touchstone to compare other detection schemes.

Complete dynamic detection as discussed above is one extreme in detection. [Boonstra and Sangster[4] have operated a CTD utilizing a partial lowest order (in $n\alpha$) correction.] The other extreme in detection is to ignore completely the sequence of charge packets preceding the packet of interest and to attempt to detect without compensating for the accumulated background charge. Since the average of Q_B' is 0, one would then choose for Q_i^- and Q_i^+ the following

$$Q_i^- = [\langle Q(i)\rangle + \langle Q(i-1)\rangle]/2 \tag{12}$$

and

$$Q_i^+ = [\langle Q(i+1)\rangle + \langle Q(i)\rangle]/2. \tag{13}$$

In this case, the error probability P associated with a specific detection event becomes

$$P = \sum_{i=0}^{M-1} p_i \left(\int_{-\infty}^{-[\langle Q(i)\rangle - \langle Q(i-1)\rangle]/2 - Q_B'} P(Q)dQ + \int_{+[\langle Q(i+1)\rangle - \langle Q(i)\rangle]/2 - Q_B'}^{\infty} P(Q)dQ \right) \cdot \tag{14}$$

To calculate $\langle P\rangle$, the average error probability, we must average (14) over all possible preceding signal sequences. Unlike $P_{\min}$, P is a function of the preceding sequence through Q_B'. In the remainder of this paper we shall focus attention on calculating $\langle P\rangle$.

III. NUMERICAL METHOD

Let us assume (*i*) that we are using a multilevel (M-level) code in which each size of charge packet is used equally frequently (so that $p_i = 1/M$), and (*ii*) that the levels of charge are equally spaced. Let $S^{\frac{1}{2}} \equiv [Q(i+1) - Q(i)]/2$. Then from (14), it follows that

$$\langle P\rangle = \frac{2M-2}{M} \int_{-\infty}^{-[S^{\frac{1}{2}}\cdot(1-\alpha)^{n+1}+Q_B']} P(Q)dQ. \tag{15}$$

[Note: If n and α are such that $|Q_B'| > S^{\frac{1}{2}}$ for some sequences of charge packets, then errors are made with this detection scheme even in the absence of noise. Thus using the detection scheme characterized by the Q_i^- and Q_i^+ given by eqs. (12) and (13), it is essential that n and α be such that $|Q_B'| < S^{\frac{1}{2}}$ for all possible sequences of packets. Thus, $(S^{\frac{1}{2}} + Q_B') > 0$, and $\langle P\rangle < 1$.]

For numerical calculations, it is expedient to use (3) to rewrite (15) as

$$\langle P\rangle = 2\left(1 - \frac{1}{M}\right) \int_{-\infty}^{-(S/N)^{\frac{1}{2}}(1+\alpha)^{n+1}(1+\Sigma)} e^{-x^2/2} \frac{dx}{\sqrt{2\pi}}, \tag{16}$$

where S/N, the signal-power-to-noise-power ratio, is given by

$$\mathrm{S/N} = \frac{\{[Q(i+1) - Q(i)]/2\}^2}{\Delta Q^2}, \tag{17}$$

and where Σ is given by

$$\Sigma = \sum_{N=1}^{\infty} \binom{N+n}{N} \alpha^N J_N. \tag{18}$$

In (18) each $J_N[= (Q_N - \bar{Q})/S^{\frac{1}{2}}]$ is a random variable, which for M even can take on the values $\pm 1, \pm 3, \pm 5, \cdots, \pm(M-1)$ with equal probability, and for M odd is $0, \pm 2, \cdots, +(M-1)$ again with equal probability. To evaluate (16) in this form it is now necessary to average the integral in (16) over all possible sequences $J_1, J_2, J_3, \cdots$.

In this paper, we focus attention on that range of n and α which will probably be of greatest device interest—$n\alpha \ll 1$. In this case, only the first few terms in Σ will contribute significantly to its total value. By "significantly," we mean, of course, that whether $J_N = +(M-1)$ or $J_N = -(M-1)$ for fixed $J_1, \cdots, J_{N-1}$ and $0 = J_{N+1} = J_{N+2} = \cdots$, makes an acceptably small (say, 0.1%) difference in the values of the integral in (16). Thus, we can proceed as follows. Evaluate the integral in (16) for J_1 equal to each of its possible values and $0 = J_2 = J_3 = \cdots$, sum, divide by M, and multiply by $2(1 - 1/M)$. This gives a first estimate of $\langle P \rangle$ which we call $\langle P \rangle_1$. Now again evaluate the integral in (16) for all possible pairs of J_1, J_2 with $0 = J_3 = J_4 = \cdots$, sum, divide by M^2, and multiply by $2(1 - 1/M)$. This gives a second estimate of $\langle P \rangle$ which we call $\langle P \rangle_2$. In general, $\langle P \rangle_2 > \langle P \rangle_1$. If $(\langle P \rangle_2 - \langle P \rangle_1)/\langle P_1 \rangle$ is within the desired accuracy, then we may stop here. If $\langle P \rangle_2$ differs significantly from $\langle P \rangle_1$, we calculate $\langle P \rangle_3$ in the obvious way and compare to $\langle P \rangle_2$, etc. For the numerical results presented in the next section, $\langle P \rangle_3$ is as far as it is necessary to calculate to obtain 0.1 percent accuracy. For $n\alpha \ll 1$, convergence is guaranteed.

Often knowledge of the error probability to within a factor of 2 is adequate for design purposes. Thus, computing can be greatly facilitated if use is made of the following result. If $A > 1$, then

$$D/2 < I(A) < D, \tag{19}$$

where

$$I(A) = \int_{-\infty}^{-A} e^{-x^2/2} dx \tag{20}$$

and

$$D = \exp(-A^2/2)/A. \tag{21}$$

In the following section, our results for error probability are somewhat high, as D has been used in place of $I(A)$ in evaluating (16).

IV. NUMERICAL RESULTS

We have calculated both the minimum error probability $\langle P_{\min}\rangle$ [eq. (11)] using a dynamic detection scheme [eqs. (8) and (9)] and the error probability $\langle P\rangle$ [eq. (15)] using a static detection scheme [eqs. (12) and (13)], both for two-level and four-level coding. In all cases, we have taken $\alpha = 10^{-3}$, storage capacitance $C = 0.1$ pF, detection capacitance $C_{DE} = 0.1$ pF (see Appendix A), $Q_0 = C\cdot(4$

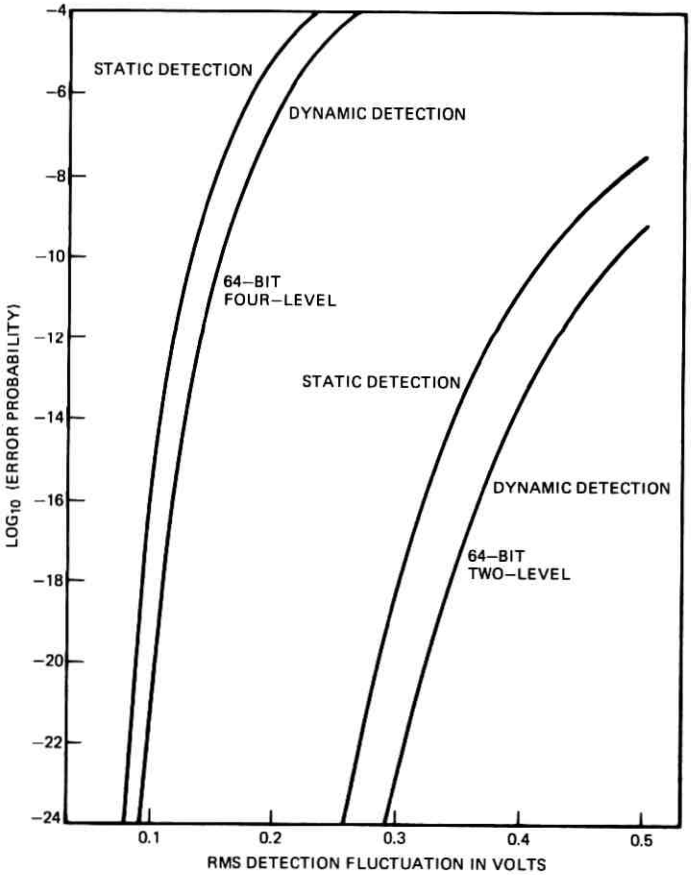


Fig. 1—Error probability as a function of the root-mean-square fluctuation in the detection level voltage for static and dynamic detection schemes of two-level and four-level coding in a 64-bit device.

volts), and $Q_{M-1} = C\cdot(10$ volts). All calculations were carried to 0.1 percent.

In Fig. 1 we have plotted error probability $\langle P\rangle$ (static detection) and minimum error probability $\langle P_{\min}\rangle$ (dynamic detection) for two-level and four-level coding as a function of the root-mean-square detection-voltage fluctuation V_d. For the two-level results $n = 128$, and for the four-level results $n = 64$. Both these cases correspond to a 64-bit device. It is quite clear from Fig. 1 that to achieve an error probability of about 10^{-14}, for two-level coding $V_d < 0.345$ V, whereas for four-level coding $V_d < 0.105$ V. This means that, to be able to use four-level coding, we must have significantly better control of detection voltage fluctuation than is necessary with two-level coding.

We might imagine that a trade-off could exist which would favor four-level coding. For example, only one-half the number of transfer stages are needed with four levels as compared with two levels. Taking α inversely proportional[1,2] to C, for four levels we can double C and thereby cut α in half relative to C and α for two levels. As α is reduced, the role of incomplete transfer is reduced as well. However, for $V_d = 0.35$ V, detection noise dominates the random noise. Thus S/N is practically unchanged as C is varied [see eq. (24) in Appendix A]. In addition, S/N for four levels is so small (≈ 8) that $\langle P\rangle$ goes only from $6.8\cdot 10^{-3}$ for $\alpha = 10^{-3}$ to $4.3\cdot 10^{-3}$ for $\alpha = 0.5\cdot 10^{-3}$. Of course, for smaller V_d the change would be more drastic, as S/N would be larger. However, for smaller V_d, two-level operation is enhanced as well.

In Fig. 2 we have plotted the error probability of a two-level code as a function of the number of transfers for three different detection-level fluctuations for both static and dynamic detection schemes. In Fig. 3 we have plotted the same quantities for four-level coding and lower detection-level fluctuations. The striking superiority of dynamic detection over static detection is evident. (The dynamic curves are not actually flat; they increase somewhat in the region shown and much more rapidly for $n\alpha > 1$.)

V. CONCLUSIONS

In this paper we have derived expressions for the probability of error in detecting the size of charge packets carrying digital information in charge transfer devices. Effects of both random noise in the transfer device and detection noise at the detector were included. Error probabilities were computed and compared for common, static detection and optimum, dynamic detection of two types of coding

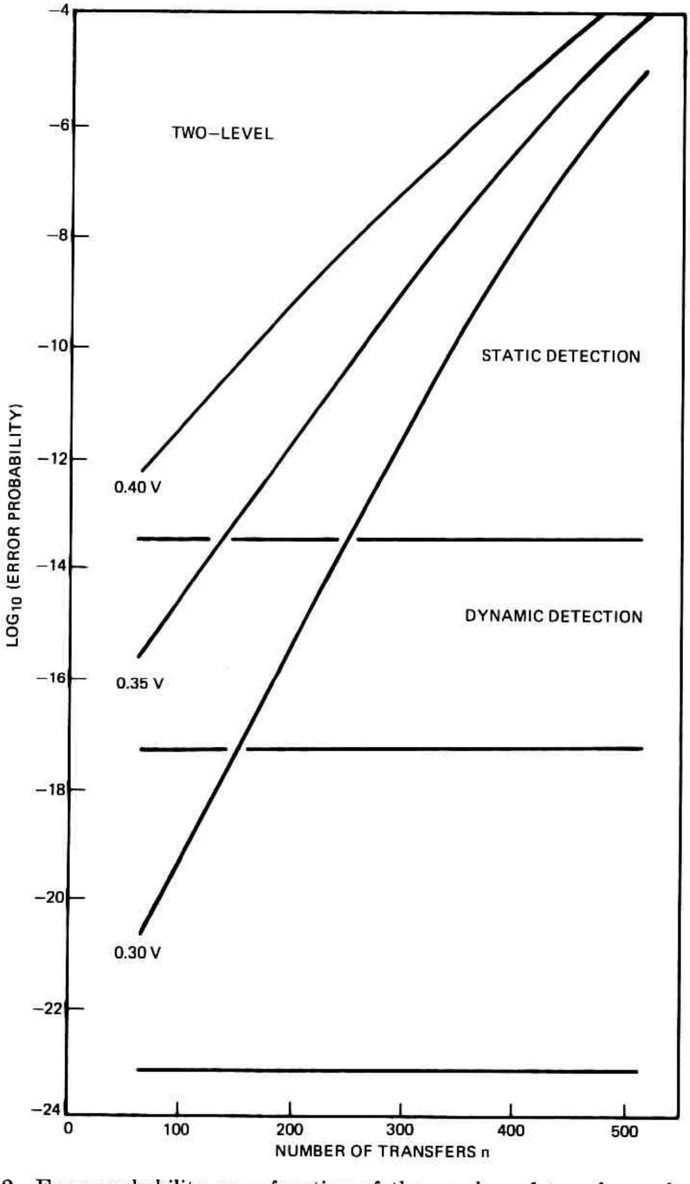


Fig. 2—Error probability as a function of the number of transfers n for static and dynamic detection of two-level coding for three values of root-mean-square detection voltage fluctuation (0.30, 0.35, and 0.40 V). For given n, the corresponding device is an $n/2$-bit device.

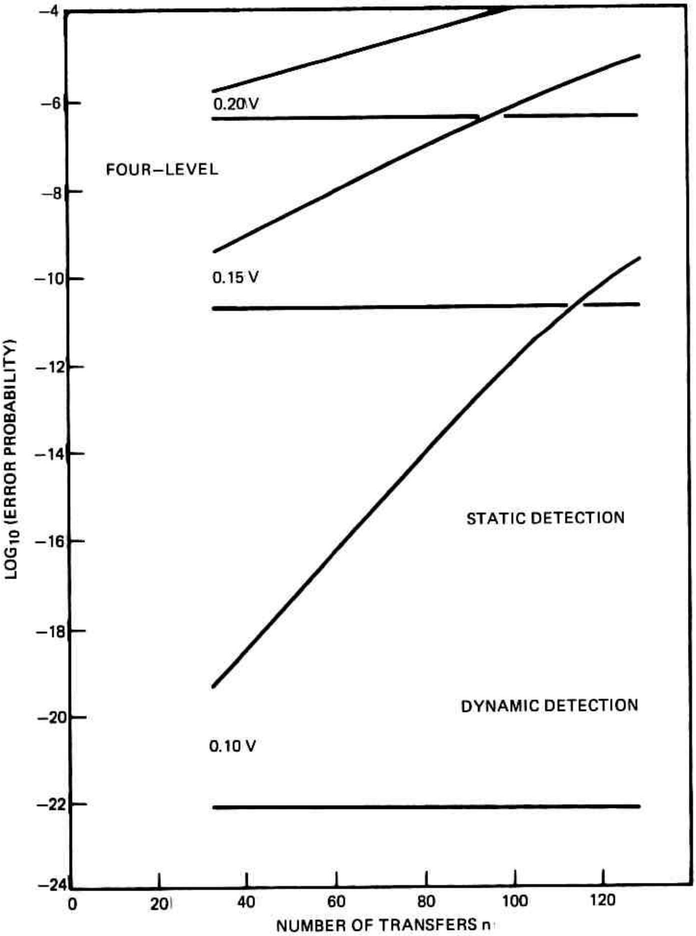


Fig. 3—Error probability as a function of the number of transfers n for static and dynamic detection of four-level coding for three values of root-mean-square detection voltage fluctuation (0.10, 0.15, and 0.20 V). For given n, the corresponding device is an n-bit device.

schemes. In the region of primary interest here (detection noise much larger than device noise), it was found that the error probability is a very sensitive function of detection noise, varying 20 orders of magnitude for a ±20 percent change in the detection noise level. Also significant was the finding that, to achieve equivalent operational performance, the rms detection noise level in a device using a four-level code must be 3.5 times smaller than that in a device using a two-level code. Thus, in designing circuits for digital signal detection, it will be necessary to focus primary attention on the detection level noise. This must be held to a few hundred millivolts at most. It was also shown how our dynamic detection scheme could maintain a very low error probability as the number of transfers, n, was greatly increased.

VI. ACKNOWLEDGMENTS

I wish to thank R. J. Strain and G. E. Smith for many helpful suggestions.

APPENDIX A

Noise

In general, we can write the mean-square noise charge, ΔQ_{rf}^2, resulting from random fluctuations in the form[8]

$$\Delta Q_{rf}^2 = \Delta Q_{\text{input}}^2(1-\alpha)^{2(n+1)} + \Delta Q_{SP}^2 H_{SP}(n+1) \\ + 2\Delta Q_{TP}^2 H_{TP}(n+1), \quad (22)$$

where $\Delta Q_{\text{input}}^2$ is the input noise contribution, ΔQ_{SP}^2 is the storage process noise acquired by a single packet during a single clock period, ΔQ_{TP}^2 is the transfer process noise acquired by a single packet during a single charge transfer, $(1-\alpha)^{2(n+1)}$ is the (square of the) attenuation from input to output, $H_{SP}(n)$ is the compounding factor for storage process noise, and $H_{TP}(n)$ is the compounding factor for transfer process noise. A derivation of eq. (22) and a discussion of the various terms therein are treated elsewhere.[6–8] For our purposes ($n\alpha \ll 1$), it suffices to let $H_{SP}(n+1) = H_{TP}(n+1) = n+1$. (For $n\alpha \ll 1$, incomplete transfer of the noise can be ignored relative to the noise itself. Thus after $(n+1)$ transfers, the accumulated noise is just $(n+1)$ times the noise resulting from a single transfer.) We shall assume that $\Delta Q_{SP}^2 \ll \Delta Q_{TP}^2$ and set $\Delta Q_{SP}^2 = 0$. For shot noise at the input, $\Delta Q_{\text{input}}^2 = qQ$, where Q is the mean total signal charge $(Q_{M-1} - Q_0)$. For thermal noise, $\Delta Q_{TP}^2 = \frac{2}{3}kTC$. As it turns out, the exact details of ΔQ_{rf}^2 are not essential because these random effects turn out to be much smaller than the detection level fluctuations discussed below. However, if these detection level fluctuations can be reduced, then eq. (22) is quite important, especially in the region of $n\alpha \geqq 1$ where devices can operate using dynamic detection.

There are two equivalent ways in which detection level fluctuations can be included. The more systematic way is to use ΔQ_{rf}^2 in place of

ΔQ^2 in eqs. (3) and (4) and then average over Q_i^- and Q_i^+ in (4) with the appropriate distribution for the detection level fluctuations. In this paper, we have restricted attention to a Gaussian distribution for these fluctuations. As the noise is also Gaussian, it follows from a straightforward integration that we can write eq. (4) in the form given in the text with

$$\Delta Q^2 = \Delta Q_{rf}^2 + \Delta Q_d^2, \tag{23}$$

where ΔQ_d^2 is the mean-square uncertainty of the detection level. [The second way is just to write (23) *a priori.*] Since some detection error will result from nonideal regeneration, once this can be more accurately simulated, a more careful analysis of detection uncertainty will be necessary.

The uncertainty ΔQ_d^2 is generated by an uncertainty V_d^2 in the detection voltage. Thus,

$$\Delta Q_d^2 = C_{DE}^2 V_d^2, \tag{24}$$

where C_{DE} is the capacitance of the detector. In our calculations, we have assumed that $C_{DE} = C$, where C is the elemental storage capacitance. If now $\Delta Q_d^2 \gg \Delta Q_{rf}^2$, then $\mathrm{S/N} \approx V^2C^2/V_d^2C^2 = V^2/V_d^2$ independent of the capacitance. (Here V represents the signal voltage.) Thus, increasing C does *not* improve the signal-to-noise ratio (S/N) when detection noise exceeds random noise.

APPENDIX B

Minimum Error Probability

It is a very general result, rederived in a previous work,[12] that if $I(A)$ is defined by

$$I(A) = \int_{-\infty}^{-A} e^{-x^2/2} dx \tag{25}$$

and if the probability that $A < 0$ is 0, then

$$\langle I(A) \rangle \geqq I(\langle A \rangle). \tag{26}$$

Thus any detection scheme for which

$$\int_{-\infty}^{Q_i^-} P[Q - Q(i)]dQ = \int_{-\infty}^{Q_i^- - Q(i)} P(Q)dQ = \int_{-\infty}^{\langle Q_i^- - Q(i) \rangle} P(Q)dQ \tag{27}$$

for each i (and the corresponding equalities for $\int_{Q_i^+}^{\infty} \cdots dQ$) will result in the minima overall error probability. The choice given in eqs. (8) and (9) does this, as it makes $Q_i^- - Q(i)$ independent of the preceding sequence of charge packets over which the average in (26) and (27) is taken.

APPENDIX C

Realizability of Dynamic Detection Scheme

Before considering the realizability of the scheme of dynamic detection discussed in the text, a point of clarification is necessary. One reason for developing dynamic detection here is to see just how low we can, in principle, make the error rate. This we have done assuming Gaussian noise, linear incomplete transfer, *and* complete knowledge of the preceding signal. If we relax the last assumption, we must take into account the fact that our detection of the preceding signal may not be perfect and, therefore, a higher error rate may in fact be the minimal rate possible physically. This problem is more difficult and will not be attempted here. What is important to distinguish, however, is the difference between "perfect" dynamic detection, which provides a minimum error rate below which one cannot hope to achieve, and the actual error rate when employing dynamic detection, which as we shall indicate below is not appreciably larger than the minimum rate under operating conditions of interest. With this in mind, let us proceed to a consideration of realizability.

In the absence of noise, the dynamic detection scheme is clearly realizable in principle. Knowledge of the preceding signal permits determining the background charge level Q_B (resulting from incomplete transfer) operationally using eq. (2). This permits placing the detection levels so that the size of the charge packet to be detected will lie midway between these detection levels. Under noiseless conditions, this permits error-free detection which, in turn, provides the signal history needed to determine Q_B for the next packet detection.

In the presence of noise, one may ask whether the dynamic detection scheme envisioned in Section II is truly realizable. If, for example, an error is made in detection, then the detection levels may be shifted far enough away from optimum so that for the next packet the error probability will be greatly increased. Fortunately, as the argument below suggests, if the probability of making a second error immediately following the first is small compared to unity, then the optimum (minimum) error probabilities presented in the text are only slightly increased (on the order of percents rather than order of magnitude).

Consider a two-level code and the dynamic detection scheme in which it is only necessary to adjust the detection levels for the first preceding signal $[Q_B = \alpha(n+1)Q_1,\ Q_1 =$ size of first preceding packet$]$. We desire the probability $P_c(i)$ that the ith packet is detected correctly. Clearly,

$$P_c(i+1) = P_{cc}(i+1\,|\,i)P_c(i) + P_{ce}(i+1\,|\,i)P_e(i), \tag{28}$$

where $P_e(i)$ $[= 1 - P_c(i)]$ is the probability that the ith packet is detected incorrectly, $P_{cc}(i+1\,|\,i)$ is the probability that the $(i+1)$th packet is correctly detected given that the ith was also, and $P_{ce}(i+1\,|\,i)$ is the probability that the $(i+1)$th packet is correctly detected given that the ith was detected incorrectly. Noting that $P_c(i+1) = P_c(i) = P_c$, we can solve (28) for P_c, obtaining

$$P_c = [1 + P_{ec}(i+1\,|\,i)/P_{ce}(i+1\,|\,i)]^{-1}, \tag{29}$$

where $P_{ec} = 1 - P_{cc}$. The error probability $P_e(= 1 - P_c)$ which we seek is, therefore, given by

$$P_e = [1 + P_{ce}(i+1\,|\,i)/P_{ec}(i+1\,|\,i)]^{-1} \tag{30}$$

$$\approx \frac{P_{ec}}{P_{ce}} = \frac{P_{ec}}{1 - P_{ee}}. \tag{31}$$

Equation (31) follows if $P_{ec} \ll P_{ce}$, as will be the case for $P_e \ll 1$, which is the region of greatest interest. If now $P_{ee} < 0.1$, then P_e will differ from P_{ec} [calculated in the text, eq. (11), as $\langle P_{\min}\rangle$] by less than 10 percent, an insignificant change. Although we have not investigated P_{ee} in detail, it is clear that P_{ee} will be closer in size to $\langle P\rangle$ (eq. 15) corresponding to static detection rather than to $\langle P_{\min}\rangle$. However, what is important is that P_{ee} can be as large as one-half without increasing $\langle P_{\min}\rangle$ by more than a factor of 2. Thus, the $\langle P_{\min}\rangle$ calculated here are not expected to be overly optimistic so long as the detection level need only be corrected on the basis of just the first preceding signal. For the present, this is the situation of primary interest. It should be kept in mind, however, that it is the random noise and not the incomplete transfer which complicates dynamic detection. With sufficiently low noise, we can in principle greatly reduce incomplete-transfer distortion without appreciable propagation of detection errors even when the detection level must be corrected on the basis of many preceding signals.

REFERENCES

1. Berglund, C. N., Thornber, K. K., "Incomplete Transfer in Charge Transfer Devices," IEEE J. Solid-State Circuits, *SC-18* (1973), pp. 108–116. See Refs. 5 to 21 of this reference for a list of prior treatments of incomplete transfer.

2. Berglund, C. N., and Thornber, K. K. "Fundamental Comparison of Incomplete Charge Transfer in Charge Transfer Devices," B.S.T.J., *52*, No. 2 (February 1973), pp. 147–182.
3. van der Ziel, A., "Noise in Solid State Devices and Lasers," Proc. of the IEEE, *58* (1970), pp. 1178–1206.
4. Boonstra, L., and Sangster, F. L. J., "Progress on Bucket-Brigade Charge-Transfer Devices," 1972 IEEE Solid-State Conference, Digest of Technical Papers *15* (1972), pp. 140–1.
5. Tompsett, M. F., "Quantitative Effects of Interface States on the Performance of Charge-Coupled Devices," IEEE Transactions on Electron Devices, *ED-20* (1973), pp. 46–55.
6. Thornber, K. K., "Noise Suppression in Charge Transfer Devices," Proc. of the IEEE, *60*, pp. 1113–4.
7. Thornber, K. K., and Tompsett, M. F., "Spectral Density of Noise Generated in Charge Transfer Devices," IEEE Transactions on Electron Devices, *ED-20* (1973). p. 456.
8. Thornber, K. K. "Theory of Noise in Charge Transfer Devices," in preparation.
9. Carnes, J. E., and Kosonocky, W. F. "Noise Sources in Charge Coupled Devices," RCA Review, *33* (1972), pp. 327–343.
10. Joyce, W. B., and Bertram, W J., "Linearized Dispersion Relation and Green's Function for Discrete-Charge-Transfer Devices With Incomplete Transfer," B.S.T.J., *50*, No. 6 (July-August 1971), pp. 1741–1759.
11. Berglund, C. N., "Analog Performance Limitations of Charge Transfer Dynamic Shift Resisters," IEEE J. Solid-State Circuits, *SC-6* (December 1971), pp. 391–394.
12. Thornber, K. K., "Operational Limitations of Charge Transfer Devices," B.S.T.J., *52*, No. 9 (November 1973), pp. 1453–1482. The residual charge, Q_R, used in this reference is the same entity as the background charge, Q_B, used in the present paper.

Copyright © 1973 American Telephone and Telegraph Company
THE BELL SYSTEM TECHNICAL JOURNAL
Vol. 52, No. 10, December, 1973
Printed in U.S.A.

Stability of a General Type of Pulse-Width-Modulated Feedback System

By R. WALK and J. ROOTENBERG

(Manuscript received May 31, 1973)

Because of its theoretical and practical interest, the stability problem in pulse-width-modulated feedback systems has received an enormous amount of attention. Much of the reported literature deals with highly approximate methods, and the exact approaches, based on Lyapunov's direct method or functional analysis, are quite restrictive and do not easily lend themselves to systematic compensation or design.

In this paper, a quite general PWM is considered, and a frequency domain stability criterion is presented, yielding a geometric interpretation in the Popov plane.

I. INTRODUCTION

The stability of pulse-width-modulated control systems has been an active area of research since the early 1960's. A variety of graphical and analytical approaches to the problem have appeared in the literature.[1–4] Aside from the approximate methods, the main contribution of the early 1960's to exact stability criteria was in the application of Lyapunov's direct method.[5,6] As is often the case, this approach yields conservative results and does not easily lend itself to system compensation. Input-output stability via functional analytic techniques was reported in Skoog[7] and Skoog and Blankenship,[8] where conditions for the L_1 boundedness and continuity of the system operator are derived for PWM systems (considered there to belong to a larger class of pulse-modulated systems, i.e., that class of modulators for which the input is sampled). One drawback to the above type of criteria is the lack of a simple geometric interpretation; e.g., a Popov-type condition. In Skoog[7] a circle criterion is derived for PWM systems, operating in the "quasi-linear" mode; that is, where the modulator does not saturate. In its exact form, however, the above condition is rather difficult to apply (the radius of the circle is in the form of an infinite sum, involving an arbitrary parameter).

In all the previous cases, the pulse-width modulator considered is the periodic sampling type, where the input to the modulator is sampled and the polarity and width of the output pulse is determined from that sample. This paper will consider a similar PWM which is a *generalization* (GPWM) of the so-called natural sampling type.[9–11,*] In this scheme, the input is compared to a repetitive reference waveform, and pulses are emitted in accordance with some specified relation between the two signals.

It is the purpose of this paper to develop a geometric stability criterion for the GPWM system. The main result of the paper is a frequency domain condition for the stability of a feedback system containing a GPWM (described below) and a linear plant that may be either lumped or distributed. The condition, similar to a Popov type, is interesting in that it allows a tradeoff between the slope of the stability line and its intersection with the real axis of the Popov plane.

II. NOTATION

In this paper we are concerned with measurable functions of a real variable defined on $[0, \infty)$. We consider the function spaces $L_p (p \geqq 1)$, where

$$L_p(p \in [1, \infty)) = \left\{ x(t) : \int_0^\infty |x(t)|^p dt < \infty \right\}$$

and

$$L_\infty = \left\{ x(t) : \operatorname*{ess\,sup}_{t \geqq 0} |x(t)| < \infty \right\}.$$

The corresponding norms are defined by

$$\|x(t)\| L_p(p \in [1, \infty)) = \left(\int_0^\infty |x(t)|^p dt \right)^{1/p}$$

and

$$\|x(t)\| L_\infty = \operatorname*{ess\,sup}_{t \geqq 0} |x(t)|.$$

Also, we shall use the extensions[13] of these spaces, defined as:

$$L_{pe}(p \in [1, \infty)) = \left\{ x(t) : \int_0^\infty |x_T(t)| dt < \infty, \quad \forall\, T \in [0, \infty) \right\}$$

and

$$L_{\infty e} = \{ x(t) ; \operatorname*{ess\,sup}_{t \geqq 0} |x_T(t)|, \quad \forall\, T \in [0, \infty) \},$$

[*] In a very recent paper, V. M. Kuntsevich[12] has treated this type of modulator by the discrete version of Lyapunov's direct method.

where

$$X_T(t) = \begin{cases} X(t), & t \leqq T \\ 0, & t > T \end{cases}.$$

And finally stability will be interpreted to mean that, for all inputs belonging to the spaces of interest, the composite system operator is a bounded mapping of those spaces into themselves.

III. SYSTEM DESCRIPTION AND ASSUMPTIONS

Consider the feedback system of Fig. 1, where the output of the GPWM is:

$$m(t) = \sum_K M\epsilon_K[\mu(t - KT_d) - \mu(t - KT_d - \tau_K)]. \tag{1}$$

The constant M is the pulse height, $\mu(t)$ is the unit step function, and T_d is the period of the modulator. Also $\epsilon_K \triangleq \text{sgn}\,[\sigma(KT_d)]$. Furthermore, if we define:

$$\begin{aligned}\omega_K(t) \triangleq [\sigma(t) - \epsilon_K A(t - KT_d)][\mu(t - KT_d) \\ - \mu(t - (K+1)T_d)], \quad \forall K \in I^+,\end{aligned} \tag{2}$$

where A (the slope) is a positive constant, then

$$\tau_K = \begin{cases} \min\,\{(t - KT_d) : \omega_K(t) = 0, & t \in [KT_d, (K+1)T_d)]\} \\ T_d, \quad \text{if} \quad \omega_k(t) \neq 0, & \forall t \in [KT_d, (K+1)T_d)] \end{cases}. \tag{3}$$

The above relations are illustrated in Fig. 2.

From eqs. (1) and (3) we see that the GPWM is a causal operator mapping L_{pe} into itself. Furthermore, it is interesting to note that the periodic PWM is derivable from the GPWM by inserting a sampler (operating every T_d seconds) and a zero order hold before the modulator, as shown in Fig. 3. Here the analog of eq. (3) would be

$$\tau_K = \begin{cases} \dfrac{1}{A}\dfrac{\sigma(KT_d)}{\epsilon_K} = \dfrac{1}{A}|\sigma(KT_d)|, & |\sigma(KT_d)| \leqq AT_d \\ T_d, & |\sigma(KT_d)| > AT_d \end{cases}$$

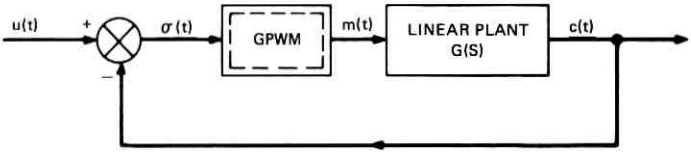


Fig. 1—GPWM feedback system.

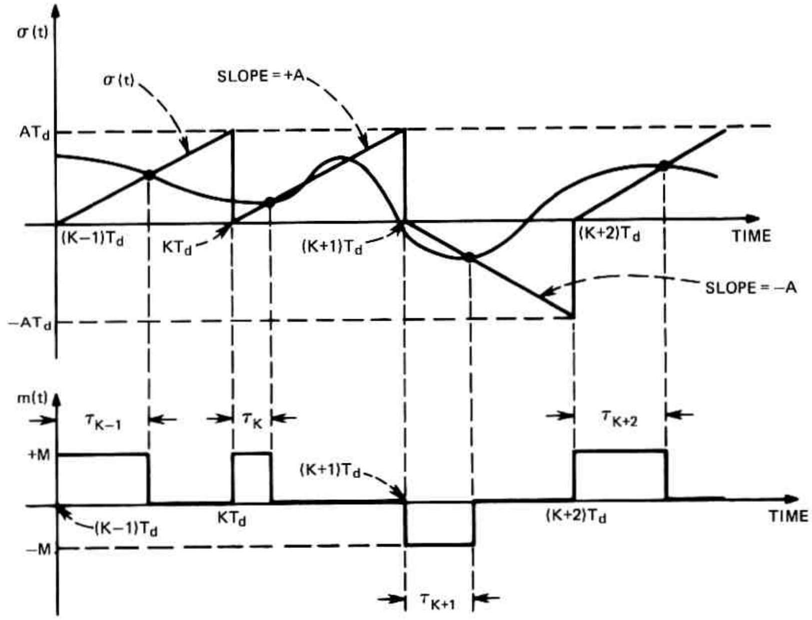


Fig. 2—Modulator definitions.

which is, indeed, the functional relation between the pulse width and the sampled input of a periodic PWM.

It is worth pointing out that various forms of the GPWM process could exist. The modulator may be one-sided ($\epsilon_{\mathcal{K}} = +1, \forall K$) as, for example, in dc power conditioning; it may emit multiple pulses period; or the reference ramp may be replaced by a symmetrical triangle or other similar waveforms. The results of this paper may be extended to any of these variations.

With the foregoing, the following assumptions are also made (see

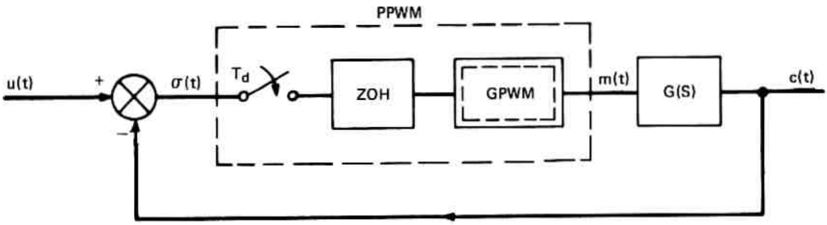


Fig. 3—Derivation of PPWM.

Fig. 1):

- A1. $U(t)$ is absolutely continuous[14] on $[0, \infty)$, and $U(t)$, $\dot{U}(t) \in L_1 \cap L_2$, where $U(t)$ includes an external input and the zero input response of the linear plant.
- A2. $g(t)$, $\dot{g}(t) \in L_1 \cap L_2$; $g(t) = 0$, $t < 0$, where $g(t)$ is the impulse response of the linear subsystem.

Normally, in input-output stability analysis the solution of the system is assumed to exist in the extended space under consideration. However, the constraints of the modulator make this unnecessary.

Lemma 1: Under assumptions A1 and A2, $\sigma(t) \in L_{pe}$ $(p = 1, 2)$.
Proof: From eq. (1), for any finite time $T \in [0, \infty)$ the modulator will produce a finite number of pulses. Thus $m(t) \in L_{pe}(p > 1)$, which implies by virtue of A2 that so does $c(t)$. Hence $\sigma(t) \in L_{pe}$ $(p = 1, 2)$ by A1 and the linearity of the L_p spaces.

IV. STABILITY

The objective of this section is to develop a geometric stability criterion for GPWM systems. Conditions for the system response to belong to $L_p(p \geqq 1)$ will be derived, yielding a geometric criterion in the Popov plane. The result will require that the linear subsystem have a measurable impulse response, satisfying A2, and thus may represent either a lumped or distributed plant. The following extension of a result due to Euler[15] will be useful in establishing the criterion.

Lemma 2: If $x(t)$ is absolutely continuous on $[0, T]$ for any $T \in [0, \infty)$, then:

$$\sum_{K=0}^{N} |x(KT_d)| \leqq \frac{1}{T_d} \int_0^T |x(t)|\,dt + \frac{1}{2} \int_0^T |\dot{x}(t)|\,dt + \tfrac{1}{2}[|x(0)| + |x(NT_d)|], \quad (4)$$

where $N = [T/T_d]$; i.e., the largest integer $\leqq T/T_d$, and the derivative $\dot{x}(t)$ exists almost everywhere.

Proof: For $K = 0, 1, 2, \cdots, N - 1$,

$$\int_{KT_d}^{(K+1)T_d} \left(\frac{t}{T_d} - K - \frac{1}{2}\right) \cdot \frac{d}{dt} |x(t)|\,dt = \tfrac{1}{2}[|x(KT_d)| + |x((K+1)T_d)|] - \frac{1}{T_d} \int_{KT_d}^{(K+1)T_d} |x(t)|\,dt$$

since both $x(t)$ and $(t/T_d - K - \frac{1}{2})$ are absolutely continuous on the interval.[16] In the integrand on the left, we can replace K by $[t/T_d]$

since, on the interval, $t/T_d - [t/T_d] - \frac{1}{2}$ differs from $t/T_d - K - \frac{1}{2}$ only on a set of measure zero. Now addition of the above for $K = 0, 1, 2, \cdots, N - 1$ and then the expression $\frac{1}{2}[|x(0)| + |x(NT_d)|]$ to both sides yields:

$$\sum_{K=0}^{N} |x(KT_d)| = \frac{1}{T_d} \int_0^{NT_d} |x(t)|\,dt + \int_0^{NT_d} (t/T_d - [t/T_d] - \tfrac{1}{2}) \cdot \frac{d}{dt}|x(t)|\,dt + \tfrac{1}{2}[|x(0)| + |x(NT_d)|].$$

Noting that:

$$\int_0^{NT_d} \left(\frac{t}{T_d} - \left[\frac{t}{T_d}\right] - \frac{1}{2}\right) \cdot \frac{d}{dt}|x(t)|\,dt \leqq \frac{1}{2} \int_0^{NT_d} |\dot{x}(t)|\,dt$$

and that $NT_d \leqq T$, we see:

$$\sum_{K=0}^{N} |x(KT_d)| \leqq \frac{1}{T_d} \int_0^{T} |x(t)|\,dt + \frac{1}{2} \int_0^{T} |\dot{x}(t)|\,dt + \tfrac{1}{2}[|x(0)| + |x(NT_d)|].$$

Q.E.D.

Along with the above result, the following observation concerning the modulator will be of interest in what follows.

Lemma 3: Consider Fig. 1. If $m(t) \in L_p$ for any $p \in [1, \infty)$, then it belongs to L_p for all $p \in [1, \infty]$.

Proof: Suppose $m(t) \in L_{\bar{p}}$, for some $\bar{p} \in [1, \infty)$. Then

$$\int_0^\infty |m(t)|^{\bar{p}} dt = M^{\bar{p}} \sum_{K=0}^{\infty} \tau_K < \infty$$

and, since M is a finite number, we see that, for any p,

$$\|m(t)\|_{L_p} = M^p \sum_{K=0}^{\infty} \tau_K < \infty$$

and thus $m(t) \in L_p$ for all $p \in [1, \infty]$ [of course, $m(t) \in L_\infty$ by virtue of (1)].

With the foregoing we are now in a position to state the main result of this paper.

Theorem: Consider the GPWM feedback system of Fig. 1. Suppose there exist two numbers, $q_1 \in R^+$, $q_2 \neq 0$, such that:

(*i*) $\dfrac{1}{q_1 q_2} > \dfrac{1}{T_d}\|g(t)\|_{L_1} + \tfrac{1}{2}[\|\dot{g}(t)\|_{L_1} + |g(0)|] - \dfrac{A}{M}$ *and*

(*ii*) $\mathrm{Re}\,[(1 + j\omega q_1)G(j\omega)] \geqq \dfrac{1}{q_2}, \ \forall \omega \in R^+,$

where

$$G(j\omega) = \int_0^\infty g(t)e^{-j\omega t}dt.$$

Then

$$C(t) \in L_p(p \geqq 1).$$

Remark: If $U(t) \in L_p$ as well, then the system will be termed $L_1 \cap L_2 \cap L_p$ stable (bounded).

Proof: Consider Fig. 1. We note that condition (*ii*) implies (see, for example, Ref. 17) that:

$$\int_0^T \left[\sigma(t) + \frac{1}{q_2}m(t)\right]\cdot m(t)dt + q_1\int_0^T m(t)\dot{\sigma}(t)dt \leqq \int_0^T \bar{u}(t)\cdot m(t)dt, \quad \forall\, T \in [0, \infty), \quad (5)$$

where $\bar{u}(t) = u(t) + q_1\dot{u}(t)$. Now from the defining relations of the modulator (see also Fig. 2), the GPWM is an *e*-positive operator;[18] i.e.,

$$\int_0^T \sigma(t)\cdot m(t)dt \geqq 0^*, \quad \forall\, T \in [0, \infty).$$

Thus:

$$\frac{1}{q_2}\int_0^T m^2(t)dt + q_1\int_0^T m(t)\dot{\sigma}(t)dt \leqq \left[\int_0^T \bar{u}^2(t)dt\right]^{\frac{1}{2}}\left[\int_0^T m^2(t)dt\right]^{\frac{1}{2}}, \quad (6)$$

where Schwarz's inequality has been used on the rhs of (5). Using

$$m(t) = \sum_{K=0}^{N} M\epsilon_k[\mu(t - KT_d) - \mu(t - KT_d - \tau_K)] \quad \text{for } t \in [0, T):^\dagger$$

$$\frac{M^2}{q_2}\sum_{K=0}^{N}\tau_K + q_1M\sum_{K=0}^{N}[|\sigma(KT_d + \tau_K)| - |\sigma(KT_d)|] \leqq M\|\bar{u}(t)\|_{L_2}\left[\sum_{K=0}^{N}\tau_K\right]^{\frac{1}{2}}, \quad (7)$$

in which we have used

$$\epsilon_k\begin{Bmatrix}\sigma(KT_d + \tau_K)\\ \sigma(KT)\end{Bmatrix} = \begin{Bmatrix}|\sigma(KT_d + \tau_K)|\\ |\sigma(KT_d)|\end{Bmatrix}.$$

[*] Note on L_2 this is not true for the periodic PWM.

[†] If the truncation time T should occur during the Nth pulse, then, of course, $\tau_N = T - NT_d$.

Inequality (7) follows from the fact that $\sigma(t)$ is absolutely continuous, which will be shown below.

Now in view of (2) and (3), $\tau_K \leqq |\sigma(KT_d + \tau_K)|/A$, and thus:

$$M\left(q_1A + \frac{M}{q_2}\right)\sum_{K=0}^{N} \tau_K - M\|\bar{u}(t)\|_{L_2}\left[\sum_{K=0}^{N} \tau_K\right]^{\frac{1}{2}} \leqq q_1M\sum_{K=0}^{N}|\sigma(KT_d)|. \qquad (8)$$

We observe that $C(t)$ [and hence $\sigma(t)$ by A1, Section III] is absolutely continuous, since it is the indefinite integral of a summable (Lebesque) function. Therefore, Lemma 2 is applicable to $\sigma(t)$ and:

$$\begin{aligned}\sum_{K=0}^{N}|\sigma(KT_d)| \leqq{}& \frac{1}{T_d}\int_0^T|\sigma(t)|dt \\ &+ \frac{1}{2}\int_0^T|\dot{\sigma}(t)|dt + \tfrac{1}{2}[|\sigma(0)| + |\sigma(NT_d)|] \\ \leqq{}& \int_0^T\left(\frac{1}{T_d}|u(t)| + \frac{1}{2}|\dot{u}(t)|\right)dt \\ &+ \int_0^T\left(\frac{1}{T_d}|C(t)| + \frac{1}{2}|\dot{C}(t)|\right)dt + \sup_{t\geqq 0}|\sigma(t)|. \qquad (9)\end{aligned}$$

Furthermore,

$$\int_0^T|C(t)|dt \leqq M\|g(t)\|_{L_1}\sum_{K=0}^{N}\tau_K$$

and $\qquad (10)$

$$\int_0^T|\dot{C}(t)|dt \leqq M[\|\dot{g}(t)\|_{L_1} + |g(0)|]\sum_{K=0}^{N}\tau_k.$$

Using (9) and (10) in (8) then implies:

$$\begin{aligned}MZ&\left[\left(\sum_{K=0}^{N}\tau_K\right)^{\frac{1}{2}} - \frac{\|\bar{u}(t)\|_{L_2}}{2Z}\right]^2 \\ &\leqq q_1M\left(\frac{1}{T_d}\int_0^T|u(t)|dt + \frac{1}{2}\int_0^T|\dot{u}(t)|dt\right) \\ &\qquad\qquad + q_1M\sup_{t\geqq 0}|\sigma(t)| + \frac{M\|\bar{u}(t)\|_{L_2}^{\,2}}{4Z} \\ &\leqq q_1M\left(\frac{1}{T_d}\|u(t)\|_{L}1 + \frac{1}{2}\|\dot{u}(t)\|_{L_1}\right) \\ &\qquad\qquad + q_1M\sup_{t\geqq 0}|\sigma(t)|^{*} + \frac{M\|\bar{u}(t)\|_{L_2}^{\,2}}{4Z} < \infty,\end{aligned}$$

[*] It is a simple matter to show that, under the hypotheses of the theorem, the system is L_∞ stable and, since $u(t) \in L_\infty$, $\sup_{t\geqq 0}|\sigma(t)| < \infty$.

where

$$Z = q_1 A - \frac{Mq_1}{T_d}\|g(t)\|_{L_1} - \frac{Mq_1}{2}[\|\dot{g}(t)\|_{L_1} + |g(0)|] + \frac{M}{q_2}.$$

For $Z > 0$ [condition (*i*) of the theorem], we have $\sum_{K=0}^{N} \tau_K \leqq Q$(independent of T) $< \infty$, and thus $m(t) \in L_p$, which implies by A2, Section III, that $C(t)$ does also, and the theorem is proved.

Comments: (*a*) Condition (*ii*) of the theorem is similar to a Popov condition for feedback systems with static, sector nonlinearities, although the GPWM does not strictly belong to that class.

(*b*) The condition allows a tradeoff between the slope of the stability line and its interaction with the real axis of the Popov plane.

(*c*) Because of the constraints of the modulator, the modified linear plant does not have to be a *strictly* positive operator, as is commonly the case.[18]

(*d*) Since the assumptions are sufficient to ensure that $U(t)$ [and $g(t)] \to 0$ as $t \to \infty$, the theorem also guarantees that $\sigma(t) \to 0$.

V. ACKNOWLEDGMENT

The authors are grateful to R. A. Skoog of Bell Laboratories for his very useful suggestions and a careful reading of the manuscript.

REFERENCES

1. Andeen, R. E., IRE Trans. Auto. Control, 1960, *5*, p. 306.
2. Delfeld, F. R., and Murphy, G. J., IRE Trans. Auto. Control, 1961, *6*, p. 283.
3. Polak, E., IRE Trans. Auto. Control, 1961, *6*, p. 276.
4. Ghonaimy, M. A. R., and Aly, G. M., Int. J. Control, 1972, *16*, p. 737.
5. Kadota, T. T., and Bourne, H. C., IRE Trans. Auto. Control, 1961, *6*, p. 266.
6. Murphy, G. J., and Wu, S. H., IEEE Trans. Auto. Control, 1964, *9*, p. 434.
7. Skoog, R. A., IEEE Trans. Auto. Control, 1968, *13*, p. 532.
8. Skoog, R. A., and Blankenship, G. L., IEEE Trans. Auto. Control, 1970, *15*, p. 300.
9. Black, H. S., *Modulation Theory*, New York: Van Nostrand, 1953.
10. Fallside, F., Proc. IEEE, 1968, *115*, p. 218.
11. Mokrytzki, B., IEEE Trans. Industry and General Applications, 1967, *3*, p. 493.
12. Kuntsevich, V. M., Auto. and Remote Control, 1972, p. 1124.
13. Zames, G., IEEE Trans. Auto. Control, 1966, *11*, p. 228.
14. Kolmogorov, A. N., and Fomin, S. V., *Introductory Real Analysis* (transl.), Englewood Cliffs, N. J.: Prentice-Hall, 1970.
15. Stanaitis, *Introduction to Sequences, Series, and Improper Integrals*, San Francisco: Holden-Day, 1967.
16. Natanson, I. P., *Theory of Functions of a Real Variable* (transl.), New York: Frederick Unger, 1961.
17. Aizerman, M. A., and Gantmacher, F. R., *Absolute Stability of Regular Systems* (transl.), San Francisco: Holden-Day, 1964.
18. Holtzman, J. M., *Nonlinear System Theory: A Functional Analysis Approach*, Englewood Cliffs, N. J.: Prentice-Hall, 1970.

THE BELL SYSTEM TECHNICAL JOURNAL
Vol. 52, No. 10, December, 1973
Printed in U.S.A.

Theory of Minimum Mean-Square-Error QAM Systems Employing Decision Feedback Equalization

By D. D. FALCONER and G. J. FOSCHINI

(Manuscript received June 7, 1973)

Decision feedback equalization is presently of interest as a technique for reducing intersymbol interference in high-rate PAM data communications systems. The basic principle is to cancel out intersymbol interference arising from previously decided data symbols at the receiver, leaving remaining intersymbol interference components to be handled by linear equalization. In this work we consider the application of decision feedback equalization to quadrature-amplitude modulation (QAM) transmission, in which two independent information streams modulate quadrature carriers. Extending Salz's treatment in a companion paper of decision feedback for a baseband channel, we derive the form of the optimum receiver filters via a matrix Wiener-Hopf analysis. We obtain explicit analytical expressions for minimum mean-square error and optimum transmitting filters. The optimization is subject to a constraint on the transmitted signal power and assumes no prior decision errors. The class of QAM transmitter and receiver structures treated here is actually much larger than the class usually considered for QAM systems. However, our results for decision feedback equalization show that, for nonexcess bandwidth systems, optimum performance is achievable without taking advantage of the most general structure. If the transmitter is required to have the conventional QAM structure, study of the time continuous system that gives rise to the sampled data system considered here demonstrates that under quite general assumptions a nonexcess bandwidth system is optimum. Finally, the explicit description of the optimum transmitting matrix filter follows from an information-theoretic "water-pouring" algorithm in conjunction with the determination of the form of the points of maxima of a determinant extremal problem.

I. INTRODUCTION

Interest has recently intensified in receiver structures which hopefully will permit higher data symbol rates than are possible with conventional demodulator/linear equalizer structures having the same error probability. The decision feedback equalizer is an example of a receiver component that can have important performance advantages over a linear equalizer operating over dispersive channels with additive noise.[1–7] The basic structure of a decision feedback equalizer (DFE) is shown in Fig. 1. The function of the filter in the feedback path is to cancel "postcursors" of the channel's impulse response; that is, intersymbol interference components arising from *previously* decided symbols. Thus, the job of the linear filter in the forward path is to minimize (according to some criterion) "precursors" of the channel's impulse response which cause intersymbol interference from future data symbols. Of course, there is a possibility of error propagation with this nonlinear feedback structure. We avoid this intractable problem by assuming that no erroneous decisions pass into the feedback filter. Thus, our results provide a performance lower bound. Earlier experimental studies indicated that error propagation is not a serious problem on some channels.[3,4]

Price[6] (whose bibliography on the subject is extensive) has derived asymptotic formulas (allowing for an infinite number of equalizer taps) for error probability, optimum transmitter pulse spectrum, and communication efficiency for the "zero-forcing" DFE, which minimizes the noise variance at the DFE output subject to the constraint that the intersymbol interference is zero at the receiver's sampling instants. As is the case for linear equalization, the mean-square-error (MSE) criterion is more general than the zero-forcing criterion. The MSE criterion minimizes the mean square of the total error (noise plus residual intersymbol interference) at the DFE output.[2,5] Asymptotic results and illuminating calculations of performance for MSE-minimizing DFE's are contained in a companion paper by Salz.[7]

All previous theoretical studies of decision feedback equalization have assumed a "baseband" linear PAM channel model depicted in

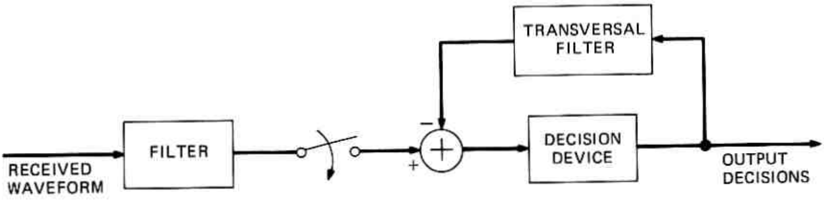


Fig. 1—Basic decision feedback equalizer structure.

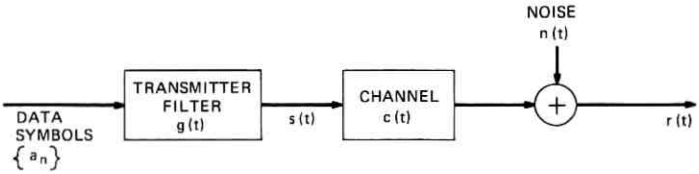


Fig. 2—Baseband channel model.

Fig. 2. The transmitted waveform $s(t)$ is

$$s^{(b)}(t) = \sum_n a_n g^{(b)}(t - nT),$$

where the data symbols $\{a_n\}$ are statistically independent discrete-valued random variables from a finite set and $g^{(b)}(t)$ is some suitable transmitted pulse waveform. The channel output waveform is then

$$r^{(b)}(t) = \sum_n a_n h^{(b)}(t - nT) + n^{(b)}(t),$$

where the overall impulse response is

$$h^{(b)}(t) = \int_{-\infty}^{\infty} c^{(b)}(\tau) g^{(b)}(t - \tau) d\tau$$

and $n^{(b)}(t)$ is additive noise. This model is certainly valid for a real linear channel accepting every T second a pulse of the form $a_n g^{(b)}(t)$. It is also valid for the important case where the linear channel $c^{(b)}(t)$ is actually the baseband equivalent of a passband channel when the modulation is either double-, vestigial-, or single-sideband. (See Ref. 8, Chapter 7.) Of course, $c^{(b)}(t)$ then depends on the carrier frequency and on any phase offset between the reference carriers at the modulator and demodulator.

In this paper we extend the asymptotic DFE theory to the case of QAM (quadrature amplitude modulation) signaling, for which the baseband model of Fig. 2 is not sufficient. We summarize our results at the end of Section II. The most general QAM transmitter structure is illustrated in Fig. 3. Two independent data sequences enter a lattice network comprising filters with impulse responses $g_{11}(t)$, $g_{21}(t)$, $g_{12}(t)$, and $g_{22}(t)$. Modulation is done with two quadrature carriers with frequency f_0 Hz. In practice, most QAM transmitters are specialized to the case $g_{11}(t) = g_{22}(t)$; $g_{21}(t) = -g_{12}(t)$.* We call the class of trans-

* Indeed, it is often assumed that $g_{12}(t)$ and $g_{21}(t)$ are zero.

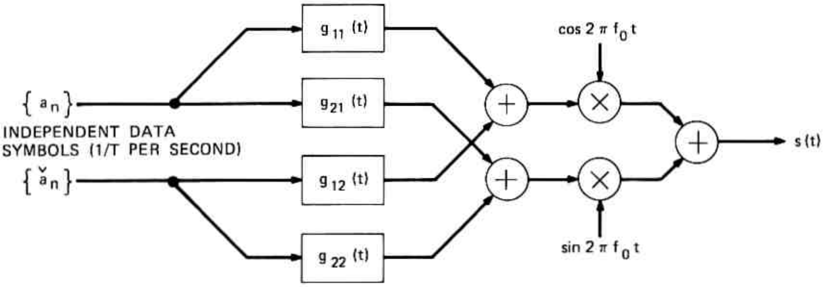


Fig. 3—General QAM transmitter structure.

mitters with this special structure the class of "passband" transmitters ($\mathcal{P}$). We show in later sections that optimum performance is in general achievable by restricting the transmitter to this class or a simple variant thereof. It is worth noting that QAM systems with passband transmitters are mathematically equivalent to baseband PAM systems, but with complex impulse responses and information symbols.[9,10]

For the most general QAM structure, the waveform $s(t)$ is expressed in terms of two-dimensional vectors and matrices. Define the vector $\mathbf{a}_n$ to be the nth pair of information symbols,

$$\mathbf{a}_n = \begin{pmatrix} a_n \\ \check{a}_n \end{pmatrix}. \tag{1}$$

The most general QAM transmitter is characterized by the matrix filter

$$g(t) = \begin{pmatrix} g_{11}(t) & g_{12}(t) \\ g_{21}(t) & g_{22}(t) \end{pmatrix}. \tag{2}$$

Then the structure of Fig. 3 yields

$$s(t) = (\cos 2\pi f_0 t, \sin 2\pi f_0 t) \sum_n g(t - nT)\mathbf{a}_n. \tag{3}$$

We assume that the data symbols are uncorrelated discrete-valued random variables with variance σ_a^2. Thus

$$\langle \mathbf{a}_n \mathbf{a}_m^\dagger \rangle = \sigma_a^2 \delta_{nm} I, \tag{4}$$

where $\langle\ \rangle$ denotes expectation, † denotes transpose,* δ_{nm} is the Kronecker delta, and I is the identity matrix. The transmitted power is

* The symbol † will denote conjugate transpose for complex vectors and matrices.

then given by

$$P \equiv \lim_{\tau \to \infty} \frac{1}{2\tau} \int_{-\tau}^{\tau} \langle s^2(t) \rangle dt = \frac{\sigma_a^2}{2T} \int_{-\infty}^{\infty} [g_{11}^2(t) + g_{12}^2(t) + g_{21}^2\,(t) + g_{22}^2(t)]dt$$

$$= \frac{\sigma_a^2}{4\pi T} \int_{-\infty}^{\infty} [|G_{11}(\omega)|^2 + |G_{12}(\omega)|^2 + |G_{21}(\omega)|^2 + |G_{22}(\omega)|^2]d\omega, \quad (5)$$

where $G_{ij}(\omega)$ is the Fourier transform of $g_{ij}(t)$. For future reference note that we can also write P as

$$P = \frac{\sigma_a^2}{4\pi T} \int_{-\pi/T}^{\pi/T} \sum_n \left[\left| G_{11}\left(\omega + \frac{2\pi n}{T}\right)\right|^2 + \left| G_{12}\left(\omega + \frac{2\pi n}{T}\right)\right|^2 \right.$$

$$\left. + \left| G_{21}\left(\omega + \frac{2\pi n}{T}\right)\right|^2 + \left| G_{22}\left(\omega + \frac{2\pi n}{T}\right)\right|^2 \right] d\omega$$

$$= \frac{\sigma_a^2}{4\pi T} \int_{-\pi/T}^{\pi/T} \sum_n \operatorname{tr} G\left(\omega + \frac{2\pi n}{T}\right)^{\dagger} G\left(\omega + \frac{2\pi n}{T}\right) d\omega, \quad (6)$$

where

$$G(\omega) = \begin{pmatrix} G_{11}(\omega) & G_{12}(\omega) \\ G_{21}(\omega) & G_{22}(\omega) \end{pmatrix}$$

is the matrix frequency response of the transmitter. We use tr to denote the trace of a matrix.

Later sections will show that without an initial assumption of the special passband transmitter structure the treatment of decision feedback equalization for two- (and hence higher) dimensional signals is a nontrivial generalization of the baseband signal case.

II. THE CHANNEL MODEL AND SUMMARY OF RESULTS

The impulse response $q(t)$ of any linear channel can be resolved about a center frequency f_0:

$$q(t) = c_1(t) \cos 2\pi f_0 t - c_2(t) \sin 2\pi f_0 t. \quad (7)$$

It is easy to show that the channel model of Fig. 4 yields exactly the above impulse response, and thus any linear channel can be conveniently represented in terms of an arbitrary center frequency f_0 by the structure of Fig. 4. We note in passing that the so-called "in-phase" and "quadrature" impulse responses $c_1(t)$ and $c_2(t)$ are often interpreted as the real and imaginary parts, respectively, of the "complex envelope" of the impulse response $q(t)$ with respect to the frequency f_0.

We assume that the low-pass transmitter impulse responses $\{g_{ij}(t)\}$

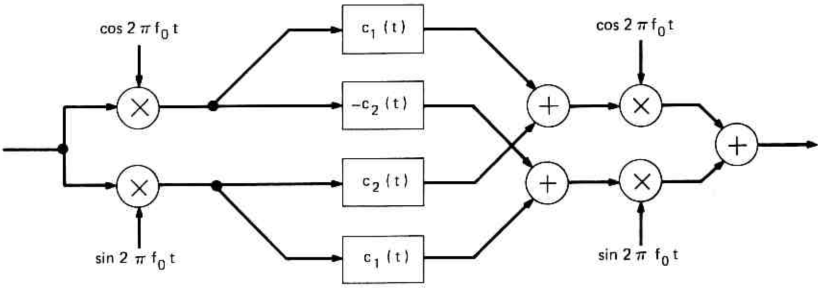


Fig. 4—A passband channel model.

are all strictly bandlimited to lie within the frequency interval $(-f_0, f_0)$; otherwise, the system would suffer distortion from aliasing effects. There is then no loss of generality in assuming that the channel's in-phase and quadrature impulse responses $c_1(t)$ and $c_2(t)$ are also strictly bandlimited to this interval.

With these assumptions, double-frequency terms disappear,[11] and it is easily shown that the noise-free channel output

$$\int_{-\infty}^{\infty} q(\tau)s(t-\tau)d\tau, \tag{8}$$

where $q(t)$ is given by (7) and $s(t)$ by (3), can be written

$$\tfrac{1}{2}(\cos 2\pi f_0t, \sin 2\pi f_0t) \sum_n \int_{-\infty}^{\infty} c(t-\tau)g(\tau-nT)d\tau\mathbf{a}_n, \tag{9}$$

where $c(t)$ is the matrix

$$c(t) = \begin{pmatrix} c_1(t) & c_2(t) \\ -c_2(t) & c_1(t) \end{pmatrix}, \tag{10}$$

the matrix $g(t)$ is given by (2), and integration of matrices and vectors means integration of each entry.

Consider receiver structures whose "front end" is the type shown in Fig. 5—sine and cosine demodulators followed by identical ideal low-pass filters that are strictly bandlimited to $(-f_0, f_0)$ and whose outputs are labelled $r(t)$ and $\check{r}(t)$, respectively. This structure causes no loss of information, since any bandlimited input signal can be reproduced exactly if the outputs $r(t)$ and $\check{r}(t)$ are multiplied by $\cos 2\pi f_0t$ and $\sin 2\pi f_0t$, respectively, and then added together. The function of the low-pass filters is to remove double frequency terms; it will turn out that the "front end" will be followed by a band-limiting matched filter, so that the low-pass filters are not necessary.

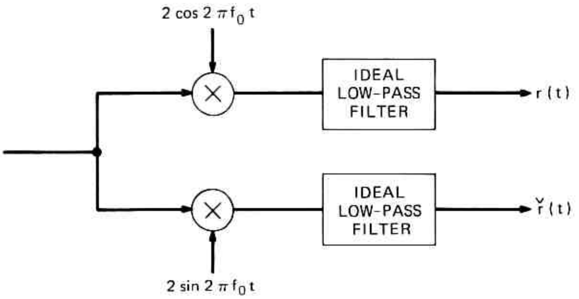


Fig. 5—Receiver "front end."

The low-pass outputs $r(t)$ and $\check{r}(t)$ can be written in vector form as

$$\mathbf{r}(t) = \begin{pmatrix} r(t) \\ \check{r}(t) \end{pmatrix} = \frac{1}{2} \sum_n h(t - nT)\mathbf{a}_n + \mathbf{v}(t), \tag{11}$$

where the matrix impulse response $h(t)$ is

$$h(t) = \int_{-\infty}^{\infty} c(\tau) g(\tau - \tau) d\tau, \tag{12}$$

and the components of the vector

$$\mathbf{v}(t) = \begin{pmatrix} n_c(t) \\ n_s(t) \end{pmatrix}$$

represent additive noise. Assuming that the additive noise in the channel is white with double-sided power spectral density $N_0/2$, it can be shown[11] that $n_c(t)$ and $n_s(t)$ are statistically independent stationary zero mean processes; each is the result of passing a stationary white noise with double-sided power spectral density N_0 through an ideal low-pass filter. Noise outside the signal bandwidth will be eliminated by a matched filter. Accordingly, we take the covariance matrix of the noise to be

$$\langle \mathbf{v}(t)\mathbf{v}^{\dagger}(t + \tau) \rangle = N_0 I \delta(\tau), \tag{13}$$

where I is the identity matrix and $\delta(t)$ is a "unit-area delta function." The mathematical model for the transmitter and channel is now complete and is summarized in Fig. 6a.

We remark in passing that linear modulation of a single stream of data symbols (e.g., single-sideband or vestigial-sideband modulation) constitutes a special case of this model. In that case, $g_{12}(t) = g_{22}(t) = 0$, and the receiver front end consists of a cosine demodulator with some phase shift θ, followed by an ideal low-pass filter. Then the overall

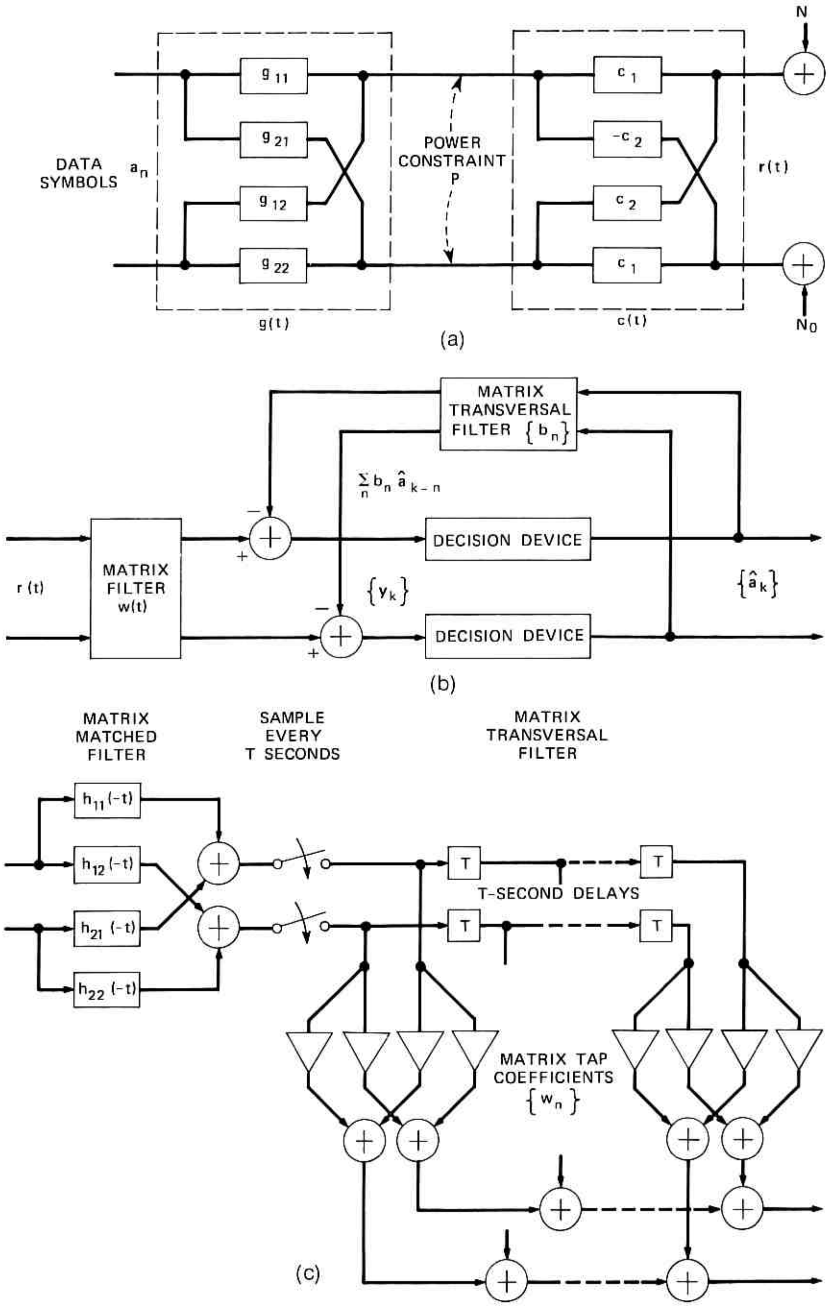


Fig. 6—(a) Canonical mathematical model of transmitter and channel. (b) QAM decision feedback equalizer structure. (c) Structure of the matrix filter $w(t)$.

impulse response is a scalar function of time (which depends on the receiver phase shift θ), and hence all two-dimensional matrices and vectors in the present treatment would be replaced by scalar quantities (see Ref. 7).

The following list summarizes our main results:

(*i*) The optimum linear forward filter at the receiver for a given transmitter channel cascade, $H(\omega) = C(\omega)G(\omega)$, is found to have the form

$$\text{Const} \times H(\omega)/\sqrt{\Phi(e^{-j\omega T}) + \frac{N_0}{\sigma_a^2} I}^{\wedge \phi},$$

where

$$\Phi(e^{-j\omega T}) = \frac{1}{T}\sum_n H^\dagger\left(\omega + \frac{2\pi n}{T}\right)H\left(\omega + \frac{2\pi n}{T}\right)$$

and $\sqrt{}^{\wedge \phi}$ denotes minimum-phase square root. This filter can be viewed as a matrix matched filter followed by an anticausal matrix tap delay line. (See Sections III and VI.)

(*ii*) For a given transmitter power spectral density: if a nonexcess bandwidth system (Section V) is required, an optimum transmitter is found and it is passband; conversely if the transmitter is taken to be passband, the optimum system is a nonexcess bandwidth one (Section III).

(*iii*) Given a passband transmitter, the MSE (the sum of the mean-square errors of the two unquantized receiver outputs) is

$$2\sigma_a^2 \exp\left\{-\frac{T}{2\pi}\int_{-\pi/T}^{\pi/T} \log\left[\frac{\sigma_a^2}{N_0}X_{eq}(\omega) + 1\right]d\omega\right\},$$

where

$$X_{eq}(\omega) = \sum_n \left|G_1\left(\omega + \frac{2\pi n}{T}\right) + jG_2\left(\omega + \frac{2\pi n}{T}\right)\right|^2 \times \left|C_1\left(\omega + \frac{2\pi n}{T}\right) + jC_2\left(\omega + \frac{2\pi n}{T}\right)\right|^2$$

and $G_1(\omega) = G_{11}(\omega) = G_{22}(\omega)$ and $G_2(\omega) = G_{12}(\omega) = -G_{21}(\omega)$ (Section VI).

(*iv*) The optimum transmitter power spectral density is found for the class of passband transmitters meeting an output power constraint (Section VII). This optimal density has a water-pouring description. (Since the processing capability considered here represents an advancement over conventional linear equalization, this emergence of an information theoretic type density is perhaps not surprising.)

(*v*) Although we do not constrain the in-phase and quadrature mean-square errors to be equal, we show that for the above-mentioned optimal systems the errors on the two data streams have equal variances and are uncorrelated (Section VI).

In a nutshell, the system optimization proceeds as follows:

(*i*) Find the optimal receiver for each transmitter.

(*ii*) Find an optimal transmitter for each transmitter power spectral density.

(*iii*) Find the optimal transmitter power spectral density.

Then we reverse, using the solution of (*iii*) to specify an optimal transmitter and then using this optimal transmitter to specify the optimal receiver.

III. THE RECEIVER OPTIMIZATION PROBLEM

The DFE structure consists of a linear matrix filter $w(t)$, quantizer, and a transversal feedback filter with matrix tap coefficients $\{b_n\}$ which processes previously made decisions as shown in Fig. 6b. The kth sampled vector input to the quantizers is written

$$\mathbf{y}_k = \int_{-\infty}^{\infty} w(\tau)\mathbf{r}(kT - \tau)d\tau - \sum_{n=1}^{\infty} b_n\hat{\mathbf{a}}_{k-n}, \tag{14}$$

where $\hat{\mathbf{a}}_n$ is the receiver's decision on the nth data symbol-pair. Note that we allow the feedforward and feedback matrix filters to have infinite-duration impulse responses. We also replace $\hat{\mathbf{a}}_{k-n}$ in (14) by the true data symbol vector $\mathbf{a}_{k-n}$ for mathematical tractibility; thus, we in effect postulate a "magic genie" preceding the feedback filter who corrects any decision errors. The genie's existence is immaterial up to the time of the first decision error, and hence our expression for MSE is certainly valid up to that time.

The error vector $\boldsymbol{\varepsilon}_n$ is defined to be the difference between $\mathbf{y}_k$ and the correct symbol $\mathbf{a}_k$,

$$\boldsymbol{\varepsilon}_k = \mathbf{y}_k - \mathbf{a}_k, \tag{15}$$

and the MSE is defined to be the trace of the *error matrix* e_0, where

$$e_0 = \langle \boldsymbol{\varepsilon}_n \boldsymbol{\varepsilon}_n^{\dagger} \rangle, \tag{16}$$

the average being with respect to the noise and the data symbol sequence. Note that e_0 is positive semidefinite and symmetric.

Substituting (14) and (15) into (16) and using the noise correlation matrix (13) and the data symbol correlation matrix (4), we can write

$$\begin{aligned} e_0 = \sigma_a^2 \sum_{n \geqq k} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} w(\tau_1) h[(k-n)T - \tau_1] h^\dagger[(k-n)T - \tau_2] \\ \times w^\dagger(\tau_2) d\tau_1 d\tau_2 + N_0 \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} w(\tau_1) w^\dagger(\tau_2) \delta(\tau_1 - \tau_2) d\tau_1 d\tau_2 \\ + \sigma_a^2 I + \sigma_a^2 \sum_{n=1}^{\infty} \left[b_n - \int_{-\infty}^{\infty} w(\tau) h(nT - \tau) d\tau \right] \\ \times \left[b_n - \int_{-\infty}^{\infty} w(\tau) h(nT - \tau) d\tau \right]^\dagger - \sigma_a^2 \int_{-\infty}^{\infty} w(\tau) h(-\tau) d\tau \\ - \sigma_a^2 \int_{-\infty}^{\infty} h^\dagger(-\tau) w^\dagger(\tau) d\tau. \quad (17) \end{aligned}$$

We immediately observe that tr e_0 is minimized with respect to the matrices $\{b_n\}$ if and only if for all $n \geqq 1$, $b_n = s_n$, where

$$s_n \equiv \int_{-\infty}^{\infty} w(\tau) h(nT - \tau) d\tau \quad \text{all } n \quad (18)$$

represents the matrix samples for $n \geqq 1$ of the impulse response of the transmitter/channel-receiver filter combination. Then, once the $\{b_n\}$ are optimized in this way, the remaining terms comprising the matrix e_0 can be written

$$\begin{aligned} e_0 = \sigma_a^2 \sum_{n \leqq 0} \left[\delta_{n0} I - \int_{-\infty}^{\infty} w(\tau) h(nT - \tau) d\tau \right] \\ \times \left[\delta_{n0} I - \int_{-\infty}^{\infty} w(\tau) h(nT - \tau) d\tau \right]^\dagger + N_0 \int_{-\infty}^{\infty} w(\tau) w^\dagger(\tau) d\tau. \quad (19) \end{aligned}$$

We wish to minimize tr e_0 with respect to the entries in the matrix $w(t)$. Notice from eq. (16) that tr e_0 is a positive quadratic form. Thus from Ref. 12 we set the gradient equal to zero to determine the stationary points which are necessarily points of global minima. We shall find that there is only one solution.

Proceeding with the calculus of variations method, we replace

$$w(t) = \begin{pmatrix} w_{11}(t) & w_{12}(t) \\ w_{21}(t) & w_{22}(t) \end{pmatrix}$$

by

$$w(t) + \begin{pmatrix} \epsilon_{11}\eta_{11}(t) & \epsilon_{12}\eta_{12}(t) \\ \epsilon_{21}\eta_{21}(t) & \epsilon_{22}\eta_{22}(t) \end{pmatrix},$$

where the $\eta_{ij}(t)$ are arbitrary. Setting

$$\begin{bmatrix} \dfrac{\partial \text{tr } e_0}{\partial \epsilon_{11}} & \dfrac{\partial \text{tr } e_0}{\partial \epsilon_{12}} \\ \dfrac{\partial \text{tr } e_0}{\partial \epsilon_{21}} & \dfrac{\partial \text{tr } e_0}{\partial \epsilon_{22}} \end{bmatrix} = \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix} \quad \text{at } \epsilon_{11} = \epsilon_{12} = \epsilon_{21} = \epsilon_{22} = 0,$$

we get

$$-\sigma_a^2 h^\dagger(-\tau) + \sigma_a^2 \sum_{n \leqq 0} \int_{-\infty}^{\infty} w(\tau_1) h(nT - \tau_1) d\tau_1 h^\dagger(nT - \tau) + N_0 w(\tau) = [0]$$

or

$$w(\tau) = \sum_{n \leqq 0} w_n h^\dagger(nT - \tau), \tag{20}$$

where

$$w_n = \frac{\sigma_a^2}{N_0} (\delta_{n0} I - s_n) \quad n \leqq 0. \tag{21}$$

This means that the matrix filter $w(t)$ can be interpreted as a matrix matched filter with impulse response $h^\dagger(-t)$ followed by a sampler and matrix transversal filter with matrix tap coefficients $\{w_n\}$. Note that the transversal filter is "anticausal"—that is, $w_n = [0]$ for $n > 0$. The structure of $w(t)$ is illustrated in Fig. 6c.

Furthermore, substitution of the optimum filter (10) back into expression (19) for the error matrix e_0 results in

$$e_0 = \sigma_a^2 (I - s_0)^\dagger \tag{22a}$$

and from (21)

$$e_0 = N_0 w_0^\dagger. \tag{22b}$$

An explicit solution for the optimum tap coefficient matrices $\{w_n\}$ can be obtained by postmultiplying (20) by $h(mT - \tau)$ and integrating, using (21) and (18) and the definition

$$\phi_n \equiv \int_{-\infty}^{\infty} h^\dagger(-\tau) h(nT - \tau) d\tau$$

to yield

$$\sum_{m \leqq 0} w_m \left[\phi_{n-m} + \frac{N_0}{\sigma_a^2} \delta_{m-n,0} I \right] = \delta_{n0} I \quad \text{for } n \leqq 0. \tag{23}$$

We recognize eq. (23) as a classical Wiener-Hopf equation for which we are assured the existence of a unique solution.[13]

We attach a plus (minus) subscript to any matrix sequence whose value is the zero matrix on the strictly negative (positive) integers.*

* A matrix sequence u_+, zero on the strictly negative integers, is referred to as *causal*. A sequence u_-, zero on the strictly positive integers, is referred to as *anticausal*.

By 1 we mean the matrix sequence vanishing everywhere but zero, where the value is I. Then (23) is written

$$w_{-} * \mathscr{r} = 1 \quad (n \leqq 0), \tag{24}$$

where we have used $\mathscr{r}$ to denote the sequence sum which we observe is the Fourier coefficient sequence for a positive definite Hermitian matrix function whose determinant is uniformly above (N_0/σ_a^2). Hence $\mathscr{r}$ admits a causal, anticausal deconvolution of the kind provided by Wiener and Akutowicz[14] (generalizing a result of Szegö). Based on Ref. 14, we can say

$$\mathscr{r} = u_{-} * u_{+}, \quad \{(u_{-})_n = [(u_{+})_{-n}]^{\dagger}\}_{n=0}^{-\infty}, \tag{25}$$

where we have used $(u_{-})_n$ to denote the nth entry in the u_{-} sequence (similarly for u_{+}). Corresponding to u_{-}, we have its convolution inverse, $[u_{-}]^{-1}$, which is also anticausal.

From what we have just said,

$$[(u_{+})_0]^{-1}[u_{-}]^{-1} * u_{-} * u_{+} = 1, \tag{26}$$

and so $w_{-} = [(u_{+})_0]^{-1}(u_{-})^{-1}$ and, in particular,

$$w_0 = (u_0 u_0^{\dagger})^{-1}, \tag{27}$$

where in the last equation the negative subscripts have been suppressed. Thus the anticausal transversal filter is found from eqs. (23) through (26) to have a frequency response inversely proportional to

$$\sqrt{\Phi(e^{-j\omega T}) + \frac{N_0}{\sigma_a^2} I}^{\,\wedge \phi},$$

where the notation $\sqrt{}^{\,\wedge \phi}$ means minimum-phase square root and $\Phi(e^{-j\omega T})$ is the discrete Fourier transform of the matrix sequence $\{\phi_n\}$. Recalling eq. (22b), we have the following expression for the error matrix:

$$e_0 = N_0 w_0^{\dagger} = N_0 [u_0 u_0^{\dagger}]^{-1}. \tag{28}$$

We remark that the development so far is analogous to that of the baseband decision feedback equalizer.[5] Further progress toward achieving a closed-form expression for tr e_0 thus depends on obtaining a closed-form expression for the matrix $[u_0 u_0^{\dagger}]^{-1}$ or for its trace, corresponding to the result recently developed for the baseband case.[7] It has not been possible to do this directly for the QAM case when the most general transmitter matrix is allowed. We shall prove under quite general conditions that the minimum of tr w_0 is achieved with a transmitter of passband structure, and that, given this transmitter structure,

$$\operatorname{tr} w_0 = \tfrac{1}{2}\sqrt{\det w_0}$$

and

$$\det w_0 = \exp\left\{-\frac{T}{2\pi}\int_{-\pi/T}^{\pi/T}\log\det\left[\Phi(e^{-j\omega T}) + \frac{N_0}{\sigma_a^2}I\right]d\omega\right\},$$

where "det" denotes the determinant.

IV. CLOSED-FORM EXPRESSION FOR DET e_0

For the most general matrix filter, attainment of a closed-form MSE expression for tr e_0 in terms of the matrix $\Phi(e^{-j\omega T})$ has so far proved intractable. However, we shall see that such a general expression is unnecessary to describe the behavior of optimum systems. Our approach is to employ the following easily proven lower bound for 2 × 2 positive semi-definite symmetric matrices

$$\operatorname{tr} w_0 \geqq 2|\det^{\frac{1}{2}} w_0|, \tag{29a}$$

which holds with equality if and only if w_0 is a scalar matrix (i.e., multiple of the identity). In this section we develop a closed-form expression for det w_0. In the following sections where we deal with optimum systems, we can always perform the analysis in a context where eq. (29) holds the equality.

We begin the analysis of $\sqrt{\det w_0}$ by recalling (25a), from which follows

$$\det\left[\Phi(e^{-j\omega T}) + \frac{N_0}{\sigma_a^2}I\right] = \det U_-(e^{-j\omega T})\det U_-(e^{j\omega T}).$$

Then from the one-dimensional theory we have[15]

$$\det(u_0 u_0^\dagger) = \exp\left\{\frac{T}{2\pi}\int_{-\pi/T}^{\pi/T}\log\det\left[\Phi(e^{-j\omega T}) + \frac{N_0}{\sigma_a^2}I\right]d\omega\right\},$$

and from (28) and (29a)

$$\frac{1}{N_0}\operatorname{tr} e_0 = \operatorname{tr} w_0 \geqq 2\sqrt{\det w_0}, \tag{29b}$$

where

$$\det w_0 = \exp\left\{-\frac{T}{2\pi}\int_{-\pi/T}^{\pi/T}\log\det\left[\Phi(e^{-j\omega T}) + \frac{N_0}{\sigma_a^2}I\right]d\omega\right\}. \tag{29c}$$

V. FOR A NONEXCESS BANDWIDTH SYSTEM, PASSBAND TRANSMITTERS CANNOT BE OUTPERFORMED

In this section we begin by expressing Φ explicitly in terms of the transmitter and channel matrices. Then we define the notion of a nonexcess bandwidth system. The primary result of this section is that, for a nonexcess bandwidth system, if the transmitter power density function

$$f(\omega) = \operatorname{tr} G^{\dagger}G \qquad \left(f(\omega) \geqq 0, \frac{1}{2\pi}\int_{-\pi/T}^{\pi/T} f(\omega)d\omega = \frac{2TP}{\sigma_a^2}\right)$$

is specified, then there exists a passband transmitter in the class of all matrix transmitters optimal under the constraint that $f(\omega)$ is the power density function.

To display the dependence of our results so far on the transmitter frequency response $G(\omega)$, we first rewrite the matrix $\Phi(e^{-j\omega T})$ using the definition of ϕ_n as

$$\Phi(e^{-j\omega T}) \equiv \int_{-\infty}^{\infty} h^{\dagger}(-\tau)\mathcal{H}(\omega, \tau)d\tau, \tag{30a}$$

where

$$\mathcal{H}(\omega, \tau) \equiv \sum_n h(nT - \tau)e^{-j\omega nT}. \tag{30b}$$

Expression (30b) is a Fourier series. Thus,

$$h(nT - \tau) = \frac{T}{2\pi}\int_{-\pi/T}^{\pi/T} \mathcal{H}(\omega, \tau)e^{j\omega nT}d\omega. \tag{31}$$

But the matrix impulse response $h(nT - \tau)$ can also be written as the inverse Fourier transform of a matrix frequency response $H(\omega)$,

$$h(nT - \tau) = \frac{1}{2\pi}\int_{-\infty}^{\infty} H(\omega)e^{j\omega(nT-\tau)}d\omega,$$

which, upon splitting up the range of integration and changing the variable of integration, can be written

$$h(nT - \tau) = \frac{1}{2\pi}\int_{-\pi/T}^{\pi/T} e^{j\omega nT} \times \left[\sum_m H\left(\omega + \frac{2\pi m}{T}\right)\exp\left[-j\left(\omega + \frac{2\pi m}{T}\right)\tau\right]\right]d\omega. \tag{32}$$

Equating the integrands in (31) and (32), we obtain an explicit expression for $\mathcal{H}(\omega, \tau)$ which when substituted into (30a) yields

$$\Phi(e^{-j\omega T}) = \frac{1}{T}\sum_n H\left(\omega + \frac{2\pi n}{T}\right)^\dagger H\left(\omega + \frac{2\pi n}{T}\right). \tag{33}$$

Furthermore, denoting the Fourier transforms of $c(t)$ and $g(t)$ by the channel matrix $C(\omega)$ and the transmitter matrix $G(\omega)$ respectively, we can write $H(\omega) = C(\omega)G(\omega)$ and

$$\Phi(e^{-j\omega T}) = \frac{1}{T}\sum_n G\left(\omega + \frac{2\pi n}{T}\right)^\dagger \times C\left(\omega + \frac{2\pi n}{T}\right)^\dagger C\left(\omega + \frac{2\pi n}{T}\right) G\left(\omega + \frac{2\pi n}{T}\right). \tag{34}$$

A *nonexcess bandwidth system* is defined by the property that for any radian frequency ω there is no more than one nonzero term in the above sum. It can be taken to be the $n = 0$ term by making a trivial frequency translation where necessary. Hence for a nonexcess bandwidth system

$$\Phi(e^{-j\omega T}) = \frac{1}{T} G(\omega)^\dagger C(\omega)^\dagger C(\omega) G(\omega) \quad \left(|\omega| \leqq \frac{\pi}{T}\right). \tag{35}$$

In this section we deal exclusively with nonexcess bandwidth systems. In Section VIII we refer to a recent theorem of H. Witsenhausen which enables us to do a complete analysis of excess bandwidth systems by transforming them to a canonical nonexcess bandwidth "equivalent" and then transforming back.

To model the class of transmitter frequency responses $G(\omega)$, we introduce $\mathcal{G}$ to denote the (Hilbert) space of all 2×2 matrices whose entries are Hermitian symmetric $\{G(\omega) = [G(-\omega)]^*\}$ finite energy functions on $(-\pi/T, \pi/T)$. The Hermitian symmetry of the entries is required so that each entry represents the Fourier transform of a real-time function. As in Section I, we use $\mathcal{P}$ to denote the passband subspace of $\mathcal{G}$ consisting of matrices of the form

$$\begin{pmatrix} G_{11}(\omega) & G_{12}(\omega) \\ -G_{12}(\omega) & G_{11}(\omega) \end{pmatrix}.$$

We shall be dealing only with matrix filters G of finite power P, given by (6). Thus we use $\mathcal{G}_P$ and $\mathcal{P}_P$ to denote

$$\left\{ G \,\middle|\, \frac{1}{2\pi}\int_{-\pi/T}^{\pi/T} \operatorname{tr} G(\omega)G(\omega)^\dagger d\omega = \frac{2TP}{\sigma_a^2} \right\}$$

in $\mathcal{G}$ and $\mathcal{P}$, respectively. In the sequel, all transmitter filters will be assumed to have power P.

We now optimize $\det\,(G^\dagger C^\dagger CG + (N_0/\sigma_a^2)\,I)$ at each radian frequency ω for a fixed amount of power transmitted at ω.

Fix $N_0 > 0$, $f > 0$ and $C = \begin{pmatrix} C_1 & C_2 \\ -C_2 & C_1 \end{pmatrix}$ (C_i's complex functions of frequency). Explicitly, we shall show that

$$\max \det \left\{ G^\dagger C^\dagger CG + \frac{N_0}{\sigma_a^2}\, I \right\}$$

over all complex G such that $\operatorname{tr} G^\dagger G = f(\omega)$ is achieved for a G of the passband form $\begin{pmatrix} G_{11} & G_{12} \\ -G_{12} & G_{11} \end{pmatrix}$. (For linear QAM systems, the same determinant extremal problem arises in the optimum selection of a transmitter with a specified power spectral density function. To our knowledge, this aspect of linear QAM systems has escaped the literature.)

Notice the unitary transformation $\Psi = 2^{-\frac{1}{2}}\begin{pmatrix} 1 & j \\ j & 1 \end{pmatrix}$ diagonalizes matrices of the form $\begin{pmatrix} a & jb \\ -jb & a \end{pmatrix}$ in that

$$\Psi^\dagger \begin{pmatrix} a & jb \\ -jb & a \end{pmatrix} \Psi = \begin{pmatrix} a+b & 0 \\ 0 & a-b \end{pmatrix}. \tag{36}$$

Since $C^\dagger C$ is of the form $\begin{pmatrix} a & jb \\ -jb & a \end{pmatrix}$ (a, b, real, $a > b$), if we let $G = \Psi B$ the problem becomes

$$\max \det \left\{ B^\dagger \begin{pmatrix} a+b & 0 \\ 0 & a-b \end{pmatrix} B + N_0 I \right\}, \quad \operatorname{tr} B^\dagger B = f(\omega).$$

Let $D = \begin{pmatrix} a+b & 0 \\ 0 & a-b \end{pmatrix}$ and rewrite the problem as

$$\max \{\det(B^\dagger DB) + N_0 \operatorname{tr}(B^\dagger DB) + N_0^2\}, \quad \operatorname{tr} B^\dagger B = f(\omega).$$

At this stage we denote the Hermitian matrix $BB^\dagger$ by Q and write

$$\max \{\det QD + N_0 \operatorname{tr} QD + N_0^2\}, \quad \operatorname{tr} Q = f(\omega).$$

Of course, an optimum Q exists, since we are maximizing a continuous function over a compact set. A nonzero off-diagonal entry in Q would only affect the determinant and not the traces. Since Q is Hermitian, the optimal Q is diagonal. Retracking, $Q = BB^\dagger = \Psi^\dagger G(\Psi\Psi^\dagger)G^\dagger\Psi$. Now Q is positive definite and so has a positive definite square root $Q^{\frac{1}{2}}$. From the definition of B,

$$G = \Psi Q^{\frac{1}{2}} \Psi^{\dagger}$$

which shows that the optimal G has the form

$$G = \begin{pmatrix} G_{11} & G_{12} \\ -G_{12} & G_{11} \end{pmatrix}. \tag{37}$$

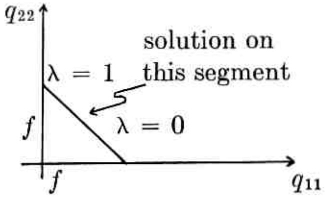
Although the proof is now complete, we go further and find G_{11} and G_{12}, as this will be used in the sequel. To find G_{11} and G_{12} we have seen that we must first find the entries q_{11} and q_{22} of Q so as to maximize

$$\{q_{11}q_{22}(a+b)(a-b) + N_0[q_{11}(a+b) + q_{22}(a-b)] + N_0^2\}$$

on the triangle in the (q_{11}, q_{22}) plane described by

$$q_{11} + q_{22} \leqq f, \quad q_{11} \geqq 0, q_{22} \geqq 0.$$

Since $a > 0$ and $a \geqq b$, the optimum (q_{11}, q_{22}) is achieved with $q_{11} + q_{22} = f$. Let λ linearly parametrize the segment joining $(f, 0)$ and $(0, f)$ as shown below



so $(q_{11}, q_{22}) = [(1-\lambda)f, \lambda f]$ where $(0 \leqq \lambda \leqq 1)$.

The criterion becomes: Maximize

$$f\{\lambda(1-\lambda)f(a^2-b^2) + N_0[(1-\lambda)(a+b) + \lambda(a-b)]\} + N_0^2, \tag{38}$$

which is a parabola concave in λ. Our problem is to determine $\lambda_{\text{opt}}(0 \leqq \lambda_{\text{opt}} \leqq 1)$. Now the parabola is maximized at

$$\bar{\lambda} = \frac{1}{2} - \frac{N_0}{f}\left(\frac{b}{a^2-b^2}\right).$$

If $\bar{\lambda}$ satisfies $0 \leq \bar{\lambda} < 1$, then $\lambda_{\text{opt}} = \bar{\lambda}$. If $\bar{\lambda} < 0$, $\lambda_{\text{opt}} = 0$ and if $\bar{\lambda} > 1$, $\lambda_{\text{opt}} = 1$.

So from $G = \Psi Q^{\frac{1}{2}} \Psi^{\dagger}$, we obtain

$$G_{11} = \frac{[|\sqrt{(1-\lambda_{\text{opt}})}| + |\sqrt{\lambda_{\text{opt}}}|]|f^{\frac{1}{2}}|}{|2^{\frac{1}{2}}|} \tag{39a}$$

$$G_{12} = \mathrm{j}\frac{(|\sqrt{1-\lambda_{\text{opt}}}| - |\sqrt{\lambda_{\text{opt}}}|)|f^{\frac{1}{2}}|}{|2^{\frac{1}{2}}|} \text{ signum } b. \tag{39b}$$

The determination of the signs attached to G_{11} and G_{12} was made by noticing that at each frequency

$$\det\left(G^{\dagger}C^{\dagger}CG + \frac{N_0}{\sigma^2}I\right) \tag{40}$$

is invariant to the sign of G_{11}, while (40) is maximized if signum b is used for G_{12}.

VI. CLOSED FORM EXPRESSION FOR ALL PASSBAND G

We have seen that, for nonexcess bandwidth systems, an extremal G for

$$\det w_0 = \exp\left\{-\frac{T}{2\pi}\int_{-\pi/T}^{\pi/T}\log\det\left[\Phi(e^{-j\omega T}) + \frac{N_0}{\sigma_a^2}I\right]d\omega\right\}$$

exists in the space $\mathcal{P}$. Next we show that for each $G \in \mathcal{P}$, whether or not it has excess bandwidth, $\operatorname{tr} w_0 = 2\sqrt{\det w_0}$. To do this we must show that w_0 is a scalar matrix. First observe that the matrices G and C are in $\mathcal{P}$, and their entries, being Fourier transforms of real-time functions, are Hermitian symmetric. The matrix $\Phi(e^{-j\omega T}) + (N_0/\sigma_a^2)I$, which is designated by $\mathcal{R}$ and is the Fourier transform of the matrix sequence $\mathscr{r}$ in (24), can be expressed in terms of the channel matrix $C(\omega)$ and a passband transmitter matrix

$$G(\omega) = \begin{pmatrix} G_1(\omega) & G_2(\omega) \\ -G_2(\omega) & G_1(\omega)\end{pmatrix}$$

as in (34) to yield

$$\mathcal{R} = \begin{pmatrix} \mathcal{R}_1(\omega) & j\mathcal{R}_2(\omega) \\ -j\mathcal{R}_2(\omega) & \mathcal{R}_1(\omega)\end{pmatrix},$$

where

$$\begin{aligned}\mathcal{R}_1(\omega) = \frac{1}{T}\sum_n\Bigg[\bigg|&G_1\left(\omega+\frac{2\pi n}{T}\right)C_1\left(\omega+\frac{2\pi n}{T}\right) \\ &- G_2\left(\omega+\frac{2\pi n}{T}\right)C_2\left(\omega+\frac{2\pi n}{T}\right)\bigg|^2 \\ &+\bigg|G_1\left(\omega+\frac{2\pi n}{T}\right)C_2\left(\omega+\frac{2\pi n}{T}\right) \\ &+ G_2\left(\omega+\frac{2\pi n}{T}\right)C_1\left(\omega+\frac{2\pi n}{T}\right)\bigg|^2 + \frac{N_0}{\sigma_a^2}\Bigg]\end{aligned} \tag{41}$$

and

$$\begin{aligned}\mathcal{R}_2(\omega) = -\frac{2}{T}\sum_n \mathrm{Im}\Bigg\{\bigg[&G_2\left(\omega+\frac{2\pi n}{T}\right)C_1\left(\omega+\frac{2\pi n}{T}\right)\\ &+G_1\left(\omega+\frac{2\pi n}{T}\right)C_2\left(\omega+\frac{2\pi n}{T}\right)\bigg]^*\\ &\times\bigg[G_1\left(\omega+\frac{2\pi n}{T}\right)C_1\left(\omega+\frac{2\pi n}{T}\right)\\ &-G_2\left(\omega+\frac{2\pi n}{T}\right)C_2\left(\omega+\frac{2\pi n}{T}\right)\bigg]\Bigg\}. \qquad (42)\end{aligned}$$

The entries $\mathcal{R}_1$ and $\mathcal{R}_2$ are real functions of ω. $\mathcal{R}_1$ is a positive even function and $\mathcal{R}_2$ is an odd function. The matrix $\mathcal{R}$ is positive definite; i.e., $\mathcal{R}_1^2 > \mathcal{R}_2^2$. It is also Hermitian and passband.

We have previously noted in eq. (25) that the matrix $\mathcal{R}$ can be factored into the anticausal and causal matrices $U_-(e^{-j\omega T})$ and $[U_-(e^{-j\omega T})]^\dagger$, respectively. The matrix $u_0u_0^\dagger$, which is proportional to the error matrix inverse, is unique and the factor $U_-(e^{-j\omega T})$ is unique up to an arbitrary unitary matrix post-multiplicative factor Ψ. We now pick a particular unitary matrix.

The matrix $\mathcal{R}$ is diagonalized by the unitary matrix

$$\Psi = \frac{1}{\sqrt{2}}\begin{pmatrix}1 & j\\ j & 1\end{pmatrix};\quad \text{i.e., } \Psi^\dagger\mathcal{R}\Psi = \begin{pmatrix}\mathcal{R}_1-\mathcal{R}_2 & 0\\ 0 & \mathcal{R}_1+\mathcal{R}_2\end{pmatrix}. \qquad (43)$$

Now the entries $\mathcal{R}_1 - \mathcal{R}_2$ and $\mathcal{R}_1 + \mathcal{R}_2$ are nonnegative real functions on $-(\pi/T) \leqq \omega \leqq (\pi/T)$. Since

$$-\infty < \int_{-\pi/T}^{\pi/T} \log\,(\mathcal{R}_1 \pm \mathcal{R}_2)d\omega,$$

we have from Szegö's theorem[15] that

$$\mathcal{R}_1 - \mathcal{R}_2 = |\alpha_-|^2 \qquad (44a)$$

and

$$\mathcal{R}_1 + \mathcal{R}_2 = |\beta_-|^2, \qquad (44b)$$

where α_- and β_- are anticausal functions of ω, i.e.,

$$\alpha_- = \sum_{m\leqq 0} \alpha_m e^{-j\omega mT} \qquad (45a)$$

and

$$\beta_- = \sum_{m\leqq 0} \beta_m e^{-j\omega mT}; \qquad (45b)$$

the $\{a_m\}$ and $\{\beta_m\}$ being sequences of complex numbers. We can assume

α_0 and β_0 are real and positive without loss of generality. Therefore

$$\Psi^\dagger \mathcal{R} \Psi = VV^\dagger, \tag{46}$$

where

$$V = \begin{pmatrix} \alpha_- & 0 \\ 0 & \beta_- \end{pmatrix},$$

and since Ψ is a unitary matrix,

$$\begin{aligned} \mathcal{R} &= (\Psi V \Psi^\dagger)(\Psi V^\dagger \Psi^\dagger) \\ &= U_- U_-^\dagger, \end{aligned} \tag{47a}$$

where

$$U_- = \Psi V \Psi^\dagger = \begin{pmatrix} \alpha_- + \beta_- & j(\alpha_- - \beta_-) \\ -j(\alpha_- - \beta_-) & \alpha_- + \beta_- \end{pmatrix}. \tag{47b}$$

Thus in this factorization, U_- can be taken to be passband. Furthermore, since α_0 and β_0 are real,

$$u_0 = \begin{pmatrix} \alpha_0 + \beta_0 & j(\alpha_0 - \beta_0) \\ -j(\alpha_0 - \beta_0) & \alpha_0 + \beta_0 \end{pmatrix} \tag{48}$$

is both Hermitian and passband, and so therefore is the error matrix $e_0 = [u_0 u_0^\dagger]^{-1}$; i.e., its off-diagonal terms are purely imaginary.* But we know that e_0, defined by (15), must have real equal off-diagonal terms, and therefore e_0 must be a scalar matrix. Thus $w_0 = (1/N_0)e_0$ is also scalar and

$$\text{tr } w_0 = 2\sqrt{\det w_0}. \tag{49}$$

Summarizing the development so far, we have shown that, for nonexcess bandwidth systems, if the transmitted power spectrum is specified, the passband transmitter structure is optimum. We then showed that if the transmitter has the passband structure, the MSE is given by eqs. (29b) and (29c), (29b) holding with equality. Incidentally, using the results of the last paragraph it can be shown that $\alpha_0 = \beta_0 = \sqrt{N_0/2\text{MMSE}}$; hence u_0 is known. In Section III the optimum linear receiver filter $w(t)$ was found up to the constant (matrix) multiplier u_0^{-1}. For nonexcess bandwidth passband systems, we can now make the more complete statement that the matrix Fourier transform of $w(t)$ is

$$\sqrt{\frac{\text{MMSE}}{2N_0}}\, G^\dagger(\omega) C^\dagger(\omega) \Big/ \sqrt{G^\dagger(\omega) C^\dagger(\omega) C(\omega) G(\omega) + \frac{N_0 I}{\sigma_a^2}}^{\,\wedge\,\phi}$$

(where $\sqrt{\quad}^{\wedge\,\phi}$ means minimum-phase square root).

* It is important to notice that, although u_0 is unique only up to a postmultiplicative unitary factor, the matrix $u_0 u_0^\dagger$ is unique.

We shall now express (29c) in a somewhat different form, which avoids the use of the determinant. As pointed out earlier, $\mathcal{R}_1(\omega)$ and $\mathcal{R}_2(\omega)$ are real even and odd functions, respectively. Thus

$$\mathcal{R}_1(\omega) - \mathcal{R}_2(\omega) = \mathcal{R}_1(-\omega) + \mathcal{R}_2(-\omega)$$

and

$$\int_{-\pi/T}^{\pi/T} \log\,[\mathcal{R}_1(\omega) - \mathcal{R}_2(\omega)]d\omega = \int_{-\pi/T}^{\pi/T} \log\,[\mathcal{R}_1(\omega) + \mathcal{R}_2(\omega)]d\omega, \quad (50)$$

from which it follows that

$$\begin{aligned}\operatorname{tr} e_0 &= 2N_0 \exp\left[-\frac{T}{4\pi}\int_{-\pi/T}^{\pi/T} \log \det \mathcal{R}(\omega) d\omega\right] \\ &= 2N_0 \exp\Bigg(-\frac{T}{4\pi}\left\{\int_{-\pi/T}^{\pi/T} \log\,[\mathcal{R}_1(\omega) - \mathcal{R}_2(\omega)]d\omega\right. \\ &\qquad\qquad \left. + \int_{-\pi/T}^{\pi/T} \log\,[\mathcal{R}_1(\omega) + \mathcal{R}_2(\omega)]d\omega\right\}\Bigg) \\ &= 2N_0 \exp\left\{-\frac{T}{2\pi}\int_{-\pi/T}^{\pi/T} \log\,[\mathcal{R}_1(\omega) + \mathcal{R}_2(\omega)]d\omega\right\}. \qquad (51)\end{aligned}$$

Substituting (41) and (42) into (51) gives the following expression for MSE:

$$\operatorname{tr} e_0 = 2\sigma_a^2 \exp\left\{-\frac{T}{2\pi}\int_{-\pi/T}^{\pi/T} \log\left[\frac{\sigma_a^2}{N_0} X_{eq}(\omega) + 1\right]d\omega\right\}, \quad (52)$$

where

$$\begin{aligned}X_{eq}(\omega) = \frac{1}{T}\sum_n \left| G_1\left(\omega + \frac{2\pi n}{T}\right) + jG_2\left(\omega + \frac{2\pi n}{T}\right)\right|^2 \\ \times\left|C_1\left(\omega + \frac{2\pi n}{T}\right) + jC_2\left(\omega + \frac{2\pi n}{T}\right)\right|^2.\end{aligned}$$

This expression is valid for any passband transmitter and, as shown in the previous section, it is valid for optimum general QAM transmitters with no excess bandwidth.[*] We show in Appendix A that, under very general assumptions, optimum *passband* transmitters will have no excess bandwidth.

[*] We remark at this point that if we had restricted attention to passband transmitter structures from the outset, we could have derived the MSE expression (52) more directly by using the complex envelope notation referred to in Section II instead of the matrix formulation.

VII. OPTIMUM TRANSMITTER

Here we continue under the assumption of a nonexcess bandwidth system. So far, we know that, if $f = \operatorname{tr} G^{\dagger}G$ is specified, an optimal passband G exists yielding a minimum for MSE and possessing power spectral density f. Our next step is to free $\operatorname{tr} G^{\dagger}G$ and to find G, which minimizes MSE subject only to the constraint that

$$\frac{T}{2\pi}\int_{-\pi/T}^{\pi/T} f = \left(\frac{T}{2\pi}\int_{-\pi/T}^{\pi/T} \operatorname{tr} G^{\dagger}G = \right) = P. \tag{53}$$

Notice

$$\int_{-\pi/T}^{\pi/T} |G_1 + jG_2|^2 = \int_{-\pi/T}^{\pi/T} (|G_1|^2 + |G_2|^2) \tag{54}$$

since $G_1^*G_2 - G_1G_2^*$ is odd. Thus our problem becomes to find $|G_1 + jG_2|^2$, minimizing

$$2 \exp - \left\{\frac{T}{2\pi}\int_{-\pi/T}^{\pi/T} \log\left(\sigma_a^2 \frac{|G_1 + jG_2|^2|C_1 + jC_2|^2}{TN_0} + 1\right)\right.$$

subject to $\frac{T}{2\pi}\int_{-\pi/T}^{\pi/T} |G_1 + jG_2|^2 = P.$

It is shown in Appendix B that the solution to this problem is given uniquely by

$$|G_1 + jG_2|^2 = \left(\mathcal{C} - \frac{N_0T^2}{\sigma_a^2}|C_1 + jC_2|^{-2}\right)_+$$

$$(\text{where } (\xi)_+ \triangleq \max[(\xi, 0)]),$$

where $\mathcal{C}$ is a constant set at a value so that

$$\frac{T}{2\pi}\int_{-\pi/T}^{\pi/T} (|G_1|^2 + |G_2|^2) = P.$$

This solution also occurs in a related context in information theory, where it is dubbed "the water-pouring solution."[16]

Since $(G_1^*(\omega)G_2(\omega) - G_1(\omega)G_2^*(\omega))$ is odd and $f(\omega)$ is even, we average $|G_1^*(\omega) + jG_2(\omega)|^2$ and $|G_1^*(-\omega) + jG_2(-\omega)|^2$ to get*

$$f(\omega) = \frac{1}{2}\left\{\left[\mathcal{C} - \frac{N_0T^2}{\sigma_a^2}|C_1(\omega) + jC_2(\omega)|^{-2}\right]_+ + \left[\mathcal{C} - \frac{N_0T^2}{\sigma_a^2}|C_1(-\omega) + jC_2(-\omega)|^{-2}\right]_+\right\}.$$

To find $G_1(\omega)$ and $G_2(\omega)$, use the above $f(\omega)$ in Section V.

* Note that for $N_0 \to 0$, the optimum $f(\omega)$ tends to a constant.

VIII. THE ROLE OF NONEXCESS BANDWIDTH SYSTEMS

In the previous sections we have determined the optimum transmitter under the hypothesis that the system is nonexcess bandwidth. Here we point out that this hypothesis is not very restrictive.

In systems in which the transmitter is required to be passband, it follows, under very mild assumptions on the channel characteristics, that the optimum transmitter (subject to an output power constraint) is a nonexcess bandwidth system. The mathematical proof of the optimality of the nonexcess bandwidth system is considered in detail in Appendix A. For an example, if for each $\omega(|\omega| < \pi/T)$

$$|C_1(\omega) + jC_2(\omega)| > \left|C_1\left(\omega + \frac{2\pi k}{T}\right) + jC_2\left(\omega + \frac{2\pi k}{T}\right)\right| \qquad (k \neq 0),$$

then the optimal transmitter has no energy outside

$$\left\{\omega \middle| |\omega - 2\pi f_0| \leqq \frac{\pi}{T}\right\}.$$

For systems allowing any matrix transmitter, the question arises whether or not the optimal transmitter is passband. If the answer to the question is negative, the next question is whether or not the optimal transmitter is nonexcess bandwidth. The answers to these questions depend on the system parameters, and there are channels for which the answers to both questions are negative. It is beyond the scope of this paper to give a detailed mathematical discussion of these more complex systems. Such systems are still under investigation, and so we shall limit ourselves to mentioning without proof some important facts concerning the analysis of such systems.

The analysis begins by returning to Section IV fixing ω and posing the extremal problem of

$$\max \det \left(\sum G_k^\dagger C_k^\dagger C_k G_k + N_0 I\right),$$

subject to $\operatorname{tr} \sum G_k^\dagger G_k = f$. If for each ω it is optimal to expend all of f on one of the G_k's, then we are in the line pursued in the previous sections. However, to achieve optimality one may need to use more than one G. Indeed, H. Witsenhausen has solved this determinant extremal problem showing that at most two G_k's are required to achieve optimality, and there are instances where two G_k's are necessary. Even when two G_k's are needed, the w_0 matrix remains a scalar matrix and once again the trace and the determinant optimization are equivalent. The fact that two G_k's are required means the transmitter is excess bandwidth.

Witsenhausen has shown that when two G_k's are needed, one can be taken to be a multiple of $\begin{pmatrix} 1 & j \\ -j & 1 \end{pmatrix}$ and the other a multiple of $\begin{pmatrix} 1 & j \\ j & -1 \end{pmatrix}$. Although both G_k's cannot have the passband form, the $\begin{pmatrix} 1 & j \\ j & -1 \end{pmatrix}$ matrix corresponds to a very simple structural variation of a passband filter.

We mention in closing that systems whose optimization takes us outside the realm of passband structures can be analyzed via equivalent canonical nonexcess bandwidth passband systems. The equivalence is in the sense that MMSE versus P curves for the two systems are identical, and optimum design can be carried out in the canonical system and then transformed to the more complicated system.

IX. ACKNOWLEDGMENTS

We gratefully acknowledge very helpful discussions with R. D. Gitlin, S. Halfin, J. Salz, and H. S. Witsenhausen.

APPENDIX A

Optimality of a Nonexcess Bandwidth System

Fix $P > 0$ and $q(\omega)$ a positive continuous real function on $(-\infty, +\infty)$. In the text we are confronted with the optimization

$$\sup \int_{-\pi/T}^{\pi/T} \log \left[\sum_{k=-\infty}^{\infty} r\left(\omega + \frac{2\pi k}{T}\right) q\left(\omega + \frac{2\pi k}{T}\right) + 1 \right] d\omega,$$

where the sup is over all nonnegative Lebesque integrable $r(\omega)$ for which

$$\int_{-\infty}^{+\infty} r(\omega) d\omega \leqq P > 0.$$

We show here that, under weak conditions on $q(\omega)$, the optimization problem can be replaced by an equivalent "nonexcess bandwidth problem," namely, find

$$\sup \int_{-\pi/T}^{+\pi/T} \log \left[\tilde{r}(\omega)\tilde{q}(\omega) + 1\right] d\omega,$$

where $\tilde{q}(\omega)$ is a given continuous function and $\tilde{r}(\omega)$ is any nonnegative integrable function satisfying

$$\int_{-\pi/T}^{\pi/T} \tilde{r}(\omega) d\omega = P > 0.$$

Define $\tilde{q}(\omega)$ on $[-\pi/T, \pi/T]$ to be the envelope sup $q(\omega + 2\pi k/T)$. To avoid annoying pathologies, assume $q(\omega)$ is such that for each ω

$$\left\{k \mid q\left(\omega + \frac{2\pi k}{T}\right) = \tilde{q}(\omega)\right\}$$

is not empty. Moreover, assume that $(-\pi/T, \pi/T)$ can be expressed as a disjoint union of subsets $\{V_m\}_1^m$ of total measure $2\pi/T$ such that on each V_m there exists a k_m so

$$q\left(\omega + \frac{2\pi k_m}{T}\right) = \tilde{q}(\omega)$$

holds uniformly in ω on V_m. So $\tilde{q}(\omega)$ is continuous on $(-\pi/T, \pi/T)$. Define

$$V = \bigcup_1^m \left(V_m + \frac{2\pi k_m}{T}\right).$$

Given any $r(\omega) \geqq 0$ satisfying $\|r\|_1 = P$, define ρ on $(-\infty, \infty)$ by

$$\rho\left(\omega + \frac{2\pi k_m}{T}\right) = \begin{cases} \sum_{-\infty}^{\infty} r\left(\omega + \frac{2\pi k}{T}\right) & \text{for } \omega \in V_m \quad m = 1, 2, \cdots, M \\ 0 & \omega \notin V. \end{cases}$$

So

$$\int_{-\infty}^{\infty} \rho d\omega = \sum_{-\infty}^{\infty} \int_{-\pi/T}^{\pi/T} \rho\left(\omega + \frac{2\pi k}{T}\right) d\omega$$
$$= \int_{-\pi/T}^{\pi/T} \sum_h r\left(\omega + \frac{2\pi k}{T}\right) = \sum_k \int_{-\pi/T}^{\pi/T} r\left(\omega + \frac{2\pi k}{T}\right) d\omega = P,$$

where the second equality results from the definition of ρ and the third equality is from the Lebesque Dominated Convergence Theorem. Now for $|\omega| < \pi/T$

$$\sum_k r\left(\omega + \frac{2\pi k}{T}\right) q\left(\omega + \frac{2\pi k}{T}\right) \leqq \sum_k r\left(\omega + \frac{2\pi k}{T}\right) \tilde{q}(\omega)$$
$$= \tilde{q}(\omega) \sum_k \rho\left(\omega + \frac{2\pi k}{T}\right) = \sum_k \rho\left(\omega + \frac{2\pi k}{T}\right) q\left(\omega + \frac{2\pi k}{T}\right),$$

where the very last equality follows from the fact that ρ vanishes off V. Since in $L[-\pi/T, \pi/T]$ $\rho(\omega)$ always fares at least as well as $r(\omega)$, we have the fact that the supremum can be taken over the class of nonnegative functions vanishing off V.

In the applications it often occurs that $V \subseteq (-\pi/T, \pi/T)$, in which case $q(\omega) = \tilde{q}(\omega)$ on V and the optimization problem becomes

$$\sup_{\|r\|=P} \int_{-\pi/T}^{\pi/T} \log\,[r(\omega)q(\omega) + 1]d\omega.$$

Even when $V \not\subset (-\pi/T, \pi/T)$, we need only solve

$$\sup_{\|\tilde{r}\|=P} \int_{-\pi/T}^{\pi/T} \log\,[\tilde{r}(\omega)\tilde{q}(\omega) + 1]d\omega$$

with the optimand "rearranged" to produce the desired $r(\omega)$. The rearrangement procedure is simply that, for each $\omega \in (-\pi/T, \pi/T)$, we define $r(\omega + 2\pi k_m/T) = \tilde{r}(\omega)$. Elsewhere $r(\omega)$ is defined to be zero. In dealing with even $q(\omega)$, if $V \not\subset (-\pi/T, \pi/T)$, the rearrangement produces an uneven $r(\omega)$. When this occurs, $[r(\omega) + r(-\omega)]/2$ provides an even optimand.

APPENDIX B

Maximization of the Exponent Functional

Let $q(\omega)$ be a continuous positive function on an interval $[a, b]$. Fixing a real number $P > 0$, let Γ be the convex set of nonnegative continuous functions with integral less than or equal to P. We seek γ to maximize the nonlinear functional

$$I(\gamma) \triangleq \int_a^b \log\,(1 + \gamma q).$$

This same problem occurs in classical information theory where, for reasons we shall see, it is dubbed "the water-pouring problem." Although the solution is correctly described in the literature, the supporting arguments are formal (for example, see Ref. 16 or 17). We give a rigorous proof here, although our argument is not constructive in that the extremal function is "pulled out of the air." To motivate the extremal function, the reader can turn to the references or supply for himself a variational derivation.

Now $I(\gamma)$ is concave on Γ, as we see by employing the Liebnitz rule to confirm the strict negativity of $I''[\lambda\gamma_1 + (1 - \lambda)\gamma_2]$ on $0 \leqq \lambda \leqq 1$ with γ_1 and γ_2 in Γ (differentiation is with respect to λ). It is clear that if the extremal function exists it has integral equal to P and so we can redefine Γ to require equality of the integrals.

For each constant $\mathcal{C}$, the function $(\mathcal{C} - q^{-1})_+$ denotes the function equal to $\mathcal{C} - q^{-1}$ when $\mathcal{C} - q^{-1} > 0$ and equal to zero otherwise. Now $\int (\mathcal{C} - q^{-1})$ is a continuous strictly increasing function of $\mathcal{C}$ with range

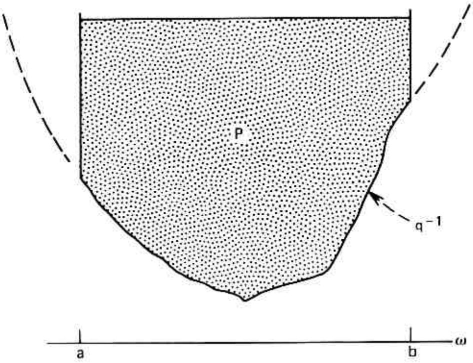


Fig. 7—Optimal power spectral density.

$[0, \infty]$. Fix $\mathcal{C}$ so $\int(\mathcal{C} - q^{-1})_+ = P$ and call the resulting function $\bar{\gamma}$. To show $I(\bar{\gamma})$ is the global maximum of $I(\gamma)$ over Γ, let γ_1 denote any other function in Γ and let us investigate the segment $\{\lambda\bar{\gamma} + (1 - \lambda)\gamma_1, 0 \leqq \lambda \leqq 1\}$. Now $I[\lambda\bar{\gamma} + (1 - \lambda)\gamma_1]$ is concave in λ and straightforwardly

$$I'[\lambda\bar{\gamma} + (1 - \lambda)\gamma_1]|_{\lambda=1} = \mathcal{C}^{-1}\left\{P - \int_{\bar{\gamma}=0} \mathcal{C}q\gamma_1 - \int_{\bar{\gamma}>0} \gamma_1\right\}$$

which is nonnegative as $\mathcal{C}q \leqq 1$. By definition, for a concave function the graph lies above any chord joining two points on the graph. So $\lambda = 1$ must be a point of global maxima of the segment.

Also, $\bar{\gamma}$ is the unique point of maxima since, if there were another point of maxima $\bar{\bar{\gamma}}$, we would have $I(\gamma)$ constant on the line segment joining $\bar{\gamma}$ and $\bar{\bar{\gamma}}$ contradicting the strict negativity of I''.

To understand the water-pouring terminology, look at Fig. 7 where we consider the graph of q^{-1} with vertical walls based at $[a, q^{-1}(a)]$ and $[b, q^{-1}(b)]$ to be a vessel into which water of amount (area) P is poured. Relocate the ω axis to the water level line. Then reflecting the water accumulation about the level line gives the shape of $\bar{\gamma}$.

We mention in closing that $\bar{\gamma}$ is optimal in a larger set than Γ obtained by requiring integrability rather than continuity in the definition of the constraint set. The optimality over the larger set follows from a function space continuity argument.

REFERENCES

1. MacColl, L. A., "Signaling Method and Apparatus," U. S. Patent No. 2,056,284, October 6, 1936.

2. Austin, M. E., "Decision—Feedback Equalization for Digital Communication over Dispersive Channels," Technical Report 461, Research Laboratory of Electronics, M.I.T., August 11, 1967.
3. Keeler, R. J., "Construction and Evaluation of a Decision Feedback Equalizer," Record of IEEE Int. Conf. on Comm. 1971, Montreal, Canada, June 1971, pp. 21–8 to 21–18.
4. George, D. A., Bowen, R. R., and Storey, J. R., "An Adaptive Decision Feedback Equalizer," IEEE Trans. Comm. Tech., *COM-19*, June 1971, pp. 281–293.
5. Monsen, P., "Feedback Equalization for Fading Dispersive Channels," IEEE Trans. on Info. Theory, *IT-17*, January 1971, pp. 56–64.
6. Price, R., "Nonlinearly Feedback-Equalized PAM vs. Capacity for Noisy Filter Channels," Record of IEEE Int. Conf. on Comm. 1972, Philadelphia, Pa., June 1972, pp. 22–12 to 22–17.
7. Salz, J., "Optimum Mean-Square Decision Feedback Equalization," B.S.T.J., *52*, No. 8 (October 1973), pp. 1341–1373.
8. Lucky, R. W., Salz, J., and Weldon, E. S., Jr., *Principles of Data Communication*, New York: McGraw-Hill, 1968.
9. Kobayashi, H., "Simultaneous Adaptive Estimation and Decision Algorithm for Carrier-Modultated Data Transmission Systems," IEEE Trans. Comm. Tech., *COM-19*, October 1971, pp. 835–848.
10. Dugundji, J., "Envelopes and Pre-Envelopes of Real Waveforms," IRE Trans. Info. Theory, *IT-4*, March 1958, pp. 53–57.
11. Wozencraft, J. M., and Jacobs, I. M., *Principles of Communication Engineering*, New York: John Wiley and Sons, 1965, pp. 492–504.
12. Vainberg, M. M., *Variational Methods for the Study of Nonlinear Operators*, San Francisco: Holden-Day, Chapter III.
13. Dunford, N., and Schwartz, J., *Linear Operators, Part III*, New York: Wiley-Interscience, 1971, Chapter XV.
14. Wiener, N., and Akutowicz, E. J., "A Factorization of Positive Hermitian Matrices," J. Math. and Mech., *8*, 1959, pp. 111–120.
15. Doob, J. L., *Stochastic Processes*, New York: John Wiley and Sons, 1967, pp. 160–164.
16. Fano, R. M., *Transmission of Information*, New York: John Wiley and Sons, 1961, pp. 168–178.
17. Walvick, E. A., "On the Capacity of an Ensemble of Channels with Differing Parameters," B.S.T.J., *49*, No. 3 (March 1970), pp. 415–429.

THE BELL SYSTEM TECHNICAL JOURNAL
Vol. 52, No. 10, December, 1973
Printed in U.S.A.

A Universal Digital Data Scrambler

By DAVID G. LEEPER

(Manuscript received May 22, 1973)

Analyses in the literature of digital communications often presuppose that the digital source is "white," that is, that it produces stochastically independent equiprobable symbols. In this paper we show that it is possible to "whiten" to any degree all the first- and second-order statistics of any binary source at the cost of an arbitrarily small controllable error rate. Specifically, we prove that the self-synchronizing digital data scrambler, already shown effective at scrambling strictly periodic data sources, will scramble any binary source to an arbitrarily small first- and second-order probability density imbalance δ *if (i) the source is first passed through the equivalent of a symmetric memoryless channel with an arbitrarily small but nonzero error probability* ϵ*, and (ii) the scrambler contains* M *stages where*

$$M \geqq 1 + \log_2[(\ln 2\delta)/\ln (1 - 2\epsilon)].$$

Some interpretations and applications of this result are included.

I. INTRODUCTION AND SUMMARY

Digital transmission systems often have impairments which vary with the statistics of the digital source. Timing, crosstalk, and equalization problems usually involve source statistics in some way. While redundant transmission codes may be used to help isolate system performance from source statistics, the isolation is not always complete, and such codes generate additional problems by increasing the required symbol rate or the number of levels per symbol which must be transmitted. In addition, with or without transmission codes, it is always easiest to analyze or predict system impairments if we assume that the source symbols are stochastically independent and equiprobable. We shall refer to such a source as "white" because of the obvious analogy to white Gaussian noise. Methods for "whitening" the statistics of digital sources without using redundant coding generally come under the heading of *scrambling.*

We describe here a nonredundant scrambling/descrambling method which in principle will satisfactorily whiten the statistics of *any* binary source. The technique is based upon the self-synchronizing digital data scrambler. Savage has shown[1] that this device is very effective at scrambling strictly periodic digital sources. In this paper it is proven that the same device will scramble any binary digital source to an arbitrarily small first- and second-order probability density imbalance δ if (*i*) the source is first passed through the equivalent of a binary symmetric memoryless channel with an arbitrarily small but nonzero error probability ϵ, and (*ii*) the scrambler contains M stages where $M \geqq 1 + \log_2 [(\ln 2\delta)/\ln (1 - 2\epsilon)]$. In other words, *at the cost of an arbitrarily small controllable error rate, one can "whiten" to any degree all the first- and second-order statistics of any binary source.* This relaxes the restriction frequently found in the literature in which the source is assumed *a priori* to produce only independent equiprobable symbols. An auxiliary result is that the above relation for M is useful when designing a standard self-synchronizing scrambler for a given application. Heuristically speaking, the relation expresses the "power" of the scrambler by linking the "randomness" of the input and output to the scrambler length, M.

In Sections II and III of this paper we examine some properties of scramblers, maximal length sequences, and mod-2 sums of binary random variables. With these discussions as background, we prove the main theorem in Section IV. In Section V we derive bounds for the autocorrelation of the scrambled sequence. Section VI contains some practical considerations involved in applying the theorem of Section IV. Beacuse they add insight, we give simple direct proofs for the lemmas and theorem of Sections III and IV.

II. SCRAMBLERS AND MAXIMAL LENGTH SEQUENCES

Figure 1 shows a five-stage self-synchronizing scrambler and descrambler.[1] As seen, both are linear sequential filters, the scrambler utilizing feedback paths and the descrambler feedforward paths. Each cell represents a unit delay. We restrict our attention to the binary case and use the symbols $\oplus$ and $\sum$ to denote mod-2 addition. Representing the data as shown, we have

$$b_k = a_k \oplus b_{k-3} \oplus b_{k-5}$$

and

$$c_k = b_k \oplus b_{k-3} \oplus b_{k-5} = a_k,$$

which shows that the descrambled sequence is identically equal to the

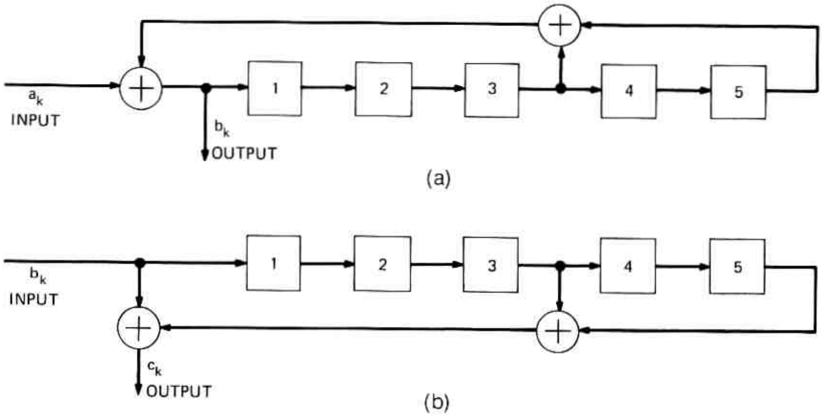


Fig. 1—(a) Five-stage scrambler. (b) Five-stage descrambler.

original data sequence. The descrambler is self-synchronizing because the effect of a channel error, insertion, or deletion lasts only as long as the total delay of the register, five bit-intervals in this example.

Let us consider the general scrambler of Fig. 2a with the input stream disconnected. Under such a condition, the scrambler becomes a sequence generator whose output must ultimately become periodic because (*i*) future states of the register are completely determined by the present state (the *state* of the register is the contents of its stages) and (*ii*) only the finite number 2^M states are possible, where M equals the number of stages. One of these, the all-zeros state, simply leads to an all-zeros output. Discounting this state, we see that the longest possible period from the generator must be $2^M - 1$ bits. It is proven in the literature[2,3] that with the proper choice of feedback taps we can generate such a *maximal length sequence* for any M.

Registers which generate maximal length sequences make very effective scramblers because of their ability to *dissociate* one scrambler output bit from another. This property will enable us to show that two arbitrarily chosen output bits tend to be very weakly correlated. We state this essential property here in the form of a lemma.

Lemma 1: From Fig. 2a it is evident that each "b" bit is equal to a lengthy mod-2 summation of selected "a" bits. Choose two bits, b_m and b_n, $m > n$, and define J_{mn} to be the number of "a" bits which enter the summation for b_m but not *the summation for b_n. That is, b_m is dissociated from b_n by the mod-2 sum of J_{mn} "a" bits.*

Then, if $n > 2^{M+1}$ (that is, the scrambler has processed at least 2^{M+1}

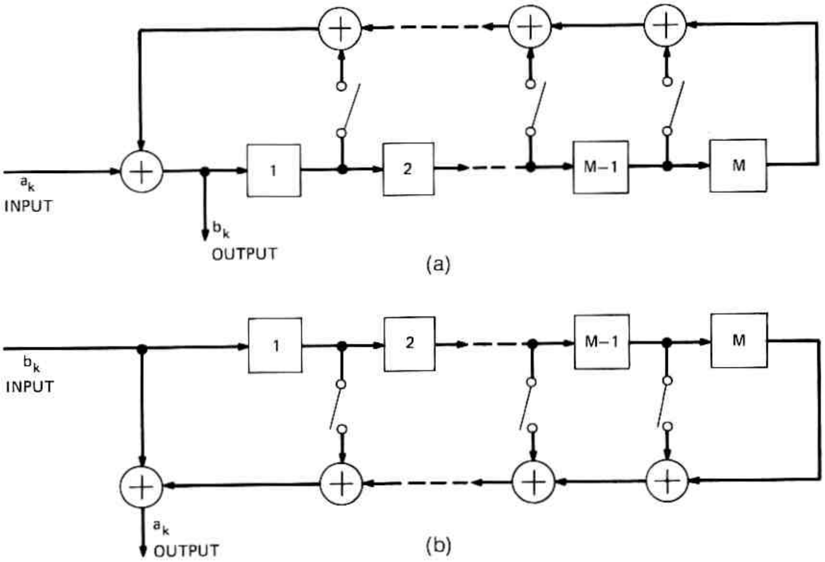


Fig. 2—(a) M-stage scrambler. (b) M-stage descrambler.

"a" bits),

$$J_o \equiv \min_{m,n} [J_{mn}] = 2^{M-1}. \tag{1}$$

In other words, after a settling time of 2^{M+1} bits, any chosen pair of output bits will differ by the mod-2 sum of at least 2^{M-1} input bits.
Proof: See appendix.

III. MOD-2 SUMS OF BINARY RANDOM VARIABLES

Throughout this paper we assume that a data sequence may be modeled as a sequence of binary random variables defined on a suitable probability space. In this section we state as lemmas two essential properties of mod-2 sums of binary random variables. Since the scrambler output is formed from mod-2 sums of input bits, these properties play a key role in determining the scrambler output characteristics. We include the proofs in the text because the equations involved will be useful later on.

Lemma 2: Consider two independent binary random variables r_1 and r_2. A third binary random variable $r_3 = r_1 \oplus r_2$. Let

$$p_i = P(r_i = 1) = 1 - P(r_i = 0), \qquad i = 1, 2, 3.$$

Then

$$|p_3 - \tfrac{1}{2}| \leqq \min [|p_2 - \tfrac{1}{2}|, |p_1 - \tfrac{1}{2}|]$$

with equality if and only if p_1 or $p_2 = \frac{1}{2}$, 0, or 1. In other words, r_3 is as close or closer to being equiprobable than either r_2 or r_1.

Proof: Since r_1 and r_2 are independent,

$$p_3 = p_2(1 - p_1) + p_1(1 - p_2). \tag{2}$$

Let

$$d_i = p_i - \tfrac{1}{2} \qquad i = 1, 2, 3.$$

Then by substitution

$$|d_3| = 2|d_1||d_2|.$$

But since

$$|d_i| \leqq \tfrac{1}{2},$$

we have

$$|d_3| \leqq \min [|d_1|, |d_2|]$$

with equality if and only if $|d_1|$ or $|d_2| = 0$ or $\frac{1}{2}$.

Corollary to Lemma 2: If $p_1 = \frac{1}{2}$, then $p_3 = \frac{1}{2}$ and r_3 is independent of r_2.

Proof: Since $r_3 = r_1 \oplus r_2$, $P(r_3 = 1|r_2 = 1) = 1 - p_1 = \frac{1}{2}$. But by eq. (2), $p_3 = \frac{1}{2}$. Thus, $P(r_3 = 1|r_2 = 1) = P(r_3 = 1) = \frac{1}{2}$, which implies r_3 and r_2 are independent.

Lemma 3: Consider now a sequence of independent binary random variables $\{r_k, k = 1, 2, \cdots\}$ with

$$P(r_k = 1) = 1 - P(r_k = 0) = \epsilon \quad \text{for all } k.$$

We form the mod-2 sum

$$R_n = \sum_{k=1}^{n} r_k \tag{3}$$

and let $P_n = P(R_n = 1)$. Then

$$P_n = \tfrac{1}{2}[1 - (1 - 2\epsilon)^n]; \qquad n \geqq 1. \tag{4}$$

Note that, as $n \rightarrow \infty$, P_n converges to $\frac{1}{2}$ for all $0 < \epsilon < 1$. However, we shall be concerned only with finite values for n.

Proof: By applying eq. (2) repeatedly, it is easily shown that the sequence P_n satisfies

$$P_n = (1 - 2\epsilon)P_{n-1} + \epsilon; \qquad n \geqq 2,$$

and

$$P_1 = \epsilon.$$

The solution to this first-order linear difference equation is given by eq. (4).

IV. A UNIVERSAL DIGITAL DATA SCRAMBLER

With the help of the lemmas, we may now derive the main result. We model the source as a device which generates a sequence of binary random variables $\{s_k\}$ with completely unknown statistics. Our goal is to find a scrambling/descrambling method such that the scrambled sequence $\{b_k\}$ will have statistics which approach those of the independent equiprobable ("white") sequence $\{w_k\}$. If we attempt to scramble $\{s_k\}$ directly as in Fig. 2, we are faced with a dilemma. The scrambler simply provides a one-to-one mapping between its input and output. As long as we have no knowledge or control of the statistics of $\{s_k\}$, the statistics of $\{b_k\}$ must likewise remain unknown and uncontrolled. Hence, the self-synchronizing scrambler *alone* cannot be universal.

Instead of scrambling directly, we proceed as shown in Fig. 3. The source output is first passed through the equivalent of a binary symmetric memoryless channel (BSC) with crossover probability $\epsilon > 0$. Remarkably, no matter how small ϵ may be, this modification of the source sequence is sufficient to guarantee that the first- and second-order probability densities for $\{b_k\}$ will approach those of $\{w_k\}$ to within an arbitrarily small difference δ. The only requirement is that M, the length of the scrambler, be dependent upon the choice of ϵ and δ. This is the essence of the theorem which we derive below. (We note in passing that the descrambled sequence will now differ from the original source sequence by the error rate ϵ, but since ϵ may be chosen arbitrarily small, we assume for now that this is of no consequence.)

To begin, we observe that because of the BSC the scrambler input sequence may be written

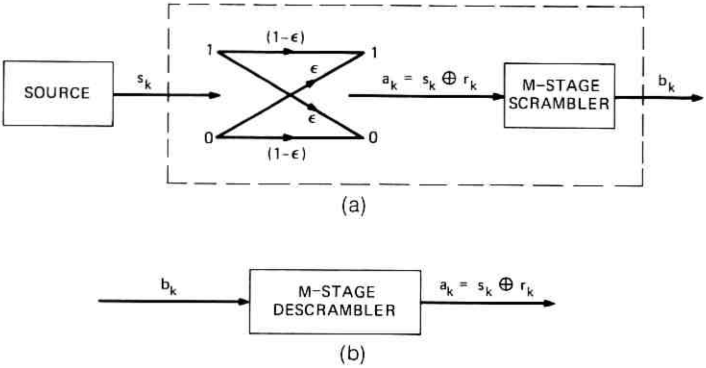
$$a_k = s_k \oplus r_k, \qquad k = 0, 1, 2, \cdots, \tag{5}$$

where

$$P(r_k = 1) = 1 - P(r_k = 0) = \epsilon.$$

From Lemma 1 we have seen that the action of the M-stage scrambler is to dissociate any chosen pair of bits (b_m, b_n) by the mod-2 sum of at least 2^{M-1} "a" bits. Let us assume that b_m and b_n are dissociated by *exactly* 2^{M-1} "a" bits and that they are related by

$$b_m = b_n \oplus \sum_{l=1}^{2^{M-1}} a_l. \tag{6}$$



(a)

(b)

Fig. 3—(a) Universal scrambler. (b) Descrambler.

(Here the subscript l is unrelated to the original position of a_l in the scrambler input stream.) In what follows we show that $P(b_m = 1) \approx \frac{1}{2}$ and that b_m and b_n are nearly independent. For these purposes the use of eq. (6) represents a worst-case analysis. By substitution from eq. (5) we may write

$$\begin{aligned} b_m &= \left[b_n \oplus \sum_{l=1}^{2^{M-1}} s_l \right] \oplus \left[\sum_{l=1}^{2^{M-1}} r_l \right] \\ &\equiv \qquad A \qquad\quad \oplus \qquad R \end{aligned} \tag{7}$$

where A and R equal the first and second bracketed terms, respectively. Since the bits comprising R are independent from those comprising A, R is independent of A. Furthermore, by Lemma 3,

$$P(R = 1) = \tfrac{1}{2}[1 - (1 - 2\epsilon)^{2^{M-1}}]. \tag{8}$$

Therefore, by Lemma 2, no matter what the value of $P(A = 1)$,

$$\delta' \equiv |P(b_m = 1) - \tfrac{1}{2}| \leqq \tfrac{1}{2}[(1 - 2\epsilon)^{2^{M-1}}] \equiv \delta. \tag{9}$$

It follows that so long as $\epsilon > 0$ we may force δ and δ' to be arbitrarily small by choosing a large enough M. Specifically, for a given δ,

$$M \geqq 1 + \log_2 \left[\frac{\ln 2\delta}{\ln (1 - 2\epsilon)} \right]; \qquad 0 < \epsilon < \tfrac{1}{2}. \tag{10}$$

Since δ may be made arbitrarily small, the density function $p(b_m)$ may be made nearly white, and it follows that all first-order statistics of the scrambled sequence may be made nearly white.

Our having shown $P(b_m = 1) \approx \frac{1}{2}$ does not by itself show that the source has been effectively scrambled. For example, consider a sequence

$\{x_n\}$ which consists of consecutive blocks of 100 symbols each. All the symbols in each block are alike; with probability $\frac{1}{2}$ they are all ones, and with probability $\frac{1}{2}$ they are all zeros. Here $P(x_n = 1) = \frac{1}{2}$ for all n, yet the sequence has a very "nonrandom" nature. The implication is that to determine the effectiveness of the scrambler, we must also evaluate the statistical dependence between scrambler output bits.

By definition, the variables b_m and b_n are independent if

$$p(b_m, b_n) - p(b_m)p(b_n) = 0.$$

Accordingly, we define the function

$$\begin{aligned} d(b_m, b_n) &\equiv p(b_m, b_n) - p(b_m)p(b_n) \\ &= p(b_m | b_n)p(b_n) - p(b_m)p(b_n) \end{aligned} \tag{11}$$

and show that the universal scrambler (Fig. 3) bounds the maximum value of $|d(b_m, b_n)|$.

We do a worst-case analysis by assuming that b_m and b_n are related by eq. (7). Further, we ignore the "s" bits appearing in eq. (7) because, being independent of R, they can only weaken the dependence between b_m and b_n. Hence, we may compute the maximum value of $|d(b_m, b_n)|$ by assuming

$$b_m = b_n \oplus R. \tag{12}$$

From eqs. (8) and (9) we note $P(R = 1) = \frac{1}{2} - \delta$ and for convenience we temporarily let $P(b_n = 1) = b$. Substituting these relations and eq. (12) into eq. (11), we find that

$$|d(b_m, b_n)| = 2\delta[b(1 - b)];$$

for

$$b_m, b_n = 0, 1; \qquad m > n > 2^{M+1}.$$

Hence, for $b = \frac{1}{2}$ we obtain the general result

$$|d(b_m, b_n)|_{\max} = \delta/2.$$

Since δ may be forced arbitrarily small if M is given by eq. (10), it follows that any pair of output bits may be made nearly independent, and we may whiten to any degree all the second-order statistics of the source sequence.

We may also show that the joint (second-order) density $p(b_m, b_n)$ approaches that for the white sequence. The derivation of eqs. (6) to (9) shows that both the density $p(b_i)$ and the conditional density $p(b_i | b_j)$ must have values on the interval $[(\frac{1}{2} - \delta), (\frac{1}{2} + \delta)]$ for all

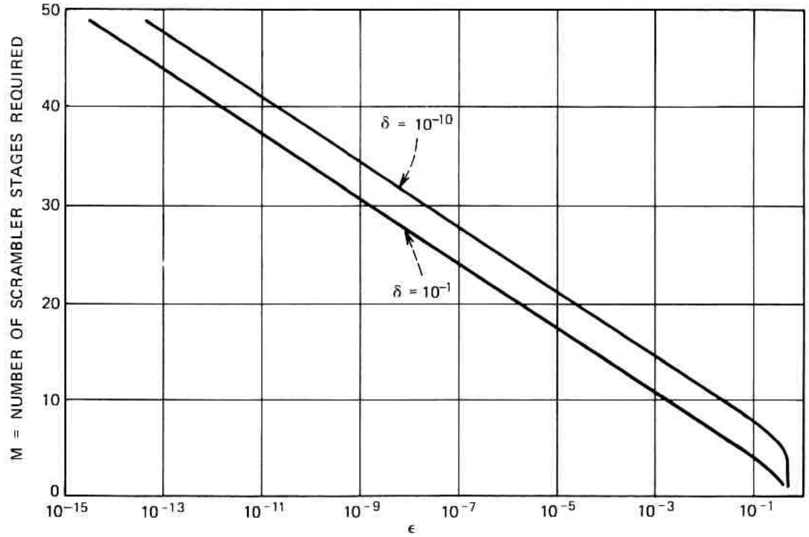


Fig. 4—Scrambler stages required as a function of ϵ and δ.

possible values of b_i and b_j. Hence,

$$(\tfrac{1}{2} - \delta)^2 \leqq p(b_m, b_n) = p(b_m | b_n)p(b_n) \leqq (\tfrac{1}{2} + \delta)^2,$$

or

$$|p(b_m, b_n) - \tfrac{1}{4}| \leqq \delta + \delta^2 \approx \delta,$$

where

$$b_m, b_n = 0, 1; \qquad m > n > 2^{M+1}.$$

For the white sequence $\{w_k\}$, we know $p(w_m, w_n) = \frac{1}{4}$ for $w_m, w_n = 0, 1$. Thus the joint density $p(b_m, b_n)$ may be whitened to any degree by choice of δ, ϵ, and M.

The discussion above constitutes a proof of the following theorem.

Universal Scrambler Theorem: A binary source with unknown output statistics is connected to a binary symmetric memoryless channel and an M-stage self-synchronizing scrambler as shown in Fig. 3. The channel has error probability ϵ where $0 < \epsilon < \frac{1}{2}$. The scrambler output is represented by a sequence of random variables $\{b_n, n = 0, 1, \cdots\}$ and we define $p(b_n)$ to be the first-order and $p(b_m, b_n)$ the second-order density functions for $\{b_n\}$. Then for all $\delta > 0$; $m > n > 2^{M+1}$, and $b_m, b_n = 0, 1$,

$$|p(b_n) - \tfrac{1}{2}| \leqq \delta, \tag{13}$$

and

$$|p(b_m, b_n) - \tfrac{1}{4}| \leqq \delta + \delta^2, \tag{14}$$

provided that

$$M \geqq 1 + \log_2 \left[\frac{\ln 2\delta}{\ln (1 - 2\epsilon)} \right]. \tag{15}$$

Figure 4 shows the relation between M, ϵ, and δ. As seen, M is primarily dependent upon ϵ. This may be clarified by rewriting eq. (15) for small values of ϵ. We then obtain

$$M \geqq \log_2 (1/\epsilon) + \log_2 [\ln (1/2\delta)]; \qquad \epsilon \ll \tfrac{1}{2}.$$

The primary importance of this theorem is conceptual. To avoid inordinate difficulties, many analyses in the literature of digital transmission must assume *a priori* that the digital source is white. The theorem relaxes this restriction by showing that in concept the first- and second-order statistics of *any* source may be made asymptotically white. The practical application of this theorem is discussed in Section VI.

V. AUTOCORRELATION OF THE SCRAMBLED SEQUENCE

An important second-order statistic of the scrambled sequence is its autocorrelation. We define the autocorrelation as the expectation

$$R(k) = E[b_n b_{n+k}],$$

and for convenience we let the value of b_n be $+1$ or -1. Clearly, $R(0) = 1$. For $k \neq 0$, we compute a bound on $|E[b_n b_{n+k}]|$. Following the argument which led to eq. (12), we have

$$|E[b_n b_{n+k}]| \leqq |E[b_n(b_n \oplus R)]|; \qquad n, n+k \geqq 2^{M+1}, k \neq 0.$$

By definition,

$$E[b_n b_m] = \sum_i \sum_j ij P(b_m = i | b_n = j) P(b_n = j).$$

We let $b_m = b_n \oplus R$ and for convenience

$$P(b_n = 1) = b = 1 - P(b_n = -1).$$

Substituting, the dependency on b vanishes, leaving us with

$$E[b_n(b_n \oplus R)] = 2\delta.$$

Hence,

$$\begin{aligned} R(k) &= 1 && \text{for} \quad k = 0, \\ |R(k)| &\leqq 2\delta && \text{for} \quad k \neq 0. \end{aligned} \tag{16}$$

Note that, by forcing δ to a small value with proper choice of M and ϵ, this autocorrelation approaches that for a "white" digital source

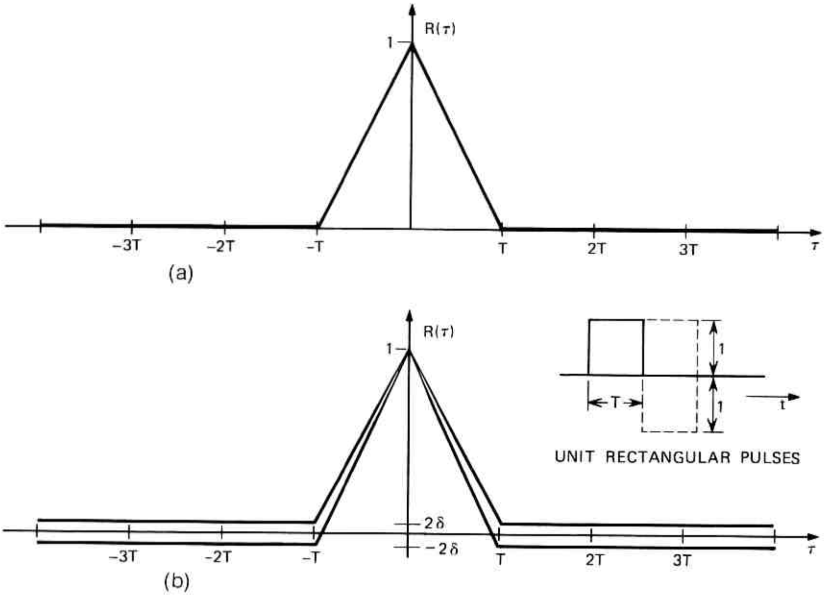


Fig. 5—(a) Autocorrelation for white sequence $\{w_k\}$. (b) Autocorrelation bound for scrambled sequence $\{b_k\}$.

which has $R(k) = 0$, $k \neq 0$. This is shown graphically in Fig. 5 which shows the autocorrelation of "white" and scrambled sources for unit rectangular pulses.

VI. PRACTICAL CONSIDERATIONS

In practice, the binary symmetric channel required by the theorem might be implemented as shown in Fig. 6. The bit r_k is a logic "one" only when the level from the noise generator exceeds some threshold. The threshold is set such that $P(r_k = 1) = \epsilon$. The noise source need not be white, but values of $n(t)$ separated by the baud interval should

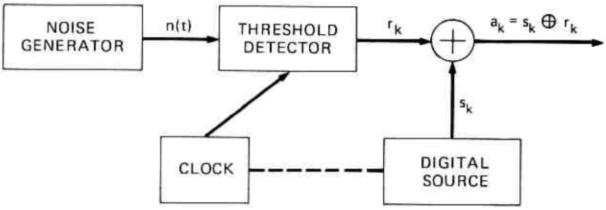


Fig. 6—One possible implementation of the BSC.

be independent. In principle, the combination of this simulated BSC and an M-stage self-synchronizing scrambler will form a universal scrambler capable of satisfactorily whitening the statistics of any binary source. Such a scrambling structure could be used wherever randomized bit statistics are essential and a small error rate can be tolerated (or perhaps corrected by an error-correcting code).

There are, of course, good reasons to avoid actual implementation of the BSC. First, it may be difficult to generate the r_k sequence accurately if ϵ is very small. Second, the deliberate generation of errors, if not impractical, is at least unpalatable. Third, and most important, many commonly encountered sources do not need it. Self-synchronizing scramblers have been used successfully without any prior randomization of the source.[4] In this section we consider the operation of the scrambler without the BSC and show how a designer may use eq. (15) to estimate the required scrambler length for a given application.

From eqs. (2) and (5) we deduce that the net effect of the binary symmetric channel in Fig. 3 is

$$\epsilon < p(a_k \mid a_0, a_1, \cdots, a_{k-1}) < 1 - \epsilon; \qquad a_k = 0, 1, \tag{17}$$

for all k. In other words, because of the BSC there remains a small uncertainty as to the value of any "a" bit, even though all the other "a" bits might be known. As shown in the theorem, this and the dissociation property are sufficient to guarantee effective scrambling. Hence, if the designer knew to begin with that the source itself had the characteristic

$$\epsilon < p(s_k \mid s_0, s_1, \cdots, s_{k-1}) < 1 - \epsilon; \qquad s_k = 0, 1, \tag{18}$$

then no BSC would be necessary, and eq. (15) could be applied directly. For example, bit streams encoded from analog waveforms (such as frequency-division multiplexed speech) often have such a property, and a value for ϵ could be obtained from the coding rule and the amplitude distribution of the analog signal.

For those cases in which a value for ϵ cannot be computed, let us assume that the designer has at least some knowledge of the source pulse density. He could then proceed by estimating a nominal value for ϵ and then decreasing the value to allow some margin. For example, a source which produces bit streams known to vary from 10 to 90 percent "ones" over short periods (say, several hundred bits) would have a nominal $\epsilon = 0.1$. It seems reasonable to allow at least one order of magnitude "margin" in the estimate, resulting in $\epsilon = 0.01$. Then from Fig. 4 we see that an eight-stage scrambler should be sufficient.

Of course, estimating ϵ from the source pulse density does not guarantee that eq. (18) really holds, but if the source sequence is not strictly periodic (the case covered comprehensively by Savage), it is a reasonable procedure. The point here is that even when we are unwilling to commit deliberate errors to guarantee fixed source statistics, we may still use eq. (15) to estimate how large a scrambler is required. Heuristically speaking, eq. (15) is an expression for the "power" of the scrambler, relating the "randomness" of the input and output to the number of scrambler stages.

VII. CONCLUSIONS

We have shown that at the cost of an arbitrarily small error rate it is possible to "whiten" to any degree all the first- and second-order statistics of any binary digital source. This relaxes the restriction frequently found in the literature in which the digital source is assumed *a priori* to produce only independent equiprobable symbols. The key equation in our result [eq. (15)] is useful when designing a standard self-synchronizing scrambler for a given application.

We leave unsolved the problem of whether universal scramblers exist for the M-ary source.

VIII. ACKNOWLEDGMENTS

I wish to thank M. B. Romeiser for editorial suggestions and Miss J. M. Michel for assistance in computer programming. I also had the benefit of stimulating discussions on the subject of scrambling with T. M. Chien and R. J. Deaton.

APPENDIX

Proof of Lemma 1*

For convenience, we assume that in Fig. 2a the scrambler initially contains all zeros. Since each scrambler output bit is ultimately a mod-2 summation of selected input bits, we may write

$$b_n = \sum_{k=0}^{n} h_k a_{n-k}, \tag{19}$$

where the binary sequence h_k performs the selection. We note that if $a_0 = 1$ and $a_i = 0$ for all $i > 0$, then $\{b_n\} = \{h_n\}$. But under these

* Independently of the author, U. Henriksson has developed[5] a proof of a similar lemma.

conditions, as described in Section II, $\{b_n\}$ will be a maximal length sequence. Hence, $\{h_n\}$ must itself be a maximal length sequence.

Now we consider the two output bits b_n and b_m. We wish to count the number of "a" bits which entered the summation for b_m but not b_n. We have

$$\underset{\text{(a)}}{b_n = \sum_{k=0}^{n} h_k a_{n-k}} \qquad \underset{\text{(b)}}{b_m = \sum_{k=0}^{m} h_k a_{m-k}.} \tag{20}$$

Since $m > n$,

$$\begin{aligned} b_m &= \sum_{k=0}^{m-n-1} h_k a_{m-k} \oplus \sum_{k=m-n}^{m} h_k a_{m-k} \\ &= \sum_{k=0}^{m-n-1} h_k a_{m-k} \oplus \sum_{k=0}^{n} h_{m-n+k} a_{n-k} \end{aligned} \tag{21}$$

Examination of the subscript range shows that all the "a" bits selected by the first summation in eq. (21) are unique to b_m. By comparing the second summation with eq. (20a) we see that the additional "a" bits which enter b_m but not b_n are those for which

$$h_{m-n+k} - h_{m-n+k}h_k = 1.$$

Hence,

$$J_{mn} = \sum_{k=0}^{m-n-1} h_k + \sum_{k=0}^{n} [h_{m-n+k} - h_{m-n+k}h_k],$$

or

$$J_{mn} = \sum_{k=0}^{m-n-1} h_k + \sum_{k=0}^{n} h_{m-n+k} - \sum_{k=0}^{n} h_{m-n+k}h_k, \tag{22}$$

where addition is now in the usual sense.

We examine this expression in detail, recalling that the sequence $\{h_k\}$ has period $p = (2^M - 1)$ and the given condition $n > 2^{M+1}$.

Case (1): If $m - n = Kp$, $K = 1, 2, \cdots$, then

$$h_k = h_{m-n+k} = h_{m-n+k}h_k$$

for all values of k. Hence the second and third summations cancel. But then the first summation contains K periods of a maximal length sequence. Since each period contains exactly $2^{(M-1)}$ ones,[6] the first summation totals at least $2^{(M-1)}$.

Case (2): If $m - n \neq Kp$, then it is easily shown[6] that the sequence formed by the term-by-term product $h_{m-n+k}h_k$ has period p and contains $2^{(M-2)}$ ones per period. The sequence $\{h_{m-n+k}\}$ contains $2^{(M-1)}$ ones per period. Hence the net contribution of the second and third summations is $2^{(M-2)}$ ones per period. Since $n > 2^{M+1} > 2p$, the summations cover at least two periods. Thus their net total is at least $2^{(M-1)}$.

Thus for either case,

$$J_0 = \min_{m,n>2^{M+1}} [J_{mn}] = 2^{M-1}.$$

REFERENCES

1. Savage, J. E., "Some Simple Self-Synchronizing Digital Data Scramblers," B.S.T.J., *45*, No. 2 (February 1967), pp. 449–487.
2. Gallager, R. G., *Information Theory and Reliable Communication*, New York: John Wiley and Sons, 1968, pp. 225–238.
3. Peterson, W. W., *Error Correcting Codes*, Cambridge: M.I.T. Press, 1961, pp. 251–270.
4. Fracassi, R. D., and Tammaru, T., "Megabit Data Service with the 306A Data Set," Bell Laboratories Record, *49*, No. 10 (November 1971), pp. 310–315.
5. Henriksson, U., "On a Scrambling Property of Feedback Shift Registers," IEEE Trans. on Commun., *20*, No. 5 (October 1972), pp. 998–1001.
6. Golomb, S. W., *Shift Register Sequences*, San Francisco: Holden-Day, 1967, pp. 23–59, 88.

THE BELL SYSTEM TECHNICAL JOURNAL
Vol. 52, No. 10, December, 1973
Printed in U.S.A.

Dispersion and Equalization in Fiber Optic Communication Systems

By D. M. HENDERSON

(Manuscript received July 2, 1973)

The additional optical power required at the repeater input in a fiber optic communication system due to intersymbol interference is experimentally measured. In the experiment, the intersymbol interference which results from differential mode delay in multimode fibers is minimized with a five-tap transversal equalizer. Error rate measurements are performed using five fibers ranging from 0.01 km to 1.25 km in length. In this manner, the additional optical power required to achieve a given error rate is found as a function of pulse width. The measured values compare favorably with the power penalties predicted by Personick. The trade-off between excess optical power and equalization penalty in dispersion-limited fiber systems is discussed.

I. INTRODUCTION

The temporal spreading of light pulses in an optical fiber can impose a limit on the highest data rate transmitted by a fiber optic communication system. Such spreading arises from differential mode delay in multimode fibers and material dispersion in both single-mode and multimode fibers.[1] The fiber materials and geometry together with the type of light source determine the magnitude of each effect. In this paper, we report the measurement of the additional optical power required to compensate for the loss in sensitivity resulting from the need to equalize detected light pulses that experience mode-delay spread. The experiment was carried out to determine the feasibility and practicality of equalization in dispersion-limited fiber systems.

In the experiment, light from a Burrus[2]-type gallium arsenide light-emitting diode (LED) digitally modulated at 48 Mb/s is coupled into a liquid-core fiber.[3] Intersymbol interference in the detected pulse train is reduced with a transversal equalizer[4] by forcing zero crossings in the pulse response at all sampling times but one. Error rate measurements are made as a function of optical power on five sections of fiber ranging from 0.01 kilometer to 1.25 kilometers in length. From these measurements, the power required to assure a given error rate can be determined for the five pulse widths encountered. It is found that the measured power penalty due to intersymbol interference compares favorably with the values calculated by Personick.[5] The trade-off between excess optical power and increased repeater spacing afforded by equalization is discussed in the concluding section of this paper.

II. DESCRIPTION OF EXPERIMENT

A block diagram of the experimental setup is shown in Fig. 1. For a source we use a Burrus-type diffused junction GaAs LED driven with a 48-Mb/s pseudorandom pulse stream. The LED output is first collimated, then attenuated as necessary with neutral-density filters, and finally focused onto the input of a liquid-core fiber. For the half-duty-cycle, return-to-zero, input light pulses used in the experiment, a maximum of −14.4 dBm average optical power can be coupled into the fibers.

In Table I, the measured loss and pulse width are shown for the five lengths of fiber used. The pulse width for the shortest section is deter-

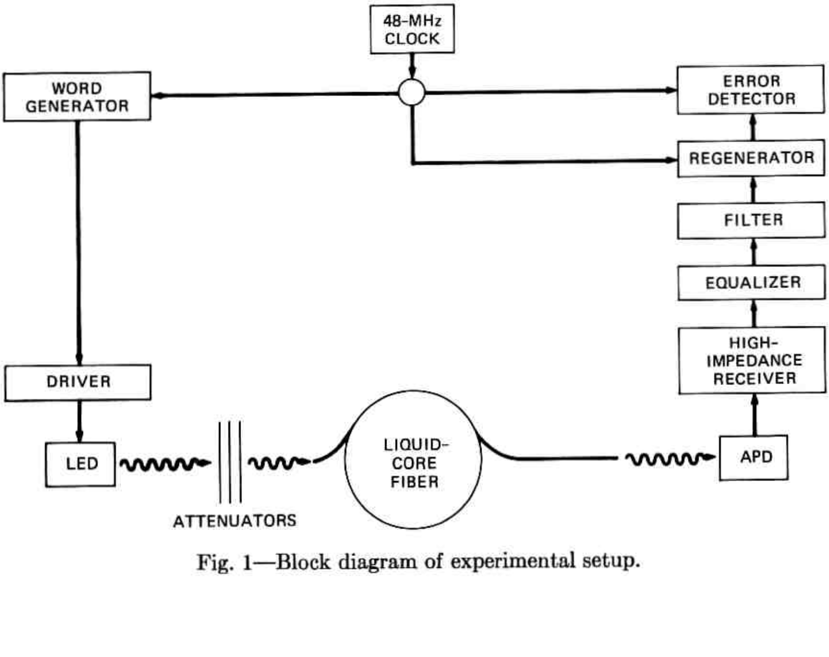


Fig. 1—Block diagram of experimental setup.

TABLE I—MEASURED LOSS AND PULSE WIDTHS FOR THE FIVE FIBERS TESTED

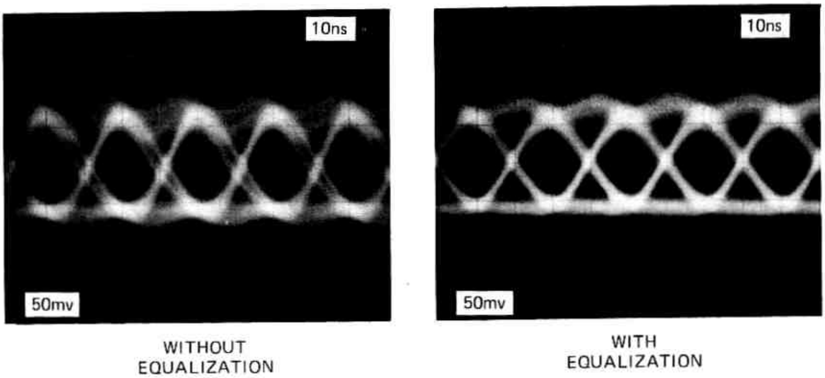
Fiber length (km)	Differential loss (dB/km)	Total loss (dB)	Rms pulse width (ns)
0.01	27.8	≈0	3.0
0.50	27.8	13.9	7.5
0.75	30.0	22.5	6.8
1.12	25.5	28.6	10.9
1.25	29.6	37.0	10.5

mined by the input pulse width and the bandwidth of the RCA C30817 avalanche photodiode (APD) detector. The additional width that is observed for the remaining four fibers is due to differential mode delay. Note that the 0.75-km fiber has greater differential loss and exhibits less pulse spreading than the fiber 0.50 km in length. Higher-order modes are presumably more highly attenuated in the 0.75-km fiber. The 1.25-km fiber is obtained by splicing the two together.

A high-impedance receiver is used, the first stage of which tends to integrate the detected light pulses. Incorporated in the receiver is an appropriate compensating network to assure that the receiver response is flat over the bandwidth of interest. Such a design has been shown to give improved signal-to-noise performance and reduced avalanche gain over a conventional, nonintegrating receiver.[5] At near-optimal avalanche gain of ≈ 60, a 10^{-9} error rate is realized with -53.7 dBm average optical power for the shortest section of fiber.

In order to equalize the detected optical pulses of various widths and shapes, a five-tap transversal equalizer is utilized. The tapped delay line with one time slot between taps is realized with RG188 coaxial cable. Variable gain and polarity of the signal picked off at each tap are achieved with MC1733 wideband differential amplifier integrated circuits. An additional wideband amplifier serves as a summing amplifier to recombine the signals. The equalizer is manually adjusted for each fiber tested.

Figure 2 shows the eye diagrams of the output of the "Nyquist" filter both with and without equalization for the shortest section of fiber. Here the equalizer serves to modify slightly the combined bandwidth response of receiver and filter, giving an improved response. The eye diagrams for the 1.12-kilometer fiber are shown in Fig. 3. In this case, differential mode delay has resulted in significant intersymbol interference. Adjusting the equalizer for zero crossings results in the equalized signal shown.



WITHOUT EQUALIZATION

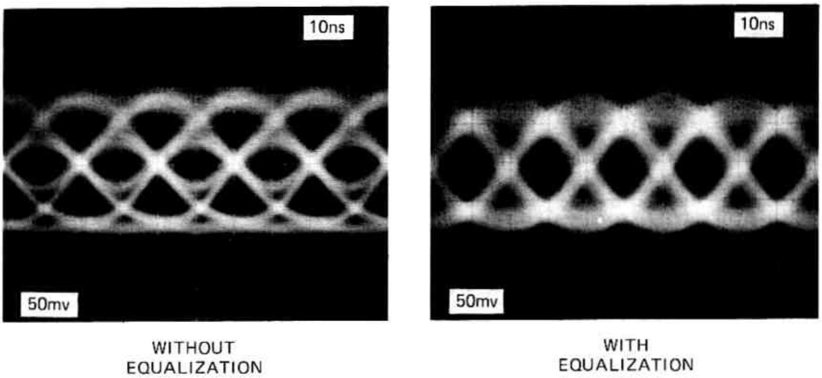
WITH EQUALIZATION

Fig. 2—Eye diagrams of input to regenerator with and without transversal equalizer for the 0.01-km fiber.

The equalized pulse train is regenerated and then compared with the original pseudorandom sequence in an error detector. Error rate measurements are then made as a function of optical power for each fiber. Optical power readings are taken with a Coherent Radiation Model 212 power meter.

III. RESULTS

The shape of the detected optical pulse strongly influences the amount of intersymbol interference and accordingly the additional optical power required. To accurately determine these shapes, a wideband, low-noise, 50-ohm amplifier is substituted for the high-impedance receiver. Through signal averaging with boxcar integration,



WITHOUT EQUALIZATION

WITH EQUALIZATION

Fig. 3—Eye diagrams of the input to regenerator with and without transversal equalizer for the 1.12-km fiber.

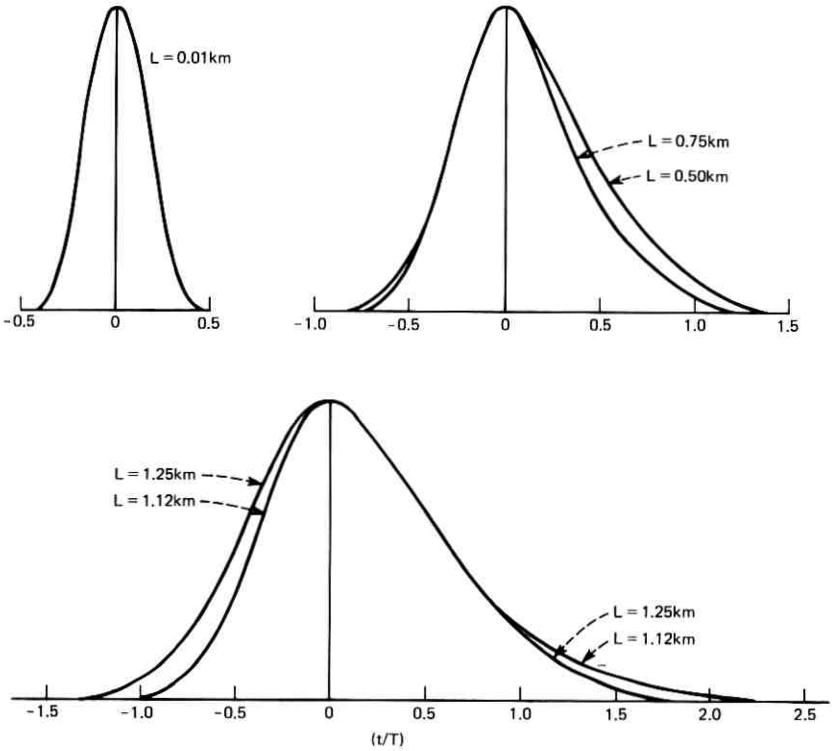


Fig. 4—Detected pulse shapes for the five fiber lengths tested.

the pulse shape can be recorded with signal-to-noise ratios in excess of 100. The pulse shapes for the five fiber lengths are plotted in Fig. 4 versus the normalized parameter t/T, where T represents one time slot (20.83 ns). From these data, the rms pulse widths σ of Table I were computed from the relation $\sigma^2 = [\int t^2 f(t)dt)] - [\int t f(t)dt]^2$. Here $f(t)$ is the measured pulse shape normalized to unit area. The integration was numerically performed. In the figure, it is seen that for the shortest length, the pulse is confined to a single time slot. For the intermediate lengths the pulses are effectively confined to two time slots; for the longest lengths to three time slots.

A compilation of the error rate measurements is shown in Fig. 5. We have not attempted to fit the best curve to the data for each fiber. Instead we show superimposed on the data a curve of fixed shape which minimizes the deviation from the mean for all the fibers. The measured values fall within ±1/8 dB of the selected curve.

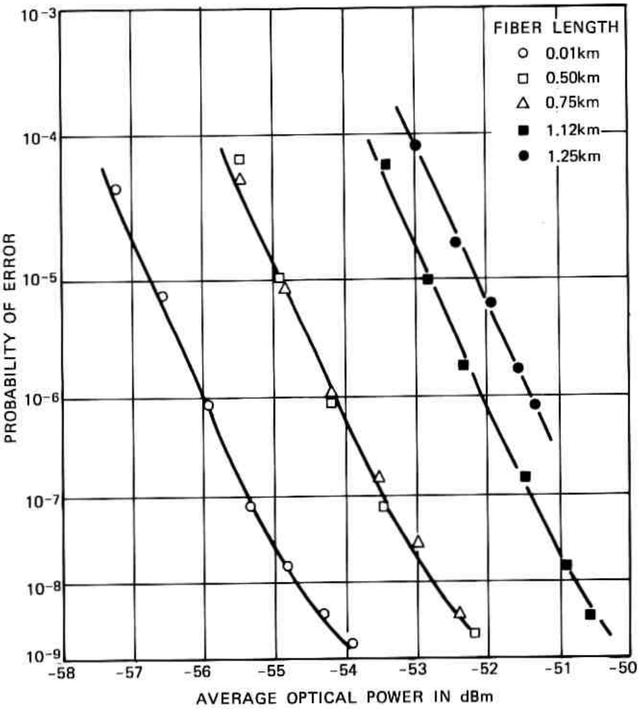


Fig. 5—Results of error rate measurements for the five fiber lengths tested.

The dependence of required optical power on the detected pulse width and shape has been calculated by Personick.[5] In this calculation, the degradation in signal-to-noise ratio is found when equalizing from the detected pulse shape to a raised cosine shape. These results are presented in Fig. 6 in terms of the additional optical power required to maintain a fixed error rate. The dependence is shown for an exponential pulse $(1/\sigma)\cdot\exp(-t/\sigma)$ and a Gaussian pulse $(1/\sqrt{2\pi}\sigma)\cdot\exp(-t^2/2\sigma^2)$. When $(\sigma/T) < 0.25$ the power penalty for the two shapes is about equal. This behavior follows from the fact that little difference exists between the frequency spectra of the pulses over the range of interest $(0 < \omega/2\pi < 1/T)$ in the narrow pulse limit. As the pulse width increases, the frequency spectrum of the Gaussian falls off much more rapidly than the exponential, resulting in a much larger power penalty.

Also shown are the measured points taken from Table I and Fig. 5. As no measurement is performed in the limit $\sigma/T \rightarrow 0$, the point at $\sigma/T = 0.15$ has been assigned the value of 0.25 dB to coincide with

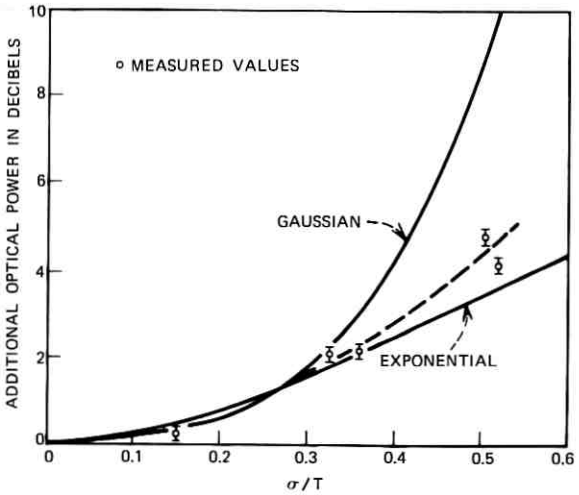


Fig. 6—The additional optical power required to maintain a fixed error versus normalized pulse width.

the Gaussian curve at that point because the measured pulse shape appears Gaussian rather than exponential. All other points have been scaled upward by this amount. Scaling this point to the value for the exponential pulse shape would merely shift all points up by an additional 0.12 dB and not affect the results significantly.

The measured points do not fall on a continuous curve but rather define a range of values. A dashed line which bisects the measured points is shown. It is interesting to note that the 0.75-km and 1.25-km fibers which lie above the dashed line have a more symmetric pulse shape and less of a tail than the 0.50-km and 1.12-km fibers which fall below the line. Such dependence is expected because the presence of the tail leads to spectra which fall off less rapidly in the frequency domain and therefore will suffer less power penalty. Calculations confirm this dependence.[6]

These results point out the important role the detected pulse shape plays. The decidedly asymmetric pulses that result from differential mode delay in liquid-core fibers lead to power penalties that increase much more slowly with pulse width than the penalty predicted from Gaussian pulses.

IV. DISCUSSION

In order to point out the benefits and limitations of equalization in dispersion-limited fiber optic communication systems, we treat a specific case below. In the example, optical pulses from an LED source are transmitted by a low-loss fused-silica multimode fiber.

The fiber loss is considered first. Define

$$R(\mathrm{dB}) = 10 \log [P_{\mathrm{in}}/P_{\mathrm{req}}(T_0)], \tag{1}$$

where P_{in} is the optical power coupled into the fiber and P_{req} is the power required in the absence of intersymbol interference to assure a given error rate at the data rate $1/T_0$. The distance $L(\mathrm{km})$ over which one can communicate is governed by the fiber attenuation coefficient $\alpha(\mathrm{dB/km})$ according to

$$R(T_0) - \alpha L \geqq 0. \tag{2}$$

Consider the fused-silica multimode fiber announced by Corning Glass Works[7] for which $\alpha = 4$ dB/km at 0.85 μm. If we take for P_{req} the measured value of -53.7 dBm at 1 error in 10^{+9} and assume that -15 dBm can be launched into the fiber, then the inequality in eq. (2) would hold for distances less than ≈ 9.6 km. Additional signal-to-noise margin that is required would reduce this value in proportion to the fiber loss.

The required power is dependent on the data rate. For the high-impedance receiver it has been shown that[5] $P_{\mathrm{req}} \approx (T)^{-7/6}$. By defining $\tau = 10 \log (T_0/T)^{7/6}$, eq. (2) can be written

$$R(T_0) - \alpha L \geqq \tau(T_0/T). \tag{3}$$

From this result the maximum separation can be found as a function of data rate. In Fig. 7 the curve marked 4 dB/km loss shows the dependence. The inequality is satisfied for distances L below the line.

Personick, et al.,[8] have studied the pulse spreading in such a low-loss fused-silica multimode fiber excited by an LED source. For a fiber with a tailored index-of-refraction profile which gives an effective numerical aperture of 0.12 and an effective diameter of 96 μm, they find that material dispersion is the dominant pulse-spreading mechanism due to the large spectral width (400 Å) of the LED. The measured rms pulse width is found to be $\sigma/L \equiv \sigma' = 1.75$ ns/km.

If we do not equalize, but restrict the pulse width to avoid intersymbol interference, then an additional limit is imposed on L. As an example, take the arbitrary restriction $\sigma' \cdot L/T < 0.35$. In Fig. 7, this inequality is satisfied to the left of the curve marked material dispersion. Therefore, the area in the lower left-hand side bound by the two limiting curves shows the available working distances versus data rate. It should be noted that another possible way to reduce this

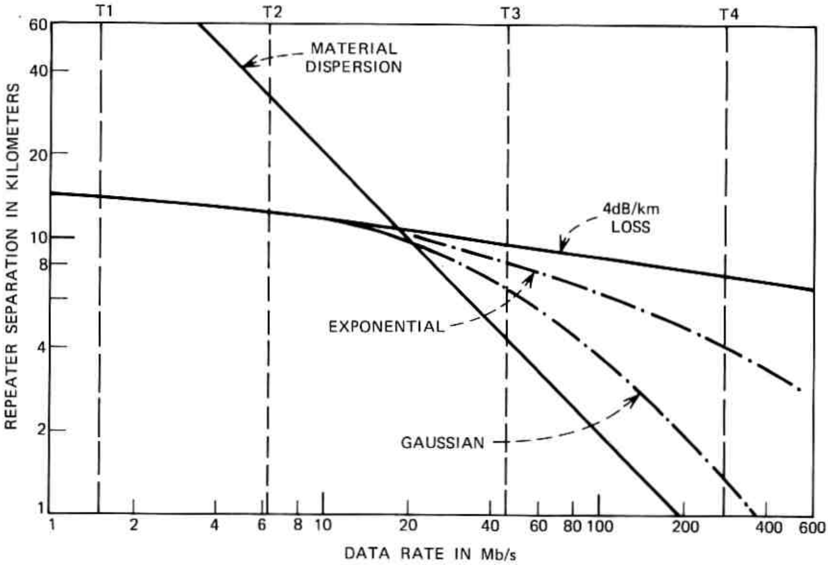


Fig. 7—Maximum repeater spacing versus data rate for the fiber system described in text.

dispersion is to trade off optical power and LED bandwidth by reducing the spectral width of the output. We deal with equalization alone in this paper.

At higher data rates, dispersion limits the repeater spacing before the fiber loss limit occurs. Below we consider the effect of using this excess power for equalization to increase the spacing. The power penalty has been presented as a function of pulse width in Fig. 6. Here σ/T become $\sigma' L/T$. Let $p(L, T)$ expressed in dB represent the power penalty. Equation (3) then becomes

$$R(T_0) - \alpha L \geqq \tau(T_0/T) + p(L, T). \tag{4}$$

The solutions to this equation are shown in Fig. 7 for both the Gaussian and exponential pulses. At the data rate used in this experiment (48 Mb/s) the repeater separation is increased from ≈ 4.2 km to ≈ 6.5 km for the Gaussian-shaped pulses. At that distance the rms pulse width is $\sigma/T \cong 0.55$. For exponential pulses the spacing could be as large as 8.4 km with accompanying pulse width $\sigma/T = 0.70$. The exact distance will depend strongly on the detailed shape of the detected pulses as noted previously. In any case, increases of at least

50 percent would be possible. At higher data rates the potential increases are greater still.

V. SUMMARY

We have experimentally measured the additional optical power that is required to maintain a fixed error rate for digitally modulated light pulses that encounter differential mode delay in multimode fibers. The resulting power penalty versus pulse width compares favorably with the values predicted by Personick. We find that the asymmetric shape of the mode-delay-spread pulses results in penalties which increase much more slowly with pulse width than does the penalty predicted for Gaussian pulses. Each potential dispersion-limited fiber system must be examined separately to determine the potential increase in repeater spacing that equalization offers as the pulse spreading mechanisms vary significantly among different sources and fibers. The example presented shows graphically how one can utilize excess optical power in equalizing delay distortion and thereby maximize the repeater spacing for a given data rate.

VI. ACKNOWLEDGMENTS

The author wishes to express his appreciation to J. S. Cook, S. D. Personick, and J. Stone for many helpful discussions. Peter K. Runge generously furnished the high-impedance receiver. The technical assistance of A. R. McCormick was invaluable.

REFERENCES

1. Gloge, D., "Dispersion in Weakly Guiding Fibers," Appl. Opt., *10*, No. 11 (November 1971), pp. 2442–2445.
2. Burrus, C. A., and Miller, B. I., "Small-Area, Double-Heterostructure Aluminum-Gallium Arsenide Electroluminescent Diode Source for Optical-Fiber Transmission Lines," Opt. Commun., *4*, No. 4 (December 1971), pp. 307–309.
3. Stone, J., "Optical Transmission in Liquid-Core Quartz Fibers," Appl. Phys. Lett., *20*, No. 7 (April 1972), pp. 239–240.
4. Lucky, R. W., Salz, J., and Weldon, E. J., Jr., *Principles of Data Communication*, New York: McGraw-Hill, 1968, pp. 128–165.
5. Personick, S. D., "Receiver Design for Digital Fiber Optic Communication Systems," B.S.T.J., *52*, No. 6 (July-August 1973), pp. 843–886.
6. Personick, S. D., private communication.
7. Keck, D. B., Mauer, R. D., and Schultz, P. C., "On the Ultimate Lower Limit of Attenuation in Glass Optical Waveguides," Appl. Phys. Lett., *22*, No. 7 (April 1973), pp. 307–309.
8. Personick, S. D., Hubbard, W. M., Gloge, D., Holden, W. S., Dawson, R. W., Chinnock, E. L., and Lee, T. P., "Measurements of Material Dispersion in Optical Fibers," presented at the Conference on Laser Engineering and Applications, Washington, D. C., May 30–June 1, 1973.

Copyright © 1973 American Telephone and Telegraph Company
THE BELL SYSTEM TECHNICAL JOURNAL
Vol. 52, No. 10, December, 1973
Printed in U.S.A.

Phase Dispersion Characteristics During Fade in a Microwave Line-of-Sight Radio Channel

By M. SUBRAMANIAN, K. C. O'BRIEN, and P. J. PUGLIS

(Manuscript received February 5, 1973)

Measurements of phase and amplitude dispersion over a 20-MHz band have been made on a 42-km, 6-GHz, line-of-sight microwave link. A novel technique is introduced for measuring the phase dispersion induced by the propagation path. Specifically, the amplitudes and relative phases of four tones separated equally by 6.6 MHz have been continuously monitored over a period of four months. The data show that there is usually measurable (0.02 degree/(MHz)^2) phase distortion over the 20-MHz band during those fades whose depth exceeds about 20 dB. These dispersive fades, which usually last a few seconds, typically occur along with shallow and essentially nondispersive fades that have durations of several minutes. However, only the dispersive fades exhibit a phase nonlinearity. Analysis of 16 events measured in the autumn of 1970 yield the following results.

(*i*) *The distribution curve describing the fraction of time that phase nonlinearity (quadratic) exceeds a given value follows a log-normal distribution.*

(*ii*) *The quadratic phase nonlinear coefficient exceeds an average value of 0.1 degree/(MHz)^2 for fades with depth larger than 34 dB from the nominal level. This corresponds to a time delay distortion of 0.55 nanosecond over 1-MHz band.*

(*iii*) *The correlation between log-amplitude and phase nonlinear coefficients yields a correlation coefficient whose magnitude is fade-depth dependent and whose sign varies from event to event.*

The experimental technique of measuring phase dispersion reported here may be of interest not only for propagation studies but also in other systems such as measurement of characteristics of electrical networks. The statistical results obtained on the phase characteristics may prove of interest in formulating an analytical model. Further, they may be of significance in the design of existing and future microwave systems.

I. INTRODUCTION

Fading in microwave communication channels has been the subject of investigation by many workers for a considerable length of time. However, the emphasis on these studies has been more toward the behavior of the amplitude characteristics of the signal rather than of its phase characteristics. The purpose of this study was to investigate the phase characteristics of a microwave signal transmitted over a typical tropospheric, line-of-sight link.* Specifically, the following topics were addressed in the experiment.

(*i*) Measurement of phase variations over a microwave radio channel, as a function of both frequency and time; obtaining from the experimental results statistics on the phase nonlinearity in a microwave radio channel.

(*ii*) Correlation, if any, between the amplitude and phase distortions.

Measurements were made in late 1970 on a TH-3 radio channel operating between Atlanta and Palmetto, Georgia. The experiment was conducted as an adjunct to an ongoing study by other members of Bell Laboratories. The transmitter located in Atlanta and the front end of the receiver situated in Palmetto were common to both experiments. However, a different set of apparatus was employed for the measurement of amplitude and phase in the present study. G. M. Babler[1] has reported on the experimental layout of the microwave link.

This paper addresses three major areas: experimental technique and arrangement, measured data, and statistical analysis. The experimental technique is a novel one in that it measures directly the phase difference between pairs of transmitted tones separated in frequency. Only the phase difference induced by the propagation path is measured, not the transmitter and receiver beat oscillator fluctuations. More specifically, at a carrier frequency of 6 GHz, four tones separated by 6.6 MHz and with a definite phase relation are transmitted over a 42-km line-of-sight path. At the receiving end, the signals are brought to a 70-MHz IF and filtered out. The amplitude of each tone is continuously monitored. The phase difference between each adjacent pair is measured.

Data were recorded during roughly 100 events in which the fade

* As a result of the findings reported here, a more comprehensive, higher resolution measurement program has been undertaken to characterize more completely the dispersive microwave channel. It is expected that a summary of these results will be published here in the near future.

depth exceeded 10 dB. These fades can generally be divided into two categories. The majority of them are relatively shallow (<20 dB), long-lasting (minutes) events which show little amplitude and nonlinear phase distortion, although linear phase dispersion corresponding to path length variations may be observed. A second class of events are those that exhibit deep and brief (seconds) fades showing substantial amplitude dispersion and nonlinear phase dispersion. This second class could present difficulties to communication systems.

Twenty-six dispersive fading events whose fade depth exceeded 20 dB were observed during the autumn of 1970, a smaller number than usual. Only 16 of these were analyzable as a result of equipment malfunction. This analysis yielded the following results.

(*i*) The distribution curve describing the fraction of time that the phase nonlinearity (quadratic) exceeds a given value follows an almost log-normal distribution.

(*ii*) The quadratic phase nonlinear coefficient exceeds an average value of 0.1 degree/(MHz)^2 for fades that are deeper than 34 dB from the nominal level. This corresponds to a time delay distortion of 0.55 nanosecond over 1-MHz band.

(*iii*) No simple relationship seems to exist for the correlation between the quadratic log-amplitude and the quadratic phase nonlinear coefficients.

II. DESCRIPTION OF PHASE DISPERSION MEASUREMENT

This experiment measures the effect of the transmission path on the relative phase of signals at different frequencies. Specifically, a "picket fence" of tones in the 6-GHz range, separated from each other by 0.55 MHz, are transmitted over an approximately 42-km path. These tones are generated at the transmitter by means of a "picket-fence" generator developed by G. A. Zimmerman of Bell Laboratories. The experimental apparatus measures the phase difference between pairs of these tones separated by 6.6 MHz in such a way that only the phase difference induced by the transmission path is measured, and not the transmitter and receiver beat oscillator fluctuations. Although a closer spacing between the tones would have been more desirable, practical considerations limited the selection to four tones distributed uniformly over the 20-MHz radio channel. Further, at the time of initiation of this experiment, the available statistics[2] on the amplitude dispersion during deep microwave fading did not seem to indicate any significant fine structure over a 20-MHz band, and consequently no fine structure of phase nonlinearity of significant magnitude was expected as a result of fading by multiray phenomena.

To understand the measurement, it is important to trace a signal all the way through the system. A 70.4-MHz signal inserted into the picket-fence generator is divided by 128 and fed into a pulse generator. The resulting pulses are then mixed with the 70.4-MHz signal producing a picket fence of tones separated by 0.55 MHz and centered at 70.4 MHz. In addition to having equal amplitude, the tones have the same initial phase. The time-dependent frequency error of the 70.4-MHz signal caused by oscillator fluctuations may conveniently be expressed as 128 $\Delta\omega_o$ (t). The nth tone can then be written, with $\omega_o = 70.4/128$ MHz, as

$$A_o \cos\left[n(\omega_o + \Delta\omega_o)t\right], \tag{1}$$

where A_o is the amplitude of each picket. Up conversion at the transmitter by a beat oscillator with angular frequency ω_T yields

$$A_T \cos\left(\{[\omega_T + n(\omega_o + \Delta\omega_o)]\}t + \phi_T\right), \tag{2}$$

where A_T is the amplitude of the transmitted up-converted signal and Φ_T is the phase shift introduced by the beat oscillator. After transmission through the atmosphere, the signal becomes

$$A'_T A_n \cos\{[\omega_T + n(\omega_o + \Delta\omega_o)]t + \phi_T + \phi^a_n\}, \tag{3}$$

where A_n and Φ^a_n are the amplitude and phase modulation introduced by the atmosphere on the nth tone and A'_T is the amplitude of the received signal without any modulation by the atmosphere. Down conversion at the receiver yields

$$A_R A'_T A_n \cos\{[\omega_T - \omega_R + n(\omega_o + \Delta\omega_o)]t + \phi_T - \phi_R + \phi^a_n\}, \tag{4}$$

where $A_R A'_T A_n$ is now the amplitude of the IF signal, ω_R is the angular frequency of the receiver beat oscillator, and Φ_R is the phase shift introduced by the receiver beat oscillator. Consider that this IF signal is mixed with a signal from a standard oscillator (phase changes are included in $\Delta\omega_s$) described by

$$A_s \cos(\omega_s + \Delta\omega_s)t. \tag{5}$$

We then have, for the upper and lower sideband signals (ignoring a constant),

$$\begin{aligned} S_n = A_s A_R A'_T A_n \cos\{[\omega_T - \omega_R + n(\omega_o + \Delta\omega_o) \pm (\omega_s + \Delta\omega_s)]t \\ + \phi_T - \phi_R + \phi^a_n\}. \end{aligned} \tag{6}$$

In order to compare two tones (m and n) with phases Φ_n^a and Φ_m^a, $n > m$, choose

$$\omega_s = \left(\frac{n - m}{2}\right)\omega_o. \tag{7}$$

By appropriate filtering, S_n and S_m are isolated. After being mixed with ω_s, the lower sideband of S_n denoted by S_{nl} and the upper sideband of S_m denoted by S_{mu} are separated, yielding

$$S_{nl} = A_s A_R A'_T A_n \cos\left\{\left[\omega_T - \omega_R + \left(\frac{n + m}{2}\right)\omega_o + n\Delta\omega_o - \Delta\omega_s\right]t \right. \\ \left. + \phi_T - \phi_R + \phi_n^a\right\} \tag{8}$$

$$S_{mu} = A_s A_R A'_T A_m \cos\left\{\left[\omega_T - \omega_R + \left(\frac{n + m}{2}\right)\omega_o + m\Delta\omega_o + \Delta\omega_s\right]t \right. \\ \left. + \phi_T - \phi_R + \phi_m^a\right\}. \tag{9}$$

Measurement of the difference in phase of S_{nl} and S_{mu} yields

$$[(n - m)\Delta\omega_o - 2\Delta\omega_s]t + \phi_n^a - \phi_m^a \equiv \Delta\phi. \tag{10}$$

If the first term is small, we have a direct measure of the quantity of interest. In this case, ω_s and ω_o are derived from similar highly stable sources (Hewlett-Packard 105B quartz oscillators and General Radio 1165 and 1163 frequency synthesizers). The phase drifts, $12\Delta\omega_o t$ and $2\Delta\omega_s t$ [in eq. (10) corresponding to the selected value for $(n - m) = 12$ in the experiment], are approximately -1 degree/day. Note that, if the frequency deviation is the same for both oscillators, even this small phase drift cancels out. Further, the transmitter and receiver oscillators are manually synchronized every day, except on Sundays. However, this small drift produces a running phase difference (which is linear with time) between tones and can be subtracted out during the data analysis by measuring the linear slope before and after a fading event, thus causing no error to the data. The short-term rms frequency deviations (<1 s) are on the order of 1×10^{-8}/ms or 1×10^{-11}/s. At the frequency of 6.6 MHz, this corresponds to a rms phase noise of 0.024 degree. When the picket-fence generator was run directly into the phase and amplitude measurement system, the observed rms phase noise out of the network analyzers was 0.03 degree. This is considerably smaller than the accuracy of the measuring equipment which is ±1 degree.

Figure 1 is a block diagram of a simplified system. Of the entire assembly of pickets, those numbered "1" and "13" are separated out by the narrow bandpass filters. Both ω_1 and ω_{13} are mixed with an $\omega_s = (\omega_{13} - \omega_1)/2$ derived from the same source. The second pair of filters labeled $[(\omega_1 + \omega_{13})/2]$ separates the appropriate sidebands.

Figure 2 is a block diagram of the actual measuring system. The entire picket fence is received and divided into four channels. In each channel one tone is filtered through. The four frequencies used are 53.35, 59.95, 66.55, and 73.15 MHz, spanning the desired range of approximately 20 MHz, the width of radio channel. The nominal level of the individual tones at the input to the system is approximately −46 dBm.

Each filtered tone is amplified and mixed with a signal of frequency ω_s from the standard oscillator. The output is then split into its two sidebands, thus producing three pairs of signals, each pair having the same frequency difference.

The network analyzer (Hewlett-Packard Model 676A-H05, which is a modified version of 676A to meet our requirements) measures the relative phase (10 mV/degree), the amplitudes (50 mV/dB), and the ratio of the amplitudes of the two input signals. Amplitudes can be measured over a maximum of 80-dB range with up to 0.01-dB resolution and ±1.5-dB accuracy. Phase difference can be measured with up to 0.02-degree resolution and ±1-degree accuracy if the tone levels are not too widely different. The measured results on the system indicate that for tones whose amplitude difference is less than 30 dB, which is well within the limits on requirements of our experiment, these specifications were satisfactorily met. The network analyzer performs its amplitude and phase measuring functions at an IF frequency of 100 kHz. The accuracies quoted above can be achieved only if this 100 kHz is stable to less than 100 Hz. Because of drifts in the beat frequency oscillators at transmitter and receiver, the frequencies of the signals at the input of the network analyzers drift (together) by a few hundred hertz. In order to maintain the network analyzer's IF frequency constant, it is necessary to track these drifts. This is accomplished by sampling in each network analyzer one of the input signals, amplifying it to a constant level, mixing it with a 100-kHz signal, filtering out one (the upper) sideband, and employing this signal as a local oscillator in the network analyzer. The IF strip in the network analyzer then operates at a constant 100 kHz. This arrangement is shown in Fig. 3. The dc voltage outputs of the network analyzers are recorded on a 7-channel FM tape recorder with a dc-to-625-Hz bandwidth. The four

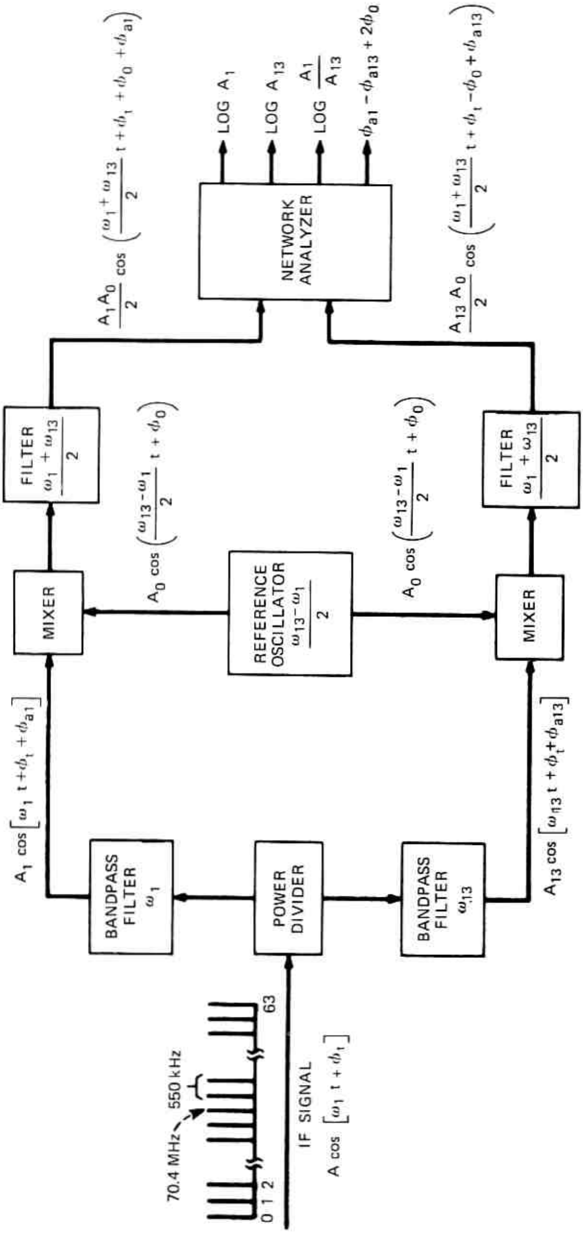

70.4 MHz
550 kHz
0 1 2
63
IF SIGNAL
$A \cos [\omega_1 t + \phi_t]$
POWER DIVIDER
BANDPASS FILTER ω_1
BANDPASS FILTER ω_{13}
$A_1 \cos [\omega_1 t + \phi_t + \phi_{a1}]$
$A_{13} \cos [\omega_{13} t + \phi_t + \phi_{a13}]$
MIXER
MIXER
REFERENCE OSCILLATOR $\frac{\omega_{13} - \omega_1}{2}$
$A_0 \cos \left(\frac{\omega_{13} - \omega_1}{2} t + \phi_0 \right)$
$A_0 \cos \left(\frac{\omega_{13} - \omega_1}{2} t + \phi_0 \right)$
FILTER $\frac{\omega_1 + \omega_{13}}{2}$
FILTER $\frac{\omega_1 + \omega_{13}}{2}$
$\frac{A_1 A_0}{2} \cos \left(\frac{\omega_1 + \omega_{13}}{2} t + \phi_t + \phi_0 + \phi_{a1} \right)$
$\frac{A_{13} A_0}{2} \cos \left(\frac{\omega_1 + \omega_{13}}{2} t + \phi_t - \phi_0 + \phi_{a13} \right)$
NETWORK ANALYZER
LOG A_1
LOG A_{13}
LOG $\frac{A_1}{A_{13}}$
$\phi_{a1} - \phi_{a13} + 2\phi_0$

Fig. 1—Basic schematic of two-channel phase and amplitude comparator.

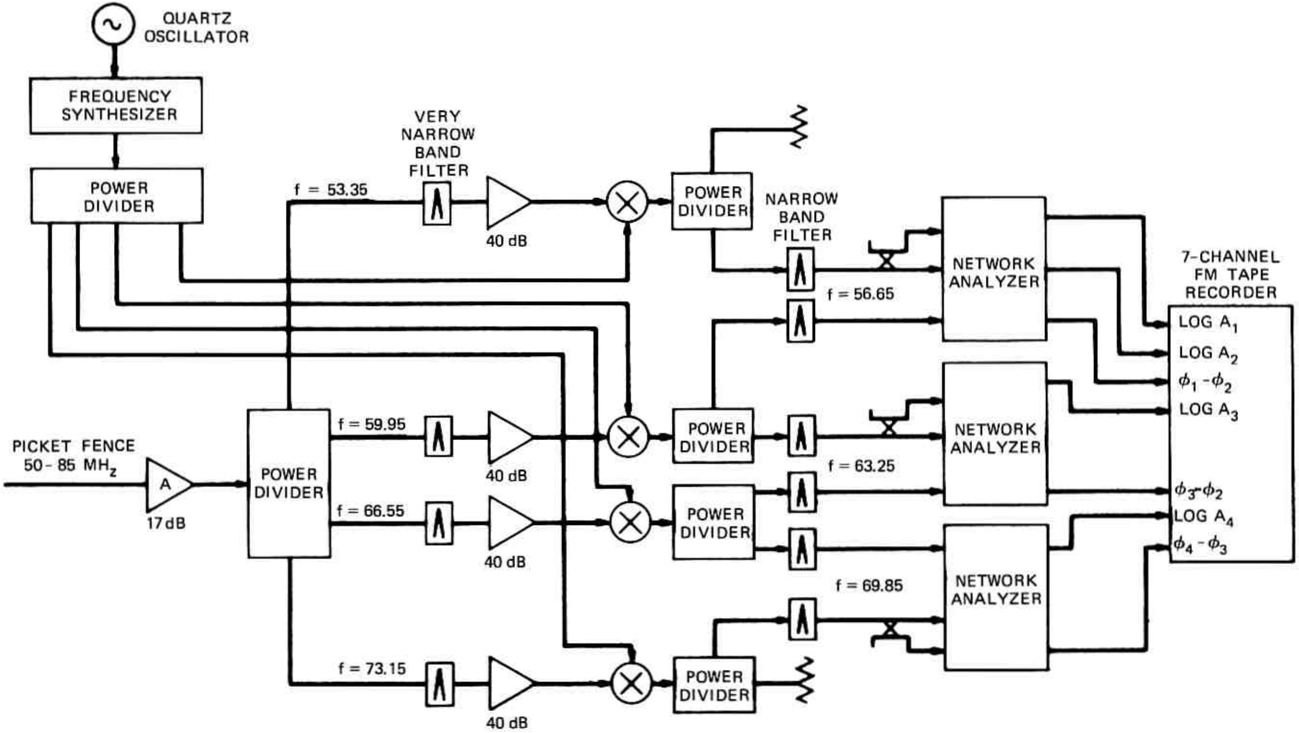


Fig. 2—Schematic of the system.

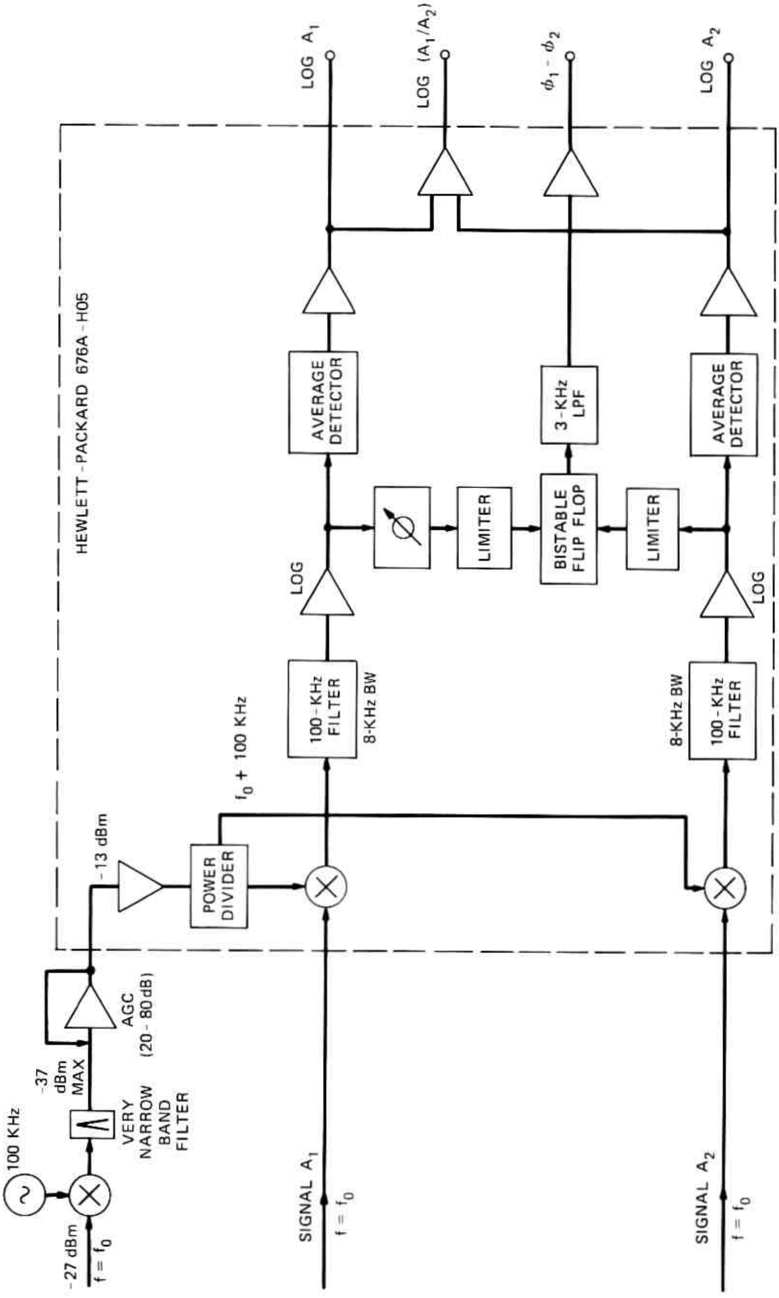


Fig. 3—Network analyzer.

amplitudes and three phase differences are recorded. Note that if one knows $\Phi_2 - \Phi_1$ and $\Phi_3 - \Phi_2$ one can compute $\Phi_3 - \Phi_1$, etc.

III. DISCUSSION OF EXPERIMENTAL TECHNIQUE

The experimental technique described here has several advantages and limitations compared with others used in the past.[3–5]

The distinguishing characteristic of this technique is that the actual phase difference between two signals is measured. Thus, if we consider $\Phi = \Phi(\omega)$, we measure $\Delta\Phi/\Delta\omega$ rather than simply $d^2\Phi/d\omega^2$, which is the delay distortion. Further, the tones are generated and measured at IF in such a way that the phase and frequency variations of the up and down converters do not affect the measurements. Hence, the same apparatus could be used at any desired carrier frequency. A knowledge of the envelope delay, $d\phi/d\omega$, could be of use in the analysis of multipath fading of line-of-sight microwave link.

The basic problem facing anyone wishing to measure phase differences of the same signal reaching two widely separated receivers, as in very-long-baseline interferometry (VLBI) or, as in our case, of two signals generated at one place and received at another, is one of a time reference. One technique is to transmit a timing signal either over the air or through a cable from one place to another. This approach suffers from unknown variations in the signal path because of atmospheric changes in the broadcast case or temperature variations in the cable case. Following the lead of the workers in VLBI, similar standard oscillators have been set up at the transmitter and receiver. These oscillators have sufficient short-term (<1 s) and long-term (<1 degree/day) stability to enable measurements to be made within the desired accuracy (0.03 degree rms for 6.6-MHz tone separation).

IV. CHARACTERISTICS OF FADES

This section describes the temporal behavior of the signal during fading. The data were recorded on analog FM tapes continuously from September 11 to December 31, 1970. The system was shut down on two occasions for several days for servicing. The system was also turned down a few times for several hours at a time for making tests. The tape was manually changed as it approached the end. Each tape lasted an average of five days. The tape was turned on automatically whenever any one tone exceeded 10-dB fade and ran till all the four tones recovered from deeper than 10-dB fade. The nature of the fades can be classified into the following broad categories.

(*i*) All the four tones in the 20-MHz band fade simultaneously, with the level of fade being approximately the same for all of them. In other words, there is no significant frequency selectivity present. Such fades last several minutes. These fades are usually relatively shallow (less than or about 20 dB).

(*ii*) The four tones fade selectively; that is, the fade levels of the four tones are significantly different. Such selective fades are usually deep (greater than about 20 dB) and last no more than a few seconds and often much less than one second. The dispersive fades are usually superimposed on shallow, nondispersive ones. The deepest fade level on the four tones may occur either simultaneously or separated in time by a small fraction of a second. In the former case the minimum amplitude in the band occurs at a single frequency in the band, whereas in the latter case the minimum traverses across the band in time.

There is no significant phase nonlinearity during the nondispersive fades mentioned in case (*i*) above. A temporal presentation of the amplitudes of the four tones and phase differences between them during a typical nondispersive fade is given in Fig. 4. For the sake of clarity, the traces are displaced from one another. The fade level of A_1 is with respect to its nominal unfaded value. The nominal unfaded levels of A_2, A_3, and A_4 are displaced by approximately 20, 40, and 60 dB, respectively. Similarly, the three phase difference traces are displaced from each other by about 50 degrees. The fade duration is about 1 minute, and the maximum fade depth is about 15 dB. There is no noticeable variation in the magnitudes of the difference between adjacent tones. Although the traces of the phase differences in Fig. 4 are nearly horizontal, they have, in general, a linear slope with time which is the same for all three. As mentioned in Section II, this is caused by the difference in the quartz oscillator frequencies between the one driving the picket-fence generator at the transmitter and that of the other in the phase/amplitude measuring system at the receiver. Any path length change is reflected on the three traces as a change from the linear slope and will be the same in all the three traces. However, when there is a phase nonlinearity present within the 20-MHz band, the three traces will have different slopes.

Figure 5 represents a dispersive fade of the type discussed in case (*ii*) above. The total duration of the fade is in excess of 5 minutes, and the maximum fade depth is 39 dB (on A_4). Once again, the time axis has been referenced with respect to that at which the deepest

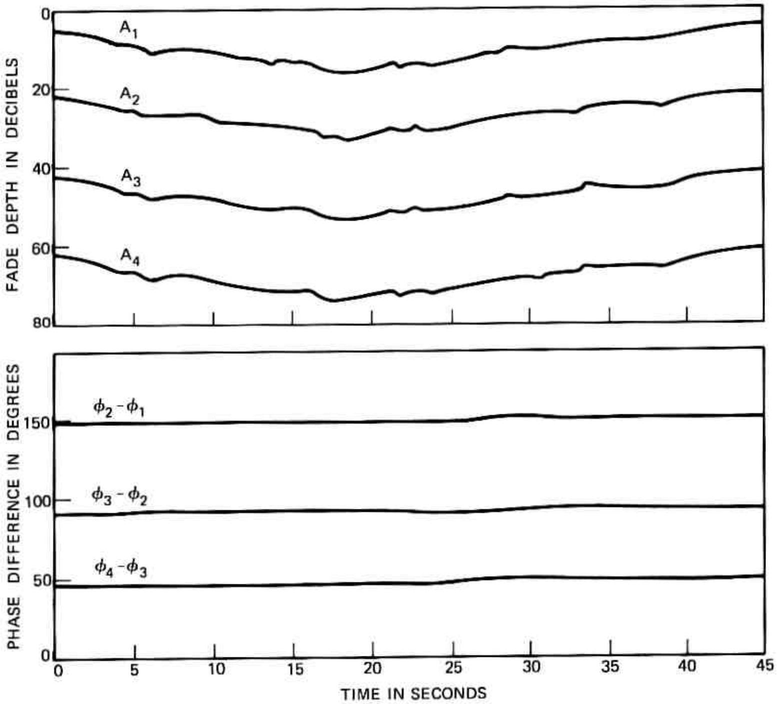


Fig. 4—Nondispersive fading.

fade occurs. It could be observed in this particular event that there are two deep fades separated by about 2 minutes superimposed on the prolonged shallow fades. It is further observed from the phase data that significant phase nonlinearity is present only during the deep fade periods that last only in the order of seconds. The magnitude of the phase nonlinearity is dependent on the depth of the fade and its amplitude dispersion. Thus, while the first fade in Fig. 5 exhibits significant phase nonlinearity, the second does not.

A time-expanded representation of the first fade in Fig. 5 is given in Fig. 6. Large selective fading is exhibited during this event for an extended period of time. During the few seconds around the deepest fade period, the fade dispersion curve changes slope and the high-frequency end of the band (A_4) fades deeper than the rest. The phase nonlinearity is exhibited predominantly during this period of deepest fade. Figure 6 has been redrawn in Fig. 7 by interchanging the role of time and frequency parameters. The dispersion with respect to fre-

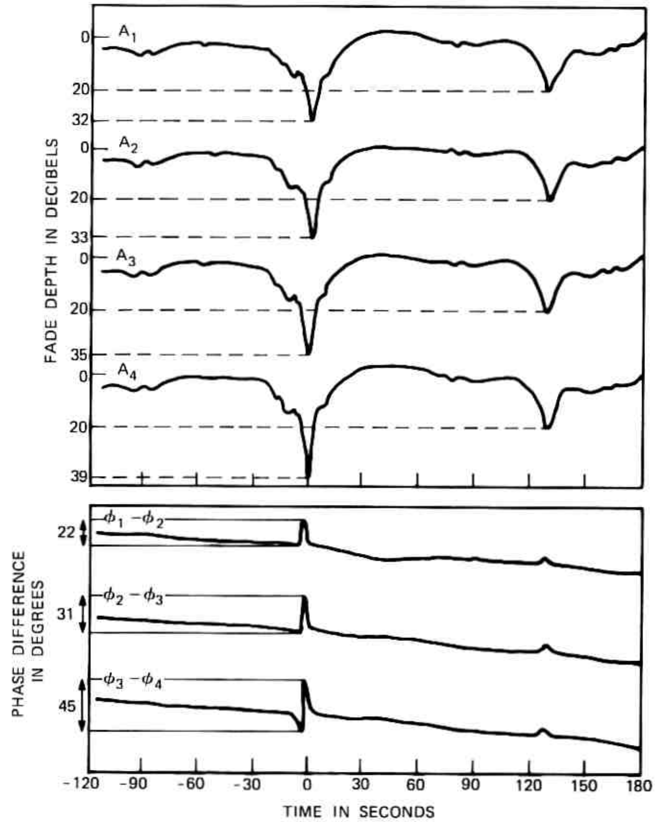


Fig. 5—Dispersive fading.

quency of the fade depth and the phase difference are plotted as discrete functions of time. This presentation visually portrays the nonlinearities as greatest around zero time. We can also observe that the phase difference curve changes from a convex to a concave shape between 0 and 0.2 second. Thus, the phase nonlinearity could assume zero value around the time of deepest fade.

The relationship between the amplitude distortion and phase nonlinearity was explored by fitting a second-order polynomial curve over the log-amplitude and phase dispersion curves and then correlating the coefficients of the quadratic terms.

The following second-order polynomial expansion was assumed for log-amplitude dispersion $\chi(f)$ (in dB) and phase dispersion $\phi(f)$ (in

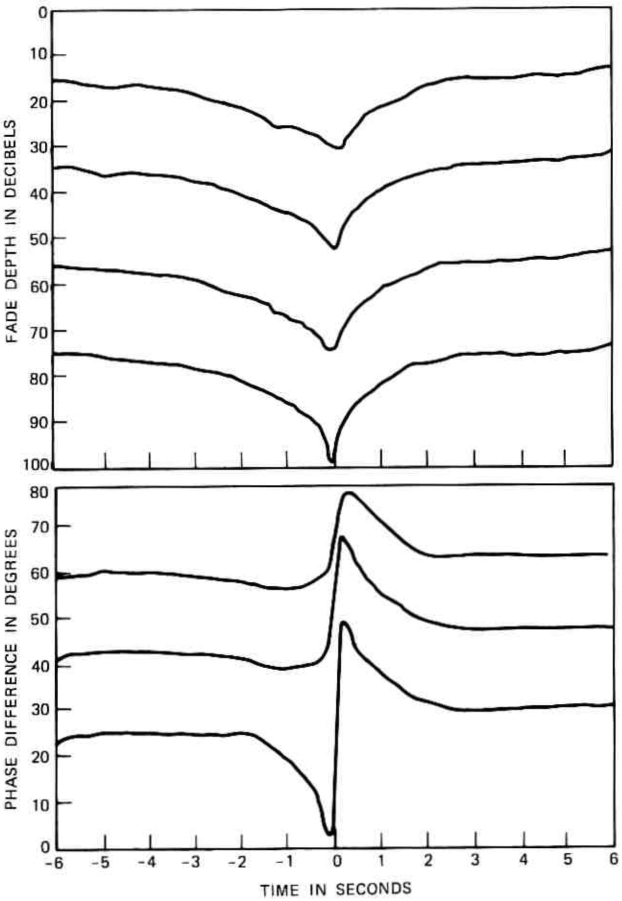


Fig. 6—Dispersive fading.

degrees).

$$\chi(f) = a_\chi(f_3) + b_\chi(f - f_3) + c_\chi(f - f_3)^2 \tag{11}$$

$$\phi(f) = a_\phi(f_3) + b_\phi(f - f_3) + c_\phi(f - f_3)^2. \tag{12}$$

The temporal behavior of the quadratic log-amplitude and phase nonlinear coefficients designated by c_χ and c_ϕ, respectively, are given in Fig. 8. There appears to be some degree of correlation between these coefficients except around zero time reference. As explained in the description of Fig. 7, the reason for the deviation around zero time is the change in the convexity of the phase dispersion curve.

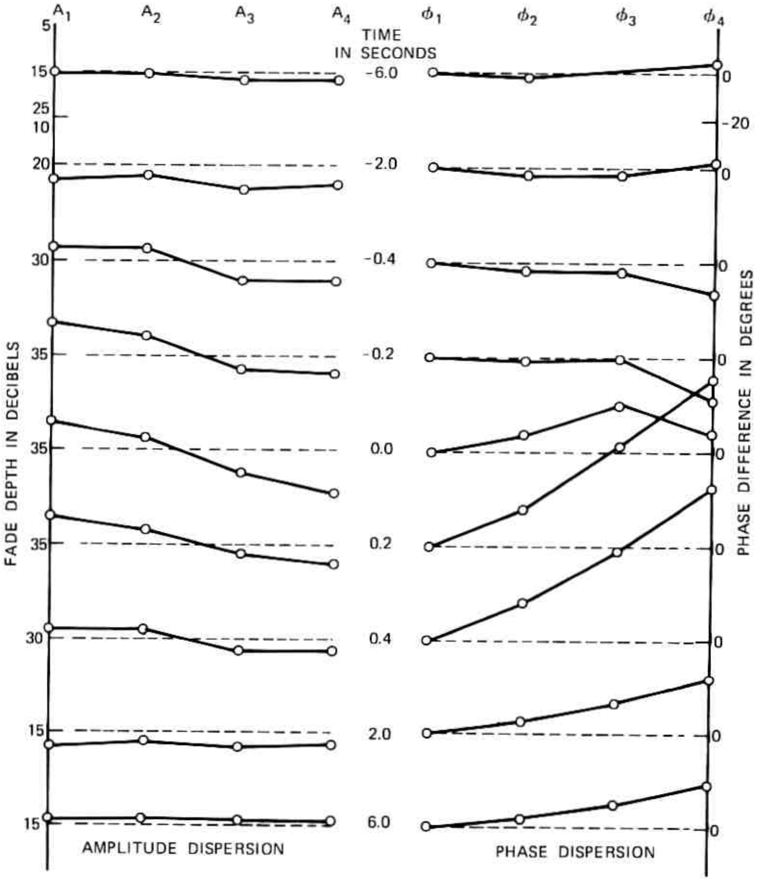


Fig. 7—Amplitude and phase dispersion during dispersive fade.

V. DATA ANALYSIS

The amplitude and phase dispersion data have been fitted with second-order polynomial curves, and the nonlinear coefficients of the two have been studied. Specifically, results on the following have been obtained.

(*i*) The distribution curve describing the fraction of time that the phase nonlinearity exceeds a given value.

(*ii*) The average of the magnitude of the phase nonlinearity as a function of the depth of the fade.

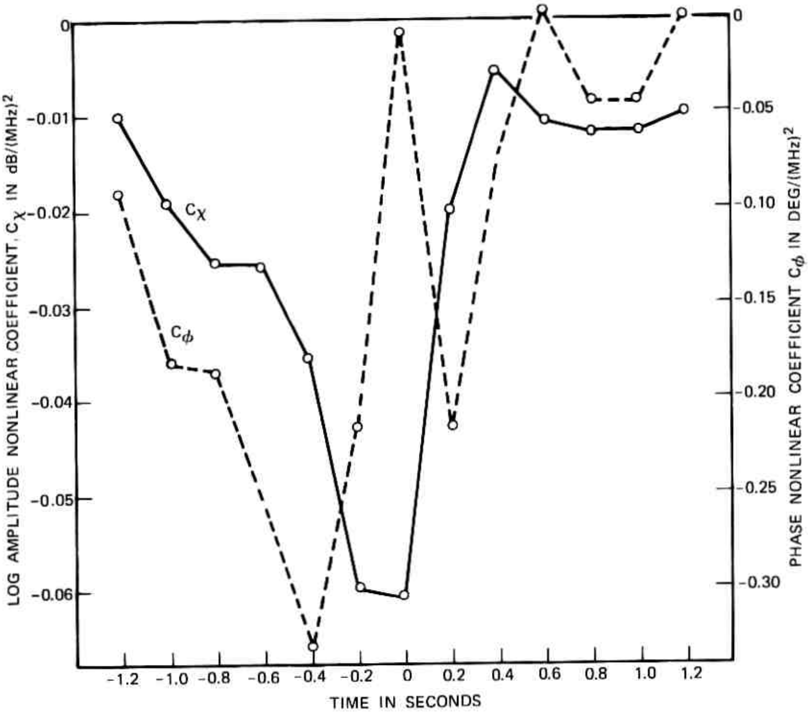


Fig. 8—Temporal behavior of log-amplitude and phase nonlinearities (c_χ, c_ϕ).

(*iii*) The nature of the correlation between log-amplitude and phase nonlinear coefficients.

Letting $f_3 = 0$ and $f - f_3 = \Delta f$ in eqs. (11) and (12), we obtain

$$\chi(\Delta f) = a_\chi(0) + b_\chi(\Delta f) + c_\chi(\Delta f)^2 \tag{13}$$

$$\phi(\Delta f) = a_\phi(0) + b_\phi(\Delta f) + c_\phi(\Delta f)^2. \tag{14}$$

The data were taken on four tones, at $\Delta f = -$ 13.2 MHz, -6.6 MHz, 0, and $+6.6$ MHz. For the sake of convenience in the analysis, the frequency separation was normalized with respect to $\Delta f = 13.2$ MHz. Thus, the four data points corresponding to eq. (14) were designated by the following set of simultaneous equations.

$$\begin{aligned} \phi(-1) &= \tilde{a}_\phi(0) + \tilde{b}_\phi(-1) + \tilde{c}_\phi(-1)^2 \\ \phi(-\tfrac{1}{2}) &= \tilde{a}_\phi(0) + \tilde{b}_\phi(-\tfrac{1}{2}) + \tilde{c}_\phi(-\tfrac{1}{2})^2 \\ \phi(0) &= \tilde{a}_\phi(0) \\ \phi(+\tfrac{1}{2}) &= \tilde{a}_\phi(0) + \tilde{b}_\phi(\tfrac{1}{2}) + \tilde{c}_\phi(+\tfrac{1}{2})^2. \end{aligned} \tag{15}$$

Comparing the coefficients of eqs. (14) and (15), we obtain

$$\begin{aligned} a_\phi &= \tilde{a}_\phi \\ b_\phi &= \tilde{b}_\phi \left(\frac{1}{2 \times 6.6 \times 10^6} \right) \\ c_\phi &= \tilde{c}_\phi \left(\frac{1}{2 \times 6.6 \times 10^6} \right)^2 . \end{aligned} \tag{16}$$

A similar set of equations can be obtained for the log-amplitude dispersion, yielding

$$\begin{aligned} a_\chi &= \tilde{a}_\chi \\ b_\chi &= \tilde{b}_\chi \left(\frac{1}{2 \times 6.6 \times 10^6} \right) \\ c_\chi &= \tilde{c}_\chi \left(\frac{1}{2 \times 6.6 \times 10^6} \right)^2 . \end{aligned} \tag{17}$$

The measured log-amplitude and phase dispersion data of each event was digitized at a sampling period of 0.2 second. Equations (15) contain three unknowns ($\tilde{a}_\phi$, $\tilde{b}_\phi$, and $\tilde{c}_\phi$) and four equations. The four data points for ϕ were obtained by arbitrarily setting $\phi_1 = 0$ and then computing ϕ_2, ϕ_3, and ϕ_4 by the measured differences. These four data points were then fitted with a smooth curve with the least-square-fit criterion, and c_ϕ was computed for the fitted curve. Note that this procedure does not affect the computational result on c_ϕ. Further, it will not cause any serious error as discussed by Babler,[1] as in the case of fitting a third-order polynomial curve that passes through all four data points. The log-amplitude coefficient c_χ was obtained by fitting a second-order polynomial curve through the four data points, χ_1, χ_2, χ_3, and χ_4, which are directly available from measurements. The coefficients c_ϕ and c_χ were then computed using eqs. (16) and (17).

The quadratic phase nonlinearity can easily be related to the more familiar parameter of delay distortion. The delay distortion, denoted by $\tau(\omega)$, is defined as the departure of the envelope delay, $D(\omega)$, from a constant value.[6] The envelope delay is given by[6]

$$D(\omega) = \frac{d\phi}{d\omega} . \tag{18}$$

From eqs. (12) and (18), we obtain

$$\tau(\Delta\omega) = \frac{c_\phi}{180 \times 10^{+6}} (\Delta f), \tag{19}$$

where $\tau(\Delta\omega)$ is the delay distortion in s/MHz, c_ϕ is expressed in degrees/(MHz)^2, and Δf is expressed in MHz.

VI. RESULTS

Numerous fading events were recorded exceeding 10-dB fade. Of these, 26 events faded deeper than 20 dB and were selected for data analyses. This selection was based on the fact that only these display phase distortion and consequently are of interest here. The first selected event occurred during the period of September 16 to 18 (recorded in the tape that ran during this period). The twenty-sixth event occurred during the period of November 20 to 23. No event exceeded 20-dB fade between November 23 and December 31. Babler[1] has observed, in the amplitude dispersion experiment run during approximately the same period in the same microwave link, 40 counts of the tone at the center of the band dipping below 20 dB. In the present analysis, the interval of an event is defined beginning from when at least one tone exceeds a fade depth of 10 dB and lasting until all the tones have recovered above 10-dB fade level. Thus, an event of ours could include more than one count of Babler's experiment. Considering the loss of time because of system shut-downs and tape run-outs mentioned in the previous section, the number of 20-dB fade events in the two experiments agree with each other satisfactorily. Note that the number of fading events during the 1970 autumn season appears to be significantly below normal. Unfortunately, not all the events could be used in the analyses because of failure in part of the instrumentation. Data of 16 events were used in connection with the analysis of phase nonlinearity [items (*i*) and (*ii*) mentioned in the previous section], and the correlation between log-amplitude and phase nonlinearities [item (*iii*)] was made for 14 events.

6.1 *Distribution of Phase Nonlinearity*

The distribution of the phase nonlinear coefficient c_ϕ for the pooled data of 16 events (consisting of 2284 samples) is shown in Fig. 9. Three distribution curves represent the positive, the negative, and the absolute values of c_ϕ. Observe in Fig. 9 that 70 percent of the samples yielded positive values for c_ϕ and the remaining 30 percent were negative. No attempt was made to study the distributions of the individual events, since the sample size was considered inadequate. The distribution of $|c_\phi|$ is presented on a log-normal graph in Fig. 10. Observe that the smooth curve fitted over the data points is "close"

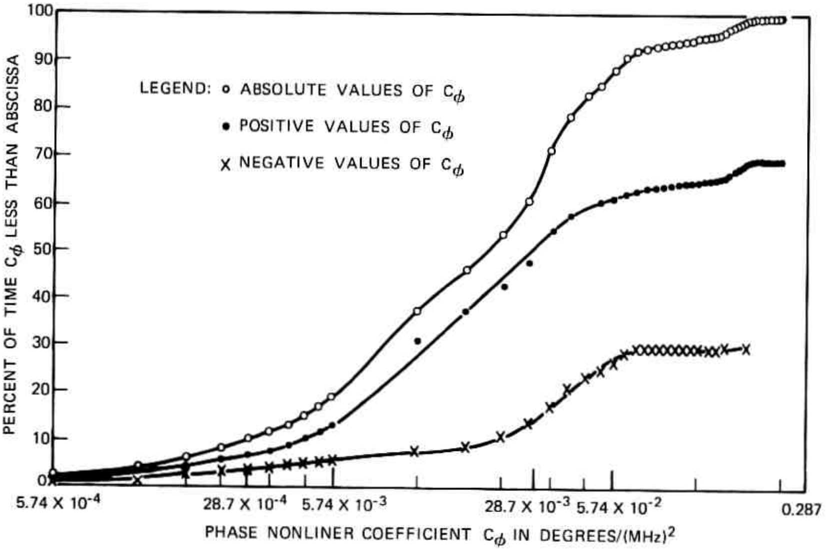


Fig. 9—Distribution curves for c_ϕ.

to a straight line. Hence, it appears that the distribution of the phase nonlinearity is close to log-normal. This conclusion is substantiated more quantitatively in the appendix.

6.2 *Dependence of Phase Nonlinearity on Fade Depth*

Figure 11 shows the dependence of phase nonlinearity on the depth of fade. The data points represent the magnitude of quadratic nonlinear coefficient $|c_\phi|$ as each event reached the fade depths of 10, 15, 20, 25, 30, 35, and 40 dB as well as when they come out of the fade. This approach of plotting the data points was adopted over that of recording the value of $|c_\phi|$ at the maximum level of the fade for each event for the following two reasons. First, the sample size is larger. Second, and more important, as explained in Section IV, $|c_\phi|$ could momentarily assume a zero value at the peak of the fade because of its changing sign, and thus could lead to an erroneous result. The smooth curve represents the average value of $|c_\phi|$ as a function of the fade depth. We see that $|c_\phi|$ increases with fade depth beyond 20-dB fade. The magnitude of $|c_\phi|$ remains constant at 0.02 degree/(MHz)^2 below 20 dB, which is due to the limitation in the accuracy of equipment and the variance of $|c_\phi|$ in curve fitting. ($|c_\phi|$ should eventually go to zero at 0-dB fade.)

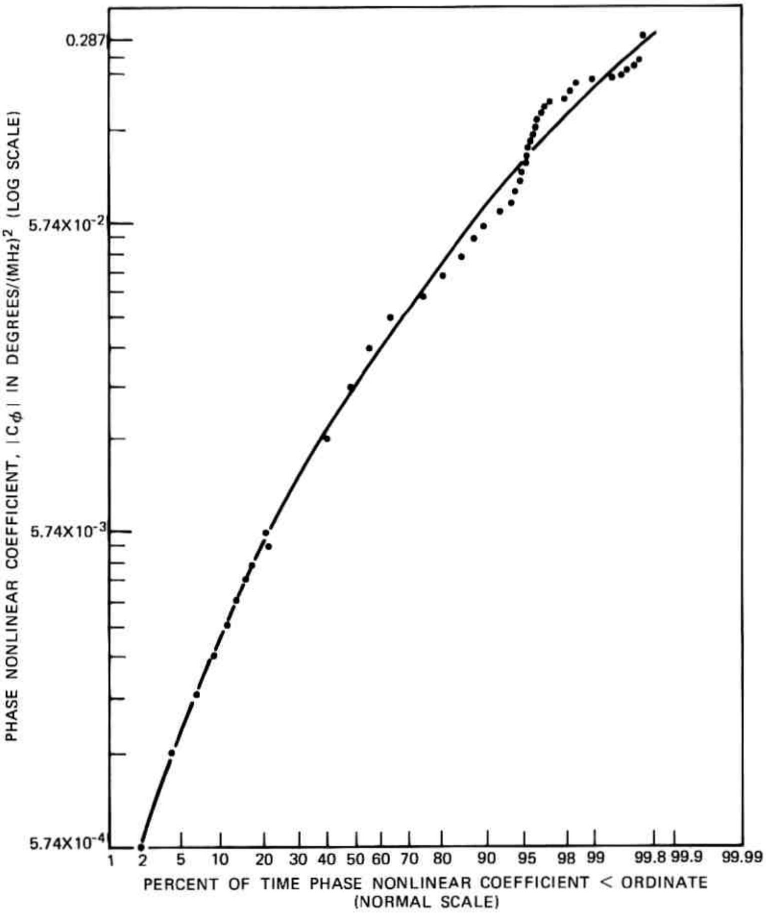


Fig. 10—Distribution curve for $|c_\phi|$: log-normal plot.

Also presented in the ordinate scale is the delay distortion $\tau(\Delta\omega)$ in eq. (19), in seconds over a 1-MHz band. Observe that corresponding to $|c_\phi| = 0.1$ degree/(MHz)^2 and $\Delta\omega = 2\pi$ the delay distortion is calculated to be 0.55 nanosecond, and this occurs at a fade depth of about 34 dB. This reasoning does not take into account the phase dispersion ripples being present between the tones (i.e., frequency separation of less than 6.6 MHz). Babler[1] and Ho[7] have reported the presence of such ripples (some of them of significant magnitude) superposed on an overall smooth amplitude dispersion curve in a 20-MHz band. This fine structure is currently being investigated in a

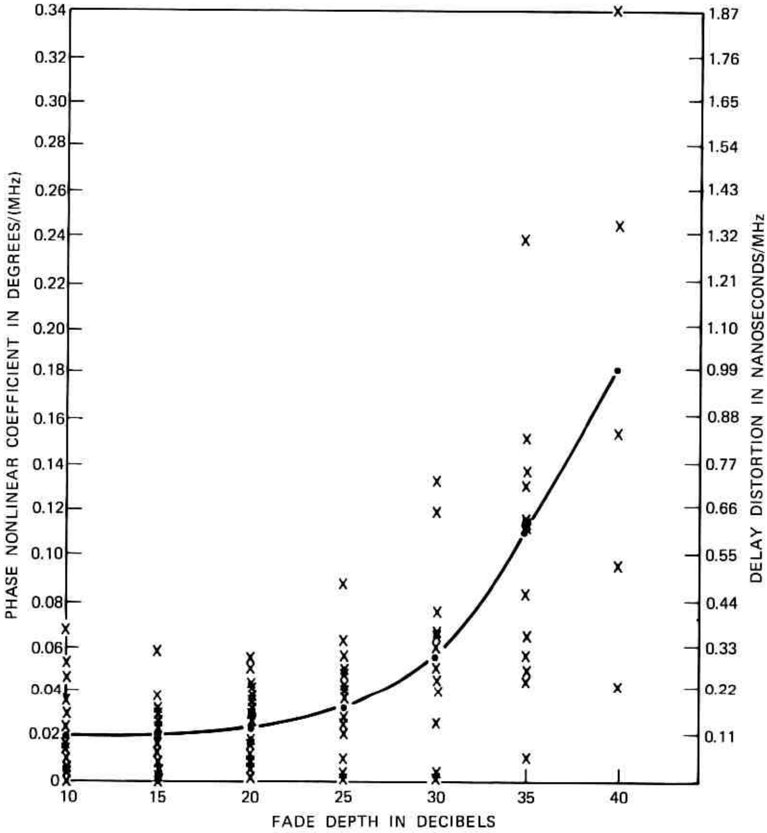


Fig. 11—Dependence of phase nonlinearity, c_ϕ, on fade depth.

more comprehensive study being conducted at Bell Laboratories by choosing tones that are closer than 6.6-MHz separation.

6.3 *Correlation Between Log-Amplitude and Phase Nonlinearities*

The correlation between the log-amplitude nonlinearity $\tilde{c}_\chi$ and phase nonlinearity $\tilde{c}_\phi$ was computed for each event. Table I shows the results for 14 events. The correlation coefficient was calculated for each event using digitized data points for that part of the fade below the 10-, 15-, 20-, and 25-dB levels in order to establish the dependence of the correlation coefficient R on fade level. The results shown in Table I do not easily lend themselves toward making any firm conclusions about the physical model that would yield such data, yet they are presented here

Table I—Correlation Between Log-Amplitude and Phase Nonlinearities

Event	$R(\tilde{c}_\chi; \tilde{c}_\phi)$			
	10 dB	15 dB	20 dB	25 dB
70-3-1	+0.58	+0.58	+0.68	+0.77
70-3-2	−0.38	−0.40	−0.71	—
70-6-1	+0.80	+0.80	+0.60	+0.36
70-6-2	+0.68	+0.68	+0.70	+0.70
70-6-4	+0.50	+0.43	—	—
70-6-5	+0.49	+0.49	−0.70	—
70-6-6	+0.36	+0.39	+0.45	−0.48
70-6-8	+0.68	+0.93	—	—
70-13-1	−0.74	−0.74	—	—
70-16-1	−0.65	−0.67	−0.70	−0.71
70-16-2B	−0.39	−0.62	−0.74	—
70-18-1A	−0.65	−0.64	−0.71	−0.72
70-18-1B	−0.97	−0.97	−0.98	−0.99
70-23-1	−0.42	−0.42	−0.48	−0.48

to give an idea of the complexity of the problem. The following observations can be made on the results.

(*i*) The positive and negative coefficients are equally probable.

(*ii*) The magnitude of the correlation coefficient, in general, increases as the fade-depth increases.

(*iii*) The correlation coefficient, in some cases, changes sign near the peak of the fading.

(*iv*) It appears difficult to predict the behavior of the phase nonlinear coefficient from a knowledge of the amplitude nonlinear coefficient. One primary aim in pursuing this course of analysis was to investigate whether a deterministic relationship could be established between the two coefficients. If this were possible, it would have helped a communication system designer to build an automatic phase compensator that could be controlled by measuring the amplitude dispersion, measurement of the amplitude dispersion being far simpler than that of the phase dispersion.

VII. SUMMARY

Measurement of phase and amplitude dispersion have been made over a 20-MHz band at 6 GHz on a 42-km line-of-sight radio link. The present experimental technique is unique in that the direct phase dispersion, instead of delay distortion, has been measured using the scheme of very-long-base interferometry. The base delay between adjacent pairs of four tones equally spaced over a 19.8-MHz band was measured. Twenty-six events were recorded during the autumn season of 1970 that produced phase distortion of greater than 0.02 degree/(MHz)^2. All these events had fades which exceeded 20 dB below the nominal level. Although the number of events observed appears to be considerably less than normal, it was considered adequate to derive some preliminary statistics about the phase dispersion characteristics during deep dispersive microwave fading.

The quadratic phase nonlinear coefficient which is linearly proportional to the delay distortion is observed to have a log-normal distribution. To the best of the authors' knowledge, this is the first time that statistics have been obtained on such phase characteristics. These results may be useful in the formulation of a statistical model of microwave fading.

The quadratic phase nonlinear coefficient and hence the delay distortion increase with the depth of fade. On the average, fades deeper than 34 dB below nominal level cause a delay distortion in excess of 0.55 nanosecond/MHz. For a monochrome television signal with about 4-MHz bandwidth, the delay distortion permitted is about 25 nanoseconds.[6] If the same bandwidth is assumed at the radio frequency band as would be if it were an amplitude-modulated system, the average delay distortion for a 40-dB fade would be, from Fig. 11, about 4 nanoseconds, which would be well within the tolerance. Over the measured 20-MHz band, the delay distortion at 40-dB fade would be about 20 nanoseconds.

During the deep dispersive fades, the nonlinear phase and amplitude coefficients were found to have nonzero correlation which could be either positive or negative. No simple relationship seems apparent between the two coefficients. Although simple two-ray models can be made to account for both amplitude and phase dispersion,[7] the complex temporal behavior indicated in our results along with that reported by Babler[1] lead us to believe that the multiray (more than two rays) phenomenon is the cause of these deep fades.

VIII. ACKNOWLEDGMENTS

The authors wish to recognize the assistance of D. E. Kreichbaum in the data reduction and A. M. Franz in the data processing. W. T. Barnett's comments concerning the improved presentation of the results is gratefully acknowledged.

APPENDIX

To determine the nature of the distribution curve for the absolute value of c_ϕ, an entirely empirical approach to the problem was adopted. Plotting the raw data of the phase nonlinear coefficient $|\tilde{c}_\phi|$ on a log-normal graph paper indicated that the distribution was close to log-normal. In the literature[8] a powerful technique exists which transforms, in most cases, a given distribution to a normal distribution by a suitable linear transformation. Use of this technique was expected to yield information on how closely the observed set of data approximated a log-normal distribution. Following Box and Cox[8] who have treated such an analysis of transformation, the following transformation was chosen.

$$\tilde{c}_{\phi t}(\lambda) = \begin{cases} \dfrac{\tilde{c}_\phi^\lambda - 1}{\lambda} & (\lambda \neq 0) \\ \\ \log \tilde{c}_\phi & (\lambda = 0). \end{cases} \tag{20}$$

Here, λ is the parameter that defines the transformation. (For the sake of convenience, the absolute value signs have been omitted.) In accordance with our assumption, a value exists for λ such that $\tilde{c}_{\phi t}(\lambda)$ is normally distributed. The value of λ can be determined using the maximum likelihood theory. The method of maximum likelihood involves maximizing the log-likelihood estimate $\mathfrak{L}_{\max}$ with respect to the unknown parameters of μ, σ^2, and λ, where μ and σ^2 are the mean and variance of the transformed data $\tilde{c}_{\phi t}(\hat{\lambda})$. The maximum likelihood estimate of λ, denoted by $\hat{\lambda}$, yields the best possible estimate that would make the transformed variable $\tilde{c}_{\phi t}$ closest to a normal distribution. The maximum log-likelihood function is given by

$$\mathfrak{L}_{\max}(\lambda/\tilde{c}_{1\phi}, \tilde{c}_{2\phi}, \cdots, \tilde{c}_{n\phi}) = -\frac{n}{2}\log\sigma^2 + (\lambda - 1)\sum_{i=1}^{n}\log\tilde{c}_{i\phi}, \tag{21}$$

where n is the number of data points.

The value of λ varied between -1 and $+1$ in steps of 0.1. The maximum value of $\mathfrak{L}_{\max}(\lambda)$ as a function of λ occurs at $\lambda = -0.1$. Thus, the maximum likelihood estimate for λ, $\hat{\lambda}$ is -0.1. Figure 12 shows the data points of $\tilde{C}_{\phi t}(\hat{\lambda})$ on a log-normal graph. The straight line through the data points corresponds to $\lambda = 0$. It is seen from eq. (20) that $\lambda = 0$ yields a log-normal distribution. Except at the large value of nonlinearity, the distribution appears log-normal. This is further justified by the fact that $\hat{\lambda} = -0.1$ is statistically quite

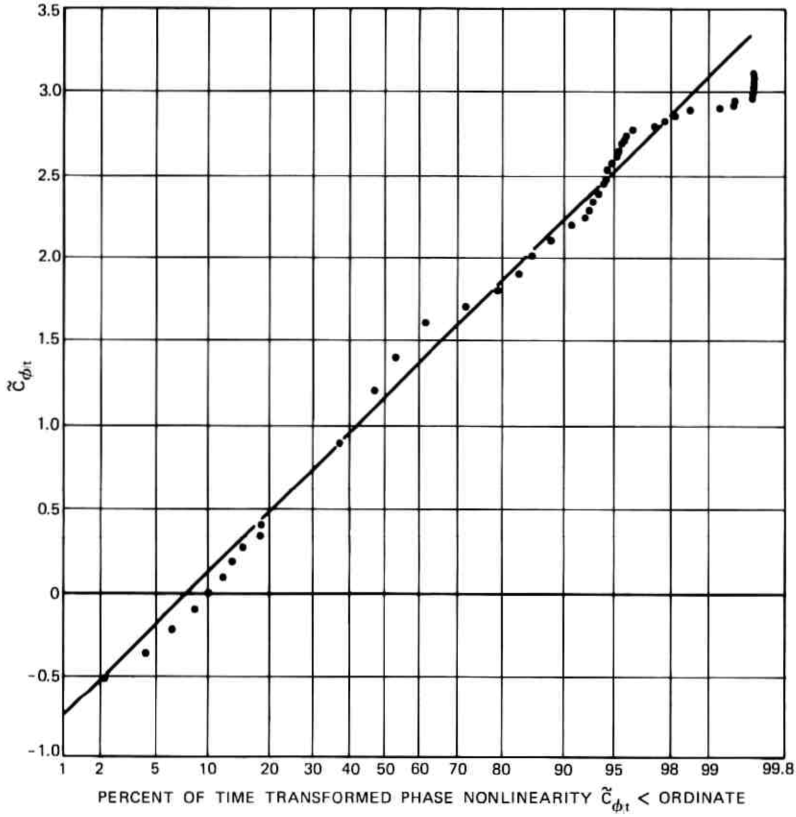


Fig. 12—Distribution of $\tilde{c}_{\phi t}(\hat{x})$: log-normal plot: $\hat{\lambda} = -0.1$.

close to zero. The deviation at the high end can be largely attributed to poor sample size.

REFERENCES

1. Babler, G. M., "A Study of Frequency Selective Fading for a Microwave Line-of-Sight Narrowband Radio Channel," B.S.T.J., *51*, No. 3 (March 1972), pp. 731–757.
2. Kaylor, R. L., "A Statistical Study of Selective Fading of Super-High Frequency Radio Signals," B.S.T.J., *32*, No. 5 (September 1953), pp. 1187–1202.
3. Thompson, M. C., Jr., and Vetter, M. J., "Single Path Phase Measuring System for Three Centimeter Radio Waves," Rev. Sci. Instr., *29*, No. 2 (February 1958), pp. 148–150.
4. Thompson, M. C., Jr., and Janes, H. B., "Measurements of Phase Stability Over a Low-Level Tropospheric Path," J. Res. NBS, *63P*, July-August 1959, pp. 45–51.

5. Janes, H. B., "Correlation of the Phase of Microwave Signals on the Same Line-of-Sight Path at Different Frequencies," IEEE Trans. Ant. Prop., November 1963, pp. 716–717.
6. Bell Telephone Laboratories, Incorporated, *Transmission Systems for Communication Engineers,* Fourth Edition (February 1970), p. 721.
7. Ho, T. L., private communication.
8. Box, A. E. P., and Cox, D. R., "An Analysis Transformation," Royal Statistical Society, Series B, *26*, No. 2 (1964), pp. 211–252.

Contributors to This Issue

David T. Chai, B.S.E.E., 1959, Purdue University; M.S. and Ph.D. in Communication Science, 1961, 1967, University of Michigan; IBM Development Laboratory, 1959–1960; Bunker-Ramo Corporation, 1967–1969; Bell Laboratories, 1969—. Mr. Chai has worked on the language design for interactive data management systems. He was responsible for the design and implementation of the CPAS (Construction Program Administrative System) Query System along with BISP. He was principally responsible for the retrieval part of the SNS70 (Switching Network Survey of 1970) System. He is now a supervisor of the Quality Information Systems Group of the Quality Assurance Center. Member, Eta Kapp Nu, Tau Beta Pi, Association for Computing Machinery, American Linguistic Society.

David D. Falconer, B.A.Sc., 1962, University of Toronto; S.M., 1963, and Ph.D., 1967, Massachusetts Institute of Technology; postdoctoral research, Royal Institute of Technology, Stockholm, 1966–67; Bell Laboratories, 1967—. Mr. Falconer has worked on problems in coding theory, communication theory, channel characterization, and high-speed data communication. Member, Tau Beta Pi, Sigma Xi, IEEE.

Gerard J. Foschini, B.S.E.E., 1961, Newark College of Engineering; M.E.E., 1963, New York University; Ph.D. (Mathematics), 1967, Stevens Institute of Technology; Bell Laboratories, 1961—. Mr. Foschini initially worked on real-time program design. Since 1965 he has mainly been engaged in analytical work concerning the transmission of signals. Currently he is working in the area of data communication theory. Member, Sigma Xi, Mathematical Association of America, American Men of Science, New York Academy of Sciences.

Thomas A. Gibson, B.S.(E.E.), 1960, Kansas State University; Bell Laboratories, 1960—. Mr. Gibson has worked on network, control, and traffic aspects of No. 1 ESS. He also participated in studies of small electronic switching offices. He is now engaged in the study and design of time-shared computing techniques for providing information management capabilities. Member, Eta Kappa Nu, Tau Beta Pi, ACM.

LEE E. HEINDEL, B.S., 1965, The Pennsylvania State University; M.S., 1967, and Ph.D., 1970, University of Wisconsin; Bell Laboratories, 1970—. Mr. Heindel is engaged in various areas of computer science research, notably algebraic algorithms and information management systems, and is currently co-authoring a book on interactive language design systems. He is a visiting lecturer at Stevens Institute of Technology. Member, Association for Computing Machinery, Sigma Xi, Pi Mu Epsilon.

D. M. HENDERSON, B.A. (Physics), 1962, M.S. (Physics), 1964, Miami University; Ph.D. (Engineering and Applied Science), 1969, Yale University; Bell Laboratories, 1969—. Upon joining Bell Laboratories, Mr. Henderson was involved in optical communication at 10.6 μm. Presently he is engaged in optical fiber communication. Member, American Physical Society, IEEE, Phi Beta Kappa.

DAVID G. LEEPER, B.S.E.E., 1969, Washington University; M. Eng. (Electrical), 1970, Cornell University; Bell Laboratories, 1969—. Mr. Leeper has worked on problems associated with digital transmission. Member, IEEE, Eta Kappa Nu, Tau Beta Pi.

JAMES MCKENNA, B.Sc. (Mathematics), 1951, Massachusetts Institute of Technology; Ph.D. (Mathematics), 1961, Princeton University; Bell Laboratories, 1960—. Mr. McKenna has done research in quantum mechanics, electromagnetic theory, and statistical mechanics. He has recently been engaged in the study of nonlinear partial differential equations that arise in solid-state-device work and in the theory of stochastic differential equations.

KEVIN C. O'BRIEN, B.S. (Physics), 1963, Boston College; Ph.D. (Physics), 1969, Brown University; Bell Laboratories, 1969—. Mr. O'Brien's work at Bell Laboratories has included antenna design, microwave propagation on earth-space as well as tropospheric paths, digital carrier maintenance, and currently the development of an automatic voice response system. He is supervisor of the Maintenance Information Processing Group. Member, Sigma Pi Sigma, Sigma Xi, IEEE.

BRUCE W. PUERLING, B. A. (Math), 1966, Franklin and Marshall College; M.A. (Math), 1967, University of Michigan; Bell Laboratories, 1966—. Mr. Puerling has been engaged in the design and implementation of interactive computer applications including engineering aids, data base query languages, data base management systems, and tools for the design of interactive applications. Member, Association for Computing Machinery, Pi Mu Epsilon.

PHILIP J. PUGLIS, JR., Electronics Technology, 1967, RCA Institutes; B.S.E.E., 1972, Polytechnic Institute of Brooklyn; M.S.E.E., 1973, Stanford University; Bell Laboratories, 1967—. From 1967 to 1972, Mr. Puglis was involved in various antenna and propagation studies which included laser simulation of microwave relay links aimed at studying the fading phenomena. He has recently completed the OYOC (one year on campus) assignment, 1972–1973 at Stanford University. He is currently a member of the Transmission Measurement Circuits Department.

JERRY T. ROBERTO, B.S.(E.E.), 1965, Newark College of Engineering; M.S.(Computer Science), 1966, Harvard University; Bell Laboratories, 1965—. Mr. Roberto has been involved in electronic switching systems and development of interactive management information systems. He is currently co-authoring a book on interactive language design systems. Member, Eta Kappa Nu, Tau Beta Pi, Association for Computing Machinery.

JACOB ROOTENBERG, B.S. (E.E.), 1960, M.S. (E.E.), 1962, D.Sc. (E.E.), 1967, Technion-Israel Institute of Technology. Mr. Rootenberg joined Columbia University in 1967 where he is currently an Associate Professor of Electrical Engineering and Computer Science. His fields of interest include nonlinear and optimal control systems, simulation, and power processing. Since 1971 he has been teaching a series of courses in control systems theory as part of the In-Hours Education Program at Bell Laboratories. Member, IEEE, SIAM.

N. L. SCHRYER, B.S., 1965, M.S., 1966, and Ph.D., 1969, University of Michigan; Bell Laboratories, 1969—. Mr. Schryer has worked on the numerical solution of parabolic and elliptic partial differential equations. He is currently studying problems of this type which arise in semiconductor device theory.

PETER F. STOCKHAUSEN, B.S.(Math), 1969, Marquette University; M.S.(Applied Math), 1971, Stevens Institute of Technology; Wisconsin Telephone Company, 1966–1969, 1973, Data Processing Systems Support; Bell Laboratories, 1969–1973. Mr. Stockhausen has been engaged in the design and implementation of computer-supported information management systems. Member, Pi Mu Epsilon.

M. Subramanian, B.S., 1953, Madras University (India); Dip., 1956, Madras Inst. Tech. (India); M.S.E.E., 1961, and Ph.D., 1964, Purdue University; Bell Laboratories, 1966—. Mr. Subramanian's earlier research included work on receivers, parametric amplifiers, and ferroelectric materials in the microwave region as well as nonlinear optics and cathodoluminescence in the quantum electronics field. His experience in the field of electromagnetic wave propagation includes studies on characterization of the atmosphere and its effects on optical and microwave propagation. He is currently involved with digital carrier maintenance. Member, IEEE, Eta Kappa Nu, Sigma Pi Sigma, Sigma Xi.

K. K. Thornber, B.S., 1963, M.S. (E.E.), 1964, Ph.D. (E.E.), 1966, California Institute of Technology; Research Associate, Stanford Electronics Laboratories, 1966–68; Research Assistant, Physics Department, University of Bristol, 1968–69; Bell Laboratories, 1969—. Mr. Thornber is a member of the Unipolar Integrated Circuit Laboratory. Member, Sigma Xi, Tau Beta Pi.

Ralph Walk, B.S.E.E., 1966, Tufts University; M.S.E.E., 1968, Columbia University; Bell Laboratories, 1966—. Mr. Walk has worked on the design of power conditioning circuitry, using ferroresonant devices. Member, IEEE, Sigma Xi, Tau Beta Pi, Eta Kappa Nu.

Joseph M. Wier, B.S.(E.E.), 1949, and M.S.(E.E.), 1950, Iowa State University; Ph.D.(E.E.), 1956, University of Illinois; Bell Laboratories, 1956—. Mr. Wier has worked on problems in data transmission, data switching, electronic switching, data processing, and information management since coming to Bell Laboratories. Member, IEEE, AAAS.

THE BELL SYSTEM TECHNICAL JOURNAL
Vol. 52, No. 10, December, 1973
Printed in U.S.A.

B.S.T.J. BRIEF

Perturbation Calculations of Rain-Induced Differential Attenuation and Differential Phase Shift at Microwave Frequencies

By J. A. MORRISON and T. S. CHU

(Manuscript received August 24, 1973)

In a recent note[1] calculated results of differential attenuation and differential phase shift, as a function of rain rate, were given at frequencies of 4, 18.1, and 30 GHz. The calculations have since been done at 11 GHz also. These results are based on scattering of a plane electromagnetic wave by oblate spheroidal raindrops. The point matching procedure used to obtain nonperturbative solutions to the problem was briefly described, and full details will be presented later.[2] Somewhat similar calculations have been carried out by Oguchi[3] at 19.3 and 34.8 GHz.

The purpose of this note is to point out that a modification of Oguchi's earlier first-order perturbation approximation,[4] for spheroidal raindrops with small eccentricity, gives results which are quite close to those obtained by the point matching procedure. We also give these modified perturbation results at frequencies in the range up to 100 GHz, although they may be less reliable at the higher frequencies, particularly at the heavier rain rates.[4] We remark that the perturbation results are obtained quite inexpensively, whereas the point matching procedure is very costly.

The surface of an oblate spheroidal raindrop is given in spherical coordinates by

$$r = R(\theta) = a(1 - \nu \sin^2 \theta)^{-\frac{1}{2}} = a[1 + \tfrac{1}{2}\nu \sin^2 \theta + 0(\nu^2)], \quad (1)$$

for $0 \leqq \theta \leqq \pi$, independently of the azimuthal angle φ. It was assumed[1] that the ratio of minor to major axis depends linearly on the radius

$\bar{a}$ (in cm) of the equivolumic spherical drop; specifically $a/b = (1 - \bar{a})$. Thus, from (1), $a = \bar{a}(1 - \bar{a})^{\frac{1}{3}}$ and $\nu = \bar{a}(2 - \bar{a})$. We may rewrite (1) in the form

$$R(\theta) = \bar{a}[1 + 2\bar{a}(\tfrac{1}{2}\sin^2\theta - \tfrac{1}{3}) + 0(\bar{a}^2)], \tag{2a}$$

or

$$R(\theta) = \bar{a}[1 + \nu(\tfrac{1}{2}\sin^2\theta - \tfrac{1}{3}) + 0(\nu^2)]. \tag{2b}$$

Then, rather than perturbing about a spherical drop of radius a, with perturbation parameter ν, as did Oguchi,[4] we perturb about the equivolumic spherical drop of radius $\bar{a}$, and take either ν or $2\bar{a}$ as the perturbation parameter.

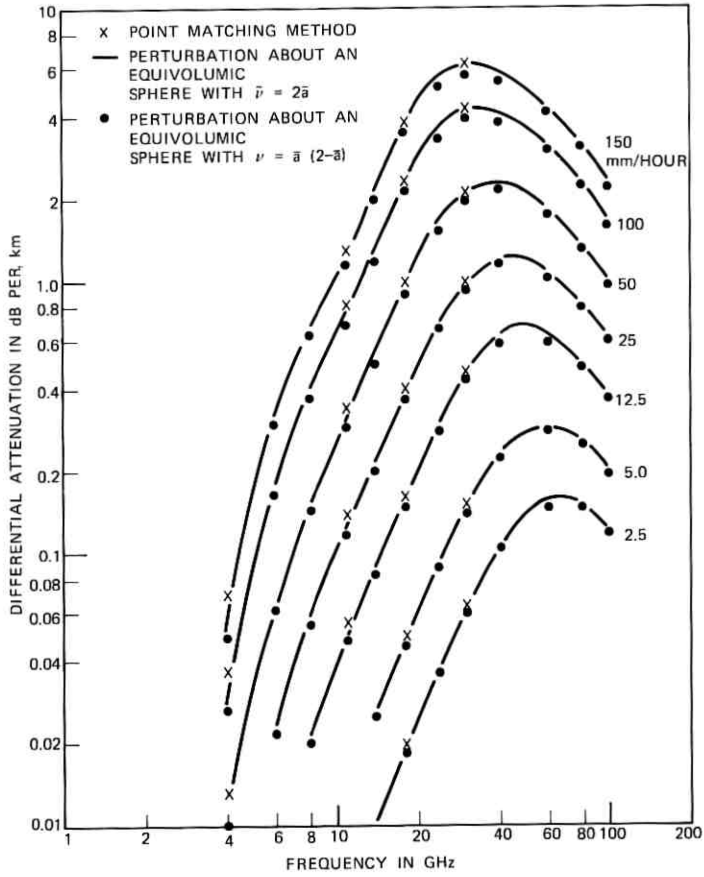


Fig. 1—Rain-induced differential attenuation.

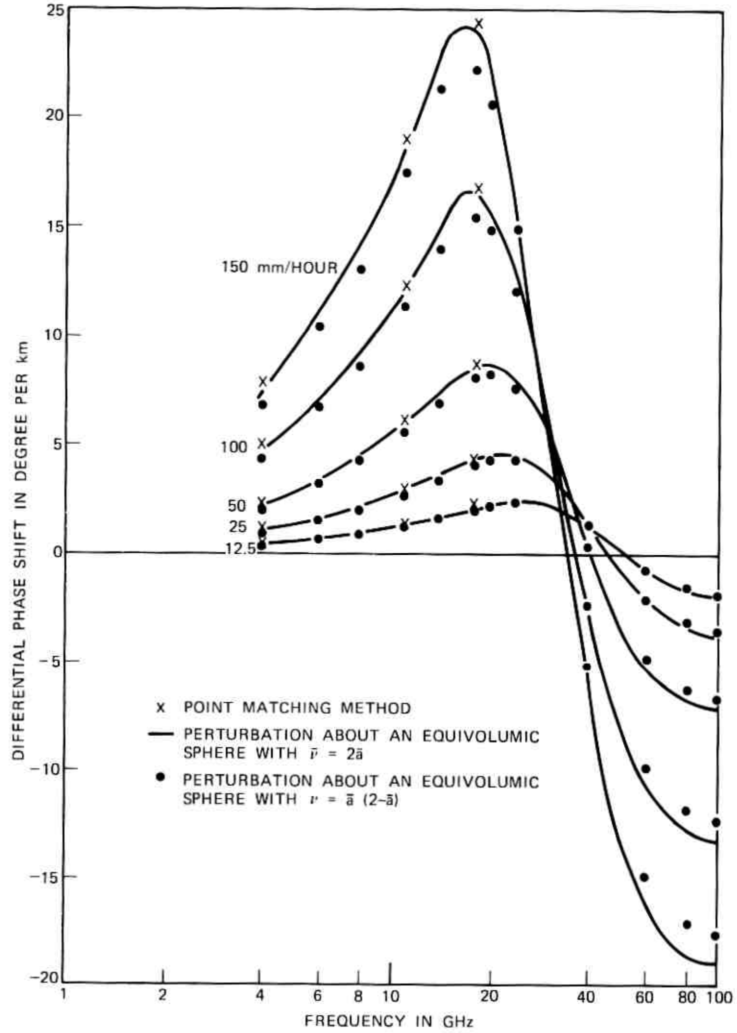


Fig. 2—Rain-induced differential phase shift.

Oguchi's first-order perturbation results have been generalized to axisymmetric raindrops which are nearly spherical, as will be discussed in the detailed paper.[2] There the first-order approximations to the forward scattering functions $S_{\rm I}(0)$ and $S_{\rm II}(0)$, for horizontally disposed oblate spheroidal raindrops, will be compared to the values obtained by the point matching method, for the 14 different drop sizes $\bar{a} = 0.025$

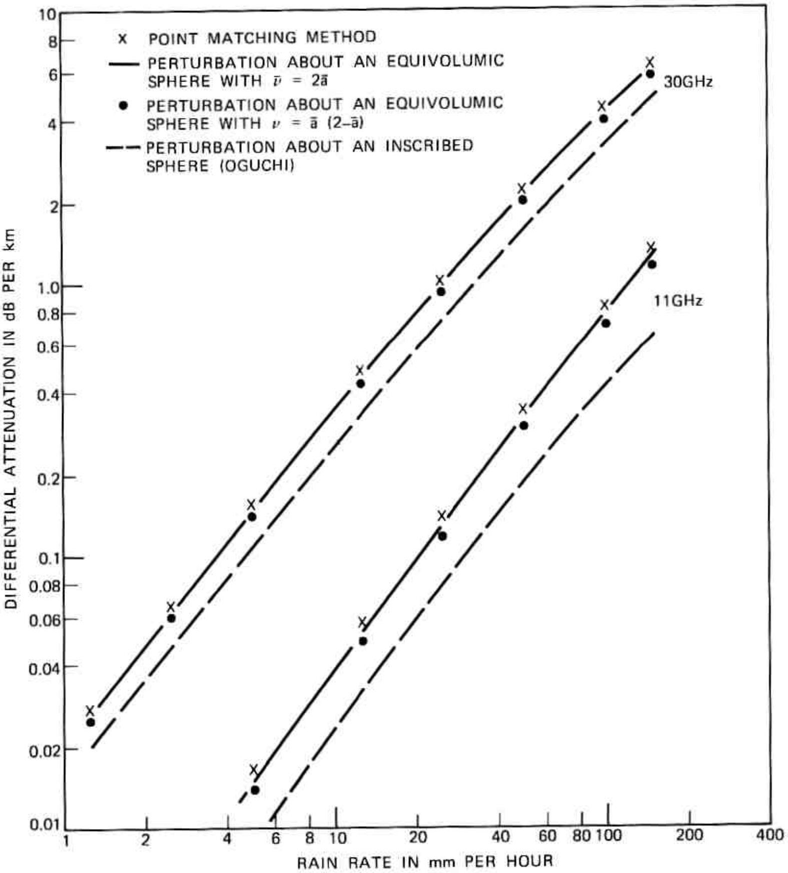


Fig. 3—Comparison between point matching and three perturbation methods for differential attenuation.

(0.025) 0.35. The subscripts I and II correspond to vertical and horizontal polarizations of the incident electric field, respectively. Here we consider only the differential attenuation and differential phase shift, which are obtained[1] by summing the real and imaginary parts of $S_{II}(0)$–$S_{I}(0)$ over the Laws and Parsons drop size distribution.[5] We comment that for the larger drop sizes the perturbation parameter is not small.

Although extra first-order correction terms arise in the expansions about the equivolumic spherical drop, given in (2), they correspond to a constant change in the radius of the drop. Hence the corresponding

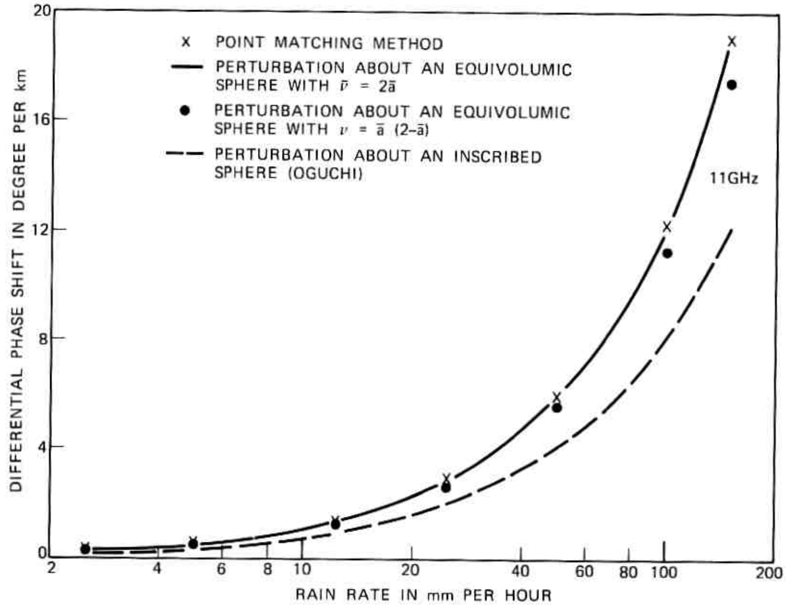


Fig. 4a—Comparison between point matching and three perturbation methods for differential phase shift at 11 GHz.

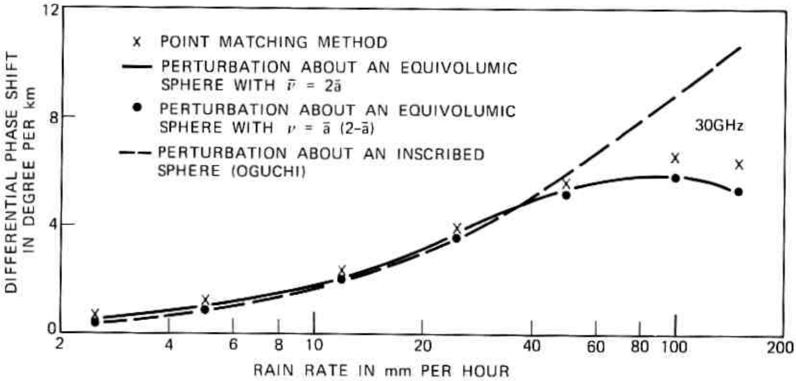


Fig. 4b—Comparison between point matching and three perturbation methods for differential phase shift at 30 GHz.

increments in the forward scattering functions are the same for both polarizations, and therefore do not affect the difference $S_{II}(0)$–$S_{I}(0)$. Thus Oguchi's formulas[4] may be applied directly to calculate this difference, by replacing a by $\bar{a}$, and using either ν, or $\bar{\nu} = 2\bar{a}$, as the perturbation parameter. We remark, however, that some simplifications may be made[2] in the four expressions given in equation (37) of Oguchi's 1960 paper.

Using the approximate forward scattering functions from the perturbations about an equivolumic sphere, we obtained the differential attenuation and differential phase shift versus frequency for various rain rates as shown in Figs. 1 and 2. The refractive indexes of water at 20°C were obtained as described in the previous note.[1] The curves are calculated with the perturbation parameter $\bar{\nu} = 2\bar{a}$, while the dots are calculated with the perturbation parameter ν. The point matching solutions are included as crosses; these show good agreement with the curves, whereas the dots deviate more from the crosses. (In order to avoid confusion, we have omitted the dots and crosses corresponding to the differential phase shift at 30 GHz.) On the other hand, we found much greater discrepancy between the point matching results and the approximate results from the perturbations about an inscribed sphere. This discrepancy is illustrated in Figs. 3, 4a, and 4b for 11 and 30 GHz. Discrepancies for 4 and 18.1 GHz are similar to those for 11 GHz. The above comparison between the point matching solution and three perturbation solutions is consistent with the order of geometrical errors in the three approximations to the oblate spheroid, i.e., the largest error corresponds to (1) and the smallest error corresponds to (2a).

The differential attenuation and differential phase shift recently presented by Watson and Arbabi[6] from 4 through 36 GHz are based upon Oguchi's perturbation solution. These numerical values are in general considerably lower than those of the point matching solution, except for the differential phase shift around 30 GHz, where the differential phase shift from the point matching method decreases sharply. The differential phase shift becomes negative at millimeter wavelengths, and hence remains a significant factor in depolarization.

The authors are indebted to Susan Hoffberg who wrote the programs for calculating the first-order perturbation approximations, and to Diane Vitello who performed the summation over the drop size distribution.

REFERENCES

1. Morrison, J. A., Cross, M. J., and Chu, T. S., "Rain-Induced Differential Attenuation and Differential Phase Shift at Microwave Frequencies," B.S.T.J., *52*, No. 4 (April 1973), pp. 599–604.
2. Morrison, J. A., and Cross, M. J., "Scattering of a Plane Electromagnetic Wave by Axisymmetric Raindrops," to be published.

3. Oguchi, T., "Attenuation and Phase Rotation of Radio Waves Due to Rain: Calculations at 19.3 and 34.8 GHz," Radio Sci., *8*, No. 1 (January 1973), pp. 31–38.
4. Oguchi, T., "Attenuation of Electromagnetic Wave Due to Rain With Distorted Raindrops," J. Radio Res. Labs. (Tokyo), Part 1 in *7*, No. 33 (September 1960), pp. 467–485; Part 2 in *11*, No. 53 (January 1964), pp. 19–44.
5. Kerr, D. E., *Propagation of Short Radio Waves*, New York: McGraw-Hill, 1951.
6. Watson, P. A., and Arbabi, M., "Rainfall Crosspolarization at Microwave Frequencies," Proc. IEE, *120*, No. 4 (April 1973), pp. 413–418.

Erratum

"A Theory of Traffic-Measurement Errors for Loss Systems With Renewal Input," B.S.T.J., Vol. 52, No. 6, July-August 1973, pp. 967–990, by S. R. Neal and A. Kuczura.

A transcription error occurred in eq. (18). It should be

$$E[K_1X_1] = \mu_1(c) + \frac{2}{\nu_1}\mu_1(c)\sum_{k=1}^{c}\mu_1(k) + \nu_1 D_c{}^{(001)} - D_c{}^{(100)}, \qquad (18)$$

and was derived in Ref. 4. This is a transcription error only and does not alter the numerical results or the conclusions (S.R.N. and A.K.).