Massachusetts
Institute
of Technology

Project MAC
Progress Report III
July 1965 to
July 1966

Massachusetts Institute of Technology
Project MAC
545 Technology Square
Cambridge, Massachusetts
02139

Work reported herein was supported by Project MAC, an M. I. T. research program sponsored by the Advanced Research Projects Agency, Department of Defense, under Office of Naval Research Contract Number Nonr-4102(01). The primary support for some of this work came from the M. I. T. departments and laboratories participating in Project MAC, whose research programs are, in turn, sponsored by various government and private agencies. This support is acknowledged by specific mention of agency and contract number in the appropriate sections.

Government contractors may obtain copies of this Progress Report, and all Project MAC Technical Reports listed in Appendix D, from the Defense Documentation Center, Document Service Center, Cameron Station, Alexandria, Virginia 22314. Orders will be expedited from DDC if placed by your librarian, or some other person authorized to request documents.

Other U. S. citizens and organizations may obtain copies of the above items from the Clearinghouse for Federal Scientific and Technical Information (CFSTI), Sills Building, 5285 Port Royal Road, Springfield, Virginia 22151.

BLANK PAGE

ADMINISTRATION AND SERVICES

ARTIFICIAL INTELLIGENCE

BIOLOGY DEPARTMENT

CIVIL ENGINEERING DEPARTMENT

COMPUTATION STRUCTURES

COMPUTER OPERATION

COMPUTER SYSTEM RESEARCH

ELECTRONIC SYSTEMS LABORATORY

LIBRARY RESEARCH

LINCOLN LABORATORY

MAN-MACHINE COMMUNICATION

NON-M.I.T. USERS

RESEARCH LABORATORY OF ELECTRONICS

SCHOOL OF ENGINEERING

SCHOOL OF HUMANITIES AND SOCIAL SCIENCES

SCHOOL OF SCIENCE

SLOAN SCHOOL OF MANAGEMENT

TABLE OF CONTENTS

TABLE OF CONTENTS (continued)

TABLE OF CONTENTS (continued)

TABLE OF CONTENTS (continued)

TABLE OF CONTENTS (continued)

LIST OF ILLUSTRATIONS

PERSONNEL
TO JULY 1, 1966

Administration

Prof. R. M. Fano
- Director

R. G. Mills
- Assistant Director

Committee On Information Processing

Prof. P. M. Morse, Chairman
Dean G. S. Brown
Prof. P. Elias
Prof. R. M. Fano
Prof. R. I. Hulsizer, Jr.
Dean H. W. Johnson
Prof. C. L. Miller
Prof. C. F. J. Overhage
Dean J. B. Wiesner

Academic Staff

S. S. Alexander
A. E. Amstutz
G. W. Angell
D. M. B. Baumann
C. A. Berg
G. D. Bernard
J. M. Beshers
K. Biemann
J. M. Biggs
M. Blum
S. C. Brown
D. C. Carroll
J. G. Charney
S. A. Coons
P. H. Cootner
F. J. Corbató
J. DaSilva
J. B. Dennis
M. L. Dertouzos
A. G. H. Dietz
W. M. Evan
A. Evans, Jr.
R. M. Fano
D. E. Farrar
W. R. Fey
D. W. Fife
M. M. Flood
J. W. Forrester
W. D. Getty
M. Greenberger
M. Halle
K. F. Hansen
H. A. Haus
F. C. Hennie
A. M. Hershdorfer
L. N. Howard
D. A. Huffman
R. Y. Kain
R. Kaplow
E. Kuh
T. W. Lambe
H. B. Lee, Jr.
C. Levinthal
J. D. C. Little
C. L. Liu
R. D. Logcher
D. G. Marquis
D. Martin

Academic Staff (continued)

S. J. Mason
F. A. McClintock
R. McNaughton
H. S. Mickley
C. L. Miller
M. L. Minsky
J. V. Oldfield
C. F. J. Overhage
H. M. Paynter
P. L. Penfield
J. W. Poduska
I. d. Pool
K. F. Reinschmidt
J. F. Reintjes
E. B. Roberts
A. L. Samuel
R. L. Schiffman
L. D. Smullin,
K. N. Stevens
T. G. Stockham, Jr.
G. M. Sturman
A. K. Susskind
H. M. Teager
A. Toomre
D. E. Troxel
J. R. Walton
J. Waelbroeck
H. M. Weingartner
J. Weizenbaum
W. L. White
R. V. Whitman
J. B. Wiesner
J. M. Wozencraft
H. J. Zimmerman

Research Associates, Instructors and Lecturers

G. W. Angell
L. N. Beckreck
D. G. Bobrow
J. W. Brackett
T. H. Crystal
E. P. C. Fernando
A. T. Foster
E. L. Glaser
G. A. Gorry, Jr.
W. L. Henke
M. M. Jones
F. F. Lee
A. Leon
W. H. Linder
J. R. Miller, III
D. N. Ness
S. A. Papert
A. L. Pugh, III
D. Roos
R. L. Rosenberg
J. A. Saltzer
R. L. Schiffman
H. M. Schneider
C. R. Sprague
C. Strachey
J. M. Sussman
C. V. Swanson
R. A. Walter

Non-Academic Research Staff

A. W. Armenti
T. H. Asselin
E. C. Bartels
H. K. Beller
G. L. Benton
J. W. Bex
R. J. Bigelow
W. R. Bjerstedt
G. M. Bonham
H. E. Brammer
T. H. Brooks
M. L. Cabral
E. J. Campbell
J. P. Casey
T. B. Cheek
M. E. Child
G. Clancy
P. A. Crisman
C. A. Cushing
M. M. Daggett
R. C. Daley
R. L. Daniels
T. F. Dempsey
R. A. DePrisco
E. A. Dole
S. D. Dunten
D. J. Edwards
B. J. Fay
C. G. Feldmann
G. M. Fratar
C. Garman
R. V. Goodman
E. P. Gord
R. W. Gosper
M. G. Gottlieb
R. M. Graham
J. W. Grondstra
U. F. Gronemann
J. A. Gunn
E. G. Guttmann
H. C. Haig
D. R. Haring
M. C. Hennemann
W. H. Hennemann
F. B. Hills
M. M. Homburger
E. R. Jensen
G. A. Johanson
D. H. Johnson
I. Y. Johnson
J. Katzenelson
B. W. Kernighan
M. M. Kessler
H. K. Knudsen
P. Kugel
P. T. Ladd
L. Lamport
R. Leopold
H. D. Levin
A. Malhotra
C. Marceau
K. J. Martin
W. D. Mathews
E. M. Mattison
W. D. Maurer
P. McCrensky
S. D. McIntosh
V. E. McLoud
K. S. Menger
E. F. Miller
R. G. Mills
N. I. Morris
H. G. Murray, Jr.
R. M. Nacamuli
R. C. Nelson
W. R. Noftsker
J. F. Nolan
L. P. Odland
M. A. Padlipsky
J. A. C. Parisot
M. M. Pennell
T. Pitidis-Poutous
R. B. Polansky
E. Quisenberry
R. Rappaport
J. M. Reed
E. C. River
S. L. Rosenbaum
D. T. Ross
A. J. Saltalamacchia
P. Samson
R. A. Sayers
G. C. Schroeder

Non-Academic Research Staff (continued)

O. G. Selfridge	J. W. Spall	D. B. Wagner
D. J. Shea	N. C. Spencer	J. F. Walsh
P. M. Sheehan	G. Spielman	J. E. Ward
P. E. Smith	R. H. Stotz	D. R. Widrig, Jr.
E. Soares	W. D. Stratton	B. L. Wolman
M. V. Solomita	M. K. Thompson	C. Wylie
D. Sordillo	D. E. Thornhill	
R. L. South	T. H. Van Vleck	

Resident Guests

S. I. Ackley	System Development Corp.
D. Barovich	I. B. M. Corporation
A. J. Berger	Boeing Company
L. M. Bousquet	French Government
A. Forte	Yale University
D. H. Johnson	National Security Agency
W. L. Johnson	Ford Motor Company
J. H. Jones	McDonnell Aircraft Corp.
R. O. Ladson	UNIVAC Division, Sperry Rand
F. Luccio	Olivetti & Company
R. S. Lynn	North American Aviation, Inc.
A. K. Mills	Dow Chemical Company
J. V. Oldfield	University of Edinburgh
D. E. Oppert	National Security Agency
J. H. Porter	Chevron Research Company
R. Silver	MITRE Corporation
H. L. Stambler	U.S. Weather Bureau
R. B. Wise	IIT Research Institute

Non-M. I. T. Users

I. E. Sutherland	ARPA - Department of Defense
R. W. Taylor	ARPA - Department of Defense
R. A. Kirsch	Bureau of Standards
R. Des Maisons	Harvard University
R. R. Fenichel	Harvard University
I. Hazel	Harvard University
M. C. Hennemann	Harvard University
A. G. Oettinger	Harvard University
S. M. Pizer	Harvard University
A. S. Priver	Harvard University
A. Ruyle	Harvard University
P. J. Stone	Harvard University
K. Winiecki	Harvard University
S. Lorch	Massachusetts General Hospital
M. T. McGuire	Massachusetts General Hospital
G. C. Quarton	Massachusetts General Hospital
J. McCarthy	Stanford University

Research Assistants

W. A. Bailey
R. R. Bartsch
P. Berman
W. T. Beyer
R. J. Bobrow
J. R. Brach
A. A. Bushkin
D. Chase
P. Clermont
L. O. Craft
L. C. Erdmann
D. S. Evans
R. C. Feldman
S. Geffner
H. L. Graham
A. D. Griffith
A. Guzmán
E. J. Hall
T. P. Hart
B. J. Huffman
E. G. Hurst, Jr.
E. L. Ivie
T. J. R. Johnson
A. R. Kessler
B. V. Koen
J. N. Kogan
L. Krakauer
J. F. Kramer
B. R. Kusse
R. B. Lapin
M. I. Levin
J. L. Linderman
G. C. Ling
S. Lipner
F. L. Luconi
A. W. MacEwan
H. S. Magnuski
H. S. Marcus
W. A. Martin
L. F. McPherson, III
R. L. Miller
J. Moses
L. M. Norton
R. P. Parmelee
A. Pawlikowski
G. Piotrowski
I. Plotkin
J. E. Rodriguez
L. J. Rotenberg
P. J. Santos, Jr.
R. A. Schlumff
H. L. Selesnick
A. Setnick
E. H. Sibley
D. Somekh
S. L. Strong
L. C. Teague, Jr.
W. Teitelman
C. W. Therrien
C. C. Tillman, Jr.
E. C. Van Horn, Jr.
J. W. Weber
M. Weinberger
J. I. Weiner
P. A. Wieselmann
D. U. Wilde
R. D. Wright
E. Yourdon

Other Graduate Students

J. M. Alanko
D. M. Auslander
B. H. Bloom
W. Brody
J. R. Brown
P. B. Clark
J. A. Davis
P. J. Denning
R. Diephuis
J. L. Doane
M. Edelberg
D. D. Falconer
D. L. Flamm
Z. Fluhr
J. C. Free
R. C. Gammill
M. L. Gentry
J. Gertz
M. M. Gold
J. Golden
D. Haccoun
T. L. Hawk
T. E. Hlavac, Jr.
W. Inglis
H. Ishii
T. L. Jones
D. A. Kendrick
B. Kwok
M. A. Lieberman
D. T. Llewellyn-Jones
R. E. Marks
T. McElroy
J. I. Meltzer
P. F. Meyfarth
J. H. Morris, Jr.
N. P. Negroponte
J. Rockart
L. L. Roos, Jr.
D. D. Y. Ryu
J. G. Schwartz
H. S. Schwenk, Jr.
C. L. Seitz
L. L. Selwyn
A. W. Slawson
D. H. Slosberg
E. Soltero
M. J. Spahn
J. E. Spencer
J. Sukonick
J. L. Sussman
C. N. Taubmann
B. T'sou
J. M. Wallace
R. N. Wallace
G. A. Walpert
D. Wright
S. Zucker
M. Zwick

Undergraduates

W. C. Albertson
E. L. Anderson
W. L. Bass
P. Bos
N. R. Cohler
J. E. Davidow
A. A. Dvorak, Jr.
J. L. Fields
D. Griffel
R. Haber
F. Hickey
S. Kidd
T. Knight
A. N. Kramer
D. A. Lasher
K. W. MacLean
S. T. Mooallem
L. H. Morton
R. E. Raffo
J. S. Roe
G. L. Rosen
T. P. Skinner
C. R. Smith
S. W. Smoliar
G. J. Sussman
S. A. Ward
P. M. Wolk
B. Zimmermann

Technical Assistants

W. Bennett
G. D. Blum
F. J. Cole
R. Fidler
R. Greenblatt
J. T. Holloway
W. M. Parks
C. H. Willson
A. W. Wilson

Operating Staff

G. Andrews
D. Borsini
R. Brenner
K. Carley
T. Dattilo
R. A. Degan
M. Dickie
C. Ferguson
R. G. Hart
J. McGillivary
R. McNamara
R. Moore
S. Nelson
F. Noseworthy
M. Pagliarulo
R. Parker
J. Payson
D. Peaslee
R. Poole
E. Reardon
J. W. Waclawski

Technicians

R. J. Porcaro

Administrative and Supporting Staff

C. S. Alles
N. H. Arold
M. E. Baker
T. Bergeron
A. W. Bowen
S. Buell
J. M. Constantine
B. L. Cullen
K. Day
F. H. Dilworth
D. J. Dolan
G. Gulledge
A. Handelman
D. L. Jones
D. Kontrimus
T. Liotta
L. Martin
E. G. Moore
W. Needre
S. A. O'Leary
J. E. Piñella
D. Rain
C. L. Robinson
J. Robinson
D. C. Scanlon
E. L. Schneider
R. Slosberg
J. Wallen
P. Wester
R. M. Yunetz

PREFACE

Project MAC was organized at the Massachusetts Institute of Technology in the spring of 1963 for the purpose of conducting a research and development program on Machine-Aided Cognition and Multiple-Access Computer systems. It operates under contract with the Office of Naval Research, acting on behalf of the Advanced Research Projects Agency of the Department of Defense.

The broad goal of Project MAC is the experimental investigation of new ways in which on-line use of computers can aid people in their individual intellectual work; whether research, engineering design, management, or education. One envisions an intimate collaboration between man and computer system in the form of a real-time dialogue where both parties contribute their best capabilities. Thus, an essential part of the research effort is the evolutionary development of a large, multiple-access computer system that is easily and independently accessible to a large number of people, and truly responsive to their individual needs. The MAC computer system is a first step in this direction and is the result of research initiated several years ago at the M. I. T. Computation Center.

Project MAC was organized in the form of an interdepartmental, interlaboratory "project" to encourage widespread participation from the M. I. T. community. Such widespread participation is essential to the broad, long-term project goals for three main reasons: exploring the usefulness of on-line use of computers in a variety of fields, providing a realistic community of users for evaluating the operation of the MAC computer system, and encouraging the development of new programming and other computer techniques in an effort to meet specific needs.

Faculty, research staff, and students from fourteen academic departments and four interdepartmental research laboratories are participating in Project MAC. For reporting purposes, they are divided into seventeen groups, whose names correspond in many cases to those of M. I. T. schools, departments and research laboratories. Some of the groups deal with research topics that fall under the heading of computer sciences; others with research topics which, while contributing in a substantive way to the goals of Project MAC, are primarily motivated by objectives outside the computer field.

The purpose of this Progress Report is to outline the broad spectrum of research being carried out as part of Project MAC. Internal memoranda of Project MAC are listed in Appendix A, and MAC-related theses are listed in Appendix B. Some of the research is cosponsored by other governmental and private agencies, and its results are described in journal articles and reports emanating from the various M. I. T. departments and laboratories participating in Project MAC. Such publications are listed in Appendix C of the report. Project MAC Technical Reports are listed in Appendix D.

Robert M. Fano
Cambridge, Massachusetts

BLANK PAGE

ADMINISTRATION AND SERVICES

Administrative Facilities and System Management

A. The Basic Approach and External Appearance

B. The User Viewpoint

C. The Group Supervisor Viewpoint

D. The Central System-Administration Viewpoint

E. Security

Academic Staff

R. M. Fano - Director

Non-Academic Research Staff

R. G. Mills - Assistant Director

Administrative Staff

M. L. Cabral
J. A. Gunn
M. M. Homburger
A. J. Saltalamacchia
T. H. Van Vleck

Administrative Facilities and System Management - Richard G. Mills and Thomas H. Van Vleck

This is a report on the general nature and state of development of various programming tools that aid in the management of the Project MAC IBM 7094 Compatible Time-Sharing System (CTSS). The set of programs which will be described are referred to collectively as the "administrative subsystem". It is entirely proper to consider the subsystem as part of CTSS, even though most of the programs are in the same class as any other user-generated programs. That is, they do not involve specially privileged interaction with the supervisor or other parts of the central facilities of CTSS, with one significant exception: some problem numbers under which the various subsystem programs operate are granted extraordinary privileges, in the sense described in Section AG.7.03 of the CTSS manual.*

In this general description, it is not possible to cover all of the details of program operation and interaction with the central system. In particular, it will not be possible to trace through the full complexity of the security structure surrounding critical files of the administrative subsystem.

One additional general comment is in order. The subsystem described here is only partially the result of a careful design effort. Much of it has grown in ad hoc fashion, and many aspects of its structure are influenced by vestiges and artifacts of the past. It is important to note that, because there is minimum dependence upon any particular features of the central system, other subsystems with totally different exterior aspects could be implemented as easily as the present one. In fact, were CTSS expected to have a longer life, there might be sufficient motivation to refine the present system. However, very likely what is described here is essentially the final version.

*The Compatible Time-Sharing System: A Programmer's Guide, Second Edition, The M.I.T. Press, Cambridge, Mass., August 1965 (Library of Congress No. 65-25206)

A. THE BASIC APPROACH AND EXTERNAL APPEARANCE

The general system-administration point of view is that CTSS offers two primary resources: secondary storage space and processor time. System access is recognized as a resource rather indirectly, and other attributes, such as responsiveness, which might be viewed as resources, are not controlled directly.

The process of adding a user to the system consists of giving him an identity which the system can recognize (a problem-number/programmer-number pair) and a secure means for the system to challenge and receive authentication of this identity (a "password" scheme). The new user may then be issued certain of the system's resources in the form of quotas of secondary-storage space and processor time; he may also be assigned other attributes relative to his particular use of the system. All of this information is conveyed to the system in the form of a multi-field record, which is inserted in a particular CTSS file containing one such record for each authorized user.

This "resource-allocation file", together with a small set of commands whose use is restricted to the system administrator, constitutes the entire interface between the administrative subsystem and the core supervisor of CTSS.

The LOGIN command is the guardian of access to the system; it decides whether a user attempting to enter may do so, on the basis of information in the resource-allocation file, and communicates time allotments and user privileges for the individual to the A-core supervisor.

The scheduling algorithm in A-core accounts for processor usage and prevents a user from exceeding his time allotments. The file system is responsible for keeping a user from using too much secondary storage.

The "external" administrative subsystem serves to facilitate and organize manipulation of information in the resource-allocation file, and periodically initiate certain operations related to both resource-allocation file updating and reporting of system-resource consumption. The external administrative subsystem consists of about 25 special-purpose "saved" files and commands, which involve about 10,000 to 15,000 words of code.

B. THE USER VIEWPOINT

Users see a reasonably simple system. They are uniquely assigned, for system-administrator purposes, to one of about eighteen "groups". The user is admitted to the system by the action of a group supervisor and is allotted a quota of system resources. In general, secondary-storage quotas are assigned for relatively long periods; processor-time quotas are adjusted more frequently. Also, time quotas are assigned in units of processor minutes per month, so a user's quota is refreshed on the first of each month.

Both time and storage quotas are subdivided. Users may be allotted space on drum, disk, or magnetic tape (no drum space is available for normal users). Processor time quotas are subdivided among five numbered "shifts". These are defined as follows:

Shift 1:	0800 - 1700	For period beginning 0800 Monday and ending 0800 Saturday.
Shift 2:	1700 - 2400	
Shift 3:	0000 - 0800	
Shift 4:	0800 Saturday to 0800 Monday	
Shift 5:	Foreground-Initiated Background (FIB), described in CTSS Manual Section AH.1.03.	

A user may be allotted time in any or all shifts. If he consumes his allotment on a given shift, the system automatically logs him out, and he must wait until the beginning of a shift in which he has resources remaining before he is permitted to log in again. He can make a request to his group supervisor for an adjustment in his allotments, and the group supervisor may, at his own discretion, choose to do so. As previously mentioned, time quotas are valid for an arbitrarily-selected accounting period of one month; thus, to refresh a user's time allotment, the "time used" accumulations are automatically reset to zero at the end of the month.

C. THE GROUP SUPERVISOR VIEWPOINT

Group supervisors see a bit more of the structure of the administrative subsystem. First, each supervisor is assigned a special problem number for maintaining a "group allocation file" (GAF) containing his distribution of a block of computer resources, which he has been allotted by the central administration for further subdivision. For this purpose, he has access to a few special-purpose programs, the most important of which is the so-called ALOCAT command. ALOCAT is a special-purpose editor tailored to the job of maintaining GAF's. It contains convenient features for summing allocated resources, notifying the group supervisor if he has exceeded the limit of his block, etc. ALOCAT is designed to be highly interactive and informative; it is designed for the group supervisor who knows little and cares less about details of using CTSS.

The functions delegated to a group supervisor are as follows: 1) he may add or delete users; 2) he may allot resources of secondary-storage space and processor time, within the overall limit of the block allotment to his group; 3) he may change the password of a user; and 4) he may influence the probability that a user will be permitted to log in. In particular, for demonstrations and other special purposes he can give a user absolute assurance of permission to log in.

After a group supervisor has altered the privileges or resources of some of his users, he executes a command which notifies the central section of the administrative subsystem of his changes. Once each day, in an operation colloquially referred to as "turning the crank", the central section takes note of all such requests and makes the indicated adjustments in the resource allocation file, thus effecting the change requested by the group supervisor. This operation is now done automatically once each day by the FIB facility.

D. THE CENTRAL SYSTEM-ADMINISTRATION VIEWPOINT

With day-to-day resource-allocation problems handled by group supervisors, the central administration need only be involved with rather infrequent adjustments of the block allotments to groups, and maintenance and operation of the various elements of the system-administration subsystem. For example, this is the level where responsibility rests for assuring that group-supervisor-initiated changes are made in the resource allocation file regularly and correctly.

In addition, the daily, weekly, and monthly reporting of resource consumption is automatically performed at this level. The central administration is also in a position to do special monitoring of any aspects of the system performance.

E. SECURITY

The issue of system security is a vital one in the context of a future information public utility. In CTSS, security is still an important issue, but the emphasis is slightly different. For example, it is extremely important that a user be severely impeded, if not completely thwarted, in any attempt to take control of the computer away from the CTSS supervisor. On the other hand, the issue of user-to-user file security, although important, is less critical. From a purely human point of view it is desirable to assure each user that other users cannot access his personal files without his permission; but, in fact, very few users have files whose absolute security must be maintained at any cost.

It is much more important to increase the difficulty of unauthorized consumption of a user's precious processor-time resources. This has the curious effect of enormously upgrading the importance of file-system security. The reason, obviously, is that a particular file within the normal CTSS file system contains the resource allotments and passwords of all users. Penetration of the security of this file could severely compromise the ability of various administrative levels to control system resources.

As might be expected in a community like M. I. T., over the years there have been attempts -- some quite successful -- to penetrate the various security bars which have been erected from time to time in the evolving system. No earnest attempt to defraud the administration has been discovered; most of these activities were undertaken by sophomores (in fact or in temperament) who were interested in demonstrating their ability to do anything advertised as difficult, and, in some cases, their general contempt for authority. There have been a few unthinking pranks and a few ventings of frustration which caused serious and deplorable interference with other system users, but these have been rare.

It is an interesting commentary on the psychology of the M. I. T. sophomore (again, both in fact and in temperament) that the amount of (detected) probing of system security has diminished considerably since "The Management" pointed out that penetration for frivolous purposes is quite possible but somewhat anti-social, and penetration for fraudulent or destructive purposes is also possible but so undesirable as to warrant punishment if detected. Those very few individuals whose actions have caused major interference with the using community have either departed or assumed a more social attitude.

ARTIFICIAL INTELLIGENCE

An Autonomous Manipulator System

A. Computer and Related Hardware

B. Special Input Devices

C. Special Output Devices

D. Programming

E. Vision/Manipulation Experiments

Unrecognizable Sets of Numbers

Linearly Unrecognizable Patterns

CONVERT: A Pattern-Driven Manipulation Language

POLYBRICK: Parallelpiped Pattern Recognition

PDP-6 Display Subroutines

Nonlinear Optimization Applied to Checkers

The Automata Theorist's Helper

A Syntax-Based Score-Reading Program

Computer Processing of Musical Data

Some Experiments in Computer Composition of Tonal Music

ADEPT: A Heuristic Program for Proving Theorems of Group Theory

LISP Maintenance

Manipulation of Algebraic Expressions

Computer Experiments in Finite Algebra

PILOT: A Step Toward Man-Computer Symbiosis

On-Line CTSS Instruction

Academic Staff

M. Blum
D. G. Bobrow
M. L. Minsky
S. A. Papert
A. L. Samuel

Non-Academic Research Staff

H. K. Beller
H. E. Brammer
D. E. Eastlake
D. J. Edwards
E. P. Gord
W. R. Gosper
R. Greenblatt
W. H. Henneman
J. T. Holloway
L. Lamport
W. D. Maurer
W. R. Noftsker
P. Samson
O. G. Selfridge
D. Sordillo
R. L. South
G. Spielman

Research Assistant and other Students

R. Baecker
W. Bennett
B. H. Bloom
G. D. Blum
R. J. Bobrow
S. Geffner
A. K. Griffith
A. Guzman
T. P. Hart
T. L. Jones
T. Knight
L. J. Krakauer
M. I. Levin
S. M. Libman
W. A. Martin
J. Moses
L. M. Norton
A. Reed
J. S. Roe
G. L. Rosen
A. W. Slawson
C. R. Smith
S. W. Smoliar
M. J. Spahn
J. Sukonick
G. J. Sussman
W. Teitelman
J. M. Wallace

Guests

A. Forte - Yale University
R. Silver - MITRE Corporation

An Automomous Manipulator System - Marvin L. Minsky

Our goal is to develop techniques of machine perception, motor control, and coordination that are applicable to performing real-world tasks of object-recognition and manipulation. A suitable mechanism will then be able to assemble simple mechanical devices according to goal-directed (rather than programmed) instructions (e. g., disassemble an object on a table, or construct an object described by a drawing). Our aim is to have a computer-controlled system accept a relatively uncomplicated command and, without human assistance, locate, grasp, and assemble parts of a simple mechanical device.

The problem is difficult, but in an engineering sense rather than a theoretical one -- it is complex and intricate, rather than intellectually baffling. Numerous attempts that tested specific schemes for "pattern-recognition" have failed, because no isolated special-purpose technique can solve a problem that needs a large-scale system -- not because of a lack of efficient techniques. A machine that can "see" will need all the resources generated by advances in Artificial Intelligence and programming.

The principal parts of the autonomous manipulator system are a general-purpose digital computer, its input and output devices, and its programs. Only a large, general-purpose computer can deal with the intricacies of the processes involved and, as the mechanisms of the systems become clearer, certain components may become candidates for special-purpose-hardware treatment. For the present, though, it is important to retain flexibility for research, even at the cost of a serious loss of real-time speed.

The system needs visual and tactile input devices capable of unusual discrimination ability. We are developing a series of new devices to meet this need. A number of more conventional input devices are also warranted for kinesthetic feedback. These requirements have necessitated the design and construction of a versatile array of input devices and channels. The output system also demands a variety of output channels -- to the motor organs for motion, and to the human users for monitoring and display.

Finally, and most important, the system needs computer programs. They must deal with analysis of the visual (and tactile) scene; coordinate the motor actions; and provide its internal information storage, planning, and general goal-directed problem-solving activity.

In order to press toward our goals, we have had to build a number of tools which were not readily available. We have had to design and construct computer and related hardware, set up support facilities, develop systems and support programs, perform experiments, and analyze and document results. One might characterize the project's work from August 1965 through April 1966 as a tool-building phase.

Although some tools are not yet complete, we are now able to begin serious study of the most difficult problem facing the project: The analysis of real-world three-dimensional scenes. This problem will occupy most of our attention for an indefinite period. Because programming for increased dexterity in manipulation (and for the interpretation of goal-directed commands) is better understood, it is less difficult. However, a considerable amount of work still is necessary in this area.

A brief summary is presented to sketch the present status of the main portions of the project.

A. COMPUTER AND RELATED HARDWARE

The system is based on a DEC PDP-6 computer and related devices. With this is associated an assortment of tapes, printers, memories, input-output channels, and CRT displays. Unique to the manipulator project are a number of special interface connections for the devices described next. Because of the unusual size and complexity of the programming systems involved, a very large and fast core memory (2^{18} words of 40 bits; 2.5-microsecond cycle-time) is to be added during the summer of 1966.

B. SPECIAL INPUT DEVICES

The sensory equipment includes two visual-input devices: TVA, a vidicon television camera, and the more precise TVB, an image-dissector device for controlled-scan analysis. Currently in use is a continuous-position-and-pressure-detecting tactile sensor, based on a time-domain reflectometer. For position sensing, there are a variety of inputs from the hand and arm. A more versatile system has been designed and is under construction.

C. SPECIAL OUTPUT DEVICES

The manipulator, shown in Figure 1, consists of a hydraulically-powered, electrically-controlled industrial arm, and a hand with five degrees of freedom (but with only a simple gripping ability). The arm was built and given to us by the American Machine and Foundry Company, and is sufficient for our present simple goals. More versatile hands and arms are being developed to be available when programming is able to cope with more intricate manipulation.

A new arm, almost completed, is shown in Figure 2. This arm, specifically designed for the project, will be less massive and more versatile than previous designs. Its two main parts are a shoulder and an arm. The shoulder is a strong but slim stand, that terminates in a two-axis rotation joint (not shown in these pictures). The arm is a modular jointed system. Each section consists of a long tetrahedral "bone" with two hydraulic-cylinder

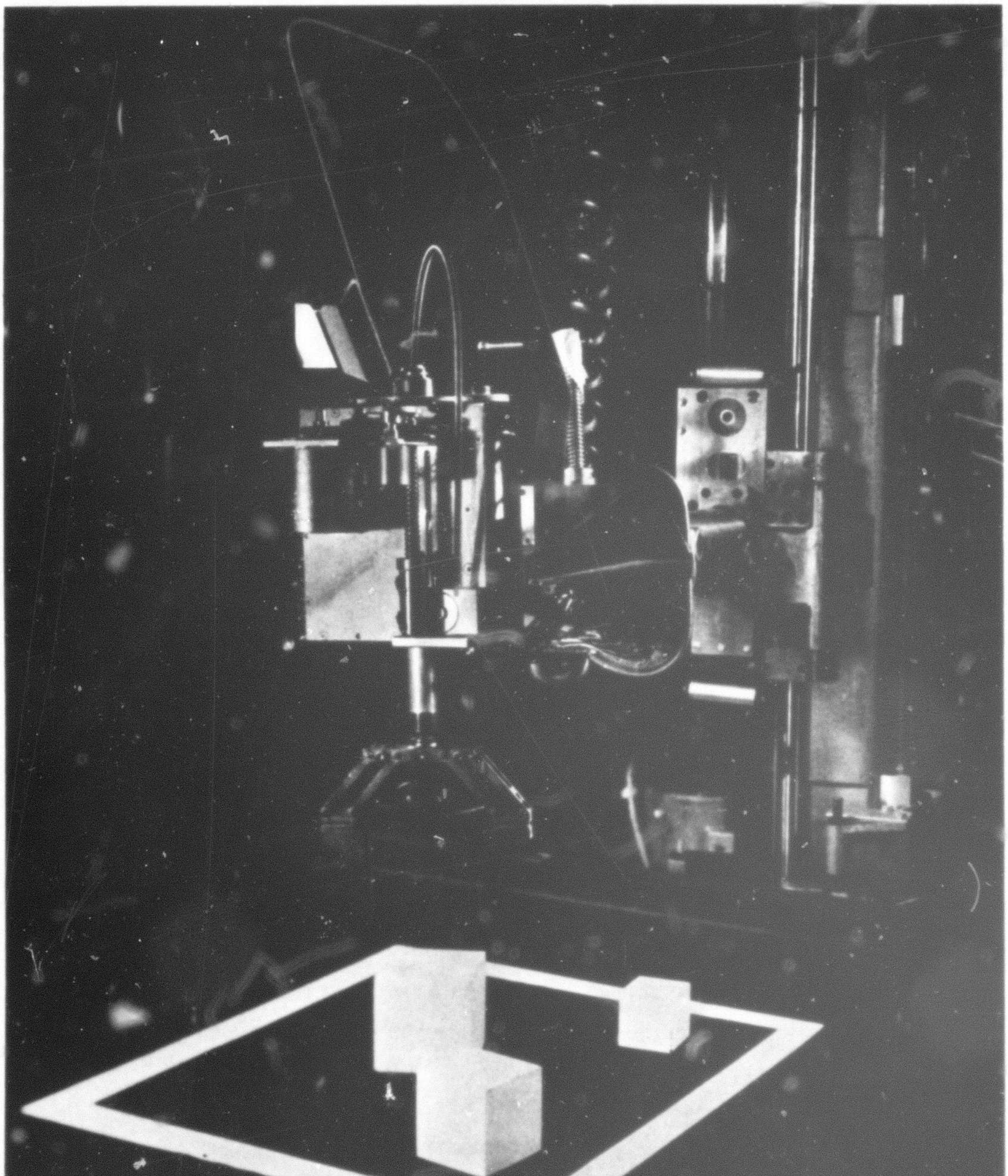


Figure 1. Modified AMF "Versatran" Arm

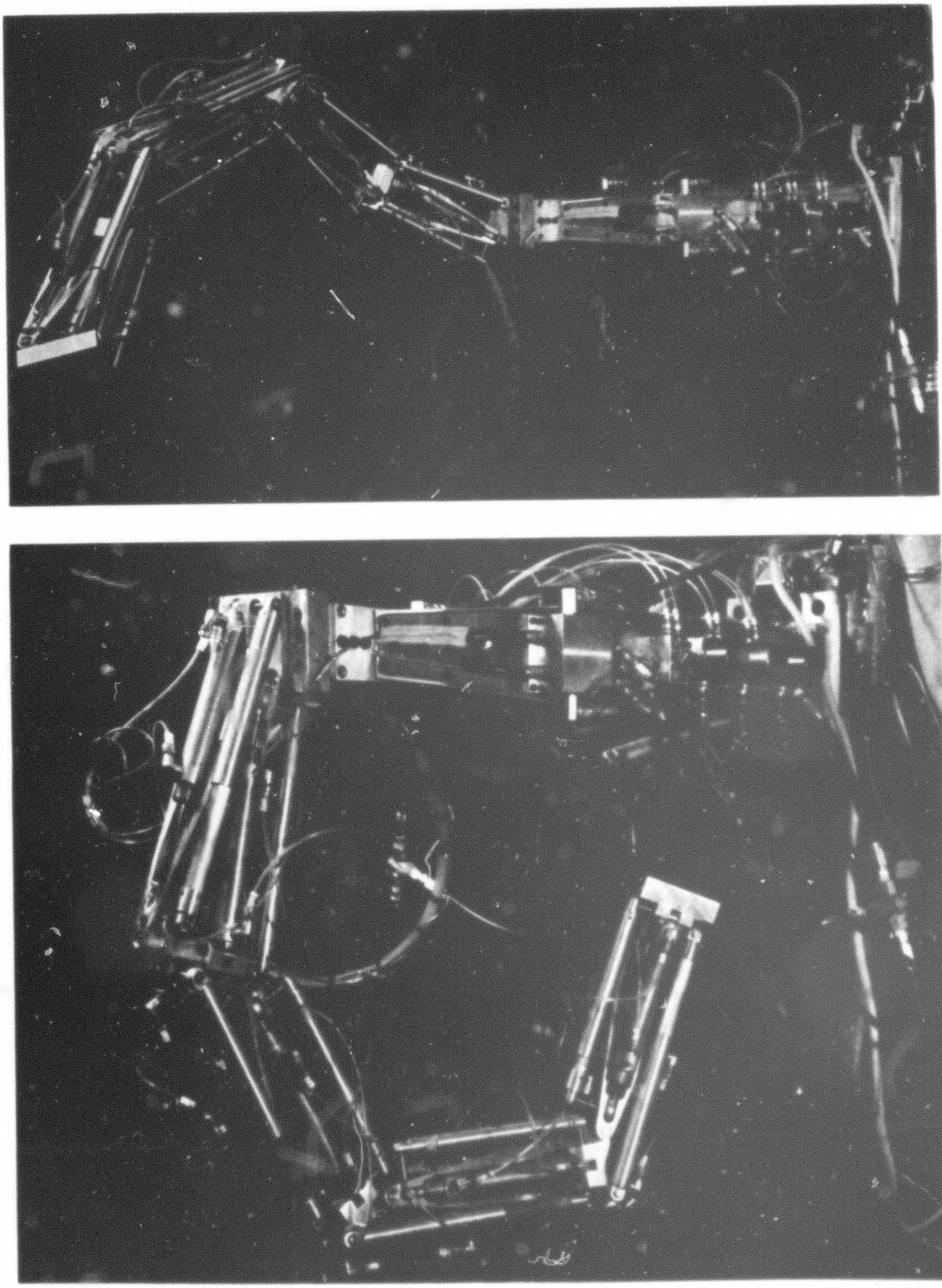


Figure 2. "Lobster-Claw" Arm Mechanism

"biceps". With four sections, the system resembles a tentacle (or really a lobster arm) rather than a humanoid or industrial-type arm. Since it is important that the "hand" be able to reach any point in a portion of space from any direction, this requires at least six axes, in addition to grasping.

D. PROGRAMMING

Programs have been developed to read selected parts of the visual scene, analyse them for parts of polygonal objects, and then transform them to real-world coordinates. The present programs are still rudimentary, and their extension is vital to the project.

On the output side, we have programs that handle the rather complicated relations between arm-control signals and the resulting movements, and programs are being developed for better Hand-Eye coordination. There remain non-trivial problems of controlling motion through a cluttered space without dangerous or destructive consequences.

For planning and control of the overall activity of the machine, we are taking into account texture and color, as well as object boundaries and the like. The system will be able to partition a scene into objective regions, combine these regions into proposed objects, and finally represent this collection of pseudo-objects as an abstract model of the objects and background in real space.

A program already completed and documented can do something of this nature -- given an isometric two-dimensional drawing, it constructs an abstract three-dimensional model of a scene composed of rectangular objects. (See Guzman's report, elsewhere in this section.) Extensions to this program are under way. A first-generation system for <u>seeing</u> should be in operation this fall, resulting from the merger of the above-mentioned systems.

E. VISION/MANIPULATION EXPERIMENTS

Two experimental demonstrations have been completed -- "ball catching" and "cube handling". Their primary purpose was to consolidate all points in the project's development. Each experiment requires all equipment in the project to work smoothly at one time: and close coordination between staff members who develop the parts is necessary.

"Ball catching" is the interception by the hand, under TVA visual tracking, of a thrown ball. The visual problem is simple -- the target is a small, moving, high-contrast object, and no analysis of its geometric form is necessary. This was to check out the many new programs and devices involved.

"Cube handling" requires visual and tactile location of a cube on a table and picking it up. This demonstration checked out the next stage of complexity of the visual-analysis system.

The next experimental demonstration -- building a specified structure from various objects piled on a table (using visual feedback) -- will be the first to require a truly complex visual system.

Unrecognizable Sets of Numbers -- Marvin L. Minsky and Seymour A. Papert

Let A be some set of positive integers written in binary notation. It is natural to ask what kind of computing machine could recognize the set in the sense of deciding whether a given binary sequence represents a number belonging to A. The technique described in this note enables one to show that certain sets cannot be recognized by finite-state automata (i. e., these sets are not "regular"). The essential idea is this: Let $\pi_A(n)$ be the number of members of A less than the integer n. We show that the asymptotic behavior of $\pi_A(n)$ is subject to severe restraints if A is regular. These constraints are violated by many important natural numerical sets whose distribution functions can be calculated, at least asymptotically. These include the set P of prime numbers for which $\pi_p(n) \sim n/\log n$ for large n, the set of integers A(k) of the form n^k, for which $\pi_{A(k)}(n) \sim n^{1/k}$ and many others. The technique cannot, however, yield a decision procedure for regularity, since for every infinite regular set A there is a non-regular set A' for which $|\pi_A(n) - \pi_{A'}(n)| \leq 1$, so that the asympatotic behaviors of the two distribution functions are essentially identical.

A remarkable set of equivalence theorems has been investigated recently by S. A. Papert and R. McNaughton. (See McNaughton, this report.) Consider the class of "extended" regular expressions obtained by adding not and and to the or, star, and concatentation of the Kleene Algebra. It is easily shown that to obtain not in the McCulloch-Pitts formulation, we do not require anything more than a loop around a single cell. So, if we confine ourselves to extended regular expressions that contain no stars, we can obtain McCulloch-Pitts nets that have only single-cell loops of this kind. It can be shown that the converse is true -- that nets with only this kind of loop can be described by "star-free" extended regular expressions.

It turns out that machines of this class cannot "count" cyclically (i. e., they cannot even determine whether the number of ONE'S in the input is even or odd). A more general definition of "non-counting" would be: a regular set E in a "non-counting set", if there exists a number n such that for all strings U, V, and W, and all positive integers p, if $\underline{U V^n W}$ is in E, then so is $U V^{n+p} W$.

It was first shown that any star-free E is non-counting: the converse, which is also true, was more difficult to show. This theorem points toward a connection between the formulations we have been considering and a rather different way of looking at machines -- namely, the approach through semi-groups.

In the semi-group formulation, one thinks of a machine as having a set of states, which are transformed into one another by the input signals. From the group-theory point of view, the existence of permutation induced by a signal on some of the states means that there is a subgroup in the semigroup of transformations (on the states) generated by the input signals. The important theorem here is that a machine that is "permutation-free", in this sense, is also a star-free machine, and vice-versa. It turns out that this is equivalent to the class of machines that can be constructed out of the "unit automata" of Krohn and Rhodes.

Finally, there is a connection with the predicate calculus. Consider an alphabet $X = \{x_1 \ldots x_r\}$. For each i we define the propositional function $F_i(t)$ which asserts that the t^{th} input is x_i. Let t_i, $t_2 \ldots t_s$ be a set of integral valued variables and let P be the class of expressions defined by:

(a) $F_i(t_j) \varepsilon P$ for all i, j
(b) $t_j < t_k \varepsilon P$ for all i, k
(c) Any proposition form in members of P is in P
(d) If $A \varepsilon P$ and A contains the free variable t_i, then $(t_i)A$ and $(\exists t_i)A$ are in P.

Let P_o be the set of members of P with no free variables. Then P_o defines the class of words on X which satisfies it.

A regular set E is in L if it can be defined as the set of words satisfying a predicate calculus expression, P_0, as defined above. The class L is the "L-languages" of McNaughton and these also turn out to be equivalent to the star-free machines. (See Minsky, Appendix C.)

Linearly Unrecognizable Patterns - Marvin L. Minsky and Seymour A. Papert

We have succeeded in clarifying, to a certain extent, the enormously confused situation with regard to the usefulness of perceptron-like, pattern-recognition devices. This theory is widely-regarded as the first clear step toward theoretical understanding of the range and limitations of these devices. We include here an abstract of a paper to be published in the Proceedings of the American Mathematical Society's 1966 Symposium on Applied Mathematics, and also as a MAC technical report.

"Pattern Recognition", in the context of computational techniques, generally refers to the construction of algorithms to imitate functions of the human visual system. We have been studying its most elementary problem: given "Figures" as subsets of points of a quantized Euclidean plane; decision procedures for membership of mathematically defined classes, or "patterns" (i. e., triangles or connected figures) are considered.

Vast numbers of published papers have described ad hoc experiments using various decision procedures. The majority use linear discriminant functions, either explicitly or implicitly. Our first aim is to provide a rational measure of the complexity of patterns with respect to such algorithms. This leads to a mathematical explication of appropriate sense of "local property": convexity and triangularity are local in that the corresponding decision procedures can be expressed linearly in terms of functions of small numbers of points. Fairly elaborate mathematical arguments are necessary to prove that connectivity is not local in this sense, and (perhaps surprisingly) that the disjunction and conjunction of local properties are not necessarily local. These results are closely related to the more general (and more urgent) problem of constructing a mathematical theory to formalize the intuitive distinction between "serial" and "parallel" computation. Although we have not, so far, been able to find the proper general concept, we have analyzed some special cases that seem instructive and that show parallel computation is not as universally powerful as some are tempted to believe. We also deduced some incidental corollaries relevant to the theory of "learning machines" (such as perceptrons). In particular, we qualify the well-know perceptron-convergence theorem: the time needed for convergence, and the size of the coefficients, can become so large that it would be easier and better to store the entire set of figures used to "teach" the machine. Our results are not all negative: we have described some perceptron algorithms for properties which might have appeared to be beyond this scope, and we describe a remarkably economical serial algorithm for connectivity.

<u>CONVERT: A Pattern-Driven Manipulation Language</u> - Adolfo Guzmán

CONVERT is a programming language, resembling COMIT and SNOBOL, which is applicable to problems conveniently described by transformation rules . By this we mean that patterns may be prescribed, each being associated with a skeleton, so that a series of such pairs may be searched until a pattern is found which matches an expression to be transformed. The conditions for a match are governed by a code, which also allows subexpressions to be identified and eventually substituted into the corresponding skeleton. The advantages of the language are that: it allows one to apply transformation rules to lists and arrays as easily as strings; both patterns and skeletons may be defined recursively; and, consequently, the programs may be stated quite concisely. Design of the language was done at the Instituto Politécnico Nacional of México. (See Guzmán, Appendix C.)

An interpreter for CONVERT, imbedded in LISP, has been constructed and runs in the Project MAC 7094. Experimentation with the interpreter and with programs written in the language suggested several changes, which are described in some detail in a recent memo. (See MAC-M-305, Appendix A.)

An improved version of the processor runs in the PDP-6 of the Artificial Intelligence Group, and care has been taken to assure compatibility of both the 7094 and PDP-6 versions of CONVERT.

POLYBRICK: Parallelepiped Pattern Recognition - Adolfo Guzmán

Given a particular class of objects, an interesting problem in pattern recognition consists of the identification, selection, and description of those objects from any others present in a picture or drawing, which may often contain partially hidden bodies. In addition, we want the recognition of such objects to be substantially independent of their position, orientation, or size.

POLYBRICK is a program which finds the three-dimensional parallelepipeds (solids limited by six planes, parallel two-by-two) present in a picture, such as Figure 3. Input data is the picture in an idealized, two-dimensional, symbolic form; that is, a connection matrix plus the two-dimensional coordinates of its points. No noise or perspective deformation is assumed, but "extraneous" (i.e., non-parallelepiped) objects are allowed.

For Figure 3, the program has to determine the number and location of all cubes. POLYBRICK's answer to this example is:

(CUBE 1 is HG) (CUBE 2 is U V S T K)
(CUBE 3 is XW) (CUBE 4 is A Y C B D)
(CUBE 5 is P M N Q)

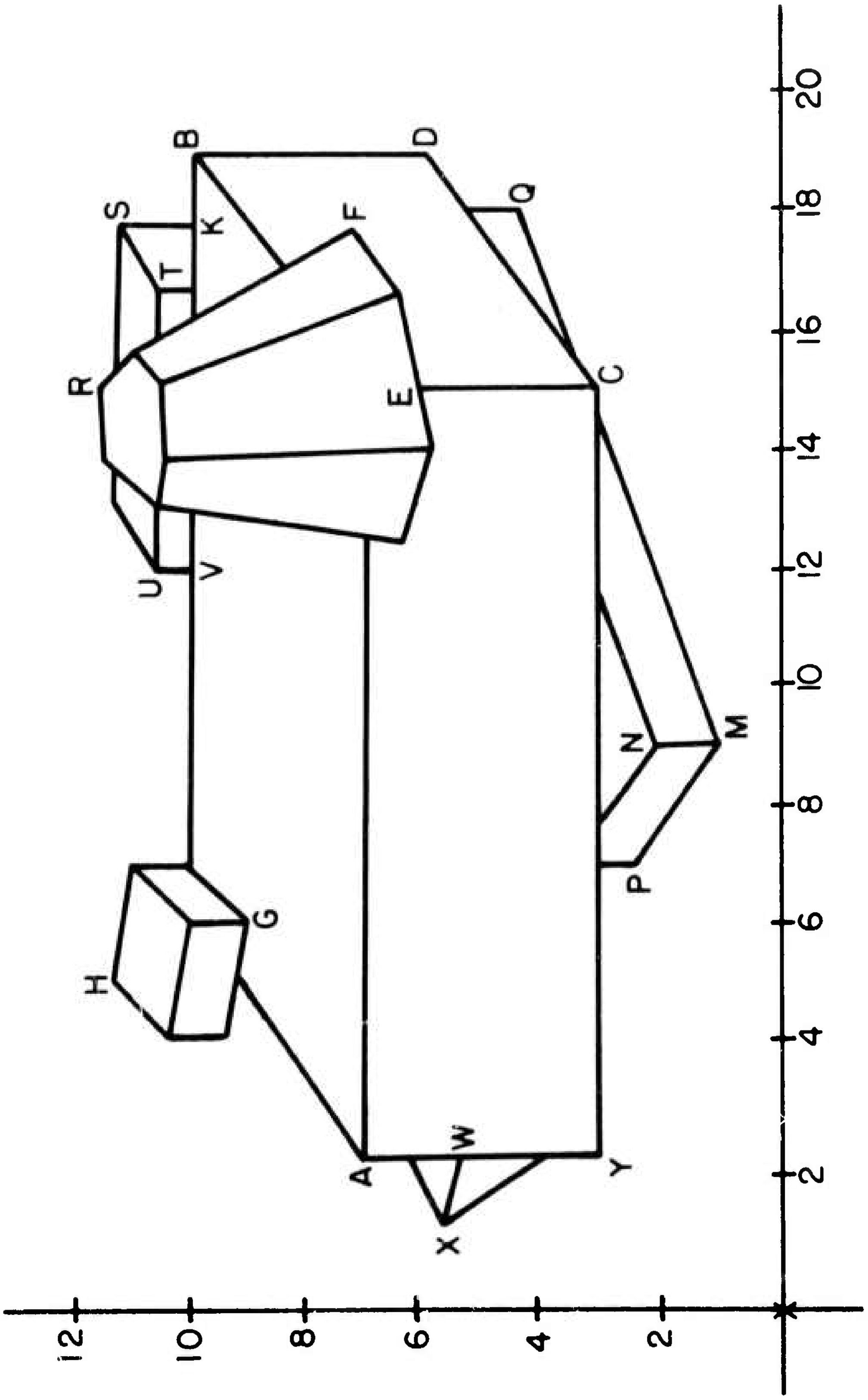


Figure 3. Parallelepiped Patterns

The program has been successfully tested with a variety of pictures, but its answers to optical illusions are somewhat objectionable. A recent memo discusses the method used in some detail and, in the light of its successes and failures, a more general method is proposed. (See MAC-M-308, Appendix A.) The program itself is written in CONVERT, a pattern-driven symbolic manipulation language. (See Guzmán, Appendix C.)

PDP-6 Display Subroutines - Elaine Gord

I have been working on a display program for the PDP - 6 in connection with the intelligent automata project. At present this program will display n (integer) transparent cubes, any of which can be rotated around any of three axes and/or moved anywhere on the face of the scope.

I am currently debugging the subroutines which will also construct tetrahedra and octahedra, remove all hidden lines, and cause the display to be seen in perspective.

Nonlinear Optimization Applied to Checkers - Arnold K. Griffith

Checkerboard positions are characterized by quintuples of parameter values. Each parameter takes on only a small number of values, so that there are only a finite number of possible quintuples. Each quintuple is assigned an index indicating position strength. Plausible board positions are derived from tabulared games of checker masters. Positions resulting from moves actually made in a game are considered strong: those resulting from possible alternative moves are considered weak. The index assigned to a particular quintuple is determined from the relative number of strong and weak positions characterized by the quintuple. A more detailed description of this scoring technique is given in a recent memo. (See MAC-M-299, Appendix A.)

An extension of this technique, not described in the above-mentioned memo, has been partially explored. Twenty-seven parameters, grouped in sets of three, are used to describe a board position. An index is assigned to each possible triplet of values of a set of parameters. Thus, nine such indices are associated with such a given board position. These indices take on only a small number of values, and are themselves grouped in sets of three. The above process is repeated, with triplets of indices taking the place of triplets of parameter values. This gives a board position three more indices, which are added to give a strength evaluation.

The quintuplet procedure was extensively tested on actual checker game positions. It could correctly identify the strongest position from a set of alternatives about twenty-five percent of the time. It proved to be superior to the linear-weighting technique employed by Samuel for the same purpose.*

The Automata Theorist's Helper - William H. Henneman

I have written a set of functions in LISP, for both the time-shared 7094 and the PDP-6, which allow a user to construct examples of finite-state automata and investigate several aspects of their structure.

The finite-state automata may be represented by either a flow table or a regular expression, whichever is more convenient. Presently, there are functions available to do the following things:

*A. L. Samuel, "Some Studies in Machine Learning Using the Game of Checkers", IBM Journal of Research and Development, July 1959, pp. 211-229

1. Calculate the derivative of a given regular expression with respect to a finite regular set;
2. Construct the state graph of a machine from its regular expression representation;
3. Construct a regular expression to represent the machine from its state-graph representation;
4. Calculate the semigroup of the machine;
5. Find all non-trivial subgroups of a semigroup;
6. Tell whether a given group is cyclic, abelian, a direct product, a sub-direct product, etc.;
7. Construct the lattice of ideals in a semigroup;
8. Construct the Krohn-Rhodes decomposition of an automaton into series-parallel connections of pure group and reset machines.

A Syntax-Based Score-Reading Program - Allen Forte

As part of a larger research project in musical structure, a program has been written which "reads" scores encoded in an input language (designed by Stefan Bauer-Mengelberg) isomorphic to music notation. The program is believed to be the first of its kind.

From a small number of parsing rules, the program derives complex configurations, each of which is associated with a set of reference points in a numerical representation of a time-continuum. The logical structure of the program is such that only the defined classes of events are represented in the output.

Because the basis of the program is syntatic (in the sense that parsing operations are performed on formal structures in the input string), many extensions and refinements can be made without excessive difficulty.

The program can be applied to any music which can be represented in the input language. At present, however, it constitutes the first stage in developing a set of analytic tools for the study of so-called atonal music (the corpus of a revolutionary and little understood music, which has exerted a decisive influence upon contemporary practice of the art).

The program and the approach to automatic data-structuring may be of interest to linguists or to scholars in other fields concerned with basic studies of complex structures produced by human beings. (This research will soon be a Project MAC Technical Report.)

Computer Processing of Musical Data - Peter Samson

A PDP-6 computer program is being developed to aid composers and other persons involved in the creation and processing of music. When the program, named BIG, is completed it will have two kinds of routines: those to input and output music in both human-oriented and machine-oriented forms; and those to edit and analyze pieces of music in the computer.

These routines are at the command of the user through the PDP-6 on-line Teletype. The user types in a language whose words, including commands, modifiers, and names of the musical objects in memory, are stored in an internal dictionary, which the user may augment. For each word typed in, the command processor finds best and second-best matches in the dictionary, and computes a "score" for each match. If the best match is perfect, or if its score is above a certain threshold and a certain margin above the second-best score, then the first dictionary entry is assumed. Otherwise, the word is printed out as ambiguous and the command is not performed. This scheme has proved its worth in practice, since nearly every mistake in typing has been corrected by the command processor, and no mistake has been corrected erroneously.

The input and output routines handle the following forms and media: organ keyboard input, electronic organ output, six-voice square wave output (externally filtered), four-voice complex-wave output (weighted sum of harmonics), punched-paper-tape input (in language derived from musical notation), DECtape input and output, and musical scores on the display scope. Also, a composing routine written by Stephen Smoliar is treated as an input function by BIG. All of these features are debugged and in use, except for the organ output whose hardware is not yet completed, and the musical score display, which is an integral part of the editing functions.

Some of the editing commands will have close analogs in text-editing programs: searching for a given string of notes, proceeding to a numbered measure or line, and inserting and deleting material. Others will resemble operations in symbolic-manipulation systems: pattern matching and re-lacement, naming of subexpressions, and copying and rearranging portions. And some will perform operations unique to music: for example, transposition, part doubling, and perhaps simple harmonizing. At each step of the editing, the user will get feedback in the form of a scope diaplay.

System editing features have not yet been written, and development of the system is momentarily dormant, because currently active portions fill the 16,000-word PDP-6 core memory. Work on BIG will resume when additional memory arrives in July, 1966.

Some Experiments in Computer Composition of Tonal Music - Stephen W. Smoliar

For approximately one year, I have been conducting experiments with the MAC PDP-6 in an attempt to write a program which will generate tonal melodies. The current results to date consists of a program, described below, which, according to Allen Forte, simulates with a fair degree of authenticity the modal forms of twelfth-century plainsong. Spontaneous results from this program are accessible through BIG, Peter Samson's program for computer processing of musical data.

The program, named MELGEN, produces melodies using the tetrachord as the fundamental melodic element. A tetrachord is a diatonic progression of four tones within the interval of a perfect fourth; the stem "chord" does not imply that these four tones are to be sounded together, it merely refers to the set of those four tones. MELGEN employs major tetrachords, such as are found in the first four tones of any major scale: e.g., C - D - E - F.

The current copy of MELGEN produces four-measure melodies in 4/4 and 6/8 meter, and eight-measure melodies in 2/4 and 3/4 meter. The rhythmic pattern for a single measure is determined by probability tables, which allow for rhythmic variety, but are strongly biased against complex syncopations. This pattern may be repeated through several measures in the melody. The corresponding melodic notes for a measure of rhythm are selected totally at random; however, since the domain of selection is restricted to the major tetrachord, the results are fairly conventional. Adjustments are included so that the melody ends on a strong rhythmic beat, and the first and last notes are the uppermost tone of the tetrachord which, from music theory, turns out to be the principal tone (or tonic) of the tetrachord. When a rhythmic pattern is repeated, the corresponding melodic pattern may either be repeated or altered in one of five ways:
1) Each note may be raised one scale degree, if the result remains within the domain of the tetrachord; 2) Each note may be lowered one scale

degree, if the result remains within the domain of the tetrachord; 3) The melody may be rewritten in retrograde, i.e., backwards; 4) The melody may be rewritten in inversion, i.e., the first and fourth tones are swapped and the second and third tones are swapped; 5) The melody may be rewritten in retrograde-inversion, i.e., reversed melody with tone swapping. When a new rhythmic pattern is introduced, the old one is discarded and a new measure of melody is written. Examples of the original MELGEN are shown in Figure 4.

Attempts were made to have MELGEN produce several phrases in the above manner using different tetrachords, but the results tended to be long, rambling, and musically uninteresting. Currently, the tetrachord approach is being abandoned in favor of a theory involving chords of simple harmonic progressions; this may tend to produce memodies which are tonal rather than modal.

ADEPT: A Heuristic Program for Proving Theorems of Group Theory - Lewis M. Norton

A program, named ADEPT (A Distinctly Empirical Prover of Theorems), has successfully proved almost 100 theorems from the abstract theory of groups, and a number of heuristics have been tested using it. Numerous insights have been gained about the difficulties of attempting to prove group-theoretic facts by computer.

An executive routine has been written which uses the basic program to establish isomorphisms. There is a subroutine which, given two sets as input, generates a canonical map between them (if possible), and then the executive calls for proofs that this map is well-defined, homomorphic, onto, and one-to-one. In this way, a proof that $A/B \approx (A/C)/(B/C)$ has been obtained using about six minutes of CTSS computer time. This is the most impressive result obtained by the system. It is assumed by the executive that the problem is well-posed; in fact, ADEPT can and has proved the justifying result that if B is normal in A, the factorgroup B/C is normal in A/C.

Figure 4. Examples of MELGEN Melodies

ADEPT has been written using a fresh viewpoint. It does not start with a particular logical system or so-called "complete" proof procedure, as did much previous work in theorem-proving. Instead, possible procedures which a student might use have been considered, in an attempt to encode these methods and heuristics into an algorithm which can be run on a computer and appear to have some degree of "awareness" of where it is in a proof. The result is a program which: 1) is not logically complete, 2) enables the computer to produce proofs for theorems of greater complexity than those done by any previous programs reported in the literature, and 3) constructs proofs which are often quite like those which a student would give (i.e., the development of the proof progresses in a "reasonable" manner).

This research, also reported in Project MAC Progress Report II, is near completion, and will soon be presented in a doctoral dissertation and as a Project MAC Technical Report.

LISP Maintenance - Joel Moses

We have written a new version of the LISP system for Project MAC. The new version provides additional data storage and several new functions and constants. The I/O capabilities, EXCISE, the error comments, and several routines have been improved. Much irrelevant code and many bugs have all been removed.

Source decks and BCD listings of the new version are available. The decks are organized to ease the job of assembling private LISP systems in which unneeded features are absent. For instance, without reassembling, a user can create a private LISP system in which the data storage space has been arbitrarily allotted among binary program space, the push-down list, full wordspace, and free storage. (See MAC M-296, Appendix A.)

Manipulation Of Algebraic Expressions - Joel Moses

In a recent paper, we present and evaluate a symbolic method, due to L. H. Williams, * for solving systems of polynomial equations. The method was found to be very effective for small systems, where it yielded all solutions, without the need for initial estimates. However, by itself, the method appears inappropriate for solving large systems of equations, due to explosive growth in the intermediate equations, and the hazards which arise when coefficients are truncated. The paper also contains a discussion of recent results of Daniel Richardson, showing the recursive unsolvability of the symbolic integration and matching problem. (See Moses, Appendix C.)

We have also written a program which is capable of integrating all but two of the problems solved by Slagle's Symbolic Automatic INTegration (SAINT) program. In contrast to SAINT, it is a purely algorithmic program and has frequently achieved running times that are two or three orders of magnitude faster than SAINT.

The program uses a routine, for matching algebraic expressions, which is capable of performing the necessary pattern recognition to transform the equation

$$a \sin^2 (b) + a \cos^2 (b) + e = a + e$$

A new heuristic for integration, used in the program, is called the EDucated GuEss (EDGE) heuristic. It is claimed that this heuristic, with the aid of a few algorithms, is capable of solving all the problems unsolved by the basic program. The heuristic is an extension of the method of integration by parts. It makes guesses as to the form of the integral, obtains the integral by differentiating the form, and codes the relations between the derivation and the intergral. (See MAC-M-310, Appendix A.)

* L. H. Williams, "Algebra of Polynomials in Several Variables for a Digital Computer", Journal of the Association of Computing Machinery (January 1962), vol. 9, pp. 29-40

Computer Experiments in Finite Algebra - W. Douglas Maurer

The computer system described in Progress Report II, has been more than doubled in volume. It was presented at the April 1966 Symposium on Symbolic and Algebraic Manipulation in Washington, D. C., and will be documented in the August 1966 issue of the Communications of the ACM. Among the capabilities which have been added are:

1. Input and output of single tables to a remote console or to the memory in S-expression format;
2. Input and output of table files to the disk memory in S-expression format;
3. A routine which determines, for any two semigroups, whether they are isomorphic, and if so, constructs an isomorphism (revised and shortened);
4. Input and output of polynomials, product and quotient of polynomials, a divisibility test for polynomials, and an irreducibility test for polynomials of degree 5 or less;
5. Input and output of permutations and sets of permutations; generation of semigroups of permutations from sets of permutations, and of the Cayley table of a semigroup of permutations; input of a state graph as a set of permutations of the states;
6. Adding a zero or unit to a semigroup, or constructing left or right coset maps of a group, or determining whether a semigroup has a zero, or how many idempotents it has, or how many elements it has of a specific order;
7. Dynamic loading and unloading of routine directories, and automatic introduction of new entries in a routine directory;
8. Dynamic loading of routines when and only when requested, and adjustment of memory size for data by reassembly of a single 20-instruction FAP program.

Two major developments are incomplete, although well under way, at this writing. The first is an inferential compiler, which accepts a source language of mathematical statements and proceeds to compile code to verify the statements over a file of special cases. This has been coded, but has been found impractical within the confines of the April 1966 system (ALGEBRA II). The second is a complete revision of this system, to be known as ALGEBRA III. It is being coded in AED, rather than MAD, and features list processing, the ability to handle rings and fields, dynamic loading and unloading of routines, the ability to handle several routine directories, and a greater profusion of data types (each of which has a type number that may be referenced by routines). As this is written, 26 AED routines have been written to handle 30 separate functions.

PILOT: A Step Toward Man-Computer Symbiosis* - Warren Teitelman

PILOT is a programming system constructed in LISP. It is designed to facilitate the development of programs by easing the familiar sequence: write some code, run the program, make some changes, write some more code, run the program again, etc. As a program becomes more complex, making these changes becomes harder and harder because the implications of changes are harder to anticipate.

In the PILOT system, the computer plays an active role in this evolutionary process by providing the means whereby changes can be effected immediately, and in ways that seem natural to the user. The user of PILOT feels that he is giving advice, or making suggestions, to the computer about the operation of his programs, and that the system then performs the work necessary. The PILOT system is thus an interface between the user and his program, monitoring both the requests of the user and the operation of his program.

* This research will soon be a Project MAC Technical Report

The user may easily modify the PILOT system itself by giving it advice about its own operation. This allows him to develop his own language and to shift gradually onto PILOT the burden of performing routine but increasingly complicated tasks. In this way, he can concentrate on the conceptual difficulties in the original problem, rather than on the niggling tasks of editing, rewriting, or adding to his programs.

PILOT is a first step toward computer systems that will help man to formulate problems in the same way they now help him to solve them. Experience with it supports the claim that such "symbiotic systems" allow the programmer to attack and solve more difficult problems. (See Teitelman, Appendix B.)

On-Line CTSS Instruction - Oliver G. Selfridge and J. Anthony Gunn

An initial system has been designed and constructed to aid a naive user in acquiring practice in using CTSS at Project MAC. The user is guided through a number of primitive CTSS Commands with appropriate responses; including login, logout, listf, printf, ttpeek, and others. Then he is led into the first steps of using these commands in the context of running and editing simple arithmetic programs.

We believe such a system should be used, ideally, in conjunction with a beginner's manual; that is, one written with the naive user in mind. At present, there is no specific user's manual for CTSS on Project MAC, though some sections of the Programmer's Guide are very relevant and useful, such as Section AA. 2.

We believe that the techniques being explored here are a set of the most natural ways to use time-sharing for on-line instruction. The chief idea is to give the user experience in using the system, while protecting him from his errors and explaining how to correct them, so that he can learn as fast as possible how to proceed on his own. Figure 5 is a small piece of a sample session.

```
r login a
W 1739.6
*TYPE YOUR LAST NAME PLEASE
*HIT THE CARRIAGE RETURN AFTERWARDS
gunn
*THANK YOU




 LOGIN PLEASE.
READY.

login t666 pupil
*YOU HAVE TYPED YOUR NAME OR PROGRAMMER NUMBER WRONG.
*THE SYSTEM WOULD TYPE

W 1740.2
Password
*BUT WHATEVER YOU TYPED THEN, IT WOULD SAY
  T666 PUPIL NOT FOUNO IN DIRECTORY.
 LOGIN COMMANO INCORRECT.
R 1.250+.950


*YOU MUST START LOGGING IN ALL OVER AGAIN.
*TYPE EXACTLY -
login t263 gunn


 LOGIN PLEASE.
READY.

login t263 gunn
W 1740.9
Password
 T0263  4278 LOGGED IN  09/07/66   1741.0 FROM 20000E
 LAST LOGOUT WAS 08/26/66  1712.4
R 3.033+.516


*SUPPOSE YOU HAO TYPEO THE WORO 'LOGIN' WRONG,BY TYPING, SAY, 'LGIN'
*THE SYSTEM WOULO HAVE SAID -
 LOGIN PLEASE.
READY.
*ANO YOU WOULD HAVE HAD TO LOGIN ALL OVER AGAIN



*SUPPOSING YOU HAD TYPED YOUR PASSWORD WRONG.
*THE SYSTEM WOULD HAVE REPLIED -
 PASSWORD    NOT FOUNO IN DIRECTORY.
 LOGIN COMMAND INCORRECT.
R 1.383+.716
*ANO YOU WOULO HAVE HAD TO LOGIN ALL OVER AGAIN


*NOW LOGIN AGAIN AS YOU OID BEFORE
```

Figure 5. Simplified On-Line CTSS Instruction

Academic Staff

C. Levinthal

Non-Academic Research Staff

H. G. Murray, Jr.
S. A. Ward

Research Assistants and Other Students

W. Brody
A. Pawlikowski
M. Zwick
J. L. Sussman

Guests

Robert Langridge - Harvard University

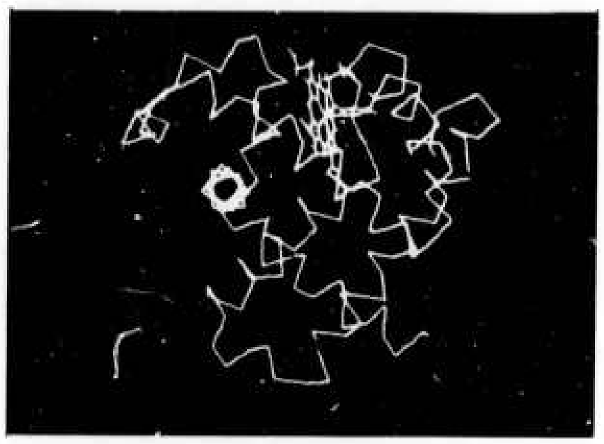
Andrew W. MacEwan - Children's Hospital Medical Center

Molecular Model Building - Cyrus Levinthal

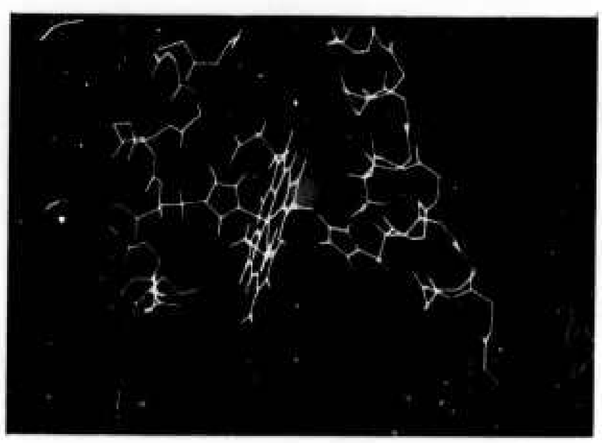
During the past year, our work on molecular model building had two objectives. First, we have been developing the programs to make them more affective and more convenient. And second, we have been using the programs in their present state in attempts to predict the three-dimensional arrangement of the atoms in a protein from the known chemical composition of the molecules. We are using known molecular structures, such as myoglobin and lysozyme, as the basis for a set of construction rules. Though the system is programmed to keep track of known forces between the atoms within a single molecule, the interaction between various parts of the structure and the surrounding water are put in among this set of rules which the investigator should follow as he manipulates the structure. However, the program is designed in such a way that a user sees his error on the oscilloscope display whenever he violates one of the established rules. The investigator can use the information he obtains from the oscilloscope screen to decide how the structure should be changed, and by means of simple teletypewriter commands he can produce changes in the computer program.

In general, we have used our ability to interact with the computer program to simulate the process of building molecule models by hand. However, the computer-aided system has many advantages over any system of mechanical model building. We can combine the advantages of each of the various types of mechanical models, as well as the ability to visualize any internal part of the structure and its relationship to the whole molecule. For example, Figure 6 shows a sequence of myoglobin displays; each more detailed than the last. The (a) frame show the central heme group, and the backbone structure with every ninth atom connected. The (b) frame shows the same structure, this time connecting every third atom in the backbone. The star-shaped cluster at the left of the frame is an end view of the backbone following an α-helix. And the (c) frame shows approximately one-fifth of the myoglobin molecule, as a (5A°) cube around the heme group. This last frame is obtained by pointing the light pen of the ESL Display Console at the heme structure and requesting the computer to display this region in detail.

(a) Heme group and every ninth atom



(b) Heme group and every third atom



(c) 5 A° volume around heme group

Figure 6. Displays of a Myoglobin Molecule Structure

One histidine side chain (a pentagonal shape and its connectors to the backbone) is shown binding the heme group on the left, while the other histidine (right side of the picture) is available for binding. The central region is shown in full detail, while the periphery of the display contains less detail. [We are indebted to J. C. Kendrew and H. Watson for the coordinate data from which this structure is displayed.] Figure 7 is a simplified lysozyme molecule display, showing a cleft (bottom of the picture) where an enzyme which acts on a substrate molecule would fit. The star-like configuration at the top of the figure is an axial view of an α-helical portion of the lysozyme backbone. [We are indebted to Prof. D. C. Phillips of Oxford University, England, for the coordinate data from which this structure is displayed.]

Whenever any procedure for altering the structure has been tested by means of manually introduced changes, it is then possible to alter the program so that the computer can carry out the sequence of changes automatically. The basic program guarantees that we satisfy all of the chemical constraints, as well as any rules which have been obtained from chemical or structural studies with other molecules. The system for model building is being extended so that we can also study the interactions between molecules of a crystal, and between an enzyme molecule and its substrate.

In addition to the alterations and expansions of the system, we are using the present version as an aid in attempts to predict a three-dimensional structure of a protein from its known chemical composition. This work is being done with the protein cytochrome C, a relatively small protein for which there is a vast amount of chemical data available from many laboratories. We have tried to make use of the differences and the similarities which have been observed when this molecule is extracted from different animals. The sum total of all of this information, as well as the information which results from chemical alterations of the structure, allows the formulation of a large number of rules which any three-dimensional model of this structure must satisfy. Using these rules we have obtained a tentative prediction for the structure of the molecule, but we do not as yet know whether the set of rules

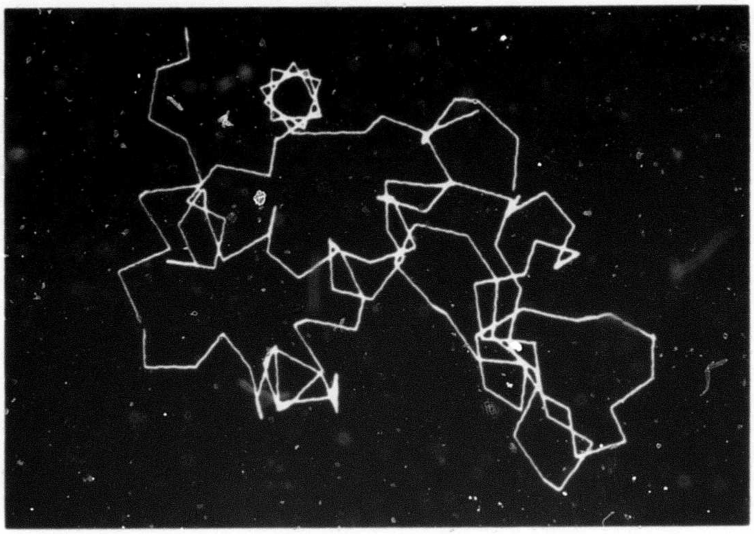


Figure 7. Display of a Lyzozyme Molecule Structure

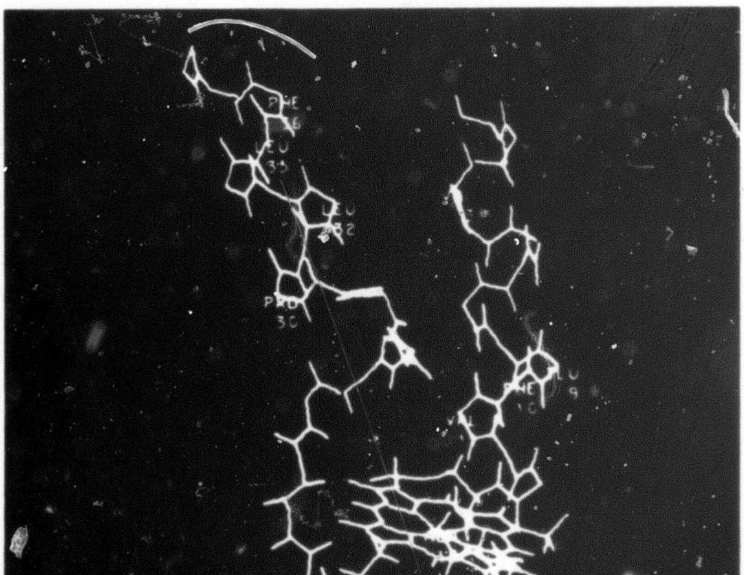


Figure 8. Display of a Cytochrome C Molecule Structure

used is sufficient to make our model unique. Figure 8 is a guess at the possible structure of cytochrome C, showing a different display method. The myoglobin was shown as an entire molecule and as a detailed area around a particular atom: the cytochrome C is shown in a linear display, from amino acid 3 through 33, with special-interest hydrophobic amino acids numbered and labeled. This display shows approximately one-third of the molecule, with the backbone wrapped around the heme group. The backbone is in an α -helix configuration, except in the vicinity of the heme group, and no side chains are displayed. The final test of our predictions must rely on the X-ray crystallographic work which is currently being done on this molecule by Dr. Richard Dickerson and his collaborators at the California Institute of Technology.

We are also trying to utilize "incomplete" crystallographic information (intensities without phases) advantageously. Standard X-ray methods make no assumptions about protein stereochemistry until the final stages of structure solution; we do, however, know a great deal about the configurations of groups of atoms in proteins, particularly when they occur in large sub-structures (such as alpha helices) in advance of any such solution. It may be possible to solve for the positions of these large groups in the unit cell without phase information. This would provide further limiting constraints on our model-building efforts and may be a useful supplement to standard crystallographic techniques.

Since our procedure still involves a large amount of manual manipulation by the operator, we cannot use the structures of the few known proteins as tests for our procedures of predicting an unknown structure. All of the rules we impose must, of course, be consistent with the known structures, but the only way of establishing the validity of these rules for predictive purposes is to make the prediction and then see if it is verified by further experimental work.

BLANK PAGE

CIVIL ENGINEERING DEPARTMENT

Structural Design Language
Construction Cost Estimating and Accounting
Critical Path Scheduling
Stability Analysis by Limiting Equilibrium
Automatic Flow-net Design
Settlement Problem-Oriented Language

Academic Staff

D. Beltran-Maldonado
J. M. Biggs
F. F. Brotchie
A. G. H. Dietz
A. M. Hershdorfer
K. Hoeg
T. W. Lambe
W. H. Linder
R. D. Logcher
W. M. Pecknold, III
F. E. Perkins
K. F. Reinschmidt
J. M. Roesset
D. Roos
R. L. Schiffman
G. M. Sturman
J. M. Sussman
R. A. Walter
J. R. Walton
R. V. Whitman

Non-Academic Research Staff

R. V. Goodman
J. M. Hodnick

Research Assistants and other Students

T. H. Asselin
W. A. Bailey
L. N. Beckreck
A. J. Bodhe
J. R. Brach
R. L. Daniels
R. C. Feldman
A. J. Ferrante
R. Fidler
E. J. Hall
J. C. Jordan
S. Lipner
W. K. MacLean
B. J. McKenna
R. McPhail
J. A. Rionda, Jr.
R. A. Schlumff
B. Schumacker
J. G. Schwartz
L. C. Teague, Jr.
J. W. Weber

Structural Design Language - J. Melvin Biggs, Robert D. Logcher, Richard V. Goodman, Gerald M. Sturman, Edward J. Hall, Ankoor J. Bodhe, Shing C. Chan, Augustin J. Ferrante

The initial development of the STRUctural Design Language (STRUDL) was accomplished during fiscal year 1964-65 and was covered in Progress Report II. This development is also being sponsored by the M. I. T. Inter-American Program in Civil Engineering.

STRUDL is a large-scale computer system designed to aid a structural engineer throughout the design process. Its most important characteristic is a high degree of man-machine interaction, and complete flexibility of use, which is accomplished by a modular structure. This structure consists of a large number of subroutines, each representing an operation within the engineering design process, which may be called by the user in any order and executed within any constraints that have been momentarily imposed.

During the past year, the initial version of STRUDL has been extended and improved. Of primary importance is an extension of the dynamic memory allocator, to permit reorganization using secondary storage as a logical extension of core. This permits the solution of much larger problems than was previously possible. Data retrieval capabilities have also been expanded: a single function has been developed to process data by reading a disc file, which contains information concerning the location of the data requested.

The PRELIMINARY ANALYSIS command has been fully implemented, and enables an engineer to make an approximate analysis of a structure, based on behavioral assumptions which are input at the console. The subcommand PORTAL has also been designed and fully implemented, and contains a standard set of assumptions commonly used for the approximate analysis of building frames subjected to horizontal loads.

Of considerable significance is the past year's experience gained in the practical use of a computer-aided design system. During this time, STRUDL has been used by students and staff for practical engineering purposes, and has also been used for classroom demonstrations of engineering design procedures. In addition, demonstrations have been given to practicing engineers and their reactions have been recorded. All of this experience has reinforced the previously held opinion that flexible, comprehensive design systems, operating in the time-sharing mode, produce a decided increase in the design capability of an individual engineer. It appears certain that this concept will have a profound influence on the practice of engineering.

Construction Cost Estimating and Accounting - Robert L. Daniels

I have been attempting to consolidate many of the routine activities performed by the office staff of a construction company into an integrated time-shared computer system. The main effort was expended on construction estimating and accounting. The system uses, as its building blocks, a set of accounts that are normally encountered in building construction. With the use of specially designed "time and materials-installed" reports, the system creates and maintains a basic set of cost and time information which is used in pricing and scheduling future work. An important influence in the design of the cost library, where the set cost and time information is stored, has been the storage of data useful in the Critical Path Method (CPM) of scheduling. As a useful by-product of the accumulation of information necessary for estimating and scheduling, daily and weekly payroll calculations are issued. Also, the accounting of moneys spent is kept and output at the user's request.

The system is immune to changes in labor rates, as it only maintains crew composition and production rates in some relevant quantity for each of the construction accounts. No attempt was made to include pricing of materials and equipment on the initial system. Estimating is done through input of dimensions of the various components of the building. Quantities are computed and stored in the accounts for summation at a later time.

As an estimating tool, the system is an interacting "partner" of the estimator, performing his computations and bookkeeping and keeping him informed of all relevant information developed since the start of the job at hand. The capability for estimating of alternatives has been included. Much work will have to be done before the system becomes operational. (See Daniels, Appendix B.)

Critical Path Method Scheduling - Thomas H. Asselin and Robert C. Feldman

An extension to the Critical Path Method (CPM) system, the "CPM Bar Chart", was developed during the reporting period. This graphic CPM schedule combines advantages of two project-scheduling techniques, the bar chart and CPM.

A CPM Bar Chart can be generated in either of two modes. In the time-sharing mode, the Civil Engineering Systems Laboratory IBM 1620 computer is used as a remote terminal to the Project MAC IBM 7094 computer. After the early start, early finish, and late finish of each project activity have been calculated by the IBM 7094, the user can request the plotting of a CPM Bar Chart on either the Calcomp Plotter Model 565 or the Gerber Verifier Plotter, both of which can be operated on-line with the IBM 1620 computer. The batch-processing mode allows independent calculation of the scheduling times, which are then input on cards to the IBM 1620. The CPM Bar Chart is then generated in the same manner as in the time-sharing mode. (See Asselin, Appendix C.)

Research during the past year has resulted in the development of a heuristic technique which allows the use of linear convex and non-convex activity time-cost curves in dynamic network scheduling. The technique is designed as a substitute for the present dynamic network-analysis capabilities of the Critical Path Method, which are not directly applicable to this more general problem.

A number of computer programs have been developed to implement this approach. They provide the capability to perform dynamic network analysis, plot project time-cost curves on the 1050 console, and present scheduling information as desired by the user. Advantage has been taken of the man-machine interaction made possible by a time-shared computer. Thus the user can exercise his judgement at each stage of the solution process. (See Feldman, Appendix C.)

Stability Analysis by Limiting Equilibrium - William A. Bailey

STABLE (Stability Analysis By Limiting Equilibrium) is a system designed to perform stability analyses of natural or man-made earth slopes. The user is essentially unrestricted in describing the slope geometry, ground water pressures, and properties of soil comprising the slope.

In limiting-equilibrium analysis, a failure mechanism is assumed and a set of forces computed that will satisfy the equations of static equilibrium. In general, the number of unknowns exceeds the number of available equations and additional assumptions have to be made to proceed with the stability analysis. A simplified method, which gives good results for circular slip lines, has been suggested by Dr. A. W. Bishop of the University of London and is included in STABLE. Since the set of assumptions used is built into the method, this part of the system is suited for batch processing.

More sophisticated procedures of analysis allow a more general failure mechanism to be investigated and provide both control of and a means of ascertaining the influence of the various assumptions made in the analysis. Dr. N. R. Morgenstern of the University of London has recently developed a method to make the slope stability analysis more general, and these considerations are incorporated in STABLE. Since this part of the system is designed to allow the user to modify the analysis as a consequence of information gleaned from preliminary computations, this part of STABLE is ideally suited for time-sharing.

STABLE has been used extensively as a teaching aid in soil engineering.

Automatic Flow-net Design - Robert McPhail

In the design and analysis of any earth slope, it is essential that the soil engineer be in a position to predict the ground water seepage pattern and the resulting pore water pressures. The time-sharing system under Project MAC has been used to supplement the classroom discussion of these topics in a course in soil engineering.

Settlement Problem-Oriented Language - Robert L. Schiffman, Jane C. Jordan, and Richard A. Schlumff

The Settlement Problem-Oriented Language (SEPOL) is a system designed to calculate, via time sharing, the magnitude and progress of settlement of an earth mass when the soil surface is subjected to a specified loading. During this reporting period several new capabilities have been added to the previous system, SEPOL I (Soil Engineering Problem-Oriented Language). The present system allows a wider choice of loading types and load configurations, a detailed analysis of the resulting subsoil stresses, and calculations for progress of settlement for a multilayered soil profile.

BLANK PAGE

COMPUTATION STRUCTURES

A General Purpose Macrogenerator

Research in the Theory of Automata

Repeatable Multiprocess Computation

Memory Allocation in Multiprogrammed Computers

Digital Logic Simulation

Development of the PDP-1-X Multiuser System

Waveform Processing

A Table-Driven Compiler Generator

A Table-Driven Syntactic Analyzer

Scheduling and Resource Allocation

Fact Retrieval by Finite-Set Theoretic Models

Automatic Flowcharting

Implementation of a Flowchart Compiler

FLOW-DEBUG: An On-Line Graphical Debugging Aid

Manipulation of Approximating Functions on a Graphical Display

Braille Translation and Interactive Information Retrieval

A Teaching-Machine Simulator

Academic Staff

J. B. Dennis
J. J. Donovan
D. W. Fife
E. L. Glaser
R. M. Gray
F. C. Hennie, III
D. A. Huffman
R. Y. Kain
H. B. Lee, Jr.
C. L. Liu
E. G. Manning
D. Martin
R. McNaughton (Visiting)
T. G. Stockham, Jr.
J. M. Wozencraft

Research Assistants and other Students

W. T. Beyer
A. Buckles
G. J. Burnett
A. A. Bushkin
R. H. Campbell
C. D. Y. Chang
J. E. Davidow
P. J. Denning
D. D. Falconer
R. C. Gammill
H. Hellman
D. A. Lasher
H. F. Ledgard
F. L. Luconi
H. S. Magnuski
R. E. Marks
L. J. Rotenberg
H. S. Schwenk, Jr.
T. P. Skinner
A. A. Smith
E. C. Van Horn, Jr.
E. Westerfield
D. U. Wilde
S. Zucker

A General Purpose Macrogenerator - Christopher Strachey

A macrogenerator, developed in England, has been implemented on CTSS and is available as the public file command GPM. This command is described in Section AJ. 2.02 of the CTSS Programmer's Guide. The program is described in detail in "The General Purpose Macrogenerator", The Computer Journal, October 1965, Vol. 8, No. 3, pp. 225-241; the abstract of which is reproduced below.

"The macrogenerator described in this paper is a symbol string processor, both its input and its output being strings of symbols. It operates by a form of substitution which is completely general in its application in that it is allowed anywhere. The result is a powerful system including such features as recursive functions and conditional expressions which can be implemented with very few instructions.

"Part 1 describes the operation of the macrogenerator and gives some indication of how it has been used at Cambridge. Part 2 contains a sufficiently detailed account of its implementation to make it a relatively simple task to transfer it to any suitable computer.

"Part 2 describes in some detail an implementation of the general purpose macrogenerator described in Part 1. The implementation is based on the use of a single stack and is described both as a series of transformations on the state of the stack, and by a CPL program. Various error checking features are described which greatly simplify the discovery of errors in macro programs."

Research in the Theory of Automata - Robert McNaughton

During the last year, research in this area has resulted in several informal memoranda and drafts. The following is a summary of this written material. Research in this field is not planned for 1966-1967 at Project MAC.

A. TESTING AND GENERATING INFINITE SEQUENCES BY A FINITE AUTOMATON

This paper, originally an informal memorandum, has been published in Information and Control (vol. 9, no. 5, Oct. 1966, pp. 521-530). Given that an ω-event is a set of infinite sequences, the paper defines "regular ω-event" in terms of state graphs. It then proves an analogue to the so-called Kleene-Myhill theorem, to the effect that an ω-event is finite-state if and only if it is regular.

B. FINITE-STATE INFINITE GAMES

This writing was distributed as Computation Structures Memo No. 14 in September, 1965. An infinite game consists of two players playing to infinity, while a finite-state game is one where there are only a finite number of states or configurations possible. A deterministic game is one in which the two contestants play in turn, each with full knowledge of the past history of the game. The winning and losing conditions of a finite-state infinite game are defined in terms of states occurring infinitely often as a result of the play. The paper sets out to prove that, in any deterministic finite-state infinite game, one of the players has a winning strategy, and such a strategy is a finite-state strategy. Thus, if what the paper claims to prove is true, it will follow that for any such game there is a finite automaton (having, in general, more states than the number of states of the game itself) that plays one of the two sides and always wins. Recently, a fallacy in my proof was discovered by Mr. Lawrence Landweber, a student of Professor J. R. Buchi of Purdue University. The error is deep, and I have not yet been able to patch it up. The concepts of this paper are intimately related to those of A.

C. TECHNIQUES FOR MANIPULATING REGULAR EXPRESSIONS

This paper was issued as Computation Structures Memo No. 10, November, 1965, and will also appear in the Proceedings of the Conference on Systems and Computer Science at the University of Western Ontario, London, Canada, September 1966. Various ways of proving equality of regular expressions are described in detail: proof of conversion to state graphs, proof by reparsing, proof by axiomatic method, and proof by manipulating transition graphs. The transition-graph method is most fruitful in situations where one does know exactly which two regular expressions are to be proved equal; such as, when one is given a regular expression and told to find a certain property equal to it. Several difficult examples of proof by transition graphs are given.

D. THE LOOP COMPLEXITY OF REGULAR EVENTS

Released as Computation Structures Memo No. 18, January 1966, this paper will also appear in the Proceedings of the 1966 Symposium on Logic, Computability and Automata, Rome, New York. The loop complexity of a regular event is defined as the smallest star height of all restricted regular expressions representing the event. ("Restricted" means that operator signs for intersection, complementation, and set difference may not occur.) This paper contains results that make it possible to ascertain the loop complexity of certain events. The general problem of finding an algorithm to ascertain the loop complexity of an event remains open. The main technique involves the use of transition graphs.

E. PARENTHESIS GRAMMERS

Written as Computation Structures Memo No. 23, March, 1966, this paper has been submitted for publication in the Journal of the A.C.M. A parenthesis grammar is context-free, and every rule is of the form $A \rightarrow (\Omega)$, where Ω is a non-empty string and contains no parentheses. The language of a parenthesis grammar turns out to be very visibly structured and unambiguous. The main result of this paper is showing that the equivalence problem for such grammars is soluable: there is an algorithm for determining whether two given grammars generate the same language. The algorithm and the proof are quite non-trivial. As such, it is the only example of an equivalence problem having a non-trivial affirmative solution, and hence it is an interesting case history for those interested in equivalence problems in general. The paper leaves several interesting problems still open: e.g., the equivalence problems for certain classes of grammars only slightly more general than parenthesis grammars.

F. ON NON-COUNTING REGULAR EVENTS

Written in collaboration with S. A. Papert, draft material for several sections of this paper have been completed. A table of contents for the envisioned paper is as follows:

1. Introduction
2. Characterization in Extended Regular-expression Language
3. Fundamentals of Algebraic Theory of Machines
4. Monoid Characterization of Non-counting Events
5. Symbolic Logic Characterization
6. Nerve Net Characterization
7. Decomposition Theory for Group-free Events
8. L_c, L_u, and L_{uc}
9. Possible Directions for Future Research

The paper concerns itself with a certain subclass of the class of regular events: the paper as a whole will show that there are several characterizations (2, 4, 5, and 6, above) by proving that these are equivalent. We have decided the term "non-counting" is preferable (for this subclass) to the term "topological", which we had previously used. Briefly, a non-counting regular event is one whose reduced state graph contains no loop that counts any word modulo m, for $m \geq 2$. (Two helpful alternative definitions: an event expressible as a regular expression containing free use of all the Boolean operators, but with star [denoting closure, or iteration]; an event whose semigroup has no non-trivial subgroups.)

At present we have draft material for the first six sections of the paper. The first two sections were written during the previous year and must be extensively revised: sections 3-6 have been written quite recently. Towards the end of the contract year, we discovered a flaw in our proof of Section 7, which we intend to work on during the months ahead.

Repeatable Multiprocess Computation* - Earl C. Van Horn

When a multiple-access computing system executes a program for an individual user, the system appears to the user to have certain characteristics; these characteristics are conveniently discussed as the properties of a virtual, or apparent computer. The present research seeks to help system designers answer the question, "What kind of virtual computer should be provided to a user?" In particular, I have assumed that certain characteristics are desirable in a virtual computer, and I have attempted to show that it is both possible and feasible for a system to provide a virtual computer having these characteristics.

It is assumed that a virtual computer should be able to execute a multiprocess program. A typical multiprocess program directs virtual arithmetic units and virtual data channels to cooperate; i.e., to communicate by both writing into and reading from shared data quantities.

It is also assumed that a virtual computer should maintain a certain relationship between the initial computation state a user specifies and the output symbols produced during a computation begun from this initial state. When a user specifies an initial computation state, he specifies three kinds of information: 1) a multiprocess program, 2) an initial state word for each virtual processing unit whose activity the program directs, and 3) input data to be read during the program's execution.

*This Ph.D. research will soon be available as a Project MAC Technical Report.

The relationship desired between initial computation state and output symbols is expressed in the form of two virtual computer characteristics. The first characteristic is output-functionality, which means that each output symbol produced by the virtual computer during a computation should be a function only of the virtual computer's initial computation state, and not of events that occur because the host computing system is simultaneously providing virtual computers to other users. The second desirable virtual computer characteristic is limit definiteness, which means that the number of output symbols produced by each output device of the virtual computer should similarly be a function only of the virtual computer's initial computation state, provided the user does not prematurely terminate the virtual computer's activity.

In contemporary virtual computers that execute multiprocess programs, skillful programming is usually required to obtain output functionality and limit definiteness, because the relative sequencing of the executions of several virtual processing units is usually affected by the scheduling strategy of the host computing system, and so is usually a function of other factors in addition to the virtual computer's initial computation state. The present research seeks a way of designing a virtual computer so that output functionality and limit definiteness are properties, not of a program, but of the virtual computer itself; i. e., so that these two characteristics prevail for every initial computation state, and hence regardless of programming mistakes or improper input data.

A method has been discovered for achieving output functionality and limit definiteness in a virtual computer, and a proof has been constructed showing that this method does, in fact, achieve these two characteristics. It appears that a virtual computer whose design is based on this method can be both implemented and programmed at reasonable cost.

The nature of the method can be understood using the following analogy. Suppose a particular shared data quantity, (which might be a segment of data) has associated with it a "rubber band". Whenever a processing unit holds the rubber band, the unit may read the quantity. If a processing unit does not "hold" the band, and attempts to read the quantity, the unit ceases operation until it does hold the band. In order to write into the quantity, a processing unit must be the only unit holding the rubber band. A processing unit holding the band, may not only read, and perhaps write, the quantity, but in addition it may execute an instruction that causes some other processing unit to hold the rubber band; and the first unit may also execute an instruction that causes itself to let go of the rubber band.

The above rules are valid for each shared data quantity, separately. The actual method that has been studied is somewhat more general, in that it allows processing units to have several "holds" on any rubber band, or several "negative holds". By considering every data quantity, and each processing unit state word, to be a shared quantity, rules have also been formulated to govern the execution of forks, quits, creation and deletion of segments, and changes in segment length.

Memory Allocation In Multiprogrammed Computers - Peter J. Denning

Preliminary investigations into the problem of dynamic allocation of core storage have been conducted. The problem is: what policy should be used to decide which information is to occupy core memory at any given time, especially when the demand for memory exceeds the supply? Before outlining the proposed policy, we must state briefly the context in which our ideas have evolved.

We suppose that the computing system under consideration is multiprogrammed. The data pertinent to each computation is divided into "segments", each segment into "pages". A segment is defined by the programmer in some manner which appears logical to him. A page is a constant-length set of memory words; it is used as the unit of allocation, storage, and transmission of information by the system; it is invisible to the programmer. It is apparent that memory allocation policies should be concerned with pages rather than segments, since the page is a unit natural to the system hardware. Consequently, the policies we consider fall under the heading of "page-turning". One danger in page-turning is that, if improperly done, it could result in such heavy traffic of pages in and out of memory that the system would become congested. On the other hand, if properly done, page-turning can assure more efficient operation by permitting only pages in current use to occupy memory. The current trend of multiprogrammed systems toward a "Computer Utility" is resulting in computations about which no prior information regarding dynamic memory requirements is available. This means that the system is forced to base allocation decisions only on the behavior of a computation.

One important behavioral aspect of a computation at any given time is its "Working Set of Information". Roughly speaking, this is the minimum amount of data that must be present in main memory to permit efficient operation of a process. It is the set of pages that a process is currently referencing. If the system knew at every time what the working set of each process was, it could insure that the working set was loaded before any processor was assigned to that process. The problem, then, is to measure dynamically the pages belonging to a process working set. A simple method for doing this is to use a sampling interval, T, equal to the swap-in time of a page. All pages referenced by a process during T are assumed to be in the working set and must be loaded before a processor is assigned to that process. Any pages belonging to a process, but not referenced during the most recent sampling interval, are assumed to have fallen out of the working set for that process, and may be moved from main memory. T is measured in process time (for example, memory cycles used by the given process) so that process suspensions for page faults (I/O waits) or other system interrupts do not result in erroneously small measurements of the working set. Information regarding page use can be kept in the descriptors for each segment, so working set measurements do not require excessive housekeeping.

In order to minimize page traffic two things are necessary:

1. Keeping the total system working set (the union of the individual working sets) within specified bounds, so that the total demand for core will not be excessive;

2. Choose for removal from core those pages least likely to be used in the near future. (Such pages belong to processes low in the Ready List.)

It should be clear that whenever a page is referenced, a corresponding "use bit" can be set. These bits are reset by the system every T seconds, as described. Whenever pages must be removed from core, only the descriptors of processes low on the Ready List must be consulted. Whenever a process is about to be run, the "working-set" pages are compared with "in-core" pages; the process is not allowed to run until any discrepancies are resolved.

As a basis for comparison a probabilistic model was constructed, and was used to predict the page traffic into core for each of the following policies:

1. Delete at random;
2. Delete the oldest unused page, that is, the one which has been unreferenced for the longest time;
3. Use the Ferranti ATLAS scheme to detect and predict page-use cycles;
4. Detect Working Sets.

Schemes (1), (2), and (4) require comparable bookkeeping, while (3) uses considerably more. (In the ATLAS computer the computation involved in the allocation decision required almost a full swap time.) For typical parameters, scheme (2) was about twice as efficient as (1); (3) and (4) were an order of magnitude better. Scheme (3) has an additional defect: it will not work, for example, on list-structure programs which do not display any particular page-use patterns. Detection of Working Sets will work for any process which does not change working-set size too rapidly with respect to the sampling interval. Inasmuch as the optimum page-removal decision used information regarding processes in the Ready List, an intimate relationship between Scheduling and Allocation has been uncovered. (See Denning, Appendix B; MAC-TR-21, Appendix D; and Project MAC Computation Structures Group Memo No. 24, March 1966.)

Digital Logic Simulation* - Donald L. Smith

A data structure has been developed for describing complex digital systems for purposes of simulation, design evolution, and logic verification. The principle criteria used in formulating the data structure were: 1) speed of simulation for large systems having asynchronous timing, 2) provision for incremental design changes without regeneration of the entire data structure, 3) ability to fully detect timing hazards if this is desired by the designer.

The basic elements of the data structure represent flip-flops, combinational logic blocks, delays, and events. These elements are tied together in a list structure that facilitates alterations and makes efficient simulation possible. The simulation algorithm queues future events according to their time of occurrence, and evaluates a combinational level only when its value is needed and may have changed since last evaluated.

The proposed data structure is viewed as the basis for a digital logic design system that would be implemented on a time-shared computer, and would include a conversational design translator as well as the simulator and logic checking aids.

* See Smith, Donald L., Models and Data Structures for Digital Logic Simulation, M.I.T. Department of Electrical Engineering, M.S. Thesis, June 1966 (soon to be a Project MAC Technical Report).

Development of the PDP-1-X Multiuser System - Jack B. Dennis and Leo J. Rotenberg

Experience obtained operating the PDP-1 time-sharing system, at the M. I. T. Research Laboratory of Electronics, together with experience at Project MAC, has taught us much about desirable properties of time-shared computer systems, or, in general terms, multiprogrammed computer systems. Features of a Multiprogrammed Computer System, (MCS) are generally evident in the services that an object program may obtain from the system by giving commands that we call meta-instructions. From work done at Project MAC, Dennis and Van Horn have published a set of meta-instructions and defined their meaning in terms of a program structure felt to be particularly suitable for an MCS. Their principal ideas concern parallel processing, relations among computations, and the organization and manipulation of directories of files, etc., retained by the system on behalf of its users. (See MAC-TR-23, Appendix D.)

With the purpose of providing a realization of many of these ideas, a new multiprogramming executive system has been designed around the Digital Equipment Corporation PDP-1 computer, donated to the Department of Electrical Engineering in 1961. The new system involves hardware additions and modifications, as well as preparation of a new executive program. The modified machine will be called the PDP-1-X. The more important services provided by this system are:

1. File Memory

   The PDP-1-X will include a file memory system capable of storing information on miniature tape units (DECtape), so that a user may retrieve programs and data by command from his console. Consistent with the informal way in which this system is operated, a user will ordinarily bring a tape containing his programs with him and mount it on a free unit at the beginning of a session of use, and take an updated tape with him when he leaves.

2. Parallel programming

We have found the notion of process to be more fundamental to the design of an MCS than the concept of program. A process is the sequential execution of the steps of a procedure. One process may, by means of a fork meta-instruction create a second process, which is thought of as running concurrently with the first. The use of a join meta-instruction allows a process to test whether two or more asynchronous activities have been completed. Execution of a quit meta-instruction terminates a process. The fork, join, and quit meta-instructions form one set of primitive operations for parallel programming. In the PDP-1-X, use of these primitives to control concurrent input/output activity replaces the interrupt feature that serves this function on most present machines.

3. Computations

We use the term computation to mean a group of processes that collectively accomplish a computing task. Each computation runs on behalf of a user of the system and has an associated sphere of protection that determines what files, services, or directories the processes of the computation are allowed to access or use. In our revised system, the sphere of protection is established by a 64-entry program reference list through which the processes of the computation execute input/output operations, gain access to directories and files, and obtain other services provided by the executive system. A primary computation will be able to run a second computation under its surveillance. This is a valuable asset in program debugging; it allows a programming system running as an object computation to monitor a second object computation under test, without the danger of being disabled if the test process should run wild or address incorrectly.

4. Directory Structure

The system design includes a hierarchical directory structure. Each item in a directory associates an alphanumeric name with a pointer to a file, an input/output function, an entry point giving access to some system service, or a directory giving further associations of names with pointers. Processes may reference items in a subtree of the directory lattice by specifying a downward path from a directory that is accessible through the capability list of the computation. A system user's root directory is always accessible through the capability list of a computation operating on his behalf. By means of meta-instructions (similar to those defined in MAC-TR-23), processes will be able to share files in a rather flexible way.

The most important innovation of this system is the provision of environment in which parallel programming may be used and developed. Most functions of the executive program are performed by executive processes, scheduled without preferential treatment, that run concurrently with user processes and use the parallel programming facilities for control of input/output functions.

Because of our emphasis and reliance on parallel programming, it is essential that the use of these features involve minimal cost in processing time. For this reason, specially designed process-scheduling hardware is being built into the PDP-1-X, so that a particularly fast implementation of the fork and quit primitives is obtained. There is little bookkeeping to be done by the executive when switching control between processes, and we anticipate switching times in the range of 200 to 300 microseconds. (See also, L. J. Rotenberg, Proposal: An Implementation of an Almost-Segmented Multiprogrammed Computer System for the PDP-1, Project MAC Computation Structures Group Memo No. 19.)

Waveform Processing - Thomas G. Stockham, Jr.

A study of fast-Fourier-transform techniques has resulted in an algorithm for effecting fast convolution and correlation. This work constitutes a major breakthrough in the computation of lagged-products and has had a profound effect upon the cost and character of computations requiring such calculation.

Signal filtering and spectral analysis are processes which play central roles in waveform processing; but they use computer time extravagantly, since they involve the summation of lagged products. The recently disclosed Cooley-Tukey method for evaluating Fourier series has been applied to relieve this drawback. If N is the number of lagged products to be summed and M is the number of lags to be considered, computation is effected in time $T_{fast} \leq K_1 M(\log_2 N + C)$, provided $M \geq N$. The term K_1 is a proportionality factor that depends on the nature of the basic interactive step, the efficiency of the programming language, and the speed of the computing maching. C is a small overhead factor which accounts for data movement and indexing. Normally, the time required is $T_{standard} = K_2 (M \times N)$. In one realization, K_1 has been measured to be 2.6 K_2. Computation times for various N, and quantitative measures of the relative accuracy of the method, are presented in a recently published paper. (See Stockham, Appendix C.)

A program has been completed by H. Ledgard to allow a user of the ESL display console to sketch functions having a prescribed analytic form. This algorithm allows the user to approximate an arbitrary numeric series with segments of smooth curves that can be selected from a broad class. (See Ledgard, this section, and also Appendix B.)

A Table-Driven Compiler Generator - Chung L. Liu and Gabriel D. Y. Chang

Generality has been our primary design criterion, since we do not intend to provide users with a generalized compiler-generator system, but rather to provide an environment within which users can freely design and produce their own compilers. We, therefore, do not limit the users to specific ways of doing syntax analysis, or doing storage allocation, or producing binary programs in our system. What we do provide are mechanisms which we believe are general enough for users to build compilers in their own ways.

The system is divided into three parts; a syntax analyzer, a table-processing package, and an assembler. The syntax analyzer accepts input strings of a source program, recognizes various syntactic types, transmits information to tables in the table-manipulation package (which will be used for storage-allocation purposes), and emits macros that can be interpreted by the assembler. The table-processing package accepts information from the syntax analyzer and puts it into tables. These tables are processed (sorted, merged, and so on) and used for allocating storage space. Utilizing information from the table-processing package, the assembler accepts information from the syntax analyzer and puts it into tables. These tables are processed (sorted, merged, and so on) and used for allocating storage space. Utilizing information from the table-processing package of the object machines, the assembler interprets the macros generated by the syntax analyzer and translates them into machine codes.

So that a minimum amount of reprogramming will be necessary when we move the system to another computer, most of the compiler system is written in the MAD language.

The table-processing package and the assembler of the table-driven compiler system have been designed and implemented, and two input languages were designed. The Table Declaration and Manipulation Language is used to declare both the number of tables and the formats of the tables which will be used by the compiler. The language is also used to declare how these tables should be manipulated (sorted, merged and so on) after they are set up. The Macro Interpretation Language is used to specify how the macros (generated by the syntax analyzer) should be interpreted, for the assembler to generate machine codes. (See Chang, Appendix B.)

A Table-Driven Syntactic Analyzer - Richard E. Marks

Many special-purpose computer languages have been developed: there are languages for civil engineering problems (COGO), business problems (COBOL), list-structure processing (COMIT and LISP), as well as a large number of general-purpose, scientific-algebraic languages. The purpose of this research was to develop part of a computer-software system which, given a set of appropriate instructions, could translate many languages into the proper machine code for any computer.

This software device is, in a very real sense, a computer. It contains a set of relatively general subroutines which, when called in the proper order and when given the proper arguments, can perform virtually all of the tasks necessary to translate one language into another. This software device also contains a number of instruction tables which contain an encodement of a sequence of calls to these subroutines and on encodement of the arguments to use with these calls. The set of subroutines corresponds to the instruction set of a computer, and the instruction tables correspond to the machine instructions. When performing language translations, it is these instruction tables which contain an encodement of the algorithm used to translate the source language into the target language.

Normally, this device is used to translate a problem-oriented computer language into machine code for some computer: however, it could also be used to translate a description of the compilation process (written in some suitable language) into instruction tables (machine code) for itself. Thus this device, besides being used as a compiler in the normal sense, could be used as a compiler for compilers.

Compilers are written in a compiler generation language (CGL). Any CGL and the compiler it produces would most likely be designed to optimally utilize some syntactic-analysis technique. CGL's could be described by a third language -- the bootstrap language -- so that different CGL's, designed for different analysis techniques, could be implemented. This compiler-generator system consists of three basic sections; with a control section and interface routines to allow the three sections to interact. These three sections are a table processor, a syntactic analyzer, and a code-generator. The code generator produces machine code from the output of the syntactic analyzer.

The syntactic analyzer, which includes a lexical-analysis mechanism, was the subject of a thesis. Basically, the syntactic analyzer operates on an input string and produces some contaxt-free structure, such as a tree representation of the input structure; the analyzer consists of thirteen routines, which require 3400 locations of 7094 storage, and the instruction tables, which require 12,000 locations of storage. (See Marks, Appendix B.)

Scheduling and Resource Allocation - Dennis W. Fife

Since January, 1966, a part-time research effort has been spent on the scheduling and resource allocation function of time-shared computers that have the general characteristics of MULTICS. The function may be structured in four parts:

1. System load control - techniques for controlling user demand on the system, such as restricting the number of logged-in users;

2. Scheduling - an algorithm for computing the priority and quantum assignment for each process in the ready list;

3. Memory load control - an algorithm for determining if an unloaded ready process is eligible to be loaded and executed by a processor;

4. Page-Turning - an algorithm for determining the physical pages to be reclaimed in order to satisfy needs for free core space.

The objective has been to gain insight as to the form these algorithms should take. Behind this effort has been the viewpoint that this objective is best obtained through simulation, analysis, and optimization of stochastic models of the queueing phenomena in the system.

The algorithm for scheduling and memory load control has been of particular interest, since its primary influence is on the effectiveness of multiprogramming. An appropriate algorithm should utilize observations of the characteristics which describe memory usage for the processes being scheduled. For example, if it were true that page faults in a process were independent of the number of pages the process had in core, then memory loading ought to depend simply on the number of processes already loaded. But here the algorithm is not trivial, even though the basic assumption is a gross simplification.

This rudimentary example points out the need to develop some characterization of the memory usage of processes. Without a model to guide the choice of algorithms, one may as well use the most easily implemented technique and hope for success. The model must ultimately correlate with monitored behavior of processes in the operating system, but preliminary analyses of the consequences of various models will help develop insight in anticipation of the operational system. (See Fife, Appendix C.)

Fact Retrieval by Finite-Set Theoretic Models - Robert C. Gammill

An experimental list-processing language, called the Set Theoretic Language (STL), was embedded in MAD. The primitive operations of this language are very similar to those of SLIP, and were coded in FAP. The actual form of the lists is similar to LISP.

The purpose of the language is for fact retrieval and manipulation. The system builds up a structure which represents a finite model from set theory. Each list of the structure points not only to its own sublists, but to all lists of which it is a sublist. Each such list represents a set and its relations to the other sets of the model.

The structure is built using the basic operation ASSERT followed by a sequence of closed sentences of the form <a, b, . . . , c> . The model represents the meaning of the set of assertions given. Information can be collected from the model using the ANSWER operation. This operation produces answers to a sequence of open or closed questions (sentences). At the present time, only direct answers can be found. It is planned to develop a proof mechanism so that answers may be drawn from relations between relations of the model. This will allow questions to be answered even though the answer has not been explicitly included in the assertions.

Finite-set theoretic models, of the type built by this system, are extremely useful in representing factual information. This type of information, not normally well-suited to computer manipulation, is non-algorithmic in character, and is usually listed in directories, tables, charts, or graphs. Particular examples of information which is handled well by finite-set theoretic models are:

1. Relationships between people and real-world objects (eg., directories, business information, and family trees) (See MAC-TR-2, Appendix D.)

2. Relationships between the objects of a complex computer system, such as CTSS (objects being programs, commands, files, formats, links, etc.);

3. Relationships between graphical objects, as in computer-aided design. *

* Sutherland, Ivan E., SKETCHPAD: A Man-Machine Graphical Communication System, Lincoln Laboratory Technical Report-296

The SIR and SKETCHPAD computer systems, designed at M. I. T., used finite models as their foundation. SKETCHPAD used a ring structure mechanism, and SIR used the property lists of LISP. The Set Theoretic Language is an attempt to provide a more consistent and theoretically lucid means for describing such models.

Automatic Flowcharting - Daniel U. Wilde

A comparison of the properties of non-modifying and self-modifying programs leads to the definition of "independent" and "dependent" instructions. If a program is non-modifying, the set of all possible outcomes for each instruction is a function of the instruction itself and is independent of all other program instructions. For example, an absolute transfer instruction is "independent", because all of its outcomes are determined by the instruction itself. On the other hand, a tagged transfer instruction is "dependent", because its outcomes depend not only on the contents of its index register, but also on the instructions and data which affected that register.

Because non-modifying programs contain only "independent" instructions, such programs can be analyzed by a straightforward, two-step analysis procedure. First, program control flow is detected; then the control flow is used to determine program data flow (information processing). However, a self-modifying program can also contain "dependent" instructions, and its control flow and data flow exhibit cyclic interaction. This cyclic interaction suggests use of an iterative or relaxation analysis technique. The initial step in the relaxation procedure determines a first approximation to control flow; the second step then finds a first approximation to data flow. These two steps are repeated recursively until a steady-state condition is reached.

Algorithms for implementing the first iteration have been developed and written. These algorithms are capable of analyzing programs which modify both control and processing instructions during execution. In addition, data structures have been devised which permit the construction of functional expressions for the data flow (information processing). These latter algorithms have been used to produce flowcharts of self-modifying programs for test cases. (See Wilde, Appendix B.)

Implementation of a Flowchart Compiler - Eric C. Westerfeld

This work is an implementation of the 1965 Masters Thesis "Flowchart Compiler Using Teager Board Input". The mechanism translates a simulated input from a Teager Board or Rand Tablet (of the text characters and locations, the lines of flow, and the process boxes of the flowchart), and produces hard copy output that is a correct MAD program in flow and context.

Major features of the implementation are those of a network form of list-structure processing in the various stages of translation which include:

1. Flow of boxes and lines;
2. Text in a box, in algebraic format of exponents and subscripts;
3. Card images, associated with the lines and text in the boxes.

The steps involved in the implementation are:

1. Connecting the flow of boxes and lines;
2. Processing the characters (in arbitrary algebraic form) into card images that are compatible with the MAD compiler;
3. Producing groupings of generated MAD coding for a "clean" program output, with as few redundancies in transfers and labels as possible.

The scope of this implementation was by choice limited to MAD. Future usage can allow a special language, and applications in any field using graphic representation of ideas. Figures 9 and 10 are a simulated input flowchart and the resultant MAD program.

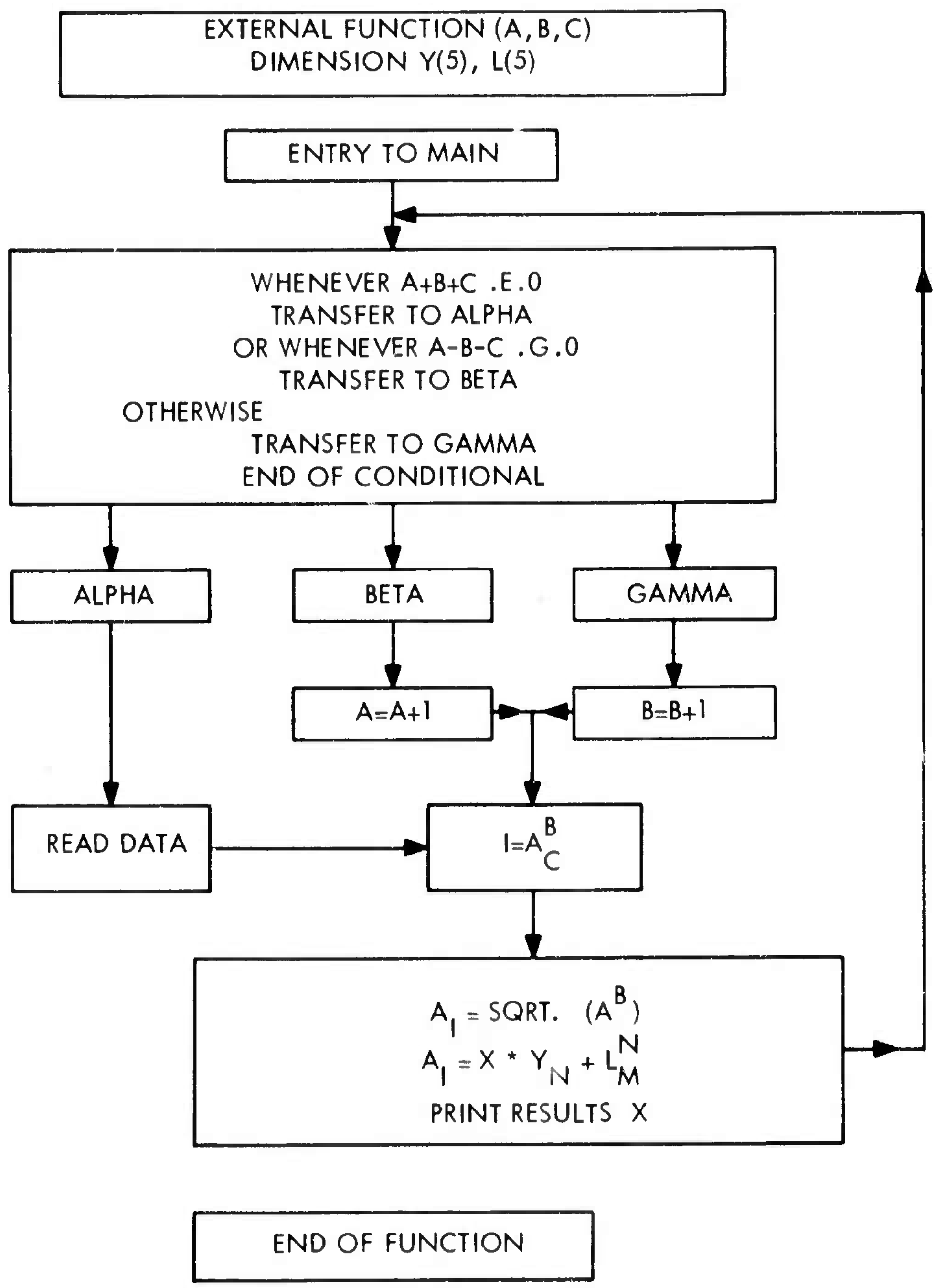


Figure 9. A Simulated Flowchart Input

```
printf flwcht mad
W 1508.5
00100                  EXTERNAL FUNCTION (A,B,C)
00200                  DIMENSION (5),L (5)
00300                  ENTRY TO MAIN
00400    000003
00500                  WHENEVER A+B+C .E. 0
00600                  TRANSFER TO ALPHA
00700                  OR WHENEVER A-B-C .G. 0
01000                  TRANSFER TO BETA
01100                  OTHERWISE
01200                  TRANSFER TO GAMMA
01300                  END OF CONDITIONAL
01400    ALPHA
01500                  READ DATA
01600    000025
01700                  I = A (C) .P. (B)
02000                  A (I) = SQRT. (A .P. (B) )
02100                  A (I) = X*Y (N) + L (M) M .P. (N)
02200                  PRINT RESULTS X
02300                  TRANSFER TO 000025
02400    BETA
02500                  A = A + 1
02600    000021        TRANSFER TO 000025
02700    GAMMA
03000                  B = B + 1
03100                  TRANSFER TO 000021
03200                  END OF FUNCTION
R .933 + .650
```

Figure 10. Flowchart Compiler Output Listing

FLOW-DEBUG: An On-Line Graphical Debugging Aid - Thomas P. Skinner

With increased man-machine interaction, provided by such large-scale time-sharing installations as Project MAC, interactive debugging aids become quite useful. Most of these aids fail in one major respect; they do not provide detailed information while the program is executing. The user merely has access to information at stratigically placed points of program interruption.

The increased popularity of graphic display terminals has suggested investigation into the area of graphical debugging. During the past year, the author developed an experimental system for CTSS utilizing the ESL Display Console. This system, known as FLOW-DEBUG, enables a user to obtain run-time information in the form of a flow chart, so that he can obtain an object-time flow chart of his program. This allows immediate error detection without requiring repetitive program execution.

FLOW-DEBUG is an entire system, but it works as a command to the existing debugging system known as FAP-DEBUG. The flow-chart subsystem shares symbolic information with FAP-DEBUG, so a user may proceed as usual and introduce FLOW-DEBUG as he sees fit. The error-detection system consists of three major parts, which are shown in Figure 11:

1. The INTERPRETER, which runs the program under test and records any rupture in the normal flow;
2. The FLOWMAP string processor, which takes the information from the interpreter and builds a list-structured representation of the program's topology;
3. The MAP routine which takes the list-structured topology and forms the actual display.

The interpreter, which is machine dependent, was written in FAP. The remainder of the FLOW-DEBUG package is written in the AED-0 language and will require minimum reprogramming for future implementations. The

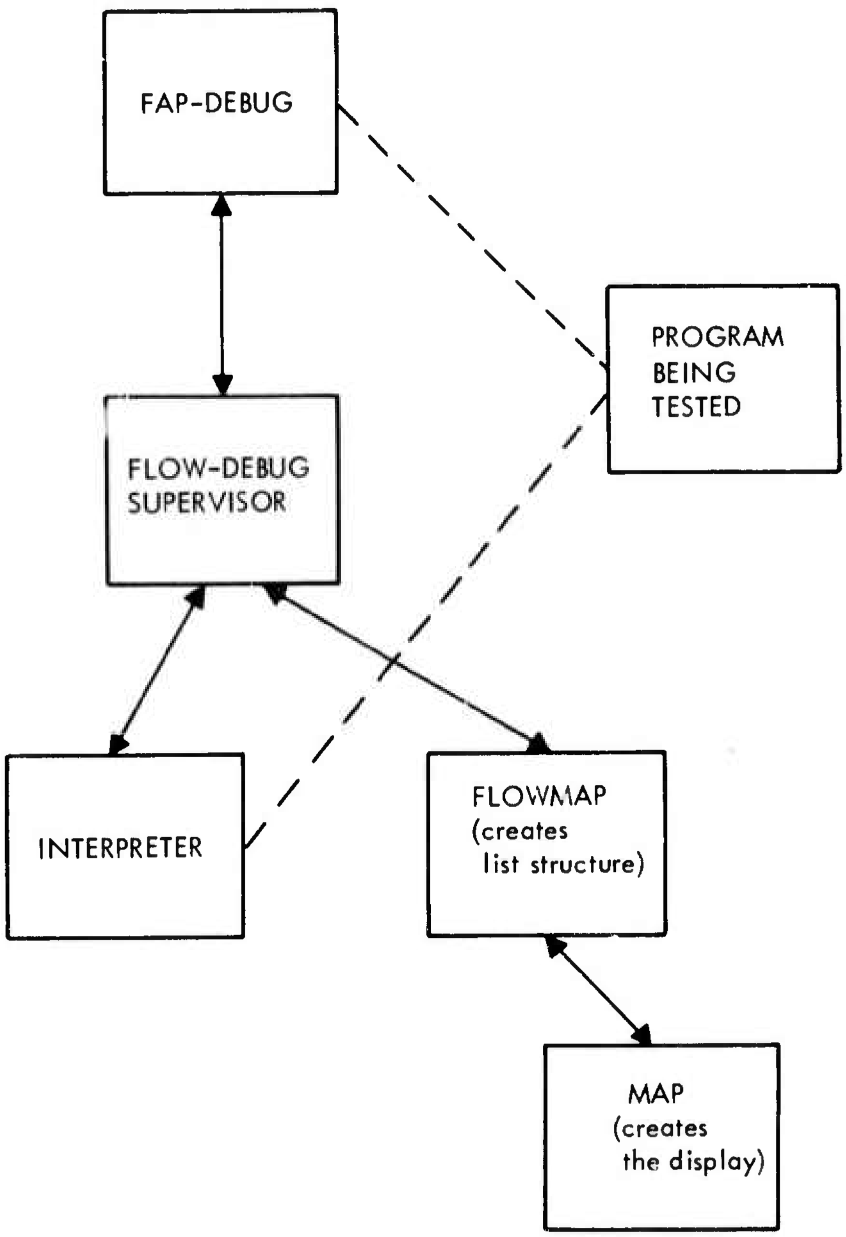


Figure 11. The FLOW-DEBUG On-Line System

display routine is also dependent on the particular display used, but only simple operations, such as line and circle constructs, were used and these subroutines could quite easily be programmed for any display.

While the display obtained was not ideal, the system worked extremely well, considering the limited amount of programming time invested. (See Skinner, Appendix B.)

Manipulation of Approximating Functions on a Graphical Display - Henry F. Ledgard

This work has explored the development of a computer program to provide for the specification and immediate display of piecewise approximating functions through the ESL display facility. A method has been developed for:

1. setting of adjusting points through which an approximating function must pass,
2. setting or adjusting the number of approximating function parts, and
3. setting or adjusting the number of domains of the piecewise approximating segments;

while at the same time

1. maintaining restrictions on the value of approximating function or its derivative,
2. minimizing the error in the last-mean-square sense.

To allow the manipulation and display of functions when no approximation is involved, the method is extended to cases where the function is completely specified by restrictions on its values.

The created program supplements the basic method with the facility to:

1. specify graphically the standard function parts which comprise the approximant, and

2. combine the functions generated under the operations of addition, subtraction, multiplication, and division.

The resultant system allows a user to generate and manipulate a large class of functions through graphical communication. (See Ledgard, Appendix B.)

Braille Translation and Interactive Information Retrieval - Henry S. Magnuski

This work can be divided into two major projects. The first is completion of the on-line Braille translation system, and the second deals with interactive information retrieval.

During the Fall term, finishing touches were put on a system which makes computer-translated text available to the blind. A PDP-8, coupled to the Braille equivalent of a teletype machine (Figure 12), has been used to translate English text into Grade I and Grade II Braille. The translation of English text into Grade I Braille is fairly straightforward, and, with few exceptions such as inserting the sign of a numeral before a string of numbers, simply requires the transliteration of English text characters to their Braille equivalent. Grade II Braille is much more complex, however, and contains many contractions and special rules. The output Brailler generates eight cells per second and produces 38-cell lines of text.

In September of 1965, both the Grade II translation program and the embossing mechanism itself had been completed. The main problems encountered were in the electronics interface equipment between the PDP-8 and the Brailler. The purpose of the interface equipment is to accept embossing commands from the computer and control the mechanical motions of the Brailler so that the embossing is done properly. After the equipment was checked for proper operation, a variety of test and maintenance programs were written to insure correct functioning of the machine. The Brailler is now capable of embossing 10 to 15 characters per second.

Figure 12. Computer-Driven Braille Output Device

In order to make the Compatible Time-Sharing System available to the blind, a program was written which would accept teletype characters from the PDP-8's Dataphone and turn the characters into Grade I Braille. In this application the PDP-8 acts as a small satellite processor, taking the normal output from the 7094 and converting it to a form suitable for blind users of the system. Future plans for the Brailler include the installation of a Braille keyboard, and coding some of the Grade II translation program. (See Magnuski, Appendix C.)

Since January, time has been devoted almost entirely to a study of information-retrieval that is based on active man-machine communications. This work is centered on the idea that man can be very useful in guiding computers to do information retrieval.

Information searching with an active participant can be accomplished several ways. One of the simplest of these is to present a man with a hierarchy of directories of the information, and then let him work his way through the directories until he finds a desired document. This type of search can be done quite easily using a graphic display device and a light pen: as the user makes appropriate selections in his search, he comes closer and closer to the article he is searching for until it finally appears in its full text on the screen. Another method of doing an active search, which is especially suitable when the information being searched has a structure of its own, is through use of an "information structure diagram" which relates a document name to its contents and presents the information in a two-dimensional drawing.

Both of these types of information retrieval require the use of a remote-access computer and a display console. Although they have been rather expensive, time-shared computers and graphic display equipment are now becoming less costly, and the possibility of widespread use of information retrieval and display consoles is increasing. In order to establish how such consoles might be used, a system incorporating the above techniques, and others, was simulated on the PDP-8 with a DEC 338 display. The 338 is a very-high-resolution display, and can support over 1000 flicker-free characters simultaneously. The character generator provided both upper and lower case characters, plus many special graphics. Data for the simulation came from documents describing the MULTICS time-sharing system.

The two main purposes of the simulation were: 1) to outline the needs of users of the MULTICS information system, and 2) to design a set of commands and console operating techniques which would service these needs.

A set of "Page Independent" commands were specified, which are valid no matter where a user is situated in the information-directory hierarchy; and a set of "Page Dependent" commands were specified, which change with each scene being viewed. These commands allow the searcher to jump from one directory to another, "leaf" through documents, browse, and conveniently find his way through the information available to him.

This approach to information retrieval shows promise, since it does not rely on lengthy searches by the computer, and allows a man to use the best of his own abilities in retrieving a desired document. (See Magnuski, Appendix B.)

A Teaching-Machine Simulator - Wendell T. Beyer

While developing a preliminary teaching machine simulator, I wrote three subsystems of independent interest. These subsystems were: a dynamic storage allocation system, for use by MAD programs; a rudimentary character-string-processing language, embedded in MAD; and a program for simulating the action of a BCD file-editing program, such as EDL.

The long-term goal of my project was to construct a program which would teach a novice to use CTSS. Since I had no previous experience with teaching machines, I began by considering the limited problem of teaching a novice to use EDL, one of the BCD file-editing programs.

As part of the teaching process, the student was to have complete control of an EDL-like program. Rather than use EDL and real disk files for this purpose, I constructed a subroutine which simulated the action of EDL on BCD "files" stored in core memory. This method had a threefold advantage. First, no time was spent reading and writing files, since all files were stored internally. Second, closer supervision of the student's operation of "EDL" was possible. Third, the teaching machine script was stored in simulated BCD files, and this storage made possible dynamic editing of the script during the course of an instructional session.

I next designed a simple language for writing teaching machine scripts and constructed a program to implement the scripts. The teaching program was operating by the end of the reporting period, but is still undergoing many changes.

Non-Academic Research Staff

L. A. Berry
W. R. Bjerstedt
M. M. Daggett
R. A. DePrisco
M. G. Gottlieb
L. P. Odland
M. Padlipsky
D. Shea
M. V. Solomita
J. W. Spall
T. H. Van Vleck

Operating Staff

G. Andrews
D. Borsini
C. Brandon
R. Brenner
K. Carley
R. Cerrato
J. Cohan
G. Dardis
T. Dattilo
R. A. Degan
M. Dickie
C. Ferguson
R. G. Hart
J. McGillivary
R. McNamara
R. Moore
S. Nelson
G. Noseworthy
M. Pagliarulo
R. Parker
J. Payson
D. Peaslee
R. Poole
E. Reardon
J. W. Waclawski

Operation of the 7094 Time-Sharing System - Richard G. Mills and Thomas H. Van Vleck

With the installation and shakedown of the so-called DAEMON file-system backup facility, the operation of the 7094 time-sharing system (CTSS) has settled down to a straight-forward, 24-hour-per-day routine. CTSS has been the only regular system run on the computer for well over a year, with the exception of about ten hours per week of scheduled maintenance. Also for about a year the system has been quite heavily loaded, particularly during the day and evening shifts, and lately the heavy-load period has been observed to extend into the first hour or two of the midnight-to-8:00 A. M. shift. Weekend use of the system is now becoming heavy during the afternoon and early evening, with very early morning use (2:00 to 6:00 A. M.) still light.

Reliability of the 7094 hardware has been acceptable. The disk, which is so critical to system operation, has been unexpectedly reliable. However, when there is disk trouble it is usually severe; on three separate occasions we have experienced scoring of recording surfaces which required shutdowns of several days. An interruption of CTSS service to this extent is considered catastrophic.

During the year 1 July 1965 to 30 June 1966, 4785 hours of system time were "charged" to users; of which 3505 hours* represent remote-terminal use. (The other use was batch-processing "background".) These figures represent, respectively, 55 percent and 40 percent of the 8, 760 hours in a year.

---

*At a conservatively estimated ratio of 20 hours of "logged-in" time to 1 hour of charged time, this represents some 70, 000 console-hours and, presumably, 70, 000 man hours!

Operation of the GE 635 - Richard G. Mills and Thomas H. Van Vleck

The GE 635 computer*, which went on rent October 18, 1965, is being used for the single purpose of development and checkout of the GE 645 MULTICS system. One aspect of this use involves the so-called "6.36 system", under which certain users of the 7094 CTSS can prepare an input tape to the GECOS operating system on a 7094 tape drive. This tape, which usually involves a test run using the 645 simulator, is manually carried to the 635 and run, and the resulting output tape is carried back to the 7094 where it is loaded into the user's CTSS file space on the disk. The user may then use special-purpose debugging tools to evaluate the results of his run. Another major aspect of the use of the 635 is the so-called "6.45 system", in which card decks are input to GECOS for running under the 645 simulator, and output is produced on the 645 printer.

After a rather long period of unsatisfactory reliability, the 635 hardware and GECOS operating-system software are now operating satisfactorily.

*The machine referred to here as a 635 differs from the GE product-line 635 in that it has prototype core-memory units whose cycle time is approximately 1.6 microseconds instead of 1.0 microsecond for the standard 635. In all other respects, it is a standard product-line machine.

COMPUTER SYSTEM RESEARCH

The Transition from CTSS to MULTICS

A. CTSS Maintenance

B. CTSS Documentation

C. MULTICS Design Philosophies

D. MULTICS Software Development

E. MULTICS Implementation

Selected Topics in Computation

Academic Staff

F. J. Corbató
A. Evans, Jr.
E. L. Glaser
J. W. Poduska
J. H. Saltzer
C. Strachey

Non-Academic Research Staff

M. J. Bailey
M. E. Child
G. Clancy
P. A. Crisman
C. A. Cushing
R. C. Daley
S. D. Dunten
D. J. Edwards
C. Garman
R. M. Graham
H. C. Haig
B. W. Kernighan
C. Marceau
N. I. Morris
M. Padlipsky
E. Quisenberry
R. Rappaport
S. L. Rosenbaum
G. G. Schroeder
P. E. Smith
M. K. Thompson
T. H. Van Vleck
L. Varian
D. Wagner
D. R. Widrig, Jr.

Research Assistants and other Students

H. Magnuski
B. D. Wessler

Development of the MULTICS system is being carried out with the cooperation of the research staff of the Bell Telephone Laboratories and the General Electric Company.

The Transition from CTSS to MULTICS - F. J. Corbató

A. CTSS MAINTENANCE

During the reporting period, the group's primary effort has been shifted from maintaining the Compatible Time-Sharing System (CTSS) to developing the MULTiplexed Information and Computing Service (MULTICS). The final major activity on the development of CTSS was completion of a new file system which has allowed increased flexibility; enabling users to simultaneously read files as well as "link" to each others files. The file system was finally installed during August and September of 1965. The file system was difficult to install, not only because of the considerable magnitude of programming required, but also because almost all commands had to be modified. In addition, it was necessary to write special programs for dumping and reloading the disc, and salvaging the disc in the event of system mishaps (i. e., making the disk syntax correct); requesting the retrieval of files which had been removed from the disc; and programming for the standard input and output of information to and from the disc file. The final step required to make the new file system complete was addition of the DAEMON program for incrementally dumping a user's newly created files whenever he has logged out. This system, which is now completely installed, obviates the periodic total dump which was formerly required. Furthermore, the DAEMON should prevent file losses from being catastrophic when a system mishaps occurs, inasmuch as the backup is more closely tailored to user behavior than was the old system.

As would be expected with an evolving and maturing system, there are numerous new commands, and improvements have been made to old commands. Some of these changes are for improved administration of the system, such as upgrading of the LOGIN command, and proper record keeping of the console identification of each user, the system identification, and each user's last LOGOUT time. Another added administrative command is TTPEEK, which allows a user to inspect both the allotments and usage of his central processor time, as well as his disc, drum, and tape records.

Other changes have included the introduction of a MAIL command, which allows users to send notes to other users, on-line, to be deposited in individual "mailboxes" within each user's file directory. This feature has rapidly become a standard communication technique within the system. A further change in the command system has been the production of special commands to produce smooth operation of the RUNCOM command, which allows sequences of commands to be treated as macro-commands. RUNCOM has turned out to be one of the most powerful additions to the CTSS system, since it essentially allows users to arbitrarily create new commands of great complexity.

Inclusion of magnetic tape operation from foreground consoles within the time-sharing system has been another important step within the past period. Tape usage has turned out to be a valuable service, although out of necessity it must remain limited as there are not that many tape devices compared to users of the system. This feature has been a key factor in providing remote access to the simulation system for the GE 645.

A long-awaited feature of the time-sharing system has been the ability to leave Foreground-Initiated Background (FIB) jobs to be run in the absence of the user. In other words, the benefits of batch processing are flexibly brought to a time-sharing user such that he is able to disassociate himself from operating a well-behaved program. The FIB system has not only been useful as an extension of the system for time-sharing users, but has been of great assistance in implementating various automatic-accounting procedures; wherein programs "go to sleep" for periodic times and are subsequently reawakened to do various administration chores during intervals varying from minutes to weeks.

Additionally, there are restricted classes of users who are only able to utilize a subset of commands, and various changes and modifications have been made to improve and tighten system security against both accidents and vandalism.

B. CTSS DOCUMENTATION

In the area of documentation, one problem has been that as system size and complexity has grown, responsibility for maintenance has become more diffusely distributed: it has been more and more important to develop a systematic approach to maintenance. Shifting the maintenance staff away from CTSS was a fitting opportunity to draw up an inventory of the system components. This inventory consisted of an itemization of the modules in: 1) the main supervisor, 2) the command system, 3) the library system of subroutines, and 4) the auxiliary programs required to continue to operate the system (e. g., the Salvager, etc.). This inventory of the system has allowed responsibility to be sensibly delegated, and has been valuable as a check list when changes to the system have been contemplated.

The second edition of the CTSS Programmer's Guide was published at the beginning of this period in looseleaf form for easy maintenance. Furthermore, an up-to-date version of the manual is maintained on-line in CTSS using the TYPSET program, so that users may examine this copy from their consoles for last-minute changes. (See Crisman, Appendix C.)

C. MULTICS DESIGN PHILOSOPHIES

The primary efforts of the Multics workers were aimed at determination of final system hardware specifications, and the design and development of system software. Inherent in both of these efforts was the underlying requirement that all work on the project be thoroughly documented. Thus, a large body of reports and manual sections have been built up during the past year.

Multics is a joint effort of Project MAC, Bell Telephone Laboratories, and the General Electric Company. Its basic design philosophies were formally presented in a series of six papers, by representatives of all three organizations, at the 1965 Fall Joint Computer Conference. In their paper, "Introduction and Overview of the Multics System", F. J. Corbató of M.I.T., and V. A. Vyssotsky of BTL described important features of the new system:

"The overall design goal of the Multics system is to create a computing system which is capable of comprehensively meeting almost all of the present and near future requirements of a large-computer-service installation. It is not expected that the initial system, although useful, will reach the objective; rather, the system will evolve with time in a general framework which permits continual growth to meet unknown future requirements. Use of the PL/I language will allow major system software changes to be developed on a schedule separate from that of hardware changes ... It is expected that the Multics system will be ... available for implementation on any equipment with suitable characteristics ..."

Some of the important hardware features of the Multics system were also reported in this paper. They include: two-dimensional addressing (segments); "paged" memory; execute-only segments, which have particularly important applications for service bureaus and other applications where a program is to be executed but not read; a General Input/Output Controller (GIOC) which removes much of the input/output responsibility from the central processor, and, at the same time, makes the use of different input/output devices more flexible from the point-of-view of the programmer; and a full 128-character ASCII code. Another major hardware feature inherent in the basic design of the Multics system is the high level of system reliability that may be achieved. This is partly due to the fact that, because there will generally be more than one of each major component, a failure in any one component will not disable the entire system. This feature will also provide for more flexible maintenance, since the system can be partitioned into two distinct systems.

In addition to the paper already mentioned, five other papers presented at the conference were: "System Design of a Computer for Time-Sharing Applications", by E. L. Glaser of M.I.T., and J. F. Couleur and G. A. Oliver, both of G.E.; "Structure of the Multics Supervisor", by V. A. Vyssotsky of BTL, and F. J. Corbató and R. M. Graham, both of M.I.T.; "A General Purpose File System for Secondary Storage", by R. C. Daley of M.I.T., and P. G. Neuman of BTL; "Communications and Input/Output Switching in a Multiplexed Computing System", by J. F. Ossanna, L. E. Mikus, and S. D. Dunten, of BTL, G. E., and M.I.T., respectively;

and "Some Thoughts about the Social Implications of Accessible Computing", by E. E. David, Jr., of BTL, and R. M. Fano of M.I.T. (See Proceedings of the 1965 Fall Joint Computer Conference; Spartan Press; Washington, D.C.; pp. 185-247.)

D. MULTICS SOFTWARE DEVELOPMENT

Development of the software system has centered in four major areas, the first of which was development of a first design for the File System which is used to store segments. This generalized version of the file system used in CTSS introduces a hierarchial file structure which permits referencing to any number of levels. The system of links and permissions has been made more general, with linkage permission being associated directly with a permitted file. In addition, the file system incorporates such additional considerations as selective levels of user privacy, ease of data movement, and a backup mechanism which is readily-available in the event of either user error or system malfunction.

The design for the Central Traffic Controller was documented in a Ph.D. dissertation by J. H. Saltzer. (See Saltzer, Appendix B.) In his abstract, the author briefly describes his scheme for handling many users simultaneously:

> "The scheme is based upon a distributed supervisor which may be different for different users. The processor multiplexing method provides smooth inter-process communication, treatment of input/output control as a special case of inter-process communication, and provision for a user to specify parallel processing or simultaneous input/output without interrupt logic. By treatment of processors in an anonymous pool, smooth and automatic scaling of system capacity is obtained as more processors and more users are added. The basic design has intrinsic overhead in processor time and memory space which remains proportional to the amount of useful work the system does under extremes of system scaling and loading. The design is not limited to a specific hardware implementation; it is intended to have wide application to multiplexed, multiple-processor computer systems."

A third area of development involved I/O software for the previously mentioned General Input/Output Controller. The I/O system incorporates such features as stream switching, where the internal program may accept input and generate output without concern for the specific I/O device being used (indeed, these devices may be switched during the execution of a process); a standard character set, with both standard internal codes, external graphics, and a standard escape character, all irrespective of the particular I/O device being used; and a canonical string, where all strings of characters that look alike are represented alike within the system (this concept has been partially implemented within the context editor TYPSET in CTSS, but has now been made a standard part of the Multics I/O system).

The fourth key area of effort has been directed toward development of a command system. This includes a functional command language, similar in philosophy to CTSS. One feature of the command language is that many of the elaborate details need not be learned by the user if they are not needed in his particular work. In structure, the command-language interpreter is a system for calling upon a large library of subroutines to actually execute individual commands. Because of this, commands may be called and executed at any level; from a user console or within a running program.

As mentioned earlier, thorough documentation is an important part of the research effort in the Multics project. Consequently, details of the command language and other components of the system have been assembled in the Multics System Programmer's Manual. This document is being constantly changed, updated, and added to. A basic groundrule is the standard engineering practice that all software subsystems must be thoroughly designed, and implementation plans specified, before actual coding begins. We expect that this procedure will result in better integration and more effective operation of the resultant system than would otherwise occur.

E. MULTICS IMPLEMENTATION

To implement and debug the Multics system in a reasonable amount of time after it has been designed, a parallel tool-building effort has been carried out by various members of the three organizations involved. A GE 635 was delivered in July 1965 and was operating one month later. The following tools have been implemented:

1. A compiler for a subset of PL/I (called EPL for "Early PL") in CTSS;
2. An editing and input program (EDA) in CTSS using the Multics standard 128 ASCII characters which are typed in using appropriate escape conventions;
3. A program (merge-edit) in CTSS which writes a magnetic tape on the IBM 7094, in a format which is acceptable as an input tape for the GE 635 monitor (GECOS);
4. An assembly program for the GE 645 which runs in GECOS;
5. A program which simulates the GE 645 and runs in GECOS;
6. A program running in GECOS which loads a Multics process and starts simulation;
7. A collection of procedures (written in GE 645 assembly language) which provide the minimum services necessary to execute a Multics process;
8. A program to write on magnetic tape, the simulated GE 645 memory (when the simulated process terminates), and the results of any assemblies, in a format suitable for the CTSS disk editor;
9. An on-line, symbolic, interactive program in CTSS (GEBUG) which is used to examine the contents of the simulated GE 645 memory after it has been returned from the GE 635.

This complex of tools allows a system programmer to input, edit, compile, assemble, execute, and examine the results of a Multics process from <u>any</u> remote console attached to CTSS.

First units of the GE 645 were scheduled for September 1966 delivery. The simulation and debugging system just described will be used on the GE 645 until such time as the first version of Multics can stand by itself. It is expected that this will be in mid-1967.

Selected Topics in Computation - John W. Poduska

The effect of paging on computer execution times has been investigated by the writer and an undergraduate, David Chase. A simulator written for a small computer (the DDP-24) was run on the 7094 under CTSS, and to this simulator was added an associative-paging capability using a round-robin replacement scheme. A number of tests were then made, executing various test programs under simulation. (See Chase, Appendix B.)

The total results can be summarized briefly by stating that reloads of the associative memory occur less frequently than one might suppose. For example, a Fortran IV compiler was simulated under the following conditions:

Memory Size	8192 words, fully used
Page Size	256 words
Sticky Register	4

A 20-statement test program was compiled in about 300,000 memory cycles and required about 14,000 associative memory reloads. In other words, fewer than one-in-21 memory references required a load of the associative store -- smaller than a five percent overhead.

Two other small topics were investigated. First, a set of primitive list-processing subroutines was written in a package called MLPL (Mad List-Processing Language). These programs were written for a dual purpose: 1) to provide a primative alternate to SLIP, and 2) to provide a very efficient list-processing structure for system programs. (See MAC-M-303, Appendix A.)

Finally, the problem of table organization, searching, and sorting was being investigated primarily for classroom use: a tutorial paper on the problem was generated. (See MAC-M-302, Appendix A.)

ELECTRONIC SYSTEMS LABORATORY

Academic Staff

J. F. Reintjes	M. L. Dertouzos	A. K. Susskind
L. A. Gould	J. V. Oldfield (visiting)	

Non-Academic Research Staff

R. J. Bigelow	J. Katzenelson	D. T. Ross
T. B. Cheek	P. T. Ladd	R. H. Stotz
C. G. Feldmann	R. O. Ladson	W. D. Stratton
J. W. Grondstra	R. B. Lapin	D. E. Thornhill
U. F. Gronemann	H. D. Levin	J. F. Walsh
E. G. Guttmann	J. A. C. Parisot	J. E. Ward
D. R. Haring	R. B. Polansky	B. L. Wolman
F. B. Hills	J. M. Reed	C. Wylie

Research Assistants and other Students

L. O. Craft	W. Inglis	P. J. Santos, Jr.
T. Cruise	B. K. Levitt	D. H. Slosberg
R. Diephuis	G. C. Ling	C. N. Taubman
M. Edelberg	A. Malhotra	C. W. Therrien
D. S. Evans	W. H. Matthews	G. A. Walpert
J. Gertz	J. I. Meltzer	Y. D. Willems
H. L. Graham	J. E. Rodriguez	

Guests

S. I. Ackley	-	System Development Corp.	R. O. Ladson	-	UNIVAC Div., Sperry Rand
D. Barovich	-	I.B.M. Corporation	F. Luccio	-	Olivetti & Co.
A. J. Berger	-	Boeing Company	R. S. Lynn	-	North American Aviation
L. M. Bousquet	-	French Government	A. K. Mills	-	Dow Chemical Co.
H. J. Cilke	-	Sandia Corporation	J. V. Oldfield	-	University of Edinburgh
W. L. Johnson	-	Ford Motor Company	J. H. Porter	-	Chevron Research Co.
J. H. Jones	-	McDonnell Aircraft Corp.	R. B. Wise	-	IIT Research Institute

Introduction - J. Francis Reintjes

The Project MAC time-sharing computer system continues to stimulate the research activities of a substantial number of faculty, staff, and graduate students of the Electronic Systems Laboratory. A large segment of the Laboratory's research is motivated by staff interest in using time-sharing machines for on-line engineering design. Electrical-network design and the manipulation of graphical data pertinent to mechanical design have been explored through use of the MAC facility.

The pressing need for transient visual display of design information, such as that provided by cathode ray tubes, has resulted in a vigorous Laboratory effort toward developing graphical display systems for time-sharing machines. Recent emphasis has been placed on developing low-cost displays and on improving existing consoles previously developed by the Laboratory for general use at the MAC facility.

The Laboratory staff has continued to develop a family of computer-language systems which will enable the computer/designer team to work together effectively in the execution of creative design problems. In this effort the Laboratory staff has been augmented by guests from industry who expect to build upon research results of the group.

The MAC facility is also serving as an excellent vehicle for studies being conducted on aerospace navigation systems and associated instrumentation.

The impact of Project MAC on the educational goals of the Electronic Systems Laboratory is gratifying. During the past year, a total of 22 graduate students in the Laboratory conducted all or part of their thesis research using the MAC facility; and a new undergraduate electrical engineering subject in elements, circuits and systems has been inspired by research of a faculty member in computer-aided design.

Our MAC-related research is sponsored by the following agencies: Fabrication Branch, Air Force Materials Laboratory, WPAFB; Avionics Laboratory, USAF, WPAFB; Naval Underwater Ordnance Laboratory, Newport, Rhode Island; National Aeronautics and Space Administration; and ARPA (through Project MAC).

Reports of the groups are presented in the following sections.

Display Systems Research - John E. Ward and Robert H. Stotz

A. ESL DISPLAY CONSOLE

The ESL Console system (known as the Kludge) has continued to be available at Project MAC CTSS, and has been used by many groups at both M. I. T. and Harvard. No hardware changes have been made during the reporting period, except for minor additions to the command decoding for control of the Milliken synchronized camera. A number of program modifications have been made in the A-core Kludge Module for time accounting, camera control, one-shot storage displays, etc. A general overhaul of the entire A-Core/B-Core programming is underway (see discussion under Computer-Aided Design Project) in connection with installation of the PDP-7 buffer computer, as described in the next section.

One utility program of interest is a stereo routine for the ESL Console. This addition to the Kludge library makes use of the existing picture subroutine facilities of the A-core adapter to create a stereo pair for any display file. The two images appear on the right and left halves of the screen, and the left image is reversed so that viewing can be accomplished with a mirror projecting out from the vertical center line of the screen. Stereo images may be rotated, translated, or scaled in the usual manner.

The Number 2 Console for the M. I. T. Computation Center, fabricated by Digital Equipment Corporation from M. I. T. drawings, was received in February and installed in the display laboratory for checkout on the Project MAC Computer. In packaging this console, a different physical and wiring layout was used, rather than that in the original console, creating a number of "bugs" which have had to be tracked down. The Number 2 Console will be transferred to the Computation Center as soon as a few remaining circuit troubles are resolved and the necessary length of data-channel extension cable can be obtained.

The remote "one-shot" storage-tube displays discussed in the previous progress report have been placed in operation as a standard part of the

Project MAC facilities. These scopes are operated from the ESL Display Console by "borrowing" its X and Y deflection signals and its intensification signal. The beams of the remote scopes are always driven in parallel with the 16-inch display tubes of the ESL Console, but are only intensified when specifically addressed in a display list. As previously reported, logic for addressing up to 10 remote scopes has been installed, although only two scopes are now in use. Connection to the ESL Console consists of three coaxial cables, which may be up to several hundred feet in length. Examples of the display quality on line drawings and alphanumeric text were shown in the previous report.

Work has also continued on using telephone lines to extend the distance over which scopes can be driven in this mode (analog data transmission). For this purpose three telephone lines are used; two are equipped with Type 602B analog data sets, and one is equipped with a Type 202B digital data set (1200 bits per second). Because of the narrow bandwidth of the analog data sets (dc to approximately 1 kc), the clock rate of the ESL Console is automatically slowed down by a factor of 100 when a telephone one-shot display is called for. Noise introduction by the telephone circuits degrades resolution to the point where alphanumerics are quite unsatisfactory. However, line drawings are still of sufficient quality for many purposes.

The Harvard University Computation Center, under Professor A. G. Oettinger, installed the necessary telephone circuits (extensions of the M.I.T. Switchboard) in May 1966 and started regular operation in June. Typical displays are shown in their section of this report. The use of analog data transmission is not considered to be a long-range solution to the remote display problem, but merely a convenient expedient until suitable, digitally-driven, low-cost graphic terminals become available. Progress in this direction is described in Section C.

B. DISPLAY BUFFER COMPUTER

When the ESL Display Console was originally designed in 1963, it was recognized that a central computer should not have to be bothered with repetitively supplying data for display maintenance; i.e., continuously rewriting a flicker-free picture. The solution, of course, is to provide a buffer memory in the display system itself. Many buffered displays had been built over the previous decade, mostly using drum memories or special-purpose sequentially addressed core stacks. Memories of this type are relatively inflexible, however, and do not fit in with the dynamic computer-display interaction desired for computer-aided design. Therefore, the ESL Console was initially built without a memory, with display data buffered in the central computer core memory until such time as proper requirements for a buffer system could be determined. At the same time, the console was designed to minimize computer load by including hardware to perform many normally time-consuming computer functions, such as picture location, pen tracking, size changes, translation in two axes, etc.

As previously reported, the console has been running quite successfully in this "semi-buffered" mode on the Project MAC Time-Shared 7094 (CTSS) since January, 1964. There is a mutual interference problem, however, which has been of increasing concern as the load on the Project MAC computer has increased: the percentage of the 7094 memory cycles (data access plus interrupts) taken by the console on typical graphics is considerably higher than that used by a teletypewriter station, and the new high-speed 7094 drum system blocks display data whenever a CTSS memory swap is occurring, which causes the display to blink every few seconds. Also, total display list space in the A-core of the 7094 has been limited to 1200 words. Of this, 300 words are set aside for the one-shot storage scopes; the remainder is divided between the two console stations according to user requirements. Many users have had to sign up for both console stations in order to get sufficient display space. Finally, it is clear that display buffering is a requirement for any displays located remotely from the computer (such as are desired by many M.I.T. groups using Project MAC). Thus, interest was intensified this past year in solving this buffer problem.

Study of the display buffer problem over the past two years has led us to the conclusion (shared by many other organizations) that to provide the desired flexibility in real-time interactions at a remote display console, the buffer system should, in fact, be a small general-purpose computer. This conclusion was strengthened by introduction during the same period of quite powerful small commercial computers in the \$20,000 to \$50,000 price range, with clear indication that even these low prices will drop in the near future. Thus the Display Group recommended in November, 1965, that a small computer be purchased by Project MAC for buffering the ESL Console. The particular computer chosen was the Digital Equipment Corporation PDP-7, an 18-bit machine with convenient input-output provisions and a 1.75 μsec memory cycle. The PDP-7 circuitry is, of course, directly compatible with the ESL Console, which was constructed with DEC logic modules. The choice was also influenced by a desire to be software-compatible with other groups at M.I.T. and elsewhere that had already chosen the PDP-7 for display buffers. Cooperative efforts on design of communication formats, executive routines, etc. are already under discussion.

The PDP-7 computer, with an 8K memory, was received at Project MAC in February, 1966, for installation of the M.I.T.-constructed interface equipment. The 8K memory will provide approximately 4000 18-bit words of space for display lists (assuming that executive and real-time programs will require about 4K). This is the equivalent of 2000 words of 7094 memory, twice the amount which has been available to us in the A-core of the 7094.

Figure 13 shows the manner in which the PDP-7 will be spliced into the communication channel between the 7094 and the ESL Console. The interface being constructed by the Display Group (shown by the heavy outlines) contains a word-forming buffer, to match the 36-bit words of the 7094 and the ESL Console to the 18-bit word length of the PDP-7. Display lists and programs will be transmitted from the 7094 to the PDP-7, and then used to drive the

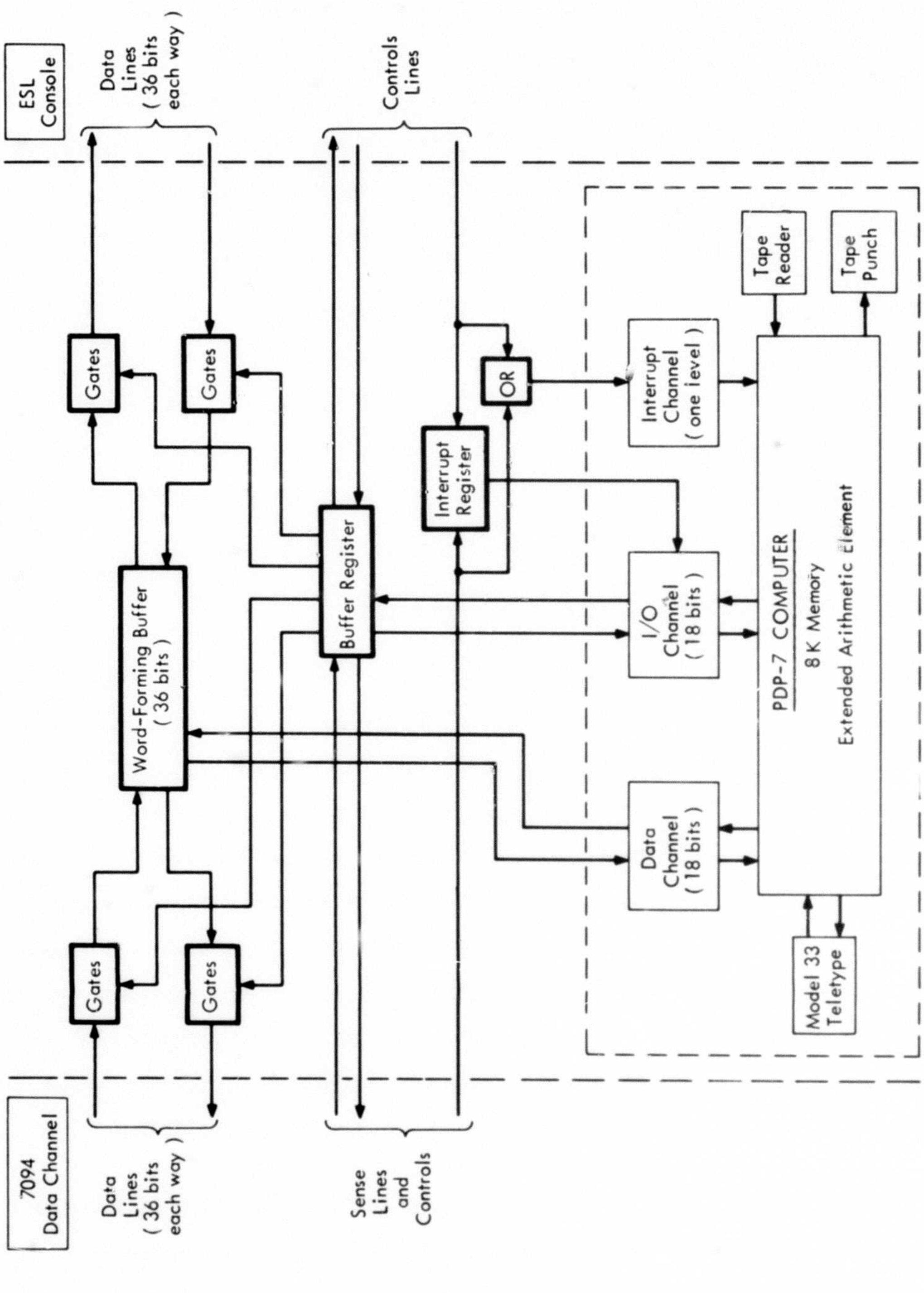


Figure 13. Display Buffer Computer Between ESL Display Console and Project MAC 7094

console in the same manner that it now operates from A-core of the 7094. Each display-list word (36 bits) will occupy two memory locations (18 bits each) in the PDP-7. The interface also contains the address register for the PDP-7 channel, a word counter, an interrupt register, and necessary control logic. All control signals, sense lines, and interrupts which formerly passed directly between the 7094 and the console will now be intercepted and interpreted by the PDP-7 executive program, which will act as a "traffic cop".

Present status of the display buffer is that the interface has been installed in the input/output section of the computer and is being checked out. Of course, these hardware aspects are only part of the problem of getting the system into operation -- perhaps a minor part compared to the design, writing, and debugging of the necessary software. It is hoped that buffered operation can be achieved before the end of the summer.

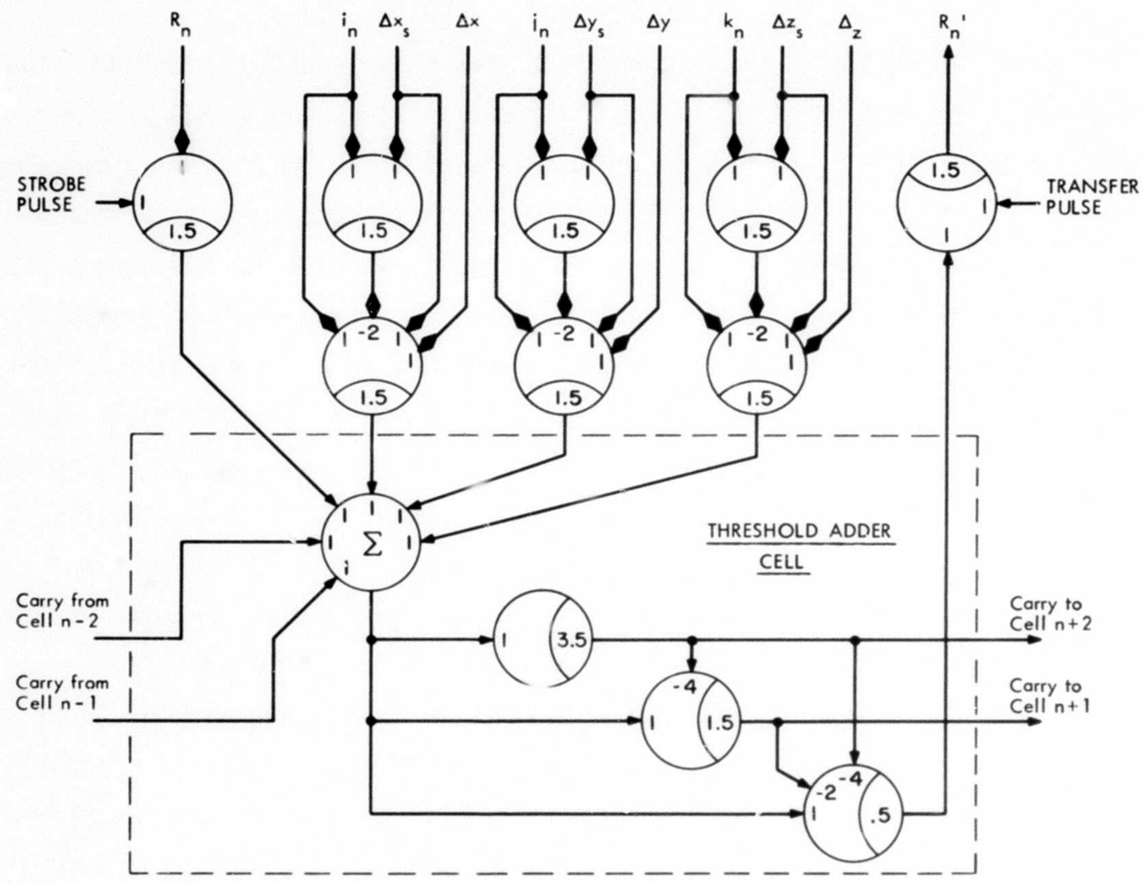
C. INVESTIGATION OF DDA ROTATION MATRIX

As discussed in the previous progress report, it is desired to replace the present binary-rate-multiplier (BRM) rotation matrix in the ESL Console by a digital-differential-analyzer (DDA) type to improve display quality. The primary advantage of the DDA is that it will eliminate the accumulation of roundoff errors in the matrix multiplications. Although the BRM and the DDA make similar roundoff errors, the DDA has a remainder register which "remembers" the roundoff in a particular operation and applies it to the next one. The BRM, on the other hand, "forgets" a roundoff as soon as it is made and errors accumulate from operation to operation. As pointed out previously, this has been an annoyance rather than a serious problem in the ESL Console, but certainly one which should be corrected if possible.

The general configuration for a DDA rotation matrix for the ESL Console was outlined in the previous progress report. This DDA is unusual in that four 10-bit signed numbers must be added (with arbitrary sign), in a time not exceeding one microsecond. At the time of the previous report, a four-input (plus carries) threshold adder cell with simple one-transistor threshold gates was working, but much too slowly. During the intervening period, an apparently satisfactory adder cell using differential threshold gates (current-mode logic) has been built and tested in breadboard form, and all the necessary sign-preparation and other gating needed to incorporate the DDA into the ESL Console has been worked out.

Figure 14 shows the present configuration of the general threshold adder cell and its associated gating. (The cells at each end of the adder will be slightly different.) In the "eyeball" representations of the threshold elements, the weight associated with each input is shown by the numbers at the arrowheads (note that weights can be negative). The numbers in the inscribed circles are the threshold settings. Two remainder registers (one for each axis) will be added to the ESL Console as part of the DDA installation. As shown in Figure 14, the contents (R_n) of these registers are gated into each addition cycle. Except for R, the inputs shown at the top of Figure 14 come directly from the present registers and pulse sources of the ESL Console. A slight change in usage of the rotation matrix data registers will occur, however. The present BRM rotation matrix performs sign-magnitude multiplication, and the i, j, and k registers are set in sign-magnitude by the computer. The line generator also works in the sign-magnitude system. Since the threshold adder cell performs ones-complement addition, all its inputs must be in this form (or converted to it). The gating shown in Figure 14 takes sign-magnitude inputs from the line generator (Δx and Δx_s, Δy_s, Δz and Δz_s), and twos-complement inputs from the i, j, and k registers and combines them as ones-complement inputs to the adder cell. Gating was saved by changing the computer setting of the data registers from sign-magnitude to twos-complement form. Also, this will be easier for future twos-complement computers (PDP-7 and GE 645) which will drive the ESL Console.

Three cells, as shown in Figure 14, have been constructed in breadboard form and tested as a three-stage adder. Cell transition time for the critical C_1 carry outputs is 70 nanoseconds. Also, all three cells exhibit identical characteristics with standard-tolerance (unselected) components. We are thus



KEY: i_n, j_n, k_n = n^{th} bits of Rotation Registers i, j, k

Δx, Δy, Δz = pulses from x, y, z outputs of line generator

Δx_s, Δy_s, Δz_s = sign bits of line generator data words

R_n = n^{th} bit of Remainder Register

R_n' = new contents of bit n of Remainder Register

Figure 14. Final Design of Threshold Adder for DDA Rotation Matrix

confident that a complete 10-bit DDA can be constructed, and that it will complete an addition cycle in about one microsecond. Since the available time between line-generator outputs is 1.5 ms, sufficient time will remain for input and output gating. Current work involves dynamic testing of the three-stage adder as a complete subsystem, and planning packaging for the DDA. It appears that each adder cell and its associated gates will fit on a single printed wiring card about 4 x 6 inches in size.

Although the preceding discussion has stressed the ESL Console as the ultimate destination for the DDA rotation matrix, the first one will actually be built for a different display system -- the DEC 340 in the Science Teaching Center at M.I.T. This machine, which will be used to display molecular models (discussed elsewhere in this report), has no hardware rotation system at present, but one is urgently desired. If we attempted to install the first DDA in the ESL Console, considerable down time would result; and in the event that problems were encountered, many users now dependent on the ESL Console and its rotation system would be unable to work. The DDA can, however, be installed in the DEC 340 with much less down time, since there is less equipment to modify or remove. Also, any delays in achieving satisfactory DDA operation will not interfere with existing users who depend on a rotation system.

It is hoped that the initial rotation system can be completed and installed by October of this year. Depending on problems (or lack of them) in this installation, plans will be made for modification of the ESL Console.

D. LOW-COST DATAPHONE-DRIVEN GRAPHIC DISPLAY

As described in the previous progress report, there is a clear-cut need for a low-cost (not more than $5,000) display console which can serve as a replacement for the present teletypewriters as time-sharing terminals in scientific and engineering applications. The unit should handle alpha-numerics at a 200-character-per-second rate, and should have a line-drawing capability to permit graphical output comparable in quality to that now obtained with the ESL Console. In addition, the unit should use a single voice-grade switched-service telephone connection to the central computer.

At the present time, there are a number of commercial display consoles on the market which will meet most of these objectives except the graphical requirement. Most of these use delay-line storage for display refreshment and are entirely character-oriented in their basic design. Conversations with various manufacturers indicate that it is difficult, if not impossible, to add a graphical capability to these existing units. In the most feasible proposal received, addition of a graphic capability would have more than doubled the price; and the resulting unit would not have been easy to program because of the rigid timing imposed by the delay-line storage system. Also, the amount of data which could be displayed at a flicker-free rate would be lower than desired, both for text and for graphics.

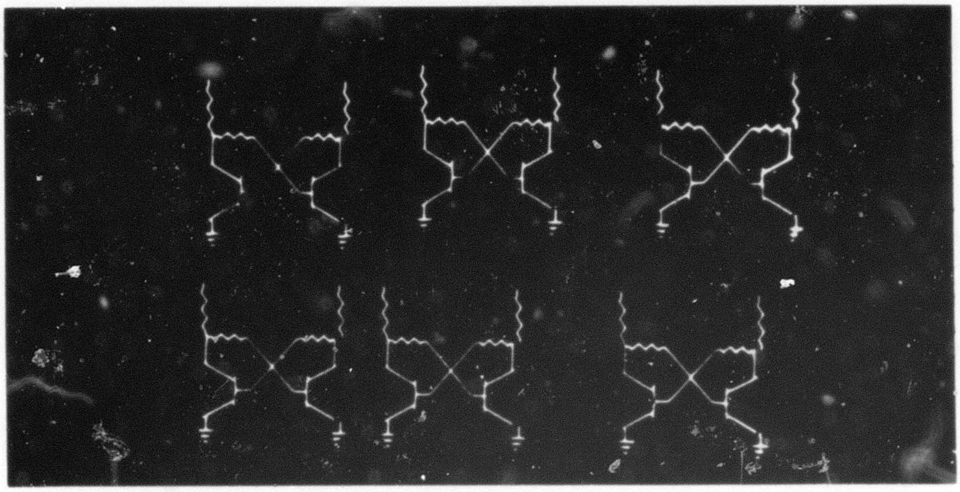
The approach being followed by the ESL Display Group in developing a low-cost unit is to store the picture in image form (direct-view storage scopes, electroluminescent panels, etc.) rather than continuously refresh a non-storage screen from some sort of electronic memory. A major advantage of this approach is that display generation and scope deflection circuits can be matched to the low input-data-rate of a telephone line, instead of having to operate at very high speed to avoid flicker. Also, the amount of information which can be displayed is not limited by circuit speeds or electronic memory size. Of course, dynamically changing displays, such as obtained with the present ESL Console, are not feasible with an image-storage approach, but these are largely ruled out anyway by the low bandwidth of the telephone line. Computer-buffered displays (such as the PDP-7/ESL Console combination) with a higher-bandwidth link to the computer would be used where real-time dynamic interaction is needed.

During the past year, a breadboard display system, including both a line generator and a character generator, was constructed and successfully operated from the Project MAC Computer via a 1200-bit-per-second telephone connection.* A Tektronix Type 564 five-inch storage oscilloscope (approximately $1,250) was used as the display device. Parts cost for all electronics including the Dataphone interface was less than $2,000. Typical line and character outputs are shown actual size in Figure 15(a) and (b). Note that the breadboard character memory had only eight words implemented (the first eight letters of the alphabet) and used a 5 x 7 dot matrix. The final design of the character generator will be a 7 x 9 dot matrix, and will include the full ASCII character set (94 printable symbols). Figure 15(c) shows a simulation of the final character generator using the point-plotting mode of the breadboard display unit. The line generator uses a hybrid approach (partly analog, partly digital) to achieve low cost. Certain problems, such as drift in the operational amplifiers, have been evident in the breadboard unit, but recent circuit improvements seem to have reduced these to tolerable levels. The ASCII character generator will require a 96-word, 64-bit, read-only memory; and both printed-resistor arrays and integrated-diode arrays have been considered as the best possibilities for low-cost realization. The Autonetics Division of North American Aviation, Inc., has silicon-on-sapphire (SOS) diode arrays of this size in pilot production, with the actual chip being only one-half-inch square.** Two such chips with the symbol patterns of Figure 15(c) were ordered for delivery in August. Work is proceeding on design of a complete interactive storage-tube display system (including an input keyboard), and it is hoped that a prototype can be ready for experimental use on the Project MAC Time-Sharing System before the end of the year.

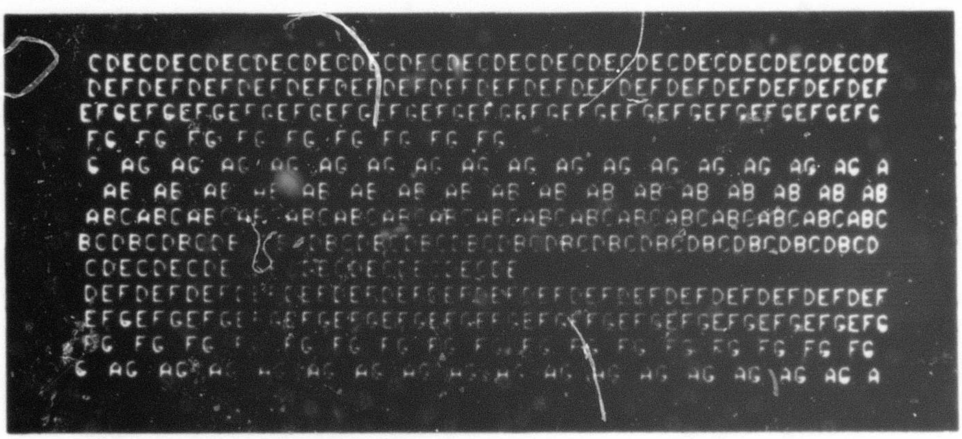
---

* <u>Design of a Low-Cost Character Generator for Remote Computer Displays</u>, T. B. Cheek, Report MAC-TR-26 (Thesis), Project MAC, M.I.T., March 1966, DDC No. AD-631-269.

** <u>Electronics Magazine</u>, May 30, 1966, pp. 152a-152d, and June 27, 1966, pp. 48-50.



(a) Circuit drawn with breadboard line generator



(b) Characters drawn by breadboard character generator (5x7 matrix)

(c) ASCII symbols (7x9 matrix) simulated with point mode of breadboard display

Figure 15. Typical Storage-Tube Displays from Breadboard Low-Cost Dataphone-Driven Console

E. IMPROVED DISPLAY TECHNOLOGY

During the reporting period, the thesis study of a novel hardware pen-tracking system using analog pulse-height comparison was completed.* As previously reported, this work was undertaken in an effort to reduce the approximately ten percent of display time taken by the present pen-tracking system, which counts points seen by the pen in a standard tracking cross (64-point pattern). The thesis results indicate that pulse-height comparison in a four-point tracking pattern is indeed a feasible technique, and that tracking time can easily be reduced to one percent (or less) of display time.

In the breadboard tracking system, the four points in the tracking pattern were displayed at 5-μsec intervals, and the digital error signal was available 25-μsec after the first pulse. Assuming a 35-μsec delay for initial positioning of the display to the tracking-cross location, and a parallel-processing system, the extrapolated time for a complete tracking cycle is 84 μsec. This compares with 1 msec for the present tracking system in the ESL Console. Circuit speed can no doubt be increased still further.

Work continued on the beam pen discussed in the previous report, with particular emphasis on reducing the field-of-view through improvement in the signal-to-noise-ratio, and use of a two-channel coaxial design. A gated coherent oscillator has been developed to replace the former pulse-excited ringing circuit. Also, further studies have been made on sources of the noise that is evident in the pen output signals, and it has been determined that the major source is in the pen amplifier and detection circuits, and not in the electron beam as had previously been suspected. A thesis study of

* Investigation of an Analog Technique to Decrease Pen-Tracking Time in Computer Displays, W. D. Stratton, Report MAC-TR-25 (Thesis), Project MAC, M.I.T., March, 1966, DDC No. AD-631-396.

the detection problem has been initiated. A paper on the pen was presented at the Fall Joint Computer Conference, Las Vegas, November 30-December 1, 1965.* Experimental work is continuing on the ESL Console.

The thesis research on the use of an infrared (IR) data link to operate a remote display was completed with a demonstration of satisfactory picture transmission.** As was previously reported, the link was an experimental one on loan from the M.I.T. Lincoln Laboratory and used a gallium-arsenide diode transmitter and a multiplier-phototube receiver, both cooled by liquid nitrogen.*** Also, despite a link design for 10-megapulse operation, previous tests at ESL had indicated that a transmission rate of 1.6×10^6 pulses per second was the maximum reliable rate on random pulse sequences because of certain circuit problems in the pulse driver for the diode.

In the remote display tests, the link was used to sample the contents of the digital horizontal and vertical output buffers of the ESL Console each increment time and transmit them from Project MAC to the Electronic Systems Laboratory. The ESL Console was operated in slow-clock mode for these tests -- 18 μsec per point. Transmission of the 28-bit messages (a start pulse, 13 bits for each buffer, and an intensification bit) was in serial form. At the receiving station, located in the Electronics Systems Laboratory, the data was stored in shift registers, digital-to-analog converted, and used to drive a Tektronix Type 564 five-inch Storage Scope. Transmission was error-free, as indicated by the photos shown in Figure 16. The actual transmission path was about 300 yards, and included two glass

* "The Beam Pen: A Novel High-Speed, Input/Output Device for Cathode-Ray-Tube Display Systems", D. R. Haring, AFIPS Conference Proceedings, Vol. 27, Spartan Books, 1965, pp. 827-835.

** Investigation of a Semiconductor Laser Data Link for Remote Computer Displays, G. C. Ling, Master of Science Thesis, Electrical Engineering Department, M.I.T., February 1966.

*** Optical Communications Employing Semiconductor Lasers, E. J. Chatterton, Technical Report 392, Lincoln Laboratory, M.I.T., (ESD-TR-232), 9 June 1965.

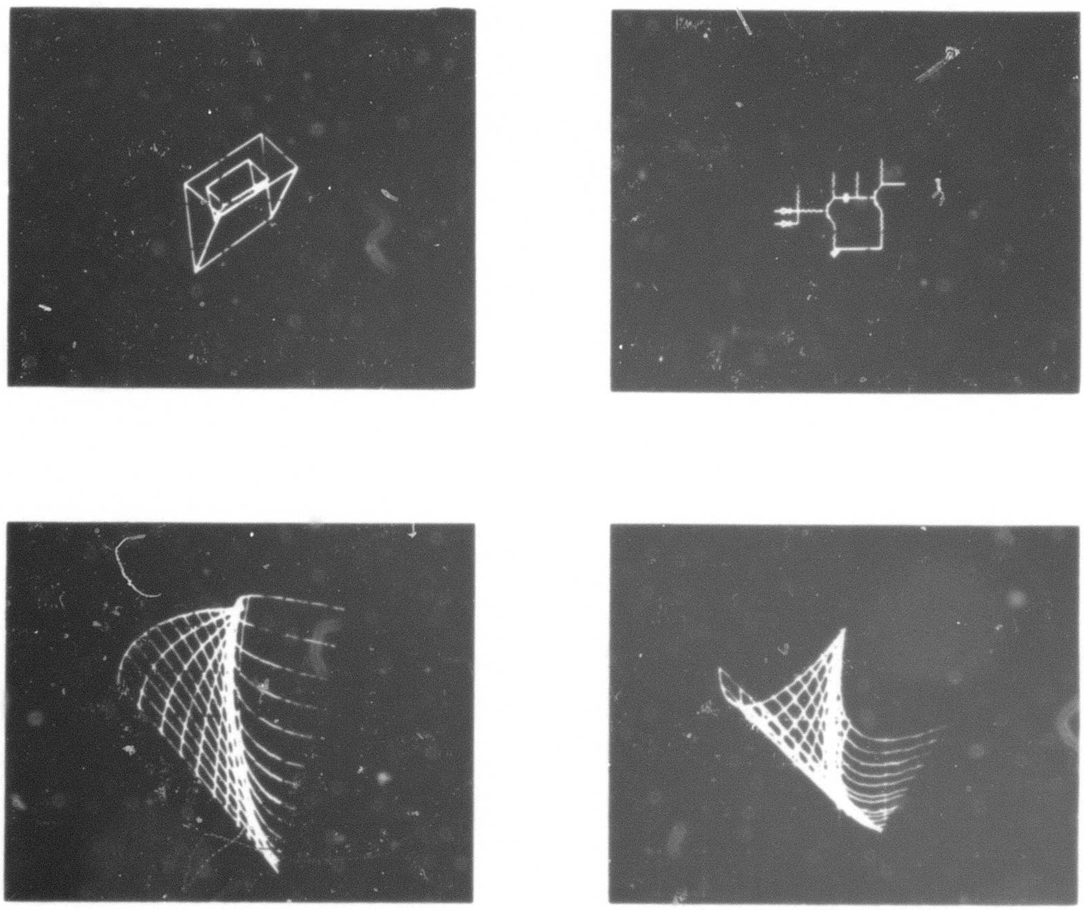


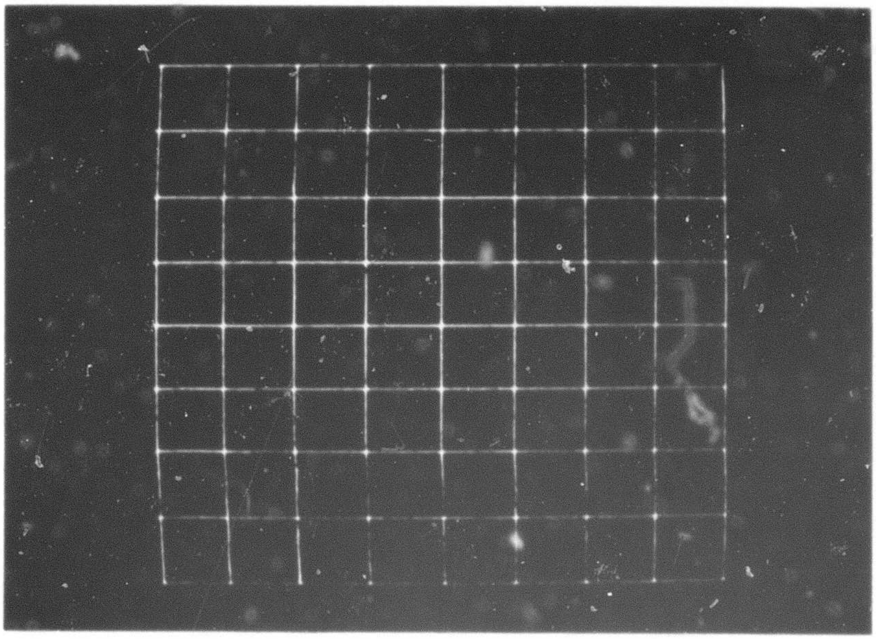
Figure 16. Typical ESL Console Displays Transmitted over an IR Link to a Slave Oscilloscope

windows. Also, a 24-db attenuating filter was used because of the short range (the link had been designed by Lincoln Laboratory to operate at a range of two miles). Transmission of the complete buffer contents each increment time (rather than sending only incremental data, which would be faster) was chosen as an expedient in these tests to simplify the transmitting and receiving logic.

Despite the success of these tests, no further work on IR data links is contemplated in connection with remote displays at M.I.T. The main difficulty, of course, is the line-of-sight restriction on location of transmitter-receiver pairs, which is a severe handicap in a complex of buildings. Considerable engineering work also would be required to develop reliable, continuous-duty cooling systems (liquid-nitrogen temperature) for the transmitter and receiver. Current planning is for local buffering of display data sent over available wire links, as has been described.

During the year, the Display Group completed construction and testing of the image-maintaining display system for the U.S. Naval Underwater Ordnance Station, Newport, Rhode Island, under contract N140(122)-76148b. As indicated in the previous progress report, a design study completed in March 1965 had recommended a system based on a TV scan converter for their particular application. Pictures are drawn on the storage surface of the scan-conversion tube by slaving its deflection inputs to a computer-driven display system (DEC Type 340), and are then viewed on a standard TV monitor by switching the deflection inputs to a TV raster. A 1203-line scan is used to provide minimum degradation of the resolution inherent in the original display, which has a 1024 x 1024 raster.

The system was experimentally tested by connecting it to a DEC Type 340 Display associated with the PDP-6 computer at Project MAC. It could have been connected to the ESL Console, but programming of the various tests was considered to be easier on the PDP-6 than on the Project MAC 7094 Time-Sharing System. Figure 17 shows typical stored displays as photographed from the 17-inch kinescope.



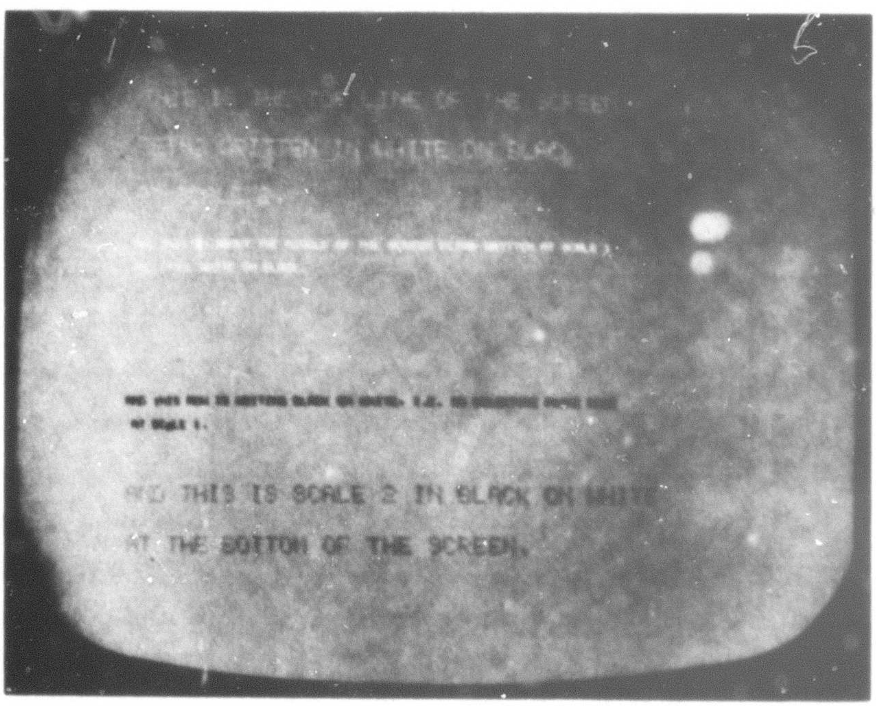


Figure 17. Displays on a TV Scan-Conversion Image-Maintaining System

Several features of the system are worth noting. First, the system was designed for selective erasure (or writing) of up to three characters during each vertical retrace time; thus text material can be altered and updated during viewing without requiring a complete erasure and rewriting. Second, it was discovered that by proper sequencing of the priming and writing modes of the scan converter, it was possible to write either white-on-black, or black-on-white. An example of the black-on-white writing is shown in Figure 17. Finally, although the rated viewing time for a picture stored in the scan converter is five minutes, it was found that this is quite conservative, and that some pictures did not degrade noticeably for viewing periods of up to one hour. The system has now been delivered to the Navy, and no further work on TV scan conversion is contemplated at this time. This project has, however, demonstrated the feasibility of the scan converter as an image-maintaining device for computer-generated displays.

In an effort to find an alternate method of image storage (in place of direct-view storage tubes) for the low-cost displays, an investigation was made of thin-film display screens consisting of an electroluminescent (el) thin film on a photoconductive (pc) thin film with an alternating voltage source applied across the combination. When light from a CRT falls on the pc from the side opposite the el, the pc conducts and causes the el to emit light as the voltage across it increases. The light emitted from the el produces the image and feeds back to the pc to keep it conducting. It has been determined that an effective way of eliminating the problem of image spreading (the lighting up of sections of the screen adjacent to the image) is to block the light being fed diagonally back from the electroluminescent screen to the photoconductive layer. By assuming an equivalent circuit for the device, the interaction of the pc-el has been studied and points of equilibrium in the operation have been determined. Effective placement of the stable and unstable points reduces the necessity of blocking 100 per cent of the feedback light to that of blocking as little as 60 per cent. In fact, it is mathematically possible to predict material properties that would require blocking less than 10 per cent of the cross-feedback light. This reduction of the cross-feedback could be implemented in a number of ways.

Using characteristics of commercial el thin films, constraints on the properties of the pc film have been determined which provide the best operating characteristics. This has been done because very little information is available on the bulk conductivity properties of pc materials. An experimental research program was undertaken to find or develop a photoconductor with these characteristics. Several photoconductive thin films were investigated, but none have the properties which are required. These results do not preclude the existence of such a photoconductor, but, if one cannot be found, the device will not operate properly unless all cross-feedback is eliminated, a problem which has not yet been successfully solved.

This work was conducted by W. H. Matthews, a Research Assistant working under Professor Chernow in the Photoconductive Semiconductors and Devices Laboratory, M.I.T. Professor Chernow left M.I.T., and Mr. Matthews transferred to another project; therefore, no further work is contemplated at this time.

Computer-Aided Design Project - Douglas T. Ross

The objective of the M.I.T. Computer-Aided Design (CAD) Project, sponsored by the U.S. Air Force under Contract AF-33(657)-10954, is to evolve a generalized man-machine system which will permit the human designer and the computer to work together on creative design problems. Such a system can improve the efficiency and precision of modern design, engineering, processing, and manufacturing efforts, especially in the aerospace industry, where problems are highly complex. The broad scope of computer-aided design requires a system that is general-purpose, natural to use, inherently evolutionary and efficient, and which can operate in a manner that is essentially independent of the specific problem, language, computer, and display system selected by the user. It should enable the user to readily derive specialized, problem-oriented systems from the parent system, and tailor them to his own needs. Although the major emphasis of the Project is on design applications, it is clear that such a generalized computer-aided design capability will be of broad utility in a wide range of man-machine problem solving.

From the system designer's point of view, the primary problem becomes not only how to solve problems, but how to state them. The Computer Applications Group of the Electronic Systems Laboratory has recognized this distinction with its Automated Engineering Design approach. The AED approach has been to isolate, define, and resolve sub-problems which are components of the larger problem of computer-aided design and man-machine communication, and to assemble from these components, a series of systems which approach the above goals in an evolutionary manner.

The AED-0 System, the major completed plateau of the CAD Project thus far, operates in both batch-processing and time-sharing modes on the IBM 7090 series of computers. It is a stable, working system, serving as the programming vehicle for all the Project's activities, and also is used by several other groups within Project MAC and outside companies. Rough statistics indicate that approximately 70 MAC users employ AED-0 in their work.

The current focal points of Project activities are 1) the AED-1 System, whose domain is general programming, compiling, and operating of programs on essentially any large-scale computer; and 2) the CADET System (Computer-Aided Design Experimental Translator), aimed at a generalized approach to computer-aided design applications as outlined above. An additional system, called AED JR, provides a uniform "first pass" of linguistic processing for both verbal and graphical language forms, and will be an integral subsystem of both AED-1 and CADET.

Although AED JR is a subsystem developed for subsequent integration into the total processing system, it is also a valuable tool by itself and has already been used in a number of language experiments. The AED JR System has provisions for the meta-linguistic description of arbitrary problem-oriented language, including the generation and debugging of control tables which enable the system to parse statements in the new language. AED JR performs both the syntactic and semantic parsing of an input statement, and in the process builds a "First-Pass Structure" model of the statement. In this structure the context of each word is shown by its location in a binary "tree", and the sequence in which the words and phrases should be considered to extract meaning is also represented. An appropriate "Second-Pass" operator can follow this sequence to complete the translation by compiling a program, building a model for graphical display, or whatever is needed for the problem being solved.

A. THE AED-1 PROCESSOR

Figure 18 shows the major phases of the AED-1 translation process. In short, the input string of characters is read and converted into a string of multi-character items, parsed into the first-pass structure, translated into the delayed-merge structure, and finally converted into the output character stream in assembly language for the target computer. Each of the major phases shown in Figure 18 is modular, so that it can be improved without disturbing the other portions of the general translation scheme.

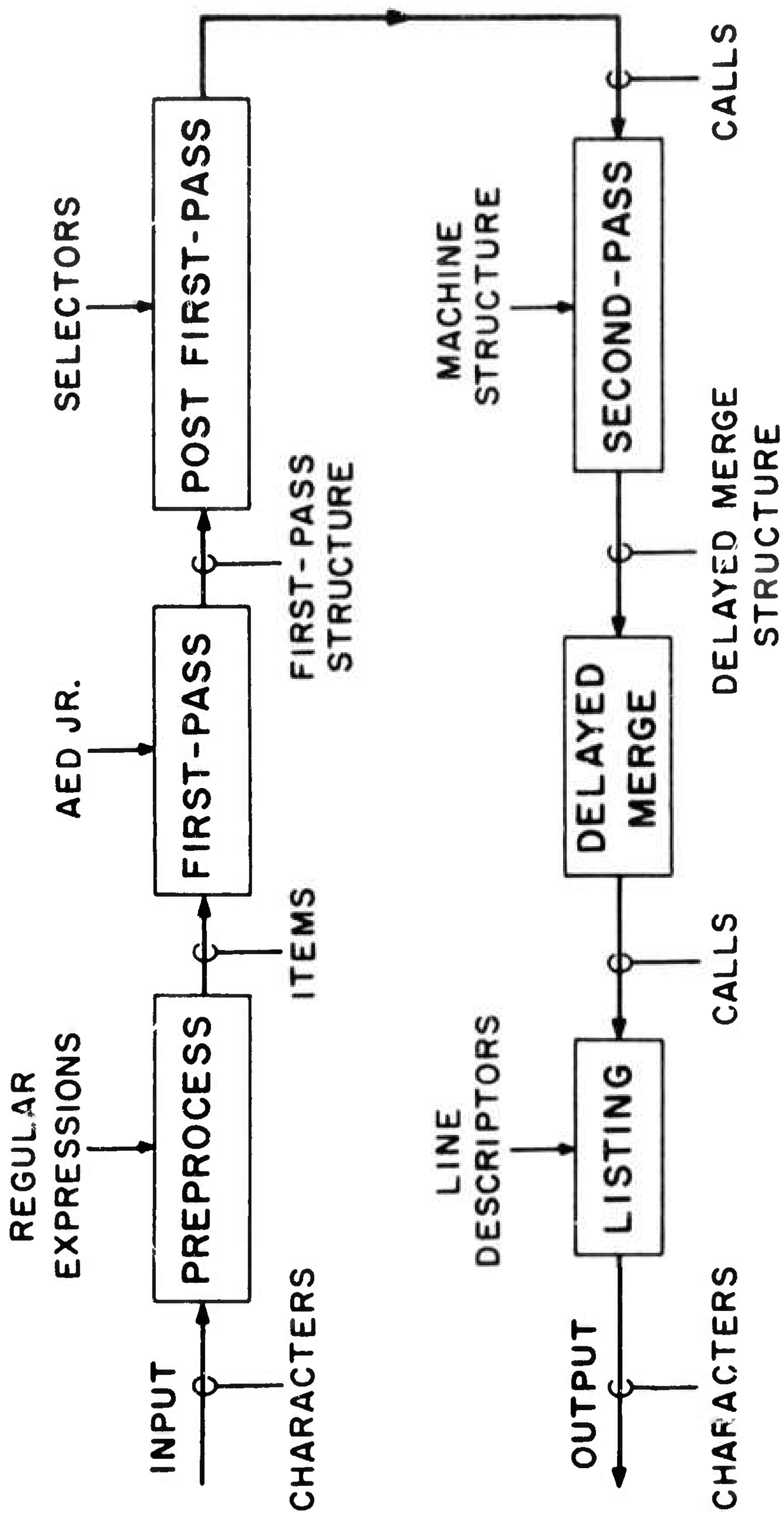


Figure 18. Phases of the AED-1 Processor

For the AED compiler to be machine-independent and language-independent, each of these major portions represents a general-purpose "machine" for carrying out some major portion of the over-all translation process; and since each of the "machines" is programmable (or table-driven), a large degree of machine and language independence is obtained in the AED compiler. Control information at each step is indicated in Figure 18. This modular approach to system building is implemented by the use of "integrated packages" of atomic and molecular routines which can be assembled in many ways to give a high-level approach to system programming.

The initial part of the Preprocessing Pass of AED-1, in which input characters are joined into multi-character items, is carried out by the routines of the RWORD Package. The established language of regular expressions (used for many years in studies of finite-state machines and automata) describes the various items to be built. We selected one of the simpler subsets of the language, since we feel it is adequate, and will also yield efficient finite-state machines for incorporation into AED-1. Items recognized by RWORD are immediately placed in a hash-coded Spelling Table Structure (written using the generalized String Package), so that each is stored only once, and a unique internal code can be used for efficient equality checks and identification inside the system. The elaborate input-string form of macro definition and expansion found in the AED-0 language is also carried out in Preprocessing by a special sub-language which is parsed and "second-passed" on-the-fly. At the same time, the declarations of the language being translated cause the block-structured Symbol Table to be built, and the Spelling Table provides immediate access to "active meanings" throughout the parsing process. The Symbol Table also uses the String Package.

After Preprocessing, the First-Pass Algorithm of AED JR (top center in Figure 18) is applied to the full language (the macro-definition and declaration portions having already been translated) to parse the input statement to produce the "First-Pass Structure" for the statements to be compiled. The Post-First-Pass section provides a uniform way to structure the semantic routines of the system and to drive the Second Pass.

"Second Pass" (bottom of Figure 18) is concerned with translating the meaning of a source-language statement (which has been parsed into the first-pass-structure form) into the equivalent object program. A Second-Pass Algorithm keeps track of the use of the machine registers of the object machine by means of a complete "environment vector" and its past history so that it can automatically detect conflicts in the way that atomic semantic units (isolated object-code sequence) use registers. When a conflict does arise, the Second-Pass Algorithm calls a "Hint" program, saying, for example, "Hint: I find that the value of A is now in storage and it should be placed in the accumulator." The Hint program then selects an appropriate plan to resolve the problem. The Second-Pass Algorithm with its Hint mechanism is a particularly attractive scheme, because it separates the problem of program optimization from the problem of generating workable programs.

The output of the Second Pass is the Delayed Merge Structure which is used to drive a table-driven Listing Program. The Listing Program may be set to produce an output form that will be accepted as input by the machine-language Assembly System for the target machine. In this way, the initial version of AED-1 does not have to be concerned with loader and operating system characteristics.

An initial version of the total AED-1 processor is scheduled for completion early in 1967. The AED-1 Processor will initially be set up to compile the existing AED-0 language, after which elaborations to provide AED-1 and successor languages can begin. We plan to make AED-1 operational on the GE 645 Computer, on the IBM 360 and 7094, and on the Univac 1108 at about the same time. Many provisions are included in the system to facilitate boot-strapping the system from one machine to another with minimal hand-coding of machine-dependent portions. Even so, the problems make such an operation a major undertaking.

B. THE CADET SYSTEM

CADET extends AED translating and compiling techniques to the area of graphical language and modeling. The intent of the CADET System is to provide a generalized, man-machine, problem-solving system which is problem-independent for the class of computer-aided design applications requiring mixed verbal and graphical modeling and calculation (in the same sense that the FORTRAN language is problem-independent over the class of scientific calculations requiring algebraic notation).

A viewpoint which we have held since 1962 is that communication by language ultimately resolves into the processing of a time sequence of distinguishable events, that verbal and graphical languages are merely different mechanical forms for those events, and that, in reality, there is only a single common linguistic processing involved. Our objective in CADET is a system in which both forms may be used interchangeably, and in which the characteristics of each type of language form are enriched by the application of features from the other form.

During the reporting period a preliminary system, CADET-0 was written and debugged. CADET-0 is a close relative of the "May 6" demonstration system (1964) described in previous reports, but uses the new graphical RWORD package which permits light-pen and pushbutton actions at the display console to be treated as an elaborate character string. Thus, this system demonstrates to an even finer degree the similarity between graphical input processing and character-string processing of verbal language. The principal goal of the new CADET-1 System is to pursue these similarities, and differences, still further by studying the relationship between block-structured symbol tables (as in AED-1) and problem-modeling structures, in order to derive a systematic structure for the CADET System as a whole.

C. AED COOPERATIVE PROGRAM

A contractual task of the Computer-Aided Design Project is to make its research results known and available to industry. To this end, the Project conducts the AED Cooperative Program in which staff members from industry visit M.I.T. for a nominal one-year period to learn the capabilities of AED, while contributing to its construction and improvement. Upon return to their sponsoring firms, they communicate to others their knowledge of the AED integrated packages, the completed "plateau" systems, and many innovations in programming and problem modeling learned at M.I.T. During the reporting period, the original Cooperative AED Project (which began in March 1964) was completed, and its successor was begun. The 1966 AED Cooperative Program includes twelve visiting staff members. Past and present participants in the program number 22 from 17 organizations.

The AED Cooperative Program applies to those aspects of the total work of the Project which are well enough developed to merit research and development with industry participation. The cooperative work at the moment is centered exclusively on the AED-1 Processor. As soon as feasible, the cooperative program will be extended into the areas of graphics and actual reduction-to-practice of computer-aided design to selected areas.

Copies of system releases and documents are also distributed to companies who do not participate directly in the AED Cooperative Program, so that they may experiment with AED-0 and AED JR on their own problems. As of 1 May 1966, the batch-processing version of AED-0 has been supplied by M.I.T. to seven companies who are past and present participants in the AED Cooperative Program, and eight copies have been distributed by IBM.

D. DISPLAY INTERFACE PROGRAMMING

The Computer-Aided Design Project is participating in designing and writing a new software interface for use when the PDP-7 computer is installed to buffer the ESL Console (see discussion under Display Systems Research). Although this system will be used initially with the IBM 7094, its major usage will be with the GE 645, and this is being taken into account in the system design. This software system will reside partly in the 7094 B-Core, partly in A-Core, and partly in the PDP-7. Programming is being done in AED-0.

As is the case with the present B-Core System, which has been described in previous reports, the new generalized B-Core Executive Program is being designed to provide a convenient standardized software interface between the user program and the display system. The B-Core Executive will also control communications to the small computer, sending blocks of display commands or programs for the small computer, and requesting attentions whenever appropriate. The programs sent to the small computer will come from a special library of PDP-7 programs, which is part of the software system. Routines in the library are being designed to interlock properly so as to build up extensions to the small computer executive; and also so that they are easy to connect with arguments in display command sequences. The present A-Core System will be relieved of its display-list buffering and real-time computing chores, which will be handled by the PDP-7; and the new A-Core System will be primarily an input/output adapter between the B-Core System and the PDP-7.

The requirement for absolute reliability of programs executed in the small computer (which could be relaxed if suitable memory-protection devices were incorporated into the small computer), means that initially a user will only be able to assemble programs from pre-coded subroutines existing in the library. A later development will include a special compiler (probably treating a subset of standard AED language) which will incorporate the needed checks automatically, and which then will allow the user to write his own programs for real-time execution.

To place the maximum small-computer capacity at a MULTICS user's disposal, a "minimal" executive program, which will be concerned solely with communication and data transfer, is being designed for the PDP-7. A "message" from the GE 645 to the PDP-7 will have three parts: 1) a block of words of binary information; 2) a data address in the PDP-7, where that block of information is to be placed; and 3) a control address in the PDP-7, to which control is transferred after the block of information has been stored. The data block may be an extension to the PDP-7 executive, a piece of display file, a real-time user program, or just raw data. If it is desired to send data back from the PDP-7 to the 645, an extension to the basic executive can be sent over from the 645 to send back "attentions" when they are generated. Other parts of the minimal executive will be concerned with establishing the basic execution cycle (the PDP-7 never stops running) and interrupt processing.

Computer-Aided Electronic Circuit Design - J. Francis Reintjes, Michael L. Dertouzos, Jacob Katzenelson

A. CIRCAL (CIRCUIT ANALYSIS PROGRAMS)

The CIRCAL system computer model for an electronic network consists of lists of storage registers, representing branches, connected to other lists which represent nodes. The method for solution at each instant of time is as follows: The nodes are assigned fixed node potentials. The voltage across each branch, and hence the current through it, is determined by the two nodes to which it is connected. Equilibrium requires the sum of the branch currents at each node to be zero. If the sum is not zero, new node potentials closer to the equilibrium values are computed by perturbing the old ones in a prescribed manner. This process is repeated until the equilibrium values are obtained within a specified tolerance.

Development of the CIRCAL-1 circuit-analysis programs has continued during the reporting period. The most significant change has been provision for current and voltage sources which may depend on a node-pair voltage. These are brought into the network in the same manner as independent sources. The voltage on which the source depends is identified by specifying the appropriate nodes. The library of source functions has been augmented to include (as standard waveforms) impulse functions and exponentials, in addition to the step functions and sinusoids previously available. Further, new branch elements have been permanently added to the system. These are the diode, the zener diode, and the tunnel diode. Previously, only one of these types could be incorporated into any particular circuit. A motion picture demonstrating the capabilities of CIRCAL-1 was completed in February.

Most recent work has been directed toward modifying CIRCAL-1 programs to make them compatible with various peripheral programs. In particular, a structure has been set up that allows a user to call programs which permit nonlinear resistors to be described and included in the circuit being analyzed. An input-output package has been completed which allows the user to plot results of on-line circuit analysis on the ESL Console, or on a remote Teletype or IBM 1050 teletypewriter. Also, a compiler has been developed for expanding CIRCAL by permitting inclusion of memoryless non-linear elements, nonlinear capacitors and inductors, and arbitrary voltage and current excitations. The technique is a method of compiling on-line descriptions of arithmetic expressions that are of the general form G = f (x, y, z), where G is a single-valued function of variables x, y, and z. The maximum number of variables in the expression can be increased from three to any size by a simple extension. A standard precedence grammar of algebra is employed; making a form of bounded-context translation possible. In this translation, action to be taken is determined solely by the symbol currently being scanned, and at most three symbols to its left.

In addition to this work on CIRCAL-1, time was spent on the design of the next generation of circuit-analysis programs: CIRCAL-2. Conferences with circuit designers have provided considerable data regarding features that would be desirable for the new system. In the light of these requirements, a new method of simulation seems desirable and is being considered. The compiler programs mentioned above were written so that they are completely independent of the CIRCAL programs. Hence, the compiler which is presently compatible with CIRCAL-1 will also be compatible with CIRCAL-2, no matter how different the new circuit-analysis programs may be.

B. DIGITAL-SYSTEM SIMULATION

Work on digital-system simulation is being continued. The simulation system is organized into an on-line command-type structure, with three main subsystems. These three subsystems deal with graphical input of block diagrams, on-line definition of special blocks and input signals, and the actual simulation and debugging of a digital system specified by the first two. The portions completed to date are core routines for the third part and a master program to control the entire process of block definition, block-diagram description and wrap-up, and simulation.

The "core" routines are the group that -- when given an array of input values, identification of the block class, and an array in which to store the results -- will iterate the block function through one time cycle at all levels. New inputs and previous states are used to compute the new states. Any logical errors are reported as they occur. At present, these routines are equipped to handle only standard blocks and "wrapped-up" structures with single-line inputs. However, specific planning has gone into the work so that addition of the capability to handle user-defined blocks and "bundled" signal lines is straightforward and will require no change in the existing system.

The sample block diagram used to check out the core routines was a scale-of-three counter using two-level flip-flops, four delays, and seven gates. A dummy main program was written to call on the core routines in the proper repetitive fashion with a square-wave input. When simulated, the counter did not appear to operate correctly, and it was found that a mistake in the logical design of the counter had been made and a fifth delay had to be included in the counter to make it operate as desired. The above experiment, though somewhat crude, provided a startling example of the usefulness of such a simulation.

The graphical input-output package allows arbitrary digital systems to be described to the computer by drawing them in block-diagram form on the screen of the ESL Display Console through use of the program-controlled passive inputs (shaft encoders, digital switches, and toggle switches). The blocks of the diagram may range in detail from an individual gate to a complete digital subsystem. Concurrent with the formation of this display, a comprehensive data structure is established within the computer which stores all classification and topological information contained in the display. The structure is built in the form of an interconnected string-pointer list. When this list is completed, the data within it can be used by the simulation programs to analyze overall system behavior.

An attempt has been made to minimize the number of unnatural actions an engineer is required to perform when using this program. Specifically:

1. No programming ability of any kind is required,
2. No artistic talent is required to draw a neat diagram,
3. The display is in standard block-diagram form, so the engineer speaks to the computer in his own graphical language.

The program defines a number of common digital elements to be standard objects. These are the six common logic gates (AND, OR, NOT, NAND, NOR, EXCLUSIVE OR), the various types of flip-flops (complement, set-reset, combination), delays, and lead junction boxes or fan-out points. Each of the standard objects is accorded the following special treatment:

1. It has a unique, easy-to-recognize symbol,
2. It may be oriented in any direction,
3. It has individual buttons assigned to minimize the necessary man-machine interaction.

All an operator need do to plot one of these objects is position the tracking cross correctly on the screen and push the appropriate button (or buttons). Any block that the designer wishes to use in his system diagram that is not a standard object must be explicitly defined by him. The program gives him two alternative methods of doing this: Either he can draw a block diagram of the object using standard objects and previously defined non-standard ones; or he can give a suitable input-output description of the block. If the first method is used, the resulting block is denoted as a "wrapped-up" element. The wrap-up feature allows the engineer to test his system in small units and to use nesting of definitions.

The blocks of the diagram can be interconnected by the operator through use of the light pen. Each lead may have any number of segments, each of which may be placed and oriented as desired. Thus the operator can reproduce his pencil picture exactly on the screen. Perfect pen positioning at the ends of lines is not required, as lines are made to latch automatically to the correct picture parts.

C. CURVE-DRAWING REMOTE DISPLAY

An experimental model of a low-cost curve generator, intended for possible use in a remote graphical-display station of a time-shared computer facility, has been designed, constructed, and tested as a thesis project. This system uses a parametric technique based on the control of curve segments. Complex curves are composed of the minimum number of such segments that achieves a prescribed fitting error. Such a technique permits a compact description of graphical data; which is desirable from a communication viewpoint.

The experimental model uses a storage-type cathode-ray tube (CRT) for visual display; driven by two waveforms, x(t) and y(t), which are the outputs of linear networks T_x and T_y. These networks are characterized by their respective step responses $T_x(t)$, $T_y(t)$, and are constrained to have unity steady-state gain. Computer commands are made up of a sequence of six characters tapped from a standard 35 KSR Teletype terminal of the MAC system. Each computer command supplies a set of coordinate values which, after being converted to analog signals, are used to drive T_x and T_y. A third portion of each computer-command controls the parameters of T_x and T_y. A curve segment is plotted as follows.

Assume that at some time, t_k, the linear networks have attained steady state, such that $x(t_k) = x_o$ and $y(t_k) = y_o$, and at this time the computer supplies a new set of coordinates $(x_o + \Delta x,\ y_o + \Delta y)$. The signals supplied to the CRT will then be

$$x(t) = x_o + \Delta x T_x\,(t-t_k)$$
$$x(t) = y_o + \Delta y T_y\,(t-t_k)$$

resulting in a displayed curve segment starting at point (x_o, y_o) and ending at $(x_o + \Delta x,\ y_o + \Delta y)$. The particular trajectory, $f(x, y) = 0$, between these two points will depend upon the parameters of T_x and T_y.

The major criteria used in selecting a realization for T_x and T_y were:

1. It should provide a large and varied family of segments,
2. It should have the ability to match slopes of connecting segments,
3. It should have a configuration which can be reliably implemented.

The chosen realization is given by

$$T_x(t) = 1 - \left[\alpha_x e^{-\sigma_o t} + (1 - \alpha_x) e^{-\sigma_x t}\right], \quad t \geq 0$$

$$T_y(t) = 1 - \left[\alpha_y e^{-\sigma_o t} + (1 - \alpha_y) e^{-\sigma_y t}\right], \quad t \geq 0$$

under the constraints

$$0 \leq \alpha_x \leq 1 \qquad \alpha_x < 1 \rightarrow \alpha_y = 1 \qquad \sigma_x > \sigma_o$$
$$0 \leq \alpha_y \leq 1 \qquad \alpha_y < 1 \rightarrow \alpha_x = 1 \qquad \sigma_y > \sigma_o$$

This realization has four degrees of freedom; two degrees for determining steady-state coordinates, one degree to specify final slope, and one degree to yield a family of segments having that slope. The exponential time response results in an economical implementation.

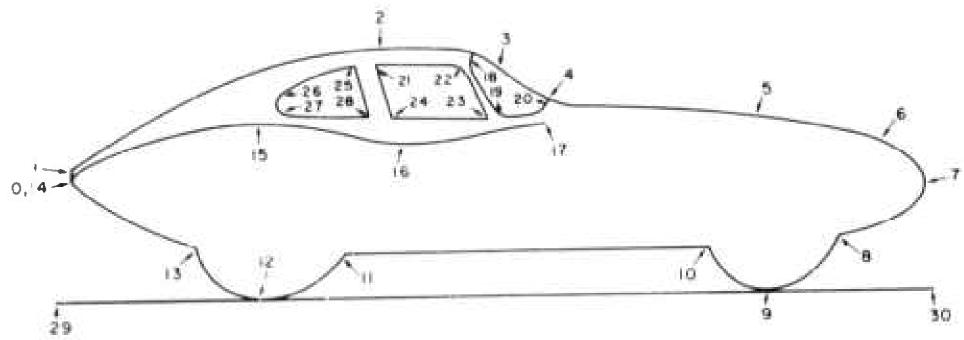
A software translator was written to convert a desired curve from a point-by-point description to a series of curve segments realizable by T_x and T_y. A given curve to be plotted is matched within a given maximum allowable error using the minimum number of segments necessary. Wherever possible, slopes of connecting segments are matched.

For testing purposes, the prototype hardware and translator software were used in connection with CIRCAL. It was found that the step response of a simple tunnel-diode circuit, computed by CIRCAL at 100 points, can be displayed by means of three to five segments, depending upon the desired fit. In another test, the automobile silhouette of Figure 19(b) was displayed with 28 segments, using the data points shown in Figure 19(a).

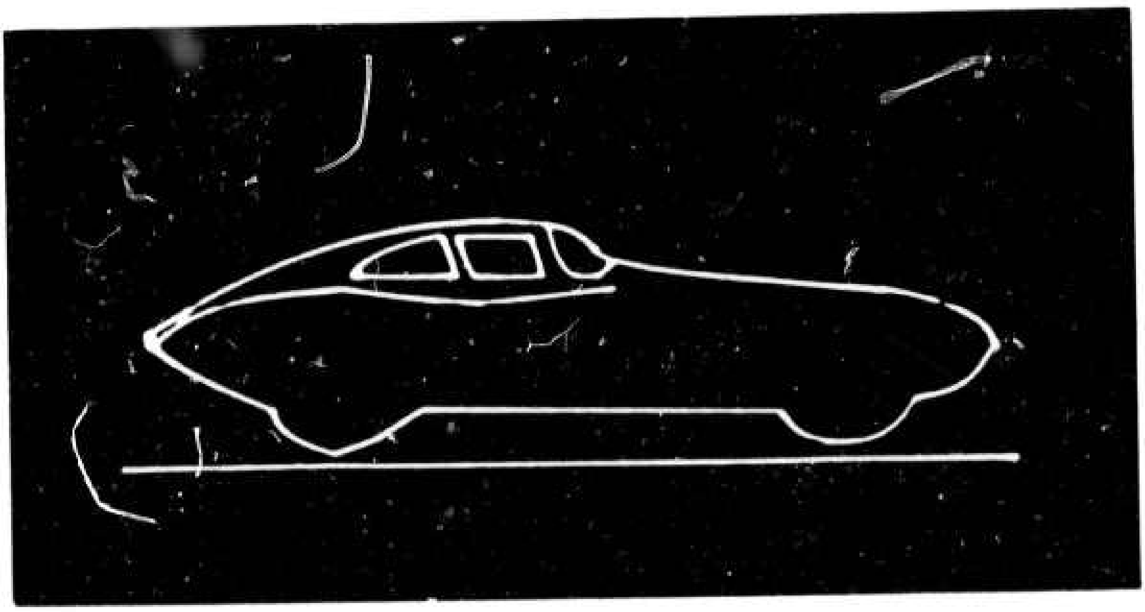
The main feature of this display technique is the utilization of few computer words for the display of a complex curve. Advantages resulting from this feature are small storage requirements for the translated curves and a relatively fast display time. The main penalty is the computational effort necessary for translation of a given curve into appropriate display commands.

D. AEDNET - A SIMULATOR FOR NONLINEAR ELECTRONIC CIRCUITS

AEDNET is a system of digital computer programs for simulation of nonlinear electronic circuits. It takes maximum advantage of the similarities between resistive nonlinear networks (where pertinent variables are voltage and current), inductive networks (where the variables are flux and current), and capacitive networks (where the variables are charge and voltage) to arrive at an approach to the solution of nonlinear RLC networks. Through use of this approach, circuit elements with a variety of types of nonlinearities can be included in the networks to be analyzed. The system has two main features: 1) the ability to simulate the behavior of a wide class of nonlinear networks; 2) the on-line interactive use of a digital computer and its oscilloscope display unit. AEDNET is written in AED-0 and



(a) Sports Car Silhouette Data Points



(b) 28-segment Output Display

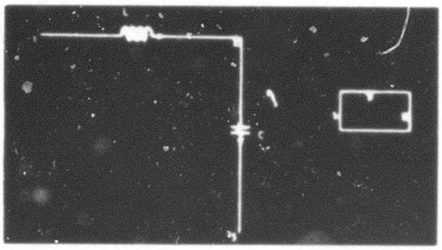
Figure 19. Automobile Silhouette on the Curve-Drawing Remote Display

it operates on the Project MAC Time-Sharing System and the ESL Display Console. AEDNET simulates networks whose elements are nonlinear time-varying capacitors, resistors, inductors, and dependent and independent sources. The network is required to satisfy certain topological conditions, and the elements' characteristics are required to satisfy suitable Lipschitz conditions. Nonlinear characteristics are represented in the computer either by a table or a subroutine. Elements with non-monotonic characteristics are allowed, and elements' characteristics are required to be continuous but not differentiable.

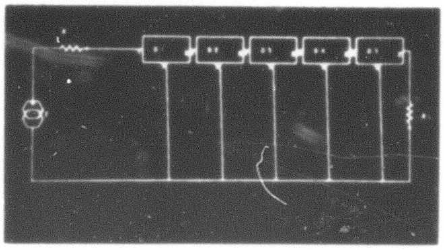
Input to the program is given mainly by drawing the network diagram on the face of the oscilloscope, and assigning values to the network elements by typing on the console's Teletype, or by drawing the elements' characteristics. The program's output consists of waveforms of the various network variables. Each variable can be displayed on the oscilloscope as a function of time or as a function of another network variable. After a network has been specified and the response calculated and displayed, the user can modify the network by changing both elements and topology and ask for recalculation and redisplay.

A feature recently introduced into the system is the use of "devices", which summarize network information by giving it another name. This includes the ability to define a subnetwork, give it a special symbol and use the result as a new network element in the construction of a larger network. Thus the device's function in the description of a network is similar to a procedure (subroutine) in writing programs. It also has some special value to AEDNET. The nonlinear capability of the system is achieved by defining some basic generalized elements. This implies, for example, that a diode is considered by the analysis part of the system as a nonlinear resistor. The user, however, would like to think of a diode as a diode, and address it by a "diode" symbol. This is done by defining a subnetwork with an appropriate resistor, giving it a "diode" symbol and using the newly-created device as a new element.

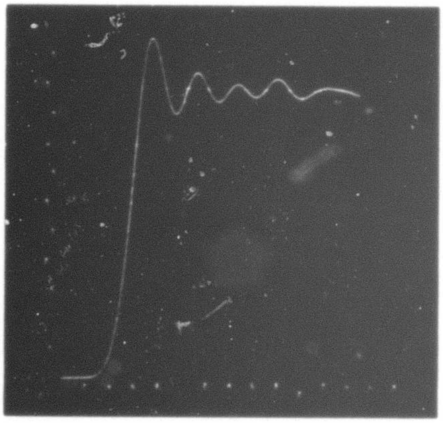
As an example of the use of AEDNET, Figure 20(a) snows a small network after it was defined and assigned the three-terminal symbol which is shown to the right of the network. The numbers indicate the correspondence between the network nodes and the symbol's poles. This filter section is now connected to make a larger filter, which is terminated by appropriate resistors as shown in Figure 20(b). A calculation of the response for a step input is now requested. The output voltage on R_L is given in Figure 20(c) and the input current in Figure 20(d): the 'flower' of Figure 20 (e) is the output voltage as function of input current.



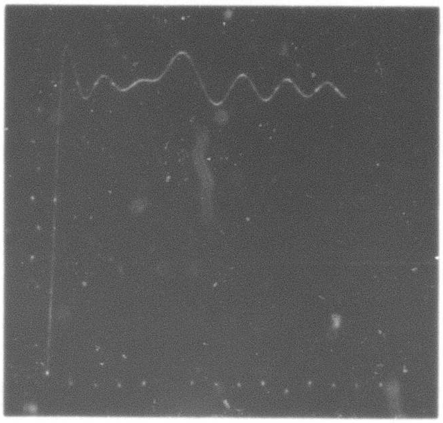
(a) A network and its equivalent "device"



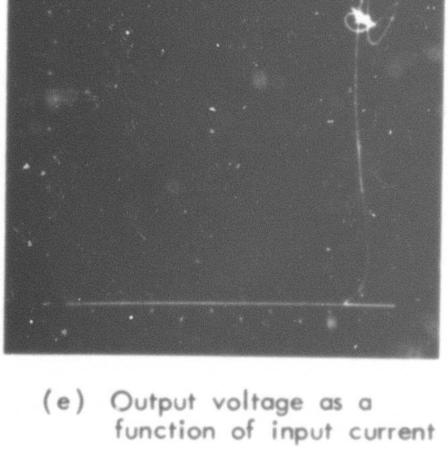
(b) A filter made up of the devices of (a)



(c) Output voltage in response to a step input



(d) Input current



(e) Output voltage as a function of input current

Figure 20. An Example of the Use of AEDNET

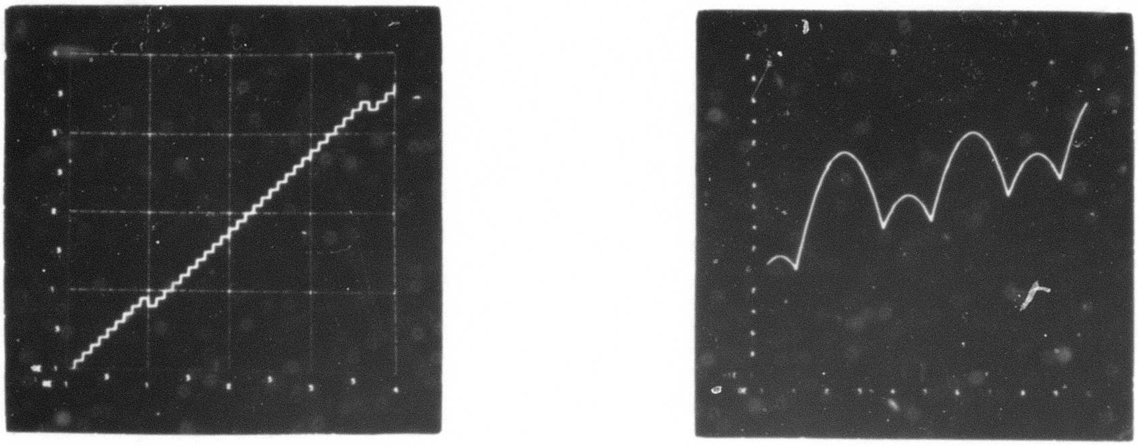
Aerospace Computer Analysis and Synthesis - Donald R. Haring, Frank B. Hills, Alfred K. Susskind

During this past reporting period, simulation studies (for the U. S. Air Force under Contract AF-33(657)-11311) of the digitally controlled accelerometer system described in the previous progress report have been completed. A thesis, a laboratory report, and a conference paper describe the system, methods, and results of these studies. Very briefly, the accelerometer system is described as a feedback control system in which the accelerometer sensor is the controlled element and the controller consists of an analog-to-digital converter, a digital filter, a digital decision element, a bidirectional counter, and a digital-to-analog converter. The counter contents is a measure of the instantaneous acceleration.

System stability was analytically investigated by the describing-function method, its performance evaluated by means of a simulation program run on the Project MAC computer, and results displayed on the ESL Console. Examples of the simulation results displayed on the ESL Console are shown in Figure 21. Simulation is necessary, because the extreme nonlinearities in the system make analytical methods of little value.

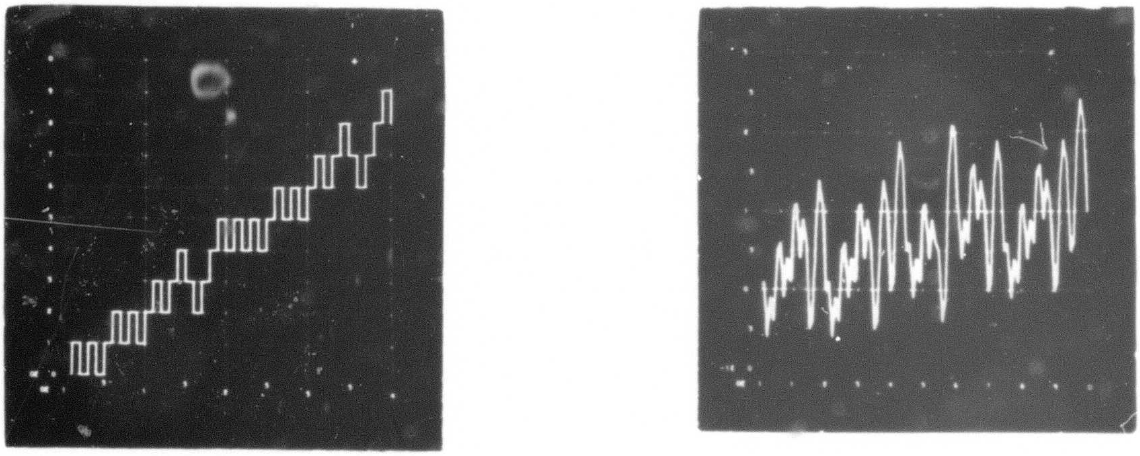
Compared to the pulsed-rebalance accelerometer system -- a very common high-performance, digital-accelerometer system -- the finer quantization of acceleration in the new system under study greatly reduces round-off errors. It was shown both analytically and by the simulation program that the new system improves upon the pulse-rebalance system by a factor of at least five, as far as steady-state errors in velocity are concerned. It was also shown by the simulation program that the new system performs better under dynamic inputs as well. The effects of noise were also evaluated, and it was shown that noise degrades the performance of the new system more seriously than that of the pulsed-rebalance system.

These studies were terminated with the recommendation that an experimental program begin to evaluate the problems of hardware implementation and the performance of an actual operating system.



OUTPUT VELOCITY BIAS ERROR

a) System Response to a Ramp of Slope 0.9 $\dot{A}_{max}$



OUTPUT VELOCITY BIAS ERROR

b) System Response to a Ramp of Slope 0.2 $\dot{A}_{max}$

Figure 21. Response of a Simulated Digitally Controlled Accelerometer

Simulation Studies of Strapped-down Navigation Systems - Frank B. Hills, Jacques A. C. Parisot

In a research program for the Air Force, under Contract AF-33(657)-11311, we are continuing to study the computations peculiar to "strapped-down" navigation systems, in which inertial sensors (gyros and accelerometers) are rigidly mounted to an air frame, rather than being mounted on a stabilized platform. The accelerometer outputs must therefore be transformed from body axes to the coordinates in which navigation computations are made. We are using the direction-cosine transformation and, because of the rotations of the vehicle, the direction-cosines are not fixed and hence must also be computed. This is done by numerically solving nine simultaneous differential equations in which gyro data are the independent variables.

Analytical studies made by the group have shown that the major errors, insofar as the computations are concerned, are due to sampling and quantizing gyro and accelerometer data. Because acceleration and rotation are noncommunicative, the effects of sampling and quantizing on the errors are very complex. Therefore, simulation studies on the PDP-6 were started to supplement the mathematical investigation.

An outline of the desired simulation program and the reasons for the separate parts were given in last year's report. During the past year this program has been written and debugged, and the simulation tests have been started. One major goal in writing the program was an organization that would permit a great variety of simulation tests with a minimum of debugging once the main program was written. This goal has been achieved, and a typical test, complete with printed output, can be set up with one page of coding. The other major goal of the program was speed. Results are being obtained in a fairly reasonable length of time, considering the computations required; however, means for speeding up the computations are being considered.

LIBRARY RESEARCH

The Technical Information Project

Reconstruction of the TIP Programs

Search Procedures Based on Measures of Relatedness Between Documents

Formatting Output Files Within the TIP System

A TIP On-Line System Dictionary

System-Initiated Response from TIP

Sorting and Inventorying of Large Files

Processing of Serials and Journals

A List Structure for Storing Dictionary Information

Creation of a First-Cast Thesaurus from TIP Files

Academic Staff

S. C. Brown	M. M. Kessler

Non-Academic Research Staff

J. P. Casey	M. C. Henneman	W. H. Scholz
T. F. Dempsey	W. D. Mathews	P. M. Sheehan
E. A. Dole	K. D. Rude	W. A. Solomon
I. Y. Johnson		

Research Assistants and other Students

A. Gevins	A. N. Kramer	A. Pawlikowski
T. Higgins	L. Marquardt	B. Zimmerman
E. L. Ivie	L. H. Morton	

The Technical Information Project - Meyer M. Kessler

The most important event in the past year has been the transition of TIP from limited experimental status to a public command facility available to all subscribers to the MAC system. Experience with this system has led to several major re-design efforts that will produce a more flexible and general information-retrieval system.

The TIP library has been expanded to thirty journals. The total number of journal entries on the disc is presently limited by available memory space to not more than thirty-thousand items. This is roughly three years of publications. Presently, as new material is placed on the disc, we are forced to delete the oldest entries in order to make room. Arrangements have been made for a much larger memory allotment in the forthcoming MULTICS system.

As the library on the disc grows, and as it is coming into more widespread use, new concepts of accuracy and maintenance must be applied. This requires a sizable editing program, which is now under way. In this connection, we shall soon switch from IBM punch-card equipment to Flexowriter paper-tape equipment. The chief advantage of this change is the expanded keyboard character set that will allow a more flexible and realistic format. It is hoped that the Flexowriter operation will also add speed and accuracy to the input process.

In addition to general on-line use of TIP by the Project MAC community, several extensive studies of TIP performance were made:

1) Processor Sanborn C. Brown, of the Physics Department and the Research Laboratory of Electronics, is using the TIP program to revise Basic Data of Plasma Physics, published in 1959 by the M.I.T. Press. This computer-based revision serves as an excellent test of user feasibility for the TIP program. The method used has been described in "A Bibliographic Search by Computer", Sanborn C. Brown, Physics Today, vol. 19, (May 1966) pp. 59-64.

2) Dr. Edward V. Ashburn has studied the laser literature and compiled a laser bibliography using TIP. The methods and analysis of this experiment will be published as a joint report by TIP and the American Institute of Physics. The resulting bibliography will be published in the Journal of the Optical Society of America.
3) A bibliography on "many-particle problems" was done on TIP by Dr. Stephen G. Brush of Harvard. This was compared with a manually-compiled bibliography.
4) Professor Elmer Hutchisson of Stanford University used TIP for a study of the productivity of physicists as a function of the size of the department.
5) A Ph. D. thesis, Search Procedures Based on Measures of Relatedness Between Documents was compiled by Mr. Evan L. Ivie. (See Ivie's report, elswhere in this section.)
6) An undergraduate seminar was give in the fall term of 1965-1966 by members of the Project TIP, with the cooperation of Professor S. C. Brown and Mrs. I. Y. Johnson.

An extensive program was developed by TIP, in cooperation with the American Institute of Physics, along a wide front of joint activities varying from the control and maintenance of editing and refereeing functions to the use and distribution of TIP-generated information. These activities make up part of a joint proposal to the National Science Fundation. Project TIP is now partially sponsored by NSF Grant N. 292.

Reconstruction of the TIP Programs - William D. Mathews

Two major subsystems make up the TIP system; a data reduction subsystem and a retrieval subsystem. Data reduction is under control of the programs REDUCE and EXPAND. REDUCE transforms formatted test input, with the TYPSET command, into a specially structured text called TIP text. The data is structured to optimize reading speeds for the retrieval subsystem. EXPAND will reverse the proceudre and put TIP text into a form which can be edited, again using TYPSET. Central to the data reduction subsystem is a user-supplied file called the "Field Table". As shown below, this file contains structuring information telling the REDUCE-EXPAND programs how to operate.

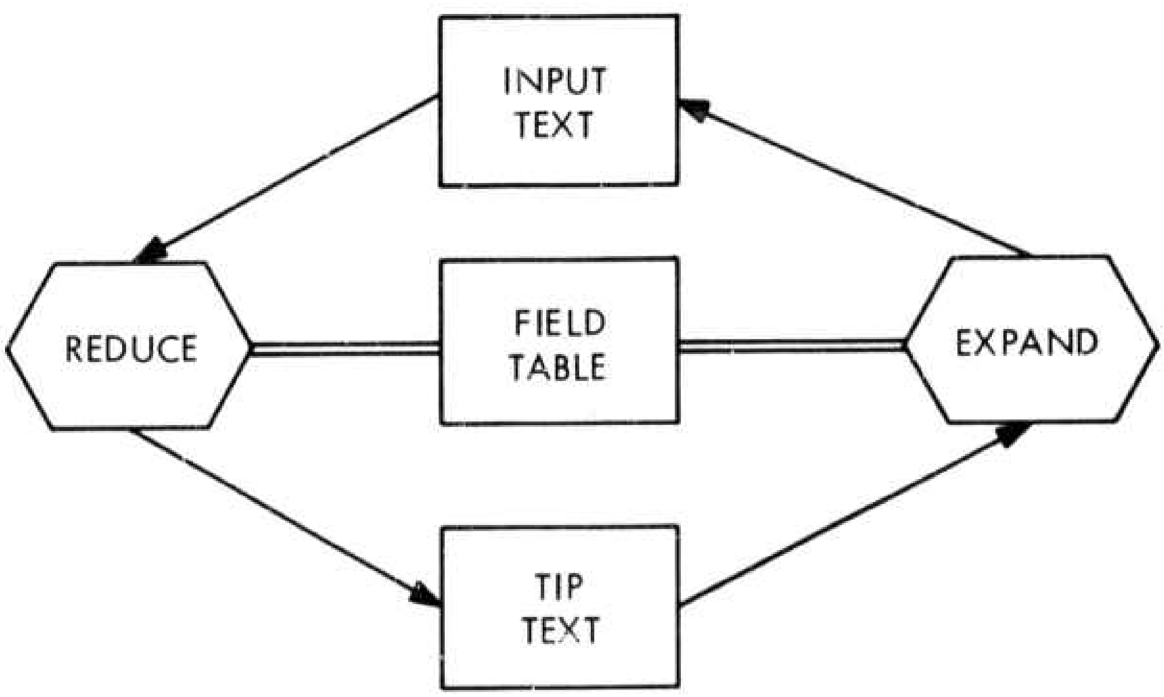


Figure 22. Project TIP Data-Reduction Subsystem

The retrieval subsystem, operating as the TIP command, under the control of user requests, reads TIP text and creats output text. User requests may be made directly from the console or stored in a disk file. The output file may be console printout, a disk file in the proper form for the RUNOFF command, or new TIP text. TIP text is read by referring to the "Field Table", and output text is formed by consulting the "Format Table", as shown below.

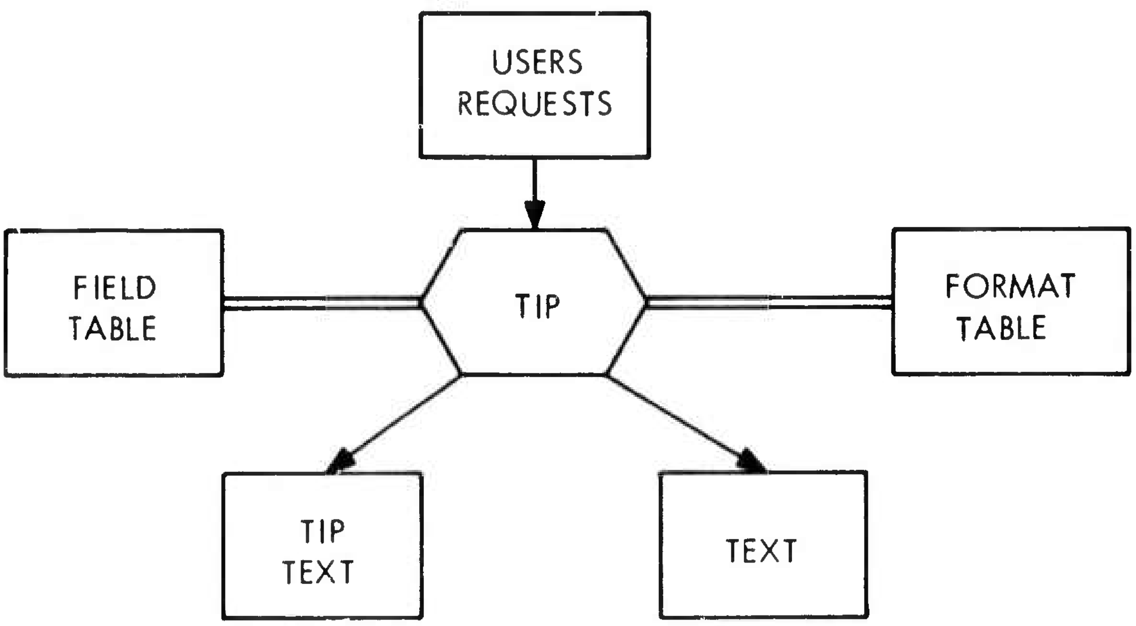


Figure 23. Project TIP Data-Retrieval Subsystem

The capability of the FIND request has been greatly expanded and now allows the user to specify strings with variable elements. Instead of finding literal occurences of specific characters, the FIND request can establish a criterion that will match on any subset of characters such as vowels or odd numbers. For example, the user may look for any string containing "c-vowel-t". With the element "-vowel-" properly defined, this request will find "cat", "cet", "cit", "cot", or "cut".

Algebraic relationships may also be included in the FIND request. For example, after writing a local file of personnel data, the user might ask for all salaries greater than $8000.

The reconstruction of TIP programs along these lines will place much greater emphasis on user-created data.

Search Procedures Based On Measures of Relatedness Between Documents - Evan L. Ivie

We have devised a new type of information retrieval system which utilizes data of the type generated by the users of a system, instead of data generated by indexers.

The theoretical model on which the system is based consists of three basic elements. The first element is a measure of the relatedness between document-pairs. It is derived from information theory. The second element is a definition of what constitutes a set (cluster) of inter-related documents. This definition is based on the measure of relatedness. The last element is a procedure which transforms a request for information into a cluster of answer documents.

Requests are made by designating one or more documents to be of interest and perhaps some to be of no interest. The requestor can continue to interact with the procedure as it locates the answer cluster by specifying as interesting or not interesting other documents which are presented to him. The answer cluster, which is generated automatically, is made as small (specific) or as large (general) as is desired, depending on the initial request and the subsequent interactions.

An experimental system was developed to test the model in a realistic environment. It was programmed for the Project MAC time-sharing system and utilized the physics data file of the Technical Information Project. Citations were used as the data base for the measure of relatedness. A file structure and retrieval language were designed which allowed close man-machine coupling.

Experiments were conducted which compared the clusters of documents produced by the experimental system with various sets of documents of known mutual pertinence. These sets included bibliographies from review articles, subject categories, and sets of documents found to be of interest to selected users of the system. It was found that between 60-90% of the documents of known pertinence were included in the corresponding clusters. Ways of improving this retrieval efficiency even further have been suggested. (See Ivie, Appendix B; MAC-TR-29, Appendix D.)

Formatting Output Files Within the TIP System - Kenneth D. Rude

Considerable attention has been given to the flexibilty of output from the TIP system. A user generally wants to see his files printed in many different ways. Sometimes he will be printing on a form which has restrictions requiring information to be printed only on certain lines or certain spaces. At other times he does not care how the information is printed so long as it is readable and identified. An initial solution to the problem of allowing the user full range of output possibilities has been to permit the construction of a "Format Table". With such a device, the user supplies a prototype item, together with control words similar to those used in the runoff command. Additional controls allow the overlaying of fields and inclusion of textual constants in any item.

A TIP On-Line System Dictionary - William A. Solomon

A new user of the MAC system is faced with several problems. One of these is mastery of the specialized vocabulary that has evolved along with the MAC system. New words have been coined, and common words are used in special ways. Acronyms are freely used in system documentation, usually without explanation or definition.

Since any user of TIP is of necessity also a user of CTSS, we seek to improve TIP user services by supplying an on-line dictionary-encyclopedia of terms relating to the MAC system as a whole. Items which are not already in the dictionary will be intercepted for insertion into the file the first time each is requested. Thus the dictionary may be automatically altered and updated.

System-Initiated Response From TIP - William A. Solomon

One of the communications problems facing a scientist is keeping up-to-date on current literature in his field. We are developing a TIP subsystem, called "MAC-TIP Service", which will notify any M. I. T. author whenever one of his articles is cited as a reference in the current TIP input literature. Notification is in the form of a computer-typed form letter which will contain information about both the quoted and the quoting articles.

An M. I. T. author receiving such information may then generate a search request, building up a typical FIND request for the retrieval system. New input to the TIP data files will automatically be screened against such requests, providing a system-initiated response on a continuing basis.

Sorting and Inventorying Of Large Files - William D. Mathews

Some work has been done on a massive sorting and inventorying of TIP text files. The Author Catalog of Books on Reserve for M. I. T. Subjects is a published printout from such a sorted file. It is now possible to obtain word indexes and frequency counts on TIP for any field. In the future, it will be possible to permute any field against any other field, allowing correlation data to be constructed.

Processing of Serials and Journals - Patricia M. Sheehan

During 1966, a disk file was created which contains information about serials and journals in the M. I. T. Libraries. Specialized catalogues and internal worksheets for library staff use have been produced from this file, as has the M. I. T. Press copy for the ninth edition of Current Serials and Journals in the M. I. T. Libraries.

The file is compatible with the TIP system of retrieval and selection. Specialized editing programs transform the file for printout purposes, allowing variations in content, upper and lower case characters, page length, and width.

Analyses made by the computer are being used to establish the practical limits of adapting subjective criteria of library filing rules to computer sorting techniques. Further steps are now being taken to efficiently use these files for more complete control of acquisition and maintenance of all journal holdings in the M. I. T. Libraries.

A List Structure for Storing Dictionary Information - Lewis H. Morton

A specialized list structure has been constructed for the TIP system. Lists are created in free storage and are maintained by the free-storage package. Each list has a name and consists of a number of cells, each containing some fixed number of registers. Cells may be added, deleted, changed, or renamed by referring to them either by name or position in a master list. Virtual lists may also be constructed and manipulated. These are directories to master lists, sorted by the contents of one of the registers in each cell of the master list. Alterations of cells in virtual lists are reflected by alterations in the master list.

Creation of a First-Cast Thesaurus from TIP Files - Elizabeth A. Dole

The basic purpose of subject-heading lists is to organize the contents of printed technical literature. The goal of my research project is to see whether there is some way to produce an acceptable thesaurus from the machine records of a large sample of titles in a connected body of scientific literature. I have been constructing and applying algorithms to word lists taken from the titles of physics-journal articles in the TIP files, and evaluating the resulting product. Title words are collected, counted, and sorted into two lists; one arranged alphabetically, the other by frequency of use. Each work is accompanied by its count. Eight algorithms have been designed, and seven have been tested on a preliminary data sample. Successive algorithms define a "word" more and more specifically. Words not meeting the requirements of an algorithm are rejected from the original list, and unit and usage counts are reduced accordingly.

The final list for 1964 title data, will be checked against the "List of Subject Index Headings" from Scientific Abstracts, Section A, Physics Abstracts, vol. 67 (1964). A single-word match will be made between the refined TIP title list and the standard list. Then a two-word match will be made using those associations between two words in the same title that appear with greater than random frequency.

Two possible applications of this research are: 1) the creation of first-cast thesauri for interdisciplinary fields, such as biomedical engineering; and 2) the delineation of changes in terminology with time, by applying these procedures to samples of the same kind of literature from different time periods.

LINCOLN LABORATORY

On-Line Data Storage and Retrieval
On-Line Experimentation
Compilation for the Digital Differential Analyzer

Non-Academic Research Staff

A. W. Armenti	J. W. Bex	J. F. Nolan
J. A. Arnow	H. K. Knudsen	

On-Line Data Storage and Retrieval - Amendio W. Armenti

The on-line data storage and retrieval system, described in previous progress reports, was modified to run under the new CTSS system which was introduced in the fall of 1965. At this point, the work of constructing the experimental model was completed, and the operating system was used as a vehicle for demonstrating the feasibility of a versatile, general-purpose, storage-and-retrieval system.

Use of the Project MAC time-sharing system was indispensable in providing to the project a convenient, expeditious way of programming the system; but more importantly, the means for making all the system functions available to terminal users as on-line operations. The five-man working group coded and debugged the system (about 30, 000 words) in three months. This is a reduction in time over use of the Lincoln Laboratory batch-processing facility by at least a factor of 10.

The finished model is now available as an operating system for general use by Project MAC subscribers. Since the system is a feasibility model, and therefore experimental, there are a number of restrictive features. However, a user can define files and relations, input entries, and modify files in the manner described in Lincoln Laboratory Technical Report TR-377.* Files created by a user are stored in the user's disk-file space in the form of pseudo tapes and he may generate as many of these pseudo tapes as his allotted disk space will provide. Each pseudo tape can hold as much as 7300 words of data, but this will vary with the number of relations defined in the files. (See MAC-M-314, Appendix A.)

*J. F. Nolan and A. W. Armenti, An Experimental On-Line Data Storage and Retrieval System, M.I.T. Lincoln Laboratory Technical Report TR-377, February 1966, EDS-TDR-65-36

On-Line Experimentation - Joseph W. Bex

Since last year's report, additional programs to simulate designs for the magnetic orientation system used on Lincoln Laboratory's experimental communications satellite (LES) were developed and debugged on-line. This work, dealing with a new system of partially unknown characteristics, was greatly expedited by the CTSS interaction facility.

An on-line searching process was developed for recalibrating the electronic systems on board the LES, now in orbit, and is currently in use.

Through the "linking" facility provided by Project MAC, Lincoln Laboratory Group 63 staff members were able to make use of Prof. Dertouzos' circuit design and analysis program (TED). TED was used to help design various circuits in LES.

Compilation for the Digital Differential Analyzer - Harold K. Knudsen

Principal parts of the DDA Compiler described in the previous progress report were coded and checked out. Work on the compiler was then suspended in order to complete the construction of the DDA. At the present time, however, because of unanticipated circumstances, the DDA program has been put into abeyance for an indefinite period.

MAN-MACHINE COMMUNICATION

Academic Staff

D. A. Huffman J. Weizenbaum

Research Assistants and other Students

S. T. Mooallem S. A. Ward

Guests

S. Lorch	-	Mass. General Hospital
M. T. McGuire	-	Mass. General Hospital
G. A. Miller	-	Harvard University
G. C. Quarton	-	Mass. General Hospital
P. J. Stone	-	Harvard University

Introduction - Joseph Weizenbaum

During the period covered by Project MAC Progress Report II, the ELIZA program was developed, which had for its objective the simulation of one participant in a two-person conversation. The input-output of this program was via a MAC teletypewriter console. During the current reporting period this program was subjected to intensive testing – at the Psychiatric Research Department of the Massachusetts General Hospital under Dr. G. C. Quarton. et al – and modification. The principal inprovement was a feature which permits the program to establish and maintain intermediate contexts in the actual course of an ongoing conversation. The ELIZA program has also seen considerable service as a MAC demonstration device.

A Teaching Script for ELIZA - Saul T. Mooallem

The use of natural-language communication, through the facilities of an existing computer program, ELIZA, has been used for the study of man-machine natural-language conversation, and it has been found that programmed instruction can be made more adaptive than with conventional techniques. The program operates on the input text in conjunction with a script, consisting of list structures headed by keywords and composed of decomposition and reassembly rules. The input is scanned for keywords and decomposed accordingly; output text is subsequently created by the application of corresponding reassembly rules. A brief script for instruction in high school physics was devised, and this method of applying computer-aided natural language communication to programmed teaching was evaluated. (See Mooallem, Appendix B.)

<u>Man-Machine Natural-Language Communication</u> - Gardner C. Quarton, Michael T. McGuire, and Stephen Lorch (Massachusetts General Hospital)

A. DIAGNOSIS AND DISPLAY OF MISUNDERSTANDING

This study was designed to assess the phenomenon of misunderstanding in man-machine natural-language communication, and establish the conditions and limits under which natural-language communication between man and machine could be made plausible. A program and "script" similar to the original ELIZA program devised by Professor Joseph Weizenbaum was used, and included the following alterations: 1) elaboration of the recognition rules for analyzing natural-language input; 2) use of a variety of different replies, based on recognition of sentence fragments; and 3) memory features within the ELIZA program, forced-choice trees, and the pattern of communication which subjects establish with the computer. The "script" contained 147 key words and key words in context.

During the study, 24 subjects were tested. They were told, "You will be communicating with another machine." Whether or not the machine was actually to be another teletype with a human operator, or a computer, was left unsaid. After one hour of typed exchanges (tests were conducted in the evenings to facilitate rapid machine replies), 15 subjects felt that they were conversing with another human, five were uncertain, and four were sure they were conversing with a machine.

In this study, 80 percent of the "machine" replies were contextually and grammatically appropriate (as judged by two raters); 20 percent of the replies were either "out of context" or "grammatically incorrect". Replies falling in either of the latter two sub-categories were considered examples of machine misunderstanding. Ninety percent of these "inappropriate" replies were not recognized as such by the subjects. Briefly, this means that under the experimental conditions of this study, a relatively large number of errant machine responses (unrelated to the ostensible meaning of a subject's input) were not recognized by the subjects.

B. ELABORATION OF THE DIAGNOSIS STUDY

The written protocols obtained in the first study have been examined, and a more elaborate recognition system for unrestricted natural-language input is presently being developed. In this empirical development of recognition rules, the probability of input recognition is expected to increase while machine misunderstanding of input should decrease. Increased precision in recognizing input allows for more thorough diagnosis of the limits within which (and the conditions under which) subjects make allowances for different kinds of machine misunderstanding. In an attempt to qualify this problem, machine replies to input are being formally categorized to allow an analysis of the subjects' responses to the language exchanges, in terms of the replies in these formal categories and their effect on misunderstanding.

An additional reason for this elaboration of recognition rules, and categorization of machine replies, is to prepare tests for studying the common models with which subjects are viewing the world at the time of the communication interchange. At the present time, crude forms of such models are recognizable through the simultaneous pattern analysis of the man-machine exchanges and each subject's particular language behavior. Two measures to be used in testing the accuracy of the machine recognition of subject models are: 1) a hypothesis-testing script which, through an analysis of a subject's language behavior in response to the script, either confirms or denies the accuracy of the recognition; and 2) paraphrases of subject models from already-stored computer models, after a finite number of man-machine exchanges.

C. EFFECTS OF COMMUNICATION ON VERBAL BEHAVIOR

It appears from our research that the language behavior of subjects can be partially determined by the type of machine reply offered; independent of content or ostensible meaning. In present studies, five types of machine replies are being used: 1) inference-based replies, computer replies containing a fragment of the input sentence, with the addition of a few words to make the reply similar to normal conversation; 3) continuitives, computer replies such as "Go on please", where the machine has not understood the input, needs more information, and wishes to continue the exchange; 4) inappropriate replies of the "out-of-context type", when a subject uses a word or words that have one meaning and the computer misunderstands the meaning of the word in forming a reply; 5) inappropriate replies of the "grammatically incorrect" type, when improper English is used in replies. Control studies are being conducted in our laboratory using two directly linked teletypes, where a research assistant simulates the computer at one teletype.

D. TEACHING STUDIES

We believe that a time-shared computer can efficiently teach students certain subjects (which have both a logical base and a preferential sequence of learning events), and we have developed a hypothesis-testing script where students study and learn hypothesis-testing behavior from the evidence given by the computer. The computer simultaneously presents a student with problems to be solved and analyzes the problem-solution methods of the student. When presently planned revisions are included in ELIZA, diagnosis of the way students behave will lead to an identification of learning inefficiency and "blocks" in hypothesis-testing behavior. This identification will activate scripts designed to specifically illustrate to the student where his difficulties lie and help him attempt to overcome them.

E. PSYCHOLOGICAL REACTION STUDIES

This study is listed last because it partially overlaps those previously outlined. Among the subjects tested thus far, we have noted the following common psychological tendencies:

1) Subjects appear to unknowingly transfer ("project") the image of either a friendly or unfriendly person onto the machine, and respond to machine behavior as though it were initiated by such a person.

2) Most subjects "test" the limits of the computer in a nonsystematic way, but use a particular type of testing behavior which is commonly used to test the tolerance limits of another person.

3) Responses to machine failure (AUTOMATIC LOGOUT, CTSS NOT IN OPERATION, etc.) elicit highly personal and strong emotional feeling towards the machine. The response to such machine behavior often results in subject (and programmer) frustration, stubborness, inefficient use, failure to recognize program errors, etc. Man-machine rapport often breaks down, with ensuing withdrawal of the man from interaction, or mistreatment of the machine.

4) Denial of machine limitations. Subjects will often imagine that the machine may have a much greater capacity than it actually does, and tend to "deny" or not recognize machine responses which refute their beliefs about the machine's capacities.

5) Competition with the machine. We have noted that, after the initial "testing" and "play" period, individuals who continue to work develop a competitive attitude towards the machine.

We intend to systemmatically explore these issues as part of our future research plans.

Time-Sharing the General Inquirer* - Philip J. Stone (Harvard University)

Early in 1964, we began to design and program the General Inquirer content-analysis procedures for use on the Project MAC time-shared system. Some of the General Inquirer programs are designed to facilitate comparing two texts. They allow an investigator to develop an efficient set of rules which enable him to effectively describe, in terms of tag assignments, the differences between two texts. Confronted with any sentence, the investigator, using these rules, should be able to identify the text to which the sentence belongs. Several standard statistical procedures based on separate tag distribution frequencies can be used for making such discriminations. Using stepwise multiple regression, a formula is developed to identify each sentence by the probability of certain tags occurring within it. The formula, however, usually considers the probability of each tag separately, and not information of co-occurrence. (Co-occurrence information can be added as additional variables to multiple regression procedures; the problem of course, is that a variable has to be added for each co-occurrence possibility that the investigator wants to consider. The number of variables soon becomes unwieldy.) Instead of using multiple regression, our procedures have instead focused on co-occurrence information by the use of tree-building techniques.

A time-shared computer may not be extremely large; however, it invariably requires a large, random-access information storage device for fetching information associated with each user's problem as the user requests it. In this way, the investigator, sitting at a typewriter console, can repeatedly regroup sentences according to different characteristics and identify patterns in the text which are relevant to his hypotheses. He can thus rapidly rework his data (by developing and pruning different trees) until he is satisfied with his analysis.

*Abstracted from Experiments in Induction, by E. B. Hunt, J. Marin, and P. J. Stone, Academic Press, 1966

The data are transferred from magnetic tape to the Project MAC time-shared computer disk at some time prior to the session at the typewriter. The investigator interacts with the computer and his data via a typewriter console and transmission line; he begins by calling the content-analysis program and specifying the names of the texts he wants to compare. In order to save disk space, only the tag descriptions, not the original text itself, are stored. Rather than print retrieved sentences, the computer types the document and sentence numbers, and the investigator can quickly look up sentences in a previously prepared "text and tag listing". Because of the relatively slow typewriter speed, the programs do not print out tables of frequency counts, but rather just print those tags and tag combinations that show as least a specified frequency relationship between the two texts.

While the time-shared system offers flexibility, ease of use, and closeness to data, two factors in the present MAC system greatly limit its utility. First, the user is held to the printing speed of a typewriter, which can be tediously slow when compared to a 600-lines-a-minute printer. A device is needed that at least matches the reading speed of most users. Second, storage costs have not as yet reached the point where it is feasible to keep large amounts of data on the disks. As mentioned, the information to be analyzed must be transferred to the disk from magnetic tape at some time before the analysis is to begin. Usually, storage allocation logistics require that this information be deleted or returned to magnetic tape before other information can be put on the disk. We expect that all these present difficulties will be resolved in the near future. Indeed, in the future reintegrated system, the user should be able to use the time-shared aspect of the system to monitor all phases of our content-analysis processing that now use batch procedures.

In the meantime, our readily accessible 1401-1460 machines have proved to be the preferred device for regrouping operations. Large files can be kept on tape and repeatedly passed through the smaller computer for regrouping;

any resulting new files being either printed or written on another tape. Complicated analysis procedures may involve handling a number of different tapes. Yet one can duplicate all the regrouping operations possible on the time-shared system.

NON-M. I. T. USERS

Interactive Social Psychology Experiments

Generalized Desk Calculator

FAMOUS: An On-Line Algebraic Manipulator

I. E. Sutherland	ARPA-Department of Defense
R. W. Taylor	ARPA-Department of Defense
R. A. Kirsch	Bureau of Standards
R. Des Maisons	Harvard University
R. R. Fenichel	Harvard University
I. Hazel	Harvard University
M. C. Henneman	Harvard University
A. G. Oettinger	Harvard University
S. M. Pizer	Harvard University
A. S. Priver	Harvard University
A. Ruyle	Harvard University
P. J. Stone	Harvard University
K. Winiecki	Harvard University

Interactive Social Psychology Experiments - Philip J. Stone
(Harvard University)

Experimental strategies have been made possible by computer-driven teletypewriter consoles. With this combination, a subject seated at a typewriter-like mechanism can type messages, and receive responses in a standardized environment. The computer can be made to take a "role" and interact with the subject within the confines of that role, even to the point of insisting that the subject stay within his own role, should he digress. The messages exchanged can cause the typewriter to type at high speed or type asynchronously, like a human, with occasional corrected mistakes. Indeed, experiments by others have shown that it is often difficult for a subject to ascertain whether he is interacting with a computer or with another typewriter controlled by a human. In this sense, the social environment provided has the potentiality of passing Turing's test.

For the past six months, under NSF Grant GS-178, a seminar group at Harvard University has been experimenting with a variety of experimental designs, using a teletypewriter connected to the Project MAC computer and the recently elaborated ELIZA programming framework of J. Weizenbaum. Following recent experiments of W. M. Evan, showing that computer-administered multiple-choice tests are answered with more candor than human-administered tests, one of our investigations has had the computer administering highly contingent, open-ended interviews. Additional investigations have explored using the computer as a standardized "other" in interactional experiments, such as competitiveness vs. cooperation in the "prisoner's dilemma". Here, the computer may represent more than one other person. For example, reproducing the Bavelas communication studies, the subject is but one man in a five-person communication net, and each message he writes must be directed toward a particular member. When he completes a message, various incoming messages are returned to him. In still other experiments, the computer itself is programmed to violate role boundaries and observe the patterns with which different subjects handle such role deviations.

The utility of standardized interactional environments is multifold. On one hand, a setting is provided for the psychological study of each subject's interactional behavior, and his performance can be compared with personality test scores, background factors, etc. On the other hand, research may be focused on the effects of making changes in an environment. Two programmed environments may be constructed that differ in but one characteristic; then response differences between subjects in each environment can be used to study the importance and effects of that characteristic.

Essentially, the computer must perform an on-the-spot content analysis of incoming messages and relate these to the context of past interaction; so the computer must have stored dictionaries for identifying the meanings of words and phrases. A rudimentary parsing system may be needed to resolve ambiguities, although the computer usually has the alternative of occasionally asking for elaboration, if an incoming statement cannot clearly be interpreted. Further knowledge is needed about the priorities and memory-storage requirements of conversational interaction. Important differences, for example, have been found between the situation where the computer is responding to the lead of the subject, and where the subject is responding to the initiative of the computer. While our initial "scripts" have often been rewarding and useful, we have only begun to explore the requirements and possible heuristics for successful interaction.

The entire conception of interactive experiments in social psychology stems from the idea of a contingent interaction that, although programmed and hence deterministic, adjusts to the behavior of the subject. The simulation models have been developed as group projects, with the interactive console offering a centerpiece for discussion, implementation, and testing of ideas suggested by a team of behavioral scientists. The syntactic- and semantic- analysis procedures would not even have been attempted without the time-sharing facilities for making repeated revisions as cases are being tested. (See Stone, Appendix C.)

Generalized Desk Calculator - Anthony G. Oettinger and Adrian Ruyle
(Harvard University)

Project TACT (Technological Aids to Creative Thought) has completed its CTSS implementation of the Culler-Fried, on-line computing system; which functions as a desk calculator, generalized to perform calculations of numerical analysis. Several features not present in the original version have been added, with the most notable being a rudimentary facility for manipulating algebraic expressions.

To provide a useful remote terminal for users of this computing system, an ESL storage-tube-display oscilloscope was installed at the Aiken Computation Laboratory, Harvard University. This oscilloscope terminal, driven from M.I.T. by the ESL Display Console over three standard data phones, is shown in Figure 24. This terminal produced the displays shown in Figures 25 and 26.

The difficulties of discretized function representation are shown in Figure 25. The function $e^{i\theta}$ $(-n\pi \leq \theta \leq n\pi)$ is represented in the complex plane by a circle, traversed n times. In Figure 25a, the function $e^{i\theta}$, $-200 \leq \theta \leq 200$ (represented as 49 data points connected by straight-line segments), is shown after the first 14 data points have been plotted. Figure 25b shows $e^{i\theta}$, $-200 \leq \theta \leq 200$, represented by 38 data points. With enough data points, the plotted function appears, of course, as a circle traced $200/\pi$ times. However, this sparsely-represented version remains unrecognizable.

Figure 26a illustrates a direction field for the differential equation $y' = y - x$, in the region $-1 \leq x \leq 1$, $-1 \leq y \leq 1$. Figure 26b shows Picard iterates $y_0 = 1$, y_1, y_2, y_3, y_4 approximating the particular solution through (0, 1) of the differential equation $y' = -5y + e^{-x}$.

Project TACT also has a terminal, shown Figure 27, connected to Glen Culler's machine at the University of California at Santa Barbara, and a direct comparison of the original system with its CTSS descendent is being

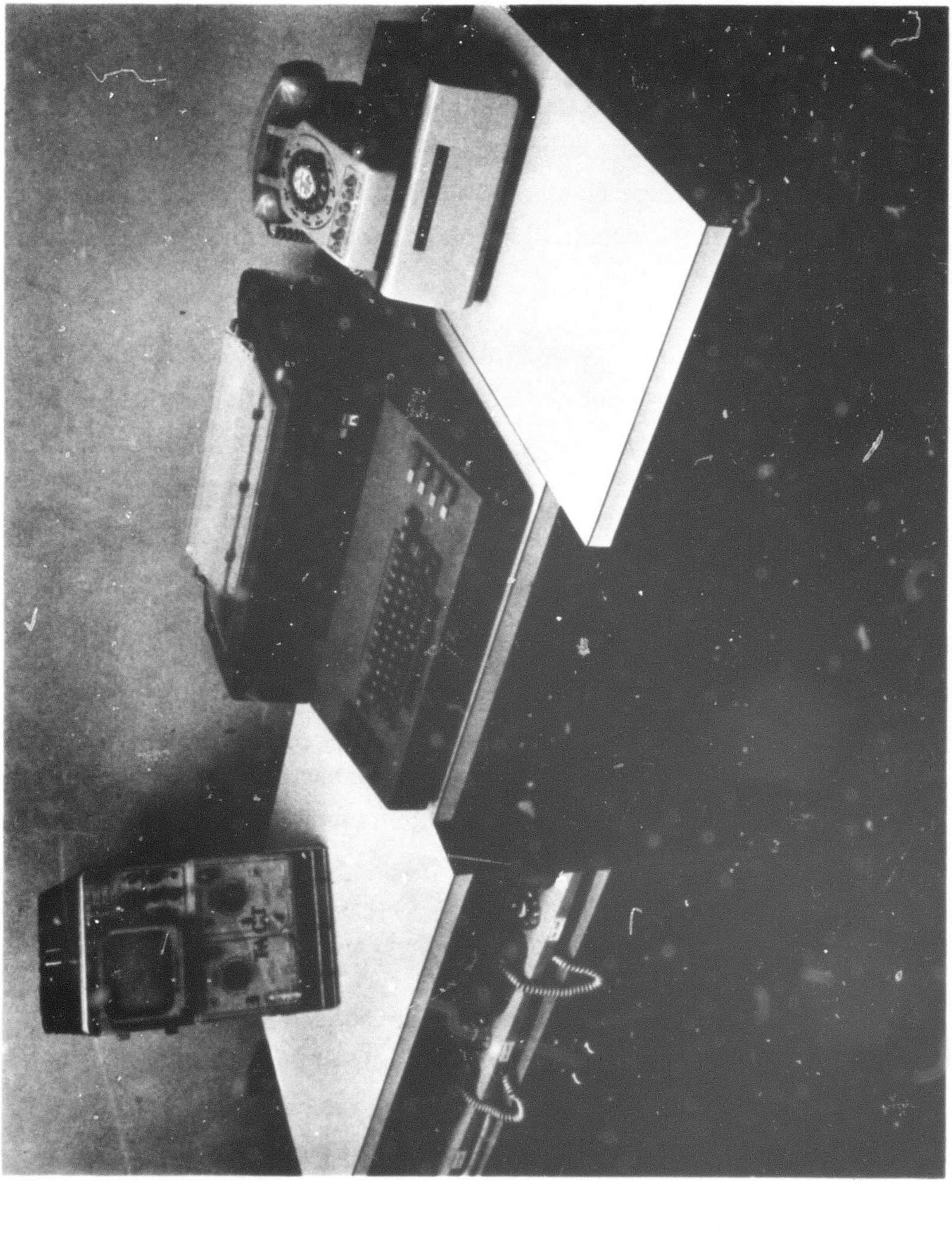
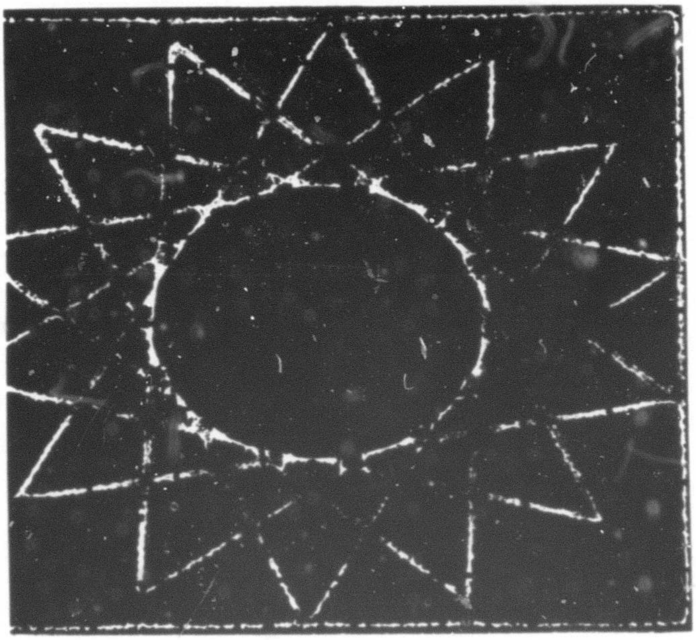
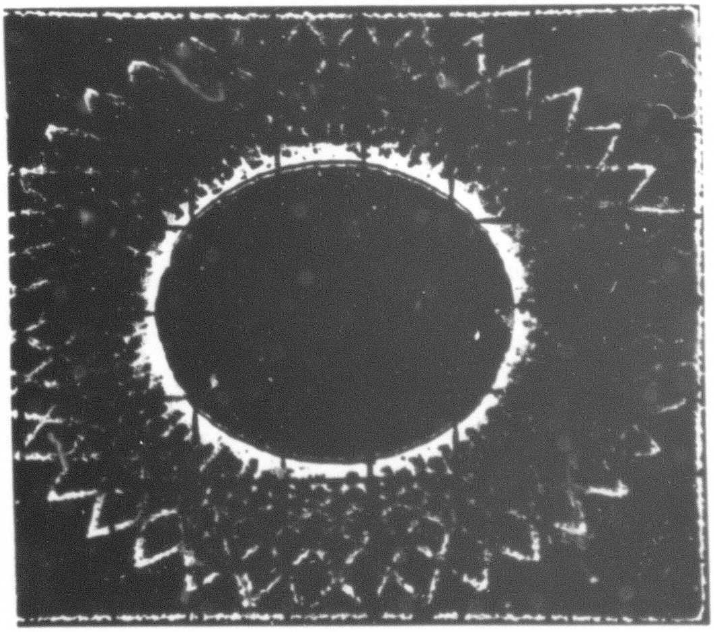


Figure 24. Project TACT Remote MAC/ESL Display Terminal

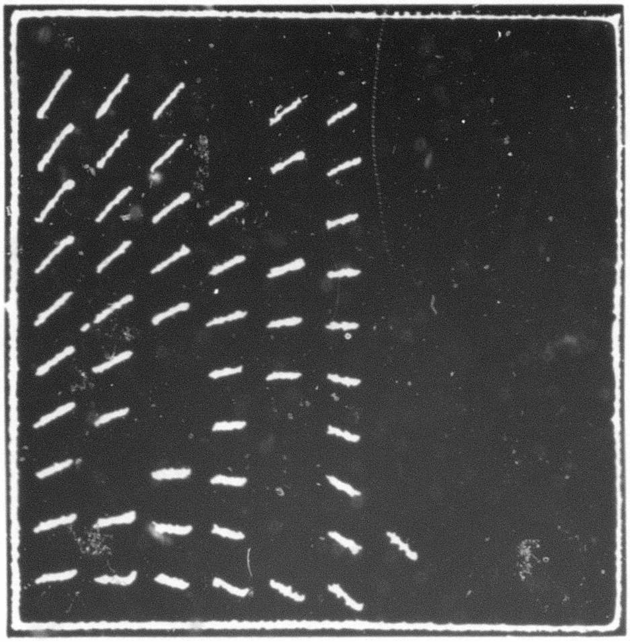


(a) 14 Points Plotted

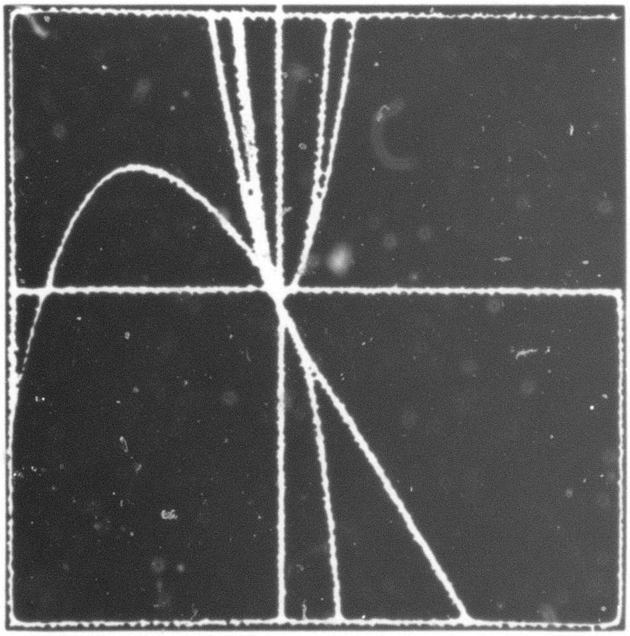


(b) 38 Points Plotted

Figure 25. TACT Display of $e^{i\theta}$ $(-200 \le \theta \le 200)$



(a) $y' = y - x$ Direction Field $(-1 \leq x \leq 1)(-1 \leq y \leq 1)$



(b) $y' = -5y \quad e^{-x}$ Picard Iterates $(y_0 = 1,\ y_1,\ y_2,\ y_3,\ y_4)$

Figure 26. TACT Display of Differential Equations

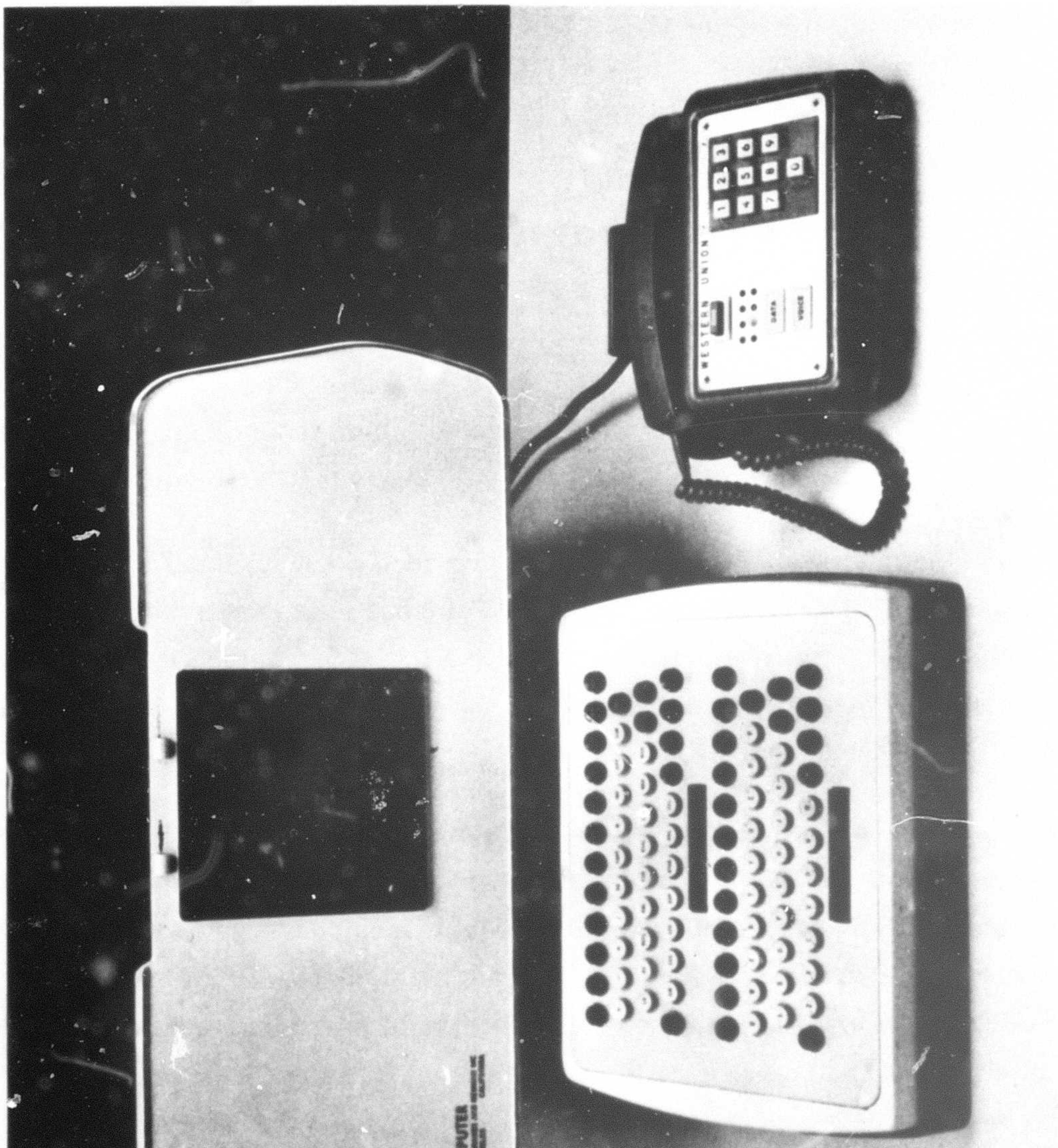


Figure 27. Project TACT Remote Culler-Fried Display Terminal

made, with emphasis on implications for design. Further comparison with the MAP and OPS-3 systems will be undertaken as soon as storage-tube-display programs for these open-ended systems can be developed. A survey paper comparing the various systems is planned for fall distribution by Adrian Ruyle, with John Brackett of M.I.T.

The TACT calculator is presently being used for research in biology and statistics, and will be used during the coming academic year for teaching a course in differential equations.

FAMOUS: An On-Line Algebraic Manipulator - Robert R. Fenichel
(Harvard University)

During the past year Project MAC has provided computational support for research work on my FAMOUS system (Fenichel's Algebraic Manipulator for On-line USe). This is a large program in LISP 1.5. Before FAMOUS was developed, I helped reorganize and improve the CTSS version of LISP. (See MAC-M-296, Appendix A.)

The primary motivations of FAMOUS have been the work on ם-theory ("Mem-theory") by A. W. Holt and others, and the recent unsolvability results of Daniel Richardson. These quite distant lines converge at the conclusion that global characterizations of algebraic manipulation is misguided. The FAMOUS system, consequently, uses bodies of unsequenced descriptions of local change. The transformations have successfully and evocatively been used to describe other algebraic manipulation schemes, and abstract automata and miscellaneous symbolic algorithms have also been described.

Work with FAMOUS has led to miscellaneous associated results; for example, limit problems, involving finite expressions of piecewise-analytic functions, are recursively undecidable. A set of FAMOUS transformations for limit problems of this sort does about as well as a college sophomore. The present line of research is completed, and will be presented shortly as a doctoral dissertation.

RESEARCH LABORATORY OF ELECTRONICS

Academic Staff

G. D. Bernard
A. Bers
S. C. Brown
W. D. Getty
M. Halle
H. A. Haus
D. Klatt
F. F. Lee
S. J. Mason
P. L. Penfield, Jr.
L. D. Smullin
K. N. Stevens
H. M. Teager
D. E. Troxel
H. J. Zimmermann

Non-Academic Research Staff

H. S. Davis
G. M. Fratar
J. M. Harwitt
E. R. Jensen
E. M. Mattison
V. E. McLoud
R. M. Nacamuli
M. M. Pennell
E. C. River
R. A. Sayers

Research Assistants and other Students

R. R. Bartsch
J. D. Bigham, Jr.
S. R. Brueck
T. H. Crystal
J. A. Davis
D. Haccoun
W. L. Henke
B. R. Kusse
M. A. Lieberman
D. T. Llewellyn-Jones
H. M. Schneider
C. L. Seitz
M. G. Smith
C. E. Speck
D. A. Wright

Introduction

This interdepartmental laboratory provides facilities for academic research in three categories; designated as general physics, plasma dynamics, and communication sciences.

Major support for this research is provided by the Joint Services Electronics Program of the Army, Navy, and Air Force, as well as the Atomic Energy Commission, the National Science Foundation, the National Institutes of Health, and the National Aeronautics and Space Administration.

As indicated by the following reports, a substantial number of R. L. E. research projects have received invaluable assistance from Project MAC facilities. These faculty and student research activities span a wide range of scientific and engineering subjects.

Grapheme-to-Phoneme Translation of English - Francis F. Lee

The result of this research is the identification and development of an efficient scheme for automatic translation of English text from letters to phonemes. The motivation arises from the specific desire for artificial-speech outputs from reading machines and also from the general desire to provide background for modeling the cognitive processes involved in human reading and speech.

One approach to the translation problem is the use of phonic rules, such as those used in the phonic method of teaching primary school children. An alternate approach is to use a word-lookup procedure. The phonic approach has a broad generative appeal; processing storage requirements will be limited to that needed only for the rules. However, phonic rules used for teaching children have been found inadequate and imprecise for use on automatic machines.

To determine whether phonic rules can be generalized to the extent that an algorithmic approach of grapheme-(primarily letter spellings) to-phoneme translation can be implemented for a reading machine, a body of 17,777 most commonly used English words, together with their pronunciation in General American, was used as experimental data. It was processed and placed in various forms so that graphemic context was plainly observable. It was found that while the consonant-letter mapping relationship is more regular, vowel-letter mapping is heavily dependent on the root word and suffixes. Furthermore, a detailed study of the cases of the paradigmatic suffixes (-ed, -er, -ing) revealed that the algorithmic approach, even when supplemented by large exception lists, not only leads to extreme rule complications, but leaves many serious problems unsolved.

The study led to the belief that phonic rules do not play a significant part in the mature reader's reading habit. His reading capability is based principally on the existence of the learned morpheme lexicon and the application of morphophonemic rules. Furthermore, it is believed that in reading unfamiliar words or nonsense words, morphophonemic rules are applied to the parts which agree with bound morphemes, if any, and phonic rules are then applied to the remaining unfamiliar or pseudo-morpheme. These phonic rules depend on personal experience and differ among individuals, thus accounting for the variation of responses to nonsense words.

The alternative approach of a straightforward word-lookup procedure for grapheme-to-phoneme translation would require a very large amount of data storage. By storing units at a level corresponding to morphemes, instead of words, it is possible to achieve a substantial (order of magnitude) reduction in this data-storage requirement. A lexicon containing on the order of 32,000 selected morphemes and words can be used, together with algorithms, to give phonemic translation for a vocabulary equivalent to what is contained in the Webster New Collegiate Dictionary. The algorithms make use of a search-and-compare procedure which provides the direct translation of simple words, and decomposes complex words into their constituent lexical entries in an orderly manner. With a lexicon, additional syntactic and semantic markers may be included. Problems the phonic algorithmic approach cannot possibly solve, such as those involving compound words containing the medial mute grapheme [e], syntactical ambiguities, and semantic symbolisms, may also be handled. (See Lee, Appendix B.)

Cognitive Information Processing - Donald E. Troxel

The Compatible Time-Sharing System has been used for both stimulus preparation and data reduction in conjunction with psychological tests probing the nature of the human reading process. A monograph describing the nature of these tests and the results is in preparation by Dr. P. A. Kolers of R. L. E.

Many additional blocks were programmed for inclusion within BLØD1. These were all related to the problem of simulating the behaviour of complex digital systems before their construction with integrated circuitry. This method of simulation via a Project MAC console proved to be virtually unusable, due to the slowness and unreliability of the 1050 console when used as a curve plotter. With a better graphical display, such as that provided by the ESL Display Console, this technique would offer a powerful aid to the designer of special-purpose digital systems.

Jackson D. Bigham, Jr., a graduate student, has been studying the degradation introduced by a Delta Modulator on the detectability of sinusoidial signals embedded in white, band-limited, Gaussian noise, as related to modulator step size and sampling frequency.

A MAD program has been written to simulate the entire system. White Gaussian noise is generated, and a sinusoidial signal is added; the resulting signal is filtered using a bilinear Z-transform filter. The resulting signal is transmitted and reconstructed by a Delta Modulator model, and the signal is extracted using a cross-correlation detector. The extracted signal, a signal extracted before transmission, and a pure sine wave are compared to determine the degradation introduced by the Delta Modulator. Most of the simulation runs have been completed; however, the results have not yet been reduced and analyzed.

A Far Infrared Plasma Physics Experiment - D. T. Llewellyn-Jones

This laboratory experiment involves a far infrared wide-bandwidth Michelson interferometer, the output of which must be Fourier-analyzed in order to be meaningful.

The magnitude of the analysis problem posed by this type of experiment requires capabilities provided only by fairly large computers, which are used extensively for infrared spectroscopy. Time-sharing provides the opportunity to link the computer directly to the experiment, so the computer can process the raw data while it is being generated. This allows feedback between the operator and his experiment, to permit his interpretation of results as soon as a run is completed, and avoids the interruption inevitably imposed by batch processing. The advantages of using time-sharing include maintaining the operator's continuity of thought, and a substantial saving in total manhours.

A device has been constructed which will feed this data automatically into the Teletype from a digital voltmeter. Thus the experiment will have a fully automated data acquisition and processing system. Data are produced in parallel BCD form at rates up to one six-character (four digits, range, and polarity) BCD word per second; a rate which is well within the capability of a 35 KSR Teletype terminal.

The device, shown in Figure 28, is a simple electromechanical parallel-to-serial converter, the output of which simulates the action of the relay between the keyboard and printer units of the Teletype. The black box interface contains three elements -- buffer memory, sampling commutator, and substitute relay. Memory holds the digital voltmeter (DVM) output for one reading and presents it in Teletype code (which is 1-2-4-8 BCD, plus fixed bits for numerals). Sequential sampling by the commutator produces a time pattern of pulses, which simulates the output of the Teletype keyboard. Thus the DVM is operating the Teletype; the computer reads the DVM output. It is now being debugged and tested under actual working conditions.

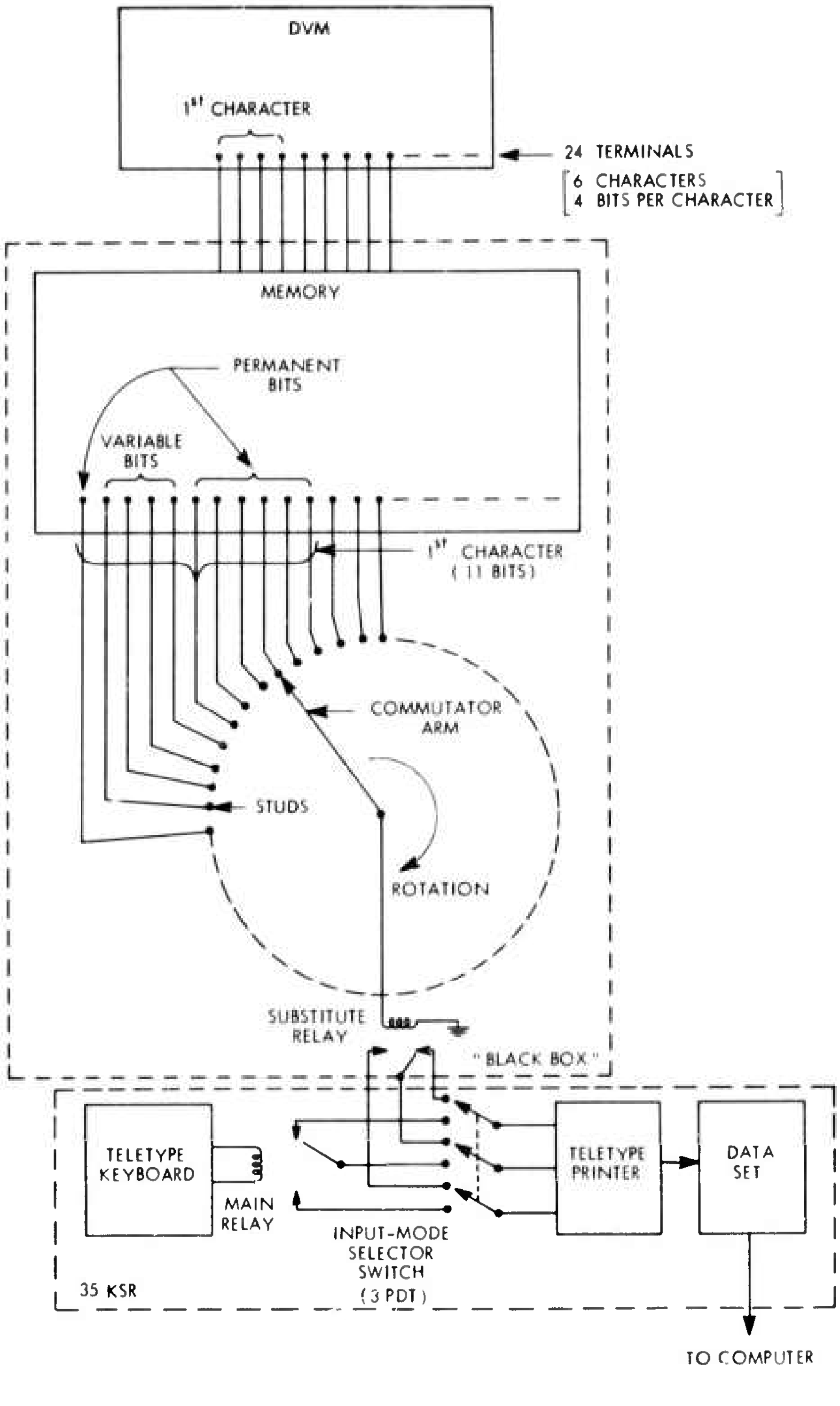


Figure 28. A Simple Electromechanical Parallel-to-Serial Converter

Analysis of Speech - Eleanor C. River

In the course of studying physiological constraints on speech production we have implemented a procedure which computes the acoustic natural frequencies of a speaker's vocal tract from appropriate measurements of the vocal tract. The input data are measurements made from tracings of lateral midsagittal cineradiographs taken during speech production. These measurements are obtained in a series of planes defined in a coordinate system which is located by certain anatomical landmarks.* Figure 29 shows the locations of the planes at which cross-dimension measurements are made. The lowest plane of interest is located at the glottis (in this case, at plane number 37); the most anterior plane is determined by the location of a marker at the corner of the mouth (shown as a dot at plane number -6 in this example).

One portion of the program applies to the measurements a set of transformations, devised from our knowledge of fixed anatomical data for the speaker, to obtain a function that specifies the cross-sectional area of the vocal tract at selected points along its length. A second portion of the program accepts the derived area function as input and determines the natural frequencies of the non-uniform tube by a Webster's equation approximation to the acoustic description of the vocal tract. By comparing the computed natural frequencies to the measured formant frequencies for utterance, we can test the validity of the above transformations. So far, the model has proven to be acceptable for cineradiographic frames selected from several utterances of one speaker.

The techniques of on-line interaction permit the acoustically sophisticated operator to perturb the assumed area parameters for a particular utterance and, by immediate inspection of the resulting natural frequencies, to determine the sensitivity of the spectral output to changes in particular regions of

* J. M. Heinz and K. N. Stevens, "On the Relations between Lateral Cineradiographs, Area Functions, and Acoustic Spectra of Speech", Proceedings of the Fifth International Congress on Acoustics, Lièges, 7-14 September 1965.

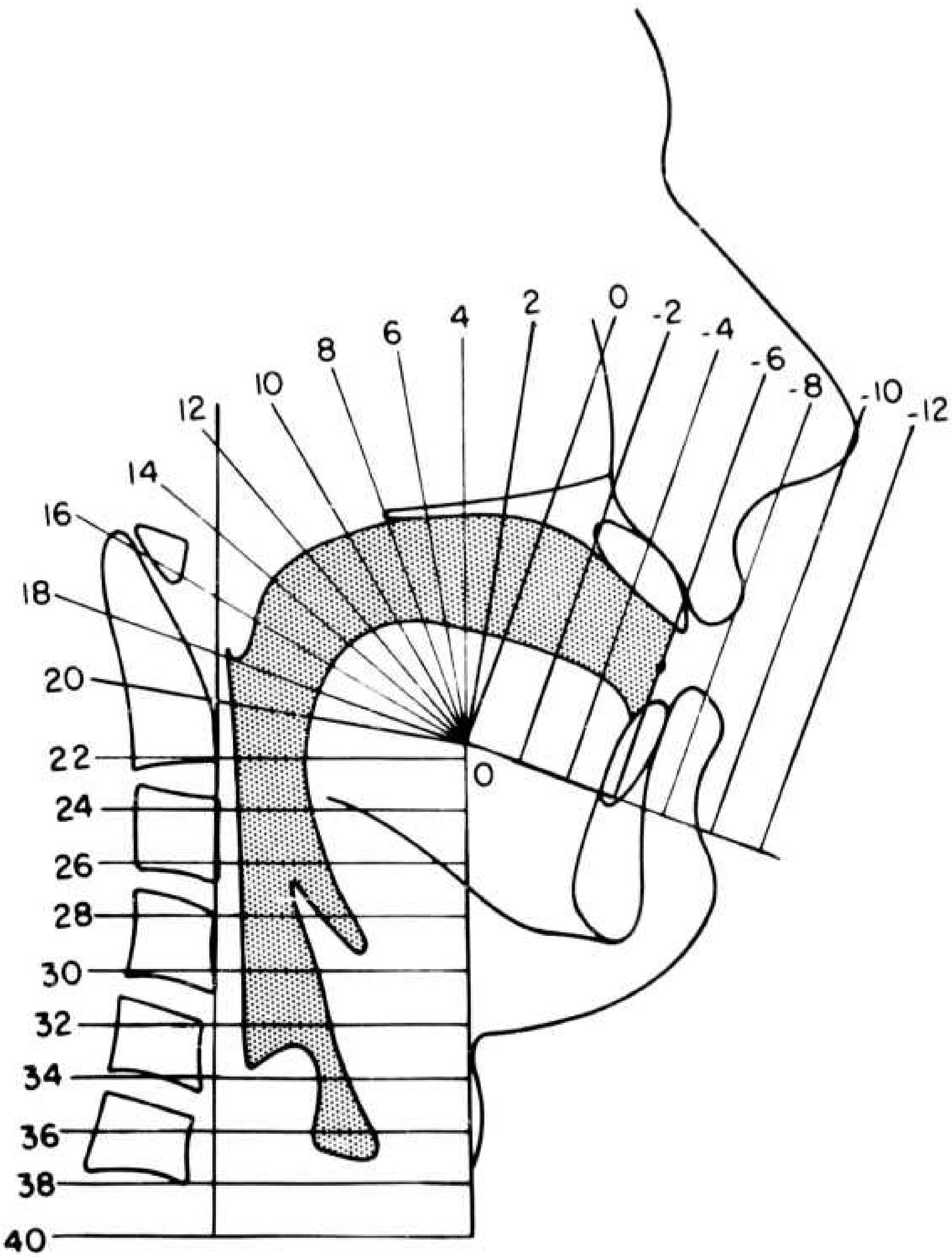


Figure 29. Coordinate System for Specifying Overall Vocal-Tract Configuration

the tract. This feature provides an indication of the accuracy with which the area function need be specified and, incidently, of the precision with which a talker must position various articulatory structures in order to obtain a desired acousic output.

Future work will be done with other speakers, to test further the generality of our assumptions. Programs written by another member of the R. L. E. Speech Group will simplify the input and storage of cineradiographic data by graphical means using the facilities of the ESL Display Console. As an educational device the projected system will permit the student to enter and to display the pertinent physiological information, to observe the idealized area function resulting from application of the transformations and, finally, to immediately obtain the resulting acoustic output.

Experiments Using a Larynx Model - Thomas H. Crystal

A model of the human larynx has been developed for simulating the portion of the pitch cycle when the vocal folds are apart and air is flowing through them. This model consists of two sub-models: one for air flow, and the other for mechanical motion of the vocal folds. The mathematics for the flow model is based on a representation of the aperture between the vocal folds as a narrow rectangular duct. The mechanical model is a simple harmonic oscillator driven by the pressure force on the walls of the duct. For low-speed air flow through the duct, the pressure is positive, tending to force the folds apart; for high rates of flow, the Bernoulli effect makes the pressure negative, acting to close the folds. The model is simulated on a digital computer in conjunction with models of the acoustic impedances looking into the treachea and into the vocal tract.

The motivation for developing such a model emanated from the desire to learn more about the speech production process and about the nature of the speech signal. Of interest in understanding the process is information regarding the function and effect of varying model parameters, representing physiological controls exercised by the talker, particularly in imparting intonation and stress to speech. Increased knowledge of the effects of larynx activity as observable in the speech signal will permit better separation and analysis of the roles of components of the speech production system. Improved analysis has implications for improved analysis-synthesis telephony and, considering the possible effect of individual differences in larynx structure and activity, improved techniques for automatic identification of talkers.

Model operation yields results which are in general agreement with findings from diverse experiments on human subjects. The results suggest that larynx operation varies between two limiting modes. When the laryngeal muscles are relaxed, operation depends on the pressure of the air between the vocal folds; when the muscles are tense, the pressure forces are negligible and the duration of the vocal source pulse is approximately the same as the period of the mechanical system. Experiments with varying model parameters indicate that there are simple relationships between parameter values and such measures of larynx activity as average flow, pulse duration, and speech intensity.

Important information has been gained about the possible nature and extent of the interaction between the larynx and the vocal tract. The high impedance associated with the first resonance of the vocal tract retards flow from the larynx into the tract at frequencies near the resonant frequency. This action, in turn, reduces the amount of excitation at the resonant frequency. The effect of this interaction on speech is a ten to thirty percent broadening of the frequency spectrum peak associated with the first resonance.

Computer simulation, rather than analog modeling, was indicated by the non-linear, time-varying nature of laryngeal events. Use of the facilities of Project MAC enabled the achievement of such a simulation without simultaneous loss of the major advantage of analog simulation -- the ability to observe behavior while making arbitrary adjustments on parameters. Such a facility was imperative for adjusting the more than two-dozen model parameters to give "normal" larynx operation, and was achieved by developing a command system for controlling model operation, examining and modifying model status, and printing and plotting the waveforms associated with model operation. (See Crystal, Appendix B.)

Dynamic Articulatory Modeling of Speech Production - William L. Henke

We are seeking to improve our understanding of the encoding of discrete linguistic signals into continuous acoustic signals by modeling the dynamic behavior of the vocal mechanism. The philosophies and methodologies of the model are beyond the scope of this report, so we will limit our discussion to computer technique aspects of the work. Since we are working with spacially distributed structures, we make extensive use of the graphical input/output facilities of the ESL Display Console. Because of the complex data structure of this model and the display requirements, the model could not have attained anything near its present form were it not for the availability and use of the AED Programming system. (See D. T. Ross, this volume.)

The dominant feature of our modeling methodology is a division of the model into a "state" which describes the articulatory mechanism, and into "operators" or "abstractions" which are related to higher-level linguistic descriptions. During a simulation run the (distributed) positions, velocities, and various types of quasi-forces pertinent to the state can be displayed for various structural parts (currently the tongue, both lips, and the mandible). Figure 30 shows an ESL Display Console view of the positions and velocities of the state of the model during a simulation of the phoneme input string /t u/. The velocities of the surfaces are shown by line segments (attached to the surfaces), which represent velocity vectors. This particular instance in time is shortly after the release of the /t/. Attributes of the current operator status can also be displayed.

A subsystem of on-line input, storage, and subsequent display of mid-sagittal-plan radiographs of human speakers has been developed for evaluation of the model. The abstract forms associated with our operators are spacial

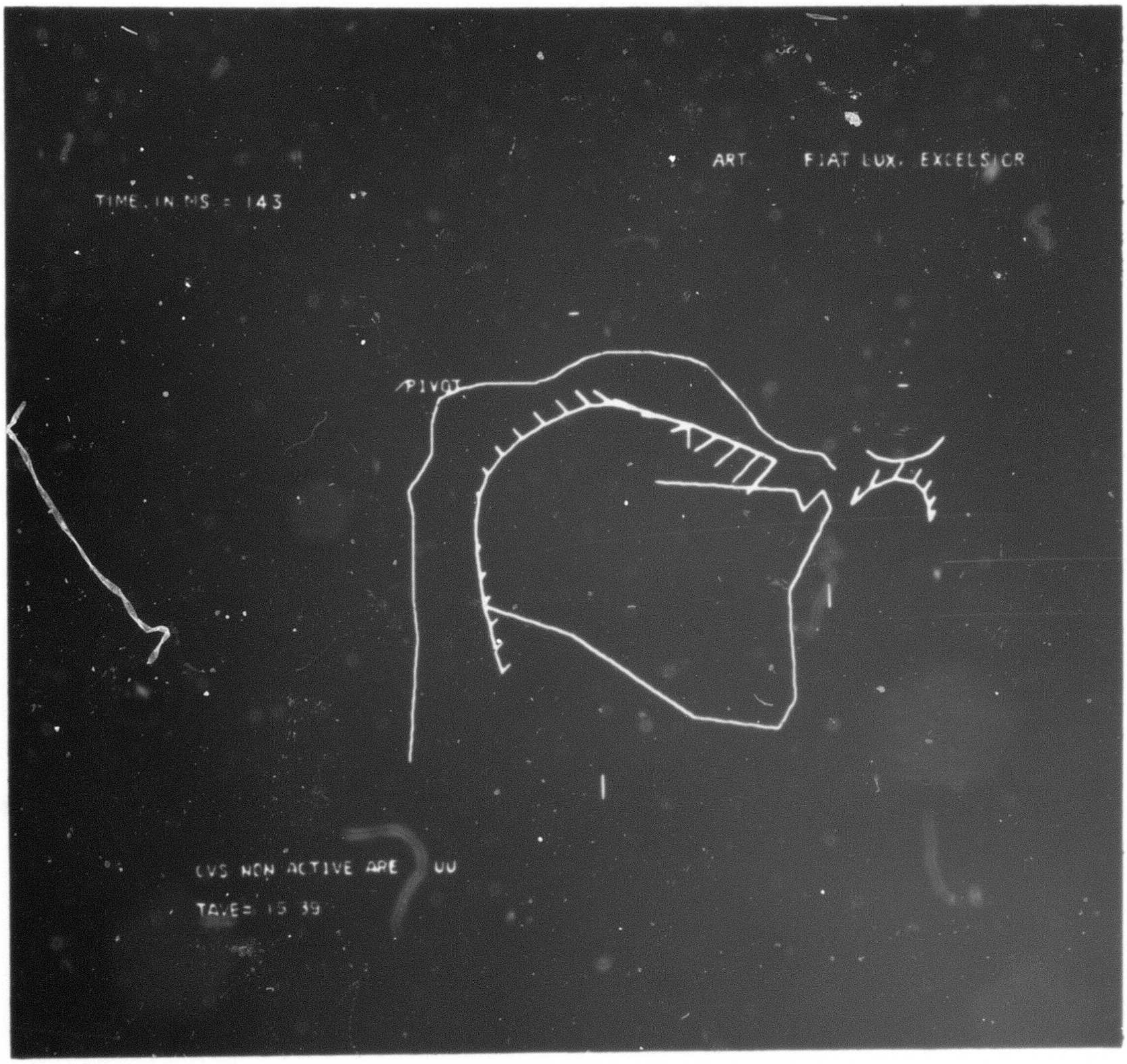


Figure 30. Display of Midsagittal Section of Modeled Vocal Mechanism

in nature, so a graphical inputing, editing and filing system (with commands similar to those of the CTSS command "ED") has been developed for these forms.

A motion picture film has been produced, using the ESL Display Console, which demonstrates both the model and these several graphical communication techniques developed in support of the model. Portions of the film were generated on a frame-by-frame basis, so that dynamic actions which cannot be observed "live" (i. e., on-line) could be animated.

Testing of Phonological Rules - Morris Halle and Francis F. Lee

Phonological rules of the general form

$$x_1x_2x_3 \longrightarrow x_1x_2'x_3 \qquad (1)$$

in which the x's are vectors in multi-dimensional space, possessing +, -, or 0 (don't care) values in each dimension, are frequently used in the study of phonology. The dimensions correspond to the distinctive features of the phonemes. In particular, it is always necessary to specify in expression (1) only certain dimensions for each of the x's.

A program has been written in SNOBOL to translate easily read phonological rules to a program in SNOBOL which can then operate on the lexicon. Rules considered are of the following specific types:

1) Replacement
2) Insertion
3) Deletion
4) Transposition
5) Minus-next rule
6) Alpha-rules for 1)-5)

Beam-Plasma Ion Heating by Beam Modulation - Gary D. Bernard

A theoretical model relating to the modulated beam experiments of the R.L.E. Active Plasma Group has been solved exactly. This model is a linearized hydrodynamic representation of a fully-ionized, macroscopically neutral, electron-ion plasma with longitudinal d-c magnetic field, excited by a sinusoidally varying line charage. Responses to line charge excitation are electron and ion velocities and densities, plus the radial electric field. Each of the responses depend on knowing the roots of the dispersion relation.

CTSS has been used in numerical work relating to this theory in the following ways:

1. Locating and approximating the cutoff frequencies, where the responses are singular,
2. Evaluating and approximating roots of the dispersion relation,
3. Learning the behavior of the responses as a function of the excitation frequency.

(See Bernard and Bers, Appendix C.)

Plasma Instabilities at Cyclotron Harmonics - Carlton E. Speck

Instabilities of quasi-static waves ($\bar{k}$ parallel to $\bar{E}$) are being studied at cyclotron harmonics for propagation across an applied field ($\bar{B}_0$) in a uniform, infinite, collisionless plasma, whose electron momentum distribution function is

$$f_0 = \frac{1}{2\pi p_{\perp 0}} \delta(p_\perp - p_{\perp 0}) \delta(p_{||})$$

where $p_\perp$ is the magnitude of the momentum across the field and $p_{||}$ is the amplitude of the momentum along the field. The ions are assumed to be infinitely massive. Figure 31 shows dispersion diagrams of slow-wave relativistic instabilities at cyclotron harmonics, and electrostatic instabilities between cyclotron harmonics.

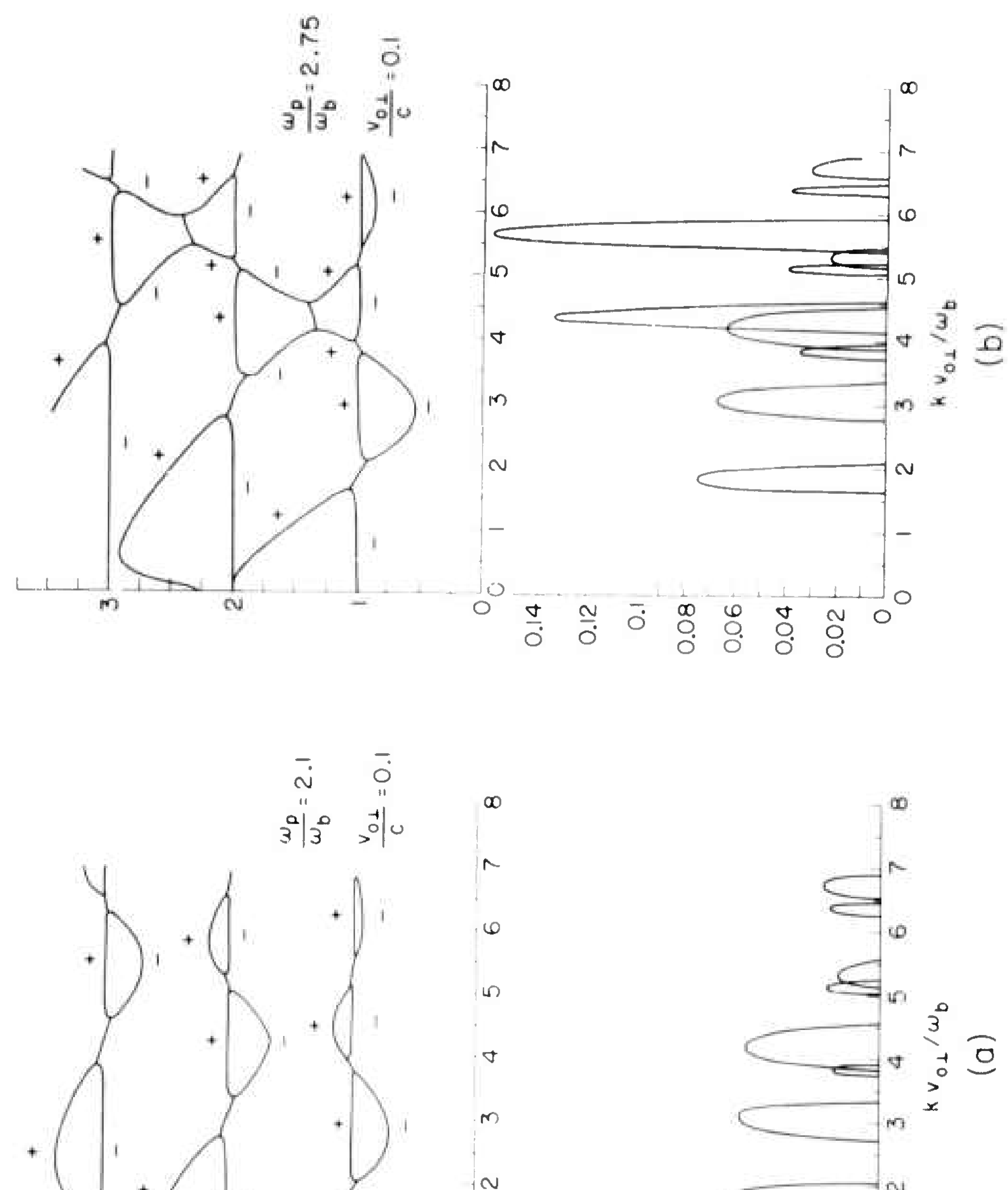


Figure 31. Dispersion Diagrams Showing Slow-Wave Relationistic Instabilities and Electrostatic Instabilities

The analysis is based on a relativistically correct dispersion relation that contains a dielectric tensor consistent with the exact Maxwell's equations. The waves have an assumed space-time dependence of the form $e^{j[\omega t - \bar{k}\cdot\bar{r}]}$. An instability is taken to exist when the frequency (ω) has a negative imaginary part for a real propagation vector ($\bar{k}$). The dispersion relation is solved numerically for its roots [$\omega = \omega(k)$], and two types of instabilities are found. The first occurs at high densities, and is interpreted in terms of the coupling of neighboring cyclotron harmonic modes. The coupling is shown to be that of an active (negative-energy) wave with a passive wave. The second type of instability occurs even at low densities, and is interpreted in terms of the coupling of the two waves which exist at each harmonic when the relativistic effects are taken into account. (See Speck, Appendix B).

A Model of a Beam-Plasma Discharge - Jon A. Davis

During the last year, Project MAC has been very helpful in debugging and testing programs for computer "experiments" concerning beam plasma discharge.

One of these "experiments" involved the heating of electrons by a travelling longitudinal wave in the presence of magnetic mirrors. It was found that the heating was considerable if the mirrors were idealized as hard walls, or if the electron reentered the field at a random phase after a mirror collision. However, if the mirrors were more realistically represented by a constant force, and the wave penetrated the mirror, there was hardly any heating.

Other computer "experiments" involved modeling, with one-dimensional charge sheets, the beam plasma interaction at electron-plasma frequency. At first, the plasma and beam were both represented by charge sheets. The plasma was cold and collisionless, and the interactions became so violent that plasma sheets were accelerated to velocities several times the injected beam velocity. This was felt to be due to the absence of energy loss.

Later "experiments" included collisions. Figure 32 shows the acceleration of a test particle and beam sheet velocity vs. distance, with time $= 300.4\ \omega_p^{-1}$ and $\nu = 0.2\ \omega_p$. Distances are normalized to $0.2\ \nu_o\ \omega_p^{-1}$ and accelerations to $0.2\ \nu_o\ \omega_p$. In this case, the plasma was treated analytically, and was assumed to remain linear. The beam was still represented by sheets. The interaction in this case was less intense, but the beam became well spread out in velocity, as shown. The electric field was a maximum at the point of initial overtaking of the beam, where a large charge bunch formed, although with considerable velocity spread. This research is being continued. (See Davis, Appendix C.)

Higher-Order Trapped Light-Beam Solutions - Hermann A. Haus, Martha M. Pennell

In their classic paper, Chiao, Garmire, and Townes† showed one beam profile they had obtained from a computer solution of the normalized equation for the normalized electric field E*

$$\frac{d^2}{dr^{*2}} E^* (r^*) + \frac{1}{r^*}\frac{dE^*}{}(r^*) - E^* (r^*) + E^{*3} (r^*) = 0$$

With the aid of Project MAC time-sharing facilities we have shown that the velocity of first passage is a monotonically increasing function of the launching height. This finding excludes the possibility of solutions with one single extremum different from that obtained in the paper given in the footnote.

The ability to interact with the computer was essential to our investigations. The correct launching height was found by intelligent guessing. We attempted to automate this scheme, but found the trial-and-error procedure to be much more efficient.

† Chiao, R.Y., E. Garmire, and C.H. Townes, "Self-Trapping of Optical" Beams", Physical Review Letters, Vol. 1964, pg. 479

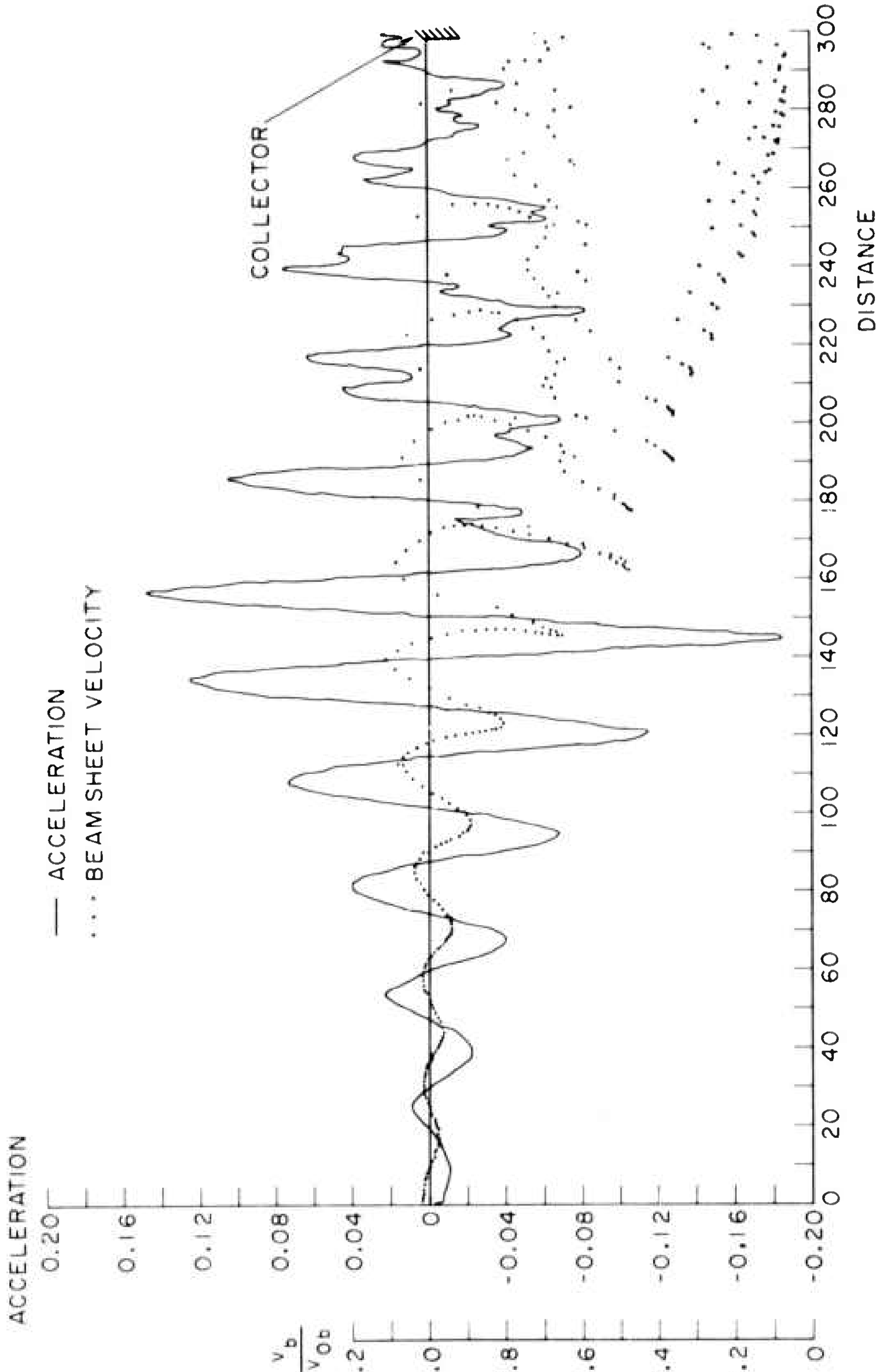


Figure 32. Acceleration of a Test Particle and Beam Sheet Velocity vs. Distance

Plasma Dispersion Relations with Infinite Roots - Martha M. Pennell

In our study of stability analysis for propagation across the magnetic field in a plasma, we have had to solve the following equation for γ $(0 \leq \gamma \leq 10)$, given real values for α and β

$$\frac{\gamma^2}{\beta^2 \nu^2} = \frac{K_{xy}^2}{K_{xx}} + K_{yy}$$

where:

$$K_{xy} = j \frac{\alpha^2}{\gamma} \sum_{n=-\infty}^{\infty} \left[\frac{n\{\gamma J_n J_n'\}'}{\nu(\nu-n)} - \frac{\beta^2 n J_n J_n'}{(\nu-n)^2} \right]$$

$$K_{xx} = 1 - \frac{\alpha^2}{\nu^2} \sum_{n=-\infty}^{\infty} \left[\frac{\gamma n^2 (J_n^2)'}{\nu(\nu-n)} - \frac{\beta^2 n^2 J_n^2}{(\nu-n)^2} \right]$$

$$K_{yy} = 1 - \alpha^2 \sum_{n=-\infty}^{\infty} \left[\frac{\{\gamma^2 (J_n')^2\}'}{\gamma \nu(\nu-n)} - \frac{\beta^2 (J_n')^2}{(\nu-n)^2} \right]$$

$J_n = J_n(\gamma)$, is the n^{th}-order Bessel function of the first kind. All derivatives are taken with respect to γ.

The availability of an on-line computation facility was extremely crucial in our investigations. Numerical techniques for solving transcendental equations such as the dispersion relation stated above rely heavily on one's intuition. We chose the well-known Newton's Method whose convergence depends on a "good" initial approximation. Where possible we obtained these mathematically. Nevertheless, shrewd guessing played an important role in regions where the roots changed from pure real to complex numbers. Here the features of time-sharing proved indispensable. (See Bers, and also Pennell; Appendix C.)

Studying Polynomials on the ESL Display Console - Maxim G. Smith

The R.L.E. Plasma Electronics Group has this quarter placed in operation a program for studying polynomial dispersion relations which can find multiple roots, follow the roots as physical parameters are varied, and plot the loci on the ESL Display Console. The program can be used by persons untrained in programming to obtain physically meaningful data about plasma system stability. Since the program can locate saddle and branch points automatically, the time-consuming trial-and-error methods formerly employed to do this can, for the most part, be eliminated.

Internally, the program uses a Newton-Rhapson iteration to simultaneously solve for zeros of two polynomial functions in two complex variables: the dispersion function and its derivative. The automatic data-graphing package is capable of drawing several graphs on each display console independently.

Programming Support and Development - Martha M. Pennell

The Computation Group has provided both programming support and development for RLE users of the MAC facilities.

Our Document Room has been given a demonstration of the time-sharing TIP command. In cooperation with Dr. Kessler s project, plans are being made to have more widespread use of the system by Laboratory members.

During the course of the year, we conducted approximately ten informal demonstrations of time-sharing. We found that experienced programmers were operational after an hour s session at the Teletype. A simple program illustrating the essential commands (LOGIN, LOGOUT, EDL, MAD, LOADGO, PRINT) together with a copy of the time-sharing primer proved sufficient for them to become productive users of the system. Perhaps the greatest difficulty we encountered was keeping these people informed on new developments (i.e., EDL instead of EDIT, etc.). Consequently, plans are being made for a refresher course.

The availability of an on-line facility is a great boon to the numerical analyst. Many numerical techniques are based on initial first guesses, which in many cases can be obtained only by trial and error. During this last year, we have spent considerable time investigating methods of solving transcendental equations. (See Pennell, Appendix C.)

Sorting of Personnel Records - Gail M. Fratar, Ronald M. Nacamuli

A computer program has been written and checked out to automate the personnel files of the Research Laboratory of Electronics. The personnel roster includes approximately 700 people who are classified by several categories: position, research group, room occupied, telephone extension, etc. Previously, these files were brought up to date and sorted manually. An up-to-date group listing took days to prepare; now such a listing is obtained as soon as we have access to the computer.

Sequential Decoding with Coherent Detection - David Haccoun

A few years ago Wozencraft proposed a procedure to decode convolutionally encoded messages. This sequential-decoding procedure is one of the few methods known for communicating over a noisy channel at very interesting rates. It is basically a tree-search algorithm and can be found in section 6-4 of Wozencraft and Jacobs.* Also, a sequential decoding simulation facility has been implemented by Neissen at Project MAC and uses a PDP-6 computer. (See Neissen, Appendix B.)

The purpose of this research is to modify Niessen's facility and simulate binary antipodal signaling over an additive Gaussian noise channel with a slowly varying phase.

---

* J.M. Wozencraft and I.M. Jacobs, Principles of Communication Engineering, Wiley, New York, 1965

The sequential decoder is being modified to include the channel, the generation of the signal and a detection method using phase reference. This phase reference is built by a technique termed "non-decision-directed-channel" measurement, in which the received signal is squared to remove the modulation, and where either past or future bauds can be used. This method, however, provides a double phase-angle estimate which has to be halved. The system will be modified to permit very easy changes in reference size, using either future or past bauds or a mixture of both.

Statistics will be gathered on the distribution of the number of computations per search, computations per block, etc., so that useful comparisons with similar systems (using different schemes) can be drawn.

Work is progressing toward finishing the different programs, and data is being collected. A thesis summarizing this work will be available shortly.

Simulation of Sequential Decoding Systems - David D. Falconer

A sequential decoder, for a telemetry system that utilizes coherent binary antipodal signaling, was simulated on the Project MAC PDP-6 computer using a sequential decoding facility developed by Charles Niessen. The system parameters were chosen to simulate operation at an information rate just below R_{comp} on an additive white-gaussian-noise channel with a signal-to-noise ratio of -6db per transmitted waveform. The simulation produced estimates of the undetected error probability, the average number of computations per decoded digit, the distributions of computation per search and per block, and the distribution of search depth. Operation of the system at a rate just above R_{comp} was also studied. (See Falconer, Appendix C; Niessen, Appendix B.)

At present, a composite decoding scheme is being simulated and studied. Basically, this scheme includes M parallel, independent communication systems, each consisting of a convolution encoder, a discrete memoryless channel, and a sequential decoder. Successive M-tuples of symbols transmitted via these M systems are words of a high-rate block code. At the receiver, after most of the M symbols of a block may have been decoded by the sequential decoders, the remainder may be decoded by an algebraic block decoder. Analysis of the system has shown that, for large M, operation rates up to the capacity of each channel is possible, with finite mean-decoding-time per block. Furthermore, the probability of buffer overflow while decoding a block goes down exponentially with M. The results of the simulation will appear shortly as an electrical engineering doctoral dissertation.

SCHOOL OF ENGINEERING

Project Intrex
The MAP System
The TREC System
Optimization Methods Applied to Ship Design
General Arrangements in Ship Design
Computer-Aided Room Arrangements
Two-Dimensional Stress Analysis
Three-Dimensional Stress Analysis
Stability of a Viscoelastic Surface Subject to Rolling Contact
General Plastic Stress-Strain Analysis
The Equilibrium Problem Solver
Identification of Nonlinear Dynamic Systems
Conformal Mapping of Free-Streamline Flow
The ENPORT System for Dynamic System Simulation
Analysis of Networks of Wavelike Transmission Elements
Symbol Manipulation and Evaluation for Engineering

Academic Staff

C. A. Berg
J. W. Brackett
S. A. Coons
K. F. Hansen
R. Kaplow
P. Mandel
F. A. McClintock
H. S. Mickley
C. F. J. Overhage
H. M. Paynter
R. C. Rosenberg
T. B. Sheridan

Non-Academic Research Staff

G. L. Benton
B. Fay
C. Garman
U. F. Gronemann
P. Kugel
R. Leopold
T. Pitidis-Poutous

Research Assistants and other Students

J. M. Alanko
W. C. Albertson
D. M. Auslander
P. Choong
D. L. Flamm
J. C. Free
M. L. Gentry
R. Haber
T. L. Hawk
N. B. Heubeck
F. M. Hickey
R. Hodges
B. V. Koen
K. J. MacCallum
H. S. Marcus
P. F. Meyfarth
R. P. Parmelee
G. Piotrowski
R. E. Raffo
R. Sayre
E. H. Sibley
D. Somekh
S. L. Strong
C. C. Tillman, Jr.
M. Weinberger
J. I. Weiner
P. A. Wieselmann

Project Intrex - Carl F. J. Overhage, J. Francis Reintjes, Gary L. Benton, Charles H. Stevens, Peter Kugel, Uri F. Gronemann, Douglas T. Ross, Alfred K. Susskind, and John E. Ward

Project Intrex (Information Transfer Experiments) was established by the School of Engineering to conduct an interdepartmental, interdisciplinary program of experimentation with new forms of library service. The primary goal is to evolve a functional design for a set of information transfer services that could be made fully operational in the early 1970's.

In the library of the future, as now visualized at M. I. T., access to books, journals, reports, and other holdings will be provided throughout the academic community by an electrically integrated network of information transfer devices, linked to sources and users outside the campus and coordinated by a central staff. In evolving a design for this network, Project Intrex will pursue experiments in three major areas:

1. Modernization of current library procedures through the application of technical advances in data processing, textual storage, and reproduction;
2. Integration of the Institute library into national and regional networks of libraries and other information centers;
3. Extension of the rapidly developing technology of time-shared, multiple-access computer systems as the central information-processing and control component in an information transfer network.

Experimentation has been initiated in the first two areas.

A. THE AUGMENTED CATALOG

The principal information-locating tool in the traditional library has been the union card catalog. Project Intrex plans to develop and test experimentally an unconventional catalog -- one which is digitally stored and concurrently accessible by many users through on-line computer terminals. This computer-based catalog will include books, journal articles, reviews, technical reports, theses, pamphlets, conference proceedings, and other forms of recorded information pertinent to a specific interdisciplinary field. The content, depth, and connectivity of the catalog will be considerably larger and more refined than a conventional card catalog. The availability of on-line data-processing services will also permit a functional augmentation, enabling users to interact with this store of catalog data in highly flexible and effective ways. The utility and cost of an augmented catalog will be determined through a set of operational experiments conducted with real users in an active library environment.

B. TEXT ACCESS

When a retrieval specification has been narrowed to a set of identifiers of the documents desired by the user, those documents must be delivered or displayed to the user within certain limits of time, quality, and cost. Project Intrex plans to experiment with a variety of possible solutions to the problem of text access in an information transfer network and to evaluate the following technologies:

1. For storage - print on paper, analog microimages on photographic materials, analog signals on magnetic or possibly thermoplastic materials, digitally encoded characters, and graphic elements on photographic or magnetic materials;
2. For delivery - transportation for some of the foregoing, electrical transmissions for others;
3. For display - direct inspection, Xerography, optical projection, oscilloscopic display, and the like.

In general, the text-access experiments will be designed to offer the reader a wider range of choices in speed of access, physical form of access, ability to retain materials, to use a loan, to purchase or rent a document, to use remote transmission or delivery facilities, or, if he prefers, to use the library in person. The goal will be to determine: 1) the true costs of different levels of service, as outlined above; 2) the amount of use by different readers of different modes and kinds of material; and 3) the subjective reactions of readers to different modes and levels of access.

Although Project Intrex is defined with respect to the M. I. T. community, it is expected to yield design information for the modernization of all large libraries and to advance the general technology of information transfer. (See Overhage, Appendix C.)

The MAP System - Roy Kaplow, John W. Brackett, and Stephen L. Strong

The state of the MAP System at the beginning of 1966 is described in Map: A System for On-Line Mathematical Analysis (MAC-TR-24). A number of important features have been added since the first edition of the manual was prepared in December 1964.

Of these, perhaps the most significant are the display procedures which, as a temporary measure, are available on the ESL console. It is now possible to plot from one to three one-dimensional functions simultaneously, on automatically displayed log or linear "graph paper". Functions may be plotted against independent variables or as functions of one another. These facilities have been tested with students in an undergraduate course (Metallurgy 3.51), using an assigned problem involving the calculation and display of electron wave functions. It is clear that display facilities, not requiring additional programming (as available with MAP), are a great advantage to most users, as well as in teaching. (Commands currently available are: plot, plot storage, and compare.)

Another basic innovation has been the command sequence facility, which allows sequences of up to 18 commands to be set up and executed at will (starting at any point in the sequence) and edited at any time. Since any (or all) of the included commands can themselves be requests to execute another command sequence, and since many of the ordinary commands are the equivalent of a normal "batch-processing" job, problems of enormous complexity can be handled with a minimum of effort by inexperienced computer users. (Commands available are: create, run, and edit.)

Internal programs have been developed, and subroutines made available, which now make it easy for relatively inexperienced programmers to add their own operations and procedures to the system, and to use them intermixed in any fashion with the procedures already available. All of the details of data transfer, disc access, and essential user-machine interaction are handled automatically. (The command available is: execute.)

All bulk input of data is now automatically under control of an editing program, and data can be corrected in BCD form just after input, or later if it is the most recent input.

A number of more specific operations have also been added, which extend or simplify the use of MAP. They are:

1. The derivatives of a function and the definite integral can now be referred to directly in equations. For instance, (d(x)=derif(g(x))) and (area = intf(g(x))).
2. A least-square fitting of any function, by up to five arbitrary fitting functions with linear unknown coefficients, can now be obtained. (The command available is: least square.)
3. There is a command which causes a function to be searched for those few points which will give adequate visual representation of the function in a hand- or machine-drawn plot. Both the allowable deviation and the maximum number of points to be plotted may be specified. (The command is: graph.)
4. Equations of the form g(x)=c may be solved. The answers are values of the independent variable, (x), which satisfy the equation. g(x) may be any function: e.g., (g(x)=h(x)+x**2/sinf(x)+derif(m(x))); and c any constant. (the command is: root.)
5. Experimental data is often not available at the equal intervals numerical procedures usually require. A command has been written to form, by interpolation, an equal-interval function from the required two tables of data: e.g., y(x), and x(x). (The command is: equalize.)

6. The output command print has been augmented to make convenient tabular forms of output available. (The command is: print table.)

Other work in progress, but not yet fully implemented, includes the following:

1. A function editor is being written to allow modification of non-BCD stored data. Alterations such as changing particular values, inserting or deleting values, and shifting the range of the independent variable are included in the editer.
2. An entire matrix-operation package is being developed. All normal matrix operations will be included, and MAP will detail such items as dimensions and tabular order, as well as check dimensional correspondence in combined operations.

We have also been experimenting with methods of displaying three-dimensional atomic arrangements. The rotation capability of the ESL display console is suitable for this purpose. We have been able to obtain excellent representations of perfect crystalline arrangements, as well as displays of lattice displacements due to thermal effects and other disturbances. This technique will also be very useful to illustrate disorder (as in liquids and glasses) and defects (such as dislocations or impurity strain fields).

Miss Barbara Fay of the Materials Center Computation Facility has assisted in programming some of the routines described in this report.

The TREC System - Kent F. Hansen, Billy V. Koen, James L. Lucey, Philip T. Choong, Michael L. Gentry, and Thomas L. Hawk

The Time-sharing REactor Code (TREC) system has been designed to make complicated nuclear codes readily available for all potential users in the Nuclear Energy Department. The principle system feature is collection of input data during a question-and-answer session at a console. Details of the system operation and its constituents have been given in previous reports.

In the past year, several additions and modifications have brought the total number of programs in the TREC system to four. First, experienced users have found that symbol-description printout during the input/edit cycle is unnecessary. Therefore, a new mode of high-speed input has been devised and incorporated into the basic QUIN (question-and-answer) programs. Second, output is normally printed at the console, but it seemed that for many applications a crude form of graphical output would be more suitable. For this reason, general-purpose graphical code has been added which may be shared by any of the nuclear codes. Third, a large, nuclear-cross-section library has been added to TREC for use by the CROSEC routine in connection with any particular nuclear code.

Finally, a major addition to the code library has been the modification of a multigroup, one-dimensional S_N code. By means of Carlson's Method, this code solves one-dimensional S_N equations for a maximum of 16 energy groups, 16 angular directions, and 5 spatial regions.

The necessary modifications consist primarily of storage reallocation in the BASIC array of the program's QUINTB, with corresponding changes in the data-analyzing subroutine, SNPSTD. In addition, to enable SNPSTD to more effectively consider the cylindrical case, an alteration was made to that portion of the subroutine correlating the number of energy groups with the number of angular regions. Several minor adjustments to SNPSTD were found necessary to prevent premature initiation of error routines during the edit phase of TREC.

The remaining programs of the SN code, SNPORT and SNCODE, were left intact, and the symbol table is being prepared. The entire code is believed suitable for incorporation into TREC.

TREC is now in general use by members of the Department. Users may link to the system from their own problem numbers, or may use a special departmental problem number. To control the general use of TREC, a multi-user monitor program is being written. The monitor will check for authorized users and their available time, and limit all users to TREC (i. e., only allow commands to "loadgo" or "resume" TREC). Time accounting and administrative details will also be performed within the monitor. (See Hansen, Appendix C.)

Optimization Methods Applied to Ship Design - Philip Mandel and Reuven Leopold

Drawing on techniques developed in such other fields as control engineering, operations research, and design of experiments, an optimization procedure has been developed at M. I. T. over the past two years (under partial sponsorship by the Naval Ship Engineering Center, Bureau of Ships, Contract No. Nobs 90100) that shows promise of being a very useful tool in expediting routine preliminary ship design. This optimization procedure determines the dimensions, proportions, and coefficients of the least-cost (amortized building cost, plus yearly fuel operating cost) or most-profitable (highest capital recovery factor, or any other specified economic criterion) ship for any given set of owner's requirements. The procedure relies on relatively crude empirical expressions to make all of the necessary calculations in various sub-areas of ship design process. However, as more sophisticated approaches are developed in these sub-areas, they can be readily incorporated into the basic procedure. Crude though it is, the present model of the ship design process honors most of the major constaints that are observed in the manual design of conventional merchant ships. There are three quite separate steps involved in carrying through an optimization study to its ultimate goal. These are:

1. Choice of an optimization technique,
2. Choice of an optimization criterion,
3. Choice of the mathematical model of the process whose results are to be optimized.

In the first category, the following formal methods developed in other fields were examined and compared, for their relative merit in general and in particular for the ship design problem:

1) Differential calculus, maximum-minimum techniques, with Lagrange multipliers,
2) Steepest-ascent methods,
3) Dynamic programming,
4) Various random-search techniques.

On the basis of this examination, the first three were discarded and the exponential random-search technique was chosen as most appropriate for the ship design process.

In the second category, the following criteria which should have either a maximum or minimum value were examined:

a. The sum RMS error of all independent variables,
b. Weighted multiple-parameter criterion,
c. Max-Ranking criterion.

In this category, it was found that basic differences between the ship design optimization problem and that of other fields disqualify the optimization criteria used in some of these fields entirely. For example, in control optimization, the standard of performance, which may be maximum allowable acceleration, velocity, or displacement, is assumed as fixed at the beginning of the problem. In the design problem the standard of performance, which is minimum cost or maximum profit, is not known at the start; it has to be found by solving the problem, since it is the end result. This difference eliminated the possibility of using the Max-Ranking criterion in the ship design problem. Furthermore, it was found that because of the nature of the mathematical model of the ship design process, two of the parameters had to be treated as part of the optimization criterion rather than as direct inputs to the design process. For this reason, only the weighted multiple-parameter optimization criterion was found suitable for the ship optimization problem.

The third category involves a choice of the multitude of relationships which are needed to define the ship design process in mathematical terms.

These relationships, and the order in which they are computed, are referred to as the mathematical model of the ship design process. Mathematical models for a cargo ship and a tanker were used in this research.

Neither of these models of the ship design process are composed entirely of closed-form mathematical expressions. For example, some of the relationships constituting these models are given in the form of tables. If all of the relationships were reduced to closed-form expressions, by least-square or other curve-fitting techniques, differential calculus techniques with Lagrange multipliers could conceivably be used in the optimization procedure. However, since many of the expressions of the ship design model would be highly non-linear, and therefore not solvable by direct methods, these techniques would be of dubious value. This is one of the reasons why the random-search technique was adopted as the optimization technique.

General Arrangements in Ship Design - James M. Alanko

A method has been developed, under partial support by the Naval Ship Engineering Center, Bureau of Ships, Contract No. Nobs 90100, which allows a computer to aid in optimizing general stowage arrangements in ship design. This method has been applied to the specific problem of arranging furniture in an individual stateroom.

The technique used requires that functional relationships be established between the furniture items to be arranged and the room floor plan. Relationships based on esthetics were not used; however, if they are neglected the ability to produce a satisfactory arrangement is not destroyed. Once these relationships have been established and numerically codified, the arrangement area is scanned and all possible spaces for each item are determined. All possible permutations of items in spaces are then evaluated and the permutation which optimizes the desired functional relationship is selected. In addition to its ability to scan hundreds or thousands of individual arrangements, the computer can save additional manhours by using a mechanical plotting or electronic display device to directly produce the plans of the selected arrangement. (Described by MacCallum elsewhere in this section.)

Even though the program's basic principles are very general, it will not produce acceptable arrangements for all types of rooms. The constraints which have been built into the program undoubtedly would not be compatible with a room whose function is to provide other than just living space: a classroom is a good example of this. The constraint of unobstructed floor space tends to push all furniture toward the outside edges of a room. By altering such constraints, and adding esthetic functions, arrangements of rooms other than those providing simple living space could be generated.

Also, arrangements produced by this method are only as good as the relationships taken into consideration. In the program which has been developed, only five considerations have been used.

Finally, this program is limited to the types of furniture which have been built into the program. Modifications would be necessary to allow different types of furniture. (See Alanko, Appendix B.)

Computer-Aided Room Arrangements - Kenneth J. MacCallum

Work is progressing toward developing a scheme for using a display console linked to a computer to produce arrangement plans for a ship design. It can be envisaged that a general arrangements program could eventually be combined with a background program for optimizing arrangements. (Described by Alanko elsewhere in this section.) The display console would then serve as a very flexible input device, before an optimizing program is run, and for making minor alterations to the final arrangement for functional or esthetic reasons not accounted for in the optimizing program. This would provide a good example of man-machine interaction with man contributing only the qualities most suited to him in the design process.

It was rapidly discovered on using the whole program for the first time that the initial drawing capability of the ESL Display Console was very inflexible. A "Pseudo-Pen" facility would increase the drawing flexibility, but this is not, unfortunately, where the main difficulty arises if drawing is being done in the "T" (horizontal/vertical) mode. (The ESL "Pseudo-Pen is described in memo MAC-M-256, dated July 13, 1965.) The Pseudo-Pen horizontal/vertical constraint causes lines to be horizontal or vertical from the end point of the previous line. This causes the position of any line but the first to be in error because of the previous end-point positions. The solution to this problem is to constrain the line being drawn to be horizontal or vertical not from its start point, but from its end point. This, of course, requires altering the previous line.

The first alteration was to eliminate the use of "rubber-band" lines. This makes the process of deleting and reinserting lines more straightforward. The next requirement was that a data structure of all lines drawn should be constructed so that any line could be referred to and altered as necessary. This is also convenient for other reasons as well; namely, that Pseudo-Pen requires a data structure to work with, and also for the utilization of background programs as mentioned previously.

In the chosen data structure each line bead has a pointer to the next line and a pointer to the last line. There are also pointers to each of the end points. The end points are four-word blocks and contain the data for the points. Each line, except the first, shares its start-point bead with the end point of the last line.

Now, when a new line is drawn on the display, the end point of the last line is examined and, if necessary, one of its components altered. This line is corrected and the new line can be plotted with the end point of the last line used as the start point of the new line. Simultaneous with putting the required correction on the previous line, the new line drawn will automatically be either vertical or horizontal.

An additional pushbutton feature which is closely connected to this will rotate the current drawing onto one of the orthogonal planes. If it is initially in the xy plane, it will rotate into the xz plane. Subsequent pushes of the button will rotate the picture into the yz plane and then back to the xy plane. This forms a necessary and convenient aid to drawing in three dimensions. Another pushbutton now will make the last line drawn invisible. This is necessary for construction lines when drawing in three dimension.

The above additions to the arrangements system of Murton* provide a much more natural drawing capability and, at the same time, a good basis for the addition of Pseudo-Pen and background programs. The system, of course, is not near the ideal, and much work still needs to be done. The addition of a data structure will mean that a great deal more control of data must be kept in moving or rotating pictures. The data, however, need only be altered for changes in set point and rotation matrix at discrete times; for instance, when an object is being "completed". Most of the refinements still necessary should become clear on operation of the system.

Although it is obvious that much of the initial work in this field is tedious and totally unconnected with the more classical problems of Naval Architecture, the potential of such a system is considerable and its development to a usable level rewarding.

*Murton, D. B., Computer-Aided Internal Arrrangements of Ships, M. I. T. Department of Naval Architure and Marine Engineering, June 1965

Two-Dimensional Stress Analysis - Paul A. Wieselmann

Various aspects of the solution to the two-dimensional elasticity problem, as formulated in terms of a complex variable, are expressible as functions of two complex functions. A canonical domain (e. g., the unit circle or the top half of the plane) can be used as an expedient to obtaining these functions, provided a conformal mapping of this domain to the domain of the problem can be obtained. Since the two solution functions and the conformal mapping function are analytic, they have complex power-series expansions. The series representations embody complete generality and the problem becomes that of determining the set of coefficients for a particular problem.

Previous work has been primarily in the area of determining a mapping function. A program to generate the coefficients in the Schwarz-Christoffel integral mapping function now exists. A simple program has also been written to obtain the coefficients of the mapping series from the integral expression. Parts of the programs to expand the Cauchy integrals of the boundary conditions into power series have also been written. The problem of numerical integration at a point load is a primary concern, although the previous work on similar problems associated with the Schwarz-Christoffel integral has been of some help.

An application of the Schwarz-Christoffel mapping function was made in connection with the work of P. F. Meyfarth (reported elsewhere in this section.) The problem was a free streamline potential flow and involved numerical conformal mapping of the infinite half plane into the correct Hodograph for the problem.

Three-Dimensional Stress Analysis - Richard P. Parmelee

Computer programs have been written for the stress analysis of two- and three-dimensional bodies. The programs are designed to permit engineers with little or no computer background to define their problem and obtain results quickly (typically one-half hour of console use and two minutes of computer time). The analysis, which is based on the method of Ritz, permits transforming an engineering statement of the problem (elastic properties and applied loads) into a set of linear-algebraic equations, automatically. These equations, which can be solved by over-relaxation, define approximately the displacements of an array of points (or nodes) in the body. This array of nodes, which must be topologically equivalent to a cartesian grid, can be arranged in any fashion to suit the geometry of the body, the loads and the expected solution.

Over the past two years, computer programs have been written to combine this analysis algorithm with suitable teletype input/output and graphical output. The results of test problems show that the program will produce very good results (0 to 8 percent error) for modest expense (one to three minutes of computer time). (See Parmalee, Appendix B.)

During this reporting period, the programs were modified and tested more extensively. The modifications were primarily directed toward improving the use of disk storage, in order to increase the size of problems that can be solved: because many interesting problems required more nodes (and therefore more equations) than could be directly accommodated in core memory, provision had to be made for their storage in disk memory. When complete, these modifications will raise the maximum problem size from 300 nodes to about 2400 nodes.

Other modifications were made to the input/output portion of the program to improve the ease with which the program could be operated. These improvements were part of the results of program testing carried out by Professor C. A. Berg. In addition to providing useful information about improvements in input-output, Professor Berg's use of the program provided a basis for further evaluation of the accuracy of the stress-analysis algorithm.

Stability of a Viscoelastic Surface Subject to Rolling Contact - Charles A. Berg

This study concerns the deformation of small irregularities on the surface of a viscoelastic body which is subject to rolling contact. The fundamental question to be answered by this research is whether a small irregularity (bump) on the surface will undergo net growth during a single pass of a roller. To settle this question, a surface of a Newtonian viscous incompressible body subject to rolling contact has been considered; the viscous body exhibits no recovery creep, and thus bumps on its surface should be highly susceptible to instability due to the loading of rolling contact along the surface. The equations of incremental deformation of the viscous body have been integrated by coupling the Rayleigh (1878) analogy between elastostatic deformation and viscous creep with the two-dimensional version of R. P. Parmelee's static elasticity program RITZ 2D. Preliminary indications of instability of the surface (i. e., net bump growth during a single roller pass) have been obtained. Investigations to confirm these preliminary indications are now being carried out. Investigating further details of the bump-and-roller interaction, when the bump and roller are themselves in contact, are planned.

The writer would like to comment especially on the man-machine interactions which have occurred in the course of this research. First, CTSS has afforded a flexibility in using computational techniques in mechanics which, in this writer's opinion, puts the use of these techniques on a par with experimental methods. For example, if a user desires to assess the effect of compressibility on the deformation field of a certain problem, he can quickly recall the problem data, change the material compressibility, and ask for a solution. This aspect of CTSS makes the use of computational methods lively, appealing, and extremely useful; the usefulness of computational methods employed in the flexible manner afforded by CTSS is felt especially strongly in research, such as the rolling-contact-stability study described here, in which computational procedures seem to offer the best approach for exploratory investigation.

Realization of the full flexibility potential of CTSS is hampered, however, by a lack of simple instructions about how to use the system. An example of difficult instructional material is the CTSS Programmers Guide. This document, the sheer size of which is formidable, seems to be oriented toward program writing, whereas one of the principal benefits which could be derived from CTSS is that a researcher might use existing programs to solve problems in his area of competence, irrespective of whether he could in fact write a program. It would seem, to this writer, highly worthwhile to devote attention to instructing people in methods for using the programs already on file.

General Plastic Stress-Strain Analysis - Frank A. McClintock

The analysis of metalworking operations, the calculations of the strength of designed parts, and the prediction of fracture would all greatly benefit from a facility for plastic stress analysis of arbitrary shapes. Typical situations are so complex that solutions can only be obtained by numerical methods, and at present there exist very few solutions at all.

The technique under investigation is to progressively modify assumed displacement fields in the light of the resulting unbalance of forces. The obscure behavior of the third-order, nonlinear equations requires at this stage that the programmer be included in each iteration. For the immediate problem (a crack under plane-strain tension), the iteration procedure is being organized by a variational technique.* Obtaining this solution should do much toward finding a general method for determing plastic stress and strain distributions.

The Equilibrium Problem Solver - Coyt C. Tillman and Grover C. Gregory

During the past year, development work on a console-oriented system for solving equilibrium field problems was completed. The system, called EPS for Equilibrium Problem Solver, is now available to all CTSS users.

The EPS system was designed to aid engineers and researchers in treating systems of elliptic partial differential equations. It is not an automatic device which accepts input of a restricted class, then produces tables of numbers for output. Rather, it is a command-based system which gives a user great freedom of choice in his manner of problem description and in the form and quantity of information supplied as output. This means, of course, that the user must know more about EPS than he would have to know about an "automatic program" in order to obtain answers. But, on the other hand, with EPS he may obtain answers to a much more general class of problems. In particular, he can solve non-linear problems and problems which at the outset are not completely defined in a mathematical sense – e.g., problems involving free boundaries.

To take full advantage of the EPS system, the user stations himself in the ESL display room, where he has access to both a remote keyboard and

*R. Hill, Journal of the Mechanics and Physics of Solids, vol. 11, 1963 pp. 305-326.

a display console. (EPS may be used without display hardware, but for many problems this hardware can be of great advantage.) The user types commands such as

```
SET a=e=h=1, b=c=d=f=g=0
DEFINE x=r*cos(theta), y=r*sin(theta),
    r=10*i, theta = pi*j/20
SET pi=4*atan(1) $
```

to specify various algebraic parameters needed for his problem definition, and commands such as

```
APPEND 2, 0, 10, 0, 10, 20, 2, 20 TO wedge
CLOSE wedge $
```

to specify the topology of a finite difference lattice. Commands of the form

```
REVIEW VARIABLE x, theta
REVIEW CURVE wedge
LIST variables $
```

may be used for verification of past input, and ones such as

```
PRINT u(1,5,7), uxy(1,9,20)*sqrt(y)
PLOT u(1, i, j)/(r*r) $
```

for requesting output.

Three features of EPS which give the system great flexibility and ease of use are its in-core algebraic expression compiler, its algebraic expression interpreter, and its macro processor. The first of these features makes possible the efficient treatment of algebraic parameters which have functional dependence on other parameters, while the second feature enables the user to specify numerical quantities in terms of arbitrary algebraic expressions at any point in his input. The third feature – the macro processor – provides a mechanism by which the user may actually modify the EPS command language. In particular, it enables the user to generate composite commands; constituted from any number of primitive system commands, or from other composite commands. Moreover, as the system recognizes a conditional pseudo-command, the macro mechanism admits to recursive macro definitions. (See MAC-M-284, Appendix A.)

Identification of Nonlinear Dynamic Systems - Joseph C. Free

Preliminary effort has been devoted to the search for a sufficiently ordered and general modelling structure to allow simulation of lumped non-linear dynamic systems, in such a way that circuits could be logically grown or trimmed in the synthesis process. Such a structure results from the combination of a minimum-element bond-graph set, known as gyrographs, for circuit representation, and a generalized piecewise-linear functional representation, known as hyperpolyhedral functions, for constitutive element relations.

To get a rough idea of the manipulations required to synthesize a model from measurements upon a prototype system, several "ad hoc" programs have been written and more are being developed. Concentration has been on mapping of nonlinearities, rather than growing of circuits, and the principal difficulty has been in developing a useful transient-error criterion which adequately reflects the mismatch between two structurally similar nonlinear systems.

Original research has been restricted to two-element systems. In the linear case this corresponds to those with all roots along the imaginary axis of the complex plane, if two different energy storage elements are considered; or all roots along the real axis, if an energy-storage element and a dissipative element are considered.

A digital laboratory to allow experimentation with the latter class of two element systems is now being programmed.

Conformal Mapping of Free-Streamline Flow - Philip F. Meyfarth

Generally, potential flow problems involving free streamlines (boundaries) are most conveniently treated by a conformal mapping technique proposed by Helmholtz and Kirchoff. A variety of problems involving only one free streamline have been solved analytically by this method. In problems involving two or more free streamlines, however, the required mappings cannot be expressed as simple functions.

A program for the solution of a problem with two free streamlines has been written and executed for a variety of flow conditions. The results show very close agreement with known exact solutions for those special cases in which a direct comparison can be made. The techniques developed for use in this program can be used as a basis for a more general program for treating a variety of problems with free streamlines.

The ENPORT System for Dynamic System Simulation - Ronald C. Rosenberg

ENPORT is an algorithmic procedure for the analysis of linear lumped-parameter dynamic systems, based on bond-graph notation and a state-space approach. Graphical procedures are available for transforming electrical circuits, mechanical schematics, and other notations directly into bond graphs.

A Dynamic Systems Laboratory implements the analysis procedure mentioned above and simulates the behavior of linear dynamic systems specified as bond graphs. The dynamic response variables are available as tables or plots, and rapid-access response plots are generated on a display set up by a digitally-controlled analog computer.

Professor H. M. Paynter has used the Dynamic Systems Laboratory as the basis of a self-instructional system for subjects inexperienced in the behavior of dynamic systems. Subjects were able to develop an understanding of first and second-order behavior based on experimentally derived relations. However, the process was inefficient. A test facility, in which subjects tested and modeled unknown (black-box) systems, was found to be a useful teaching tool.

Conclusions reached on the basis of these experiments with the system as a research and teaching tool indicated that a more sophisticated system was required, to permit useful simulations of large, complex dynamic systems. Accordingly, considerable effort has been devoted to studying properties of bond graphs of primitive elements, as a possible basis for ENPORT reorganization. These studies have resulted in the concept of a canonical bond-graph form for a dynamic system, and reprogramming to make use of this result is now being carried out.

An anticipated feature of the new ENPORT system will be the inclusion of aggregate bond-graph elements, such as PUMP a b, TRANSISTOR a b c, and other engineering-device models.

Analysis of Networks of Wavelike Transmission Elements - David M. Auslander

Transient analysis of dynamic systems has concentrated on simulation via lumped-parameter models. This leads to concise notation and efficient computation when dealing with systems of a basically lumped nature, but it is quite cumbersome and inefficient for dealing with systems of a distributed nature, especially when information is desired at the high-frequency end of the response spectrum.

In this work, transient analysis of dynamic systems is done by simulation with distributed-parameter models. The simulation uses a digital computer extensively, in much the same way as an analog computer is used for lumped-system analysis. The basic language of the simulation is that of bond graphs, and is part of the ENPORT system for dynamic system simulation being developed by Ronald Rosenberg, also using Project MAC facilities. Most of the simulation programming has been accomplished in the last reporting period.

General operation of the program is in two phases. In the first phase, individual components of the physical system are modelled on combinations of pure delay elements and static elements. All the dynamic properties of a component are accounted for by proper combination of delay lines, and the static properties are accounted for in the junction structure. The second phase is a system simulation that makes use of models produced in the first phase. An arbitrary network structure is input to the program by means of coded bond graphs, all necessary input parameters and initial conditions are supplied, and then the transient response of the system is calculated using wave-scattering variables as the internal representation of the state of the system. Various outputs may be specified by the user when the input parameters are specified.

Symbol Manipulation And Evaluation For Engineering - Edgar H. Sibley

This research is directed toward developing a computer system oriented to the general and specific needs of engineers, and is intended for those who may not be sophisticated in the use of computers. The system is intended to allow the user to define his symbols and syntax (or use some already available) and manipulate or evaluate the expressions he builds from them. It is assumed that the engineer's notation may at times be ambiguous; he may use the same symbol in different contexts in different ways, but the computer will be able, even so, to help in his manipulations, and ultimately resolve the ambiguities (subject to the approval of the user), or ask for user interaction if its interpretation is faulty or impossible.

A prototype system has been programmed, and some numeric and algebraic manipulations have been carried out to test its power. Examples of the possible generality have been realized by working out certain elementary procedures for recognizing and checking the validity of chemical and predicate calculus formulas. The principles of the system will be discussed in detail in a forthcoming mechanical engineering doctoral dissertation, in which the prototype system is fully described, examples are given, and discussion of necessary additions and future improvements are presented.

SCHOOL OF HUMANITIES AND SOCIAL SCIENCES

The CONCOM Project
Turkish Social Systems Analysis
Statistical Analysis of the Turkish Survey
Analysis of Turkish Voting Trends
Sociometric Analysis of Venezuelan Elites
Social Backgrounds of Political Elites
Social Science and Data Archives
The VENELITE Project
Social Science Surveys and Models
Investment Planning and Integer Programming
Dynamic Stability of Currency Systems

Academic Staff

J. M. Beshers	E. Kuh	J. A. Silva
F. Bonilla	I. d. Pool	J. Waelbroeck
J. DaSilva		

Non-Academic Research Staff

G. M. Bonham	S. D. McIntosh	J. Rizika
D. Levine	N. I. Morris	S. Sacks
P. McCrensky	J. Nagle	

Research Assistants and other Students

P. Berman	J. L. Fields	T. McElroy
P. Bos	J. Golden	R. McKelvey
T. Carroll	D. Griffel	L. L. Roos, Jr.
P. B. Clark	D. A. Kendrick	H. L. Selesnick
R. Crossley	A. R. Kessler	W. D. Selles
C. Ellison	J. F. Kramer	

The CONCOM Project - John F. Kramer and Herbert L. Selesnick

This project, directed by Professor Ithiel de Sola Pool, attempts to stimulate a mass-media communication system, so that a social scientist, given demographic-population data and media-exposure data among the population, can schedule a variety of messages via several media or vehicles and reproduce the cumulative exposure among different demographic-population types. Currently, a background version of the simulation is being re-programmed to run on foreground at Project MAC. We are nearly at the stage of being able to produce a population file which includes probabilities of exposure to each vehicle for each member of the simulation population. In addition, a new and rather versatile version of the Mosteller technique, for producing complete N-dimensional tables from several partial and subtables, has been programmed for use independent of the simulation.

The simulation is to be used as a research tool by Mr. Selesnick on mass communications in the Soviet Union, while Mr. Kramer will simulate a case study of a 1948 U.N. information campaign in Cincinnati, Ohio, complete with panel survey data.

Turkish Social Systems Analysis - Leslie L. Roos, Jr.

Research on building formal models, for aiding cross-sectional and time-series research on attitudes in developing social areas has been continued. A flexible way of reweighting respondent types, to estimate attitudinal changes under different conditions, and then compare the actual attitudinal distribution with the predicted one for purposes of model validation has been programmed. Using this technique, a number of different models of attitude change and development have been explored.

A. URBAN-RURAL COMPARISONS

Given the diversity of the Turkish peasantry, it has been possible to construct models of respondent types such that the attitudinal distribution within these types will remain constant in towns, cities, and metropolitan areas. Attitudinal differences between rural and urban areas are conceptualized in terms of distribution of these types. The models assume that a person of a given type has the same probability of holding a particular attitude where he lives in a village, in a town or city, or in a metropolitan area. From a theoretical point of view, the models assume that "urbanism" is an antecedent variable which affects the distribution of a number of control variables -- education, media exposure, and so on -- which in turn affects distribution of the given attitudes. Thus, when controls for these intermediate variables are run, and the proper weighting factors applied, urban-rural differences should vanish. If we are dealing with a model based on respondent types derived from education, then, when 30 percent of the village males with a middle-school education hold a given attitude, it is assumed that 30 percent of the males with middle-school educations living in towns, cities, or metropolitan areas hold this attitude. Naturally, this is a simplified picture of the intra-societal differences, but it should facilitate the initial approach to model-building.

Since there are about a dozen attitudinal items for which actual urban data are available, partial validation of each model is possible by generating town, city, and metropolitan data, and then comparing these data with the actual survey results from the urban areas. Such a procedure permits refinements of each model, and the testing of alternate sets of respondent types to produce a better fit between model outcomes and the actual data.

B. ATTITUDE CHANGES

Data from two available studies has permitted comparing changes in distribution of twelve responses. By various analytical techniques, it is possible to make a very strong case for the hypothesis that considerable attitude change took place between 1962 and 1963. A number of political and economic events in Turkey during this period are advanced as possible explanations for the changes. This hypothesized attitude change within respondent types formed the basis for comparison with a formal model, which dealt with changing attitudes as a consequence of changes in the frequency of different respondent types over time. In this time-series model, educational and population projections were used to estimate the distribution of village respondents for 1972 and 1982. The trend data and model output were then compared to make some generalizations concerning the relative importance of the factors influencing short-term and long-term attitude change.

Direct validation of this longitudinal model was not possible, but parts of the formulation were susceptible to indirect validation. In particular, since one variable -- education -- played a central role in creating the respondent types, it was possible that the effects of other factors, such as exposure to mass media, were being neglected in the model. Because some of the same factors almost certain to be present in the Turkish villages of the future -- more educated individuals, more radio listening, and more newspaper reading -- are found in the Turkish towns of today, it is possible to use cross-sectional data from the 1963 survey to estimate the importance of some neglected factors in the time-series model.

Statistical Analysis of the Turkish Survey - William D. Selles and Allan R. Kessler

We have summarized a large part of a rural development survey of 6600 Turkish peasants into fifty summary measures, using Project MAC to explore intimately the statistical relationships among these measures. We have utilized the statistical subroutines of Mr. James R. Miller from the OPS system (see Miller's reports, this volume.) and have developed additional routines in conjunction with the Data Banks Project.

In developing a system to perform this analysis, we have explored several new approaches for summarizing social science data; in particular, proportional-reduction-of-error measures of association and measures of consensus based on information theory. The system has been designed to offer programming flexibility and efficient response to the demands of a social science analyst, F.W. Frey, who is presently writing the final reports to U.S. A.I.D., the sponsor of the project, under contract no. AID/nesa-11.

Analysis of Turkish Voting Trends - Allan R. Kessler

By correlation and time-series analysis of voting behavior in national elections from 1950 to 1965, and of indices of economic and social development, we have been evaluating relationships among economic, social, and political factors in Turkey's development.

Sociometric Analysis of Venezuelan Elites - Allan R. Kessler

From a file of sociometric choices of friendship, communication, kinship, and common activities among 290 Venezuelan elites, structural characteristics of the Venezuelan society are being sought. Because of the disparities between our graph-theoretic models and the realities of our data, we are using our ability to closely interact with this large amount of data, to empirically aggregate the sociometric choices to reveal social cliques among our respondents, and to disaggregate the total set of choices into social groupings constructed from our a priori knowledge of the Venezuelan society.

Social Backgrounds of Political Elites - William D. Selles

The CTSS system has made possible efficient processing and analysis of a unique survey of elite groups in world politics obtained through an extensive literature search. Data were assembled for approximately four hundred distinct political units in twenty-five countries, covering a time span of about 150 years.

Comparative and trend analyses, based on group characteristics and social background factors of group members, were accomplished using analytic systems previously developed at Project MAC. The results of this analysis will furnish material for a forthcoming publication of F. W. Frey and W. B. Quandt on comparative political elites.

Social Science and Data Archives - Stuart D. McIntosh and David Griffel

Analysis of what social scientists do when working with a data base has yielded the following design criteria for a data base system:

1. The system must have the capability, under audit control, of integrating the data prototype contained in the code book and the data file. The user must have the ability to change either data or prototype; the system should maintain the correspondence between the two.
2. The system must allow a social scientist to build information and social science indexes both within and across data files. The summarization of these indexes is a major analysis tool, and may take the form of cross-tabulation or tree construction.
3. The system design should be embodied in a computer system such as Project MAC, where there is highly responsive interaction between man and machine.
4. The system must be user oriented. The goal is to place a social scientist on-line, where he may interact with his data without the aid of programmer, clerical help, or technical interpreters.

5. The system must allow the social science user to provide intimate feedback into its design and embodiment.

Our experience has shown that a system designed to solve the general problems of data management, if designed properly, meets the above design criteria. Therefore, we have undertaken the construction of a general data management system, utilizing social science data as its first application. In rough outline the substance of the system is as follows:

1. The Organizer is a subsystem which permits creation of machine-executable code-book information, both for data processing and data auditing.
2. The Processor is a subsystem which applies the executable code book to the data under control.
3. The Cross-Analyzer is a subsystem which builds indexes (trees) and also produces co-occurrence tables both within and across files.

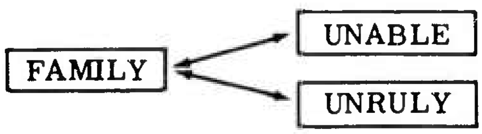
At present, the system is used as a pilot plant and is constantly improving as a result of procedural research and user feedback.

The VENELITE Project - Frank Bonilla and Peter Bos

The VENELITE Project is an attempt at a more sophisticated method of analyzing qualitative interview data. In most projects of this nature, the original data is in the form of a fixed-format survey, and the coding for program use generally breaks this down into "Who said (or did not say) what, how many times." Whole concepts are broken down into a simple "yes" or "no", with no explanation or qualification admitted. The project has developed a coding system which provides considerably more freedom in the interviews themselves, and retains a large amount of their original character and richness all the way through to the program input stage.

The basic method of coding is to reduce the interviews to the form of an outline, using a dictionary of some 650 code words, strings of which represent statements or ideas. These coded forms then serve as input to the computer.

The outline form of the coded interviews lends itself readily to the use of list-processing techniques. The short statement FAMILY UNABLE UNRULY (family problem: parents are incapable and children disrespectful of their elders) is interpreted as:



(the arrows indicate relations between words, list pointers in the program).

The programming system has or will have the following capabilities:

1. counting of symbol occurrences, similar to ordinary survey data processing,
2. printing of the coded interviews in outline format, showing word relations in tree structure, designed to give the analyst an idea of what to look for,
3. testing relations of words (i.e., which words are related to this one how many times),
4. searching through the coded data for a given logically connected string of words, or co-occurrence of unconnected words; this is the most powerful facility, and permits interrogating the data to any degree and in almost any way desired.

Social Science Surveys and Models - José A. Silva

During the academic year of 1965-66 Project MAC facilities were used as an aid in the development of three jobs:

1. A program for constructing scales based on survey responses. The program has two faces: a) item validation, based on the critical-ratio criterion, and b) a scale-reliability test, based on a split half correlation coefficient. Outputs include means, standard errors, standard deviations, critical ratios, and scale values for up to 40 different samples.
2. A program to construct rank-order indices. The program ranks items, and then computes tau coefficients, rho average coefficients of concordance, and tests for the significance of these values.
3. A model of a political system. This is a numerical experimentation model of the Venezuelan political system and is still being developed.
4. A program for transforming arrays of percentages in plus (+), minus (-), and zero (0) symbols according to the statistical limits of significance (difference from the mean). On the bases of these symbols' distribution, correlation ratios are then computed. This program is also still being developed.

Investment Planning and Integer Programming - David A. Kendrick

Research conducted in the summer and early fall of 1965 was directed toward obtaining solutions to some investment planning problems in the steel industry. In this research, an interactive program was designed so that a programmer could obtain the solution for a single lattice point of a mixed-integer programming problem, and use that result to select another lattice point which appeared to offer a better solution. Each lattice point corresponded to a combination of investment projects, located in time and space, which provided one strategy for increasing the capacity of a system of steel mills. Data from the flat products segment of the Brazilian steel industry was used. The interactive program was most useful, and on several occasions enabled us to find better combinations of investment projects than had been found by a branch-and-bound algorithm applied to the same problems. (See Kendrick, Appendix B.)

Research conducted in the winter and spring was directed toward the development of an efficient branch-and-bound algorithm for solving zero-one mixed-integer programming problems. The algorithm was programmed by adding a new subroutine and making suitable modifications in the RAND Corporation linear programming code MFOR. The code was loaded into the MAC system with the idea of taking advantage of the interactive effects that had proven so useful on the previous problem. Two efforts were made to debug the system (one in December-January and one in March-April) but neither effort was successful. The project was transferred to batch processing at the M. I. T. Computation Center, and development of the algorithm has since proceeded satisfactorily. (See Kendrick, Appendix C.)

Dynamic Stability of Currency Systems – Jan Grondstra and Jacques Waelbroeck

This study has involved designing and analyzing a world currency system which provides for creation of foreign exchange reserves, in amounts adequate to satisfy transaction needs of countries, and automatically adjusts exchange rates, in response to balance-of-payments surpluses and deficits so as to equalize the system.

The ESL display system was used to examine properties of non-linear model of international trade and to simulate its behaviour under different sorts of automatic exchange-rate adjustment and exogenous random perturbations.

Academic Staff

K. Biemann	L. N. Howard	A. Toomre
J. G. Charney	R. I. Mateles	

Non-Academic Research Staff

E. C. Bartels	B. A. Johanson	J. E. Spencer
E. J. Campbell	E. F. Miller	N. C. Spencer

Research Assistants and other Students

R. C. Gammill	D. D. Y. Ryu

Laboratory for Nuclear Science - Elmer C. Bartels

During the past year, the MAC time-sharing system and all its capabilities have been used extensively by professional programmers, graduate students, and professors associated with the Laboratory for Nuclear Science in the solution of their problems. Uses range from generating small programs for a single calculation, to use of a large program for solving a group of nuclear physics calculations, to use of the TIP library-search program.

Several demonstrations of TIP, the M. I. T. Libraries' literature-searching program, were given for members of the laboratory and their guests. We found that these demonstrations aroused a great deal of interest among many of those who were previously unaware of the possibilities and capabilities of the Project MAC system. We found TIP to be an excellent illustration of the ease with which a non-programmer can access and utilize a computer system. As a result of these demonstrations, more groups within the laboratory were introduced to the MAC system and began to make use of it.

Special-interest demonstrations for visitors from outside the Institute were also arranged. We have used programs involving nuclear physics calculations to illustrate the capabilities of time-sharing to a group of programmers, from a laboratory similar to ours at the University of Paris, and various other physicists from other parts of the country.

Modifications have been made to an existing program which calculates neutron radial bound-state wave functions and binding energies, and further calculates radial integrals of the wave functions. Additions included the option to calculate the harmonic oscillator radial wave function, or the continuum radial wave function for neutrons, and for use in the calculation of radial integrals. This program has been used extensively, for it is the basis for many nuclear physics calculations. Further modifications are being considered, to include calculation of differential-scattering cross-sections from partial-wave phase shifts, as well as specification of various types of nuclear-well potentials.

Another of the current projects is the adaptation of a code for a "boundary-condition model" for nucleon-nucleon interaction. The object of the program is to find a minimum set of parameters which gives a "best" fit to the available nucleon-nucleon data. Thus, the program involves minimizing a function of many variables, some of which are coupled together. When the program is run off-line, the parameters are varied sequentially in a fixed manner. However, with the aid of CTSS one can observe the interdependence of various parameters and either vary more than one parameter at a time or change the order of variation of the parameters. In this way, the search time required can be greatly shortened.

Investigation of the MAP and FORMAC programs is being carried out to determine their applicability to problems encountered by the LNS Programming Group.

Access to the MAC time-sharing system has proved to be a valuable asset to the members of the LNS staff, because it supplements and enhances the computing facility which has its own IBM-7044 for batch-processing non-conversational programs.

On-Line Meteorological Experiments - Jule G. Charney and Robert C. Gammill

A two-level, one-dimensional model of the atmosphere was programmed in MAD. The MAP system (see MAC-TR-24) was used for on-line specification of initial conditions, and graphic display on the ESL console, of any data array at any point in the iteration of the model. The extreme simplicity of the model, and the flexibility of the input-output capabilities of the MAP system, allowed the program to be used in the fashion of a "dish-pan" experiment. (Meteorologists frequently observe the activity of a fluid in a rotating pan to verify theories of fluid behavior.) Experiments could be formulated, tested, reformulated, and data collected for more careful study, all in one session with the computer.

The program was completed rather late in the school year, so it was not used in a succession of increasingly complex experiments paralleling the classroom development, as was originally hoped. Instead, most of the results were available only at the very end of the year. The facilities of the ESL Display Console produced easily understood graphical results which were photographed and shown in the classroom.

Unsteady-State Growth Rates of Micro-organisms - Richard I. Mateles, Dewey D. Y. Ryu, and T. Yasuda

So far, emphasis has been placed on data reduction and analysis of experimental results obtained from the studies of dynamic behavior of continuous culture. A new method of computing unsteady-state growth rates of micro-organisms has been developed with the aid of the Compatible Time-Sharing System.

The working equation for the unsteady-state growth rate of micro-organisms is given below in the form of a differential equation:

$$\mu = D - \frac{ds}{dt} / (s_o - s)$$

where μ is specific growth rate of micro-organism, D is dilution rate of fermentor, s_o is inlet substrate (for limiting nutrient) concentration, and s is concentration of the limiting substrate in the fermentor. The growth rate (μ) can be computed from the readings of substrate concentration (s) as a function of time (t) for a given dilution rate.

Currently, we are e[illegible] in trying to test some mathematical models which, we believe, wil[illegible]o describe dynamic response of a continuous culture under certain [illegible]onditions, such as microbial cell concentration, substrate concentra[illegible] flow rate of substrate, and other chemical and physical conditions of culture. We are also looking into the possibility of computer simulation of a system of continuous culture with the aid of both CTSS and some experimental results of dynamic behavior of continuous culture. (See Mateles, Appendix C.)

SLOAN SCHOOL OF MANAGEMENT

Priority Scheduling
On-Line Simulation
The OPS-3 System
Compilation in the OPS-3 System
Model Testing
Optimization by Iterative Search
Stockmarket Analysis
Computerized Micro-Analytic Simulation
Solution of Multi-Dimensional 0/1 Knapsack Problems
Console-Operated Statistical Routines
Gathering Social Science Data
An Extended On-Line Experiment
Information Utility Service Costs and Prices
General Diagnostic Processes
Large-Scale Interactive Simulation
Synchronization of Traffic Signals
Marketing Model Construction
Industrial Dynamics
The DYNAMO System

Academic Staff

S. S. Alexander	G. A. Gorry, Jr.	J. L. Openshaw
A. E. Amstutz	M. Greenberger	A. L. Pugh, III
P. H. Cootner	M. M. Jones	E. B. Roberts
W. M. Evan	J. D. C. Little	C. R. Sprague
D. E. Farrar	D. G. Marquis	C. V. Swanson
W. R. Fey	D. N. Ness	H. M. Weingartner
M. M. Flood	W. B. Ogilvie, Jr.	W. L. White
J. W. Forrester		

Research Assistants and other Students

J. R. Brown	T. J. R. Johnson	I. Plotkin
D. Chase	L. S. Kanodia	J. Rockart
P. Clermont	E. Kanstroom	L. L. Selwyn
L. C. Erdmann	J. N. Kogan	A. Setnick
J. Ever	B. Kwok	E. Soltero
E. P. C. Fernando	A. Leon	P. M. Wolk
M. M. Gold	J. L. Linderman	R. D. Wright
T. E. Hlavac, Jr.	L. F. McPherson, III	E. Yourdon
B. J. Huffman		

Technical Assistants

F. J. Cole	C. H. Willson

BLANK PAGE

Priority Scheduling - Martin Greenberger

The priority problem analyzed in this study is the scheduling of a time-shared computer among its concurrent users. Service requirements are not known in advance of execution. To keep response times short for small requests, service intervals are partitioned and segments are served separately in round-robin fashion, using one or more levels of priority. A mathematical analysis pinpoints the tradeoffs implicit in this procedure, and allows alternate strategies to be costed. The objectives of time-sharing are reviewed, and implications drawn for the design of future priority and pricing systems.

Future work on the problem will be directed toward a better understanding of the operating characteristics of the user: while switching in and out of context, the user behaves in a manner similar to the computer swapping programs in and out of core memory. The analogy is interesting, and may be fruitful if the user can be modeled in a meaningful way. (See Greenberger, Appendix C; MAC-TR-13, Appendix D.)

On-Line Simulation - Martin Greenberger and Malcolm M. Jones

The OPS system includes an on-line capability for building models and running simulations. Simulation activities are scheduled, cancelled, or rescheduled dynamically on an AGENDA either at a specified time or when a prescribed condition is met. Activities can be made to consume simulated time by means of a delay for a certain period or a wait until given conditions are satisfied. The AGENDA is a time-ordered list of conditionally and unconditionally scheduled activities.

Working within the multi-purpose framework of the OPS system, the user may inspect the AGENDA or some index of performance without stopping the simulation. He can also interrupt the run to make unprogrammed inspections and alterations. Before resuming, he can roll the simulation back to an earlier state that has been preserved, or perturb it in some other manner. Reference to data and activities is symbolic.

Extensive tracing facilities permit the user to follow the flow of control during a simulation to any desired level of detail. He may modify his experimental design as he views partial results, as well as conduct interim statistical analyses without relinquishing title to the computer or losing his place in the simulation. By running independent components of his model singly or in selected combinations from standard initial conditions, he is able to examine different aspects of his simulation in a controlled way. This flexible mode of operation encourages him to build and validate his model incrementally, thus giving him a measure of protection against the problem of initial over-complexity that can plague a monolithic simulation. (See Greenberger, Appendix C.)

The OPS-3 System - Martin Greenberger, Malcolm M. Jones, James H. Morris, Jr., and David N. Ness

The manual of the OPS system, On-Line Computation and Simulation: The OPS-3 System, was published in the Fall of 1965, and fully describes the system. (See Greenberger, Appendix C.) Work on the system has continued throughout the course of the year, and the system formed the basis of course 15.599 (Management Implications of Interactive Computer Systems) given by Jones and Ness. Professor Greenberger has given several lectures on the system while on leave at Berkeley, and he presented an invited paper "Incremental Simulation" at the national meeting of ORSA on 19 May 1966.

The OPS group is in the process of developing the specifications for a version of the system which can operate in a MULTICS environment.

Compilation in the OPS-3 System - James H. Morris, Jr.

Modifications and additions were made to the OPS system to allow compilation of otherwise interpreted procedures. The procedures (called KOP's) are translated into MAD subroutines. Aside from its use in speeding up OPS runs (by a factor of 25 to 200) the translater may be used to generate programs that can be used independently of the OPS system.

Model Testing - Merrill M. Flood

A MADTRN program, called CSTØ1, has been developed and used to make comparative tests of various stochastic learning theories against data for recent learning experiments with rats and human subjects. Using iterative search techniques, the program calculates maximum likelihood estimates for the parameters appearing in each stochastic model, and also yields confidence-region estimates based on likelihood-ratio distributions.

The program provides a highly interactive on-line capability for altering the iterative process during each calculation, including a variety of convenient options for controlling intermediate console output. Extensive experience with CSTØ1 shows a considerable saving of computer processing time, compared to similar calculations done by batch processing. Other benefits of on-line techniques include considerable flexibility in the extent of tests made for validity of alternative models.

The CSTØ1 program and its use have been detailed in a recent paper which also references earlier papers on theoretical and computational aspects of the work. (See Flood, Appendix C.)

A MADTRN program, called CSYMM5, provides an on-line capability to synthesize experimental data similar to that produced by a subject who behaves exactly in accordance with any one of several stochastic learning models. This synthetic data can then be statistically compared -- in various ways -- with actual experimental data, as a means of constructing and testing alternative learning models.

We have made considerable use of tape strategy, to ease the handling and analysis of a considerable volume of experimental data, since we have data from some 200 separate experiments filed on magnetic tape. Any one of the 200 files is readily moved from tape to disk for on-line analysis within a few seconds -- a capability that may be very easily extended beyond our 200 files.

This easy on-line access to data from many experiments has enabled us to construct and test alternative theories against data from experiments selected on-line for the purpose.

Optimization by Iterative Search - Alberto Leon

Six non-algorithmic iterative search procedures (previously developed for batch-processing systems) have been converted for CTSS on-line use. Three of the methods adopted -- BU1MAC, BU2MAC, and LOOMAC -- are classified as local direct-search techniques. The other three - SHIMAC, SATMAC, and RANMAC -- include variations of purely random search. Another, but very complex code, GROPE, a universal adaptive code for optimization, is still in the conversion process.

Iterative non-algorithmic optimization techniques make use of several program parameters which must be intuitively guessed by the user. Efficient operation of the computer codes is a direct function of the correct selection of these parameters; furthermore, if the problem on hand is not familiar to the user this selection is not an easy matter.

The interactive mode included in our codes permits the user to modify program parameters at convenient stages of the calculation process. The previous feature decreases the effect of an incorrect selection of initial values for the parameters; such an initial incorrect selection is critical in batch-processing systems. The interactive mode accelerates convergence to an answer by convenient use of the investigator's judgment and knowledge of the problem. The on-line capability affords the user an easy way to test different formulations of the same problem.

The solutions of several complex problems with different formulations, using our six computer codes, indicates substantial savings in central processing versions of the codes. Considerable savings in total elapsed time (both user and machine time) are also obtained when several formulations of the problem are tried.

The methods and computer programs are applicable to a wide variety of problems in many fields of science, engineering, and management. We have cooperated in the successful application of these techniques to applied problems of interest to faculty and graduate students in several fields, at M. I. T. and elsewhere.

Stockmarket Analysis - Donald E. Farrar and Richard Wright

Professor Farrar has been investigating patterns of corporate financial behavior with John R. Meyer and Robert Glauber. Multivariate statistical techniques, primarily factor analysis and canonical correlation, are being used to stratify firms into homogeneous groups, and to examine patterns of financial behavior within groups obtained. Initial computations have been completed and analyzed: manuscript is scheduled for completion within one year, with publication in book form anticipated.

Professor Farrar is also researching the seasonal behavior of common stock prices with Kendrick Melrose. Alleged seasonal regularities in common stock price movements with specified industry groups is being examined with spectural techniques. Computations performed during the summer of 1965 are being repeated on more appropriate data during the summer of 1966. An article length manuscript will be produced within a year.

Project MAC has been used as a utility by Richard Wright to aid software development for the Compustat data system. [Compustat is a financial data service of the Standard and Poor's Company.] Basic programs have been debugged for the general system, and specific programs have been checked out for users in the Sloan School and the Department of Economics.

Inability to multiprogram the data tapes on the CTSS has been a serious setback. At the present time, segments of the tape are stored on disk and used for check-out programs. Efficient integration of these tapes with CTSS would greatly facilitate student and faculty use of this exceptional body of financial information.

Computerized Micro-Analytic Simulation - Arnold E. Amstutz

The initial proposal for this project noted two implications of direct management access to a simulated environment which would be investigated. The first related to the contribution of the interaction capability to the development of simulation models; the second was the impact of direct access to model-based information on decision making in artificial problem-solving situations.

Investigations directed toward the first area of research are now in progress. An interactive executive program, controlling the development and modification of sub-system models and total system simulation under historical and assumed future conditions, has been developed. The first full-scale use of this system is scheduled for this summer when it will serve as the basis for extensive laboratory sessions in a special summer session on computer simulation of competitive market response. (See Amstutz, Appendix C.)

Solution of Multi-Dimensional 0/1 Knapsack Problems - David N. Ness and H. Martin Weingartner

The purpose of this project was to create a program which would allow us to solve multi-dimensional 0/1 Knapsack problems of a substantially greater magnitude than had previously been reported. To accomplish this task, some new algorithms were developed. They were implemented in such a way as to allow a user of the program to determine the course of the solution.

Briefly stated, results indicate that problems of 28 variables, which would have required hours of computer time by previous methods, can be solved in a few machine seconds, and much larger problems (involving 100 or so variables) can be solved in a few machine minutes. (See Ness, Appendix C.)

Console-Operated Statistical Routines - James R. Miller and James H. Morris, Jr.

The library of statistical routines, which has been undergoing evolutionary development since 1964, has been both expanded and improved. In addition, most of the library has been incorporated within the OPS system.

Additions to the library during the past year include:

1. A two-way analysis of variance,
2. Polynomial regression analyses,
3. A least-squares routine to fit exponential and logistic functions to data,
4. A generalized routine to convert the results of factor analysis into a parametric predictive equation.

Improvements have been made to several existing routines. Substantial improvements have been made to:

1. The T-test of the difference between two sample means,
2. The simple and multiple regression analyses.

Most of the statistical routines (both old and new) have been incorporated by Stephen Whitelaw and James Morris into the OPS system. In addition, it is planned to increase the scope of these routines to include hypothesis formulation and data preparation.

Gathering Social Science Data - William M. Evan and James R. Miller

A laboratory experiment was performed to determine the bias-reducing effect of administering social science questionnaires by means of an on-line teletype console. It was assumed that experimental subjects would regard typing responses directly into a computer as a situation guaranteeing greater privacy, and a greater sense of anonymity and confidentiality, than the conventional situation wherein questionnaires or psychological tests are filled out with pencil and paper and are then scrutinized individually by other human beings. Major conclusions of the experiment were as follows:

1. Whenever the content of a question is regarded by subjects as highly personal and possibly disturbing, answers typed directly into a computer will reflect greater honesty and candor than if the same question were written out in a conventional, pencil-and-paper manner.
2. No such difference occurs in the case of emotionally neutral questions. (See Evan, Appendix C.)

<u>An Extended On-Line Experiment</u> - James R. Miller

An eleven-week laboratory experiment in decision making was conducted using an on-line computer as a controller. The computer proved to be extremely beneficial because:

1. It permitted immediate rescheduling of the experimental procedure in the face of unanticipated and uncontrollable events;
2. It provided immediate analysis of results while the experiment was still in progress, thus permitting the experimenter to capitalize on developing response trends;
3. It diagnosed misunderstanding and confusion on the part of experimental subjects and guided immediate remedial action;
4. It provided immediate feedback to the subjects, thus substantially increasing their motivation to participate. (See Miller, Appendix B.)

<u>Information Utility Service Costs and Prices</u> - Lee L. Selwyn

The next large information utility, such as the MULTICS system being developed at Project MAC, will present its users with a wide choice of services and combinations of services, which we call applications. In order to operate efficiently and provide service on a competitive basis with conventional computing operations, such a "utility" must establish and maintain a specific and consistent pricing structure. That is, the utility must necessarily share its costs among its users, according to their respective use, as determined by some formula. However, the problem is complicated by the complexity of possible applications, and the difficulties that may arise in trying to determine their costs. In addition, prices to users need not be directly related to costs, but rather may reflect the relative demand on the utility for various types, grades, and times of service.

It is the goal of this project to develop a set of guidelines and ground-rules for the establishment of price schedules. Included in this will be a methodology for cost determination, for demand analysis, and for establishing goals of system utilization, based upon the pricing structure.

During the past few months, some preparation for this study has been made. A demand-analysis experiment using the Sloan School of Management users group on the MAC system has been proposed, and work has begun on the necessary software to implement this experiment. Under this pilot study, users will be able to "buy" their MAC resource allocation with "points" given them periodically. The goal is to develop an equilibrium set of prices that will maximize usage of the MAC 7094 computer for all shifts during the week.

General Diagnostic Processes - G. Anthony Gorry

In many areas the performance of some system is of interest, and the failure of that system to perform to expectation is a matter of concern. The problem-solving process involved in determining the causes of system malfunction in these areas is a diagnostic process. Initial study indicates that there are elements of the diagnostic process which are independent of the particular diagnostic application.

To test this hypothesis, a computer program is being written to perform general diagnosis. This program will embody those elements of the diagnostic process which can be generalized to more than one area. It will accept parametric data to specialize its activities to a given area. Hopefully, the program will be able to perform diagnostic functions in more than one problem area.

An initial version of the program has been written on-line in the SLIP-MAD system of Project MAC and is currently being debugged. When completed, the program will be used in further study of the diagnostic process.

Large-Scale Interactive Simulation - Christopher R. Sprague

I developed a long-range planning game for Professor D. C. Carroll, which was presented at the Fall 1965 meeting of the Industrial Research Institute. The game consisted of information about long-range returns from research, plus an interactive simulation program on the MAC system. The information was reduced to parameters for a simple pay-off model and was presented as printed background material. Participants read the printed material and budgeted their "research expenditures" accordingly. They then used the MAC system as a tool to improve the budgets for optimum pay-off over a simulated ten-year period. This provided both an exercise in budgeting and a demonstration of the power of an interactive time-sharing system for problem-solving. This work was done in collaboration with Mr. A. H. Bobis of the American Cyanamid Company. *

During the latter part of the year, I worked on a variety of time-shared simulation techniques. While it is generally felt that time-sharing is a poor medium for massive simulation, there are three areas of simulation where time-sharing is outstanding:

1. Conversational input and checking of parameters;
2. On-line modification of model structure (as opposed to parameters)
3. Allowing a model to call for human aid when it is faced with a particularly difficult decision.

* An article, "An Exercise in Long-Range Planning", will be published in the November 1966 issue of Research Management.

I did considerable investigation of the first two points, using several private commands which facilitated them. These were used to some advantage for a 1966 summer course, "Computer Simulation of Consumer and Competitor Response", with Professors A. E. Amstutz and P. O. Soelberg. I also spent some time investigating the third point in the context of a very large network-based planning model with heuristic "optimization" features, with Professor Carroll: individuals within the National Aeronautics and Space Administration have indicated tentative enthusiasm for such an interactive information system.

Synchronization of Traffic Signals - John D. C. Little

The problem of synchronizing traffic signals to obtain maximal bandwidth has been formulated as a mixed-integer linear program both for the case of the long arterial street and for a street network. A branch-and-bound algorithm has been developed for solving these problems. The algorithm has been developed for solving these problems. The algorithm works by solving a sequence of linear programs, which may be thought of as corresponding to the nodes of a tree. The exploration of the tree is performed interactively using CTSS, the results of one linear program being evaluated by the user before deciding what linear program to solve next. (See Little, Appendix C.)

Marketing Model Construction - John D. C. Little and Theodore E. Hlavac, Jr.

A geographic, competitive model of an urban automobile market has been constructed, fit to data, and programmed for interactive use on time-sharing. The attractiveness of a dealer to a prospective new car buyer is taken to depend on the distance from buyer to dealer, the make preference of the buyer, and on parameters characterizing the dealer. A model of dealer sales by geographic area has been built and fit to three months of new car registration data for Chicago. After fitting, the model in its interactive form can be used on-line to investigate a variety of site selection and inter-brand competition questions arising from changes in market structure. (See Hlavac, Appendix B.)

Industrial Dynamics - Jay W. Forrester

The Industrial Dynamics group has been revising DYNAMO to improve its usefulness and to make it available for the new computers at M. I. T. and elsewhere.

Explorations have continued in how best to use the interaction between a person and time-sharing computer for the study of nonlinear feedback systems. Modes of system behavior are discovered experimentally which do not appear in the available solutions for linear systems. By alternating simulation and evaluation, limited generalizations can be discovered. System simulations presented as time plots can then be generated to illustrate and summarize the kinds of dynamic behavior which have been observed. The modes of behavior in nonlinear, multiloop, feedback systems containing both positive and negative feedback are able to match the interactions of practical importance in social systems. The work should help to bridge between the descriptive and the quantitative literature. It is beginning to develop a systems structure for organizing our knowledge of economic and managerial systems.

From the viewpoint of developing a time-sharing system, the objective is to identify how the computer can facilitate learning by the user. Use of time-sharing answers questions about the research area (feedback systems) at the same time that it answers questions about how time-sharing can be improved to become a better aid in exploring unstructured learning tasks.

The DYNAMO System - Alexander L. Pugh, III

DYNAMO is being rewritten primarily to make it available as a part of MULTICS, but also to make a number of improvements that were not practical except during a major rewrite.

This past year, a true algebraic translator has been developed for the new DYNAMO. While it has been patterned on the method of Samelson and Bauer, it has been carefully developed to detect errors in a manner which gives clear error explanations.

A unique feature of this translator is the detection of ambiguous denominators. The expression

$$X = \frac{A}{B+C} \qquad (1)$$

can mistakenly be written

$$X = A/B+C \qquad (2)$$

The standard meaning of (2) is

$$X = \frac{A}{B} + C \qquad (3)$$

but the user can easily reread (2) dozens of times without realizing the error. The new translator detects this ambiguous form and retypes it with parentheses added.

$$X = (A/B) + C \qquad (4)$$

Another development just completed is a method of translating an equation (already parsed) into machine language which is highly machine independent. It is expected that simply with a change of several tables the program will generate output suitable for the IBM 7094, GE 645, or IBM 360 computers.

APPENDIX A

PROJECT MAC MEMORANDA

MEMORANDUM MAC-M-No.	SUBJECT	AUTHOR	DATE
258	Design of the Hand (AIP memo 86)	M. L. Minsky	8/13/65
259	AED Flash 19: RWORD Package (9442-M-141)	R. J. Bigelow	8/12/65
260	AEDBUG Program Description (9442-M-142)	T. B. Fox	8/16/65
261	Modifications and Addenda to the AED-O Programmer's Guide (9442-M-143)	C. Feldmann, C. Bower, P. Ladd	8/18/65
261-2	Additional Modifications and Addenda to the AED-O Programmer's Guide (9442-M-143-2) [Does not replace MAC-M-261]	C. Feldmann	9/7/65
262	A Theory of Computer Instructions (AIP memo 89)	W. D. Maurer	9/65
263	FLIP - A Format LIst Processor (AIP memo 87)	W. Teitelman	9/16/65 (not dated)
264	EDIT and BREAK Functions for LISP (AIP memo 88)	W. Teitelman	9/20/65 (not dated)
265-1	Changes to Teletype Terminals (CC-265)	R. G. Mills	11/23/65
266	Relations Between the Project MAC System and a User	A. I. Dumey	9/13/65
267	MACTAP, a PDP-6 DECtape Handling Package (AIP memo 87-A)	P. Samson	9/16/65
268	Linking Loader for MIDAS	P. Samson	1/31/66
269	Number Conversion Subroutine CNVERT (944-M-157)	J. V. Oldfield	11/19/65
270	MADIO, a Simplified Input-Output Package for MAD Programs	P. J. Denning	9/65
271	AED Flash 20: New Array Handling Language for AED-O (9442-M-145)	P. Ladd	9/24/65
272	AED Flash 21: Octal Debugging Subroutine LOPAT (9442-M-146)	J. Walsh	9/24/65
273	The ODA String Package (9442-M-150)	D. T. Ross	10/1/65

MEMORANDUM MAC-M-No.	SUBJECT	AUTHOR	DATE
274	AED Flash 22: Use of Remote Display Consoles (9442-M-148)	J. Rodriguez	10/7/65
275	AED Flash 23: Use of GENCAL (9442-M-152)	J. Walsh	10/14/65
276	Direct-View Storage-Tube Displays	R. H. Stotz, J. E. Ward	3/8/66
277	OPS-3 Goes Public (replaces MAC-M-233)	M. Greenberger, M. M. Jones, D. N. Ness, J. H. Morris	10/25/65
278	AED Bibliography (9442-M-153)	D. T. Ross	10/20/65
279	MIDAS (PDP-6 assembly program) (AIP No. 90)	P. Samson	10/65
280	Operating Statistics of the MAC Time-Sharing System	T. Hastings	12/8/65
281	Computer-Assisted Instruction (CAI)	R. Rosenberg	10/20/65
282	Computer Experiments in Finite Algebra - II	W. D. Maurer	12/1/65
283	AED Flash 26: BSS Plex Dump (9442-M-156)	R. O. Ladson	11/2/65
284	Usage Conventions for EPS SAVED	C. C. Tillman	11/30/65
285	A Useful Algebraic Property of Robinson's Unification Algorithm (AIP No. 91)	T. P. Hart	11/65
286	AED Flash 25: The Loader/Unloader (9442-M-155)	B. L. Wolman	11/16/65
287	A Display Buffer Computer for the ESL Console	J. Grondstra, R. H. Stotz, J. E. Ward	11/23/65
288	An Experiment Using a New System of Documents for Communication in the Computer Science Field	R. M. Fano	12/28/65
289	Operation Notes for the DAEMON (CC-258)	G. Clancy	12/30/65
290	Complete CTSS Inventory (CC-259) [Limited Distribution]	D. R. Widrig, G. Schroeder, M. M. Child	1/3/66

MEMORANDUM MAC-M-No.	SUBJECT	AUTHOR	DATE
291	AED Flash 27: New CTEST2 Command (9442-M-158)	C. Feldmann	1/7/66
292	Some Remarks on the Problem of Exponential Growth in Machine Theorem Proving	W. D. Maurer	1/7/66
293	Availability of the MAP System (CC-260)	J. W. Brackett	2/3/66
294	Topics in Model Theory (AIP No. 92)	M. I. Levin	1/20/66
295	The Value of Compatibility Between Time-Sharing and Batch Processing	W. D. Maurer	1/20/66
296	A New Version of CTSS LISP (AIP No. 93)	R. Fenichel, J. Moses	2/2/66
297	The Appropriateness of a Computer Programming Department As Opposed to a Computer Sciences Department	W. D. Maurer	1/20/66
298	Analysis of a Protected Service Routine (MSG Memo 21)	P. J. Denning	2/21/66
299	A New Machine Learning Technique Applied to Checkers (AIP No. 94)	A. D. Griffith	3/9/66
300	Graphical Input-Output Techniques in Power System Computation (9442-M-159)	J. V. Oldfield	2/28/66
301	Analysis and Transformation of Computational Processes (9442-M-162; MSG Memo 22)	J. Rodriguez	3/7/66
302	Notes on Searching and Sorting	J. W. Poduska	3/1/66
303	MLPL: A MAD-compiled List Processing Language	J. W. Poduska	2/10/66
304	Operators for Manipulating Language Structures (9442-M-160)	B. L. Wolman	3/18/66
305	A Program Feature for CONVERT (AIP No. 95)	A. Guzmán, H. McIntosh	4/66
306	Operating Procedures for CTSS (CC-261)	L. P. Odland	4/25/66
307	CTSS SNOBOL User's Manual (CC-235-4)	D. Shea	5/16/66

MEMORANDUM MAC-M-No.	SUBJECT	AUTHOR	DATE
308	POLYBRICK: Adventures in the Domain of Parallelepipeds (AIP No. 96)	A. Guzmán	5/66
309	AED Flash 28: Reduction of Load Time (9442-M-168)	B. L. Wolman	5/31/66
310	Symbolic Integration (AIP No. 97)	J. Moses	6/10/66
311	AED Flash 29: Description of "LAED Command" (9442-M-169)	B. L. Wolman	6/10/66
312	General Description of the AED-1 Processor and the Display Interface System (9442-M-170)	D. T. Ross	6/16/66
313	PDP-6 LISP (AIP No. 98)	P. Samson	6/16/66
315	AED Flash 30: Writing "Machine-Independent" AED-O Programs (9442-M-171)	J. Rodriguez	6/21/66
316	CONVERT (AIP No. 99)	A. Guzmán H. McIntosh	6/66

. . . REVISIONS . . .

192-2	Unrecognizable Sets of Numbers (AIP Memo 73 revised)	M. L. Minsky S. A. Papert	9/15/65
224-1	Subroutines for Automatic Plotting	T. Stockham J. W. Brackett	2/23/66
231-2	Description of SNOBOL Commands (CC-235-3)	L. Pouzin	10/1/65
265-1	Changes to Teletype Terminals	R. G. Mills	11/23/65
278-1	AED Bibliography (9442-M-153-1)	D. T. Ross	3/21/66
289-1	Operations Notes for the DAEMON (CC 258-1)	G. Clancy	5/31/66

APPENDIX B

M.I.T. THESES

Alanko, J.M., A Computer Aid to General Arrangement in Ship Design, Department of Naval Architecture and Marine Engineering, S.M. Thesis, June 1966

Bailey, W.A., Stability Analysis by Limiting Equilibrium, Department of Civil Engineering, C.E. Thesis, February 1966

Burnett, G.J., A Design Language for Digital Systems, Department of Electrical Engineering, S.M. Thesis, September 1965

Bushkin, A.A., On Generalized Syntax-directed Translation Systems and Their Implementation, Department of Mathematics, S.B. Thesis, August 1965

Chang, G.D.Y., A Table-Driven Compiler Generator System, Department of Electrical Engineering, S.M. Thesis, June 1966

Chase, D.L., The Effect of Paging on Computer Executive Time, Department of Electrical Engineering, S.B. Thesis, June 1966

Cheek, T.B., Design of a Low-Cost Character Generator for Remote Computer Displays, Department of Electrical Engineering, S.M. Thesis, February 1966 (See also MAC-TR-26, Appendix D.)

Clermont, P., Machine Substitution in a Job Shop, Sloan School of Management, S.M. Thesis, January 1966

Crystal, T.H., A Model of Laryngeal Activity During Phonation, Department of Electrical Engineering, Sc.D. Thesis, May 1966

Daniels, R.L., A Computerized Approach to Construction Cost Estimating and Accounting, Department of Civil Engineering, M.S. Thesis, February 1966

Denning, P.J., Queueing Models for File Memory Operation, Department of Electrical Engineering, S.M. Thesis, May 1965 (See also MAC-TR-21, Appendix D.)

Edwards, D.J., OCAS: On-Line Cryptanalytic Aid System, Department of Electrical Engineering, S.M. Thesis, January 1966 (See also MAC-TR-27, Appendix D.)

Evan, D.S., Man-Machine Communication for Simulation of Nonlinear Circuits, Department of Electrical Engineering, S.M. Thesis, February 1966

Everest, G.C., Data File Organization Within a Dynamic Computer System, Sloan School of Management, S.M. Thesis, September 1965

Fattal, L., Dynamic Programming and Markov Processes Applied to an Orange Grove Investment Decision, Sloan School of Management, S.M. Thesis, September 1965

Fluhr, Z.C., Single Threshold Element Realizability by Minimization, Department of Electrical Engineering, S.M. Thesis, August 1965

Free, J.C., Dynamic System Identification Using Adaptive Modelling and Manipulated Research Environment, School of Humanities and Social Science, Ph.D. Thesis, January 1966

Gertz, J.L., Graphical Input-Output Program for Digital System Simulation, Department of Electrical Engineering, S.M. Thesis, June 1966

Graham, H.L., A Hybrid Graphical Display Technique, Department of Electrical Engineering, S.M. Thesis, May 1966

Hlavac, T.E., Jr., A Competitive Model of the Chicago Automobile Market and Its Realization as an Interactive Computer System, Sloan School of Management, S.M. Thesis, December 1965

Ivie, E.L., Search Procedures Based on Measures of Relatedness Between Documents, Department of Electrical Engineering, Ph.D. Thesis, June 1966 (See also MAC-TR-29, Appendix D.)

Kendrick, D.A., Programming Investment in the Steel Industry, Department of Economics, Ph.D. Thesis, June 1965

Kogan, J.N., Real-Time Sequencing Versus Periodic Scheduling in a Job Shop, Sloan School of Management, S.M. Thesis, June 1966

Ledgard, H.F., Manipulation of Approximating Functions on a Graphical Display Facility, Department of Electrical Engineering, S.M. Thesis, September 1965

Lee, F.F., A Study of Grapheme-to-Phoneme Translation of English, Department of Electrical Engineering, Ph.D. Thesis, November 1965

Ling, G.C., Investigation of a Semiconductor Laser Data Link for Remote Computer Displays, Department of Electrical Engineering, M.S. Thesis, February 1966

Magnuski, H.S., Design for an Information Retrieval and Display Terminal, Department of Electrical Engineering, S.M. Thesis, June 1966

Marks, R.E., A Table Driven Syntactic Analyzer, Department of Electrical Engineering, M.S. Thesis, June 1966

Miller, J.R., The Assessment of Worth: A Systematic Procedure and its Experimental Validation, Sloan School of Management, Ph.D. Thesis, June 1966

Miller, R.L., A Model for Traffic Flows on a Two-Lane Two-Way Rural Highway, Department of Mathematics, Ph.D. Thesis, June 1966

Mooallem, S.T., A Teaching Script for ELIZA, Natural Language Communication Program, Department of Electrical Engineering, S.B. Thesis, June 1966

Morris, J.H., Jr., Interpretive Systems in On-Line Programming, Sloan School of Management, S.M. Thesis, January 1966

Niessen, C.W., An Experimental Facility for Sequential Decoding, Department of Electrical Engineering, Ph.D. Thesis, September 1965

Olshansky, K.J., A Low-Cost Teletype-Operated Graphical Display, Department of Electrical Engineering, S.M. Thesis, August 1965

Parmelee, R.P., Three-Dimensional Stress Analysis for Computer-Aided Design, Department of Mechanical Engineering, Ph.D. Thesis, June 1966

Piotrowski, G., Preliminary Design of a Graphical Input-Output Device Incorporating Force Output, Department of Mechanical Engineering, S.M. Thesis, September 1965

Pitidis-Poutous, T.M., The Mathematical Derivation of Preliminary Ship Lines, Department of Naval Architecture and Marine Engineering, S.M. Thesis, February 1965

Rotenberg, L.J., Formal Proofs Concerning Partial Recursive Functions, Department of Electrical Engineering, S.M. Thesis, June 1966

Saltzer, J.H., Traffic Control in a Multiplexed Computer System, Department of Electrical Engineering, Ph.D. Thesis, June 1966

Skinner, T.P., Flow-Debug: A Graphical Debugging Aid for On-Line Computer Operations, Department of Electrical Engineering, S.B. Thesis, June 1966

Smith, A.A., Input-Output in Time-Shared Segmented, Multiprocessing Systems, Department of Electrical Engineering, S.M. Thesis, January 1966 (See also MAC-TR-28, Appendix D.)

Speck, C.E., Quasi-Static Theory of Plasma Instabilities at Cyclotron Harmonics, Department of Electrical Engineering, S.M. Thesis, August 1965

Stratton, W.D., Investigation of an Analog Technique to Decrease Pen-Tracking Time in Computer Displays, Department of Electrical Engineering, S.M. Thesis, January 1966 (See also MAC-TR-25, Appendix D.)

Teitelman, W., PILOT: A Step Toward Man-Computer Symbiosis, Department of Mathematics, Ph.D. Thesis, June 1966

Therrien, C.W., A Digital Computer Simulation for Electrical Networks, Department of Electrical Engineering, S.M. Thesis, June 1965

Wantman, M.E., CALCULAID: An On-Line System for Algebraic Computation and Analysis, Sloan School of Management, S.M. Thesis, September 1965 (See also MAC-TR-20, Appendix D.)

Wilde, D.U., Program Analysis by Digital Computer, Department of Electrical Engineering, Ph.D. Thesis, June 1966

Zucker, J.S., Particle Trajectories in Magnetic Fields, Department of Electrical Engineering, S.M. Thesis, January 1966

APPENDIX C

EXTERNAL PUBLICATION

Amstutz, A. E., Simulation Techniques in the Analysis of Marketing Strategy, Sloan School of Management Working Paper 208-66, July 1966

Asselin, T. H., CPM Bar Chart: A Graphic CPM Schedule Generated by Computer, Department of Civil Engineering, Research Report R66-4A, M. I. T., Cambridge, Mass.

Bartsch, R. R., "Beam-Plasma Discharge: System D", M. I. T. R. L. E. Quarterly Progress Report No. 81, April 1966, pp. 69-72

Bernard, G. D., and A. Bers, "Theory of Plasma Excitation by a Line Charge Source", M. I. T. R. L. E. Quarterly Progress Report No. 81, April 1966, pp. 102-104

Bers, A., and C. E. Speck, "Instabilities in the Extraordinary Waves across the Magnetic Field", M. I. T. R. L. E. Quarterly Progress Report No. 81, April 1966, pp. 133-137

Bers, A., and S. R. J. Brueck, "High-Frequency Electron-Phonon Interactions in a Magnetic Field", M. I. T. R. L. E. Quarterly Progress Report No. 81, April 1966, pp. 106-111

Biggs, J. M., "On-Line Structural Design," Conference Record, 1966 Winter Convention on Aerospace and Electronic Systems, February 1966

Bobrow, Daniel G., et al, The BBN-LISP System, Bolt Beranek and Newman Inc. Scientific Report No. 1, Project 8668, February 1966

Brown, S. C., "A Bibiographic Search by Computer", Physics Today, vol. 9, no. 5, May 1966

Brown, S. C., "Computer-Programmed Basic Data of Plasma Physics", M. I. T. R. L. E. Quarterly Progress Report No. 81, January 1966

Clarkson, G., and F. D. Tuggle, "A Theory of Group Decision Behavior", Behavioral Science, vol. 11, no. 1, January 1966

Coons, S. A., "Computer Graphics and Innovative Engineering Design", Datamation, vol. 12, no. 5, May 1966

Crisman, P. A., ed., The Compatible Time-Sharing System: A Programmer's Guide, second edition, The M. I. T. Press, Cambridge, Mass., August 1965 (Library of Congress No. 65-25206)

Davis, J. A., "Discrete Particle Models of the Beam-Plasma Discharge", M. I. T. R. L. E. Quarterly Progress Report No. 80, January 1966, pp. 125-127

Davis, J. A., "Wave-Mirror Heating", M. I. T. R. L. E. Quarterly Progress Report No. 81, April 1966

Davis, J. A., "Computer Simulation of the Beam-Plasma Discharge", M. I. T. R. L. E. Quarterly Progress Report No. 82, July 1966, p. 146

Evan, W. M., and J. R. Miller, Differential Effects on Response Bias of Computer vs. Conventional Administration Of a Social Science Questionnaire: A Laboratory Experiment, Sloan School of Management Working Paper 206-66, October 1965

Falconer, D., and C. W. Niessen., "Simulation of Sequential Decoding for a Telemetry Channel", M. I. T. R. L. E. Quarterly Progress Report No. 80, January 15, 1966, pp. 183-193

Feldman, R. C., A Heuristic Approach to Using Linear Time-Cost Curves in Scheduling Project Networks, Department of Civil Engineering, Research Report R66-19, M. I. T., Cambridge, Mass., June 1966

Fife, D. W., "An Optimization Model for Time-Sharing", Proceedings of the Spring Joint Computer Conference 1966, Spartan Books, Inc., Washington D. C., 1966

Flood, M. M., Commercial Information Processing Networks -- Prospects and Problems in Perspective, (Report for the National Commission on Technology, Automation, and Economic Progress) U. S. Government Printing Office, Washington, D. C., 1966

Flood, M. M., An On-Line Version of STØ1 -- A FORTRAN II Program for Parameter Estimation and Likelihood Tests of Stochastic Learning Theories, University of Michigan Mental Health Research Institute, Preprint 173, February 1966

Frey, W., The Mass Media and Rural Development in Turkey, Center for International Studies, C/66-2, M. I. T., January 1966

Frey, W., Rural Turkey, A Regional Portrait, Center for International Studies, M. I. T., February 1966

Greenberger, M., "Simulation and the House-Heating Problem", Behavioral Science, vol. 11, no. 1, March 1966

Greenberger, M., M. M. Jones, J. H. Morris Jr., and D. N. Ness, On-Line Computation and Simulation: The OPS-3 System, The M. I. T. Press, Cambridge, Mass., August 1965

Guzman, A., and H. V. McIntosh, "CONVERT", ACM Symposium on Symbolic and Algebraic Manipulation, Washington D. C., March 1966; also published in Communications of the ACM, vol. 9, no. 8, August 1966, pp. 604-615

Haring, D. R., "Multi-Threshold Threshold Elements", IEEE Transactions on Electronic Computers, vol. EC-15, no. 1, February 1966

Haring, D. R., Sequential-Circuit Synthesis: State Assignment Aspects, M.I.T. Press, Research Monograph No. 31, 1966

Haring, D. R., "The Beam Pen: A Novel High-Speed, Input/Output Device", Proceedings of the 1965 Fall Joint Computer Conference, Spartan Books, Inc., Washington, D. C., 1965, pp. 827-835

Haus, H. A., "High Order Trapped Light-Beam Solutions", M.I.T. R.L.E. Quarterly Progress Report No. 80, January 1966, pp. 79-82

Hennie, F. C., "On-Line Turing-Machine Computation", IEEE Transactions on Electronic Computers, vol. EC-15, no. 1, February 1966

Hlavac, T. E., Jr., and D. C. Little, A Geographic Model of an Urban Automobile Market, Sloan School of Management Working Paper 180-66, April 1966

Hunt, E. B., J. Marin, and P. J. Stone, Experiments in Induction, Academic Press, 1966

Kain, R. Y., L. Clapp, and D. E. Jordan, "Symbolic Factoring of Polynomials in Several Variables" ACM Symposium on Symbolic and Algebraic Manipulation, Washington D. C., March 1966; also published in Communications of the ACM, vol. 9, no. 8, August 1966, pp. 604-615

Lieberman, M. A., and A. Bers, "Electron Beam Excitation of Ion Oscillations in an ECRD Plasma", M.I.T. R.L.E. Quarterly Progress Report No. 81, April 1966, pp. 75-80

Lieberman, M. A., and A. Bers, "Instabilities in Hot-Electron Beam-Plasma Systems", M.I.T. R.L.E. Quarterly Progress Report No. 81, April 1966, pp. 81-85

Lieberman, M. A., and A. Bers, "Theory of VHF Oscillations and Possible Interactions with Ions in the Beam-Plasma Discharge", M. I. T. R. L. E. Quarterly Progress Report No. 81, April 1966, pp. 85-93

Lieberman, M. A., and A. Bers, "Quasi-Linear Theory of Narrow-Bandwidth Convection Instabilities", M. I. T. R. L. E. Quarterly Progress Report No. 81, April 1966, pp. 94-100

Little, J. D. C., "The Synchronization of Traffic Signals by Mixed-Integer Linear Programming", Operations Research, vol. 14, July 1966

Logcher, R. D., Dynamic Memory Allocation for Engineering Data, M. I. T. Department of Civil Engineering Report T66-1, January 1966

Logcher, R. D., H. C. Reggini, A. J. Ferrante, and R. V. Goodman, "Preliminary Structural Analysis by Computer", Journal of the Structural Division, ASCE, February 1966

Logcher, R. D., G. M. Sturman, and J. M. Biggs, "Structural Design Language - A Computer System for the Total Structural Design Process", Proceedings of the International Symposium on the Use of Computers, June 1966

Magnuski, H. S., "Computer Aids to the Handicapped: The PDP-8 as a Braille Translator", Decuscope, vol. 5, no. 4, 1966

Mateles, R. I., D. Y. Ryu, and T. Yasuda, "Measurement of Unsteady State Growth Rates of Micro-organisms", Nature, vol. 208, 1965, pp. 263-265

Mathews, W. D., "The TIP Retrieval System at M. I. T.", Decuscope, vol. 5, no. 4, 1966

Maurer, D. W., "An Algebraic Structure Algorithm", Mathematical Algorithms, Preliminary Issue, January 1966

Maurer, D. W., "Computer Experiments in Finite Algebra", ACM Symposium on Symbolic and Algebraic Manipulation, Washington D. C., March 1966; also published in Communications of the ACM, vol. 9, no. 8, August 1966

Mills, R. G., "Man-Computer Interaction -- Present and Future", IEEE International Convention Record, Part 6, March 1966

Minsky, M. L., and S. A. Papert, "Unrecognizable Sets of Numbers", Journal of the ACM, vol. 13, no. 2, April 1966, pp. 281-286

Moses, J., "Solution of Systems of Polynomial Equations by Elimination", Washington, D. C., March 1966, ACM Symposium on Symbolic and Algebraic Manipulation, Washington, D. C., March 1966; also published in Communications of the ACM, vol. 9, no. 8, August 1966

Oldfield, J. V., "Graphical Input-Output Techniques in Power Computation", International Power Systems Computation Conference, Stockholm, Sweden, July 1966

Overhage, C. F. J., and R. J. Harman, et al, INTREX, Report of a Planning Conference on Information Transfer Experiments, M. I. T. Press, September 3, 1965

Parker, R. R., "Beam-Plasma Discharge: System C", M. I. T. R. L. E. Quarterly Progress Report No. 81, April 1966

Pennell, M. M., "Newton's Method for Finding Complex Roots of a Transcendental Equation", M. I. T. R. L. E. Quarterly Progress Report No. 80, January 1966, p. 263

Pennell, M. M., "Numerical Example to Illustrate Kizner's Method for Solving Nonlinear Equations", M. I. T. R. L. E. Quarterly Progress Report No. 80, January 1966, p. 266

Pennell, M. M., "Further Computations Using Newton's Method for Finding Complex Roots of a Transcendental Equation", M. I. T. R. L. E. Quarterly Progress Report No. 81, April 1966, pp. 253-254

Ross, D. T., S. A. Coons, and J. E. Ward, "Investigations in Computer-Aided Design for Numerically-Controlled Production", M. I. T. E. S. L. Interim Engineering Progress Report for December 1964 through May 1965, ESL-IR-262, March 1966

Scherr, A. L., "Time-Sharing Measurement", Datamation, vol. 12, no. 4, April 1966, pp. 22-26

Selwyn, L., "The Information Utility", Industrial Management Review, Spring 1966

Stockham, T. G., Jr., "High-Speed Convolution and Correlation", Proceedings of the Spring Joint Computer Conference, vol. 28, April 1966, pp. 229-233

Strachey, C., "A General Purpose Macrogenerator", The Computer Journal, vol. 8, October 1965, pp. 225-241

Tillman, C., "A General Console-Operated Program for Equilibrium Field Problems in Two-Dimensional Continua", Mathematical Algorithms, Preliminary Issue, January 1966

Weingartner, H., "Capital Budgeting of Interrelated Projects: Survey and Synthesis", Management Science, vol. 12, no. 7, March 1966

Weizenbaum, J., "ELIZA: A Computer Program for the Study of Natural Communication Between Man and Machine", Communications of the ACM, vol. 9, January 1966, pp. 36-45

Weizenbaum, J., "Future Directions and Challenges", ACM Symposium on Symbolic and Algebraic Manipulation, Washington, D. C., March 1966

Weizenbaum, J., On-Line User Languages in On-Line Computing Systems, American Data Processing Inc., 1966

Wolman, B. L., "Operators for Manipulating Language Structures", ACM Symposium on Symbolic and Algebraic Manipulations, Washington, D. C., March 1966

APPENDIX D

PROJECT MAC TECHNICAL REPORTS

REPORT NOS.	ASTIA NOS.	TITLE	AUTHOR(S)	DATE
MAC-TR-1 (THESIS)	AD-604-730	Natural Language Input for a Computer Problem Solving Language	Bobrow, D. G.	6/64
MAC-TR-2 (THESIS)	AD-608-499	SIR: A Computer Program for Semantic Information Retrieval	Raphael, B.	6/64
MAC-TR-3	AD-608-501	System Requirements for Multiple-Access, Time-Shared Computers	Corbató, F. J.	5/64
MAC-TR-4	AD-604-678	Verbal and Graphical Language for the AED System: A Progress Report	Ross, D. T. Feldmann, C. G.	5/64
MAC-TR-6	AD-604-679	STRESS: A Problem-Oriented Language for Structural Engineering	Biggs, J. M. Logcher, R. D.	5/64
MAC-TR-7	AD-604-680	OPL-1: An Open-Ended Programming System within CTSS	Weizenbaum, J.	4/64
MAC-TR-8	AD-604-681	The OPS-1 Manual	Greenberger, M.	5/64
MAC-TR-11	AD-608-500	Program Structure in a Multi-Access Computer	Dennis, J. B.	5/64
MAC-TR-12	AD-609-296	The MAC System: A Progress Report	Fano, R. M.	6/64
MAC-TR-13	AD-609-288	A New Methodology for Computer Simulation	Greenberger, M.	10/64
MAC-TR-16	AD-612-702	CTSS Technical Notes	Saltzer, J. H.	3/65
MAC-TR-17	AD-462-158	Time-Sharing on a Multi-Console Computer	Samuel, A. L.	3/65
MAC-TR-18 (THESIS)	AD-470-715	An Analysis of Time-Shared Computer Systems	Scherr, A. L.	6/65
MAC-TR-19 (THESIS)	AD-474-018	A Heuristic Approach to Alternate Routing in a Job Shop	Russo, F. J.	6/65
MAC-TR-20 (THESIS)	AD-474-019	CALCULAID: An On-Line System for Algebraic Computation and Analysis	Wantman, M. E.	9/65

REPORT NOS.	ASTIA NOS.	TITLE	AUTHOR(S)	DATE
MAC-TR-21 (THESIS)	AD-624-943	Queueing Models for File Memory Operation	Denning, P. J.	10/65
MAC-TR-22	AD-625-728	The Priority Problem	Greenberger, M.	11/65
MAC-TR-23	AD-627-537	Programming Semantics for Multiprogrammed Computations	Dennis, J. B. Van Horn, E. C.	12/65
MAC-TR-24	AD-476-443	MAP: A System for On-Line Mathematical Analysis	Kaplow, R. Strong, S. L. Brackett, J. W.	1/66
MAC-TR-25 (THESIS)	AD-631-396	Investigation of an Analog Technique to Decrease Pen-Tracking Time in Computer Displays	Stratton, W. D.	3/66
MAC-TR-26 (THESIS)	AD-631-269	Design of a Low-Cost Character Generator for Remote Computer Displays	Cheek, T. B.	3/66
MAC-TR-27 (THESIS)	AD-633-678	OCAS: On-Line Cryptanalytic Aid System	Edwards, D. J.	5/66
MAC-TR-28 (THESIS)	AD-637-215	Input/Output in Time-Shared, Segmented, Multiprocessor Systems	Smith, A. A.	6/66
MAC-TR-29 (THESIS)	AD-636-275	Search Procedures Based on Measures of Relatedness Between Documents	Ivie, E. L.	6/66

MAC-PR-1	AD-465-088	PROJECT MAC: Progress to July 1964
MAC-PR-2	AD-629-494	PROJECT MAC: Progress Report II July 1964 to July 1965

AUTHOR INDEX

ADMINISTRATION AND SERVICES

ARTIFICIAL INTELLIGENCE

BIOLOGY DEPARTMENT

CIVIL ENGINEERING DEPARTMENT

COMPUTATION STRUCTURES

COMPUTER OPERATION

COMPUTER SYSTEM RESEARCH

ELECTRONIC SYSTEMS LABORATORY

LIBRARY RESEARCH

LINCOLN LABORATORY

MAN-MACHINE COMMUNICATION

NON-M. I. T. USERS

RESEARCH LABORATORY OF ELECTRONICS

SCHOOL OF ENGINEERING

SCHOOL OF HUMANITIES AND SOCIAL SCIENCES

SCHOOL OF SCIENCE

SLOAN SCHOOL OF MANAGEMENT

UNCLASSIFIED

Security Classification

DOCUMENT CONTROL DATA - R&D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY *(Corporate author)* Massachusetts Institute of Technology Project MAC	2a. REPORT SECURITY CLASSIFICATION UNCLASSIFIED 2b. GROUP None

3. REPORT TITLE

Project MAC
Progress Report III: July 1965 to July 1966

4. DESCRIPTIVE NOTES *(Type of report and inclusive dates)*

annual progress

5. AUTHOR(S) *(Last name, first name, initial)*

Collection of material from many Project MAC participants; no specific author or authors

6. REPORT DATE	7a. TOTAL NO. OF PAGES	7b. NO. OF REFS
1 July 1966	312	—

8a. CONTRACT OR GRANT NO. Office of Naval Research, Nonr-4102(01) b. PROJECT NO. Nr-048-189 c. RR 003-09-01 d.	9a. ORIGINATOR'S REPORT NUMBER(S) MAC-PR-3 9b. OTHER REPORT NO(S) *(Any other numbers that may be assigned this report)*

10. AVAILABILITY/LIMITATION NOTICES

Distribution of this document is unlimited

11. SUPPLEMENTARY NOTES None	12. SPONSORING MILITARY ACTIVITY Advanced Research Projects Agency 3D-200 Pentagon Washington, D. C. 20301

13. ABSTRACT

The broad goal of Project MAC is experimental investigation of new ways in which on-line use of computers can aid people in their individual work; whether research, engineering design, management, or education.

This is the third annual Progress Report summarizing the research carried out under the sponsorship of Project MAC. Details of this research may be found in the publications listed in the Appendices at the end of the report.

14. KEY WORDS

Computers
Machine-aided cognition
Multiple-access computers
On-line computer systems
Real-time computer systems
Time-sharing
Time-shared computer systems

DD FORM 1 JAN 64 1473 (M.I.T.)

UNCLASSIFIED

Security Classification