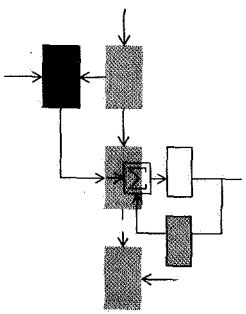
April, 1969

Report ESL-SR-379
M.I.T. Project DSR 76152
NASA Research Grant ~~NsG-496~~ (Part)
NGL-22-009-019



COMPUTER-AIDED ELECTRONIC CIRCUIT DESIGN

Michael L. Dertouzos

Status Report

June 1, 1968 - November 30, 1968

Electronic Systems Laboratory

MASSACHUSETTS INSTITUTE OF TECHNOLOGY, CAMBRIDGE, MASSACHUSETTS 02139

Department of Electrical Engineering

April, 1969 Status Report ESL-SR-379

COMPUTER-AIDED ELECTRONIC CIRCUIT DESIGN

by

M. L. Dertouzos

Status Report

June 1, 1968 - November 30, 1968

Electronic Systems Laboratory
Department of Electrical Engineering
Massachusetts Institute of Technology
Cambridge, Massachusetts 02139

ABSTRACT

This report describes progress and plans in the area of On-Line Computer-Aided Circuit and System Design. Specific topics of continuing research include: tearing of networks into subnetworks for improving computational efficiency; recursive analysis of nonlinear networks; progress in the CIRCAL-II on-line circuit-design program, and progress in LOTUS, an on-line program for block-diagram system synthesis.

New topics include: implicit computation -- a new computing approach for solving nonlinear network simulation problems -- and theoretical work in the analysis and synthesis of continuous systems.

ACKNOWLEDGMENT

The preparation and publication of this report, including the research on which it is based, was sponsored under a grant to the Electronic Systems Laboratory, Massachusetts Institute of Technology, DSR Project No. 76152. This grant is being administered as part of the National Aeronautics and Space Administration Research Grant No. NsG-496 (Part). This work was also supported in part by Project MAC, an MIT research project sponsored by the Advanced Research Projects Agency, Department of Defense, under Office of Naval Research Contract Nonr-4102(01). This report is published for information purposes only and does not represent recommendations or conclusions of the sponsoring agencies. Reproduction in whole or in part is permitted for any purpose of the United States Government.

The NASA/ERC Project Monitor for this work is Dr. William W. Happ.

CONTENTS

LIST OF FIGURES

A. INTRODUCTION AND SUMMARY

Progress in On-Line Computer-Aided Circuit and System Design proceeded along several directions: In the area of network tearing, a simple algorithm has been developed which tears a network into two subnetworks, and guarantees, on the average, a specified saving in the computational effort needed to analyze that network. In the recursive analysis of networks, work centered on the determination of network-order -- a prerequisite to computer analysis by recursive techniques. Work on CIRCAL-II continued with the development and test of (1) a "pseudo-user", i.e., a program that replaces Man for the automatic optimization of networks and (2) a method for dynamically loading and unloading program sections in order to maximize the storage available for analysis. New work in this area includes implementation of nonlinear-transient, symbolic-frequency, linear-time and recursive-analysis techniques, development of a new function-definition ability, and documentation of the CIRCAL-II system. In Computer-Aided System Design, the initial version of LOTUS was completed, and new work was initiated in the areas of Implicit Computation and Continuous-System Theory; the objective of this work is to better understand the computational capabilities of continuous systems and to develop effective continuous/discrete computing techniques for the solution of linear and nonlinear systems of equations.

B. TEARING OF NETWORKS

Professor M. L. Dertouzos

Mr. C. W. Therrien,
Research Assistant

Work has continued on the use of tearing for reducing computation and expediting the computer solution of electrical networks. Several theorems have been proven. These theorems and their associated results relate network structure to feasibility of tearing and reduction of computation. In particular, they indicate when it is unprofitable to tear a network, and they establish bounds on the connectedness or "density" of the network graph for profitable tearing.

A simple tearing algorithm has been thoroughly investigated for graphs belonging to a certain equivalence class. This class contains all graphs whose highest-degree vertex has degree λ (an integer greater than 2). The algorithm starts at any vertex of a given graph and labels it "0". Thereafter, all vertices previously unlabeled and adjacent to vertices labeled "d" are labeled "d+1"; if a vertex labeled "d" is adjacent to no unlabeled vertex, it is marked with a ($\surd$). A graph thus labeled is shown in Fig. 1. Observe on this figure that any

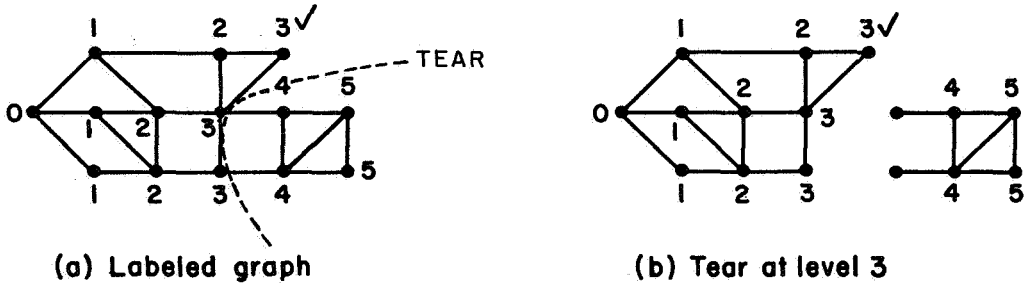


Fig. 1 Illustration of Tearing Algorithm

set of unchecked vertices with the same label, for example 3, produces a tear of the graph. For a graph having N total vertices and highest-degree vertex of degree λ, it has been shown that the tearing parameters k (number of vertices at the tearing point) and m (number of vertices in one of the torn sections), when the graph is torn at vertices labeled d, must lie in a restricted region R_d of the k-m space. For N=36, λ=3, and d=3 this region is the shaded area of Fig. 2. Recall (from Status Report ESL-SR-337) that the triangle bounds the region of all tears for N-vertex graphs, and the curved lines of Fig. 2 represent points of equal relative computation.

Additional results in this area have established the probability of a graph being torn at a given point within region R_d, and the statistical properties of the relative computation at such a point. The heavy center curve of Fig. 2 represents an average value of the computation function which over the shaded region is about 60 percent. A standard deviation of ±10 percent is indicated by the dotted lines.

Using these results, a table has been constructed that lists the optimal value of d resulting in the smallest value for the expected

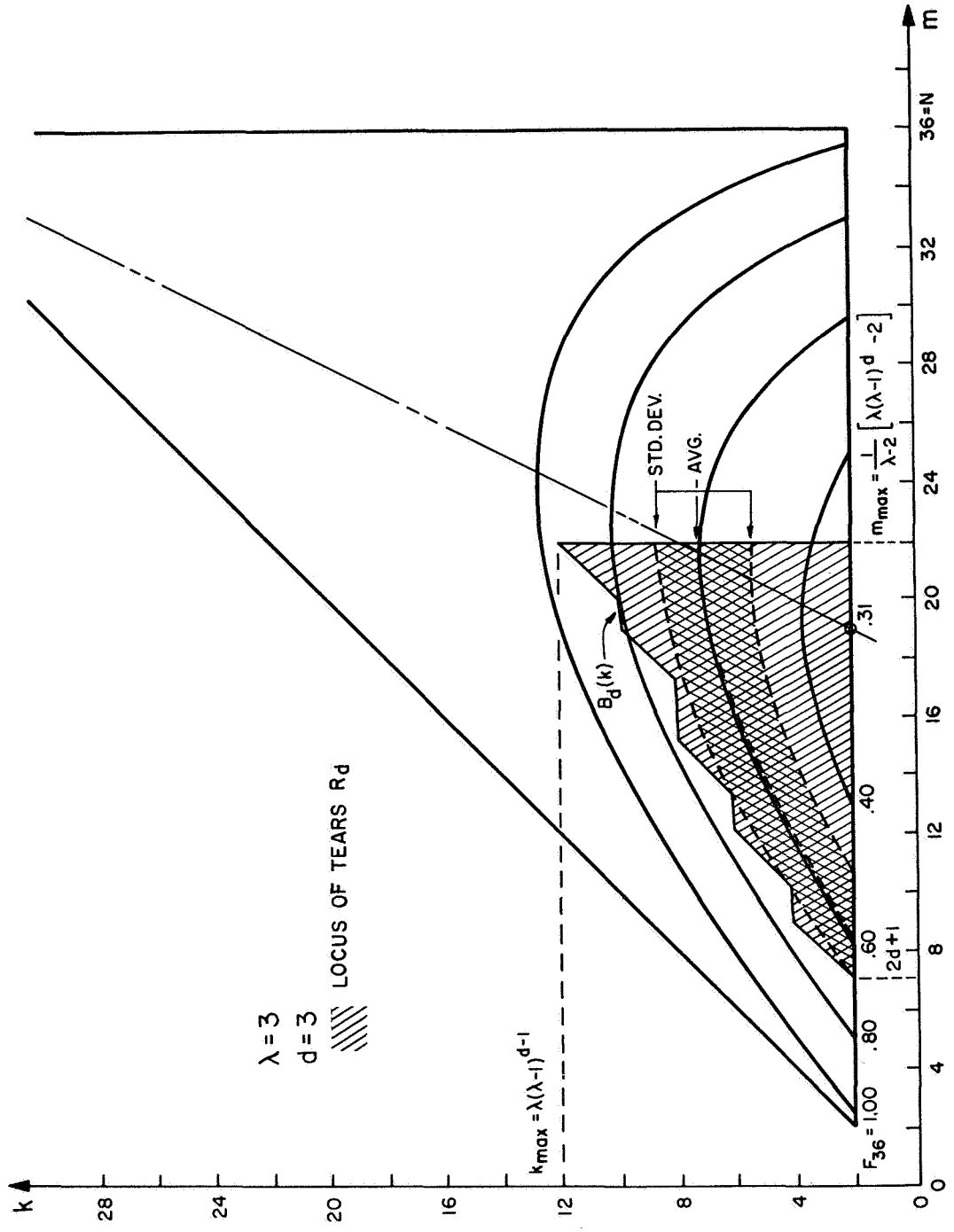


Fig. 2 Locus of Tears for Networks with Constrained Maximum Degree

relative computation of a given graph. The algorithm has been used in conjunction with this table to obtain relative savings in computation of 30 to 60 percent for a variety of networks.

C. A RECURSIVE APPROACH TO NETWORK ANALYSIS

Professor M. L. Dertouzos

Mr. H. L. Graham,
Research Assistant

Recent work in this area has centered on the development of methods for determination of network order.* This work has taken two directions; bounds have been formulated on the order of a network from properties of the network-order relation, and several methods have been explored for obtaining the exact order of a network by search techniques.

A greatest lower bound on the order of a network has been obtained which is dependent on the position of the network in the ℓ-n plane (ℓ = number of adjacent node pairs, n = number of nodes). It has been shown that if a network has order q then:

$$\ell \leq \lambda_1(q)\, n - \lambda_2(q)$$

where:
$$\lambda_1(q) = \frac{5q + 2 + \text{odd}(q)}{4}$$

in which:
$$\text{odd}(q) = \begin{cases} 1 & \text{for } q \text{ odd} \\ 0 & \text{for } q \text{ even} \end{cases} \qquad (1)$$

and:
$$\lambda_2(a) = (q + 1)\, \lambda_1(q) - \frac{q(q+1)}{2} \qquad (2)$$

These greatest lower bounds are illustrated for several values of q by the dotted lines of Fig. 3.

A previously developed* upper bound on the order of a network states that the order of a full graph on n nodes is the integer q such that $2/3(n-1) \leq q < 2/3\, n$; therefore, this number bounds the order

---

* For definition of order see "Computer-Aided Electronic Circuit Design", Electronic Systems Laboratory Status Report ESL-SR-363, May 31, 1968.

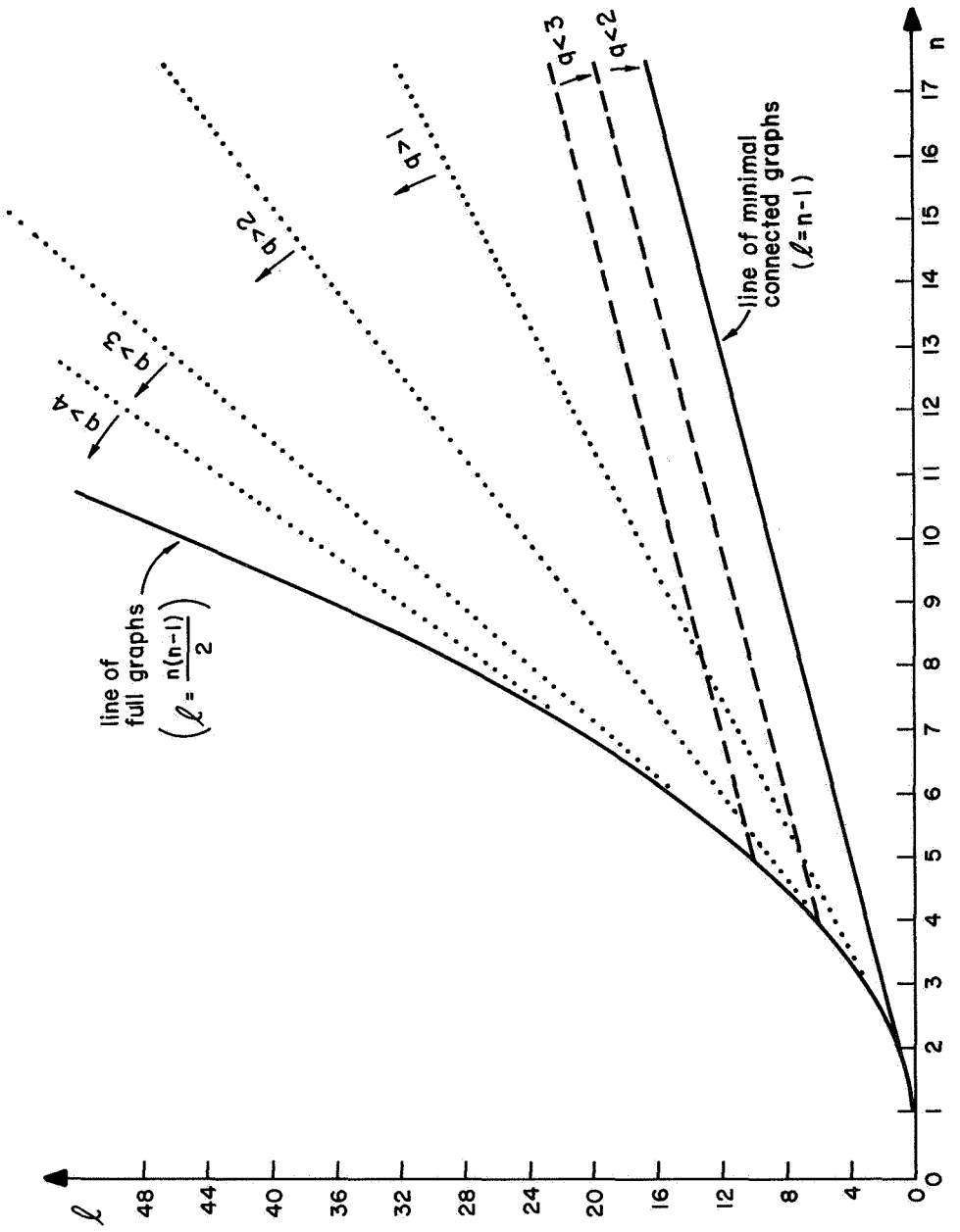


Fig. 3 Bounds on Networks with a Restriction on the Number of Adjacent Node Pairs

of any n-node network by 2/3 n. A tighter upper bound is possible for some networks. In particular, it has been shown that the graph of an order-two network must contain the full graph on four nodes, and the graph of an order-three network must contain the full graph on five nodes. These properties imply that (1) for an order-two network $\ell \geq n+2$, and (2) for an order-three network $\ell \geq n+5$. The dashed lines of Fig. 3 illustrate how these two conditions provide least upper bounds on the order of all the networks lying below these lines.

Finally, observe that an exhaustive search over all possible network construction sequences will determine, exactly the order of a given network. Some work was conducted in the exploration of non-exhaustive search algorithms for the determination of network order. Two such algorithms were developed that yield "good" network constructions; neither one, however, will guarantee finding the minimal network construction, hence the order of the network.

D. CIRCAL-II; SYSTEM DEVELOPMENTS

Professor M. L. Dertouzos

Mr. G. P. Jessel,
Research Assistant
Mr. D. Isaman,
Fellowship Student
Mr. J. R. Stinger,
Research Assistant

Four developments have taken place in the main system of CIRCAL-II. They are: (1) a preliminary investigation and implementation of the "pseudo-user" or Defined-Command feature (DEFCOM); (2) a method for controlling the dynamic loading and unloading of program sections in order to maximize available storage; (3) extension of the defitional capabilities of CIRCAL-II, including the results of (1), above; and (4) documentation.

1. The Pseudo-User

A preliminary version of DEFCOM is now available on CIRCAL-II. This feature, which allows a user to define a new command, operates interpretatively, performing automatic optimization of a network by varying the network parameters until some user-specified relation on the computed variables is satisfied. On the basis of these results,

current work is oriented toward development of a higher-level DEFCOM which will include as standard a set of "engineering oriented" commands, in addition to a more primitive set of statements for the user who has more programming experience. In addition, the new version integrates DEFCOM with the other definitional abilities of CIRCAL-II, as discussed below.

2. Loader/Unloader

The dynamic loader/unloader package of AED has been incorporated in CIRCAL-II. Acting in conjunction with the AED Free-Storage Package, it provides a method for obtaining additional space for necessary computations by unloading unneeded programs whenever, during execution, free storage becomes available. This, in general, occurs in analysis during the formulation of the network matrices. The loader provides for the automatic loading of previously unloaded programs as they are needed. Use of this feature has resulted in satisfactory analysis of 60-node circuits by CIRCAL-II.

3. Definitional Capabilities

Work has been initiated in the final development of that section of CIRCAL-II which integrates all necessary definitional requirements. These are: (1) Implementation of a DEFCOM capability on the basis of the above results so that experienced as well as inexperienced engineers may introduce within CIRCAL-II automatic optimization procedures, or "pseudo-users". Here, specific tasks involve investigation of alternative translator implementations (compiler vs. interpreter) and the judicious selection of DEFCOM primitives which are general enough to treat a multitude of cases, yet closely related to the optimization needs of network design. (2) Implementation of the function- and functional-defining capabilities of CIRCAL-II. Functions so defined will be used within the system for such tasks as : (i) identification of a voltage-current characteristic for a new element: (ii) definition of a voltage-versus-time waveform of a source, and (iii) post-processing operations, e.g., the evaluation and display by the program of functions of the computed variables for monitoring of the results by the user or the pseudo-user (DEFCOM). Functionals will be normally used for the specification of hysteresis and for the simulation of heat effects.

4. Documentation

The following documentation of CIRCAL-II is now in preparation: (1) a user's manual, (2) a programmer's manual for the writers of new analysis routines and (3) a paper for the IEEE Proceedings describing the overall objectives and structure of CIRCAL-II.

E. CIRCAL-II; NEW ANALYSIS TECHNIQUES

Professor M. L. Dertouzos

Mr. K. Hatch,
Fellowship Student
Mr. M. Lum,
Fellowship Student
Mr. C. Lynn,
Fellowship Student
Mr. K. VanBree,
Fellowship Student

One of the main attributes of CIRCAL-II is that it is a general-purpose computer-aided circuit-design program which accepts in a modular way a variety of analysis techniques. Since the main program became operational during the last reporting period, work was recently initiated on the development of several new analysis techniques. The corresponding tasks involve implementation of (1) the recursive analysis technique, the theory of which has been largely developed, as discussed in Section C, above; (2) a symbolic frequency-analysis technique for very large, linear, sparse networks; (3) linear-time and (4) nonlinear transient analysis techniques. In the following, we discuss briefly the first two of these techniques.

1. Recursive Analysis

It has been proposed (see Section C, above) that analysis of nonlinear electrical networks by recursive composition/decomposition of their graphs, should be faster than iteration for a large class of such networks. According to this method, a series of steps is first found for constructing the network from smaller subnetworks, according to a set of construction rules; each construction step corresponds to a set of algebraic manipulations on the nonlinear functions characterizing each subnetwork. After the complete sequence of construction steps has been established, and the corresponding algebraic manipulations made, a nonlinear function describing the driving-point characteristics of

the network can be formed. Any desired internal variable can then be evaluated by following the construction sequence "backwards".

The implementation and testing of these ideas requires the development of algorithms to add, combine, compose, and invert the nonlinear functions involved, and a data structure which is suitable for these operations. Initially, the functions will be represented by tables of breakpoints; current work involves the study of approximation techniques, which try to keep the number of breakpoints limited as the complexity of the functions increases through execution of the foregoing construction steps.

2. Symbolic Frequency Analysis

This approach incorporates some of the recently developed techniques for the analysis of sparse networks. Of interest here is the development of a compiler-type approach which retains the complex frequency, s, in symbolic or quasi-symbolic form. That is, given the system

$$(A - sI)\, x = y \qquad (1)$$

where A is an $m \times n$ network matrix,
s is the complex frequency,
I is the identity matrix,
x the response vector, and,
y the excitation vector;

it is desired to find x, given y, for several (about one hundred) values of s. If the matrix (A - sI) is constant, then it is possible to develop, through established techniques, a sequence of machine-code instructions which, given a value for y, will successively transform that value into a value for x. The approach normally used does not establish or store the inverse matrix and it stores only the nonzero terms of the (A - sI) matrix. In the most straightforward extension of this approach, it is necessary to compute the matrix A - sI and then to develop the compiled code, for each value of s. We feel that such an approach is wasteful since it does not exploit the fact that the location of the nonzero terms of A - sI is independent of the value of s. Our objective is to establish in one pass the compiled code, and then supply s and y as values to be processed by that code. Early results indicate

several alternatives for accomplishing this goal, and current work revolves on a comparative evaluation of the computational needs and merits of these alternatives.

F. COMPUTER-AIDED SYSTEM DESIGN

Professor M. L. Dertouzos
Mr. M. E. Kaliski,
Fellowship Student
Mr. J. R. Stinger,
Research Assistant

During the reporting period, work was conducted in three main areas: (1) continuation of the development of LOTUS, a program for the on-line simulation of block-diagram systems; (2) synthesis of continuous systems, and (3) implicit computation. These areas are further discussed in the following:

1. LOTUS

Research in LOTUS was completed in late August, 1968. An initial version of LOTUS was completed in January, 1968 and was described in a corresponding master's thesis.[1] Since that time, some work has been devoted to improving and expanding the LOTUS program. Results of this work have culminated in two papers: (1) a summary of recommendations and modifications concerning LOTUS, which has suggested several new thesis topics, and (2) a paper entitled "On-Line Simulation of Block-Diagram Systems," which will appear shortly in the IEEE Transactions on Computers.

2. Synthesis of Continuous Systems

This work, which was initiated in July, 1968, has as its main purpose the computer synthesis of generally nonlinear continuous dynamic systems in state-model form, from a given description of the desired input-output task. Our motivation here lies in the approaches used for the synthesis of sequential circuits (sc's), and in the possibility of imbedding progressively more refined discrete systems, such

---

[1] Kaliski, M. E. and K. P. Polzen, "LOTUS: On-Line Simulation of Block-Diagram Systems," S.M. Thesis, MIT, Electronic Systems Laboratory, Department of Electrical Engineering, January, 1968.

as these sc's, within a given continuous system. Some specific problems in this area are as follows:

1. Identification, if this is possible, of a language similar to that of regular expressions for characterization of the input-output tasks performed by continuous dynamic systems

2. Generation of the state-transition function from the input-output description

3. Identification of the dimensionality of the task, e.g., how many integrators are required to accomplish the task

4. Identification of the mechanics for synthesis of the system, once the foregoing have been established.

To realize this general goal, two separate directions of attack have been initially taken. One direction is to analyze various continuous dynamic-system machine forms in order to establish their capabilities. For example, continuous analogues of "Turing Machines" have been postulated and their logic capabilities will be studied. The other direction is to develop, for various continuous systems, discrete computational models which simulate them.

The notion of a system computable by a Turing Machine has been developed, and it is believed that many of the common engineering systems fall into this class. The notion of a system encoding has been defined in a fashion analogous to the canonical indexing of sets in recursion theory. Such an encoding is shown to be isomorphic, in a very natural way, to the system itself. This approach allows concentration on suitable encodings of continuous dynamic systems. For some of these encodings, computational realizations will be studied. In particular, for a subclass of the systems that can be encoded into the real numbers, it is conjectured that a synthesis in the form of a function block with an integrator feedback loop can be made.

3. Implicit Computation

This work was initiated during the latter part of the period covered by this report. Its main purpose is to establish hybrid structures which are (i) computationally effective, i.e., whose resulting accuracy may be improved through progressively more work, as in digital systems,

and (ii) fast, as are strictly analog systems. A related task is the identification of problems besides the known case of systems of equations, that can benefit from this approach. The term "implicit computation" refers in general to the "computation" performed by any implicit system (e.g., any continuous system with feedback), and, in particular, to the computation performed by a computer designed to operate in a manner similar to an implicit system.

Several structures have been proposed and investigated. These structures, which are similar to so-called "hybrid computers", have both a digital and an analog part. They differ, however, from a hybrid computer in two respects: (i) the analog part controls the digital part; and (ii) the digital part operates asynchronously. The digital part of such an implicit computer is set up from basic digital building blocks (e.g., a digital summer) to check the given equations. For example, if it is desired to find a vector $\underline{x}$, so that $\underline{f}(\underline{x}) = 0$, the digital part may compute a norm of $\underline{f}(\underline{x})$ to establish how close $\underline{x}$ is to the solution. The corrective difference equations are simulated by the analog part of the implicit computer. By making suitable connections between the analog part and the digital part of the computer (using analog-to-digital and digital-to-analog converters where necessary) the analog part corrects the operation of the digital part. As the system approaches a solution, the analog part is caused to look at progressively finer differences of the digital part, until a satisfactory accuracy has been achieved.

Present-day computers operate in a predominately sequential fashion, performing an operation only after the conclusion of the previous operation. Thus, the computation is ordered almost entirely in time. This type of operation is well suited for solving a large class of problems, e.g., the simulation of a one-dimensional explicit system, but not for others, such as the simulation of an electrical network with nonlinear elements where the characterizing equations are in implicit form. An implicit approach, on the other hand, is better suited for the latter type of problem, since it does perform computations in space as well as time, having been set up in a configuration appropriate to the problem which is being solved. Systems, such as a complicated network of pipes carrying a fluid, where the "computation" of

how much fluid flows at what rate in each pipe is distributed in space as well as time, operate on the basis of (1) pressures which build up and quickly distribute throughout the system when the input or any of the system variables change value, and (2) relaxation or flow of the system variables to reduce these pressures. This flow is such that each part of the system reacts only to the pressures exerted directly on it, without regard to other remote parts of the system.

In the above implicit computer, the analog part generates the "pressures" which are distributed to the digital blocks. Each block then operates only on the basis of the pressures that it "sees" revising its "flow" by changing the digital values of the corresponding variables.

Present research in implicit computation is concentrated in three areas: machine configuration, accuracy of computation, and simulation of linear systems. There are many possible configurations for the implicit computer, most of them concerned with the method of combining the analog and digital parts. Two major configurations are: (1) an entirely separate analog part, except for a few connections to the digital part; and (2) a duplicate approach where each digital block contains an analog part, connected to it and to the analog parts of the digital blocks. The advantages and disadvantages of these and other configurations are being investigated.

The accuracy of the implicit computer is being studied in terms of how accurate the analog part must be to guarantee a certain accuracy in the digital part. In particular, the possibility of having the system go through several cycles of pressure and flow, per solution point, with part of the solution being generated during each cycle is under investigation.

To narrow the class of problems, we have concentrated on the useful case of linear-system simulation by the implicit computer. It is expected that such a configuration will be naturally extendable to the simulation of nonlinear systems, through use of the inverse-Jacobian matrix for the necessary corrections.

G. PUBLICATIONS OF THE PROJECT

1. Current Publications

a. Reports

"Computer-Aided Electronic Circuit Design," Status Report, ESL-SR-363, September, 1968.

b. Technical Papers and Conference Participation

Dertouzos, M. L., Chairman Computer Graphics Session, and Panelist in two panel discussions, MIT-Technical University of Berlin Joint Summer Conference on "The Computer in the University," Berlin, Germany, July 22-August 2, 1968.

Dertouzos, M. L., "Computer-Aided Circuit Design," (paper and demonstration) MIT-Technical University of Berlin Joint Summer Conference on "The Computer in the University," Berlin, German, July 22-August 2, 1968.

Narud, J. A., Dertouzos, M. L., Jessel, G. P., "Computer-Aided Analysis for Integrated Circuits," Invited paper, Proceedings of the 1968 IEEE International Symposium on Circuit Theory, Miami Beach, Florida, December 4-6, 1968.

2. Past Publications

a. Reports

"Computer-Aided Electronic Circuit Design,"Part I, Status Report ESL-SR-225, December, 1964.

"Computer-Aided Electronic Circuit Design," Part I, Status Report ESL-SR-245, June, 1965.

Dertouzos, M. L., and Therrien, C. W., "CIRCAL: On-Line Analysis of Electronic Networks," Report ESL-R-248, December, 1965.

Dertouzos, M. L., and Santos, P. J., Jr.,"CADD: On-Line Synthesis of Logic Circuits," Report ESL-R-253, December, 1965.

"Computer-Aided Electronic Circuit Design," Part I, Status Report ESL-SR-256, December, 1965.

"Computer-Aided Electronic Circuit Design," Part I, Status Report ESL-SR-274, June, 1966.

"Computer-Aided Electronic Circuit Design," Part I, Status Report ESL-SR-298, January, 1967.

"Computer-Aided Electronic Circuit Design," Part I, Status Report ESL-SR-322, September, 1967.

"Computer-Aided Electronic Circuit Design," Part I, Status Report ESL-SR-337, January, 1968.

b. Thesis Proposal

Therrien, C. W., Tearing of Networks, proposal for Ph.D. dissertation, Department of Electrical Engineering, MIT, March, 1968.

c. Theses

Dvorak, A. A., "An Input-Output Program for Electronic Circuits Using a CRT," Bachelor of Science Thesis, Department of Electrical Engineering, June, 1965.

Santos, P., "CADD, A Computer-Aided Digital Design Program," Master of Science Thesis, Department of Electrical Engineering, June, 1965.

Therrien, C. W., "Digital-Computer Simulation for Electrical Networks," Master of Science Thesis, Department of Electrical Engineering, June, 1965.

Fluhr, Z. C., "Single-Threshold Element Realizability by Minimization," Master of Science Thesis, Department of Electrical Engineering, August, 1965.

Olansky, K. J., "A Low-Cost Teletype-Operated Graphical Display," Master of Science Thesis, Department of Electrical Engineering, August, 1965.

Gertz, J. L., "A Graphical Input-Output Program for Digital System Simulation," Master of Science Thesis, Department of Electrical Engineering, June, 1966.

Graham, H. L., "A Hybrid Graphical Display Technique," Master of Science Thesis, Department of Electrical Engineering, June, 1966.

Meltzer, J. R., "CIRCAL: An Input for Nonlinear Elements," Master of Science Thesis, Department of Electrical Engineering, June, 1966.

Taubman, C. N., "Computer Analysis of Electrical Analog Distribution Systems," Master of Science Thesis, Department of Electrical Engineering, June, 1966.

Walpert, S. A., "An Output Program for Representing Electrical Signals," Master of Science Thesis, Department of Electrical Engineering, June, 1966.

Edelberg, M., "A Dual Approach to Threshold Decomposition of Boolean Functions," Master of Science Thesis, Department of Electrical Engineering, June, 1967.

Willems, J. D., "Synthesis of Logical Functions with Restricted Threshold Elements," Electrical Engineer Thesis, Department of Electrical Engineering, June, 1967.

Smith, T., "Nesting of Networks for Computer-Aided Circuit Design," Bachelor of Science Thesis, Department of Electrical Engineering, June, 1967.

Kaliski, M. E., and Polzen, K. P., "LOTUS, On-Line Simulation of Block Diagram Systems," Master of Science joint thesis, Department of Electrical Engineering, MIT, January, 1968.

Stinger, J. R., "A General Data Structure for On-Line Circuit Design," Master of Science Thesis, Department of Electrical Engineering, MIT, January, 1968.

d. Technical Papers and Conference Participation

Reintjes, J. R., and Dertouzos, M. L., "Computer-Aided Design of Electronic Circuits," WINCON Conference, February, 1966, Los Angeles, California.

Reintjes, J. F., "The Role of Computers in Modern Design Technology," Conference on Computer-Aided Design, University of Wisconsin, May 3-4, 1966.

Dertouzos, M. L., and Graham, H. L., "A Parametric Graphical Display Technique for On-Line Use," presented at Fall Joint Computer Conference on November 8, 1966. Published in the Conference Proceedings of the FJCC.

Therrien, C. W., and Dertouzos, M. L., "CIRCAL: On-Line Design of Electronic Circuits," presented at the NEREM Show and published in NEREM record November, 9, 1966.

Notes: (1) Professor Dertouzos was Chairman of the Computer-Aided Electronic Circuit Design Session at the NEREM 1966 Conference, Boston, Massachusetts.

(2) CIRCAL-I was used from Munich, Germany via TELEX, June, 1966, in connection with a series of ten lectures by Professor Dertouzos at Siemens, Halske.

Katzenelson, J., "AEDNET: A Simulator for Nonlinear Networks," Proceedings of the IEEE, Vol. 54, No. 11, November, 1966, pp. 1536-1552.

Therrien, C. W., presentation on CIRCAL at N.Y.U. Conference on "Network Analysis by Computer Symposium," New York University, January, 1967.

Dertouzos, M. L., Panelist in panel discussion, "On-Line Versus Batch," held at the NASA Computer-Aided Circuit Design Seminar, April 11-12, 1967, Cambridge, Mass.

Dertouzos, M. L., "PHASEPLOT: An On-Line Graphical Display Technique," IEEE Transactions on Electronic Computers, Vol. EC-16, No. 2, April 1967, pp. 203-209.

Dertouzos, M. L., and Fluhr, Z. C., "Minimization and Convexity in Threshold Logic," Seventh Annual Symposium on Switching Circuit Theory and Logical Design, Berkeley, California, 1966; IEEE Transactions on Electronic Computers, Vol. EC-16, No. 2, April 1967, pp. 212-215.

Dertouzos, M. L., "CIRCAL: On-Line Circuit Design," Proceedings of the IEEE, Vol. 55, No. 5, May, 1967, pp. 637-654.

Dertouzos, M. L., and Graham, H. L., "A Parametric Graphical Display Technique for On-Line Use," MIT Project MAC Seminar, May 19, 1966.

Evans, D. S., and Katzenelson, J., "Data Structure and Man-Machine Communications Problems," Proceedings of the IEEE, Vol. 55, No. 7, July, 1967, pp. 1135-1144.

Dertouzos, M. L., "An Introduction to On-Line Circuit Design," Proceedings of the IEEE, Vol. 55, No. 11, November, 1967, pp. 1961-1971; also Proceedings of the Fifth Allerton Conference on Circuit and System Theory, October 4-6, 1967 (Invited Paper).

Dertouzos, M. L., "Panel Discussion on Computer-Aided Design," Proceedings of the IEEE, Vol. 55, No. 11, November, 1967, pp. 1777-1778.

Dertouzos, M. L., Chairman of Session "Computer-Aided Circuit Design; A Critical Appraisal," NEREM 1967, Boston, November 3, 1967.

Dertouzos, M. L., Co-chairman of and Lecturer at Industrial Liaison Symposium on "Computer-Aided Circuit Design," MIT, October 3, 1967.

Dertouzos, M. L., Co-chairman of and Lecturer at MIT Summer Course 6.56S on "On-Line Circuit Design," MIT July 6-July 13, 1967. (One set of Proceedings issued at that course.)

Dertouzos, M. L., Talks and Demonstration on On-Line System and Circuit Simulation to:

a. EE Department, Catholic University of America, Washington, D.C., November 10, 1967.

b. IEEE Computer Group, Boston, October 11, 1967.

d. Motion Picture

CIRCAL: Computer-Aided Electronic Circuit Design, January, 1966.