acm

SIGDA NEWSLETTER

SPECIAL INTEREST GROUP ON DESIGN AUTOMATION

Volume 5 Number 1 March 1975

Contents:

ADDRESSES

CHAIRMAN:
Charles E. Radke
IBM Corporation (B99/951)
P. O. Box 390
Poughkeepsie, New York 12602
(914) 485-7775

VICE-CHAIRMAN:
David W. Hightower
Bell Labs 2B312A
Holmdel, New Jersey 07733
(201) 949-6549

SECRETARY/TREASURER:
Lorna Capodanno
Bell Labs 2C169
Murray Hill, New Jersey 07974
(201) 582-6909

EDITOR:
Robert J. Smith, II
P. O. Box 1028
Livermore, California 94550
(415) 447-1100 x 8053

TECHNICAL COMMITTEE:
David W. Hightower, Chairman

MEMBERSHIP COMMITTEE:
Lorna Capodanno, Chairman

PUBLICITY COMMITTEE:
Lorna Capodanno, Chairman

BOARD OF DIRECTORS:

John R. Hanne
Texas Instruments
P. O. Box 5012 (MS 907)
Dallas, Texas 75222
(214) 238-3554

Steven A. Szygenda
Department of Elec. Eng.
University of Texas
Austin, Texas 78712

Donald J. Humcke
Bell Labs 2C-318
Holmdel, New Jersey 07733
(201) 949-6253

Larry Margol
Micro Electronics Division
Rockwell Instruments
D/734-057
Box 3669
Anaheim, California 92803
(714) 632-8565

Charles W. Rose
Computing & Information Sci.
Case Western Reserve Univ.
Cleveland, Ohio 44106
(216) 368-2800

MEMBERSHIP

SIGDA dues are $3.00 for ACM members and $5.00 for non-ACM members. Checks should be made payable to the ACM and may be mailed to the SIGDA Secretary/Treasurer listed above, or to SIGDA, ACM Headquarters, 1133 Avenue of Americas, New York, N. Y. 10036. Please enclose your preferred mailing address and ACM Number (if ACM member).

SIG/SIC FUNCTIONS

Information processing comprises many fields, and continually evolves new subsectors. Within ACM these receive appropriate attention through Special Interest Groups (SIGs) and Special Interest Committees (SICs) that function as centralizing bodies for those of like technical interests ... arranging meetings, issuing bulletins, and acting as both repositories and clearing houses. The SIGs and SICs operate cohesively for the development and advancement of the group purposes, and optimal coordination with other activities. ACM members may, of course, join more than one special interest body. The existence of SIGs and SICs offers the individual member all the advantages of a homogeneous narrower-purpose group within a large cross-field society.

ACTIVITIES

1) Informal technical meetings at SJSS and FJCC.
2) Formal meeting during National ACM meeting + DA Workshop.
3) Joint sponsorship of annual Design Automation Workshop.
4) Quarterly newsletter.
5) Panel and/or technical sessions at other National meetings.

FIELD OF INTEREST OF SIGDA MEMBERS

Theoretic, analytic, and heuristic methods for:

1) performing design tasks,
2) assisting in design tasks,
3) optimizing designs through

the use of computer techniques, algorithms and programs to:

1) facilitate communications between designers and design tasks,
2) provide design documentation,
3) evaluate design through simulation,
4) control manufacturing processes.

CHAIRMAN'S MESSAGE

CHARLES E. RADKE

A nominating committee headed by Dave Hightower along with Steve Krosner and Don Humcke has submitted their list of nominations. Five different organizations are represented.

Chairman

Dr. James G. Linders
Dept. of Applied Analysis and Computer Sciences
University of Waterloo,
Canada

Dr. Luther Abel
Digital Equipment Corp.

Vice Chairman

Dr. Edward Hassler
Texas Instruments

Ms. Judith Brinsfield
Bell Laboratories

Secretary/Treasurer

Mr. Donald J. Humcke
Bell Laboratories

Mr. Carl Ellison
IBM Corporation

I can only encourage you to vote; the voting is up to you.

A little over four years ago SIGDA existed in name only. I have served SIGDA for four-plus years (two and one quarter terms). In the first issue of the Newsletter (Vol. 1, No. 1; January 1971) I wrote the following as the "Message from Your Chairman":

"Is SIGDA for real? I feel that a large percentage of the membership (83 out of around 200) has shown that it is."

"In a letter dated June 10, 1970, Jean Sammet as Chairman of ACM Committee on SIG's and SIC's sent each of us a letter concerning the impending dissolution of SIGDA. Robert Hitchcock of IBM Research tried to reverse the trend by organizing a meeting of interested SIGDA members (and some non-members) at the 1970 DA Workshop in San Francisco. The result was that a list of interested people and potential candidates was generated. The SIGDA Nominating Committee, under chairmanship of J. B. O'Neill, met and increased this list of interested volunteers by four additional people. You received the list of eight candidates, and a whopping 40% of you voted. I don't know on what basis you made your choice, but I do know that we had eight interested and talented people running."

"To put first things first, I consider the most important item on the agenda to be one of communication with the membership. Our only means of achieving this at the present time is through a newsletter. ....."

"Although our first issue is small and meets only a few of the objectives which we think a Newsletter should serve, it is a start. ....."

In the last issue I listed several of the opportunities and professional activities available to you in SIGDA. The statements by the nominees which you will receive with the ballot talk about what they feel they can do for SIGDA. The present officers and Newsletter Editor have shown you what they can do for SIGDA. SIGDA is finanically sound, growing by a healthy rate each year, and its programs and newsletters are, each year, contributing more and more to the "Use of Computers in Design and Engineering."

Your vote shows your support for the candidates so that whoever is elected can continue the growth of SIGDA.

ADDENDUM:

I would like to give a special note of appreciation to our Newsletter Editors (Steve Krosner, first as Editor then as Co-Editor, and Rob Smith, first as Co-Editor and since October 1974 as Editor). During 1974 SIGDA published four full issues of the SIGDA Newsletter and to a set schedule, a first!

Volume 4 No. 1 January 1974	48 Pages
Volume 4 No. 2 June 1974	36 Pages
Volume 4 No. 3 September 1974	36 Pages
Volume 4 No. 4 December 1974	48 Pages

Thanks, Rob; keep it up!

FROM THE EDITOR

Several people have suggested that it would be desirable to organize this Newsletter in a more formal fashion, with regular sections devoted to reviews of recent publications, correspondence, architectural design automation, digital DA and a (very large) number of other topics. That's a great idea, but any significant change in the Newsletter is going to require input from people who want to distribute such material to the SIGDA membership.

As I've noted before, the most difficult task involved in editing this newsletter is obtaining contributed material that is timely and relevant. Several new approaches are going to be used during the next year including:

1) I'll arrange to reprint selected articles that may be of interest to SIGDA members, especially papers which may be difficult to obtain through normal channels. Please suggest to me candidate papers you would like to have reprinted.

2) Readers are invited to volunteer to act as associate editors in charge of special interest sections of the Newsletter. Associate editors would be responsible for preparation of camera ready material related to a specific topic. The basic idea is to promote contributions in areas of active interest. Special sections will appear on a schedule determined by material availability: in the last year, I've had such an overwhelming volume of contributions that virtually everything has been published.

   Bill van Cleemput at the University of Waterloo, Ontario, has volunteered to organize sections on "DA in Universities" and "Recent Publications." Nick Matelan (who works with me at LLL) has volunteered to do a semi-annual "DA for Microprocessor Systems" section. Other volunteers would be very enthuastically welcomed!

3) I will be attempting to solicit working papers for the Newsletter at the DA Workshop in June. If you would like to contribute a paper, letter or short note, why not discuss it with me in Boston?

4) If all else fails, I may start sending out exerpts from the LLL DA system documentation!

If you have other ideas or suggestions, please let me know.

Rob.

Rob Smith

12th Annual

DESIGN AUTOMATION WORKSHOP

BOSTON, MASSACHUSETTS

REQUIREMENTS FOR SUBMITTING PAPERS

If you plan to submit a paper, you should send three copies of the paper (rough drafts are acceptable) to the program chairman no later than January 2, 1975.

Accompanying the draft should be the full name, address, and telephone number of the principal author, with whom all further direct communication will be conducted.

Notification of acceptance will be sent to you during the first week of February, 1975. After notification of acceptance, you will receive detailed instructions on the format to be observed in typing the final copy. To insure the availability of Proceedings at the Workshop, your final manuscript will be due April 21, 1975.

Final papers should be no longer than 5000 words, and the presentation should be limited to 20 minutes. Projection equipment for 35mm slides and viewgraph (overhead projector) foils will be available for every talk. Please indicate what, if any, additional audio-visual aids you require.

Program Chairman

Rough drafts are to be sent to the Program Chairman:

S. A. Szygenda
The University of Texas
Electrical Engineering Department (ENS 515)
Austin, Texas 78712
512-471-7365

Chairman of 12th DAW

R. B. Hitchcock

Sponsors

The sponsors of the Design Automation Workshop are the ACM (Association for Computing Machinery) Special Interest Group on Design Automation and IEEE (Institute of Electrical and Electronics Engineers) Computer Society.

Design Automation

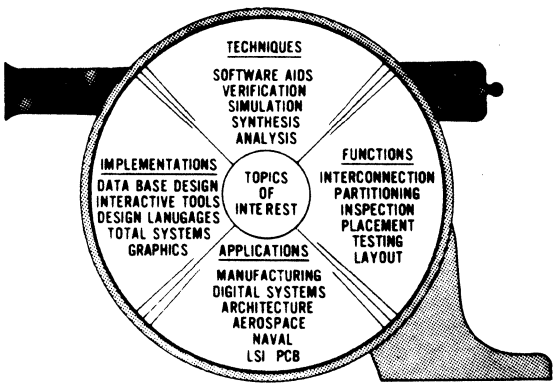
Design Automation implies the use of computers as aids to the design process.

In the broadest sense, the design process includes everything from specifying the characteristics of a product to meet a marketing objective to enumerating the details of how it is to be manufactured and tested.

Thus design automation embraces applications from one end of the design process to the other.

Site of 12th DAW

Statler Hilton Hotel
Park Square at Arlington Street
June 23, 24, 25, 1975



Call for Speakers

Workshop on Methods of Verification in Design Automation or Is the Design Correct?

October 8 - 10, 1975

Michigan State University, East Lansing, Michigan

Co-sponsored by IEEE Computer Society Technical Committees on Design Automation and Fault Tolerant Computing

The rapidly increasing complexity of digital technology has made it mandatory to verify the correctness of a design before committing to manufacturing. This need has, in turn, placed new demands on design automation. Thus, this year's workshop will focus on new methods of verification which have been or are being developed. The following sessions are planned:

Architecture and Logic Verification - Functional simulation, gate level simulation, Boolean comparison, architecture (RTL) - to - logic verification, etc. (Chairman: Steve Chappell, Bell Laboratories)

Microprogramming - Symbolic verification of microcode, interactive debugging/testing systems, verification via comparison to specifications etc. (Chairmen: William Carter, IBM and George Leeman, IBM)

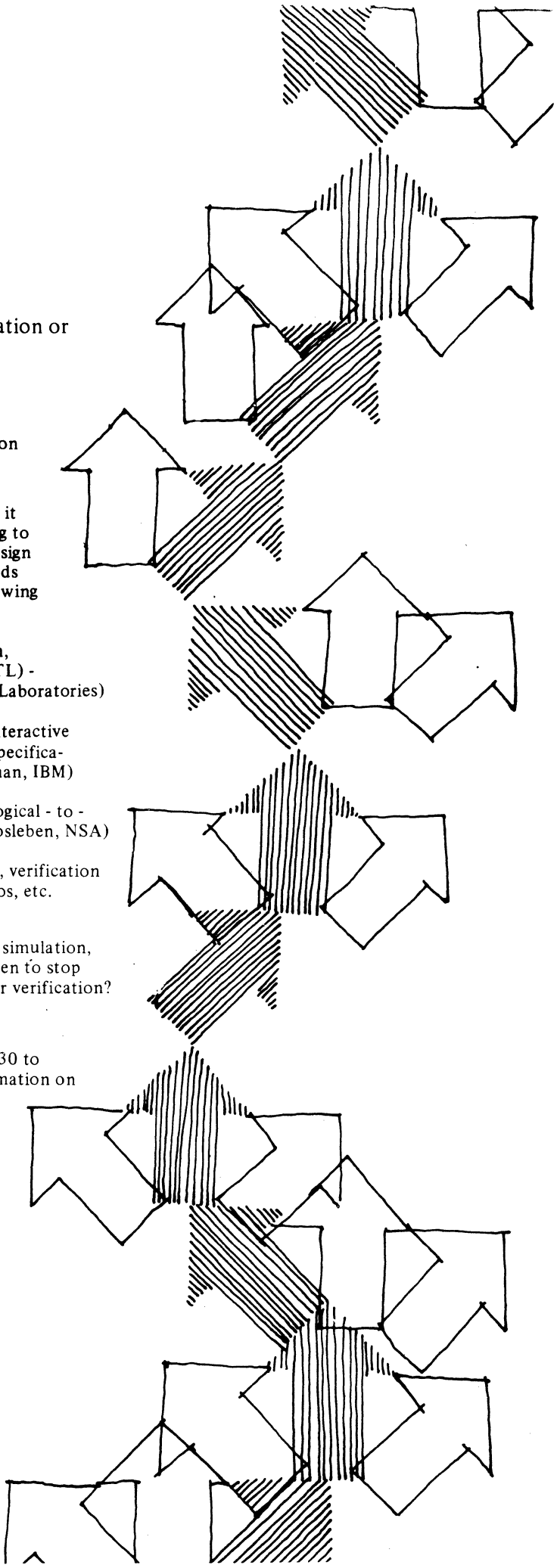
Implementation Verification - Delay and Timing analysis, logical - to - physical checking, shapes checking, etc. (Chairman: Paul Losleben, NSA)

Manufacturing Verification – Testing - DC Tests, AC Tests, verification of manually generated test patterns, testing of arrays, macros, etc. (Chairman: Gernot Metze, University of Illinois)

User Session - How are the programs actually used; e.g., for simulation, how are input patterns determined, how does one know when to stop simulating, etc.? What are the future requirements on DA for verification? (Chairman: Eckhard Schulz, Itek)

Speakers are asked to give a prepared but informal talk of about 30 to 45 minutes length, but a written paper is not required. For information on speaking or attending, contact:

Dr. Roy L. Russo
IBM T.J. Watson Research Center
Post Office Box 218
Yorktown Heights, New York 10598
Tel: 914- 945- 1643



CALL FOR PAPERS

Workshop on Data Bases For Interactive Design

Sponsored by ACM/SIGDA SIGMOD/SIGGRAPH

September 15-17, 1975

University of Waterloo Ontario, Canada

Many computer-aided design systems operate on large volumes of graphical and non-graphical data. Special consideration must be given to data base management systems and working data subsets. A need exists for standardization data base organizations in business and scientific applications.

In this spirit SIGDA, SIGGRAPH, and SIGMOD (the special interest groups of the ASSOCIATION FOR COMPUTING MACHINERY for Design Automation, Graphics, and Management of Data) in co-operation with the University of Waterloo are jointly sponsoring a workshop which will bring together designers who use large quantities of data and computer scientists who develop the interactive data base and graphics systems. The workshop is intended to explore mutual areas of interest and to promote an understanding of inter-related problems.

TOPICS

- Data Base Management Systems and their potential application in computer assisted design.
- Interactive computer graphics systems including data base support.
- Design automation involving large volumes of data.
- Data distribution in design applications (intelligent satelites, data networks).

OBJECTIVES

- Stimulate the development of data base technology and its potential applications in design.
- Examine the role of interactive graphics in the design data base environment.
- Provide a forum for interaction between computer scientists and those involved in the design process.

IMPORTANT DATES

January 20, 1975 – Notice of intention to submit a paper due.
March 15, 1975 – First draft of manuscript, 5000 words or less, due for review.
May 15, 1975 – Authors notified of acceptance of papers.
July 1, 1975 – Final revision of manuscript due.
September 15-17, 1975 – Workshop meets at the University of Waterloo, Ontario, Canada.

PUBLICATION

Papers accepted for presentation at the workshop will be published in a volume of workshop proceedings. This will be presented to participants at the workshop registration.

If you are interested in participating in this workshop, you are invited to submit a paper for presentation. Please send notice of your intention to submit a paper and a brief statement of the paper's main points by January 20, 1975 to:

Dr. Robin Williams
Program Chairman
IBM Research Division (K54/282)
Monterey & Cottle Roads
San Jose, California 95193, USA

CALL FOR PAPERS

Workshop on Computer Hardware Description Languages and Their Applications

September 3-5, 1975
New York City, New York USA

Increasing interest and attention have been given to the areas of computer-aided logic/system design and documentation. A good computer hardware description language (CHDL) can be used for describing, documenting, simulating and implementing digital systems. The 1973 and 1974 workshops have been held in New Jersey, and West Germany respectively. The Third Workshop on CHDL will be held the week before the 1975 COMPON (September 9-11, Washington, D.C.)

Papers in the following and related areas will be considered:

1. New concepts for description languages
2. Constructs and mechanisms in computer descriptive languages
3. Comparison of languages
4. Applications of languages in documentation, simulation and hardware implementation
5. Hardware compiler (translator) implementation
6. System simulation
7. Computer-aided logic synthesis, logic synthesizer implementation
8. Experience on using languages, simulators, translators and logic synthesizers
9. Analysis of Digital Systems
10. Different levels of CHDL
11. Future prognosis

Information regarding the Workshop can be obtained from:

Professor Stephen Y.H.Su
Workshop Chairman
Department of Electrical Engineering
The City College, City University of New York
New York, N.Y. 10031
Telephone: (212) 621-2392
(212) 621-2248 (secretary)

You are invited to submit 3 copies of the paper in 10 to 20 double-spaced typewritten pages to: Professor Donald L. Dietmeyer
Department of Electrical and Computer Engineering
University of Wisconsin
Madison, Wisconsin 53706
Telephone: (608) 262-3890, 262-3840

The deadling for submitting the papers is March 1, 1975. The papers will be reviewed by the referees.

There will be a session for presenting the most recent results. Authors who would like to talk in this session should submit two copies of the summary to Dr. Dietmeyer by June 1, 1975. Papers for this session will not be published in the Proceedings.

CAD 74, INTERNATIONAL CONFERENCE ON COMPUTERS IN ENGINEERING AND BUILDING DESIGN.

September 25-27, 1974
Imperial College, London, UK

Organized by Computer Aided Design

The proceedings are available from : IPC Science and Technology Press
IPC House
32 High Street
Guildford
Surrey GU1 3EW
United Kingdom

WEDNESDAY 25th SEPTEMBER

09.00 – 10.30

OPENING ADDRESS
Professor Sir Hugh Ford

D.O. ORGANIZATION AND DRAFTING
Chairman: B. Gott, CAD Centre

The computer's place in the drawing office
B. Gott, CAD Centre, Cambridge

3 PD3 – automated drafting and circuit design
R.F. Allum, U. Stockburger and M.K. Tod (Bell Northern Research, Ottawa)

15 A graphics database for engineering drawings
B. Bittner and R. Wolf (Xerox Corporation, El Segundo/ Rochester, USA)

14 Computer-aided drafting
C.B. Besant and A. Jebb (Imperial College of Science and Technology, London)

MORNING COFFEE

11.00 – 12.30

102 Computer-aided construction of technical illustrations
C.J. Richards (Lanchester Polytechnic, Coventry)

55 Computer-aided drafting in the bridge design office
I. Hamilton (CAD Centre, Cambridge)

123 The role of automatic digitizers in c.a.d.
C.M. Williams (Virginia Polytechnic Institute)

53 Multi-user digitizing systems in cartography
R.E. Grindley (Computer Equipment Co. UK)

EUCLID-II: a design and drawing language for engineers
L. Shaw (D-A Computer Services, Sheffield)

GRAPHICS
Chairman: To be announced

4 Techniques for improving the role of graphics in c.a.d.
R.F. Allum (Bell Northern Research, Ottawa)

8 Interactive graphics – an under-exploited design medium
M.D. Apperley (Imperial College, London)

44 Interactive graphics in civil engineering - portability and adaptability
R.H. Ewald, L.J. Feeser and R.L. Schiffman (University of Colorado USA)

10 Computer graphics in aircraft design and manufacture
P. Aughton (BAC, Bristol)

1 Computer graphics in the design and manufacture of node points in offshore oil rig substructures
S.A. Abbas (Teesside Polytechnic, Middlesbrough) & R.W. Davison (Whessoe Ltd, Darlington)

MANAGEMENT

Chairman: M. Sabin, BAC

87 Modelling the design process
G.L. Mallen (Royal College of Art, London)

89 Methods of improving consumer acceptance of c.a.d. programs
G.M. Mills (University of Bradford)

77 Decentralized development for design automation
T.M. Korelitz (The Badger Company, Cambridge, Mass. USA)

125 Security of data in c.a.d. systems – a cipher technique
D.L. Williams (Bell Northern Research, Ottawa)

LUNCH INTERVAL

14.00 – 15.30

CIVIL ENGINEERING
Chairman: to be announced

6 The design of portal frames by computer to strength and deflection requirements
D. Anderson (University of Warwick)

16 Computer-aided design and analysis of prestressed concrete bridge decks
D. Bond (Queens University, Belfast)

32 GENESYS: RC-Building 1. Computer design and detail of RC structures
A. Craddock (GENESYS Centre, Loughborough)

69 Highway and structural c.a.d. using a Cadmac graphics terminal
P.B. Jeffreys (Mott, Hay and Anderson, Croydon)

122 Monitoring graphic techniques in design
D.S. Willey and D. Yeomans (University of Liverpool)

64 The travelling road sweeper
H.B. Humpidge (Sheffield, England)

60 Interactive design of optical filters
A.C. Kilgour and W.D. Hay (University of Glasgow)

101 A computer program introducing size constancy psychological phenomenon in the plane representation of 3-dimensional objects
H.C. Reggini (Buenos Aires)

131 Computer animation
C. Yi et al (Imperial College, London)

HEAT EXCHANGE AND FLUID FLOW
Chairman: to be announced

19 Computer-aided design of heat exchangers in a service organization to industry
G.H. Cowan et al (HTFS, Harwell)

20 TASCI. A program for the design of shell-and-tube condensers
D. Butterworth and M.J.C. Moore (HTFS, Harwell)

47 Calculation of shell side pressure drop for segmentally baffled shell and tube heat exchangers by divided flow method
I.D.R. Grant (National Engineering Laboratory, East Kilbride)

30 Shell-and-tube exchangers with condensation on the shell side
D. Chisholm & C. Cotchin (National Engineering Laboratory, E.Kilbride)

AFTERNOON TEA

16.00 – 17.30

78 Application and development of computer-based systems for the public health engineer
V.P. Lane (General and Engineering Computer Services Ltd, Liverpool)

108 Computer simulation of lake networks
H. Sorvari (Teknillinen Laskenta, Helsinki)

91 Optimal design with fixed geometry
H. Multamaki (Teknillinen Laskenta, Helsinki)

99 The use of the computer in civil engineering design
M.R. Prince (Atkins Research and Development, Epsom)

OPTIMIZATION
Chairman: to be announced

62 NOC optima
S.E. Hersom (Hatfield Polytechnic, Herts)

45 Optimization as a design tool
R.M. Filmer (Building Research Establishment, Watford)

116 Application of optimization techniques in communications systems
M.H.E. Ward (Pye Telecommunications Ltd, Cambridge)

98 Sub-optimal solutions to combinatorial design problems
A.D. Pearman (University of Leeds)

33 The specification of circularly symmetric corrugated diaphragm configurations
D.L. Critten (Kent Instruments Ltd, Luton)

74 The thermal design of plate fin reboilers
T.D.A. Kennedy (HTFS, Harwell)

112 Mathematical modelling of radiant heat transfer
J.S. Truelove (HTFS, Harwell)

THURSDAY 26th SEPTEMBER

09.00 – 10.30

TEACHING C.A.D.

Chairman: To be announced

73 Education in c.a.d. for practising engineers
D. Rzevski and A.A. Kaposi (Kingston-on-Thames Polytechnic)

100 C.a.d. in university courses
R.E. Radley (University of Sheffield)

57 The teaching of c.a.d
D.W.H. Hampshire (Portsmouth Polytechnic)

83 The use of matrix manipulative scheme in the teaching of linear elastic analysis of structures
B.S. Lee (University of Newcastle)

9 Minimum cost allocations of dimensional tolerances – an optimization problem for courses in mechanical engineering design
G. Ashton and J. Ellis (University of Salford)

CHEMICAL ENGINEERING

Chairman: R.W.H. Sargent, Imperial College

115 Invited review:
J. Villadsen (Danish Technical University, Copenhagen)

63 The dissemination of research expertise into chemical engineering industry
D.G. Howes and M.E. Leesley (SCICON, Milton Keynes/CAD Centre, Cambridge)

127 A rational approach to distillation simulation
P. Winter (CAD Centre, Cambridge)

GEOMETRICAL/SHAPE/SURFACE DESIGN

Chairman: J. Hatvany, Institute of Applied Systems Analysis, Austria

105 Invited review. Numerical representation of shape
M.A. Sabin (BAC, Weybridge)

61 New shapes from bicubic splines
J.G. Hayes (National Physical Laboratory, Teddington)

71 Polynomial splines for both approximation and interpolation
E. Kantorowitz (University of Aarhus, Denmark)

58 Curve generation: a consideration of methods in relation to hardware implementation
D.W.H. Hampshire and R.C. Osbaldeston (Portsmouth Polytechnic)

MORNING COFFEE

11.00 – 12.30

BUILDING DESIGN

Chairman: A. Bijl, University of Edinburgh

75 PHASE: an interactive appraisal package for whole hospital design
D. Kernohan (ABACUS Strathclyde University, Glasgow)

86 Interactive building design
A. Main (British Steel Corporation, London)

40 Towards computer-aided design in a private architectural practice
J.A. Davison (Gollins, Melvin, Ward & Ptnrs, London)

97 From British research in the building sciences towards French implementation
P.T. Daniel and F. Pavageau (Flintshire County Council/SERI)

Chairman: to be announced

133 SPEED-UP – A computer-based system for the design of chemical processes
M.J. Leigh, G.D.D. Jackson and R.W.H. Sargent (Imperial College of Science and Technology, London)

13 Computer-aided design of a reducing elbow for a penstock
U. Cugini, S. Bertelli and A. Sartori (Institute of Mechanics and Machine Construction, Milan)

109 Finite element analysis, modelling and computer graphics
B. Spooner and D. Lawrence (Atkins Research and Development, Epsom)

34 Shape optimization of glass containers under external loads
M. Crochet (Unite de Mecanique Appliquee, Belgium)

MANUFACTURING AND N.C.

Chairman: M. Sabin, BAC

59 Invited survey: the use of c.a.d./c.a.m. systems in manufacture
J. Hatvany (International Institute for Applied Systems Analysis, Luxenburg, Austria)

39 Computer aid in wing design
R.S. Davies (BAC, Weybridge)

28 Cutting of two-dimensional rectangular plates
N. Christofides (Imperial College, London)

LUNCH INTERVAL

14.00 – 15.30

BUILDING DESIGN

37 An approach to computer-aided architectural design
B. David and V. Rivero (ENSIMAG, Grenoble)

111 A rationalized approach towards architectural design
R. Th'ng (ABACUS, Strathclyde University, Glasgow)

46 A cost advice aid for the building design team
A. Garnett, P. Purcell and P. Sampson (Royal College of Art, London)

82 An historical cost benefit analysis of a computer system for architectural design
W.R. Laxon and J.J. Lefevre (Informatique et Batiment, Paris)

CONTROL SYSTEMS

Chairman: to be announced

93 CADCUM – c.a.d. of control systems from UMIST
N. Munro (UMIST Control Centre, Manchester)

117 The role of minicomputers in computer-aided system identification
P.E. Wellstead (UMIST, Manchester)

38 C.a.d. of compensators for nonlinear control systems
R.H. Davis and E. Kraemer (Heriot-Watt University, Edinburgh)

41 An interactive program package for data analysis system identification and parameter estimation
M.J. Denham and L. Rigby (Imperial College, London)

65 Invited review. c.a.d. in shipbuilding
R. Hurst (British Ship Research Association, Wallsend)

26 An artificial intelligence approach to manufacturing automation
R.T. Chien and T.C. Woo (University of Illinois)

27 C.a.d. manufacture of mechanical components using Simplified Cutting Sequence
M.H. Chaudbury (Ferranti, Edinburgh)

42 Computer-aided pipe production system
B. Dodd (Imperial College, London) and J.B. Jack (British Ship Research Association, Wallsend)

AFTERNOON TEA

16.00 – 17.30

88 CASH – a computer-aided design tool for housing
K. Mathur (ABACUS, Strathclyde University, Glasgow)

80 An experiment in computer-aided architectural design
J. Lansdown (Turner, Lansdown, Holt & Ptnrs London)

56 An integrated architectural c.a.d. system
A.D. Hamlyn et al (Imperial College, London)

48 Software structures for computer-aided building
J.C. Gray (Applied Research of Cambridge)

106 Automated architectural drafting system
K. Sakakibara (Nikken Sekkei Co. Tokyo)

49 Interactive graphics in the design of nonlinear control systems
J.O. Gray and P.M. Taylor (UMIST, Manchester)

36 Classical control system design by interactive computer-aided methods
K.C. Daly (Imperial College, London)

124 C.a.d. of industrial control schemes
D.C. Williams (Warren Spring Laboratory, Herts)

118 Modelling and control of an inverted pendulum using bond graphs
B.A. White et al (UMIST, Manchester)

68 The application of c.a.d. techniques to machine tool design production and manufacture
A. Jebb et al (Imperial College, London)

121 The use of contours as an interface between c.a.d. and c.a.m.
D.G. Wilkinson (National Engineering Laboratory, East Kilbride)

84 Interactive computer graphics in shipbuilding/AUTOKON
F. Lillehagen (Central Institute for Industrial Research, Norway)

FRIDAY 27th SEPTEMBER

PROJECT MANAGEMENT

Chairman: J. Hatvany, IISA, Austria

79 Management problems of establishing computer services within the design office
V.P. Lane (General and Engineering Computer Services, Liverpool)

22 The PD3 system – project and document control
P.M. Gray and I.F. Carlisle (Bell Northern Research, Ottawa)

31 Information requirements of a design and consultancy organization
L.B. Cousins (HTFS, Harwell)

96 Computer-aided concept, design and production – a look at systems solutions
R.K. Oatman (Computervision Cpn. Amsterdam)

129 The role of human factors in the development of c.a.d. systems
J. Wood (Royal College of Art, London)

STRUCTURAL ANALYSIS

Chairman: to be announced

126 Invited survey: structural analysis programs
G.M.J. Williams (Scott Wilson Kirkpatrick & Ptnrs, London)

52 INGA – an interactive graphic system for structural analysis
I. Grieger (University of Stuttgart)

7 Computer-aided analysis and design of steel frames
L.J. Feeser et al (Rensselaer Polytechnic Institute, New York)

18 Integrated structural design suite for minicomputer
R.J. Cope and J.H. Bungey (University of Liverpool)

MECHANISMS AND COMPONENTS

Chairman: P.J. Grant, Imperial College

128 SPREAD 3 – finite element system for design and analysis of three-dimensional flat springs
J.R. Wolberg and Y. Glazer (Technion, Haifa)

113 Computer-aided engine cam design
A. Sarsten (University de Sherbrooke, Quebec) and H. Valland (Institutt for forbrenningsmotorer, Trondheim, Norway.

76 Program package for large deflection analysis of thin rods and their assemblies
M. Konopasek (UMIST, Manchester)

MORNING COFFEE

BUILDING DESIGN

Chairman: T.W. Maver, Strathclyde University

104 Computer-aided building for Oxford method
P. Richens (Applied Research of Cambridge)

24 Progress in the development and evaluation of CEDAR a computer-aided building design system
J. Chalmers et al (Property Services Agency, Croydon)

81 Grafting interactive graphics onto an existing batch system for architectural design
W.R. Laxon (Informatique et Batiment, Paris)

35 Building design
D.M. Curtis (Oscar Faber and Ptnrs, St. Albans)

94 Interactive finite element data generation using the substructuring technique
H.A. Nasreldin (University of Leicester)

23 The development of complex structures in the communications industry
A.T. Humphrey and P.D. Carter (Marconi Co. Chelmsford)

29 Repeated stiffness analysis on a mini-computer
R.J. Cope and A.G. Sinkinson (University of Liverpool)

95 LUPIM – an interactive parametric 2-D – mesh generation system
I.A. Nazlawy (formerly University of Leicester)

ELECTRICAL AND ELECTRONIC ENGINEERING

Chairman: to be announced

50 Circuit analysis programs in Europe and the USA
J.R. Greenbaum (General Electric Co. Syracuse)

5 An algorithm for the realization of a ternary logical function
R. Amer (Cairo University)

51 Third generation CAD programs – implications and applications
J.R. Greenbaum (General Electric Co. Syracuse)

132 RENDIS – a universal program system for CAD of switching circuits
H.E. Zander, Academy of Sciences, German Democratic Republic)

LUNCH INTERVAL

12 The monitoring of service networks
R. Baxter and R. Brady (University of Cambridge/Applied Research of Cambridge)

70 Computer optimization for studying summertime temperatures in buildings
R. Jones (Building Research Station, Watford)

43 Structure evaluation in a closed building method
H.M. Evans (Applied Research of Cambridge)

66 Optimization of design parameters for refrigerated dehumidifiers using iterative simulation on a digital computer
A. Iqbal (Atkins Research and Development, Epsom)

DISPLAYS

Chairman: F.E. Taylor, National Computing Centre, Manchester

54 Invited survey: interaction with visual information
D.J. Grover (NRDC, London)

110 HRD-1: a high resolution storage display
P.A. Woodsford and G.S.B. Street (Laser-Scan Ltd, Cambridge)

119 A microprogrammed processing module for displays
D.J. Woollons and P.W. Whiting (University of Sussex, Brighton)

RELIABILITY

Chairman: to be announced

2 Reliability assessment in computer-aided design
R.N. Allan (UMIST, Manchester)

72 Using c.a.d. for reliability improvement
A.A. Kaposi, C.D. Partridge and B.A. Lowrey (Kingston Polytechnic, Surrey, and REX Thompson and Partners)

17 A program to calculate mean time between failures
A.H. Boyce and E.D. Link (Marconi Research Laboratories, Great Baddow)

AFTERNOON TEA

67 Optimized silencing of high velocity air conditioning ducts
A. Iqbal and T.K. Willson (Atkins Research and Development, Epsom)

90 BIBRACS – an integrated computer system for architectural use
D.H. Mountford (Brockhouse Steel Structures Ltd, West Bromwich)

25 Structural engineering by computer
C. Chapman (Building Computer Services, Bristol)

85 STATSYS – structural analysis system, an application in CAD
P.M. McClintock et al (Imperial College, London)

130 System for assessment of aircraft vision envelopes
T. Wooderson (BAC, Weybridge)

Displays in medical physics
A. Todd-Pokropek. D. Ingram and J. Clifton (University College Hospital, London)

107 Man/computer collaboration in the design process
H.T. Smith (University of Sheffield)

SEMICONDUCTOR DEVICES

Chairman: to be announced

21 C.a.d. in semiconductor device problems
G. Pierini (CISE, Milan)

120 Design of IMPATT diodes
S.R. Wilbur and D.L. Bates (University College, London)

114 The use of graph-theoretical methods for integrated circuit design
W.M. vanCleemput (University of Waterloo, Ontario)

103 Large-signal transistor modelling
G.A. Richards (Marconi Research Laboratories, Great Baddow)

INTERNATIONAL CONFERENCE ON COMPUTER AIDED DESIGN

April 8 - 11 , 1974

University of Southampton , U.K.

Organized by the Institution of Electrical Engineers and
the Electronic Engineering Association

The proceedings are availbable from : IEE
Station House
Nightingale Road
Hitchin, Herts SG5 1RJ
United Kingdom

Page No. 37 *C. Ll. Thomas*

'THREE-D' a digital computer code for the design and analysis of three dimensional electrostatic fields

117 *H. R. Tordhol*

ELDAK—an integrated computer-aided design system for electronics

12 *E. A. Warman*

Experiences with a simple computer based drafting system

212 *G. A. Watts and C. I. Lambe*

Computer aided design of extrusion dies

123 *P. E. Wellstead, D. Galley and R. Koreman*

Interactive computer aided identification of engineering systems

85 *R. J. Wilde, J. T. Boardman and D. Farquhar*

A CAD system for distribution-network planning

200 *M. T. Wright and S. W. F. Waters*

Computer aided design of eddy-current couplings for optimum torque-speed characteristics

SOME GOOD ADVICE FROM A COMPUTER PIONEER

Following is the statement by Saul Gorn at the opening session of ACM 74 upon receiving the ACM Distinguished Service Award. While it will appear in Communications we would appreciate your printing it in Chapter and SIG Newsletters.

"When Franz Alt told me I was to receive this award, I was, of course, very pleased to be put in the same class as my predecessors, Franz Alt himself, Don Madden, George Forsythe, and Bill Atchison.

"It seemed to me, however, to be somewhat paradoxical. There are hundreds of us who keep our professional society moving in its more than two dozen special interest groups, and in its half dozen boards and their standing committees, let alone the work of the council and the editorial duties in the publications. Mostly we do these things because we would be unhappy to see those activities bog down. Questions of reward or fame have much less to do with our decision to pitch in than one might think. I believe that by and large most of us would continue these activities even if we were certain to remain anonymous.

"If, therefore, you have decided to make me less anonymous, it must be that you want me to be seen as representative of these efforts.

"This I am proud to do.

"To those of you who have been wondering whether you should pitch in, I say that there is great satisfaction to be gained in so doing. Don't hesitate anymore, but come and join us.

"Thank you again for this honor."

MPACT

Microprocessor Application to Control-firmware Translator

by

Mathew N. Matelan

The following draft describes proposed research to be performed by the author, who is working toward a Ph.D. in Computer Science at the Institute of Technology, Southern Methodist University.

Because of the recent widespread interest in microprocessor software design aids, this work may be of interest to SIGDA members. Reactions to and comments concerning the proposed research would be appreciated.

The author is currently with the Design Automation Project, Electronics Engineering Department (L-156), Lawrence Livermore Laboratory, P.O. Box 808, Livermore, California 94550.

INTRODUCTION

The use of digital computers in process control has long been an important aspect of computer technology. Further, many computer system functions not usually associated with the engineering field of process control may be cast in terms of controlling: responding in a timewise correct manner to a set of unpredictable conditions. This more general view of control and controlling will be referred to as the control problem.

It is the purpose of this proposal to outline a general method of describing the control problem and a solution. The complexity of actual control problems suggests the development of a simplifying model which abstracts the essential properties of control. From such a model, a system may be defined to solve control problems in a consistent way. The scope of such a system is very large. The feasibility of the concept, however, may be shown by developing a system tailored to a specific class of general purpose processors (called microprocessors) that have architectures and uses somewhat less complex than other classes of processors.

One of the latest advances in computer hardware technology, and one with potentially far reaching effects, is the introduction of the microprocessor - the "computer on a chip". A microprocessor may be loosely defined as from one to ten large scale integrated circuit chips capable of performing functions consistent with the needs of general purpose computing. Flexible instruction sets in microprocessors are causing them to be chosen instead of more expensive minicomputers in many application areas.

The current state of the microprocessor industry is that of intensive hardware development coupled with software development that is either minimal or left entirely to the user. There is immense potential for small general-prupose computing hardware in applications from medical sensors to powerful distributed processing systems. However, many observers have noted delay and hesitation in their use caused by arcane programming requirements and limited availability of software aids. This situation offers an opportunity for developing a system which could reduce the design engineer's workload in integrating a microprocessor into a control oriented application system, while validating the control problem model.

PROPOSAL

A software system is to be developed to allow the definition of microprocessor behavior in a control application. The MPACT (Microprocessor Application to Control-firmware Translator) system is to be designed to produce a complete microprocessor control firmware program according to definitions of the microprocessor environment and related behavioral characteristics.

The use of such a system might best be illustrated by an example. Consider a printed circuit board with components capable of realizing several useful functions, depending on the sequences of enable pulses issued to them and the interpretations given the outcomes of their actions. The use of a microprocessor to control sequencing and external signal interpretation is becoming a common method of realizing the diverse capabilities of otherwise standard hardware. This approach

often affords substantial reduction in the overall hardware on control boards due to the elimination of special purpose sequencing circuits. A similar reduction in spare part inventories, fault isolation time, and complexity of repair procedures may also be achieved through judicious use of microprocessor based assemblies. The source of these savings is the commonality of boards. Physically identical boards may at times be used for diverse functions, the only difference being the control programs used to sequence and interpret intra-component activities.

The benefits of such an approach are many, but the results of using microprocessors in actual practice have been, in many instances, disappointing. Recent contacts with microprocessor user groups [C11] reveal that software production is the major cause of product development cost and time overages, often to the point that microprocessors fail to be competitive with special purpose hardware. At present, control programs are generally produced as needed for particular applications with little consistency, other than the prejudices of individual programmers. The MPACT system would formalize the various elements of control programming and give the system designer a method of expressing a problem in a structured way based on those formaizations. From that information the proposed system would produce a complete control program.

The steadily increasing power and decreasing cost and size of the microprocessor indicates that it will be used in large numbers in high volume product lines. It seems likely that the majority of these products will involve the active control and monitoring of processes, increasing the importance of producing control-oriented programs. It is for this reason that a system to facilitate the production of microprocessor control programs was chosen as the vehicle for studying the general problem of control.

BACKGROUND SURVEY

Creation of a system for the automatic production of microprocessor control code requires a background in both software and hardware. An appreciation of the types of available microprocessor architectures is essential in forming a general method of describing them. Likewise, familiarization with a number of applications is necessary in order to isolate the essential properties of the control problem as it relates to microprocessors. Then, to realize this knowledge as a useful design tool, a background in software development procedures is needed.

The design of computer related languages is treated in several chapters of Gries [H2], Harrison [H3] and Lee [H5]. Aho [F1] is a survey of formal grammer specifications. The design and implementation of compiling systems are examined in varying detail in the texts of Appendix H.

Special problems associated with compiler-compiler design (such as meta-language specification) are discussed in Feldman and Gries [E1] and Feldman [E2], while the remaining references in Appendix E (such as Lyon [E5] and Matelan [E7 and E9]) concern various techniques. A number of articles related to language design, and especially non-procedural programming languages, is included in Appendix A; why this is needed will become evident in subsequent sections. Leavenworth and Sammet [A14] is a survey of this topic.

Parsing and storage techniques are reviewed in the articles in Appendix F, while code optimization is the subject of the papers in Appendix G. Appendix B is a list of current papers of general information concerning microprocessors (Callaghan [B3], Metzer [B11], Terrero [B18], and Waaber [B19] are surveys). Also included are papers concerning applications areas: Baskin [B1], Bell [B2], Cooper [B4], Fuller [B7] and Matelan [B10] on possible distributed architectures; Lee [B9] and Runyun [B15] on measurement devices; Whitney [B21] on calculators; and Cushman [B5], Parasuraman [B12] and Smith [B17] on the control of various processes. See also Weiss [C6] and Holt [D5] for discussions of applications with examples.

Appendix C lists recent papers covering microprocessor software, while Appendix D is concerned with hardware. The disparity in these categories is indicative of the more intense effort seen in the hardware area. Papers in these two appendices contain details of hardware and software for specific microprocessor types; see the papers in Appendix B for a more general treatment. A list of texts on digital hardware principles appears in Appendix I. Appendix J includes texts and articles on the nature of time and techniques used to describe it. Time and timing is central to any discussion of control.

DESCRIPTION OF THE WORK

Application Areas

The application areas to which microprocessor technology may be applied are literally unlimited. The technology is very young, but it has already seen use in:

- Calculators
- Terminals
- Measurement devices and systems
- Process control systems
- Medical monitoring and feed-back control
- Computer peripherals and high-capacity channels.

Such uses will continue and their number will increase. Other applications likely to be developed in the near future include:

- Replacement of combinatorial nets
- Artificial intelligence applications
- On-site preliminary data reduction
- Built-in test equipment
- Security
- Credit card verification
- Precise timining controls
- One-of-a-kind devices.

A list of this sort is never complete, being limited only by the skill of the applications engineer in configuring a microprocessor (through its control firmware) to his needs.

There are at least two other important microprocessor application areas. The first of these is the development of standard interfacing techniques and hardware. Much of computer technology is concerned with controlling or communicating with external devices. Frequently, the program needed to perform such a function is relatively trivial compared to the task of interfacing the processor to the peripheral device. The expense involved in engineering these interfaces is often a substantial part of system development costs. The microprocessor offers a means of facilitating interface design in many applications (essentially those not concerned with high transfer rates).

This technique uses a microprocessor in an interface unit or even in the interface cabling itself [D12] to perform data transformations in real time. This method could reduce hardware requirements considerably and force a measure of standardization in interfacing. For example, the mismatch of families of logic used in the controller and the controllee of an application is a common problem. A set of microprocessor based interface units can be designed with communications latches constructed to match the electrical properties of common logic families, in effect standardizing many applications. The same approach would apply to time-related controller-controllee mismatches.

However, a problem remains: each interfacing problem requires an application-dependent control program for the microprocessor involved. The proposed MPACT system is potentially capable of supplying these programs in a cost effective, completely documented form.
A second use of microprocessors is their aggregation to form large powerful distributed processor computing systems. In certain applications, readily available microprocessors might be interconnected so that their combined power would exceed that of the independent processors. If such cases exist, aggregation would allow computing devices (configured to special purpose applications, if needed) to be assembled at a cost lower than possible with current technologies.

The feasibility of such an approach is directly related to the effort required to produce efficient control firmware for each of the micro-

processors involved. This firmware would not only define individual microprocessor duties, but also specify the number and format of valid interactions allowed in the aggregation. The production of control firmware of this type might be facilitated by a fully developed version of the MPACT system.

A Generalized Model of the Control Problem

Although a large number of application problems in computer technology require a control-oriented solution, such solutions are usually produced as though all elements of the problem were completely new. This lack of consistency (and concomitant arduous development) could be reduced by generalizing the problem as much as possible. A useful tool in this regard would be a model abstracting the essential elements of the control problem.

The primary difficulty in specifying control problems is the complexity introduced by the time element. According to Webster's Third New International Dictionary: timing is "...a selection for maximum effect of the precise moment for beginning or doing something...the art or practice of regulating...observation of the elapsed time of an act, action or process". All control problems directly involve keeping pace with a clock, be it real-time or artificial (i.e., timing). A technique used in the physical sciences to isolate less complex forms of a time domain problem is that of variable separation (for example, Halliday [J5]). We therefore consider the control problem as consisting of two co-variant parts which may, however, be analyzed separately: a timing-independent part and timing-dependent part.

The timing-dependent part of a control situation consists of an enumeration of all relevant conditions which may eventuate at the controller's interface, accompanied by the responses which would be appropriate to the particular application. Each condition (or group of related conditions) and its corresponding reaction is called a sentence. "Interface" as used here is meant to include the communications registers and latches through which the controller receives information concerning its environment. Such information may be considered a definition of the current state of the controllee. However, in many control situations, the past

history of the controllee is also important. The problem-relevant portion of this information, which could be recorded in counters, buffers or registers of the controller, may be considered an encoded state history of the controllee. Therefore the term interface will encompass information which describes both current and previous controllee states needed to define an appropriate controller strategy.

The term "strategy" also requires clarification. It is more than an alternative to "procedure" as a description of the function of a control program. A controller is not a program in the usual sense of the word: a method for calculating a result predicated on initial values. It may be viewed as a referee, determining a time-wise correct response to competing service requests which present themselves at its world interface. (Perhaps the most pressing of these interface demands is that of the system clock.) Under this view, a control program represents a strategy with which to meet real-time contingencies rather than a procedure to be pursued to a result.

A strategy is a prototype, a static specification without timing or sequencing constraints. The interface conditions it expresses are not bound to be tested in a particular order; no ordering is intended by their positions in the strategy. The implementation and interpretation of condition testing is the domain of the timing-dependent part of the control problem. A strategy is therefore not a flow model since there is no flow implied. The collection of strategy sentences outlining a control problem is an aggregation of possible contingencies that are to be recognized. Strategy is symbolized in the following discussion by the letter S.

The timing-dependent part of a control situation has three aspects. Two of these aspects are essentially different methods for expressing the same timing consideration: when is a strategy sentence condition to be considered valid. The first aspect is a consideration of condition test intervals. Condition test intervals are of the form: 1) what is the smallest interval of time that must have elapsed between valid tests

for a condition, or 2) what is the maximum interval that is allowed before a valid condition may have been overlooked. Condition test intervals are symbolized by the letter C.

The second aspect is the specification and interpretation of interrupt signals (represented by I). Interrupts are viewed as asynchronous demands for response. They are not as obviously timing-dependent as are condition test intervals, but their inclusion in a strategy would invalidate the strategy's requirement for complete timing-independence. It should be stressed that the interrupts discussed here correspond to hardware features that, on receipt of a control signal, divert the attention of the computer from the current program (controller), to a specific address related to that control signal. Interrupts are timing dependent as viewed from the control application, because the processor resources needed to service them obviously impact other temporal aspects of the problem solution.

The third aspect of the time-dependent part is that of priority. Just as condition test intervals and interrupt processing overhead must be considered to determine the timing relationships of the application outlined in a strategy, so must the relative importance (in a time-response sense) of the various conditions and responses be considered. This relative order is called priority and will be referenced by the letter P.

The four features of control thus define necessary and sufficient elements needed in constructing a set-theoretic model of the control problem. They succinctly describe the dual nature of control timing.

The model consists of three sets of information (or "specifications") and a transformation which maps an ordered pair from a product set (formed from two of these sets) into the third set. It is a significant feature of this model that the elements of each set are of the same form and are completely described by the four features of control (Strategy, Condition, Interrupt and Priority).

The Control Program Model (**CP**) may be expressed as a quadruple:

$$\mathbf{CP} = (\mathbf{A},\mathbf{P},\mathbf{C},\Psi) \qquad \text{[Eq. 1]}$$

where

- **A** is that set of all possible applications of the control type. The set **A** is known as the Problem Set; the subset a of **A** is a particular application.
- **P** is the set of all possible processor capabilities adaptable to an arbitrary control application. The set **P** is known as the Processor Set; the subset p of **P** is a processor capability specification. Several p subsets may be needed to define the capabilities of particular piece of hardware.
- **C** is the set of all possible control programs expressed in symbolic form. The set **C** is known as the Program Set; the subset c of **C** is a particular control program.
- Ψ is the transformation from the product set of **A** (the Application Set and **P** (the Processor Set) into **C** (the Program Set):

$$\Psi : \mathbf{A} \times \mathbf{P} \rightarrow \mathbf{C} \qquad \text{[Eq. 2]}$$

or, for a particular application one may write:

$$c = \Psi(a,p) \qquad \text{[Eq. 3]}$$

Ψ characterizes the timing analysis that must be applied to the sets **A** and **P** in order to arrive at a control program in **C**.

The three subsets a, p, and c may be internally divided into pairs, each subset having a timing-independent part and a timing-dependent part. This double may be conveniently expressed in terms of Strategy, Condition, Interrupt, and Priority. The transformation, Ψ, must map both parts to determine a control program, c. It may be thought of as acting on these parts independently for convenience. However, the transformation of a control problem specification is a melding of many timing-dependent and timing-independent items. The establishment and maintenance of a birpartite perspective is a simplification of an extremely complex inter-relationship.

Substitution of the four previously specified set elements in the original control problem definition [Eq. 1], for a particular application, yields

$$\mathbf{CP}_{pa} = \left(\left\{ S_a, (C_a, I_a, P_a) \right\} \ , \ \left\{ S_p, (C_p, I_p, P_p) \right\} \ , \ \left\{ S_{pa}, (C_{pa}, I_{pa}, P_{pa}) \right\} \ , \ \psi \right). \quad \text{[Eq. 4]}$$

where

$\left\{ S_a, (C_a, I_a, P_a) \right\}$	is a particular application specification (specific subset a of **A**, Eq. 1)
S_a	is the timing-independent element of subset a
(C_a, I_a, P_a)	are the timing-dependent elements of subset a
C_a	is the condition test internal specifications of subset a
I_a	is the interrupt type specification of subset a
P_a	is the priority level specification of subset a
$\left\{ S_p, (C_p, I_p, P_p) \right\}$	is a particular processor capability specification (specific subset of **P**, Eq. 1)
S_p	is the timing-independent element of subset p
(C_p, I_p, P_p)	are the timing-dependent elements of subset p
C_p	is the condition test interval specification of subset p
I_p	is the interrupt type specification of subset p
P_p	is the priority level specification of subset p
$\left\{ S_{pa}, (C_{pa}, I_{pa}, P_{pa}) \right\}$	is a particular control program specification for a specific processor and application (subset of **C**, Eq. 1)
S_{pa}	is the timing-independing element of subset c
(C_{pa}, I_{pa}, P_{pa})	are the timing-dependent elements of subset c
C_{pa}	is the condition test interval specification of subset c
I_{pa}	is the interrupt type specification of subset c
P_{pa}	is the priority level specification of subset c

The expansion of the control problem quadruple [Eq. 1] into the temporally bipartite form [Eq. 4] illustrates the basic similarity of the three information sets which are involved in control program production. Note that the elements of each of the three sets are of similar form; each contains a timing-dependent and a timing-independent part. This result is significant. The complexity of the problem has been reduced not only by the initial time-wise bifurcation, but also by the recognition that this technique is applicable to each of the information sets which are needed to describe it. Such consistency may be exploited in realizing the model as a system.

Implementing the Model

The perspective afforded by the Control Problem Model (**CP**) greatly simplifies the task of conceptually defining a software system to produce symbolic control programs. The model defines concisely the major system elements which must be specified and their inter-relationships. The model is expressable as a quadruple, therefore it seems reasonable to expect that a system to realize the model should consist of four major sections.

The first constituent of the quadruple is **A**, the set of all possible control applications. In terms of a software system, **A** is that set of application programs expressable in some programming language. Returning to Equation 4 it is possible to determine the components that would be required in such a language. It would, of course, need to preserve the partitioning of timing-independent and timing-dependent components which is central to the model.

In order that a system be profitably used, the language in which the application is described must be as straightforward as possible. Toward that end, a language with English-like syntax and a non-sequential structure is proposed. A number of benefits accrue from such an approach:

1) The "code" written to describe the application forms a basis for the documentation of the firmware produced;
2) The syntax is more easily learned;
3) Expressing an application in an English-like format forces the engineer to consider in detail the function and goals of a design in terms approaching those required in readable documentation.

Perhaps the most important aspect of English-like language coding is that of self-documentation. The lack of a structured program of design documentation has led many projects into overruns. Even on one-of-a-kind developments, the extra time consumed by a wordy functional description will usually reap rewards during system integration due to self-explantory text and the existence of a well thought out control strategy. A strategy section of the language is, therefore, defined as the vehicle used to specify the timing-independent component of the control application set.

The specification of the timing-dependent component of the application seems to lend itself naturally to either a tabular format or decision tables. Sections are included in the application programming language to allow listing of the three timing-dependent subelements (Condition, Interrupt, and Priority) in a convenient manner, one which will permit these specifications to be referenced symbolically.

The second constituent of the Model is **P**, the set of all possible processor capabilities. Here again, a software system requires that a language be defined for specifying the disparate data involved in a behavioral processor description. The language components indicated by the model as necessary for the description task are identical in concept to those adumbrated for the application set, **A**. Timing-independent elements and timing-dependent elements must be defined. This correspondence allows the same general language structure to be used in defining both the application and the processor. It must be recalled, however, that while elements of the descriptions are equivalent in intent, they differ in content and interpretation. The natural language concept of dialect will be used to indicate forms of a language which have much in common, but are not the same.

The language used to describe both sets **A** and **P** will be referred to as the MPACT language. The application-oriented (user) dialect has been outlined. The processor-oriented (library) dialect is similar in structure, containing much common terminology. It must be remembered that the dialects describe different set elements, however, and that their fine structure and interpretation will often be different.

The third constituent of the model is **C**, the set of all possible symbolic control programs. From the model's view, the purpose of transformation is to locate from among the various control programs in **C**, that one which satisfies the needs of the application and the constraints of the processor. A software system must take a different interpretation of the transformation if it is to solve the problem presented in a finite amount of time. The system must generate a control program that is equivalent to the one into which (a,p) would have been mapped by ψ.

Code generation assumes an *a priori* structure to which the control program will adhere. As it was with the design of the MPACT language, the expanded form of the model [Eq. 4] indicates a convenient structure.

Note that the elements of c, subset of **C**, are again divided into a timing-independent part and a timing-dependent part as was done in the language. This division is maintained in the structure of the control program; the parts are referred to as the Driver Group and the Monitor.

The last constituent of the model quadruple is ψ, the transformation. In order to determine its systemic counterpart, recall equation 2:

$$\psi: \mathbf{A} \times \mathbf{P} \rightarrow \mathbf{C}$$

and analyze it piecewise. The set **A** is all possible control applications, while **P** is all possible processor capabilities, each expressed as a time-dichotomy. The product set, $\mathbf{A} \times \mathbf{P}$, comprises all possible pairings of the elements of **A** and **P**. That is, $\mathbf{A} \times \mathbf{P}$ is all applications paired with all processor capabilities. A specific ordered pair (a,p) of application and processor capabilities must be selected from the product set. The selection is based on the application specification and the particular processor desired for the implementation. These two items are supplied by the user. The selected pair is then mapped into c (a particular control program) which is a subset of **C**, all possible control programs.

The specification of (a,p) as an ordered pair is not accidental. Such an ordered pair may be defined by

$$(a,p) \equiv \left\{\{a\} \ , \ \{a,p\}\right\} \qquad \text{[Eq. 5]}$$

This indicates that the selection of the particular ordered pair mapped by ψ into c is predicated on a, a specific application. This is sensible in that processor capabilities should be selected for their ability to solve the application problem specification, and not contrariwise. The particular capabilities examined by the system are limited to a user defined specific processor. This divergence from the model is for the sake of system efficiency.

Recall now Equation 3, the transformation of a specific ordered pair into a specific control program:

$$c = \psi(a,p) \quad .$$

It is just such a particular set of specifications that a software system would have to process. It is necessary therefore to express ψ in terms of the system modules required to perform the transformation (or, in system nomenclature, translation).

Consider the constituents of Equation 3 in the non-rigorous terms of the MPACT system:

a	is a specific application expressed in the user dialect of the MPACT language.
p	is a specific processor description oriented toward control applications, expressed in the library dialect of MPACT language.
(a,p)	is an ordered pair consisting of an application and the processor capabilities necessary to perform it.
$\psi(a,p)$	is the timing analysis of the pair (a,p)
c	is the control program generated by the MPACT system to solve the application problem a in the symbolism of a programming language of processor p.

In the system as proposed (see Figure I), the acquisition of the processor set **P** is done by a library processor. The acquisition of the application set **A** is performed and lexically scanned by an input routine. The product set is formed by a selection process that is two-fold. A parser classifies each sentence of the application set **A**, followed by a library scan that selects the appropriate processor capabilities of the processor set **P**. This forms the specific ordered pair, (a,p), describing the problem. The transformation ψ then maps (or "integrates") the diverse packets of information which constitute the pair into a valid control problem, c. This transformation, in broad terms, characterizes the timing analysis of the system.

The Language - Specifying Sets **A** and **P**

The basic organization of an application definition might follow the usual form of a technical report. Major topics would be assigned to sections composed of paragraphs containing sentences. Sections which might be required include:

Identification Section - The Identification Section defines the user, the application, the microprocessor to be configured, the revision number and date, and any other information needed to insure configuration control and integrity.

Interface Section - The Interface Section defines the preferred nomenclature of the application in terms of the chosen microprocessor's standard interface terminology.

Timing Sections - The Timing Sections determine the timing constraints and levels of importance attached to various interface ports, flags and functions (i.e., C_a,I_a,P_a).

Strategy Section - The Strategy Section defines in terms of the interface interactions and responses, the behavioral characteristics of the microprocessor in the context of the application (i.e., S_a).

Such a language would allow a high degree of attribute assignment via defaults, which would be fully documented in the system output.

The system suggested here not only requires that each application be well defined, but that the microprocessor also be defined in such a way that control code may be generated to accomplish the application's

demands. In order to maintain consistency, the same language would be used in both types of definition. However, due to the disparity of the definition tasks (that of functional, interface-oriented applications versus code-level, hardware-dependent capabilities), two dialects of the language are required: the user dialect and the library dialect.

The major features of the user dialect were outlined in the various section descriptions above. The library dialect follows the general organization of sections, paragraphs, and sentences discussed earlier, but with added features needed in describing a generalized microprocessor.

The term "library dialect" is used to indicate the logical division between defining an application and defining the agent of the application, the microprocessor. It is intended that a "library" of microprocessor descriptions, coded in the library dialect, be maintained by those engineers most knowledgeable in the techniques associated with each microprocessor. Then when an applications engineer determines and codes his application strategy in the user dialect, the firmware program produced will be similar to that produced by the more competent engineer, multiplying his experience and capabilities. The correlation of application to firmware provided by the overall consistency of the system allows the application engineer to follow the output control program code defined by him at a higher level.

The library dialect might also contain identification, interface terminology, and timing sections as seen in the user dialect. There would probably exist, however, many strategy sections; their usage constitutes the main difference in the dialects. The strategy sections would contain a statement syntax template and a corresponding code section skeleton for each language element defined for the user of a particular microprocessor. Templates may define sentences, phrases or words. Skeletons may define in-line (table driven) pure code or routines. Skeletons may be expressed in any symbolic language, but the assembly level will probably be that most often used, due to the multiplicity of microprocessor architectures to be encompassed in this generalized system.

The Control Program - Specifying Set **C**

A result obtained from modeling the control problem is the realization that the timing duality found useful in defining the application and processor may be maintained in the specification of the control program (the primary output of the MPACT system). All programs generated by the system should (for the sake of consistency) be based on a single structure. This structure is conceptually divided into two parts: the timing-independent part (or "driver group", S_{pa}) and the timing dependent part (or "monitor"; C_{pa}, I_{pa}, P_{pa}).

The monitor consists of three sections: the sequencer, the condition test block, and the referee. The sequencer controls the time of test of each possible condition listed in the second section, the condition test block. Each time a conditon is (by some internal interpretation) true, an "event" is said to have occurred. The occurrence of an event requires an action. Actions are resident in the timing-independent part of the control program called the driver group. Each action is defined by a specific member of the driver group called a driver. The third section of the monitor is the referee, designed to respond dynamically to interrupt conditions, and perform conflict resolution based on priority information.

A driver performs duties needed to respond to a control event. Each driver is subdivided into safe blocks. A safe block is defined to be a unit of code with a maximum execution time of sufficient brevity to allow deletion of condition testing during its execution. This is not equivalent to non-interruptable code blocks sometimes seen in operating systems. Safe blocks are not code sections which must always be executed to completion; they are code sections which (due to the timing of the application) may he allowed to execute without fear of causing timing errors due to ignored events. Interruped drivers have their remaining safe blocks scheduled by the referree, while single safe block drivers require no further monitor action until their next invocation.

The structural correspondence between the languages (used in defining applications and processor capabilities) and the generated control program simplifies and unifies system documentation, while easing software-hardware integration.

The System - Specifying $\psi: \mathbf{A} \times \mathbf{P}$ Algorithmically

The Microprocessor Application to Control-firmware Translator (MPACT) System is divided into six major modules: the librarian, the scanner, the parser, the selector, the integrator, and the formatter (see Figure I).

The librarian acquires the microprocessor definition, stores it as configuration data, and catalogs it for efficient access. Previously processed specifications may be directly inserted into the library data area from the microprocessor library data base.

The scanner and parser work together, reading and processing each user supplied sentence until all have been examined. The scanner acquires a strategy sentence and performs a lexical scan. The parser parses the result in association with template data in the library area. Following application acquisition, the selector determines the skeleton data to be used and records it.

After all selector data have been gathered, the integrator analyzes the application strategy and the timing constraints in terms of the selected processor capabilities. The result of this analysis is either a control program solution, ready for the formatter, or a message to the user that the timing constraints specified are unrealizeable (due to inadequate processor rates or ambiguous strategies). Some optimization and housekeeping chores may be performed by the integrator (a reason for the symbolic notation in skeleton definitions).

The formatter arranges the skeletons in a useable order and outputs both the control firmware and documentation.

SUMMARY

The proposed system would be used to demonstrate the feasibility of generating a complete, automatically documented firmware package capable of configuring a microprocessor for control applications. Such a demonstration would provide validation of the Control Problem Model.

The benefits of a firmware generator of wide applicability are numerous:

- A structured system for firmware production would reduce the overall costs of applications development.
- The lead time required and the overall time in development of an application system would be reduced by the use of a centralized firmware generator as contrasted with the typical non-uniform, individualistic approach.
- Since both the application strategy and the microprocessor description are available to the system, automatically generated documentation would be possible.
- The availability of documentation produced concurrently with the control-firmware provides the basis for a sophisticated configuration control system, insuring that released firmware is accompanied by consistent, regulated user information.
- The centralized production of documentation automatically provides uniformity in the format of application report documentation.
- A centralized system likewise provides uniformity in the structure, style, and techniques used in the production of the actual firmware programs.
- Since all microprocessor definitions are stored as library entries, the correction of errors, the upgrading of techniques, and the addition of new language features may be accomplished at a single point, the particular microprocessor definition volume. A retranslation of earlier strategies would allow a simple method of converting them to include the latest improvements.
- As the library is the one source of microprocessor functional definitions, a convention for the naming of various interface elements might be possible, allowing uniformity of hardware-level nomenclature among the various applications engineers.
- The use of a central system for application firmware production allow an easy method of keeping historical records for backup and developmental analysis. Each use of the system following the original strategy definition would constitute a revision, which would be numbered, dated, and stored in some archival medium such as microfilm.

- The use of an English-like strategy couched in terms of the application and removed from the intricacies of microprocessor firmware would allow a reduction in the skill levels required for application engineers.
- Limiting the sphere of knowledge needed by application engineers to that of the application and its interface would produce a reduction in the learning curve associated with such a task.
- The firmware produced from library entries would be more consistent if all firmware were written by a library maintenance staff; this group would therefore be chosen according to their familiarity with the microprocessors to be used.
- The isolation of microprocessor definitions in the library would allow a strategy to be realized in several microprocessor types without additional effort. Further, the upgrade from one microprocessor to another would be possible by a rerun of the original strategy against a new library volume.
- The automatic production of firmware allows the entire process of design to reside within an engineering group, lessening the delay and communications problems caused by dealing with personnel not familiar with applications.
- The automatic production of documentation, the reduction of the "learning curve", and the uniformity of the firmware produced would lessen the impact of the rapid turnover in engineering personnel pandemic today.

Most or all of these benefits would be found in a basic system intended to demonstrate the feasibility of the MPACT approach as applied to microprocessors; they directly augment the production of control programs. There are, however, other less tangible benefits.

The model developed as a part of this work is germane to any application expressible as a control problem. This fact amplifies the significance of the two primary features of the model: that control problems may be viewed as having dual nature (timing-dependent and timing-independent), and that the three fundamental information sets of the control problem (application, processor and resultant program) are of similar form and may be expressed in terms of the duality.

These results allow a conceptual continuity in studying the control problem. They also point to a natural and simple way of defining the elements of the problem, which has been exploited in the MPACT language. The concept of strategy removes from the behavioral specification of the problem the need to explicitly determine timing relationships. Most programming languages require such an explicit determination because of the significance that is associated with statement sequencing. Sequencing is of no concern in a strategy, removing timing considerations to a more simply determined list format.

The MPACT approach to time-domain system specification is the heart of this proposal. The possibility of future extensions, expanding the concept, such as automatic selection of the processor which most advantageously realizes an application, make this feasibility study the basis of an open-ended research program.

CONCLUSION

The benefits associated with a system to automatically produce microprocessor firmware from a behavioral description of a control application are worthy of an intensive program of research and development. Demonstration of the feasibility of such a system would provide tangible rewards in both the academic and industrial domains.

APPENDIX A

Language Design and Non-procedural Languages

A1 Backus, J., Reduction Languages and Variable Free Programming, IBM Research Report RJ1010 (April 1972).

A2 Balzer, R. M., "Dataless Programming", FJCC (1967).

A3 Barbieri, et al, Computer Compiler Organization Studies, Office of Aerospace Research, Bedford, Massachusetts (May 1967).

A4 Childs, D. L., "Description of a Set-theoretic Data Structure", FJCC (1968).

A5 Crespi-Regnizzi, et al, "The Use of Grammatical Inference for Designing Programming Languages", CACM (February 1973).

A6 Dijkstra, E. W., "Cooperating Sequential Processes", Programming Languages, Academic Press (1968).

A7 Figueroa, M. A., Analysis of Languages for the Design of Digital Computers, Illinois University Report AD-762 381 (1973).

A8 Floyd, R. W., "Nondeterministic Algorithms", JACM 14 (1967).

A9 Goldsmith, C. W., "The Design of a Procedureless Programming Language", SIGPLAN Notices, 9, 4 (1974).

A10 Heidorn, G. E., "English as a Very High Level Language for Simulation Programming", SIGPLAN Notices, 9, 4 (1974).

A11 Hemming C., and R. J. Smith, "An Introduction to Register Transfer Level Simulation of Digital systems", Southern Methodist University, Technical Report CP 73013 (1973).

A12 Homer, E. D., "An Algorithm for Selecting and Sequencing Statements as a Basis for a Problem Oriented Programming System", National ACM Conference (1966).

A13 Katy, J., and W. C. McGee, "An Experiment in Non-procedural Programming", FJCC (1963).

A14 Leavenworth, B. M., and J. Sammet, "An Overview of Non-procedural Languages", SIGPLAN Notices, 9, 4 (1974).

A15 McCarthy, J., "Recursive Functions of Symbolic Expressions and their Computation by Machine", CACM (April 1960).

A16 Matelan, M. N., BETA Syntax of the MAGIC Compiler, EAGER-080, General Dynamics Corporation, Fort Worth (1973).

A17 Matelan, M. N., "A Method for Conversational Interpreter Test Language Translation", Symposium Record, IEEE Transaction on Aerospace and Electronic Systems, St. Louis (1970).

A18 Shaw, M., "Reduction of Compilation Costs Through Language Contraction", CACM (May 1974).

A19 Tirrell, A. K., "A Study of the Applications of Compiler Techniques to the Generation of Micro-code", Proceedings of the ACM SIGPLAN-SIGMICRO interface Meeting (May 1973).

A20 Wegner, P., "Data Structures Models for Programming Languages", SIGPLAN Notices, 6, 2 (February 1971).

APPENDIX B

Microprocessor General and Application Information

B1 Baskin, H. B., et al, "Modular Computer Sharing System", CACM, (October 1969).

B2 Bell, C. G., et al, "The Architecture and Applications of Computer Modules: A Set of Components for Digital Design", IEEE COMPCON (1973).

B3 Callahan, B. L. J., "The State-of-the-Art", Modern Data (May 1974).

B4 Cooper, R. G., "Micromodules: Microprogrammable Building Blocks for Hardware Development", Proceedings of the Symposium on Computer Architecture (1973).

B5 Cushman, R. H., "What Can You Do With a Microprocessor?", EDN (March 20, 1974).

B6 Kaufman, P. A., "How About The Low Cost ($1000) Minicomputer", Session 11, WESCON (1973).

B7 Fuller, S. H., et al, "Computer Modules: An Architecture for Large Digital Modules", Proceedings of the Symposium on Computer Architecture (1973).

B8 Juliussen, J. E., and F. J. Mowle, "Multiple Microprocessors with Common Main and Control Memories", IEEE Transactions on Computers, C-22, 11 (November 1973).

B9 Lee, R., "Microprocessor ICs Improve Instruments", Electronic Design (April 26, 1974).

B10 Matelan, M. N., "DELTA - The Uses of Microprocessors in the Distributed Control of Electronic Testing", Joint Conference on 'Advances in Automatic Testing Technology', University of Birmingham, Birmingham, England (1975).

B11 Metzer, J., "Forum: Its Go for Microprocessors", Electronic Products Magazine (November 19, 1973).

B12 Misuanas, D., "Petri Nets and Speed Independent Design", CACM (August 1973).

B13 Parasuraman, B., "Applications of LSI", Session 11, WESCON (1973).

B14 Reyling, G., "Performance and Control of Multiple Microprocessor Systems", Computer Design (March 1974).

B15 Runyun, Stanely, "Microprocessors Showing Promise in Test Equipment, But Haven't Made It Big Yet", Electronic Design (April 26, 1974).

B16 Schultz, G. W., And R. M. Holt, "MOS/LSI Minicomputer Comes of Age", FJCC (1972).

B17 Smith, H., "Impact of Microcomputers on the Designer", Session 11, WESCON (1973).

B18 Torrero, E. A., "Microprocessors Finding Growing Role between Calculator Chips and Minis", Electronic Design (May 24, 1973).

B19 Waaber, S., and G. C. Feth, "LSI Microprocessing Workshop Summary", Computer Decisions (January 1974).

B20 Weissberger, A. J., "MOS/LSI Microprocessor Selection", Electronic Design (June 7, 1974).

B21 Whitney, T. M., et al, "The 'Powerful Pocketful': An Electronic Calculator Challenges the Slide Rule", Hewlett-Packard Report (1972).

C1 Biewer, M., "Designers Guide to Programmed Logic", Prolog Corporation, Monterey, California (1974).

C2 Callan, David J, and R. Baskin, "Microprocessor System Design", Digital Design (February 1975).

C3 Kenny, Ron, "Microprogramming Simplifies Control System Design", Computer Design (February 1975).

C4 Weiss, C. D., "Software for MOS/LSI Microprocessors", Electronic Design (April 1, 1974).

C5 Weiss, C. D., "MOS/LSI Microcomputer Software", Electronic Design (April 12, 1974).

C6 Weiss, C. D., "Basic Microcomputer Software", Electronic Design (April 26, 1974).

C7 Anon, A Guide to PL/M Programming, Intel Corporation, Sunnyvale, California (July 1973).

C8 Anon., Preliminary MCS-8 Assembly Manual, Intel Corporation, Sunnyvale, California (1974).

C9 Anon., 8080 Preliminary Specifications, Intel Corporation, Sunnyvale, California (1974).

C10 Anon., PPS-26: Programmed Processor System Preliminary Users Guide, Fairchild Semiconductor Corporation, Mountain View, California (1974).

C11 Panel Discussion, "Microcomputers and the Software Revolution", COMPCON 75, Spring IEEE Computer Society Conference, San Francisco, California (1975).

APPENDIX D

Microprocessor Hardware

D1 Cushman, R. H., "Understanding the Microprocessor is No Trivial Task", EDN (November 20, 1973).

D2 Cushman, R. H., "Understand the 8-Bit Microprocessor: You'll See a Lot of It", EDN (January 20, 1974).

D3 Cushman, R. H., "Microprocessors are Rapidly Gaining on Minicomputers", EDN (May 20, 1974).

D4 Cushman, R. H., "The Intel 8080: First of the Second-Generation Microprocessors", EDN (may 5, 1974).

D5 Holt, R. M., and M. R. Lemas, "Current Microcomputer Architecture", Computer Design (February 1974).

D6 Lewis, D. R., "Microprocessor or Random Logic?", Electronic Design (September 1, 1973).

D7 Lewis, D. R., "How to Build a Microcomputer", Electronic Design (September 13, 1973).

D8 Lewis, D. R., "Clear the Hurdles of Microprocessors", Electronic Design (Spetember 27, 1973).

D9 Ogdin, J. L., "Survey of Microprocessors Reveals Limitless Variety", EDN (April 20, 1974).

D10 Wickes, W. E., "A Compatible MOS/LSI Microprocessor Device Family", Computer Decisions (July 1973).

D11 Wiener, H., "Computers that Fit in Your Pocket", Computer Decisions, (August 1973).

D12 Anon., OEM Survival Handbook, Computer Automation Corporation, Irvine, California (1975).

APPENDIX E

Compiler-compilers and Meta-languages

E1 Feldman, J., and Gries, D., "Translator Writing Systems", CACM (February 1968).

E2 Feldman, J., "A Formal Semantics for Computer Languages and its Application to a Compiler-compiler", CACM (January 1966).

E3 Floyd, R. W., "The Syntax of Programming Languages - A Summary", IEEE Transactions on Computers, 13, 4 (August 1964).

E4 Leavenworth, B. M., "Syntax Macros", CACM (November 1966).

E5 Lyon, G., "Syntax-directed Least Errors Analysis for Context-free Languages: A Practical Approach", CACM (January 1974).

E6 Mcilroy, M. D., "Macro Extensions of Compiler Languages", CACM (April 1960).

E7 Matelan, M. N., "A Compiler-compiler System for Avionics Test Languages", IEEE Aerospace and Electronic Systems Symposium Record (1973).

E8 Matelan, M. N., The MAGIC BETA Compiler, EAGER-066, General Dynamics Corporation, Fort Worth (1971).

E9 Matelan, M. N., The MAGIC BETA Library - Theory and Application, Department of Defense (USAF) FZE-12-451 (and General Dynamics Corporation), (1975).

APPENDIX F

Parsing and Storage Techniques

F1 Aho, A. V. and S. C. Johnson, "LR Parsing, Computing Surveys (June 1974).

F2 Aho, A. V., and Ullman, A., "A Technique for Speeding up LR(k) Parsers", SIAM Journal, 2. 2 (1973b).

F3 Bachman, C. W., "The Evolution of Storage Structures", CACM (July 1972).

F4 Bays, C., "The Reallocation of Hash-coded Tables:, CACM (January 1973).

F5 Bell, J. R., "Threaded Code", CACM (June 1973).

F6 Bell, J. R., "The Quodratic Quatient Method: A Hash Code Eliminating Secondary Clustering", CACM (February 1970).

F7 Brent, R. P., "Reducing the Retrieval Time of Scatter Storage Techniques", CACM (February 1973).

F8 DeRemer, F. L., Practical Translators for LR(k) Languages, Ph.D. Thesis, Project MAC, MIT (1969).

F9 Gantes, G., and D. A. Poplawski, "A Simple Technique for Structured Variable Lookup", CACM (September 1973).

F10 Gries, D., "Use of Transition Matrices in Compiling", CACM (January 1968).

F11 Henderson, P., "Derived Semantics for Some Programming Language Constructs", CACM (November 1972).

F12 Knuth, D. E., "Top Down Syntax Analysis", Acta Informatica, 1, 2 (1971).

F13 Lewis, A., "Syntax Directed Transduction", JACM, 15, 3 (1968).

F14 Mickunas, M. D., and V. B. Schneider, "A Parser-generating System for Constructing Compilers", CACM (November 1973).

F15 Morris R., "Scatter Storage Techniques", CACM (January 1968).

F16 Wagner, R. A., "Common Phrases and Minimum Space Text Storage", CACM (1973).

APPENDIX G

Optimization Techniques

G1 Allen, F. E., "Control Flow Analysis", SIGPLAN Notices, 5, 7 (1970).

G2 Allen, F. E., and J. Cocke, "A Basis for Program Optimization", IFTPS Congress Proceedings (1971).

G3 Cocke, J., "Global Common Subexpression Elimination", SIGPLAN Notices, 5, 7 (1970).

G4 Cocke, J., and Schwartz, Programming Languages and Their Compilers, New York University (1970).

G5 Earnest, C., "Some Topics in Code Optimization", JACM, 21, 1 (January 1974).

G6 Lowry and Medlock, "Strength Reductions", CACM (January 1969).

APPENDIX H

Software Texts

H1 Barron, D. W., Assemblers and Loaders, American Elsevier (1972).

H2 Gries, D., Compiler Construction for Digital Computers, Wiley and Sons (1971).

H3 Harrison, M. C., Data Structures and Programming, Courant Institute, New York University (1972).

H4 Knuth, D. E., The Art of Computer Programming, Volume I, Fundamental Algorithms, Addison-Wesley (1968).

H5 Lee, J. A. N., Anatomy of a Compiler, Reinhold (1967).

H6 Pollack, B. W., Compiler Techniques, Auerback (1972).

APPENDIX I

Hardware Texts

I1 Bartee, T. C., Digital Computer Fundamentals, McGraw-Hill (1966).

I2 Breuer, M. A., Design Automation of Digital Systems, Volume I, Prentice-Hall (1972).

I3 Gschwind, H. W., Design of Digital Computers, Springer-Verlag (1967).

I4 Hellerman, H., Digital Computer System Principles, McGraw-Hill (1967).

I5 Hill, F. J., and G. R. Patersen, Switching Theory and Logical Design, John Wiley and Sons (1968).

I6 Hill, F. J., and G. R. Petersen, Digital Systems: Hardware Organization and Design, John Wiley and Sons (1973).

APPENDIX J

On the Nature of Time and Techniques of Analysis

J1 Emshoff, J. R., and R. L. Sisson, Design and Use of Computer Simulation Models, Macmillan Publishing (1970).

J2 Finkbeiner, D. T., Matrices and Linear Transformations, W. H. Freeman (1965).

J3 Frailey, D. J., "Timing Features for Systems Implementation Languages", Proceedings, IFIP '74 (1974).

J4 Frailey, D. J., DSOS - A Skeletal, Real-Time Minicomputer Operating System, Technical Report CP-73012, Southern Methodist University, CS/OR Department (December 1973).

J5 Halliday, David, Nuclear Physics, John Wiley (1966).

J6 Harrary, Frank, Graph Theory, Addison Wesley (1972).

J7 Stimler, S., and K. A. Brons, "A Methodology for Calculating and Optimizing Real-Time System Performance", CACM (June 1968).

J8 Taylor, E. F., and J. A. Wheeler, Spacetime Physics, W. H. Freeman (1966).

J9 Watson, R. W., Timesharing System Design Concepts, McGraw-Hill (1970).

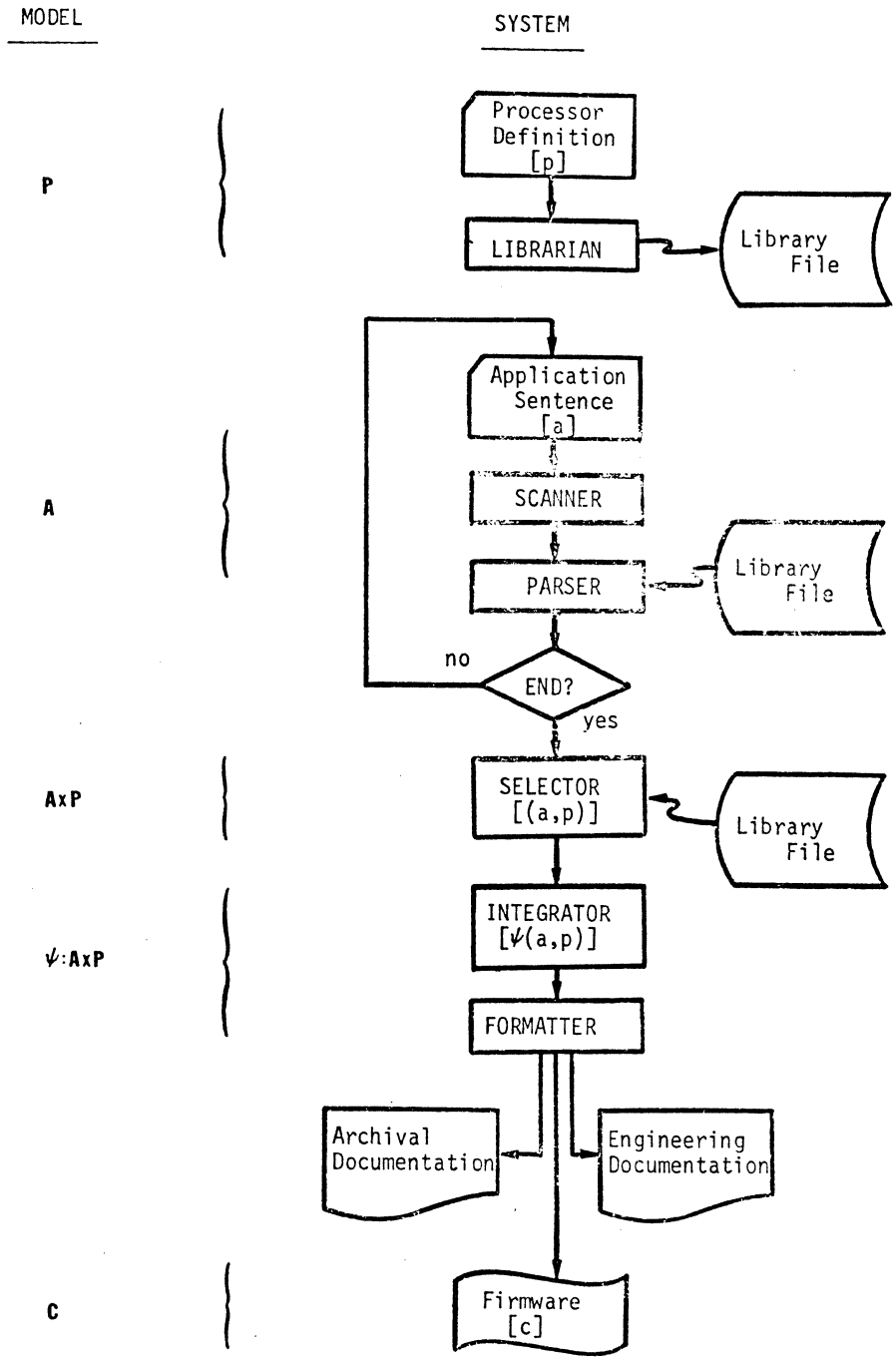


Figure I

MPACT Control Flow (ψ:AxP→C)

Interactive Package for Analysis and Synthesis of Multivariable Control Systems

****

F. Gadzinski*, J.H. Anderson, P. Anand
University of Waterloo, Ontario, Canada

ABSTRACT

This paper describes an interactive package for designing multivariable control systems. It is assumed that a model of the system, either in the form of a matrix of transfer functions or in state-space form is known. The design procedure is by successive analysis. In each iteration the designer checks the stability and transient response of the system. The package is implemented on a PDP-9 computer with precision display, disk, line printer and teletype.

Keywords and Phrases:

Software package, interactive control system design, multivariable systems.

INTRODUCTION

The classical frequency methods of designing a control system are familiar to control systems designers and include in the design procedure such requirements as overshoot, rise time, amplitude and phase stability margins. These methods deal with single loop systems and therein lies the difficulty in applying them to multivariable systems. Designing multivariable systems as a set of single loops, does not take into consideration interaction between the loops and may cause degradation in system performance and system instability.

Optimal control theory supplies a required theoretical base for designing multivariable control systems. Unfortunately, it requires accurate system models and measurability of all states of the system. The last requirement may be waived by using a state estimator at the cost of increased system complexity. Optimal control theory requires that the system requirements be specified in terms of weighting co-efficients of a performance index. In most cases, it is very difficult to translate overshoot, rise time or stability margins into values of the weighting co-efficients.

The system model is usually derived on the basis of physical considerations and it is usually non-linear. Such a model, linearized in the vicinity of some steady-state operation point gives a linear model of the system, which can be used to design optimal linear regulator (1),

* Now with Canadian General Electric, Peterborough, Ontario, Canada.

appropriate to that operating point. Identification methods give another alternative to determine system models. At the present time, the identification methods of state-space system representation are not developed well enough (4), so that they can be used in practice. On the other hand, identification methods of transfer functions of multivariable systems are more familiar to engineers and are better tested.

In the vector frequency response method of designing multivariable control systems (2) the system may be specified either in state-space form or transfer function form. It allows such requirements as amplitude and phase stability margins to enter directly into the design process. It involves some repeated and complex calculations and it seems that the computer aided design is the most convenient approach for its implementation.

There are a few available software packages for designing multivariable control systems and some of them are mentioned in (3). The present article presents another one, which is based on the vector frequency response method of designing multivariable control systems.

A VECTOR FREQUENCY RESPONSE METHOD

An assumption is made that the control system is represented in the form shown in Figure 1. The elements of the matrices K,Q and H are rational functions of the Laplace complex frequency S. It can show (3) that the characteristic polynomial of the closed loop system in Figure 1 is given by the equation:

$$C_{CL} = C_{OL} \det F \qquad (1)$$

where

C_{OL} - characteristic polynomial of the open-loop system.

$F = I+A(S)K(S)H(S)$

Similar to the way Nyquist stability criterion is proved, it can be shown that a multivariable system is absolutely stable if

$$N_f = P_o \qquad (2)$$

where

P_o - number of right half zeroes of open loop characteristic polynomial.

N_f - number of times det F(S) encircle the origin in the complex plane, when S-change along the contour D, shown in Figure 2.

The stability criterion in this form is not suitable for design purposes. An equivalent form can be obtained by expressing the determinant of the matrix F in terms of eigen-values.

$$\det F(S) = \prod_{j=1}^{m} f_j(S) \qquad (3)$$

where $f_j(S)$ called characteristic values are the eigen values of the matrix F calculated for a given value of S. Substituting (3) into (2) we have:

$$P_o = -\sum_{j=1}^{m} f_j(S) \qquad (4)$$

Let us assume that the matrix Q for some value $S=S_1$, can be written in the form:

$$Q(S_1) = [T(S_1)] \quad [\lambda(S_1)] \quad [T(S_1)]^{-1}$$

where

$[\lambda(S_1)]$ - diagonal matrix of eigen-values

$[T(S)]$ - matrix of eigen-vectors

Now, if the matrix K, for $S=S_1$, is given by

$$K(S_1) = [T(S_1)] \quad [\lambda_k] \quad [T(S_1)]^{-1}$$

then

$$Q(S_1)K(S_1) = [T(S_1)] \quad [\lambda_k{}^*\lambda] \quad [T(S_1)]^{-1} \qquad (5)$$

where $[\lambda_k{}^*\lambda]$ - a diagonal matrix with diagonal elements equal to the product of the eigen-values of matrix Q and matrix K.

The package described in this paper is based on equations (4) and (5).

DESIGN PROCEDURE

The system configuration used in this paper for the frequency response method of designing multivariable control systems is shown in Figure 3. It is assumed that the system description is given and the controller K has to be found, so that the closed-loop system is stable and meets imposed requirements. The computer design procedure is as follows:

1. Read in a description of the system.
   The system description may be specified in one of the two forms:

a) $\dot{x} = Ax+Bu$

$y = Cu$, $[A]$ n,n, $[B]$ n,m, $[C]$ m,n

b) $y = Gu$, $[G]$ m,m

Where G is a transfer function matrix with elements in the form of rational function of complex variable S.

2. Choose the controller K.
The controller K is given by the form

$$K = K_1K_2K_3\ldots K_9 \quad ; \quad K_1 = [k^i] \quad m,m$$

The parameters of these arrays K_i should be specified. Usually in the design procedure the arrays K_i are specified in consecutive iteration steps. Some of them have real entries and others have entries determined as rational function of complex variable S. The designer has the option to choose controller K_i, which is calculated on the basis of equation (5). For the specified frequencies, f_1, f_2 and for the specified regulator structure (PID, lead-log network etc....) a regulator is calculated which modifies system characteristic values for the frequencies f_1, f_2 in a desired way. This option may be helpful in setting required for phase and amplitude stability margins.

3. Inspect characteristic loci plots for the system.
If the requirements with respect to the stability of the system are not met, go to point 2 of the design procedure. Modify the current K matrix or use another K_i matrix in order to improve stability of the system.

If the system is stable, check its stability for different types of failures: Transducer failure, actuator failure, regulator failure. This is accomplished by including into the system, shown in Figure 3, matrices which simulate failure.

4. Inspect the transient response of the closed-loop system.
If the transient response does not meet requirements go to point 2 and modify the controller K.

5. Print the final parameters of the controller.

PACKAGE IMPLEMENTATION

The package is run on a PDP-9 computer (32K core) with precision display, teletype, line printer and magnetic tape unit. WATRAN was used as a programming language. Overlay techniques were used extensively to accomodate the package in core. All the input data

is entered in free format. The user, via a light pen, indicates which data to enter. Characteristic loci and transient response plots are displayed in the corresponding steps of the design procedure. On request the designer may get output on the line printer. Error and information messages are printed on the teletype. All the entered data, at any moment of the design procedure, may be displayed for verification or modification.

CONCLUSIONS

The package does not in any sense design a multivariable system automatically. It rather supports the designer in an interactive design procedure. There are some aspects which still require further work and attention, for example, guide lines on how to choose controllers K_i. The vector frequency response method of designing multivariable control systems is relatively new. There are not too many published examples relating to realistically complex processes where this method has been used effectively. It seems that in its present state there is room for development of the method and much experience to be gained. Never-the-less the method has two strong points. First, it allows to check stability of the system with interaction between loops taken into consideration. Second, it allows amplitude and phase stability margins to be set. There are indexes of performance already familiar to control systems design engineers and they are not quantities in such direct evidence using optimal control system design methods.

REFERENCES

1. A.P. Sage - Optimal System Control. Prentice-Hall, Inc. (1968)

2. A.G.J. Macfarlane - A Survey of some Recent Results in Linear Multivariable Feedback Theory. Automatica 8, pp 455-492 (1972)

3. A.G.J. Macfarlane - Round Table Discussion, Trends in Linear Multivariable Theory, Automatica 9, pp 273-277 (1973)

4. K.J. Astrom, P. Eykhoff - System Identificarion - A Survey. Automatica, Vol 7. pp 123-167 (1971)

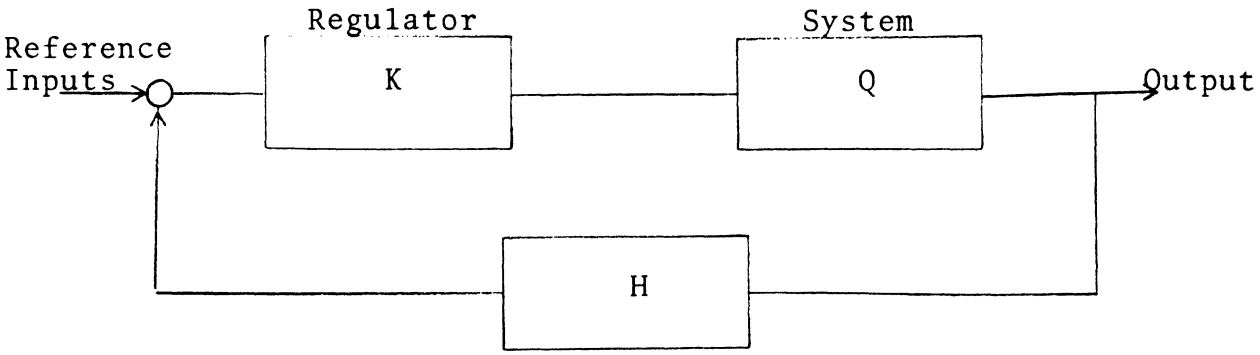


Figure 1. Feedback Regulator.

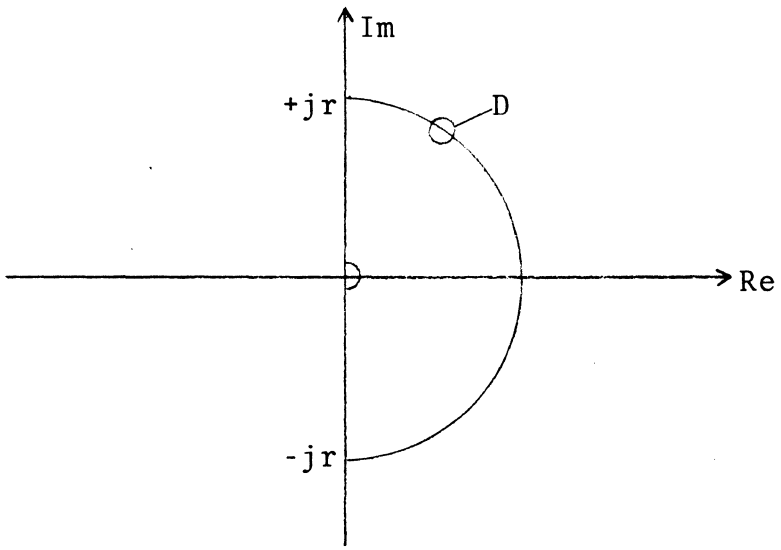


Figure 2. Nyquist Contour.

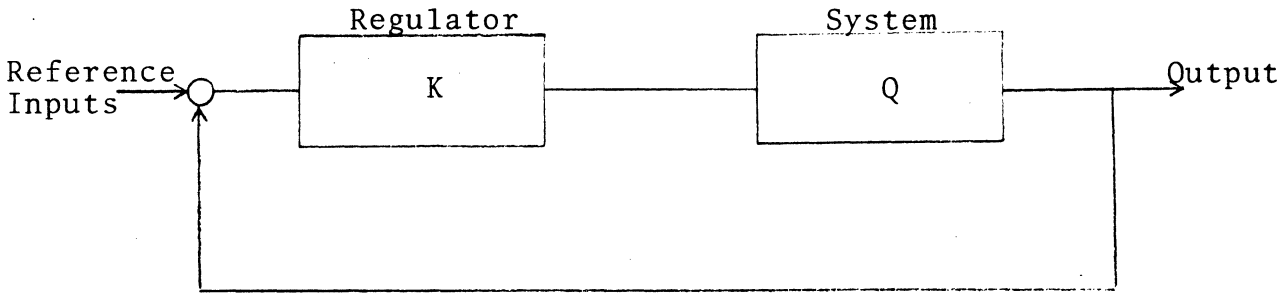


Figure 3. Feedback Regulator.

1976 acm computer science conference

disneyland hotel / anaheim / california / 10-12 february 1976

short reports on current research
by students, faculty, and researchers in the computer and information sciences
abstracts only are required/deadline for submission of abstracts is 1 december 1975

invited state of the art papers
tutorial hardware exhibits/book displays
luncheon for department chairpersons
employment register

technical symposium of
acm special interest group on computer science education (sigcse)
disneyland hotel on 12-13 february 1976

for additional information on csc 76 write to
julian feldman/chairperson csc 76
or fred tonge/vice chairperson csc 76
department of information and computer science
university of california
irvine/california 92664

for additional information on the employment register at csc 76 write to
orrin e. taulbee
department of computer science
university of pittsburgh
pittsburgh/pennsylvania 15260

PROCEEDINGS FOR 9th, 10th and 11th DA WORKSHOP

The following is the rate schedule as agreed to by SIGDA and the DA Technical Committee of IEEE.

	COPIES		SIGDA & ACM Members	All Others	Amount
1)	_____	9th DA Workshop Proceedings (1972) @ $10.00 376 Pages	$10.00	$16.00	_____
2)	_____	10th DA Workshop Proceedings (1973) @ $10.00 288 Pages	$10.00	$16.00	_____
3)	_____	11th DA Workshop Proceedings (1974) 379 Pages	$12.00	$20.00	_____
		Member Number _____		TOTAL =	_____

Please send your order prepaid to:

ACM Inc.
P.O. Box 12105
Church Street Station
New York, New York 10249

Make your checks payable to Association for Computing Machinery (an added charge is made for billing).

JOIN SIGDA JOIN SIGDA JOIN SIGDA JOIN SIGDA JOIN SIGDA JOIN SIGDA JOIN SIGDA JOIN SIGDA

_____
Name (please type or print)

_____
Affiliation

_____
Mailing Address

_____
City State Zip

Annual membership dues are $3.00 for ACM members and $5.00 for others.

____ Enclosed annual dues.

____ Please send more info.

Mail to SIGDA
ACM INC.
P.O. Box 12105
Church Street Station
New York, NY 10249

Association for Computing Machinery
1133 Avenue of the Americas, New York, New York 10036