Department of

ELECTRICAL ENGINEERING

UNIVERSITY OF NOTRE DAME, NOTRE DAME, INDIANA

COMPUTER-AIDED DESIGN AND ANALYSIS
OF CIRCUITS AND SYSTEMS

by Eugene W. Henry

Technical Report No. EE-686

July 1, 1968

Department of Electrical Engineering
University of Notre Dame, Indiana 46556

Final Report, NASA Research Grant NGR 15-004-021

The work reported here covers the period from September 20, 1966 to July 1, 1968. During the period from February 28, 1967 to February 29, 1968 this work was sponsored by the National Aeronautics and Space Administration under Research Grant NGR 15-004-021. There were no activities or expenditures on this project from June 5, 1967 to September 20, 1967. This report is published for information purposes only and does not represent recommendations or conclusions of the sponsoring agency.

RESEARCH STAFF

Eugene W. Henry, Principal Investigator

Associate Professor of Electrical Engineering

Edwin V. Harrington, Jr.

Research Assistant

William Rombalski

Research Assistant

William F. Hartman

Graduate Fellow

TABLE OF CONTENTS

I. INTRODUCTION

A formal effort in the area of computer-aided design and analysis of circuits and systems (CADACS) was initiated by the Electrical Engineering Department of the University of Notre Dame in September 1966. This project was organized to consolidate existing in-house computer routines with externally developed programs on a single program library tape which would be documented and made available to all students and faculty for purposes of education, design, and research.

Support and cooperation with the National Aeronautics and Space Administration was solicited with the hope that the following mutual benefits would accrue:

1. Information from NASA about computer programs presently in use or under development would assist in defining the scope and in developing CADACS at Notre Dame.

2. Support of full-time graduate research assistants would accelerate and lend continuity to the program.

3. NASA would be able to supply CADACS programs to its contractors and to other government agencies, thus obviating costs of duplicate program development.

4. Notre Dame would assist in developing standardized programs for NASA-oriented circuits by evaluating existing programs or by developing new approaches.

5. By making CADACS programs available to other universities, NASA could assist in the advancement of teaching and research.

The NASA Research Grant, which covered the period from February 28, 1967 to February 29, 1968, provided for the support of two research assistants during the Spring semester of 1967, and one research assistant during the Fall semester of 1967. There was no activity on this project during the summer of 1967. Another graduate student who contributed significantly to this work was supported by a fellowship. During the Spring semester of 1968 two seniors in Electrical Engineering participated in the CADACS effort by making the ECAP and MIMIC digital programs operable with the Univac 1107 on the CADACS program library tape.

Goals of this project, some of which were established at its inception, and some of which were developed during the course of the work, are listed below:

1. To develop a standard format for the description of CADACS programs in order to simplify their use and aid in the consolidation and standardization of such programs, and to develop qualification procedures for existing circuit design programs as supplied by NASA.

2. To demonstrate the applicability of the Network Analysis for System Applications Program (NASAP) to the design of acoustic, electronic, mechanical and hydraulic systems.

3. To complete the development and/or evaluation of CADACS programs to provide transfer functions, transient responses, frequency responses, sensitivity analysis,

root locus plots, and digital simulation of continuous systems.

4. To employ NASAP and other CADACS routines as integral parts of courses and laboratories in the networks and systems areas, and to evaluate their effectiveness as aids in teaching and in design.

5. To augment the capabilities of NASAP by adding a subroutine for determining sensitivity of circuit performance to each element, and to develop algorithms to facilitate data input coding and to improve the efficiency of NASAP.

As the following chapters of this report indicate, these goals have been achieved with different degrees of success. The implementation of NASAP and other CADACS programs to operational status and their application in courses and laboratories has been particularly successful. On the other hand, attempts to establish standards and qualification procedures have not been significantly effective.

There is no doubt that CADACS programs will assume continuously increasing importance in the university and in industry because of their nearly unlimited capacity for tutorial and design assistance in the area of engineering systems.

II. PROGRAM STANDARDIZATION AND QUALIFICATION

One of the initial objectives of this project was to develop a standard format for description of CADACS programs, including symbols, abbreviations, indexing, etc., to facilitate the consolidation and standardization of programs and to simplify their use. Another related goal was to develop qualification procedures for existing circuit design programs as supplied by NASA, and to compare these with similar programs on the basis of running time, accuracy, operational flexibility, ease of use and other criteria which may be formulated for purposes of program standardization.

Standardization of programs in the detail specified above does not appear to be feasible because of the widely divergent backgrounds and preferences of the program originators, and the peculiarities of the computer installations which utilize these programs. Although rigid standardization may seem to be a desireable goal from the standpoint of user convenience and interchangeability of programs by different computing centers, it may have the deleterious effect of stifling innovative techniques.

At this early stage in the evolution of computer-aided design the "standard" which would most effectively insure a reasonable interchange of programs is the coding of all routines in the Fortran IV source language. It is well known that "standardization" to Fortran is no panacea because of the inevitable debugging and recoding required to adapt an existing program to a different computer or even the same type of computer at another installation, but since Fortan is the most widely used scientific computer language it provides the best available common denominator.

It has been our experience at Notre Dame that it requires a student between one and two semesters to effect operational capability of an existing large Fortran program such as ECAP or MIMIC on our Univac 1107 computer.

The programs now in use are of the off-line, batch-processing type. The current trend toward conversational mode programs with time-shared computers brings another dimension to the problem of interchangeability.

Computer program qualification, as opposed to standardization, is clearly an attainable goal if the objectives are well defined. A detailed list of suggested qualification procedures was developed and presented to the other NASAP participants at a meeting in New York during the IEEE Convention in March, 1967. This list is included in this report as Appendix A.

No comments were received on these suggested qualification procedures from the other NASAP users, and insufficient intercommunication occurred to provide the thorough testing and revision as suggested in the procedures. Since documentation of NASAP was the responsibility of another participant, only local qualification was done.

III. AFPLICATION OF NASAP TO NON-ELECTRONIC SYSTEMS

A computer program such as NASAP which is ostensibly intended to solve electrical network problems, is in fact not so narrowly limited in scope of application. Any physical dynamic system which can be modeled by an analogous electrical circuit can be analyzed by NASAP.

Part of the CADACS effort was devoted to the development of analogs, coding procedures, and solution by NASAP of example problems involving acoustic, mechanical, hydraulic, and pneumatic systems.

Complete details of the NASAP solution of mechanical, electrical and hydraulic systems is given in a previously published technical report (Reference 1). A paper on acoustic systems was submitted for the 1968 Spring Joint Computer Conference, but it was not accepted for publication. The latter paper was presented in March 1968 at a meeting of the South Bend Section of the IEEE.

An unattractive feature of the early version of NASAP, and the most probable reason for rejection of the acoustics paper, was the cumbersome input coding, and the requirement that the user develop a certain type of topological tree from the network. Later versions of NASAP, such as developed at Northeastern University (Reference 2), require only a straightforward coding of the etwork element incidences and values as in ECAP.

It is planned to rewrite the acoustics paper in light of the new input format, and resubmit for publication.

IV. IMPLEMENTATION OF CADACS PROGRAMS

A group of CADACS programs has been developed, tested, and stored on a single magnetic tape for use on the UNIVAC 1107 digital computer. We present here a brief description of each program on the tape. Details of the use of these programs will be given in a CADACS Users Manual, to be published as a departmental report.

1. NASAP (Network Analysis for Systems Applications Program). Developed by the NASA Electronics Research Center with the cooperation of several universities including Notre Dame, this program is planned to consist of at least the following nine subroutines:

a) Construction and evaluation of flowgraph from a network code.

b) Computation of the sensitivity matrix from a network function.

c) Transient response from a system transfer function.

d) Bode plot (frequency response) from a system transfer function.

e) Construction of equivalent circuit from network code.

f) Construction of graphic display for flowgraph.

g) Derivation of approximate flowgraph.

h) Construction of display for the approximate flowgraph.

i) Monte-Carlo tolerance analysis.

NASAP subroutines a, b, c, and d are operational on the UNIVAC 1107 from the CADACS tape. Some work has been done on the development of subroutines f, g, and h, but they are not ready for implementation on the tape. No work was done by this group on subroutines e and i. The present version of subroutine a does not provide the simplified ECAP-like input coding for the network as described in Reference 2. More specific details on the work done by this group on NASAP follow:

A. The storage required by the program was reduced to eliminate the need for going to magnetic tape during the computation. This was accomplished by reprogramming to eliminate the need for checking if a first order loop had been previously formed.

B. A correction was made in the original programming of the V matrix. Previously, due to an error in programming, an incorrect V matrix would be formed whenever a dependent source was present in the circuit. Now the correct matrices are formed regardless of the type of dependent elements.

C. A program was written to form a representation of a network flowgraph on the Calcomp plotter. Information contained in the T, V, and W matrices are used by the program to generate instructions for the plotter. This program is operational, but it has not been included on the CADACS tape.

D. Some work was accomplished on the development of approximate equivalent circuits. To determine if a parameter K contributes significantly to the system transfer function J in a given network, we may use tagging techniques to evaluate J with K set to zero and again with K set to infinity. If it is found that J_0 or J_∞ do not appreciably differ from J, then the element K is not particularly critical to the performance of the circuit. If J_0 is approximately equal to J, then K should be replaced by a short circuit if it is coded as an impedance, or by an open circuit if it is coded as an admittance. On the other hand, if J_∞ is approximately equal to J, then K should be replaced by an open circuit if it is coded as an impedance, or by a short circuit if it is coded as an admittance.

Calculation of the J's is based on formulas developed from the topology equation for closed flowgraphs. The topology equation showing explicit dependence on transfer function J and parameter K is[4,5]

$$H(J,K) = 0 \qquad (1)$$

Expanding Equation (1) into the part which does not contain J, i.e. $H(\bar{J},K)$, and the part which does contain J, i.e. $JH(J',K)$, we obtain

$$H(\bar{J},K) + JH(J',K) = 0 \tag{2}$$

A further expansion in terms of K yields

$$H(\bar{J},\bar{K}) + KH(\bar{J},K') + JH(J',\bar{K}) + KJH(J',K') = 0 \tag{3}$$

From Equation (3) we may determine J, J_o, and J_∞ as

$$J = - \frac{H(\bar{J},\bar{K}) + KH(\bar{J},K')}{H(J',\bar{K}) + KH(J',K')} \tag{4}$$

$$J_o = \lim_{K \to 0} J = - H(\bar{J},\bar{K})/H(J',\bar{K}) \tag{5}$$

$$J_\infty = \lim_{K \to \infty} J = - H(\bar{J},K')/H(J',K') \tag{6}$$

Work has been done to incorporate the calculations of Equations (5) and (6) in the NASAP program. However, this project has not been completed.

E. A transient response subroutine RESP has been implemented on the CADACS tape. The input data describes the transfer function and the system input function. The computer output includes an echo of the input data, a listing of the system output as a function of time, and a graph on the high speed line printer. The user may choose any one of the following input functions: impulse, step, ramp, cosine, sine, and grid(arbitrary function).

F. The Sensitivity Analysis feature of NASAP has been implemented on the CADACS tape. This routine allows the user to find the sensitivity as a rational function of s of a specified network transfer function to a given network parameter.

G. Frequency response programs BODE1 and BODE2 have been developed and implemented on the CADACS tape. Both programs produce magnitude and phase as a function of frequency for a specified rational transfer function. BODE1 generates plots on the high speed line printer, and allows the user to specify the number of frequency points. BODE2 produces Calcomp plots in addition to plots from the line printer, using 100 frequencies between specified upper and lower limits.

It is intended to augment these programs to allow a pure time delay or transportation lag in addition to the rational part of the transfer function.

2. CALAHN, a linear network analysis program by D. A. Calahan of the University of Michigan, has been implemented on the CADACS tape. The following specifications are condensed from Reference 3:

1) Input
   a) Network elements (type, value, and node number)
   b) Network input and output node numbers
   c) Type of transfer or driving-point function desired

2) Output
   a) Coefficients of the network function
   b) Poles and zeroes of the network function
   c) Frequency response (magnitude, phase, and delay)
   d) Time response for specified input
   e) Repeated outputs with variable parameter
   f) Network function in symbolic form

3) Limitations
   a) Voltage controlled current source is the only allowable active element
   b) Network complexity limited to 30 nodes, 100 passive elements, 20 active elements
   c) Memory storage: 45K words

3. ROTLOC, a root locus subroutine developed by J. J. Brann and Emil Tejkowski at Notre Dame, has been implemented on the CADACS tape. This program accepts as input data the pole and zero locations of an open loop transfer function and produces listings and plots of closed loop pole locations as a positive or negative parameter is varied from zero to infinity. The user may specify plots on the line printer, the Calcomp plotter, or both.

4. ECAP, Electronic Circuit Analysis Program written by IBM for use on their large computers, has been adapted to the UNIVAC 1107 and implemented on the CADACS tape. This program provides transient response, dc, and ac steady state analysis with numerical output for networks having up to

50 nodes, 200 branches, 200 dependent sources, 200 switches, and 50 parameter changes.

5. MIMIC, a block-oriented simulation language developed at Wright-Patterson Air Force Base for the IBM 7094, has been adapted to the UNIVAC 1107 and implemented on the CADACS tape. This program provides a tabular listing of transient response data for a system described by the user in terms of its differential equations.

V. USE OF CADACS PROGRAMS IN THE CURRICULUM

Introduction of CADACS programs in the curriculum provides the student with the means to readily solve more realistic problems than heretofore possible. Extensive use of these programs by the students also provides a wide assortment of test cases which assist in debugging and determining the limits of program capabilities.

The principal investigator introduced CADACS programs in three different courses. Table I indicates the amount of computer utilization by the students in each course.

1. EE 131 Introduction to Control Systems:
For the Spring semester of 1968 a new laboratory experiment schedule was adopted as shown in Table II. Whereas the previous laboratory made little use of the digital computer, as indicated by 0.17 hours per student per semester in Table I for the Spring semester of 1966, the new schedule increased digital computer usage to 1.14 hours per student per semester in 1968. Programs NASAP and CALAHM were studied

TABLE I

CADACS TUTORIAL UTILIZATION OF DIGITAL COMPUTERS

Course	No. of Students	Total Hours	Total Runs	Hours/ Student	Runs/ Student
EE131 Spring '66	45	7.5	318	.17	7
EE131 Spring '68	28	32.0	966	1.14	35
EE164 Fall '67	25	28.0	1081	1.12	43
EE212 Spring '67	9	3.8	156	.42	17
CF039 Spring '67 & Fall '67	3	8.5	243	1.42	40

EE131: Introduction to Control Systems (Required course in 1966, elective in 1968).

EE164: Introduction to Computers: Analog, Digital, and Hybrid

EE212: Network Synthesis

CF039: Graduate assistants working on CADACS project. Totals are cumulative for the two semesters. Last two columns are on per semester basis.

TABLE II

EE 131L Control Systems Laboratory Spring 1968

Exp.	Title (Type of Computer)
1	Digital Computer Programs for Systems Analysis (Digital)
2	Experimental Investigation of Synchro Devices (None)
3	Experimental Determination of the Transient Response of a Two-Phase AC Servo Motor (None)
4	Analog Computer Simulation of a Position Servo with Tachometer Feedback (Analog)
5	Experimental Investigation of a Position Servo with Tachometer Feedback (None)
6	Analog Computer Design of a Spacecraft Orientation System to Minimize the ITAE Criterion (Analog)
7	Root Locus Techniques: Use of the Spirule; A Digital Computer Program for Automatic Locus Plotting (Digital)
8	Analog Computer Design of a Phase-Lead Compensator for a Position Servomechanism (Analog)
9	Position Servomechanism with Cascade Carrier Network Compensation (None)
10	Experimental Determination of the Frequency Response of an AC Servomotor and Inertial Load (None)
11	Computational Aids for System Design and Analysis by Frequency Response Techniques (Digital)
12	Describing Function Analysis of a Position Servo Containing Backlash and Saturation (Analog)
13	Design of an Automatic Ship Steering System (Analog and Digital - two week design experiment)

in Experiment 1, followed by ROTLOC and RESP in Experiment 8, and BODE1 and BODE2 in Experiment 11. Next year MIMIC will also be introduced in the laboratory. The two week design experiment, No. 13, was initiated in 1968 to require extensive use of both analog and digital computers in a significant synthesis problem.

2. EE 164 Introduction to Computers:
Although this course has required extensive use of both analog and digital computers, as indicated by 1.12 hours per student in Table I, the digital work has been limited to student-generated Fortran programs. In 1968 a new laboratory session on problem-oriented languages will acquaint the students with the CADACS programs. They will then use these programs in solving a term problem.

3. EE 212 Network Synthesis:
The NASAP program was used by the graduate students in this course to verify the transfer functions of networks which they had synthesized. Appendix B presents a portion of an examination solution obtained with NASAP. It is planned to introduce ECAP and CALAHN next time the course is offered.

4. CF039 CADACS Project:
As indicated in Table 1, the graduate students working on this project utilized 1.42 hours of computer time per student per semester. This time was spent in developing, debugging and modifying the CADACS programs.

Because of the educational benefits derived from the use of CADACS programs, and the enthusiastic response of the students, it is planned to introduce these programs at the sophomore level, and perhaps at the freshman level.

VI. CONCLUSION

Following is a list of the presentations given, papers submitted, and conferences attended during the tenure of the grant:

A. Presentations:

1) "Computer Aided Design," by Eugene W. Henry, at the Student Branch IEEE Meeting, University of Notre Dame, September 27, 1967.

2) "Digital Computer Programs for Circuit and System Analysis," by Eugene W. Henry, for the 9598th Air Reserve Squadron, Research and Development Flight, South Bend, Indiana, October 16, 1967.

3) "A Computer Oriented Approach to Coding, Evaluating and Optimizing Acoustic Systems," by Edwin V. Harrington, Jr., at South Bend Section IEEE Meeting, March 21, 1968

B. Papers and Reports:

1) Edwin V. Harrington, Jr., "Computer Oriented Approach for Coding, Evaluating and Optimizing Electrical Mechanical Hydraulic Systems," University of Notre Dame Electrical Engineering Report EE6711, October, 1967.

2) Edwin V. Harrington, Jr., "A Computer Oriented Approach to Coding Evaluating and Optimizing Acoustic Systems," submitted for the 1968 Spring Joint Computer Conference. (Rejected)

C. Conferences Attended:

1) IEEE National Convention, March 22, 1967, special meeting with NASAP participants.

2) Conference on Computer Assisted Instruction in Engineering Education, State University of New York at Stony Brook, November 16-17, 1967.

REFERENCES

1. Edwin V. Harrington, Jr., "Computer Oriented Approach for Coding, Evaluating and Optimizing Electrical Mechanical Hydraulic Systems," University of Notre Dame Electrical Engineering Report No. EE6711, October, 1967.

2. R. E. Bach, Jr. et al., "Reliable Solid-State Circuits," Semiannual Report No. 6, NASA Research Grant NGR-22-011-077, Electronics Research Laboratory, Northeastern University, January 1, 1968.

3. D. A. Calahan, "Linear Network Analysis and Realization Digital Computer Program: An Instruction Manual," University of Illinois Engineering Experiment Station Bulletin 472, vol. 62, No. 58, February, 1965.

4. W. W. Happ, "Flowgraph Techniques for Closed Systems," IEEE Trans. AES-2, No. 3, pp 252-264, May 1966.

5. W. W. Happ, "Flowgraphs as a Teaching Aid," IEEE Trans. E-9, No. 2, pp 69-79, June 1966.

APPENDIX A

NETWORK ANALYSIS FOR SYSTEMS APPLICATIONS PROGRAM QUALIFICATION PROCEDURES

1. Introduction. The purpose of this memo is to outline suggested procedures for the qualification of NASAP subroutines and to solicit comments from NASAP participants on optimizing and implementing these procedures.

A "qualified" subroutine is a computer program which may be used with minimum effort and maximum confidence to efficiently solve a given class of problems. This definition implies that certain criteria are to be observed in developing an optimal subroutine:

a) To assure minimum user effort, the input format should be simply related to the problem formulation, and the output format should contain appropriate heading to make the data presentation self-explanitory. Diagnostics should be included to alert the user against improper input format or against a problem specification which exceeds the limits of the program.

b) Before a program can be used with confidence, it must be thoroughly debugged and tested by problems of increasing complexity until the limitations are clearly defined and documented.

c) In developing an efficient program, consideration must be given to computational algorithms, error generation and propagation, storage requirements, running time, and input data coding time.

d) The class of problems to be solved by a given subroutine must be clearly defined and documented.

Some specific suggestions for implementing a qualified subroutine are outlined below.

2. Program Compatibility and Universality. All programs should be written in Fortran IV to assure minimum dependence on individual computer characteristics. Any deviations from procedures as described in "A Guide to Fortran IV Programming" by Daniel D. McCracken, Wiley, 1965, should be clearly specified in the documentation.

A program should be run on several different computers during the qualification procedure to test for adverse machine-dependent characteristics. The originator should supply to the user a Fortran listing from the operating program, and a duplicate Fortran deck. It is important that the listing and deck be dated since modifications will occur periodically.

3. Documentation. Appropriate documentation must be provided for the user, for the programmer, and for the program itself.

a) User Instructions- This should be a brief document which the user can employ to learn the operation of the program from an input-output viewpoint. Specifically, it should include the date, a brief description of the class of problems that can be solved and the algorithms employed, the program limits (maximum number of nodes, elements, loops, etc.), the machine requirements (words of core, tapes, double precision, etc.), estimated running time and accuracy, a description of input data coding procedures and program options, and one or more illustrative examples.

b) Programmer's Guide- More detailed documentation must be provided to the programmer who will check out the routine on a new computer, interface it with other subroutines, and modify it to meet particular machine or user requirements. The Programmer's Guide should include all of the information listed under User Instructions plus a detailed description of the computing algorithms, flow charts, more extensive example problems with solutions, and a listing of machines on which the program has run successfully. Any machine-dependent features should be noted. References to the literature should be cited where applicable.

c) Program Documentation- The Fortran program should be dated, and should contain sufficient comment cards to allow the programmer to correlate the Fortran listing with the flow charts given in the Programmer's Guide.

4. Implementation of Qualification Procedures. The preceding paragraphs have defined a "qualified" subroutine and its documentation. The following steps are suggested for obtaining a fully qualified computer program:

a) The originator of the subroutine will develop a preliminary program and documentation in accordance with the criteria and suggestions of the preceding sections.

b) The originator will also develop and run several example problems of increasing complexity to test the program and to determine its limitations.

c) The originator will then send copies of his program with the test problems and preliminary documentation to the other NASAP participants.

d) The other NASAP participants will run the program on their machines, and make suggestions for revisions based upon their experiences.

e) The originator will prepare the final qualified program and documentation on the basis of the accumulated experience and recommendations of all NASAP participants.

Please direct comments and suggestions on these qualification procedures to:

Eugene W. Henry, Assoc. Prof.
Department of Electrical Engineering
University of Notre Dame
Notre Dame, Indiana 46556

Phone: (219) 284-6408

The NASAP Graduate Assistants at the University of Notre Dame are:

Edwin V. Harrington
William F. Hartman
William F. Rombalski

APPENDIX B

EXAMPLE PROBLEM FROM NETWORK THEORY EXAMINATION

EE 212 Network Synthesis Examination No. 3 May 12, 1967

Consider an RC two-port network driven by an ideal voltage source E_1 and terminated by a 1 ohm resistor. The desired transfer function is given as

$$E_2/E_1 = \frac{K\ (s^2+1)}{(s+1/2)(s+2)(s+5)}$$

a) Obtain a symmetrical lattice realization, with the maximum possible gain constant K.

b) Obtain an unbalanced symmetrical realization by the Ozaki procedure, using the same gain constant.

c) Obtain a parallel-ladders realization. Compute the gain constant achieved.

d) Obtain a cascade realization by the Dasher procedure, and compute the gain constant achieved. (see Guillemin, "Synthesis of Passive Networks", Wiley, 1957.)

e) For each realization tabulate the number of resistors, the number of capacitors, the total range of element values, and the value of K. Compare and discuss these realizations.

f) For each realization code the problem for analysis on the digital computer by the NASAP program, and use the computer to verify that your realizations do produce the desired transfer function. Include the computer results in your paper.

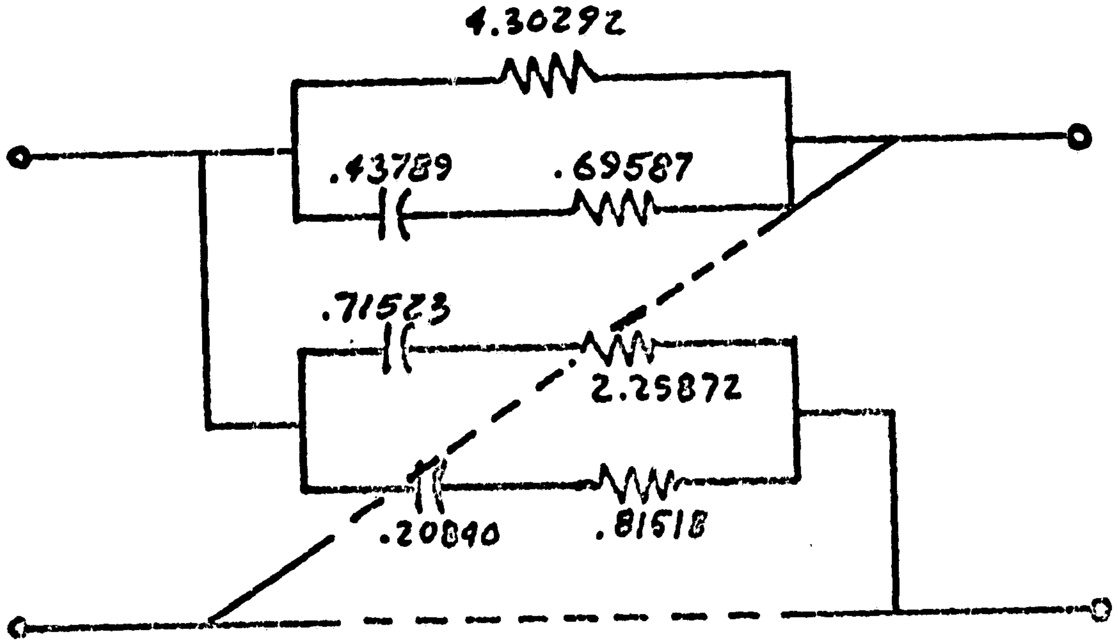
EXTRA CREDIT

a) Obtain a Fialkow-Gerst realization with the maximum possible gain constant.

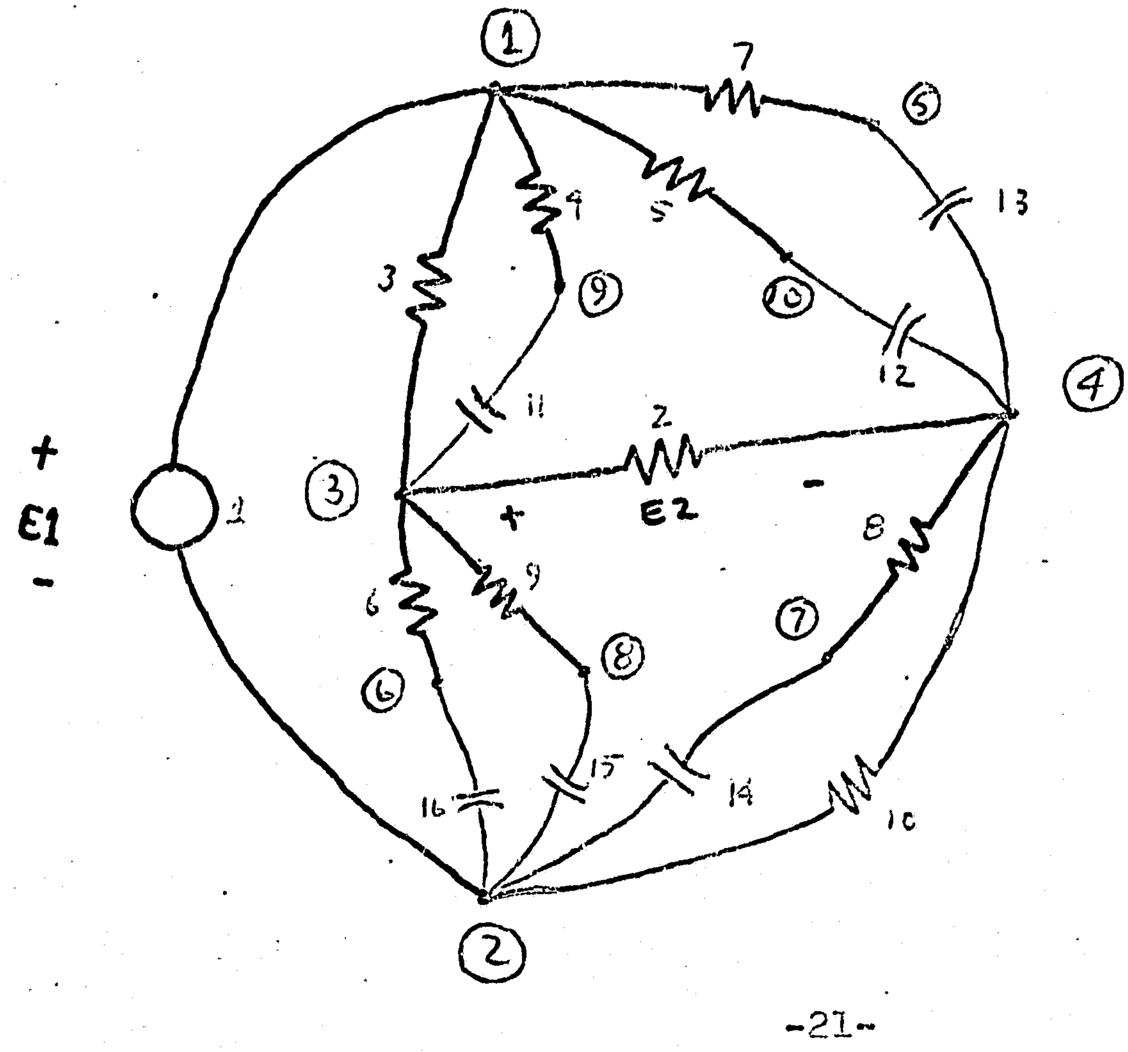
b) Use NASAP to investigate the sensitivity of the transfer function with respect to network capacitances.

c) Obtain an active RC network realization.

Symmetrical Lattice Realization :



Coding for NASAP



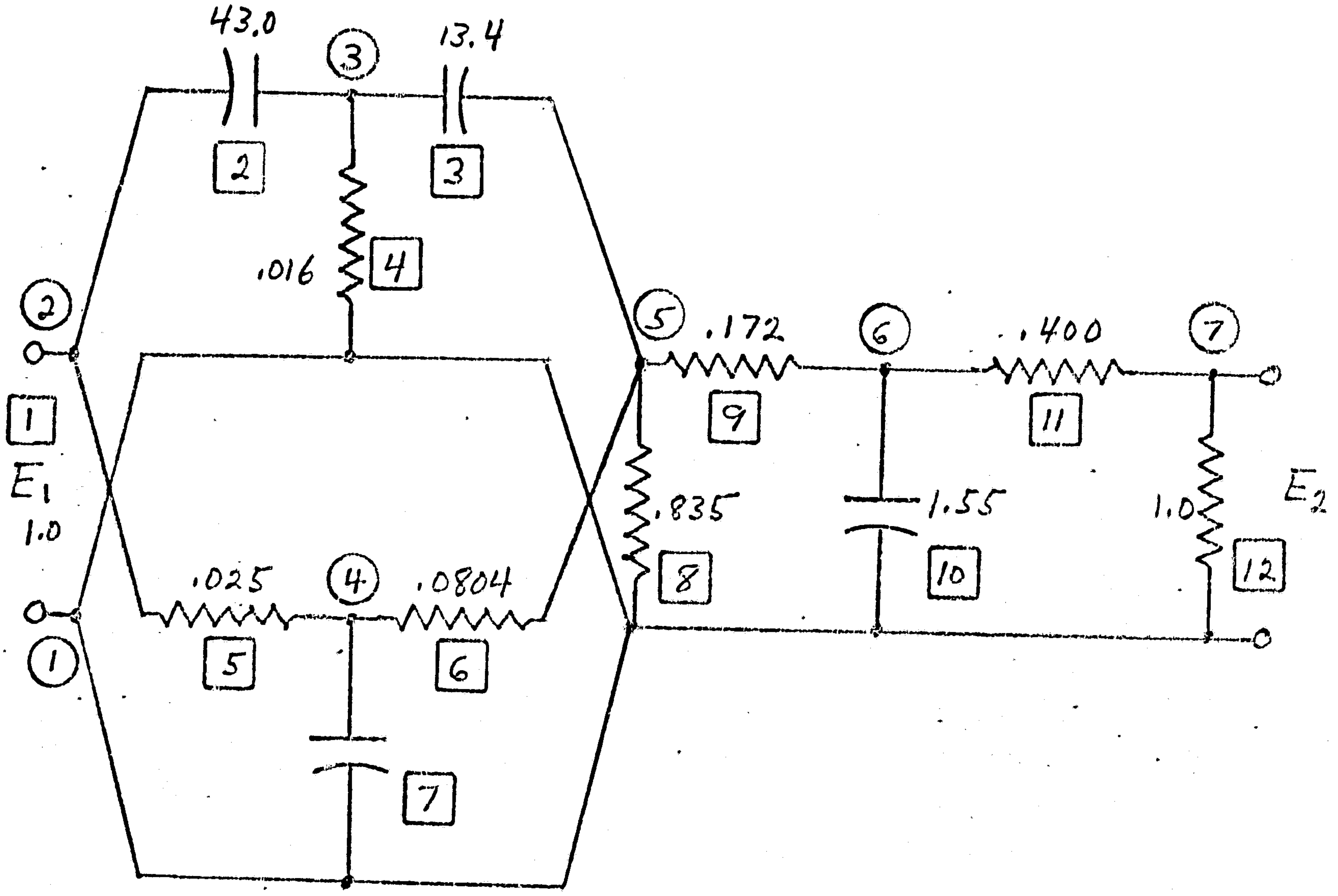
```
INPUT MATRIX BY COLUMNS
 1  2  0  2  1  0  0  1  0        .100+01
 4  3  1  2  2  0  0  0  0        .100+01
 1  3  1  3  3  0  0  0  0        .430+01
 1  9  1  4  4  0  0  0  0        .695+00
 1 10  1  5  5  0  0  0  0        .815+00
 3  6  1  6  6  0  0  0  0        .225+01
 1  5  1  7  7  0  0  0  0        .225+01
 4  7  1  8  8  0  0  0  0        .695+00
 3  8  1  9  9  0  0  0  0        .815+00
 2  4  0 10 10  0  1  0  0        .232+00
 9  3  0 11 11  1  1  0  0        .437+00
10  4  0 12 12  1  1  0  0        .208+00
 5  4  0 13 13  1  1  0  0        .715+00
 2  7  0 14 14  1  1  0  0        .437+00
 2  8  0 15 15  1  1  0  0        .208+00
 2  6  0 16 16  1  1  0  0        .715+00
```

```
    TRANSFER FUNCTION

    GAIN =-.25325-06

     NUMERATOR
0000+01S** 6 -.20553+07S** 5 -.12599+08S** 4 -.15652+08S** 3 -.14309+08S** 2 -.13597+08S** 1 -.17106+07S** 0


     DENOMINATOR
0000+01S** 6  .13630+02S** 5  .66090+02S** 4  .13820+03S** 3  .12620+03S** 2  .44314+02S** 1  .41615+01S** 0
```

```
ATTEMPT TO CANCEL COMMON FACTORS
OF NUMERATOR AND DENOMINATOR.

 GAIN =      .52050

 FACTORS OF NUMERATOR POLYNOMIAL --

     FACTORIZATION OF POLYNOMIALS BY PLT SUBROUTINE POLROT


      COEFFICIENTS OF DECREASING POWERS OF UNKNOWN

         1.00000     6.13001    7.60230   6.96200   6.61558     .83229

      BERNOULLI CONVERGENCE LIMIT --  .010
      NEWTON-RAPHSON CONVERGENCE LIMIT --  .00000100

                 ROOTS
      REAL PARTS          IMAGINARY PARTS
        -4.78698 ✓               .00000
        -1.19946 ✓               .00000
          .00060                1.00061   }
          .00060               -1.00061   } ≈ s^2+1
         -.14477 ✓               .00000

 FACTORS OF DENOMINATOR POLYNOMIAL --

     FACTORIZATION OF POLYNOMIALS BY PLT SUBROUTINE POLROT


      COEFFICIENTS OF DECREASING POWERS OF UNKNOWN

        1.00000    13.63000  66.09000 138.19999 126.20000  44.31400   4.16150

      BERNOULLI CONVERGENCE LIMIT --  .010
      NEWTON-RAPHSON CONVERGENCE LIMIT --  .00000100

                ROOTS
      REAL PARTS          IMAGINARY PARTS
        -4.99828                 .00000     — ~ s^2 + 5
        -4.78547 ✓               .00000
        -1.99927                 .00000     — ~ s + 2
        -1.20218 ✓               .00000
         -.50003                 .00000     — ~ s + 1/2
         -.14477 ✓               .00000

 COMMON FACTORS
   -4.78547 +J ✓   .00000
   -1.20218 +J ✓   .00000
    -.14477 +J ✓   .00000
```

DASHER REALIZATION, SCHEMATIC AND COMPUTER INPUT

```
INPUT MATRIX BY COLUMNS
1  2  0 12  1  0  0  1  0        .100+01
2  3  0  2  2  1  1  0  0        .430+02
3  5  0  3  3  1  1  0  0        .134+02
3  1  1  4  4  0  0  0  0        .164-01
2  4  0  5  5  0  1  0  0        .250+00
4  5  1  6  6  0  0  0  0        .804-01
4  1  0  7  7  1  1  0  0        .486+02
5  1  1  8  8  0  0  0  0        .835+00
5  6  1  9  9  0  0  0  0        .172+00
6  1  0 10 10  1  1  0  0        .155+01
6  7  0 11 11  0  1  0  0        .400+00
7  1  1 12 12  0  0  0  0        .100+01
```



DASHER REALIZATION, MASAP OUTPUT:

```
               TRANSFER FUNCTION

               GAIN = .26624+01

                NUMERATOR
.10000+01S** 3  .10779+01S** 2  .10000+01S** 1  .10779+01S** 0

                DENOMINATOR
.10000+01S** 4  .85780+01S** 3  .21581+02S** 2  .19544+02S** 1

.53855+01S** 0
```

This result may be written as :

$$E_2/E_1 = \frac{2.6624(s^2 + 1)(s + 1.0779)}{(s^3 + 7.5s^2 + 13.5s + 5)(s + 1.0779)}$$

This yields the desired transfer function when the common factor is removed.