evolutionary processes have favored the maximization of conduction velocity. These assumptions yield an optimum inside/outside diameter ratio of 0.618, and a node-to-node transit time of 14.2 μs, signal loss of 4.6 dB, and spacing of 85 outside diameters.

Acknowledgment

The author wishes to thank Prof. E. W. Lewis of the University of California, Berkeley, who called attention to the work done by previous authors and contributed many other helpful comments.

References

[1] A. C. Guyton, *Medical Physiology*, 3rd ed. Philadelphia, Pa.: Saunders, 1966, p. 67.

[2] T. C. Ruch, H. D. Patton, J. W. Woodbury, and A. L. Towe, *Neurophysiology*, 2nd ed. Philadelphia, Pa.: Saunders, 1965, p. 81.

[3] S. Deutsch, *Models of the Nervous System*. New York: Wiley, 1967.

[4] G. Falk and P. Fatt, "Linear electrical properties of striated muscle fibres observed with intracellular electrodes," *Proc. Roy Soc.* (London), ser. B, vol. 160, pp. 69–123, April 1964.

[5] C. F. Stevens, *Neurophysiology: A Primer*. New York: Wiley, 1966.

[6] R. E. Taylor, "Cable theory," in *Physical Techniques in Biological Research*, vol. 6, W. L. Nastuk, Ed. New York: Academic Press, 1963.

[7] T. C. Ruch *et al.*, *op. cit.*, p. 78.

[8] S. Ochs, *Elements of Neurophysiology*. New York: Wiley, 1965, p. 30.

[9] W. A. H. Rushton, "A theory of the effects of fibre size in medullated nerve," *J. Physiol.*, vol. 115, pp. 101–122, 1951.

[10] T. C. Ruch *et al.*, *op. cit.*, p. 65.

[11] A. L. Hodgkin and W. A. H. Rushton, "The electrical constants of a crustacean nerve fiber," *Proc. Roy. Soc.* (London), ser. B, vol. 133, pp. 444–479, 1946.

[12] T. C. Ruch *et al.*, *op. cit.*, p. 67.

[13] *Ibid.*, p. 61.

[14] E. Weber, *Linear Transient Analysis*, vol. 2. New York: Wiley, 1956.

[15] W. A. Rosenblith, *Sensory Communication*. Boston, Mass.: M.I.T. Press, 1961.

[16] T. C. Ruch *et al.*, *op. cit.*, p. 30.

[17] S. Ochs, *op. cit.*, p. 62.

[18] I. Tasaki and K. Frank, "Measurement of the action potential of myelinated nerve fiber," *Am. J. Physiol.*, vol. 182, p. 573, 1955.

[19] I. Tasaki, "New measurements of the capacity and the resistance of the myelin sheath and the nodal membrane of the isolated frog nerve fiber," *Am. J. Physiol.*, vol. 181, p. 639, 1955.

[20] R. Fitzhugh, "Computation of impulse initiation and saltatory conduction in a myelinated nerve fiber," *Biophys. J.*, vol. 2, pp. 11–21, 1962.

[21] A. L. Hodgkin and A. F. Huxley, "A quantitative description of membrane current and its application to conduction and excitation in nerve," *J. Physiol.*, vol. 117, pp. 500–544, August 1952.

[22] T. C. Ruch *et al.*, *op. cit.*, p. 45.

[23] J. B. Hursh, *Am. J. Physiol.*, vol. 127, pp. 131–140, 1939.

Correspondence

Modern Design Methods for Electronics

Abstract—**Widely used methods, such as copy, modify, cut and try, and graphical and mathematical analysis, tend to focus on circuits previously created. Mathematical synthesis of circuits is very limited in application. The synthesis of circuits or large-scale systems can be aided by a methodology called the engineering design process. To obtain the physical reality of an electronic design, so many decisions are needed that data reduction methods are required. Computer programs can simplify decision making by the analysis of interaction matrices. From architecture we get Alexander's HIDECS, and from psychology we get factor analysis programs. Their use and misuse is illustrated by the application of their rationales to the combining of subsystems of a color television receiver.**

Manuscript received February 2, 1968; revised June 4, 1968.

Introduction

This correspondence shows how the engineering design process fills the synthesis role in design. It also deals with the use of computer programs to aid in the simplification of decision making.

Widely Used Electronic Design Methods

Electronic designers use many methods to help achieve a solution to a design problem. Because they are often working on the frontiers of knowledge, formal routines do not exist for solving many circuit problems. Much cut-and-try experimentation is done. Table I is a summary of these methods. Most designs are a combination of these methods.

The recent advent of computer programs [1], [2] for circuit analysis may move the cut-and-try method from the breadboard to the computer. Once a basic concept is established, the designer can get rapid results of many trial values of components without the need to simplify and analyze the circuit himself.

The real difficulty in electronic design is in the initial devising or creating of the basic concept.

Synthesis of circuits is the putting together of components in a meaningful manner to achieve a satisfactory result. Mathematical synthesis of active and passive networks can be done if the transfer function can be expressed as a rational function. However, if one must also optimize for cost, weight, size, etc., one ends up with a problem still requiring creative effort.

The author submits that most electronic design is done without either mathematical synthesis or mathematical analysis. Synthesis of circuits is largely an "art." However, even this art has been studied and has been reduced to a systematic procedure, primarily in the field of mechanical engineering design [3].

Engineering Design Process

This process is illustrated briefly in Fig. 1. An open-end problem is one where there are many satisfactory solutions. Most design problems are open-end. Synthesis of a

TABLE I

Widely Used Electronic Design Methods

Method	Example	Comment
Copy	A circuit is copied from a manual, textbook, competitor's product, etc.	Capitalizes on experience of others and avoids "reinventing the wheel"
Modify	Change the power supply voltage on a copied circuit and change the other components to suit	A shortcut that calls for understanding of the circuit to be modified
Cut and try	Make a breadboard hookup and vary components while observing the result	Experienced designers know what to vary. This is not a fast method for a novice
Graphical and mathematical analysis	Graphical analysis of transistor curves to get operating point and distortion. Circuit analysis to predict small signal gain or transient analysis	This is what engineers are usually taught to do. Computer programs may reduce need for mathematical ability required to do circuit analysis
Mathematical synthesis	Define transfer function to relate input and output as a rational function, find roots, and derive equivalent circuit	Limited to special cases which can be expressed as mathematical functions

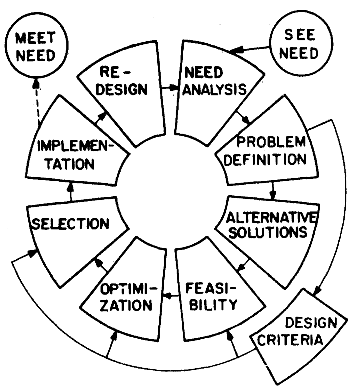


Fig. 1. Engineering design process.

solution can be reduced to the following steps [4], [5].

1) Need Analysis: Obtain sufficient information to determine the scope, objectives, and background of a project before proceeding with solution finding.

2) Definition of the Problem: Define the scope of the objectives in a general way, (e.g., performance, size, weight, etc.) with the major restraints, (e.g., cost and time) such that they are related to the need.

3) Development of Design Criteria: Establish and rank, in advance of finding solutions, sufficient quantitative criteria for judging and evaluating alternative solutions to the design problem. They may be tangible (like brightness) and intangible (like appearance).

4) Development of Alternative Solutions: Develop a range of alternative solutions, including known solutions and newly created ones.

5) Analysis of Feasibility: Analyze the reasonable alternative solutions for their physical, economic, and financial feasibility (e.g., technical feasibility, cost of producing, financing of tooling, and capital costs).

6) Optimization: Optimize the most likely solutions with respect to the principal function and the various tradeoffs required by the design criteria.

7) Selection of the Solution: Examine both good and bad outcomes of each optimized alternative. Use the established design criteria.

8) Implementation and Communication: Produce specifications, instructions, and prototypes which are understandable to those who will use them.

9) Transfer to More Detailed Phase: Recognizing that the engineering design process is cyclic and iterative in nature, repeat all steps in more and more detail until the detailed design meets the need within the established criteria for probability of success.

Decision Making

Throughout the engineering design process, decisions must be made at every step. Some of these are: What are the needs? What is the problem? What are the criteria? Which alternatives to use? How to implement the design? etc. These involve tradeoffs in the areas of time, cost, performance, space, weight, reliability, and so forth. Unfortunately, the decisions on tradeoffs are not clear-cut because of their interaction. That is, a decision on one performance criterion may involve tradeoffs with other criteria, such as cost, time, reliability, and so forth.

The human limitation in decision making is about seven related items. Beyond this, some memory assistance is necessary. This is usually in the form of a tabular array or a matrix.

The quantity of decisions is important. For a simple transistor coupling stage involving only six components, some thirty decisions are needed to define component values, tolerances, ratings, and locations. Fortunately, experience provides shortcuts to decisions such as these. However, when one looks at a practical design of any product like a television set, the decisions run into thousands. Allowing that many of these are trivial, even if a hundred remain, it is difficult to fathom. Both the number of decisions and their interactions can be shown in a symmetrical matrix like the one in Fig. 3. The interactions can be binary, i.e., yes or no, or they can be weighted, i.e., 1, 2, 3.

The lack of knowledge concerning what to do with such a matrix has prevented their use except for small matrices which could be studied visually.

The problem of decisions in design is made simpler if the components (or subsystems) are grouped into subproblems based on some rationale. Two grouping methods are described here.

1) Relationships: the grouping of components (or subsystems) by the density of interactions (HIDECS program).

2) Similarities: the grouping of components (or subsystems) by the similarities of their interactions with other parts of the system (factor analysis program).

Grouping data by similarities is known as a classificatory scheme wherein the data is grouped into categories by their similarities. If the categories are mutually exclusive, it is called a nominal scale. On the other hand, data can be grouped by the relationships between the elements. This is sometimes called a "systems" approach. Ideally one seeks to form mutually exclusive subsystems which do not interact with each other, but, as the experienced designer knows, such is seldom the case.

Alexander's Method of Grouping by Relationships

An architect and a mathematician, Alexander [6], uses graph theory to reduce the matrix to clusters. The rationale may be described as follows.

Suppose a set of variables interact (or require decisions) as in Fig. 2. In Alexander's terms, the points represent the misfit variables and the links represent the interactions. (For our purposes we will consider the interacting decisions as equivalent to the misfit variables.) He would have us first design those subsystems (or components) which have many interactions and then combine them at a new hierarchical

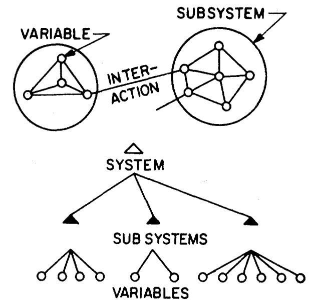
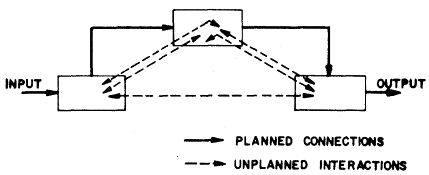


Fig. 2. Alexander's form of design.



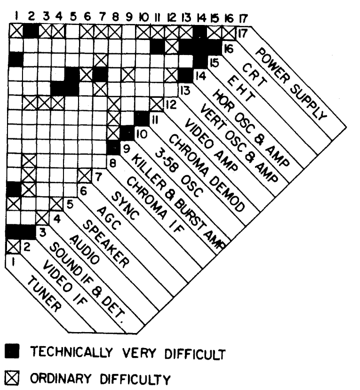


Fig. 3. Interaction matrix of subsystems of color television.

level, where the interactions are less. These subsystems are in turn combined into other subsystems by the same rule until the whole system is complete.

An interesting example of the application of this was in the theoretical design of a village for India. There were 141 misfit variables as widely different as "cattle treated as sacred," "vegetarian attitude," and "insufficient forest land." A computer program called HIDECS [7] was devised to sort the variables into meaningful groups.

To illustrate the use of Alexander's method, the author has used its rationale to work out a logic for the order of combining groups of subsystems into a whole system. A color television receiver was considered to be a system composed of seventeen subsystems, defined in the usual textbook fashion. In addition to the functional interconnections between the systems, there are the unplanned interactions causing such things as hum, jitter, oscillations, hunting, etc. These unplanned interactions occur when the subsystems are assembled into some three-dimensional array to make a real system. These interactions are responsible for the difficult design decisions which must be made as a consequence. The significant unplanned interactions between the subsystems are shown in Fig. 3.

Superimposed on this is a simpler matrix of what is considered to be the technically difficult interactions for a tube-type color television. A graphical reduction of the matrix was found to be impractical.

The result of the HIDECS 2 decomposition of the matrix into a "design tree" is shown in Figs. 4 and 5. Running time on an IBM 7040 computer was sixty-nine seconds. The author has been involved with television receiver design for sixteen years and more recently participated in the design of two different color television receivers. In the author's view the groupings of the subsystems shown in Figs. 4 and 5 are entirely consistent with good practice in color television design. This indicates that the grouping of subsystems by the density of their undesirable interactions makes good sense.

The matrix was also analyzed by programs SIMPX and EQCLA out of the HIDECS 3 package with rather unsatisfactory results on the 17 $\times$ 17 matrix. This was due to the subsystem number 17 (the power supply) interacting with almost everything else. Such elements as this, which interact with almost everything, are best left out of the matrix. By dropping number 17, a 16 $\times$ 16 matrix was formed. The result of the SIMPX decomposition into triangles was entirely satisfactory and graphical analysis of them gave the same design tree as a HIDECS 2 program. This was essentially the same as Figs. 4 and 5, omitting only number 17. A weakness of the HIDECS 2 program is the progressive subdivision into two parts, when occasionally a split into three or more parts would be more logical.

The HIDECS program is designed to work with a symmetrical matrix as large as 250 $\times$ 250 (but less if the interactions are dense). However, the interactions must be binary, and it requires considerable manipulation of the variables to get almost equal interactions. Different strengths of interactions can be handled by separate matrix analyses as shown in the example.

Factor Analysis

From the field of experimental psychology comes the method of factor analysis [8]. This is a method for determining the number and nature of the underlying variables among large numbers of measures.

The rationale is that an $n \times n$ matrix can be represented by points in n-dimensional space. Clusters of points represent "factors." When applied to symmetrical interaction matrices, the rows are grouped according to whether their patterns are similar. That is, they are similar in interactions to some hypothetical factor. These factors can then be studied to find meaningful underlying reasons for the resulting decomposition of the matrix.

A factor analysis program can be used with weighted interactions on a scale of 100 units since they were originally designed to handle correlations between psychological tests.

An interesting example of the use of factor analysis was the discovery that intelligence test scores were based on seven underlying factors such as perception, word fluency, memory, and so forth. These

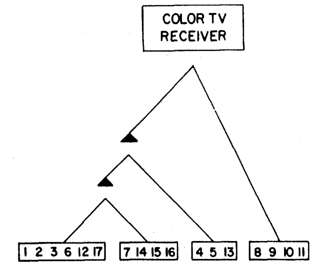


Fig. 4. Form of design for combining subsystems of color television receiver using Alexander's method.

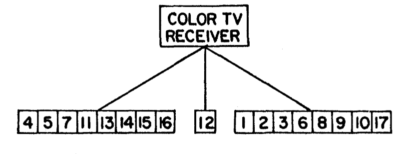


Fig. 5. HIDECS grouping of subsystems by technically very difficult interactions only.

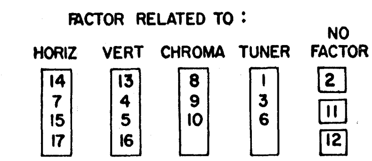


Fig. 6. Factor analysis grouping of subsystems by technically very difficult interactions only.

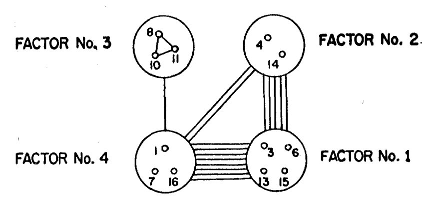


Fig. 7. Factor analysis grouping of subsystems by all interactions.

factors were identified by further study of the hypothetical factors resulting from the analysis.

A recent application of factor analysis to a design problem was done by Wright and Aasen at the University of Waterloo, Ont., Canada. After four months on the MacKenzie River Delta, as part of a team of architects, engineers, anthropologists, and sociologists, Wright [9] identified ninety functional components for the town of Inuvik. The interactions were identified on a 90 $\times$ 90 matrix. By a factor analysis he was able to group them into nine meaningful combinations.

The decomposition of the simpler 17 $\times$ 17 matrix with only the technically very difficult interactions is shown in Fig. 6. The four factors found make some sense as far as a color television design goes, but misses an important grouping of the chroma demod (subsystem 11) in the "chroma related" factor.

The whole nature of the factor analysis decomposition is better understood from the results of the more complex matrix. These are explained by Fig. 7. Here, in factor 3, the cluster of subsystems 8, 10, and 11 have interactions with each other but no organized pattern of interactions with the rest of the system. On the other hand, factor 4 has a cluster of subsystems 1, 7, and 16 (they have no interactions to each other) which are similar in their pattern of interactions with the other factors. This is a pure "similarities" grouping which turns out to be a pure "nonsense" grouping. If used to form a "tree of design," it would group the tuner, sync, and cathode-ray tube subsystems, which is not a useful grouping. Factor 4 can only be described as that factor which has eight interactions with factor 1, two with factor 2, and one with factor 3. It is clear that the factor analysis program has reduced the decision matrix to organized groupings, but in this case it would be of little value to a designer.

A factor analysis is capable of finding factors which contain interrelated elements as well as those factors which are caused by similarities of patterns. Each factor should be considered as an hypothesis, which can be used for further analysis in attempts to find groups that are significant to the design problem.

Factor analysis programs can be used on matrices as large as 90 $\times$ 90 and the interactions can be weighted. They require more postcomputer analysis than HIDECS and more computer running time also.

Summary

The engineering design process is a systematic way of synthesizing a design solution. With regard to the simplification of decision matrices, the HIDECS groups by the system principle of relationships and the factor analysis groups by a classification principle of similarities.

The relationship principle gave the more useful result when applied to the order of combining subsystems of a color television receiver.

What use can be made of the technique of determining the order of combining subsystems? In the past, this has been decided by whim and fancy or controlled by factors external to the design, such as cost, time, and personnel. The design itself contains the logic for such planning. One can use the relationship principle to determine the "ideal" plan for groupings of subproblems. Then external factors can be applied to modify the ideal to meet the exigencies of the situation.

Acknowledgment

The Department of Design at the University of Waterloo, Ont., Canada, provided the environment which permitted this exploratory study. In particular, the author wishes to thank R. S. Frew for his assistance with HIDECS and Prof. J. B. Ellis, S. W. Wright, and C. Aasen for their assistance with factor analysis.

S. F. Love
Dept. of Design
University of Waterloo
Waterloo, Ont., Canada

References

[1] J. E. McTaggart, "A high performance transistor IF amplifier for hybrid television receivers," *IEEE Trans. Broadcast and Television Receivers*, vol. BTR-10, pp. 60–65, November 1964.
[2] D. G. Mark, "Computer aided design," *IEEE Trans. Broadcast and Television Receivers*, vol. BTR-13, pp. 24–35, April 1967.
[3] J. N. Siddall, "Modern theories in design techniques," *1967 Proc. 2nd Ann. Design Seminar* (Hamilton, Ont., Canada). Also, in *Prod. Design and Value Engrg.*, pp. 24–26, September 1967.
[4] V. K. Handa, P. H. Roe, and G. N. Soulis, *The Discipline of Design*. Boston, Mass.: Bacon & Allyn, 1967.
[5] S. F. Love, "In-plant training of practicing engineering designers in the latest methodology of design," *1967 Proc. 4th Nat'l Conf. on Engineering Design and Design Education* (Hanover, N. H.).
[6] C. Alexander, *Notes on the Synthesis of Form*. Cambridge, Mass.: Harvard University Press, 1966, pp. 136–173.
[7] "HIDECS 2" and "HIDECS 3," Dept. of Civil Engineering, Massachusetts Institute of Technology, Cambridge, Res. Repts.
[8] P. Horst, *Factor Analysis of Data Matrices*. New York: Holt, Rinehart and Winston, 1965.
[9] J. B. Ellis, S. W. Wright, and C. Aasen, "A component to system construct for settlement planning," *EKISTICS*, vol. 25, pp. 55–59, January 1968.
[10] S. W. Wright, "FACPAC factor analysis program for IBM 7040 computer," Dept. of Design, University of Waterloo, Ont., Canada, 1967.

Polynomial Representation of Finite-State Machines

One of the points implicitly made in Wymore [1] is that systems, in general, possess an algebraic structure which is the same no matter what specific mathematical description is chosen to represent the algebraic structure. Nowhere can this be more clearly illustrated than by finite-state machines. The literature on finite-state systems abounds with different forms for describing them: state flow graphs, state flow tables, linear machines, etc. However, there is a well-known algebraic structure common to all of them which will be presented immediately below. In this correspondence, another mathematical description of a finite-state machine will be developed.

The following definition, found in Hartmannis and Stearns [2], gives Moore's version of the algebraic structure of a finite-state system.

Definition 1: A finite-state system is a quintuple

$$M = (S, I, O, f, g)$$

which possesses the following properties:

S is a finite set not empty
I is a finite set not empty
O is a finite set
$f: S \times I \rightarrow S$
$g: S \rightarrow O.$

For the sake of discussion, assume that S, I, and O are disjoint subsets of real numbers. Thus S consists of n distinct real numbers, I consists of m distinct real numbers, and O contains r distinct real numbers. Now inquiry is made as to a possible representation for the system functions f and g. By definition, f maps $S \times I$ into S. But since it is assumed that S and I are made up of real number elements, a geometric interpretation of this function can be obtained by plotting S versus values of the pairs (s, i) where $s \in S$, $i \in I$.

Viewing the geometry of Fig. 1, the problem of determining the function f is the same as specifying a surface which passes through the proper points in a three-dimensional space. Because finite numbers of elements are involved in all sets, this can be easily done by polynomial interpolation, as will be shown later. By a similar argument, the function g can be viewed as passing a curve through a specified set of points, with S as the domain and O as the range (see Fig. 2). The core of all the work to be done rests on this simple geometric interpretation. In the following definition a bit of notation will be used that bears explaining. If a function a is defined on a set A with values in a set B and is one-to-one and onto, then it is said that A is isomorphic (a) to the set B, and B is isomorphic (a^{-1}) to the set A. This notation is due to Wymore [1]. In the following, all symbols for polynomial functions begin with a small letter p. Thus pf is a polynomial function. The set of the real numbers is R.

Definition 2: If f is a function defined on the set A with values in B, and C is a subset of A, then the restriction to C of f is $\text{res}(f, C)$ and defined:

$$\text{res}(f, C) = [(x, y): x \in C \text{ and } (x, y) \in f]$$

that is,

$$\text{res}(f, C)(x) = f(x) \text{ for every } x \in C.$$

The necessary symbolism has been established to allow the statement of the following definition.

Definition 3: A real number representation of a finite-state system $M = (S, I, O, f, g)$ is a quintuple of sets $M' = (S', I', O', f', g')$ such that there exists a set of isomorphisms (α, β, γ) with the properties:

a) S is isomorphic (α) to S', a subset of R
I is isomorphic (β) to I', a subset of R
O is isomorphic (γ) to O', a subset of R

b) $f' = \text{res}(pf', S' \times I')$, where pf' is a polynomial defined on R^2 such that

$$\alpha[f(s, i)] = pf'[\alpha(s), \beta(i)] \tag{1}$$

c) $g' = \text{res}(pg', S')$, where pg' is a polynomial defined on R, such that

$$\gamma[g(s)] = pg'[\alpha(s)]. \tag{2}$$

Definition 3 appears quite formidable at first. However, it can be considered a section at a time. Condition a) requires that the sets S', I', O' be related to the sets of the original system by one-to-one mappings [this is important for the purposes of conditions b) and c)]. In condition b) the function f' is defined as the restriction to $S' \times I'$ of a polynomial pf'. This polynomial must be such that f' satisfies (1). The form of (1) is that of an operation preserving mapping. Condition c) is much the same as b). The function g' is the

Manuscript received February 26, 1968; revised July 15, 1968.