



DESIGN AND ANALYSIS OF ELECTRONIC CIRCUITS

E. L. CALVIN, L. C. CASADY,
J. R. EDENHOFER, R. S. MILES
Autonetics Division
North American Aviation, Inc.
Anaheim, California

DESIGN AND ANALYSIS OF ELECTRONIC CIRCUITS

E. L. Calvin, L. C. Casady, J. R. Edenhofer* and R. S. Miles
Autonetics, a Division of North American Aviation, Inc., Anaheim, California

INTRODUCTION

Very seldom does the initial design of an active circuit result in all design constraints being met. Hence, design techniques involve a process of initial design-analysis-redesign-analysis, until the circuit meets the specifications. This paper presents the general philosophy of a computer program which automates the "analysis-redesign" cycle for a large variety of passive or active, linear or nonlinear circuits. The computer program employs both circuit simulation and optimization techniques. Simulation corresponds to the analysis portion of the cycle in that it determines the circuit's response and whether or not the design constraints have been met. The optimization technique alters parameter values to minimize a function which reflects the design constraints. Thus, the optimization procedure performs the redesign.

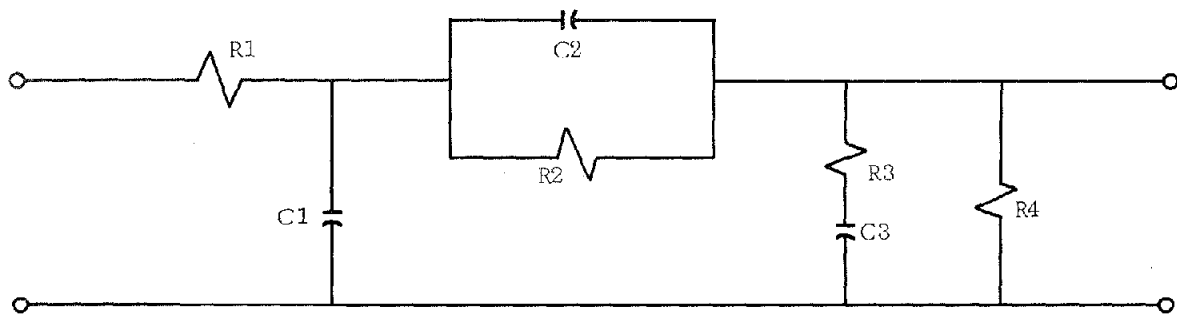
DESIGN FUNCTION

The function used by the optimization techniques will be called the "Design Function" (DF). This function is a means of defining design goals to the computer. These goals are stated by some mathematical combination of the desired circuit response and the computed response. An optimization routine alters circuit parameter values until a minimum of this design function is obtained. It should be noted that successful use of the design function results in the best rather than just satisfactory design. It should also be noted that many design functions may be used to describe the design goals of a circuit. Each of these design functions may be satisfied by a different set of parameter values. Thus, the problem of selecting the best or even a useful design function will tax the ingenuity of both the engineer and programmer. The following two examples are given to clarify the concept of the design function.

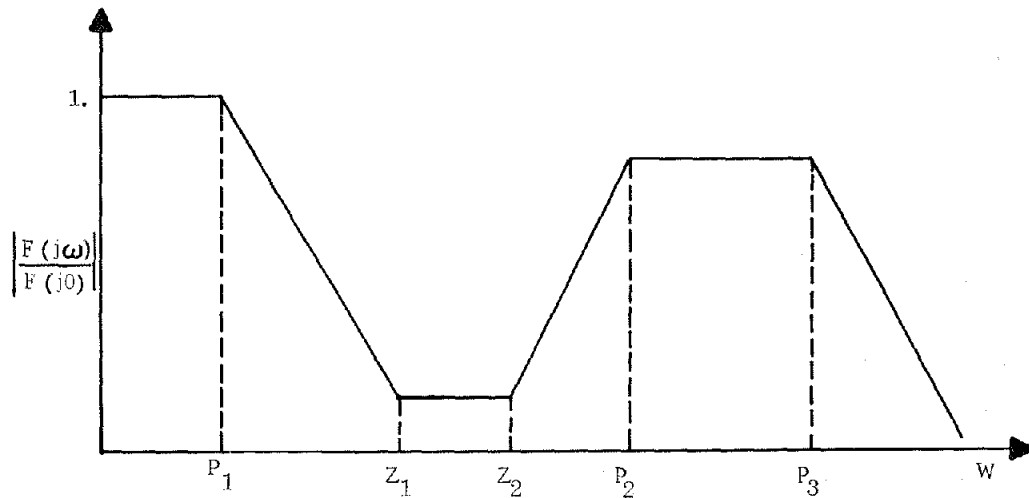
Example 1

Parameter values of the filter network shown in Figure 1a are to be determined such that the desired frequency response of Figure 1b is achieved.

*Presently at Lear Siegler Inc., Grand Rapids



(a)
FILTER NETWORK



(b)
FREQUENCY RESPONSE

Figure 1. Filter Network and Frequency Response

This desired frequency response is specified by a transfer function of the form:

$$F(j\omega) = \frac{(j\omega + z_1)(j\omega + z_2)}{(j\omega + p_1)(j\omega + p_2)(j\omega + p_3)}$$

In the circuit, R1 and R4 represent the source and load impedance and therefore must remain constant. Variable parameters are R2, R3, C1, C2 and C3. Computed circuit response at specified frequencies ω_i will be termed $G(j\omega_i)$. G is also given by the following functional relationship.

$$G = f(R2, R3, C1, C2, C3, \omega)$$

The design function must specify the relationship of the desired frequency response $F(j\omega)$ and the computed response $G(j\omega)$ such that its value becomes smaller as $G(j\omega)$ approaches $F(j\omega)$. One method of specifying a design function for the circuit of Figure 1a is outlined as follows.

1. Determine the dc gain of transfer function:

$$F(j0) = \frac{z_1 z_2}{p_1 p_2 p_3}$$

2. Calculate magnitude and phase angle of transfer function at specified frequencies.

$$\left| F(j\omega_i) \right| = \text{magnitude at } \omega_i$$

$$\text{Arg } F(j\omega_i) = \text{phase angle at } \omega_i$$

3. Determine dc gain of circuit:

$$G(j0) = \frac{R4}{R1 + R2 + R4}$$

4. Calculate magnitude and phase angle of circuit response at specified frequencies.
5. Evaluate design function defined as follows:

$$DF = \sum_{i=1}^n \left| \frac{\frac{F(j\omega_i)}{F(j0)} - \frac{G(j\omega_i)}{G(j0)}}{\frac{F(j\omega_i)}{F(j0)}} \right| + \sum_{i=1}^n \frac{|\text{Arg } F(j\omega_i) - \text{Arg } G(j\omega_i)|}{|\text{Arg } F(j\omega_i)|}$$

It should be noted that many other design functions could also be used. The real and imaginary parts of the response could have been determined and compared or only the magnitudes could have been used. The poles and zeros of the circuit could have been computed and compared with the desired poles and zeros.

Figure 2 and Figure 3 are plots of the amplitude and phase response of this example. Note that three curves are shown; the response based on the desired poles and zeros, the response based on the designers hand calculations, and the response obtained by use of the optimization program and the DF defined previously.

Example 2

The Schmitt trigger of Figure 4 is to meet the following three design constraints:

1. The upper trip voltage should be near 0 volts.
2. The output voltage should be near 2.75 volts when Q2 is on.
3. The hysteresis should be a minimum.

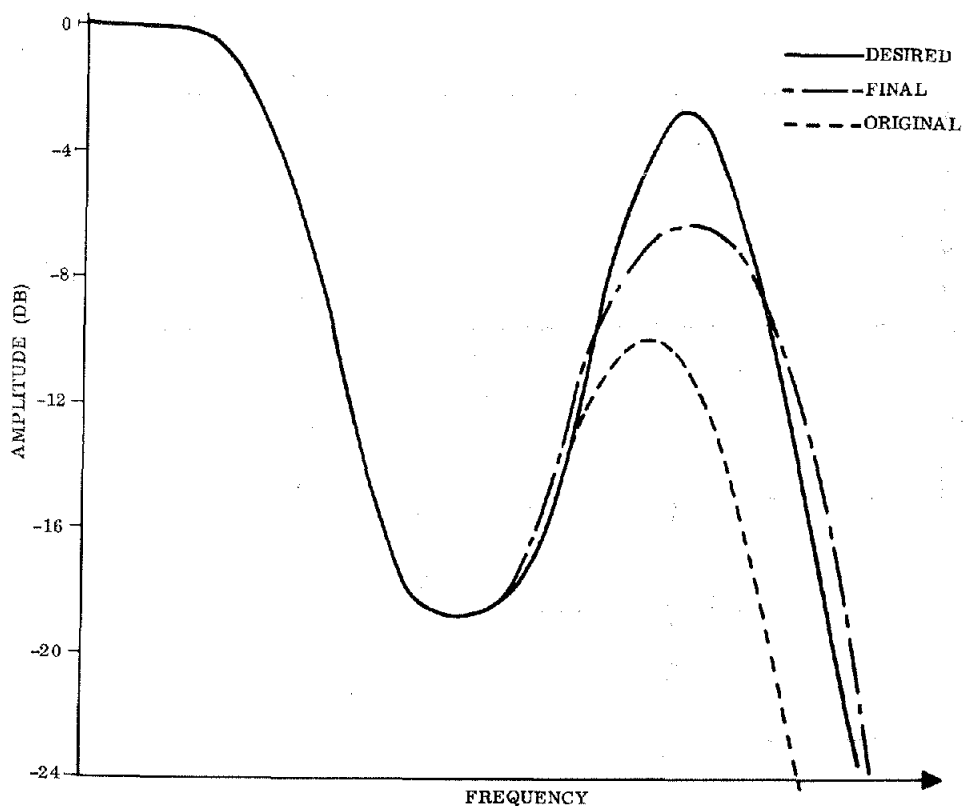


Figure 2. Amplitude Response

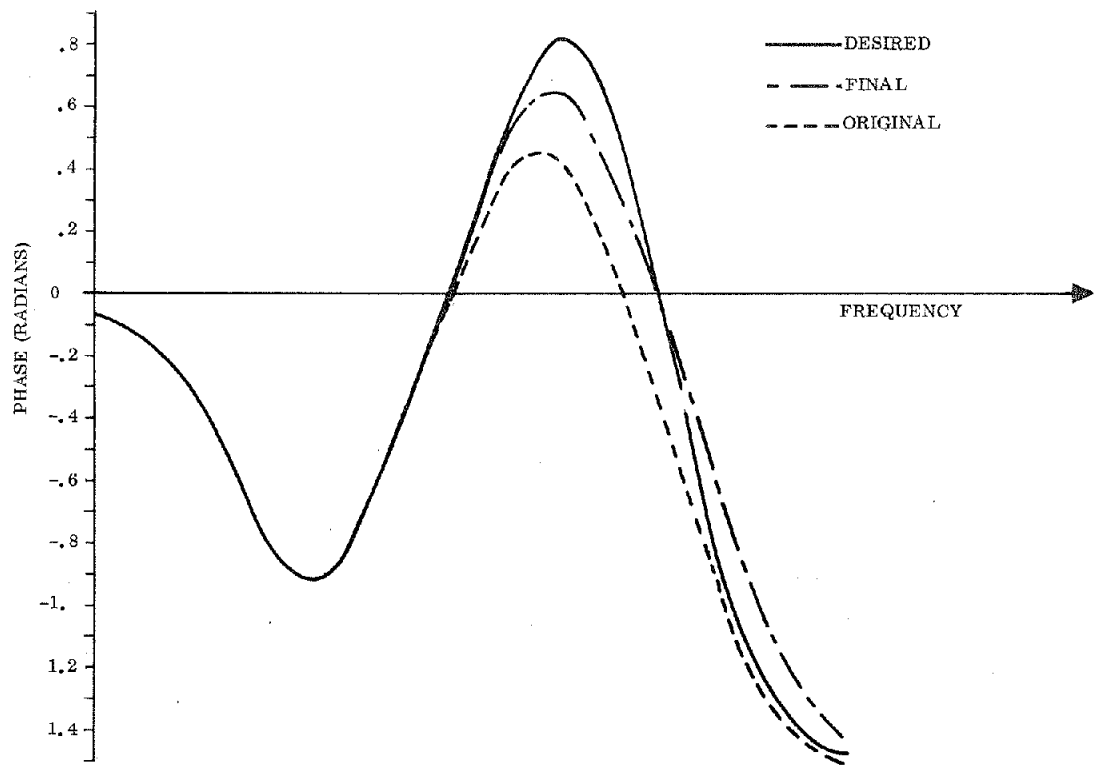


Figure 3. Phase Response

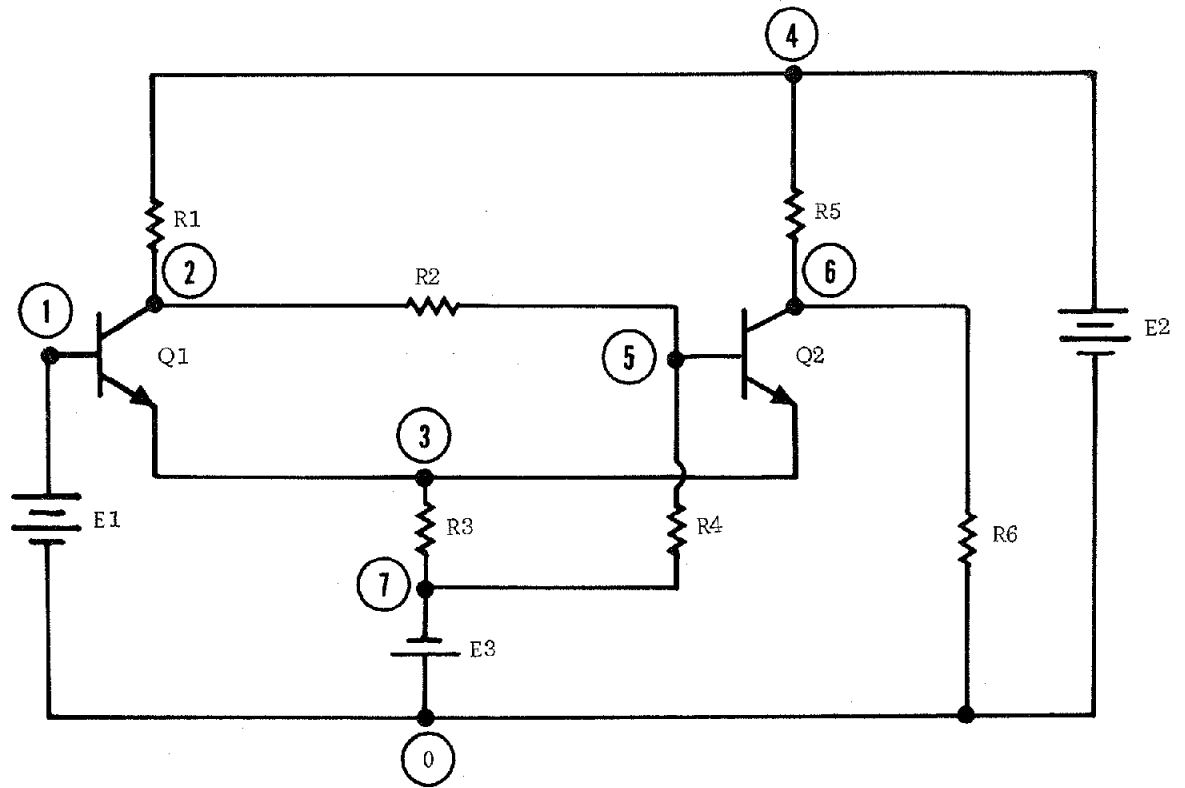


Figure 4. Schmitt Trigger

The design function used for this circuit is defined by the following equations:

$$A1 = |E_{\theta N}|$$

$$A2 = |E_{\theta UT} - 2.75|$$

$$A3 = |E_{\theta N} - E_{\theta FF}|$$

$$DF = A1 + A2 + 1000 \times A3$$

Note how each term of the design function reflects one of the design constraints. The first term is the absolute value of the upper trip voltage. When this term is zero, the first design constraint is met. The second term, when minimized will cause the second design constraint to be met. The third term is the difference between the upper and lower trip voltages. Since this term is considered to be the most important in the DF, it is weighted by a factor of one thousand. Changing the weight factor of the various terms in the design function by a small amount can result in large changes in the resulting design. This points out how critical the definition of the design function is to the optimization process.

The initial value of DF for the original design of this circuit was 1773.0. After modification of all resistors, other than the load, by the optimization routine the value of the design function dropped to 18.9 with design constraints resulting in values of:

$$E_{\theta N} = -0.0625$$

$$E_{\theta UT} - 2.75 = 3.652$$

$$E_{\theta N} - E_{\theta FF} = 0.015625$$

CIRCUIT SIMULATION ROUTINES

The circuit simulation routines perform the analysis function in the "analysis-redesign" cycle. A prime objective in the development of this program is to provide the capability of easily applying the "analysis-redesign" philosophy to as large a variety of circuits as possible. To obtain this objective, a large variety of analysis capabilities is required. In its final form, this program will be able to simulate circuits under d-c, a-c and transient operation. In each of these modes of operation, the program may perform worst case, statistical or parameter variation types of analysis. To implement these analysis capabilities and provide flexibility, the program is being written in modular form. Ease of applying the program has been provided by a user oriented input scheme. The user, be he an engineer with little or no programming experience or an experienced programmer, is being given prime

consideration in all phases of the program development. Data input and result outputs have been organized for the engineer user while the modular program construction will aid a programmer in making modifications or extensions to the program.

USER ORIENTED DATA INPUT

There are three basic types of data which the program accepts, program control data, program execution data, and format data. Program control data will determine such things as the type of circuit operation, the type of analysis desired, the optimization technique desired, required output, etc. Program execution data is that data necessary to define the circuit and its parameters. Format data defines the format by which program execution data and program control data are entered.

All data is read with a free and variable format from one input routine. This routine may be called in at any time to select new data or to read additional data. The variable format capability is obtained by building a large engineering vocabulary into the input routine. With this vocabulary the computer is capable of recognizing data items written in familiar engineering terminology. Table 1 shows how the vocabulary is used to define a format for data input. The example shown is for the Schmitt Trigger circuit of Figure 4.

MODULAR CONSTRUCTION

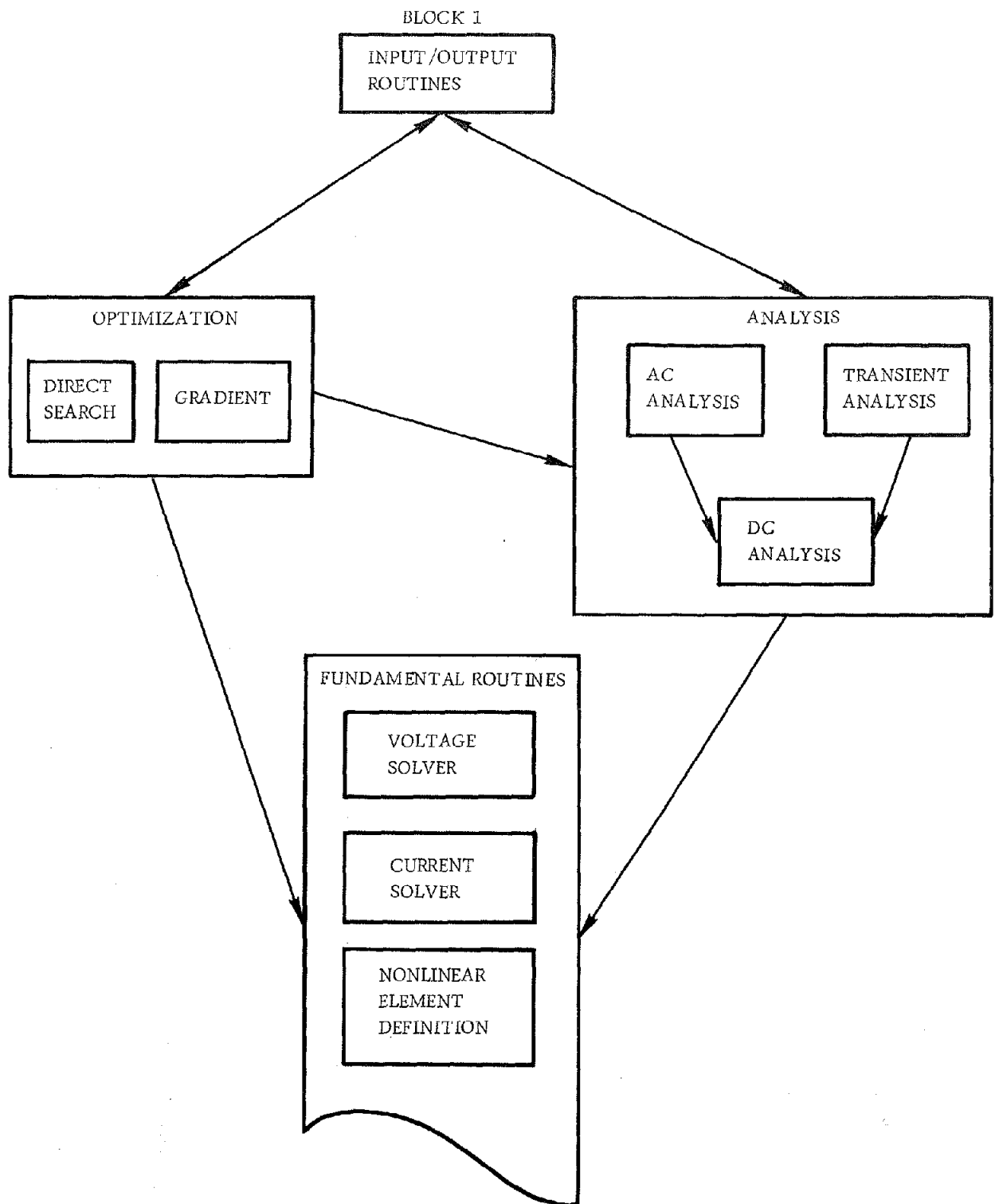
This program represents an attempt to develop a general analysis capability as noted by a wide variety of analyses available. Realizing, however, that it is impossible to generate one program to cover all applications, the modular concept was adopted. This concept tends to minimize the time and effort required to modify or extend the capabilities of the existing program. Four basic blocks of modules are included. These are: the input/output, the fundamental operations, the basic analysis, and the optimization routines. Basic analysis routines have access to the input/output and the fundamental operation routines as well as other basic analysis routines. Optimization routines may access any of the other routines. Figure 5 illustrates this concept.

OPTIMIZATION

In order to realize the minimum value of the design function, a minimization or optimization technique is required. Many optimization algorithms have been developed and tested. Each possess unique advantages. The techniques presently

Table 1. Schmitt Trigger Input Data

F , Name ,	From Node ,	To Node ,	Nom Val ,	Fixed ,	Min Opt ,	Max Opt ,	Step Size
R1 ,	2 ,	4 ,	2K ,	No ,	1K ,	10K ,	1K
R2 ,	2 ,	5 ,	2K ,	No ,	1K ,	10K ,	1K
R3 ,	3 ,	7 ,	700 ,	No ,	100 ,	10K ,	1K
R4 ,	5 ,	7 ,	2K ,	No ,	1K ,	10K ,	1K
R5 ,	6 ,	4 ,	2K ,	No ,	1K ,	10K ,	1K
R6 ,	6 ,	0 ,	10 ,	Yes			
E1 ,	0 ,	1 ,	-1 ,	Yes			
E2 ,	0 ,	4 ,	12 ,	Yes			
E3 ,	7 ,	0 ,	3 ,	Yes			
F , Name ,	BNOD ,	CNOD ,	ENOD ,	Type			
Q1 ,	1 ,	2 ,	3 ,	NPN1			
Q2 ,	5 ,	6 ,	3 ,	NPN1			
F , Type , EBE1, IBE1, EBE2, JBE2, VEB0, ALPHI, REBO, EBC1, IBC1, EBC2, IBC2, VCB0, ALPHN, RC80							
1 ,	.597 , 1E-3 ,	.659, 1.E-3 ,	30 ,	.5 ,	1.E8 ,	.597 , 1E-3 ,	.659, 1.E-3 , 30. , .99 , 1.E8



PROGRAM BLOCK CONSTRUCTION

Figure 5. Program Block Construction

incorporated into this program are, a pattern search technique called direct search and a gradient technique. Only the direct search technique will be discussed, as it is receiving the most use.

DIRECT SEARCH

The direct search algorithm is basically a trial and error method guided by programmed logic. The decisions determining the directions in which to search for new trial solutions are based solely on the success record of previous trial solutions.

The design function is of the form

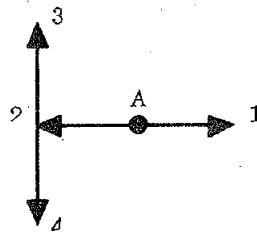
$$DF = f(X_1, X_2, X_3, \dots X_n)$$

where the X_n 's represent the set of variable circuit parameters. The purpose of the direct search program is to determine the set of circuit parameter values which result in the minimum value of the design function. There are two basic techniques utilized; "exploration moves" and "pattern moves."

Exploration consists of examining the local surface to determine if a better solution exists. After exploring the local surface, a larger move called a pattern move is made based on the results of previous trials. The implementation of these two techniques in the direct search program is best exemplified by the two dimensional problem of Figure 6. X_1 and X_2 represent the circuit parameters which will be varied to obtain the minimum design function. Point B represents the point of minimum DF while the circles represent lines of equivalent DF. Point A represents the initial or starting point.

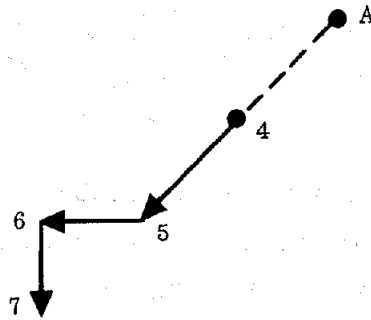
Exploration consists of examining the local surface. This maneuver is shown in Figure 6. The first step performed by the computer is evaluation of the design function at Point A (DF(A)). Parameter X_1 is then increased by a specified increment and DF(1) is evaluated. For this example, DF(1) is larger than DF(a), therefore Point 1 is rejected. Parameter X_1 is next decreased by the specified increment and DF(2) is evaluated. DF(2) is smaller than DF(a). As a result, Point A is rejected and Point 2 becomes the new "base point." X_2 is now increased by the specified increment and DF(3) evaluated. DF(3) is larger than DF(2) and Point 3 is rejected. X_2 is then decreased and DF(4) evaluated. DF(4) is smaller than DF(2) and as a result DF(4) becomes the new base point.

It should be noted at the end of the exploration phase, the program has moved from Point A to Point 4. The move is represented as the dotted vector of Figure 7.



● B

Figure 6. Two Dimensional Example Step 1



● B

Figure 7. Example Step 2

A "pattern move" will now be made from Point 4 of the same length and direction as the successful vector. This results in Point 5 of Figure 7. $DF(5)$ is evaluated and an exploratory move is now performed about Point 5.

It should be noted that in the exploratory move, X_1 and X_2 were decreased first rather than increased. This pattern was therefore tried first in the next exploration. $DF(7)$ is now compared to $DF(4)$ to determine if the pattern move and following exploration have been successful. $DF(7)$ is less than $DF(4)$ and as a result, Point 4 is rejected and Point 7 becomes the base point. A pattern move will now be made from Point 7 of the same length and direction as the successful vector from Point 4 to Point 7. This results in Point 8 of Figure 8. It should be noted that the pattern move of Figure 8 is longer than the pattern move of Figure 7. In other words, a pattern move is capable of growing and conversely of shrinking. Exploration is now performed about Point 8. $DF(11)$ is smaller than $DF(7)$ and as a result, Point 11 is chosen as the new basepoint.

Another pattern move is now executed (Point 12 of Figure 9) and the following exploration fails to reduce the design function below $DF(11)$. Exploration is then

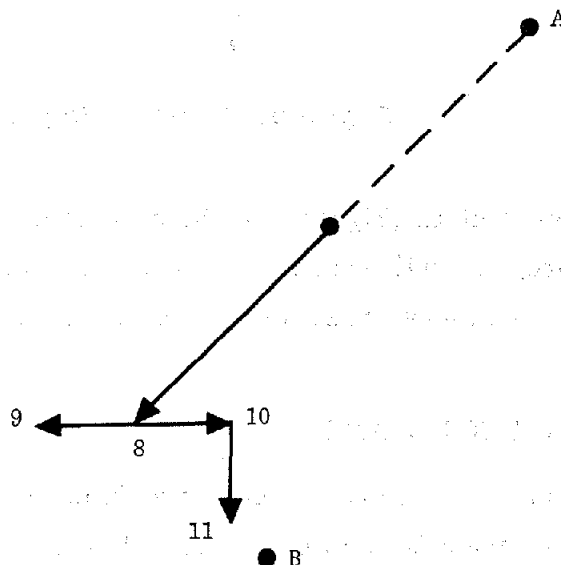


Figure 8. Example Step 3

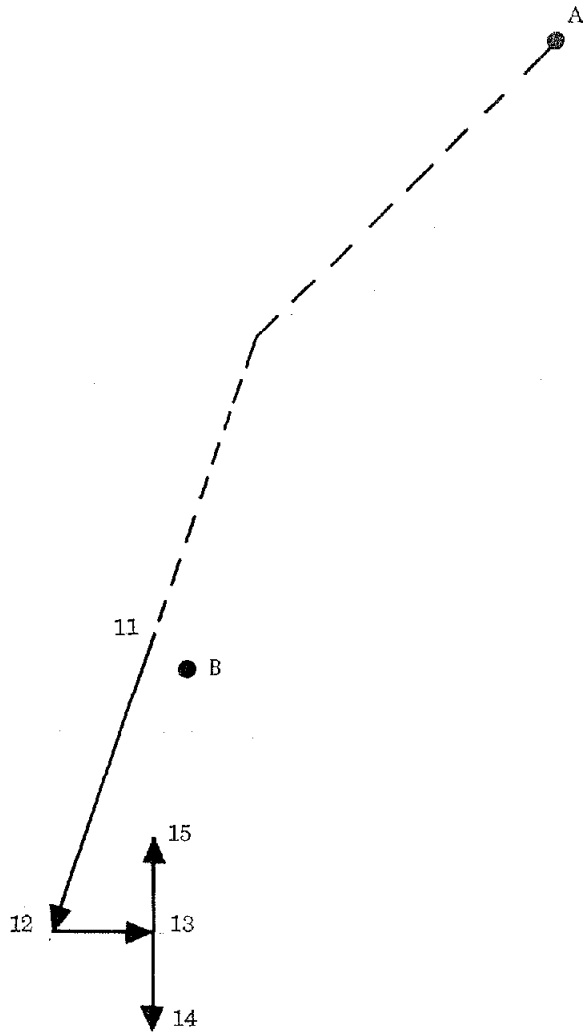


Figure 9. Example Step 4

attempted about basepoint 11 (Figure 10) which again fails to reduce the design function. At this point, the program will reduce the step size in the exploratory move and a new exploration is performed about basepoint 11 in the same manner as was performed about Point A.

ADVANTAGES OF DIRECT SEARCH

There are many sophisticated gradient techniques that will find the minimum of the example problem much faster than the direct search program. However, most design functions are not so well behaved. The direct search program often works better with functions that have discontinuous derivatives. In addition, constraints are often easier to include and in most cases, a greater variety of optimization

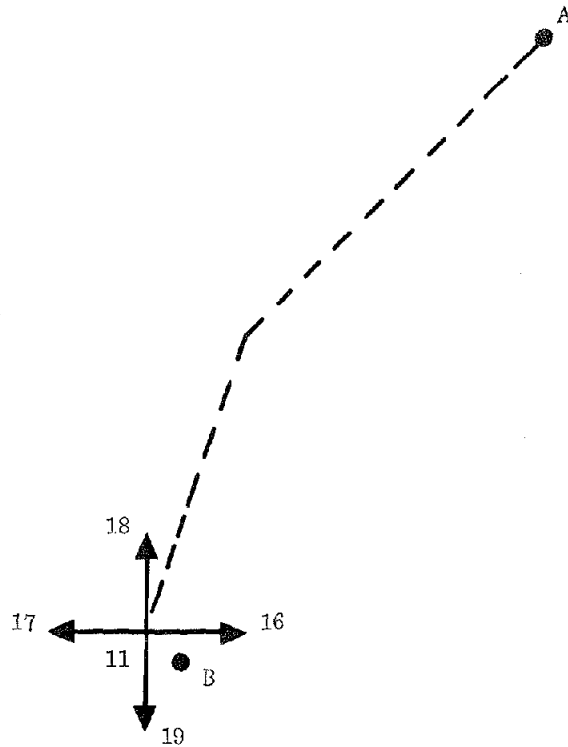


Figure 10. Example Step 5

criteria can be incorporated. It can easily be shown that the direct search program is more effective than the steepest descent gradient technique for the poorly behaved surface of Figure 11.

Point A is the initial point. The maximum step size is the same for both programs, therefore, both programs arrive at Point 1.

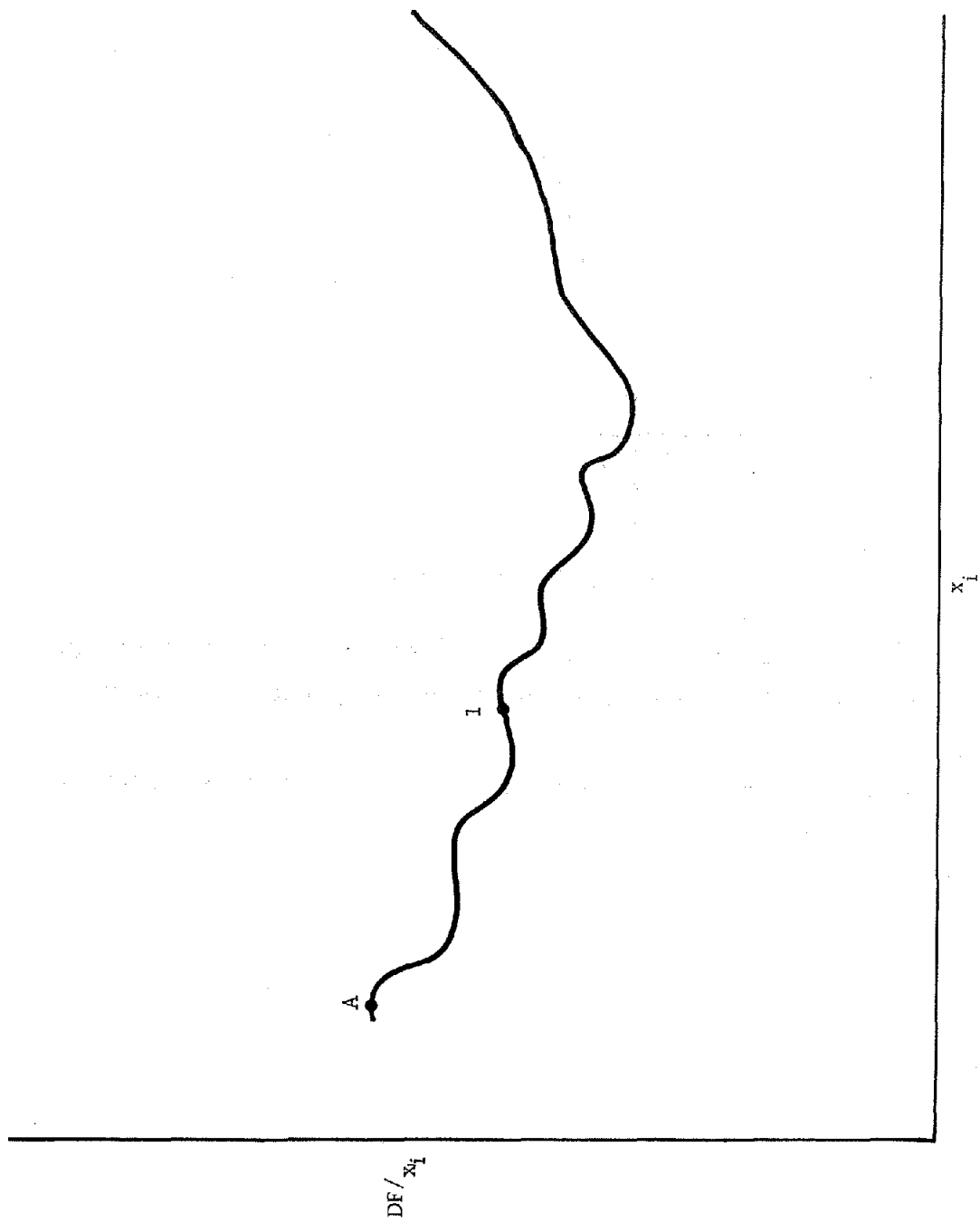


Figure 11. Poorly Behaved Surface

CONCLUSION

This paper has described a computer program which automates the "analysis-redesign" cycle for a large class of electronic circuits. At the present time, this program is incomplete and is being used to study the feasibility of the approach and gain experience in structuring a program of this complexity. Many of the conclusions reached in this study have been presented; however, there are still many interesting problems yet to be solved. One important conclusion needs to be emphasized; as the number of available functional subroutines increases, the ability to add capabilities to the program increases and the time required to add these capabilities decreases. These functional subroutines are essentially forming the basis of a circuit design and analysis language.

The ability to apply the "analysis-redesign" concept to a large class of circuits and circuit problems is an essential feature of the program. The ability to apply the concept to a large class of problems is primarily limited by the ability of the simulation and analysis routines. Simulation and analysis capabilities planned for inclusion in this program should make it applicable to most electronic circuits.

Developing and using design functions will present a challenging yet rewarding experience to the circuit designer and programmer. It will truly free the engineer from laborious calculations and place more emphasis on his creative ability. His creative ability will be needed not only to devise new and improved circuits but will also be needed to develop and apply new and more useful design functions.