NONLINEAR PROGRAMMING WITHOUT COMPUTATION

by

L. O. Chua and G. N. Lin

Memorandum No. UCB/ERL M82/21

6 April 1982

ELECTRONICS RESEARCH LABORATORY

College of Engineering
University of California, Berkeley
94720

Nonlinear Programming without Computation*

*L. O. Chua and G. N. Lin***

ABSTRACT

Using the Kuhn-Tucker conditions from mathematical programming theory, a *canonical nonlinear programming circuit* for simulating general nonlinear programming problems has been developed. This circuit is *canonical* in the sense that its topology remains unchanged and that it requires only minimum number of 2-terminal nonlinear circuit elements. Rather than solving the problem by iteration using a digital computer, we obtain the answer by setting up the associated nonlinear programming circuit and measuring the node voltages. In other words, the nonlinear programming circuit is simply a special purpose analog computer containing a repertoire of nonlinear function building blocks. To demonstrate the feasibility and advantage of this approach, several circuits have been built and measured. In all cases, the answer are obtained almost instantaneously *in real time* and are accurate to within 3% of the exact answers.

* Research sponsored in part by the Office of Naval Research Contract N00014-76-0572 and by the National Science Foundation under Grant ECS-8020640.

** L. O. Chua is with the Department of Electrical Engineering and Computer Sciences and the Electronic Research Laboratory, University of California, Berkeley,CA94720.

G. N. Lin is with the Shanghai Railroad Institute, Shanghai 201803, China. He is presently a visiting Research Fellow at the University of California, Berkeley.

1. Introduction

Nonlinear programming is a basic tool in engineering design where a set of design parameters is optimized subject to inequality constraints [1-4]. Numerous algorithms have been developed over the past 2 decades for solving nonlinear programming problems using a digital computer. One basic problem that characterizes this digital computer approach is that the computation time could be excessive even for low-dimensional problems. Moreover, for certain nonlinearities, the iteration may not even converge.

One alternative to overcome the above objection is to develop a special circuit which simulates both the objective function and the constraint functions. This approach was first proposed by Dennis [5] in 1959, and further extended by Stern [6] in 1965, for solving *linear* and *quadratic* programs. The basic building blocks they used are linear resistors, dc voltage and current sources, ideal diodes and multiport transformers. Their approach, however, had been mainly of theoretical interest because until only recently, it is difficult to build multiport transformers that work at dc. Moreover, the pn-junction diode available then did not represent a good approximation to an *ideal* diode. But perhaps the most serious objection is that this approach is not valid for *more general* nonlinear programming problems.

Our objective in this paper is to show that all of the above objections can now be overcome. First of all, advances in device technology now allows us to combine an op amp and an ordinary pn-junction diode to obtain a v-i characteristic which nearly approach that of an ideal diode [7]. Second, multiport transformers that operate at dc can now be built using solid state components [10,13].

But most important of all, we will present in this paper an entirely new circuit which is capable of simulating *any* nonlinear programming problems, not just linear or quadratic programs. The circuit is made of *ideal diodes*, and *nonlinear controlled sources* (with *multiple* controlling variables) only. In the current state-of-the-art, each nonlinear controlled source must be custom built using commercially available solid state devices. Several examples are given in this paper to demonstrate the current techniques for realizing nonlinear controlled sources.

What we have in effect is a special analog computer for simulating nonlinear programming problems. Consequently, it will clearly be less accurate and, at present, expensive to build. However, the cost should continue to decrease as better and cheaper devices become available.

There are 2 main advantages, however, of an analog computer approach over the conventional digital computer approach. First, since no iteration is involved, there is never a convergence problem so long as the nonlinear programming problem has a solution. Secondly, unlike the digital computer approach, the solution is obtained in real time.

In many applications, the variables **x** in a nonlinear programming problem must be continuously optimized as some system parameter varies. Solving this problem using a digital computer would entail storing the entire data in a buffer while the computer iterates to optimize **x** , *for each* parameter value. Clearly, this is a very time consuming task; moreover, there is no guarantee that the iteration will always converge for all parameter values.

Our nonlinear programming circuit is ideally suited for solving this problem because the solution will automatically adjusts to the optimum value as some parameter is varied. Moreover, it is done in *real time.* In certain applications (e.g., adaptive control), the variable **x** must be optimized in real time as some system parameter changes in an unpredictable way. For this class of problems, the present approach would seem to be the only solution.

2. Nonlinear Programming Circuit

Consider the following general nonlinear programming problem*:

Minimize a scalar function

$$\varphi(x_1, x_2, \ldots, x_q) \tag{1}$$

subject to the constraints

$$\begin{aligned} f_1(x_1, x_2, \ldots, x_q) &\geq 0 \\ f_2(x_1, x_2, \ldots, x_q) &\geq 0 \\ &\cdots \\ f_p(x_1, x_2, \ldots, x_q) &\geq 0 \end{aligned} \tag{2}$$

where q and p are two independent integers.

* Other nonlinear programming problems can be recast into the standard form (1) and (2).

The well-known *Lagrange multiplier approach* for solving this problem consists of defining a Lagrange function

$$L(\mathbf{x},\lambda) = \varphi(\mathbf{x}) + \sum_{j=1}^{p} \lambda_j f_j(\mathbf{x}) \tag{3}$$

where the real constants $\lambda_1, \lambda_2, \ldots, \lambda_p$ are called Lagrange multipliers [3,4]. If the program has a solution $\mathbf{x}^*$, i.e.,

$$\min \varphi(\mathbf{x}) = \varphi(\mathbf{x}^*)$$

and

$$f_j(\mathbf{x}^*) \geq 0, \quad j = 1,2,\ldots,p$$

then the following conditions must hold:

$$\frac{\partial \varphi}{\partial x_l}(\mathbf{x}^*) + \sum_{j=1}^{p} \lambda_j^* \frac{\partial f_j}{\partial x_l}(\mathbf{x}^*) = 0, \quad l = 1,2,\ldots,q \tag{4}$$

$$f_j(\mathbf{x}^*) \geq 0, \quad j = 1,2,\ldots,p \tag{5}$$

$$\lambda_j^* \leq 0, \quad j = 1,2,\ldots,p \tag{6}$$

$$(\lambda_j^*)\, f_j(\mathbf{x}^*) = 0, \quad j = 1,2,\ldots,p \tag{7}$$

Equations (4) to (7) are called *Kuhn-Tucker conditions*. The proof of the necessity of these conditions can be found in [4].

Our main objective here is to show that the nonlinear circuit shown in Fig.1 is described precisely by Eqs.(4)-(7), provided we identify each *node voltage* v_l with the *variable* x_l and *each reversed diode current* i_j *with the Lagrange multiplier* λ_j.

The diodes in Fig.1 denote *ideal* diodes with $v_d - i_d$ characteristic shown in Fig.2. Note that

$$i_{d_j} = 0\,, \qquad v_{d_j} \leq 0 \tag{8}$$

$$i_{d_j} \geq 0\,, \qquad v_{d_j} = 0 \tag{9}$$

and

$$v_{d_j} i_{d_j} = 0 \tag{10}$$

Each diamond shape symbol enclosing a plus-minus sign denotes a non-linear *controlled voltage source.* whose terminal voltage depends on the node voltage $v_1, v_2, \ldots, v_q$ in accordance with the prescribed nonlinear function $f_j(v_1, v_2, \ldots, v_q)$.

Each diamond shape symbol (in the middle column) enclosing an arrow-head is a *controlled current source* whose terminal current depends on both the reversed diode currents $i_{d_1}, i_{d_2}, \ldots, i_{d_p}$, and the node voltages $v_1, v_2, \ldots, v_q$ in accordance with the prescribed function

$$\sum_{j=1}^{p} i_j \frac{\partial f_j(v_1, v_2, \ldots, v_q)}{\partial v_l}$$

Each diamond shape symbol (on the right) enclosing an arrowhead denotes a *controlled current source* whose terminal current depends on the node voltages $v_1, v_2, \ldots, v_q$ in accordance with the prescribed nonlinear function

$$\frac{\partial \varphi(v_1, v_2, \ldots, v_q)}{\partial v_l}$$

To prove that Fig.1 indeed simulates the Kuhn-tucker conditions, note that Eq.(4) is simply obtained by applying KCL at each node l, l=1,2,...,q, in Fig.1. Next, observe that

$$v_{d_j} = -f_j(v_1, v_2, \ldots, v_q) \leq 0$$

in view of (8)-(9). Hence $f_j(v_1, v_2, \ldots, v_q) \geq 0$. Finally, note that since $i_j = -i_{d_j}$, it follows from Eqs.(9) and (10) that $i_j \leq 0$ and $i_j f_j(v_1, v_2, \ldots, v_q) = 0$, $j = 1,2,\ldots p$.

Therefore, whenever there are unique solutions $\mathbf{x}^*$ and λ^* for the original nonlinear programming problem, the node voltages $\mathbf{v}$ and the reversed currents $\mathbf{i}$ through the diodes at the operating point of this circuit will automatically give the solutions $\mathbf{x}^*$ and λ^*.

3. Circuit Implementation

Consider now the task of implementing the nonlinear programming circuit in Fig.1 using solid state devices.

The ideal diodes can be implemented by the pn-junction diode op amp feedback circuit shown in Fig.3. The diode d_1 in Fig.3(a) represents a pn-junction diode. To distinguish it from an ideal diode, we use a symbol slightly different from that of an ideal diode. The driving-point characteristic of this circuit was shown in [7] to be an excellent approximation to that of an ideal diode.

The nonlinear controlled sources can be realized by combinations of op amp and nonlinear multipliers [8] using recent techniques developed in [9-12]. At present, each nonlinear controlled source for a given problem must be custom built, and suffers from accuracy, sensitivity and stability problems. However, as more accurate and versatile components become available, it would be possible to design special analog computers with a large enough repertoire of building blocks so that the nonlinear controlled sources can be routinely assembled at ease.

In this section, we will consider a few examples of nonlinear programming problems and show how Fig.1 can be actually implemented using standard off-the-shelves components.

A. Linear and Quadratic Programs

A programming problem is called a *quadratic program* if it has the following form:

$$\textit{minimize} \quad \varphi(\mathbf{v}) = \mathbf{A}^T\mathbf{v} + \frac{1}{2}\mathbf{v}^T\mathbf{G}\mathbf{v} \tag{11}$$

$$\textit{subject to} \quad \mathbf{f}(\mathbf{v}) = \mathbf{B}\mathbf{v} - \mathbf{e} \geq 0 \tag{12}$$

where $\mathbf{A}$ and $\mathbf{v}$ are q-vectors, $\mathbf{f}$ and $\mathbf{e}$ are p-vectors, $\mathbf{B}$ is a ($p \times q$) matrix and $\mathbf{G}$ is a ($q \times q$) symmetric, positive semidefine matrix. In the special case where $\mathbf{G} = 0$, the resulting problem is called a *linear program*.

A general circuit which simulates (11)-(12) is shown in Fig.4. Note that the right part of Fig.1 is realized using only dc current sources and linear resistors. The values of the resistor R_{ij} connecting node i and node j is

$$R_{ij} = -\frac{1}{G_{ij}}, \quad i \neq j \tag{13}$$

The value of the resistor R_{ii} connecting node i to ground (the reference node) is

$$R_{ii} = \frac{1}{\sum_{j=1}^{q} G_{ij}} \tag{14}$$

The dc current source in Fig.4 can be realized by the transistor- op amp circuit shown in Fig.5 [7].

Note also that the middle part of Fig.1 reduces to a (p+q)-port transformer, which can be realized using the technique developed in [13], in addition to a set of dc voltage sources. If, as is often the case in practice, some constraints in Eq.(12) assume the special form

$$f_j(v_1,v_2,\ldots,v_q) = v_l - e \geq 0 \tag{15}$$

then since

$$\frac{\partial f_j(v_1,v_2,\ldots,v_q)}{\partial v_i} = \begin{cases} 1 & i=l \\ 0 & i \neq l \end{cases}$$

the circuit in Fig.1 corresponding to this constraint reduces to that shown in Fig.6. This circuit can be further reduced to that of Fig.7 since they both impose the same constraint on the ideal diode. The ideal diode-battery combination in Fig.7 and in the right part of Fig.4 can be realized by the two op-amp circuit shown in Fig.7, where the op amp A_2 is used to provide an adjustable bias voltage e. Let us illustrate this with an example.

Example 1.

$$\begin{aligned} &\textit{minimize} \quad \varphi(v_1,v_2,v_3) = 0.4v_1 + \frac{1}{2}(5v_1^2 + 8v_2^2 + 4v_3^2 - 3v_1v_2 - 3v_2v_3) \\ &\textit{subject to} \quad v_1 + v_2 + v_3 \geq 1 \\ &\qquad\qquad\quad v_1, v_2, v_3 \geq 0 \end{aligned}$$

The simulating circuit corresponding to Fig.3 is shown in Fig.9. In addition, we have to give adequate units to the physical circuit. To ensure that all signals (voltages and currents) are within the dynamic range of the op amp, we choose the following normalizing values:

$$V_0 = 1\ volt, \quad I_0 = 1\ ma, \quad R_0 = 1\ k\Omega \tag{16}$$

All the values of voltages, currents and resistors of Fig.9 have to be normalized with respect to these normalizing values. Using the op amp circuit in [13] to implement the (1+3)-port transformer, and using Fig.8 to implement Fig.7, we obtain the complete *quadratic programming circuit* in Fig.10. This circuit was built and its node voltage are measured:

$$v_1 = 0.258, \quad v_2 = 0.329, \quad v_3 = 0.407$$

The theoretical solutions for this problem are:

$$v_1 = 0.2520, \quad v_2 = 0.3328, \quad v_3 = 0.4150$$

The biggest relative error is

$$\frac{0.258 - 0.252}{0.252} = 2.38\%$$

B. General Nonlinear Program

Since there is an almost infinite variety of nonlinear controlled sources depending on several controlling variables, it would be impractical to catalog simplified versions of Fig.1, as we have done in Fig.3, for quadratic programs. Since each nonlinear programming circuit must presently be custom designed and built, let us consider a typical example.

Example 2.

$$\textit{minimize} \quad \varphi(v_1,v_2) = 0.4v_2 + v_1^2 + v_2^2 - v_1v_2 + \frac{1}{30}v_1^3 \tag{17}$$

$$\textit{subject to} \quad v_1 + 0.5v_2 \geq 0.4 \tag{18}$$

$$0.5v_1 + v_2 \geq 0.5 \tag{19}$$

$$v_1, v_2 \geq 0 \tag{20}$$

The simulating circuit corresponding to Fig.1 is shown in Fig.11. The nonlinear controlled current source $i=0.1v^2$ is implemented by the circuit shown in Fig.12, where the analog multiplier (see symbol in Fig.13) is defined by

$$v_{out} = \frac{(x_1-x_2)(y_1-y_2)}{10}$$

The actual characteristic shown in Fig.14 gives an excellent approximation. Taking the same normalizing values as (16), the complete nonlinear programming circuit for implementing Fig.11 is shown in Fig.15. This circuit was built and the measured node voltages are:

$$v_1 = 0.336, \quad v_2 = 0.320$$

The theoretical solutions are:

$$v_1 = 0.3395, \quad v_2 = 0.3302$$

The biggest relative error is:

$$\frac{0.330 - 0.320}{0.330} = 3.03\%$$

Example 3.

In the laboratory it is very easy to change some parameters in the simulating circuit. This corresponds to changing some coefficients in the original nonlinear programming problem. For instance, in the above example if we adjust the dc voltage V_a in Fig.15 from 0.4 to some other values, the constraint (18) will change accordingly. This is especially useful when we want to know how the minimum point of φ varies with the coefficient. For the circuit in Fig.15, we used an adjustable dc voltage source as V_a and measured the values of v_1 and v_2 corresponding to different values of V_a. All measurements are in good agreement with the theoretical solutions. Next, we used a 100Hz sinusoidal voltage source in place of the dc voltage V_a and traced the v_1–v_2 Lissajous figure (Fig.16). This is the trace of the point min $\varphi(v_1,v_2)$ when the constant of right hand part of constraint (18) changes continuously. Using SPICE[14], the simulating circuit in Fig.11 is also solved for different values of V_a. Fig.17 is the computer output of v_1 and v_2 when V_a increases from 0 to 1 with an increment of 0.01. Fig.17 is drawn according to the data in Fig.18 (only a part of the data is listed in Fig.18). A comparison between Fig.16 and Fig.17 shows that the laboratory result is reasonably close to the theoretical result simulated using SPICE. Yet the result in Fig.16 is obtained *instantly* whereas that in Fig.17 requires considerable computer time because the problem must be solved repeatedly many times, one for each parameter value.

4. References

[1] W. I. Zangwill, *Nonlinear Programming*, Prentice-Hall Inc., Englewood Cliffs, N. J., 1969.

[2] M. Avriel, *Nonlinear Programming* Prentice-Hall Inc., Englewood Cliffs, N. J., 1976.

[3] D. A. Wismer and R. Chattergy, *Introduction to Nonlinear Optimization*, Elsevier, 1978.

[4] R. J. Duffin, E. L. Peterson and C. Zener, *Geometric Programming-Theory and Application*, John Wiley & Sons, 1967.

[5] J. B. Dennis, *Mathematical Programming and Electrical Networks*, Chapman & Hall, London, 1959.

[6] T. E. Stern, *Theory of Nonlinear Networks and Systems*, Addison-wesley, 1965.

[7] L. O. Chua and S. Wong, " Synthesis of Piecewise Linear Networks ", *Electronic Circuits and Systems*, July 1978, Vol.2, No 4, pp. 102-108

[8] D. H. Sheingold, *Nonlinear Circuit Handbook*, Analog Devices, Inc., Norwood, Mass, 1974.

[9] N. R. Malik, G. L. Jackson and Y. S. Kim, " Theory and application of resistor, linear controlled resistor, linear controlled conductor network", *IEEE Trans. on Circuits and Systems*, Vol. CAS-23, no.4, pp.222-228, April 1976.

[10] J. L. Huertas and A. Gago, " On the realization of arbitrary linear resistive n-port network", *IEEJ. Electronic Circuits and Systems*, Vol.3, no.6, pp.247-252, Nov. 1979.

[11] J. L. Huertas, J. I. Acha and A. Gago, "Design of general voltage or current controlled resistive elements and their applications to the synthesis of non-linear networks", *IEEE Trans. on Circuits and Systems*, Vol. CAS-27, no.2, pp. 92-103, Feb. 1980.

[12] J. L. Huertas and A. Rueda, " Synthesis of resistive n-port section-wise piecewise-linear networks ", *IEEE Trans. on Circuits and Systems*, Vol. CAS-29, no.1, pp. 6-14, Jan. 1982.

[13] L. O. Chua, G. N. Lin and J. J. Lum, "The (p+q)-port transformer", Electronics Research Laboratory, University of california, Berkeley,Calif., Memorandum No. UCB/ERL M81/41, 11 June 1981.

[14] A. Vladimirescu, K. Zhang, A. R. Newton, D. O. Pederson and A. Sangiovanni-Vincentelli, "SPICE Version 2G User's Guide", Electronics Research Laboratory, University of California, Berkeley, Aug.10, 1981.

Figure captions

Fig. 1 The canonical circuits.

Fig.2 An ideal diode.

Fig.3 A realization of an almost ideal diode.

Fig.4 A general simulating circuit for a quadratic program.

Fig.5 A realization of a dc current source.

Fig.6 The simulating circuit corresponding to the constraint (12).

Fig.7 The simplified circuit of Fig.6.

Fig.8 A realization of a biased almost ideal diode.

Fig.9 The canonical circuit for Example 1.

Fig.10 The laboratory circuit for Example 1. All unmarked resistors are $10k\Omega$.

Fig.11 The canonical circuit for Example 2.

Fig.12 Realization of a nonlinear voltage controlled current source.

Fig.13 The symbol of an analog multiplier.

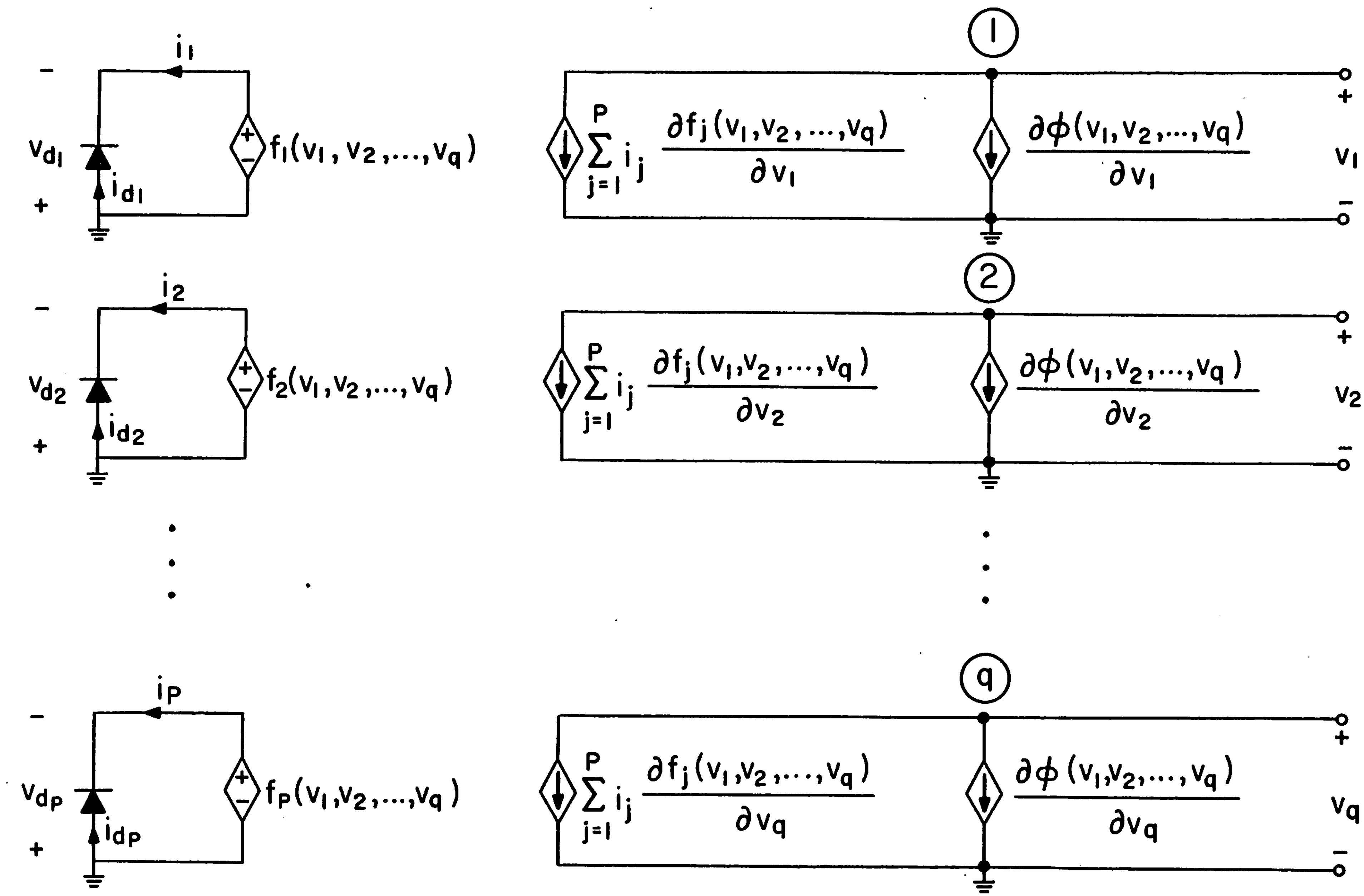
Fig.14 The v-i relation of the circuit in Fig.12. Horizontal scale: 0.1v/div; Vertical scale: 0.1ma/div.

Fig.15 The laboratory circuit for Example 2. All unmarked resistors are $10k\Omega$.

Fig.16 An oscilloscope tracing for Example 3. Horizontal and vertical scale: 0.1v/div.

Fig.17 The trace drawn according to the data in Fig.18.

Fig.18 A computer output of the canonical circuit in Fig.11.



i_1
v_{d1}
i_{d1}
$f_1(v_1, v_2, \ldots, v_q)$
$\sum_{j=1}^{P} i_j \frac{\partial f_j(v_1,v_2,\ldots,v_q)}{\partial v_1}$
$\frac{\partial \phi(v_1,v_2,\ldots,v_q)}{\partial v_1}$
v_1
i_2
v_{d2}
i_{d2}
$f_2(v_1, v_2, \ldots, v_q)$
$\sum_{j=1}^{P} i_j \frac{\partial f_j(v_1,v_2,\ldots,v_q)}{\partial v_2}$
$\frac{\partial \phi(v_1,v_2,\ldots,v_q)}{\partial v_2}$
v_2
i_P
v_{dP}
i_{dP}
$f_P(v_1, v_2, \ldots, v_q)$
$\sum_{j=1}^{P} i_j \frac{\partial f_j(v_1,v_2,\ldots,v_q)}{\partial v_q}$
$\frac{\partial \phi(v_1,v_2,\ldots,v_q)}{\partial v_q}$
v_q

Fig. 1

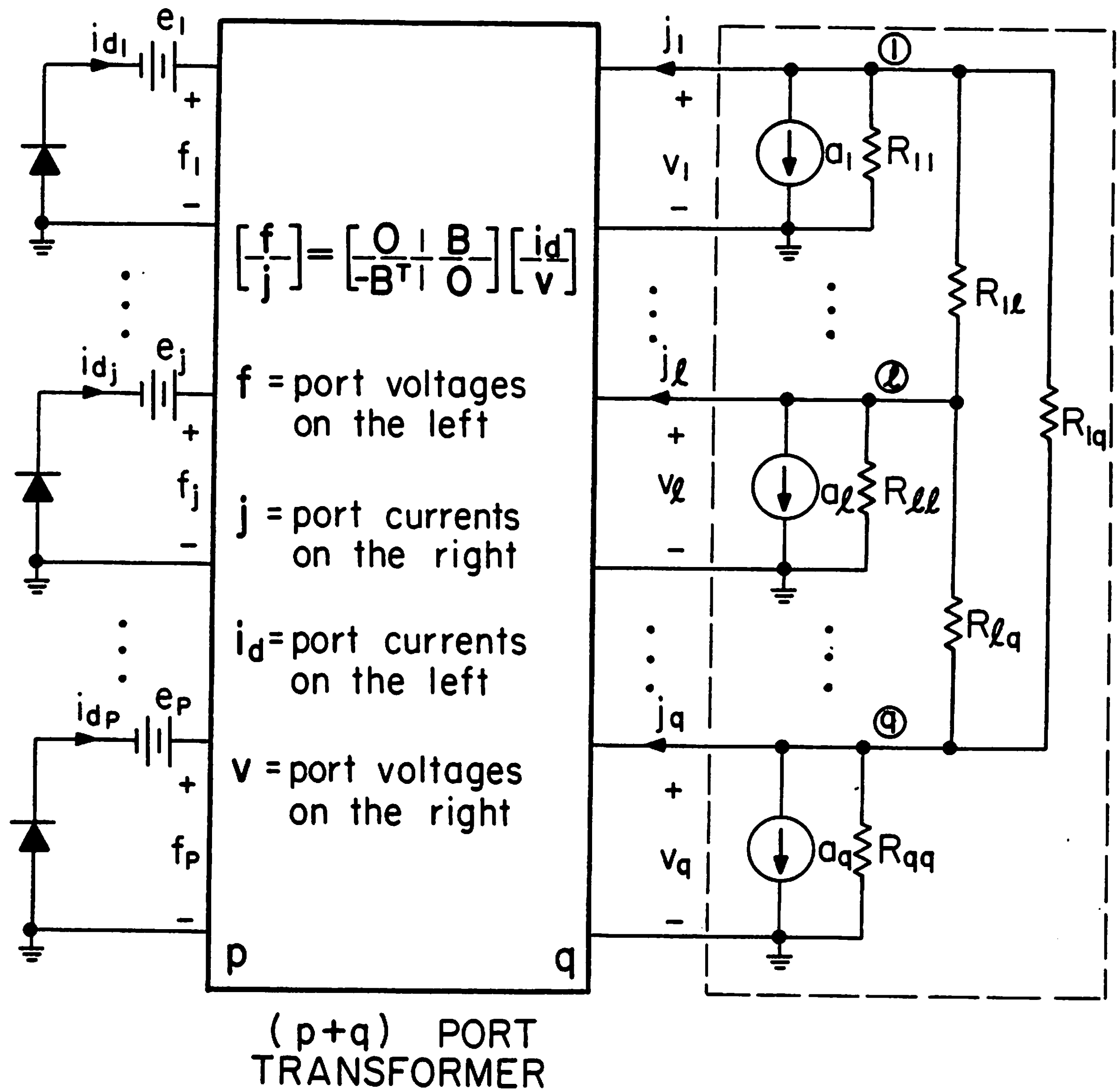


Fig. 4

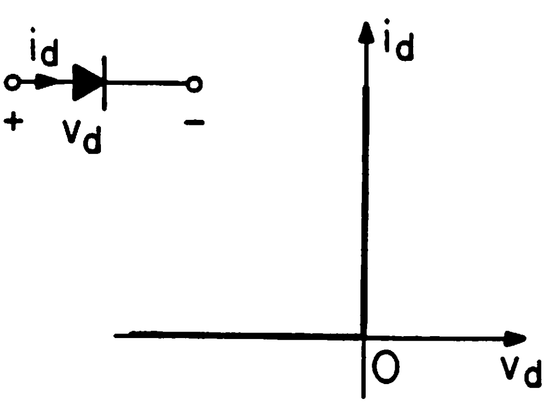


Fig. 2

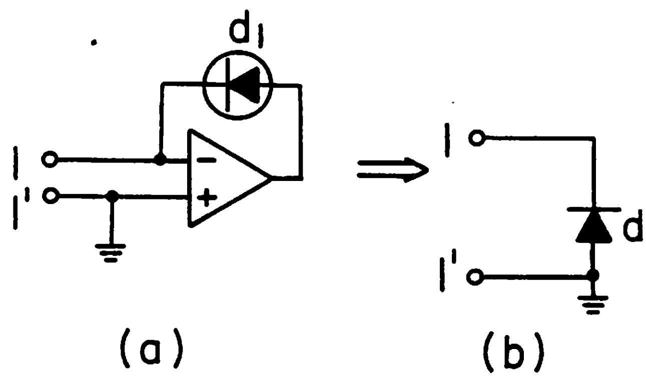
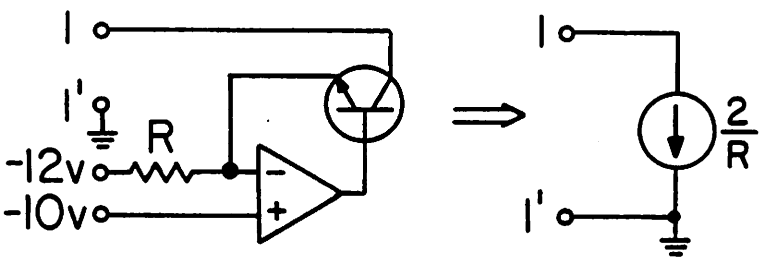


Fig. 3



(a) (b)

Fig. 5

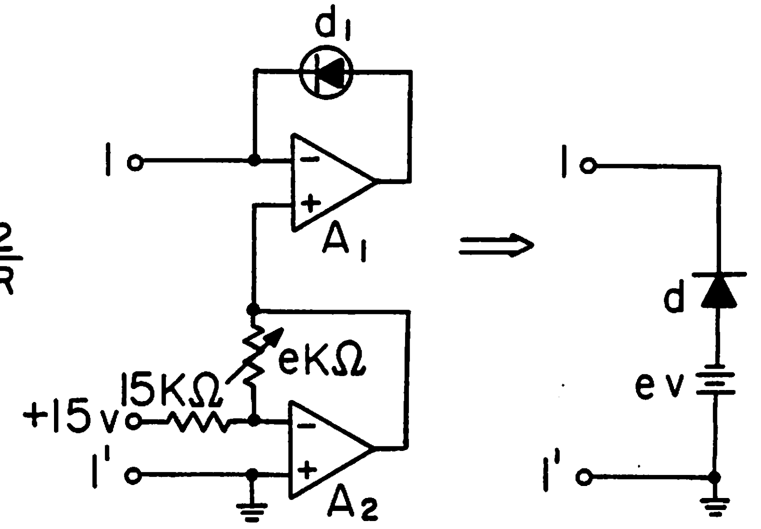


Fig. 8

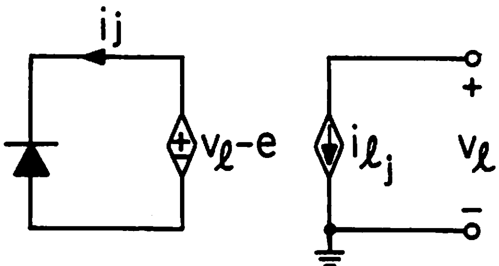


Fig. 6

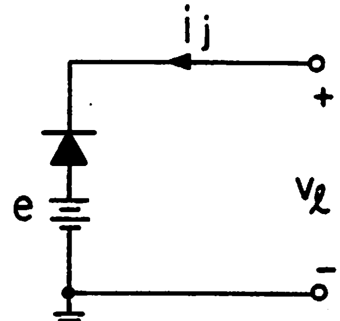


Fig. 7

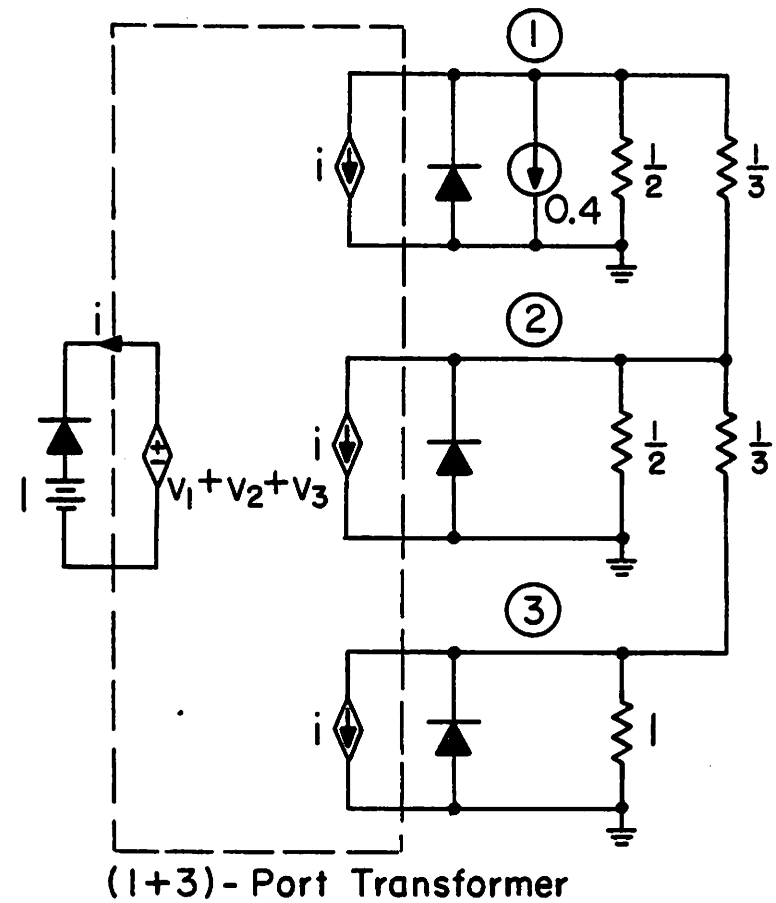


Fig. 9

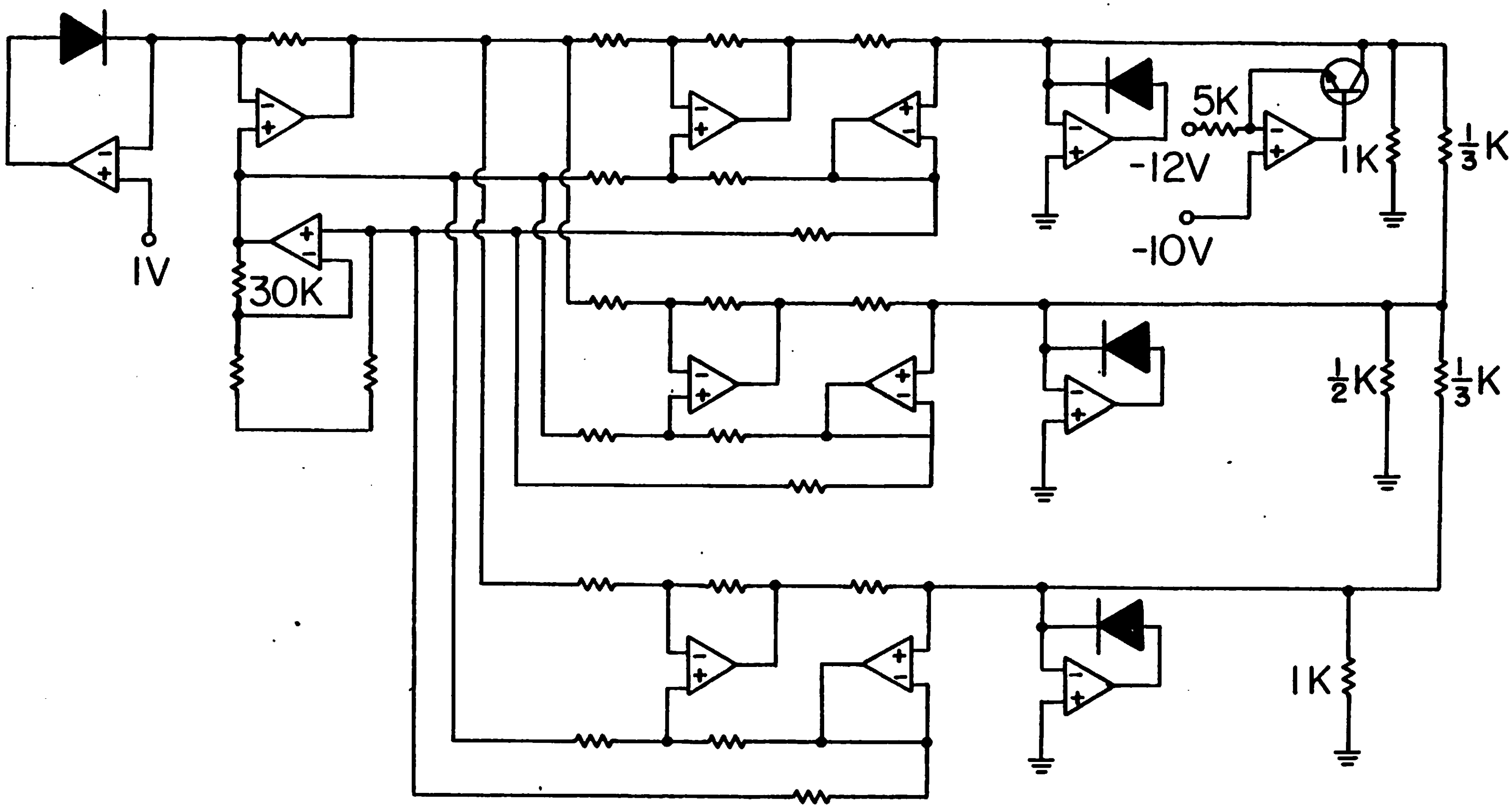


Fig. 10

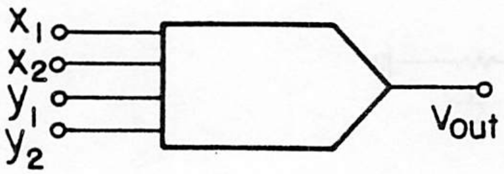


Fig. 13

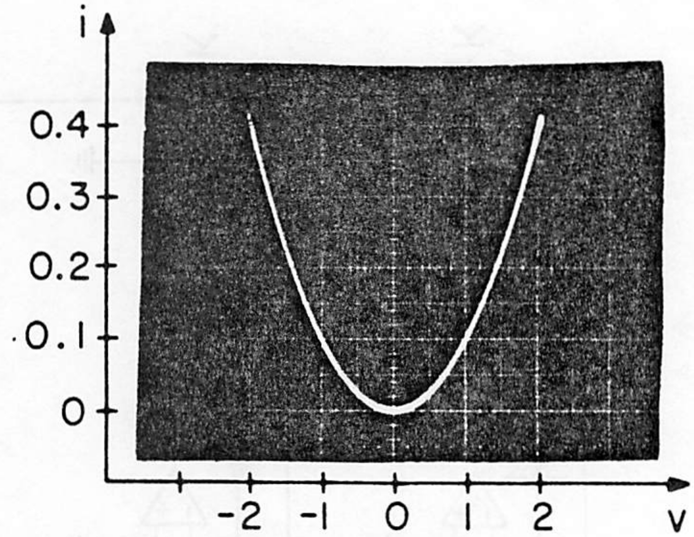


Fig. 14

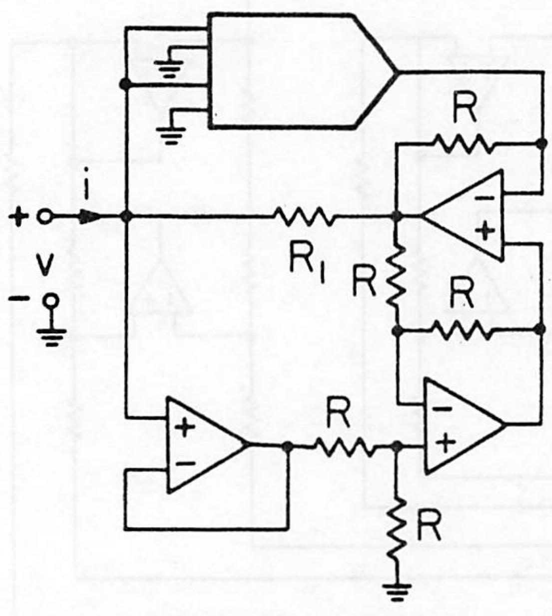


Fig. 12

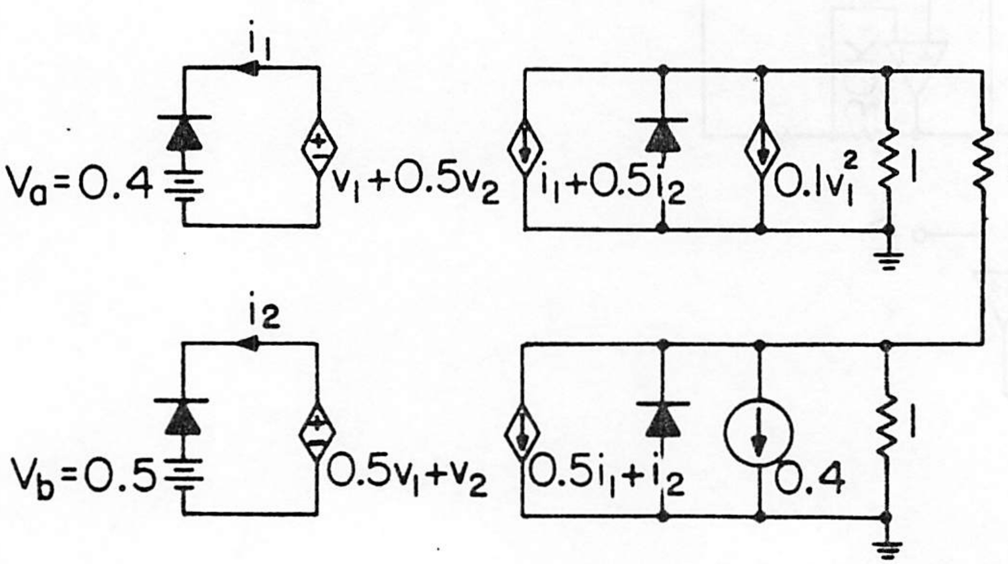


Fig. 11

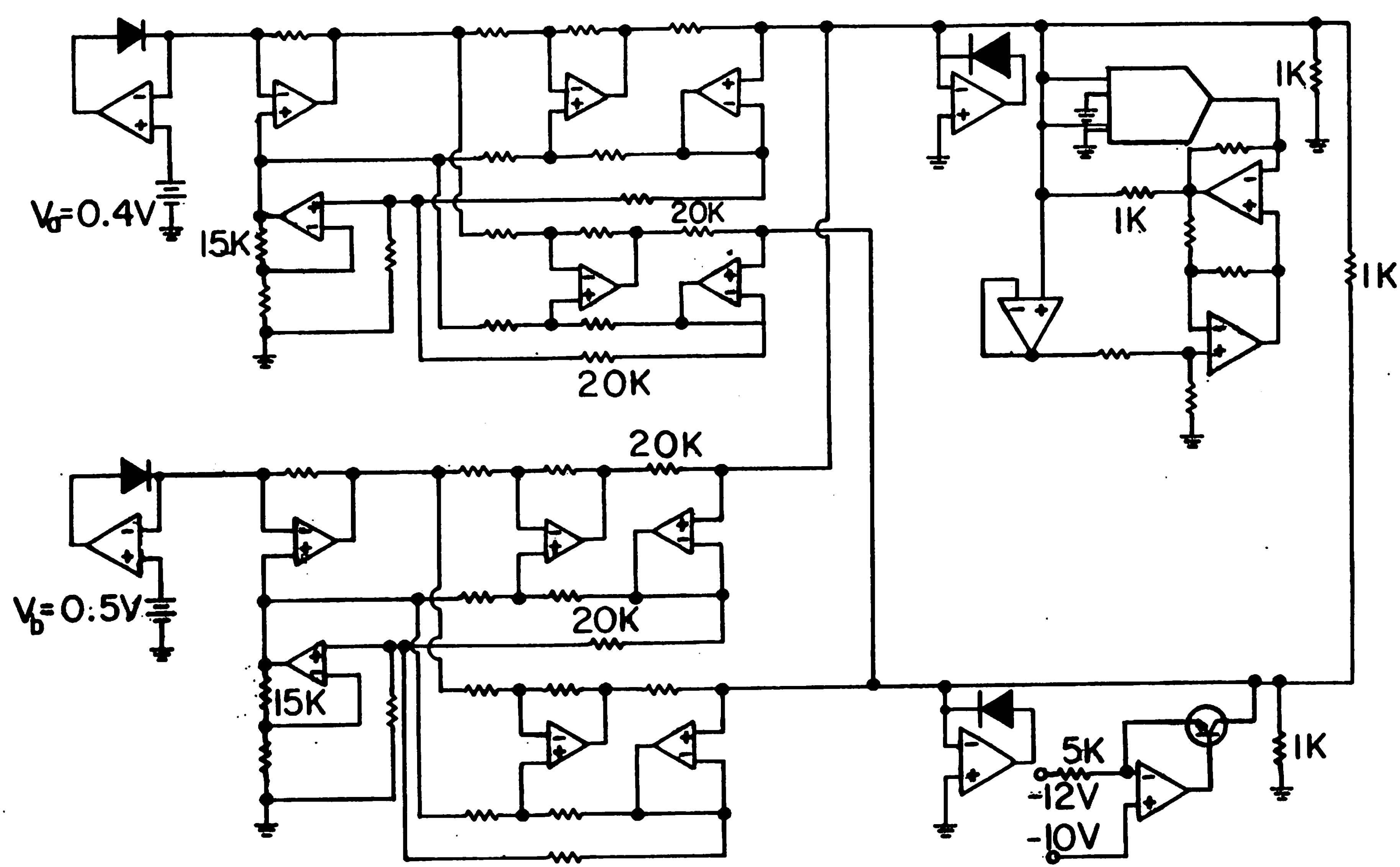


Fig. 15

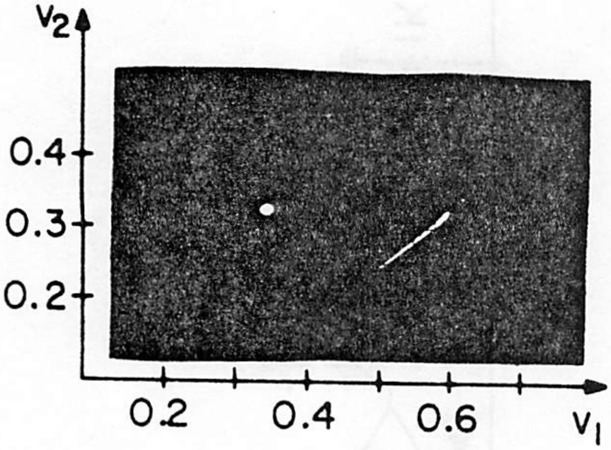


Fig. 16

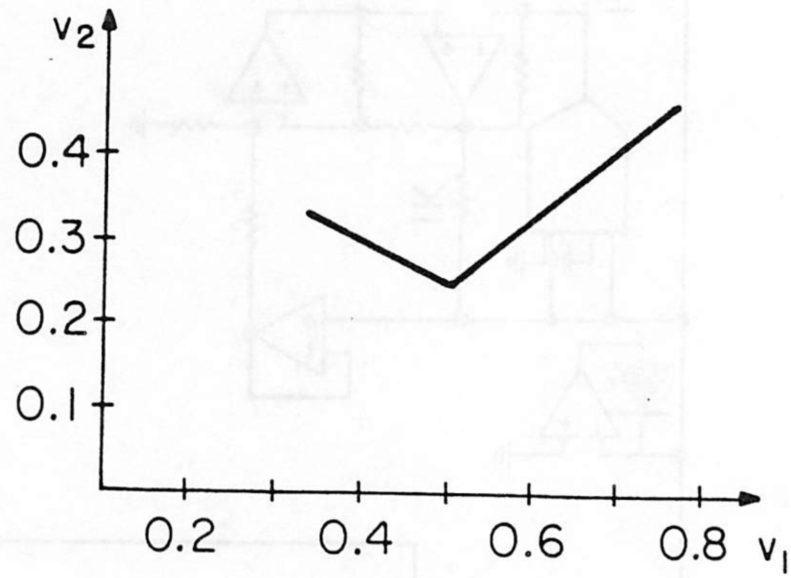


Fig. 17

Va	V(1)	V(2)
0.00d+00	3.39516d-01	3.30159d-01
5.00d-02	3.39516d-01	3.30159d-01
1.00d-01	3.39516d-01	3.30159d-01
1.50d-01	3.39516d-01	3.30159d-01
2.00d-01	3.39516d-01	3.30159d-01
2.50d-01	3.39516d-01	3.30159d-01
3.00d-01	3.39516d-01	3.30159d-01
3.50d-01	3.39516d-01	3.30159d-01
4.00d-01	3.39516d-01	3.30159d-01
4.50d-01	3.39516d-01	3.30159d-01
5.00d-01	3.39516d-01	3.30159d-01
5.50d-01	3.99947d-01	2.99945d-01
6.00d-01	4.66609d-01	2.66616d-01
6.50d-01	5.19442d-01	2.60950d-01
7.00d-01	5.54884d-01	2.90065d-01
7.50d-01	5.90308d-01	3.19216d-01
8.00d-01	6.25715d-01	3.48402d-01
8.50d-01	6.61104d-01	3.77624d-01
9.00d-01	6.96475d-01	4.06882d-01
9.50d-01	7.31829d-01	4.36174d-01
1.00d+00	7.67164d-01	4.65502d-01

Fig. 18