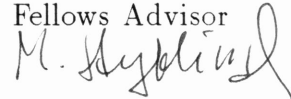


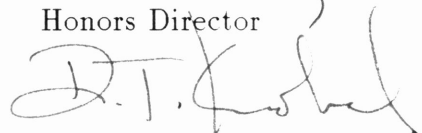
**APPLICATION OF NEURAL NETWORKS TO  
COMPUTER-AIDED DESIGN OF  
ELECTRONIC CIRCUITS:  
Solution of Linear Equations**

Jill R. Minick  
University Undergraduate Fellow, 1989-90  
Texas A & M University  
Department of Electrical Engineering

APPROVED  
Fellows Advisor



Honors Director



## I. Introduction

The objective of this research is to investigate the application of Artificial Neural Network techniques to the area of computer-aided design of electronic circuits. A major problem in this area is the analysis (especially statistical) of *large* integrated circuits, taking a substantial amount of the CPU time even on the most powerful computers available today. Two most time consuming tasks responsible for this situation are: calculation of model parameters of semiconductor devices, and solving systems of linear equations (required for all types of analyses available from such circuit analysis programs as, e.g., SPICE). Therefore, it is very important to look for methods of improving the efficiency of performing these two major tasks.

In the development of this project, the main emphasis has been directed toward solving a system of linear equations. ANNs are applicable because optimization can be used to obtain a solution to these equations. Currently, solving a system of linear equations usually involves iterative numerical techniques in the form of an algorithm implemented in a computer program. The time needed for the computer to yield an exact solution can grow exponentially with the number of equations for which the solution is desired. With ANNs, a good approximation to the solution can be obtained directly in hardware (an electrical circuit) in a parallel architecture. With a parallel architecture, operations are performed simultaneously in parallel rather than serially. As a result, expanding the number of equations to solve in parallel does not increase the time required to reach a solution. Thus the efficiency in solving a large system of linear equations is much greater with the parallel ANN.

Applying Artificial Neural Networks to the solution of a system of linear equations provides a new hardware approach which is faster and more efficient than conventional iterative implementations in software programs. One obvious disadvantage is that the accuracy of the solution obtained is

limited by the accuracy of the ANN element values and the measurements of the solution voltages, since the ANN is an *analog* circuit.

## **II. Introduction to Artificial Neural Networks in general**

Artificial Neural Networks (ANNs) have recently become a key phrase in research circles across many disciplines because of their promising outlook and far-reaching potential. The basic concept of an ANN has been around for many years, generating an interest from the late 1940's through the 1960's. However, the implementation of these concepts was not practical because researchers were limited by the analog computers of the pre-VLSI era. The rejuvenated interest in ANN concepts is a result of the tremendous advances in VLSI technology which, along with other technological advances, have made implementation possible as well as feasible [1].

Artificial Neural Networks are modelled after biological neural networks. In the human brain for example, neurons are nerve cells arranged in a network such that all thinking and learning processes are developed. Exactly how this is accomplished has always intrigued the human race and has led to many investigations over the course of time. ANNs apply what has been learned from these investigations to implementations in electronic circuits in an attempt to imitate the processing capacity of the brain. As a result, ANNs have evolved into highly interconnected, analog circuits consisting of massively parallel processing elements (neurons).

The goal of ANNs is to perform functions that humans can do very well, but that modern digital computers find very difficult. These tasks include association, classification, feature extraction, generalization, categorization, and optimization [1]. Examples might include recognizing a familiar face and recalling information associated with a certain smell or taste. Note that these tasks will involve solutions that can't be computed to an exact result, but rather involve a closest match or

best approximation.

The capabilities of the brain can be grouped into three general categories: searching, representation, and learning. These tasks are achieved with the associative and self-organizing properties of the brain. The associative property involves the ability to recall all relative information given a single detail of information. The self-organizing property refers to the ability to acquire information by a trial-and-error process which involves responding to external stimuli through organization and reorganization [1]. These are the properties that need to be implemented into ANNs.

The methodology behind ANNs involves simple processing elements (neurons) which act on analog information received from other processing elements arranged in parallel in the network. Its output, in turn, is distributed to other processing elements. The connections between elements are weighted corresponding to the relationship between the elements, which is specific to the function of the network. Because the neurons are arranged in parallel, many processes can be done simultaneously. The resulting speed and efficiency is the appeal of parallel processing. The ANN has no memory cells as a standard random access memory. The memory of the ANN resides in the state of the system where the “strengths” of the connections in the network are the “memory elements”. As an example, the neuron element can be thought of electronically as an operational amplifier in an integrator configuration. The ANN can be thought of as a large group of operational amplifiers interconnected through potentiometers which act as weights (“strengths”) in the connections. Other active elements can also be used, e.g., Schmitt Trigger devices.

Several algorithms have been developed and have proved successful in applying ANN techniques. The ANNs that have been built are application-specific, each with a separate architecture to solve a certain problem [1]. When ANNs are expanded to solve a wide range of general problems, they

tend to lose their efficiency. However, the motivation for using ANNs is still strong because of their ability to solve problems that can't be solved effectively with modern digital computers. With the advancement of technology, ANN theories and algorithms are moving closer to practical implementation. Recent developments in charge-coupled devices (CCDs) allow for dynamically reprogrammable connections which are essential in many ANN algorithms [1]. Because of the high interconnectivity of ANNs, VLSI technology may be too limited as an avenue toward implementation. Electro-optical advances may provide an alternate, more effective route.

### **III. Approach**

Several possible approaches to the solution of a system of linear equations were investigated. One such approach was to implement any iterative numerical technique and convert from the difference equations in the  $z$  domain to differential equations in the  $s$  domain. Then the differential equations could be implemented into an electrical circuit. It was discovered, however, that although going from differential equations to difference equations is relatively easy, transforming difference equations to differential equations seems to require a thorough examination into numerical methods. It was because of this difficulty that this method was abandoned.

A second approach involved a method of Least Squares which uses linear algebra to solve systems of equations. However, multiplication of matrices requires a number of multiplication operations, many of which need to be done serially. As a result, this method might be just as time consuming as the conventional software methods.

The approach used in this research is to map the Solution of a System of Linear Equations problem directly into the Linear Programming Problem of Tank and Hopfield. Tank and Hopfield showed in the Linear Programming Problem that given a function to be minimized and the constraints on

that function, a solution could be determined with a ANN implementation [2]. Tank and Hopfield's approach involved optimizing a function similar to a (quadratic) *energy* function by minimizing the square of a norm of the solution error to zero, using a modification of standard unconstrained optimization methods. The parameters of the function to be minimized, the *cost* function, and the constraint equations are mapped into a corresponding neural network. The ANN will then converge to the minimum of the corresponding energy surface within a couple of network time constants (i.e., very quickly).

This optimization involves a function to be minimized (cost function) and the constraints on that function which define the vector space where the function is to be minimized. These constraints establish the boundaries wherein the minimum is to be obtained. This minimum will correspond to the solution of the system according to standard numerical methods as proved by Tank and Hopfield.

#### **IV. Tank and Hopfield Network Solution to the Linear Programming Problem**

Tank and Hopfield proved that an ANN can be implemented into an electronic circuit where optimization is achieved by supplying initial voltages and allowing the analog circuit to converge to the point of stability. The achievement of stability implies minimization of an *energy* function which can be shown to correspond to the solution of the optimization problem. Figure 4.1 shows the organization of a network which will solve a 2-variable 4-constraint linear programming problem [2]. For the Linear Programming Problem the function to be minimized, the cost function, is

$$\Phi(\vec{\mathbf{V}}) = \vec{\mathbf{A}}^T \cdot \vec{\mathbf{V}}$$

where  $\vec{\mathbf{A}}^T$  is the transpose of the vector matrix  $\vec{\mathbf{A}}$  containing the coefficients of the variables and  $\vec{\mathbf{V}}$  is the vector containing the variables. The constraints are represented by the equation

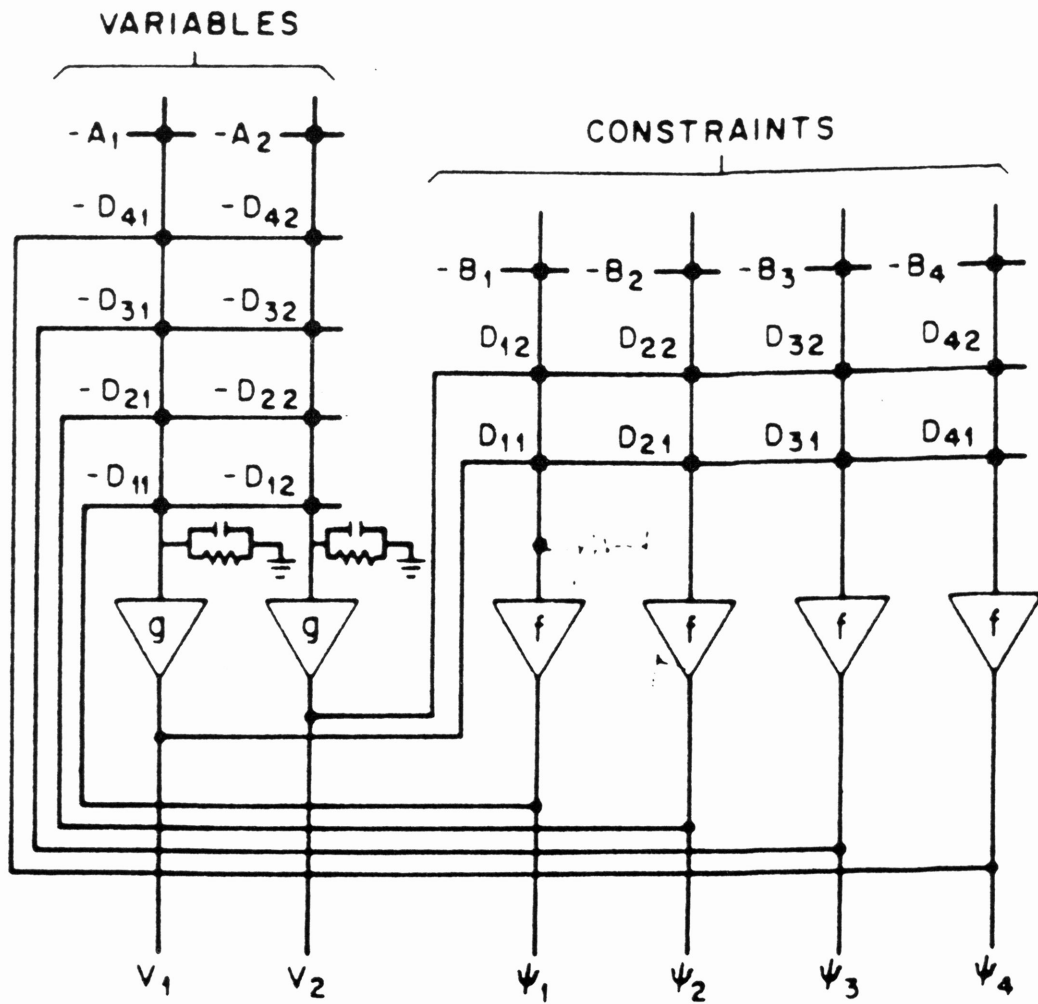


Figure 4.1. The organization of a network which will solve a 2-variable 4-constraint linear programming problem.

$$f(\vec{\mathbf{V}}) = \mathbf{D} \cdot \vec{\mathbf{V}} - \vec{\mathbf{b}} \geq 0$$

where  $\mathbf{D}$  is a matrix containing the coefficients of the variables in the constraint equations,  $\vec{\mathbf{V}}$  is the vector of the variables, and  $\vec{\mathbf{b}}$  is the vector containing the constants.

For the two-variable, four-constraint problem demonstrated by Tank and Hopfield the cost function to be minimized is

$$\Phi(V_1, V_2) = a_1 V_1 + a_2 V_2$$

where  $V_1$  and  $V_2$  are variables and  $a_1$  and  $a_2$  are the coefficients. The constraints are represented by the following equations:

$$f_1(\vec{\mathbf{V}}) = D_{11}V_1 + D_{12}V_2 - b_1 \geq 0$$

$$f_2(\vec{\mathbf{V}}) = D_{21}V_1 + D_{22}V_2 - b_2 \geq 0$$

$$f_3(\vec{\mathbf{V}}) = D_{31}V_1 + D_{32}V_2 - b_3 \geq 0$$

$$f_4(\vec{\mathbf{V}}) = D_{41}V_1 + D_{42}V_2 - b_4 \geq 0$$

where  $D_{ji}$  are coefficients of the variables in the constraint equations and  $b_j$  are the constants.

The circuit equations for the current flowing into the  $g$  amplifiers can be written



$$C_1 \frac{du_1}{dt} = -A_1 - \frac{x_1}{R} - \sum_{j=1}^4 D_{j1} \cdot f(z_j)$$

$$C_2 \frac{du_2}{dt} = -A_2 - \frac{x_2}{R} - \sum_{j=1}^4 D_{j2} \cdot f(z_j)$$

where  $C_i \frac{du_i}{dt}$  is the current across the capacitor and  $f(z_j)$  is the output voltage of the  $f$  amplifier that when multiplied by the conductance  $D_{ji}$  becomes the current contribution from the  $f$  amplifier. These equations describe the circuit implementation of the ANN which will solve the linear programming problem by minimizing an energy function of the form:

$$E = \vec{\mathbf{A}}^T \cdot \vec{\mathbf{V}} + \sum_{j=1}^n F(\vec{\mathbf{D}}_j \cdot \vec{\mathbf{V}} - b_j) + \frac{1}{R} \int_0^{V_i} g^{-1}(V_i) dV$$

The time derivative of the energy equation becomes

$$\frac{dE}{dt} = \sum_i [A_i \frac{dV_i}{dt} + \frac{dV_i}{dt} \sum_j D_{ji} \cdot f(\vec{\mathbf{D}}_j \cdot \vec{\mathbf{V}} - b_j) + \frac{x_i}{R} \frac{dV_i}{dt}]$$

making substitutions step by step the following equations result:

$$\frac{dE}{dt} = \sum_i \frac{dV_i}{dt} [A_i + \sum_j D_{ji} \cdot f(\vec{\mathbf{D}}_j \cdot \vec{\mathbf{V}} - b_j) + \frac{x_i}{R}]$$

$$\frac{dE}{dt} = \sum_i \frac{dV_i}{dt} [-C_i \frac{dx_i}{dt}]$$

$$\frac{dE}{dt} = - \sum_i C_i \frac{dV_i}{dt} \frac{dx_i}{dt}$$

Since

$$x_i = \int_0^{V_i} g^{-1}(V_i) dV_i$$

and

$$\frac{dx_i}{dt} = g^{-1}(V_i) \frac{dV_i}{dt}$$

then

$$\frac{dE}{dt} = - \sum_i C_i g^{-1}(V_i) \left( \frac{dV_i}{dt} \right)^2$$

In the above equation observe that  $C_i$  is a positive number (value of the capacitance) and that  $\left( \frac{dV_i}{dt} \right)^2$  is a positive value because it is squared. Also,  $g^{-1}(V_i)$  is defined to be a monotone increasing function. Therefore, because of the negative sign in front of the sum of these terms,

$$\frac{dE}{dt} \leq 0$$

when

$$\frac{dE}{dt} = 0 \rightarrow \frac{dV_i}{dt} = 0 \forall i$$

Thus the network achieves a steady state when the energy function is minimized. The variables are no longer changing and thus have achieved values such that the  $\Phi$  equation is minimized.

## V. Simple One-Variable Case

To illustrate the simple one variable case, let

$$\Phi(v) = v$$

subject to the constraint

$$f(v) = v \geq 0$$

A block diagram is shown in Figure 5.1. In this block diagram,  $g$  is a linear amplifier such that  $V_i = g(x_i)$  and  $f$  is a non-linear (piecewise linear) amplifier such that

$$f(v) = \begin{cases} 0 & \text{if } v > 0 \\ -v & \text{if } v \leq 0 \end{cases}$$

If we write a node-voltage equation for the voltage  $x$ , we have

$$C \frac{dx}{dt} = -1 - \frac{x}{R} - \Psi$$

$$C \frac{dx}{dt} = -1 - \frac{g^{-1}(v)}{R} - f(v)$$

Then an energy equation of the form:

$$E(v) = v + \frac{1}{R} \int_0^v x \, dv + \int_0^{g(v)} f(v)$$

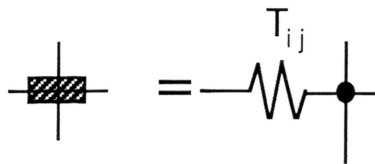
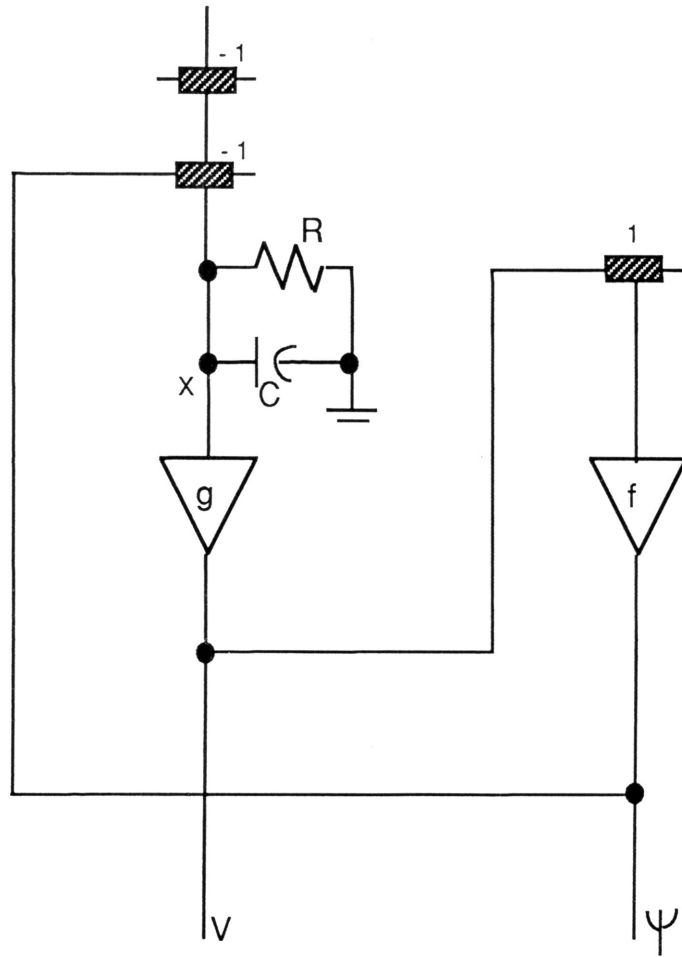
[3]

can be obtained and the time derivative of the energy function is

$$\frac{dE}{dt} = 1 + \frac{g^{-1}(v)}{R} + f(v)$$

Then, as the energy function is minimized (by supplying initial voltages and allowing the system to converge), the solution to the one-variable optimization problem will be determined.

FIGURE 5.1: ONE VARIABLE CASE



$$T_{ij} = 1/R_{ij}$$

$$V = g(x)$$

$$C \frac{dx}{dt} = -1 - \frac{x}{R} - \Psi$$

$$f(V) = \begin{cases} 0 & \text{if } V > 0 \\ -V & \text{if } V \leq 0 \end{cases}$$

## VI. Approach for Solving a System of Linear Equations

The reason for the emphasis on the Linear Programming Problem is its direct relationship to the problem of Solving a System of Linear Equations. Because of the similarity in the two problems, there is a direct mapping between them.

Solving a system of linear equations requires a variation of the Tank and Hopfield approach. In this case, the system of linear equations corresponds to the constraints on a cost function that, when minimized subject to the constraints (the system of linear equations), would simultaneously solve the system of linear equations. It was found that this cost function can be determined using numerical methods and analytic geometry. Specifically, the cost function will be of the same form as the constraint equations such that taking the gradient of the constraint equations will yield the coefficients of the cost function.

Let the system of linear equations (the constraint equations) be represented by

$$f(\vec{\mathbf{V}}) = \mathbf{D} \cdot \vec{\mathbf{V}} - \vec{\mathbf{b}} = 0$$

or

$$f(\vec{\mathbf{V}}) = \mathbf{D} \cdot \vec{\mathbf{V}} = \vec{\mathbf{b}}$$

Note that these constraints are now represented as an equality matrix equation as opposed to the inequality shown in the linear programming problem. An artificial function to be minimized, a cost function, is introduced in the form:

$$\vec{\mathbf{Y}} = \mathbf{A} \cdot \vec{\mathbf{V}}$$

The coefficients of this function need to be selected such that minimization gives the solution to  $\mathbf{D} \cdot \vec{\mathbf{V}} = \vec{\mathbf{b}}$ . Let

$$g_i(\vec{\mathbf{V}}) = \vec{\mathbf{b}}_i - \mathbf{D}_i^T \vec{\mathbf{V}} = 0$$

where  $D_i$  is the  $i^{\text{th}}$  column of  $\mathbf{D}$ , and let a Lagrange function be defined:

$$L(\vec{\mathbf{V}}, \vec{\lambda}) = Y(\vec{\mathbf{V}}) + \sum_{i=1}^n \lambda_i g_i(\vec{\mathbf{V}})$$

where the Lagrange multipliers  $\lambda_i \geq 0$ ,  $i = 1, 2, \dots, n$ . Minimization is achieved when the gradient of this Lagrange function is set equal to zero, so at the solution  $\vec{\mathbf{V}}_s$

$$\nabla L(\vec{\mathbf{V}}_s, \vec{\lambda}) = \nabla Y(\vec{\mathbf{V}}_s) + \sum_{i=1}^n \lambda_i \nabla g_i(\vec{\mathbf{V}}_s) = 0$$

Therefore

$$\nabla Y(\vec{\mathbf{V}}_s) = - \sum_{i=1}^n \lambda_i \nabla g_i(\vec{\mathbf{V}}_s) \lambda_i > 0$$

Then

$$\nabla Y(\vec{\mathbf{V}}_s) = \vec{\mathbf{A}}$$

and

$$\nabla g_i(\vec{\mathbf{V}}_s) = -\vec{\mathbf{D}}_i$$

Therefore

$$\vec{\mathbf{A}} = - \sum_{i=1}^n \lambda_i (-\vec{\mathbf{D}}_i)$$

$$A_j = - \sum_{i=1}^n \lambda_i (-D_{ij})$$

The choice of  $\lambda_i$ , the Lagrange multiplier, is somehow arbitrary, but should be chosen such that the fastest convergence is achieved [4]. If the Lagrange multipliers are chosen to be 1, then

$$A_j = \sum_{i=1}^n D_{ij}$$

Thus, for a two-variable case:

$$A_1 = D_{11} + D_{21}$$

$$A_2 = D_{12} + D_{22}$$

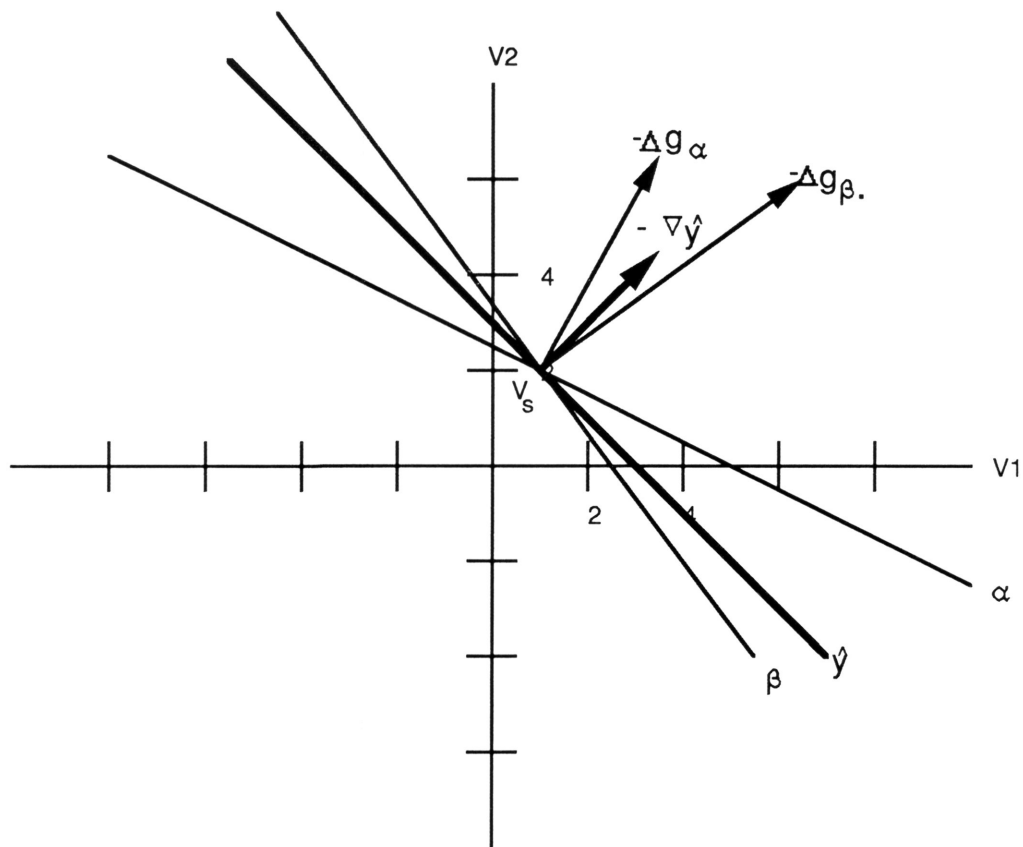
$$\vec{\mathbf{A}} = \begin{bmatrix} A_1 \\ A_2 \end{bmatrix} = \begin{bmatrix} D_{11} + D_{21} \\ D_{12} + D_{22} \end{bmatrix}$$

where  $\vec{\mathbf{A}}$  is the coefficient vector matrix of the cost function,  $i$  of  $D_{ij}$  is the number of the constraint equation and  $j$  is the number of the variable in the constraint equation, i.e.,  $D_{12}$  is the coefficient of the  $2^{nd}$  variable of the  $1^{st}$  equation. Figure 6.1 shows a graphical representation.

## **VII. Two Variable Example Voltage-Controlled Current Source (VCCS) Neuron Implementation**

Upon determination of the cost function, there is a direct mapping to the Tank and Hopfield Linear Programming Problem. Using a specific two-variable case, the mapping was performed and

FIGURE 6.1: GRADIENTS OF 2X2 EXAMPLE



$$\alpha : V_1 + 2V_2 = 5$$

$$\beta : 4V_1 + 3V_2 = 10$$

$$\hat{y} = 5V_1 + 5V_2$$



the resulting neural network was simulated with the SPICE simulation program in order to check the validity of the proposed approach. In this simulation, ideal elements were used at first to confirm the theoretical predictions. Voltage-controlled current sources were used as connective elements (corresponding to resistors in Tank and Hopfield's approach), diodes were used as the  $f$  amplifiers and a diode/voltage-controlled current source combination were implemented for the  $g$  amplifiers. In this two variable example a system of two equations is mapped to the constraint equations of the linear programming problem.

$$f_1(\vec{\mathbf{V}}) = 1 V_1 + 2 V_2 = 5$$

$$f_2(\vec{\mathbf{V}}) = 4 V_1 + 3 V_2 = 10$$

Recall the method for determining the coefficients of the cost function to be minimized (for the two-variable case)

$$\vec{\mathbf{A}} = \begin{bmatrix} D_{11} + D_{21} \\ D_{12} + D_{22} \end{bmatrix}$$

Thus, the cost function can be defined

$$\Phi(\vec{\mathbf{V}}) = \vec{\mathbf{A}}^T \vec{\mathbf{V}}$$

$$\Phi(V_1, V_2) = 5 V_1 + 5 V_2$$

Consider the energy function described by Tank and Hopfield in the linear programming problem:

$$E = \vec{\mathbf{A}}^T \cdot \vec{\mathbf{V}} + \sum_{j=1}^n F(\vec{\mathbf{D}}_j \cdot \vec{\mathbf{V}} - b_j) + \frac{1}{R} \int_0^{V_i} g^{-1}(V_i) dV$$

For this application

$$F(\vec{\mathbf{D}}_j \cdot \vec{\mathbf{V}} - b_j) = U(f_j) f_j(\vec{\mathbf{V}})$$

where

$$U(f_j) = \begin{cases} 0 & \text{if } f_j > 0 \\ f_j & \text{if } f_j \leq 0 \end{cases}$$

Recall the time derivative of the energy function

$$\frac{dE}{dt} = - \sum_i C_i g^{-1}(V_i) \left( \frac{dV_i}{dt} \right)^2$$

If both sides are divided by  $\frac{dV_i}{dt}$  the following result is obtained:

$$\frac{dE}{dV_i} = - \sum_i C_i g^{-1}(V_i) \frac{dV_i}{dt}$$

Applying this relation to the two-variable example at hand, a circuit implementation similar to that of the quadratic programming problem [3] can be visualized.

With the substitution of values, the energy equation becomes:

$$E(V_1, V_2) = [5V_1 + 5V_2] + U(f_1) \cdot (1V_1 + 2V_2 - 5) + U(f_2) \cdot (4V_1 + 3V_2 - 10)$$

$$+ \frac{1}{R} \int_0^{V_1} g^{-1}(V_1) dV + \frac{1}{R} \int_0^{V_2} g^{-1}(V_2) dV$$

Taking the derivative with respect to  $V_i$

$$-\frac{dE}{dV_1} = -5 - U(f_1) \cdot (1) - U(f_2) \cdot (4) - \frac{1}{R}g^{-1}(V_1)$$

$$-\frac{dE}{dV_2} = -10 - U(f_1) \cdot (2) - U(f_2) \cdot (3) - \frac{1}{R}g^{-1}(V_2)$$

A circuit can then be constructed from these equations. (The actual mapping to the circuit is shown in Appendix F.) The diagram for this two-variable VCCS ANN implementation is shown in Figure 7.1. In Figure 7.1, the first two long horizontal lines represent the constraint equations and the last two long horizontal lines represent the  $\frac{dE}{dV_i}$  equations. All the trapezoidal boxes are the voltage-controlled current sources (VCCSs) which multiply the input voltage by the gain (value of the coefficient) and output the result as a current. The *current-controlled voltage* sources function as the  $g$  amplifiers. The diodes correspond to constraint satisfaction,  $U(f_j)$  in the equations and the transfer function for the diode can be described by Figure 7.2 where

$$U(f_i) = \begin{cases} 0 & \text{if } f_i > 0 \\ f_i & \text{if } f_i \leq 0 \end{cases}$$

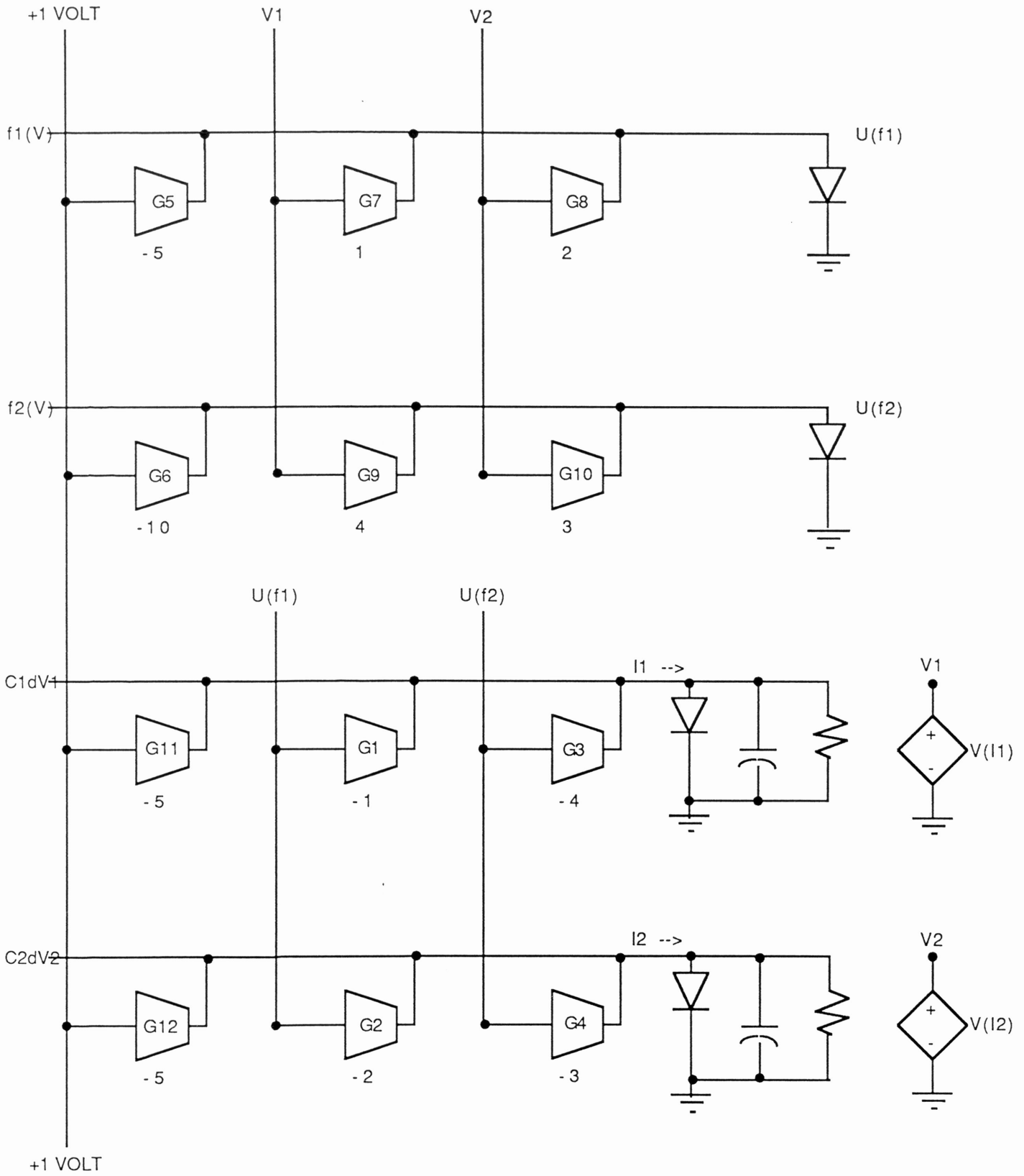
Finally, the capacitors allow for integrative analog summation of input currents from other processors and partially define the time constants of the ANN [2].

The circuit equations for the current flowing into the  $g$  amplifiers can be written

$$C_1 \frac{du_1}{dt} = -A_1 - \frac{x_1}{R} - \sum_{j=1}^2 D_{j1} \cdot f(z_j)$$

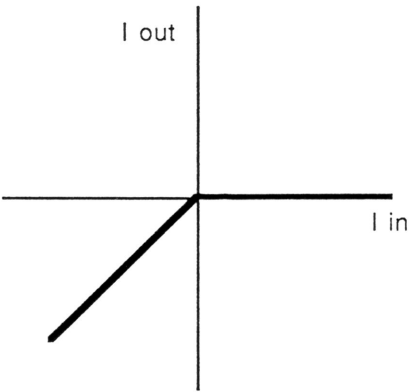
$$C_2 \frac{du_2}{dt} = -A_2 - \frac{x_2}{R} - \sum_{j=1}^2 D_{j2} \cdot f(z_j)$$

FIGURE 7.1: BLOCK DIAGRAM FOR 2X2 "VCCS" ANN

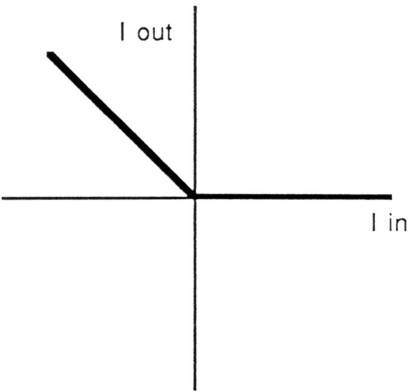


$$\frac{V}{\text{VOLTAGE}} \rightarrow \begin{matrix} \text{G\#} \\ \text{C} \end{matrix} \rightarrow \frac{I=C*V}{\text{CURRENT}} = \text{VCCS}$$

FIGURE 7.2: COMPARISON OF AMPLIFIER TRANSFER FUNCTIONS



TRANSFER FUNCTION FOR "VCCS" ANN  
(FUNCTION OF DIODES IN FIG. 7.1)



TRANSFER FUNCTION OF HOPFIELD  
("F" AMPLIFIER IN LPP)

These equations describe the circuit implementation of the ANN which will solve the system of two linear equations as the energy function is minimized (by supplying initial voltages and allowing the system to converge). In this example, the solution to the system of linear equations is at  $V_1 = 2$  and  $V_2 = 2$  so that

$$1 \cdot 2 + 2 \cdot 2 = 5$$

$$4 \cdot 2 + 3 \cdot 2 = 10$$

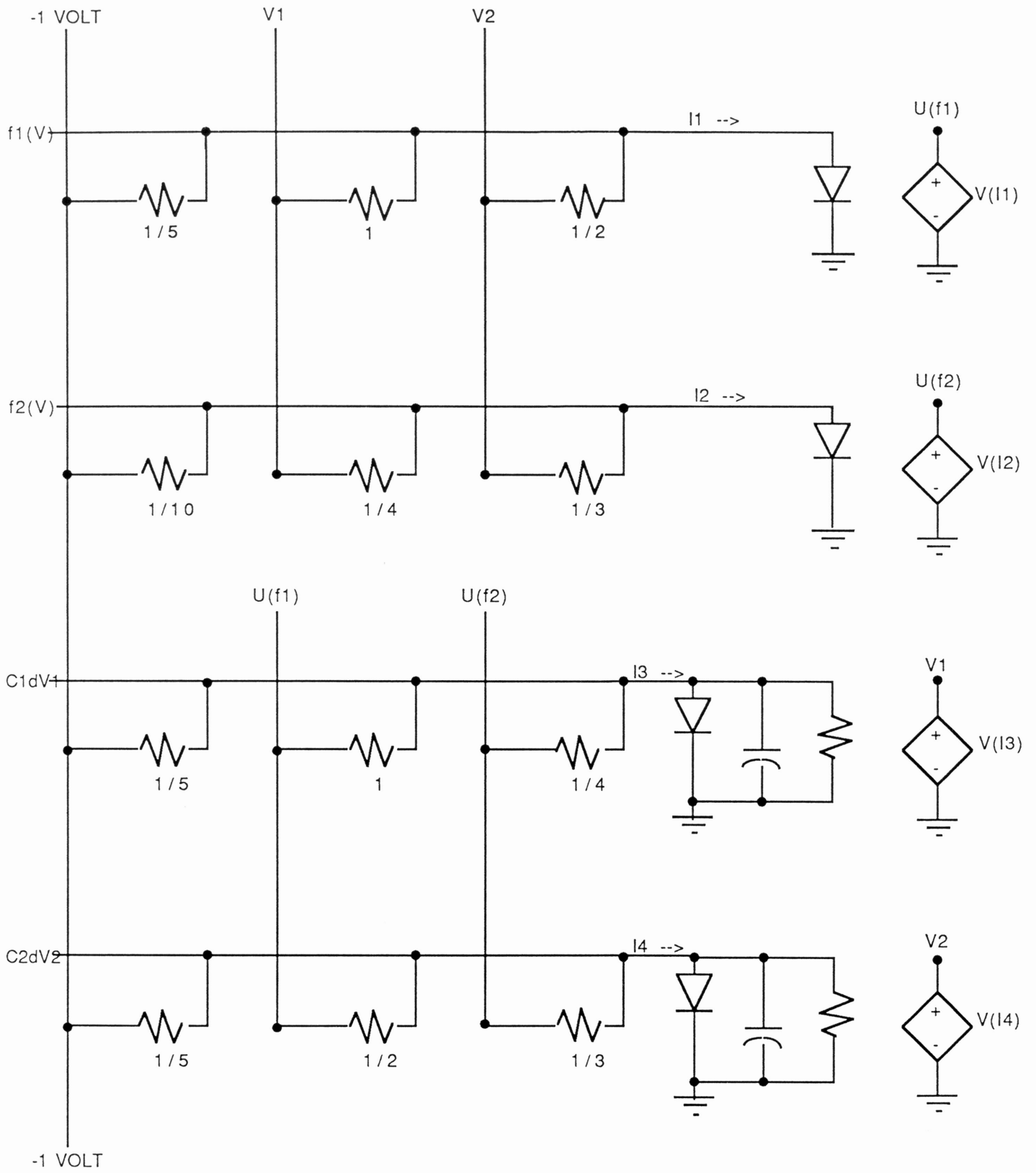
The SPICE simulation reached this solution in 10 ns (1E-8 seconds) when the above ANN was simulated. (Appendix A: SPICE File for the 2x2 VCCS ANN Implementation).

### **VIII. Two Variable Example: Resistor Neuron ANN Implementation**

The implementation using voltage-controlled current sources can be altered by substituting resistors for the voltage-controlled current sources to simplify the components in the circuit. Using the same system of equations as an example, a resistor neuron implementation was developed as shown in Figure 8.1. The difference between Tank and Hopfield's implementation and this implementation involves the functions of the  $f$  and  $g$  amplifiers. In most ANNs the processing elements require sigmoidal functions for correct operation. However, because of the nature of this problem, piecewise linear functions are appropriate. In Tank and Hopfield's Linear Programming Problem (Figure 4.1), the  $g$  amplifier was characterized by  $v_i = g(x_i)$  where  $g_i$  is a linear function. The  $f$  amplifier was defined by

$$z_j = \vec{\mathbf{D}}_j \cdot \vec{\mathbf{V}} - b_j$$

FIGURE 8.1: DIAGRAM FOR 2X2 RESISTOR NEURON ANN



$$f(z) = \begin{cases} 0 & \text{if } z \geq 0 \\ -z & \text{if } z < 0 \end{cases}$$

[3]

Whereas, in the method of Solving a System of Linear Equations, both amplifiers have the same function. If  $z$  equals the current flowing into the amplifier,

$$f(z) = \begin{cases} 0 & \text{if } z \geq 0 \\ z & \text{if } z < 0 \end{cases}$$

This resistor neuron ANN implementation did not give an exact solution like the ideal VCCS implementation did, but only had an error of approximately .1%. The VCCS implementation achieved an exact solution because it was simulated with *ideal* voltage-controlled current sources with gains equal to the values of the coefficients, whereas the resistor neuron implementation involves resistor values equal to the *reciprocal* of the coefficient values. However, the convergence for the resistor neuron implementation was achieved within 10 ns, the same convergence time as the VCCS implementation. (Appendix B: SPICE File for the 2x2 Resistor Neuron Implementation)

### **IX. Expanding to a Twenty-Variable Example**

To further test the algorithm for development of a VCCS ANN, a system of twenty linear equations with twenty unknown variables was considered. The constraints (equations) are as follows:



$$\begin{aligned}
1. & \quad 2 V_1 + V_2 = 4 \\
2. & \quad 2 V_2 + 3 V_3 - 2 V_4 = 5 \\
3. & \quad 5 V_1 + 7 V_5 - V_6 = 10 \\
4. & \quad 3 V_7 - V_8 + 2 V_9 = 7 \\
5. & \quad 5 V_{10} + V_{11} - 3 V_{12} = 1 \\
6. & \quad 8 V_{13} + 2 V_{14} - V_{15} = 9 \\
7. & \quad 3 V_1 + V_{17} + 3 V_{18} - 2 V_{19} = 4 \\
8. & \quad 3 V_3 + V_{10} - 2 V_{20} = 3 \\
9. & \quad 6 V_5 - 3 V_{14} + V_{18} = 2 \\
10. & \quad -2 V_7 + 5 V_{12} - 4 V_{13} - 2 V_{17} = 8 \\
11. & \quad 3 V_4 + V_7 - V_{16} - V_{19} = 14 \\
12. & \quad 2 V_8 + 2 V_9 - 3 V_{13} + V_{20} = 11 \\
13. & \quad 3 V_4 - 2 V_5 - 6 V_{11} + 5 V_{19} = 7 \\
14. & \quad 7 V_1 + 5 V_5 - 8 V_9 - 3 V_{13} + V_{17} = 2 \\
15. & \quad V_2 - V_7 + V_{15} + V_{20} = 6 \\
16. & \quad 3 V_3 + V_5 + 2 V_{14} - 3 V_{19} = 5 \\
17. & \quad 5 V_9 - V_{12} + 2 V_{14} + V_{15} = 8 \\
18. & \quad 3 V_6 - V_9 + 2 V_{10} = 9 \\
19. & \quad -4 V_6 + 3 V_7 + 4 V_{13} = 5 \\
20. & \quad V_4 + 7 V_{17} - 5 V_{18} = 1
\end{aligned}$$

From these equations, the coefficients of the cost function equation were determined as follows:

$$\left| \begin{array}{c|c|c|c}
A_1 = 17 & A_2 = 4 & A_3 = 9 & A_4 = 5 \\
A_5 = 17 & A_6 = -2 & A_7 = 4 & A_8 = 1 \\
A_9 = 0 & A_{10} = 8 & A_{11} = -5 & A_{12} = 1 \\
A_{13} = 2 & A_{14} = 3 & A_{15} = 1 & A_{16} = -1 \\
A_{17} = 7 & A_{18} = -1 & A_{19} = 1 & A_{20} = 0
\end{array} \right|$$

such that  $\Phi = \vec{\mathbf{A}}^T \cdot \vec{\mathbf{V}}$  is the cost function to be minimized subject to the constraints (system of twenty linear equations).

The solution to this 20x20 (twenty equations, twenty unknowns) system is as follows:

$$\left| \begin{array}{c|c|c|c}
V_1 = 1 & V_2 = 2 & V_3 = 3 & V_4 = 4 \\
V_5 = 1 & V_6 = 2 & V_7 = 3 & V_8 = 4 \\
V_9 = 1 & V_{10} = 2 & V_{11} = 3 & V_{12} = 4 \\
V_{13} = 1 & V_{14} = 2 & V_{15} = 3 & V_{16} = 4 \\
V_{17} = 1 & V_{18} = 2 & V_{19} = 3 & V_{20} = 4
\end{array} \right|$$

The 20x20 ANN was simulated in the SPICE simulation program and the above solutions were exactly determined within 11ns (1.1E-8 sec). (Appendix C: SPICE File for the 20x20 VCCS ANN Implementation).

## X. Observations

The value of the capacitors were varied during simulation of the twenty-variable example to experiment with convergence times.

<i>Cap.Value(F)</i>	<i>Time(sec)</i>
$1E - 8$	<i>oscillation</i>
$1E - 11$	$1.1E - 8$
$1E - 13$	$1.0E - 8$
$1E - 16$	$1.0E - 8$

Thus making the capacitance a little smaller makes the time constant shorter by approximately 1 ns. If the capacitance is too big, the time constant is too large for convergence to be achieved. For simulations in which convergence was achieved the solutions were the same regardless of the capacitor value.

Analysis of resistor values was also performed (this time in the two-variable example) to observe the effects of inaccuracies in these components. There are inaccuracies associated with the resistor neuron ANN implementation itself because the resistor values are reciprocals of the coefficients values. This oftentimes results in irrational fraction values (e.g.  $\frac{1}{3}$  which can only be approximated. Additionally, circuits were simulated in which the resistors values were altered slightly. When the two resistors were altered by 1%, the solutions developed errors around 5%. When six resistors were altered, the errors of the solutions varied by 13% for one and 5.8% for the other. Therefore, as more errors were introduced in the resistors, there was a corresponding error in the solutions. However, if there is only a small error in the resistor values, a close approximation to the actual solution can still be obtained.

As a final observation, note that expanding from a two variable case to a twenty variable case

with the VCCS ANN implementation increased the convergence time by only 1 ns. Present-day serial computers would have a much higher increase in the time needed to solve a twenty variable case as opposed to a two variable case. Thus, the advantage of parallel processing in the ANN is obvious. Expansion is not a problem with the parallel architecture of the ANN.

### **XI. Automating the Process**

It is obvious that the input file for SPICE simulation is quite large for a system of twenty linear equations and quite cumbersome to produce. Therefore, for the purpose of experimenting with the proposed approach, a computer program was written to automatically generate the SPICE input file given the system of equations for which solution is required. The program was written in the "Turbo C" programming language and is listed in Appendix D. The output of the software program (which is the input file for SPICE simulation) is shown in Appendix E.

### **XII. Conclusion**

It is readily apparent that the solution to a system of linear equations can be determined using Artificial Neural Networks by a direct mapping to the Linear Programming Problem done by Tank and Hopfield. This is accomplished by using the system of linear equations as the constraints and determining a cost function of the same form to be minimized by standard optimization methods. The resulting Artificial Neural Network is a highly interconnected network of simple processing elements which can be resistors or voltage-controlled current source amplifiers.

The resulting implications are that the hardware implementation of the ANN circuitry has the potential of providing a very quick solution to a problem that presently consumes a large amount of CPU time. This circuitry could potentially become a section of a neural computer as well. However, some improvements may be necessary before an ANN solution to a system of linear equations is

fully implemented. Presently, the ANN is problem specific (i.e., a distinct ANN is required for each system of linear equations). Implementation of reprogrammable connections, in which the weights of the processing elements can be altered, will expand the capabilities of the ANN to solve a variety of systems. Also, inaccuracies in components such as voltage supplies, resistors, amplifiers, etc. contribute to a less accurate solution. However, even if the errors are present in the solution, the methodology can still be used for those cases where the accuracy of the solution is not critical, e.g., in solving certain classification and decision making problems. Therefore, further improvements in technology are required to utilize the ANN to solve a variety of systems of linear equations.

# Bibliography

- [1] “Artificial Neural Networks: An Introduction”. V. Vemuri. Artificial Neural Networks: Theoretical Concepts. Computer Society Press Technology Series. pp. 1-7. 1988.
- [2] J.J. Hopfield and D.W. Tank. “Simple ‘Neural’ Optimization Networks: An A/D Converter, Signal Decision Circuit and a Linear Programming Circuit. ” IEEE Transactions on Circuits and Systems. Vol. cas-33, no.5. pp. 533-541. May 1986.
- [3] E. Sanchez-Sinencio. ELEN 489: Neural Networks and Implementations Lecture Notes. Department of Electrical Engineering. Texas A&M University. pp. A30-A40. Fall 1989.
- [4] M. Styblinski (Research Advisor). Notes. Department of Electrical Engineering. Texas A&M University. Research advisor. Fall 1989 and Spring 1990.

APPENDIX A: SPICE FILE FOR THE 2X2 VCCS ANN IMPLEMENTATION

INPUT LISTING

TEMPERATURE = 27.000 DEG C

```

***FIRST CONSTRAINT***
VS 1 0 PWL(0 0 10NS 1)
RS 1 0 1
G1 2 0 1 0 5
G2 2 0 9 0 -1
G3 2 0 13 0 -2
D1 2 3 DIODE
VOFF1 3 0 DC -.7
***SECOND CONSTRAINT***
G4 4 0 1 0 10
G5 4 0 9 0 -4
G6 4 0 13 0 -3
D2 4 5 DIODE
VOFF2 5 0 DC -.7
***X1 EQUATION***
G7 6 0 1 0 5
G8 6 0 2 0 1
G9 6 0 4 0 4
D3 6 7 DIODE
VOFF3 7 0 DC -.7
VX1 6 8 AC 0
R1 8 0 1
H1 9 0 VX1 100
C1 9 0 1E-11
***X2 EQUATION***
G10 10 0 1 0 5
G11 10 0 2 0 2
G12 10 0 4 0 3
D4 10 11 DIODE
VOFF4 11 0 DC -.7
VX2 10 12 AC 0
R2 12 0 1
H2 13 0 VX2 100
C2 13 0 1E-11
*****
.MODEL DIODE D
.WIDTH OUT=80
.TRAN 1NS 20NS
.PRINT TRAN V(9) V(13)
.END

```

APPENDIX A (CONTINUED): RESULTS OF SIMULATION

TRANSIENT ANALYSIS

TEMPERATURE = 27.000 DEG C

TIME	V(9)	V(13)
0.000E+00	-1.080E-03	3.251E-03
1.000E-09	9.998E-02	2.001E-01
2.000E-09	2.000E-01	4.000E-01
3.000E-09	3.000E-01	6.000E-01
4.000E-09	4.000E-01	8.000E-01
5.000E-09	5.000E-01	1.000E+00
6.000E-09	6.000E-01	1.200E+00
7.000E-09	7.000E-01	1.400E+00
8.000E-09	8.000E-01	1.600E+00
9.000E-09	9.000E-01	1.800E+00
1.000E-08	1.000E+00	2.000E+00
1.100E-08	1.000E+00	2.000E+00
1.200E-08	1.000E+00	2.000E+00
1.300E-08	1.000E+00	2.000E+00
1.400E-08	1.000E+00	2.000E+00
1.500E-08	1.000E+00	2.000E+00
1.600E-08	1.000E+00	2.000E+00
1.700E-08	1.000E+00	2.000E+00
1.800E-08	1.000E+00	2.000E+00
1.900E-08	1.000E+00	2.000E+00
2.000E-08	1.000E+00	2.000E+00

JOB CONCLUDED

TIME	PAGE	DIRECT	BUFFERED
CPU	FAULTS	I/O	I/O
0: 0: 3.62	84	2	1
0: 0: 4.78			
TOTAL JOB TIME		3.62	

Appendix B: Spice File for 2x2 Resistor Neuron Implementation

INPUT LISTING

TEMPERATURE = 27.000 DEG C

VS 1 0 PWL(0 0 10NS -1)

\*\*\*F1(X)\*\*\*

R1 1 2 .2

R2 5 2 1

R3 9 2 .5

D1 2 3 DIODE

VX3 2 13 AC 0

R20 13 0 1

VOFF1 3 0 -.7

\*\*VCVS\*\*

H3 12 0 VX3 1E3

R17 12 0 1

\*\*\*F2(X)\*\*\*

R4 1 4 .1

R5 5 4 .25

R6 9 4 .333333

D2 4 3 DIODE

VX4 4 15 AC 0

R19 15 0 1

\*\*VCVS\*\*

H4 14 0 VX4 1E3

R18 14 0 1

\*\*\*X1\*\*\*

R7 1 6 .2

R8 12 6 1

R9 14 6 .25

C3 6 0 1E-11

D3 6 3 DIODE

VX1 6 7 AC 0

R10 7 0 1

H1 5 0 VX1 1E3

R15 5 0 1

\*\*\*X2\*\*\*

R11 1 8 .2

R12 12 8 .5

R13 14 8 .333333

C4 8 0 1E-11

D4 8 3 DIODE

VX2 8 11 AC 0

R14 11 0 1

H2 9 0 VX2 1E3

R16 9 0 1

\*\*\*\*\*

.MODEL DIODE D

.WIDTH OUT=80

.TRAN 1NS 20NS

.PRINT TRAN V(5) V(9)

.END



APPENDIX B (CONTINUED): RESULTS OF SIMULATION

TRANSIENT ANALYSIS

TEMPERATURE = 27.000 DEG C

TIME	V(5)	V(9)
0.000E+00	-1.133E-03	3.409E-03
1.000E-09	9.904E-02	2.038E-01
2.000E-09	1.992E-01	4.042E-01
3.000E-09	2.994E-01	6.046E-01
4.000E-09	3.995E-01	8.049E-01
5.000E-09	4.997E-01	1.005E+00
6.000E-09	5.999E-01	1.206E+00
7.000E-09	7.001E-01	1.406E+00
8.000E-09	8.002E-01	1.606E+00
9.000E-09	9.004E-01	1.807E+00
1.000E-08	1.001E+00	2.007E+00
1.100E-08	1.001E+00	2.007E+00
1.200E-08	1.001E+00	2.007E+00
1.300E-08	1.001E+00	2.007E+00
1.400E-08	1.001E+00	2.007E+00
1.500E-08	1.001E+00	2.007E+00
1.600E-08	1.001E+00	2.007E+00
1.700E-08	1.001E+00	2.007E+00
1.800E-08	1.001E+00	2.007E+00
1.900E-08	1.001E+00	2.007E+00
2.000E-08	1.001E+00	2.007E+00

JOB CONCLUDED

TIME	PAGE	DIRECT	BUFFERED
CPU	FAULTS	I/O	I/O
0: 0: 2.02	75	2	1
TOTAL JOB TIME		2.02	

Appendix C: SPICE File for the 20x20 VCCS ANN Implementation

INPUT LISTING

TEMPERATURE = 27.000 DEG C

```
**** "-C" EQNS ***  
G1 2 0 9 0 -2  
G2 2 0 13 0 -1  
G3 4 0 13 0 -2  
G4 4 0 17 0 -3  
G5 4 0 21 0 2  
G6 6 0 9 0 -5  
G7 6 0 25 0 -7  
G8 6 0 29 0 1  
G9 8 0 33 0 -3  
G10 8 0 37 0 1  
G11 8 0 41 0 -2  
G12 10 0 45 0 -5  
G13 10 0 49 0 -1  
G14 10 0 53 0 3  
G15 12 0 57 0 -8  
G16 12 0 61 0 -2  
G17 12 0 65 0 1  
G18 14 0 9 0 -3  
G19 14 0 73 0 -1  
G20 14 0 77 0 -3  
G21 14 0 81 0 2  
G22 16 0 17 0 -3  
G23 16 0 45 0 -1  
G24 16 0 85 0 2  
G25 18 0 25 0 -6  
G26 18 0 61 0 3  
G27 18 0 77 0 -1  
G28 20 0 53 0 -5  
G29 20 0 73 0 2  
G30 20 0 33 0 2  
G31 20 0 57 0 4  
G32 22 0 21 0 -3  
G33 22 0 81 0 -1  
G34 22 0 69 0 1  
G35 22 0 33 0 -1  
G36 24 0 37 0 -2  
G37 24 0 41 0 -2  
G38 24 0 57 0 3  
G39 24 0 85 0 -1  
G40 26 0 21 0 -3  
G41 26 0 81 0 -5  
G42 26 0 49 0 6  
G43 26 0 25 0 2  
G44 28 0 9 0 -7  
G45 28 0 25 0 -5  
G46 28 0 41 0 8  
G47 28 0 57 0 3  
G48 28 0 73 0 -1  
G49 30 0 13 0 -1  
G50 30 0 65 0 -1
```

APPENDIX C (CONTINUED):

G51 30 0 33 0 1  
G52 30 0 85 0 -1  
G53 32 0 17 0 -3  
G54 32 0 81 0 3  
G55 32 0 61 0 -2  
G56 32 0 25 0 -1  
G57 34 0 61 0 -2  
G58 34 0 53 0 1  
G59 34 0 41 0 -5  
G60 34 0 65 0 -1  
G61 36 0 29 0 -3  
G62 36 0 41 0 1  
G63 36 0 45 0 -2  
G64 38 0 57 0 -4  
G65 38 0 29 0 4  
G66 38 0 33 0 -3  
G67 40 0 73 0 -7  
G68 40 0 77 0 5  
G69 40 0 21 0 -1  
\*\*\* "B" EQNS \*\*\*  
G70 2 0 1 0 4  
G71 4 0 1 0 5  
G72 6 0 1 0 10  
G73 8 0 1 0 7  
G74 10 0 1 0 1  
G75 12 0 1 0 9  
G76 14 0 1 0 4  
G77 16 0 1 0 3  
G78 18 0 1 0 2  
G79 20 0 1 0 8  
G80 22 0 1 0 14  
G81 24 0 1 0 11  
G82 26 0 1 0 7  
G83 28 0 1 0 2  
G84 30 0 1 0 6  
G85 32 0 1 0 5  
G86 34 0 1 0 8  
G87 36 0 1 0 9  
G88 38 0 1 0 5  
G89 40 0 1 0 1  
\*\*\* "C" EQNS \*\*\*  
G90 42 0 2 0 2  
G91 44 0 2 0 1  
G92 44 0 4 0 2  
G93 46 0 4 0 3  
G94 48 0 4 0 -2  
G95 42 0 6 0 5  
G96 50 0 6 0 7  
G97 52 0 6 0 -1  
G98 54 0 8 0 3  
G99 56 0 8 0 -1  
G100 58 0 8 0 2

APPENDIX C (CONTINUED):

G101 60 0 10 0 5  
 G102 62 0 10 0 1  
 G103 64 0 10 0 -3  
 G104 66 0 12 0 8  
 G105 68 0 12 0 2  
 G106 70 0 12 0 -1  
 G107 42 0 14 0 3  
 G108 74 0 14 0 1  
 G109 76 0 14 0 3  
 G110 78 0 14 0 -2  
 G111 46 0 16 0 3  
 G112 60 0 16 0 1  
 G113 80 0 16 0 -2  
 G114 50 0 18 0 6  
 G115 68 0 18 0 -3  
 G116 76 0 18 0 1  
 G117 64 0 20 0 5  
 G118 74 0 20 0 -2  
 G119 54 0 20 0 -2  
 G120 66 0 20 0 -4  
 G121 48 0 22 0 3  
 G122 78 0 22 0 1  
 G123 72 0 22 0 -1  
 G124 54 0 22 0 1  
 G125 56 0 24 0 2  
 G126 58 0 24 0 2  
 G127 66 0 24 0 -3  
 G128 80 0 24 0 1  
 G129 48 0 26 0 3  
 G130 78 0 26 0 5  
 G131 62 0 26 0 -6  
 G132 50 0 26 0 -2  
 G133 42 0 28 0 7  
 G134 50 0 28 0 5  
 G135 58 0 28 0 -8  
 G136 66 0 28 0 -3  
 G137 74 0 28 0 1  
 G138 44 0 30 0 1  
 G139 70 0 30 0 1  
 G140 54 0 30 0 -1  
 G141 80 0 30 0 1  
 G142 46 0 32 0 3  
 G143 78 0 32 0 -3  
 G144 68 0 32 0 2  
 G145 50 0 32 0 1  
 G146 68 0 34 0 2  
 G147 64 0 34 0 -1  
 G148 58 0 34 0 5  
 G149 70 0 34 0 1  
 G150 52 0 36 0 3

APPENDIX C (CONTINUED):

```
G151 58 0 36 0 -1
G152 60 0 36 0 2
G153 66 0 38 0 4
G154 52 0 38 0 -4
G155 54 0 38 0 3
G156 74 0 40 0 7
G157 76 0 40 0 -5
G158 48 0 40 0 1
*** "A" EQNS ***
G159 42 0 1 0 17
G160 44 0 1 0 4
G161 46 0 1 0 9
G162 48 0 1 0 5
G163 50 0 1 0 17
G164 52 0 1 0 -2
G165 54 0 1 0 4
G166 56 0 1 0 1
G167 58 0 1 0 0
G168 60 0 1 0 8
G169 62 0 1 0 -5
G170 64 0 1 0 1
G171 66 0 1 0 2
G172 68 0 1 0 3
G173 70 0 1 0 1
G174 72 0 1 0 -1
G175 74 0 1 0 7
G176 76 0 1 0 -1
G177 78 0 1 0 1
G178 80 0 1 0 0
*** OTHER INPUT INFO ***
VS 1 0 PWL(0 0 10NS 1)
RS 1 0 1
** DIODE/VOLTAGE OFFSET **
D1 2 3 DIODE
VOFF1 3 0 DC -.7
D2 4 3 DIODE
D3 6 3 DIODE
D4 8 3 DIODE
D5 10 3 DIODE
D6 12 3 DIODE
D7 14 3 DIODE
D8 16 3 DIODE
D9 18 3 DIODE
D10 20 3 DIODE
D11 22 3 DIODE
D12 24 3 DIODE
D13 26 3 DIODE
D14 28 3 DIODE
D15 30 3 DIODE
```

APPENDIX C (CONTINUED):

D16 32 3 DIODE  
D17 34 3 DIODE  
D18 36 3 DIODE  
D19 38 3 DIODE  
D20 40 3 DIODE  
D21 42 3 DIODE  
D22 44 3 DIODE  
D23 46 3 DIODE  
D24 48 3 DIODE  
D25 50 3 DIODE  
D26 52 3 DIODE  
D27 54 3 DIODE  
D28 56 3 DIODE  
D29 58 3 DIODE  
D30 60 3 DIODE  
D31 62 3 DIODE  
D32 64 3 DIODE  
D33 66 3 DIODE  
D34 68 3 DIODE  
D35 70 3 DIODE  
D36 72 3 DIODE  
D37 74 3 DIODE  
D38 76 3 DIODE  
D39 78 3 DIODE  
D40 80 3 DIODE  
VX1 42 82 AC 0  
R1 82 0 1  
VX2 44 84 AC 0  
R2 84 0 1  
VX3 46 86 AC 0  
R3 86 0 1  
VX4 48 88 AC 0  
R4 88 0 1  
VX5 50 90 AC 0  
R5 90 0 1  
VX6 52 92 AC 0  
R6 92 0 1  
VX7 54 94 AC 0  
R7 94 0 1  
VX8 56 96 AC 0  
R8 96 0 1  
VX9 58 98 AC 0  
R9 98 0 1  
VX10 60 100 AC 0  
R10 100 0 1

APPENDIX C (CONTINUED):

VX11 62 102 AC O  
R11 102 O 1  
VX12 64 104 AC O  
R12 104 O 1  
VX13 66 106 AC O  
R13 106 O 1  
VX14 68 108 AC O  
R14 108 O 1  
VX15 70 110 AC O  
R15 110 O 1  
VX16 72 112 AC O  
R16 112 O 1  
VX17 74 114 AC O  
R17 114 O 1  
VX18 76 116 AC O  
R18 116 O 1  
VX19 78 118 AC O  
R19 118 O 1  
VX20 80 120 AC O  
R20 120 O 1

\*\*\*\*\*

C1 2 O 1E-9  
C2 4 O 1E-9  
C3 6 O 1E-9  
C4 8 O 1E-9  
C5 10 O 1E-9  
C6 12 O 1E-9  
C7 14 O 1E-9  
C8 16 O 1E-9  
C9 18 O 1E-9  
C10 20 O 1E-9  
C11 22 O 1E-9  
C12 24 O 1E-9  
C13 26 O 1E-9  
C14 28 O 1E-9  
C15 30 O 1E-9  
C16 32 O 1E-9  
C17 34 O 1E-9  
C18 36 O 1E-9  
C19 38 O 1E-9  
C20 40 O 1E-9  
C21 42 O 1E-9  
C22 44 O 1E-9  
C23 46 O 1E-9  
C24 48 O 1E-9  
C25 50 O 1E-9  
C26 52 O 1E-9  
C27 54 O 1E-9  
C28 56 O 1E-9  
C29 58 O 1E-9  
C30 60 O 1E-9

APPENDIX C (CONTINUED):

C31 62 0 1E-9  
C32 64 0 1E-9  
C33 66 0 1E-9  
C34 68 0 1E-9  
C35 70 0 1E-9  
C36 72 0 1E-9  
C37 74 0 1E-9  
C38 76 0 1E-9  
C39 78 0 1E-9  
C40 80 0 1E-9

\*\*\*\*\*

H1 9 0 VX1 1E9  
R21 9 0 1  
H2 13 0 VX2 1E9  
R22 13 0 1  
H3 17 0 VX3 1E9  
R23 17 0 1  
H4 21 0 VX4 1E9  
R24 21 0 1  
H5 25 0 VX5 1E9  
R25 25 0 1  
H6 29 0 VX6 1E9  
R26 29 0 1  
H7 33 0 VX7 1E9  
R27 33 0 1  
H8 37 0 VX8 1E9  
R28 37 0 1  
H9 41 0 VX9 1E9  
R29 41 0 1  
H10 45 0 VX10 1E9  
R30 45 0 1  
H11 49 0 VX11 1E9  
R31 49 0 1  
H12 53 0 VX12 1E9  
R32 53 0 1  
H13 57 0 VX13 1E9  
R33 57 0 1  
H14 61 0 VX14 1E9  
R34 61 0 1  
H15 65 0 VX15 1E9  
R35 65 0 1  
H16 69 0 VX16 1E9  
R36 69 0 1  
H17 73 0 VX17 1E9  
R37 73 0 1  
H18 77 0 VX18 1E9  
R38 77 0 1  
H19 81 0 VX19 1E9  
R39 81 0 1  
H20 85 0 VX20 1E9  
R40 85 0 1

\*\*\* DOT COMMANDS \*\*\*

.MODEL DIODE D  
.WIDTH OUT=80  
.TRANS 1NS 20NS  
.PRINT TRAN V(9) V(13) V(17) V(21) V(25) V(29) V(33) V(37) V(41) V(45)  
.PRINT TRAN V(49) V(53) V(57) V(61) V(65) V(69) V(73) V(77) V(81) V(85)  
.END



APPENDIX C (CONTINUED): RESULTS OF SIMULATION

TRANSIENT ANALYSIS

TEMPERATURE = 27.000 DEG C

TIME	V(9)	V(13)	V(17)	V(21)	V(25)
0.000E+00	4.180E-03	-4.061E-03	-1.344E-03	-1.041E-02	-8.297E-04
1.000E-09	1.605E-01	-2.090E-02	3.114E-01	2.461E-01	4.200E-02
2.000E-09	2.606E-01	1.789E-01	6.115E-01	6.460E-01	1.419E-01
3.000E-09	3.606E-01	3.788E-01	9.115E-01	1.046E+00	2.419E-01
4.000E-09	4.604E-01	5.792E-01	1.211E+00	1.446E+00	3.420E-01
5.000E-09	5.604E-01	7.792E-01	1.511E+00	1.846E+00	4.420E-01
6.000E-09	6.606E-01	9.789E-01	1.811E+00	2.246E+00	5.419E-01
7.000E-09	7.606E-01	1.179E+00	2.111E+00	2.646E+00	6.419E-01
8.000E-09	8.604E-01	1.379E+00	2.411E+00	3.046E+00	7.420E-01
9.000E-09	9.604E-01	1.579E+00	2.711E+00	3.446E+00	8.420E-01
1.000E-08	1.061E+00	1.779E+00	3.011E+00	3.846E+00	9.419E-01
1.100E-08	1.000E+00	2.000E+00	3.000E+00	4.000E+00	1.000E+00
1.200E-08	1.000E+00	2.000E+00	3.000E+00	4.000E+00	1.000E+00
1.300E-08	1.000E+00	2.000E+00	3.000E+00	4.000E+00	1.000E+00
1.400E-08	1.000E+00	2.000E+00	3.000E+00	4.000E+00	1.000E+00
1.500E-08	1.000E+00	2.000E+00	3.000E+00	4.000E+00	1.000E+00
1.600E-08	1.000E+00	2.000E+00	3.000E+00	4.000E+00	1.000E+00
1.700E-08	1.000E+00	2.000E+00	3.000E+00	4.000E+00	1.000E+00
1.800E-08	1.000E+00	2.000E+00	3.000E+00	4.000E+00	1.000E+00
1.900E-08	1.000E+00	2.000E+00	3.000E+00	4.000E+00	1.000E+00
2.000E-08	1.000E+00	2.000E+00	3.000E+00	4.000E+00	1.000E+00

TIME	V(49)	V(53)	V(57)	V(61)	V(65)
0.000E+00	-1.158E-02	2.686E-03	2.632E-03	-4.740E-03	9.053E-03
1.000E-09	2.145E-01	3.423E-01	1.183E-01	8.175E-02	3.101E-01
2.000E-09	5.145E-01	7.422E-01	2.183E-01	2.817E-01	6.100E-01
3.000E-09	8.145E-01	1.142E+00	3.183E-01	4.816E-01	9.101E-01
4.000E-09	1.115E+00	1.542E+00	4.183E-01	6.818E-01	1.210E+00
5.000E-09	1.415E+00	1.942E+00	5.183E-01	8.819E-01	1.510E+00
6.000E-09	1.715E+00	2.342E+00	6.183E-01	1.082E+00	1.810E+00
7.000E-09	2.014E+00	2.742E+00	7.183E-01	1.282E+00	2.110E+00
8.000E-09	2.315E+00	3.142E+00	8.183E-01	1.482E+00	2.410E+00
9.000E-09	2.615E+00	3.542E+00	9.183E-01	1.682E+00	2.710E+00
1.000E-08	2.914E+00	3.942E+00	1.018E+00	1.882E+00	3.010E+00
1.100E-08	3.000E+00	4.000E+00	1.000E+00	2.000E+00	3.000E+00
1.200E-08	3.000E+00	4.000E+00	1.000E+00	2.000E+00	3.000E+00
1.300E-08	3.000E+00	4.000E+00	1.000E+00	2.000E+00	3.000E+00
1.400E-08	3.000E+00	4.000E+00	1.000E+00	2.000E+00	3.000E+00
1.500E-08	3.000E+00	4.000E+00	1.000E+00	2.000E+00	3.000E+00
1.600E-08	3.000E+00	4.000E+00	1.000E+00	2.000E+00	3.000E+00
1.700E-08	3.000E+00	4.000E+00	1.000E+00	2.000E+00	3.000E+00
1.800E-08	3.000E+00	4.000E+00	1.000E+00	2.000E+00	3.000E+00
1.900E-08	3.000E+00	4.000E+00	1.000E+00	2.000E+00	3.000E+00
2.000E-08	3.000E+00	4.000E+00	1.000E+00	2.000E+00	3.000E+00

APPENDIX C (CONTINUED): RESULTS OF SIMULATION

TIME	V(29)	V(33)	V(37)
0.000E+00	1.400E-03	1.226E-03	4.583E-03
1.000E-09	1.963E-01	2.374E-01	3.396E-01
2.000E-09	3.963E-01	5.373E-01	7.395E-01
3.000E-09	5.963E-01	8.373E-01	1.140E+00
4.000E-09	7.963E-01	1.137E+00	1.540E+00
5.000E-09	9.963E-01	1.437E+00	1.940E+00
6.000E-09	1.196E+00	1.737E+00	2.340E+00
7.000E-09	1.396E+00	2.037E+00	2.740E+00
8.000E-09	1.596E+00	2.337E+00	3.140E+00
9.000E-09	1.796E+00	2.637E+00	3.540E+00
1.000E-08	1.996E+00	2.937E+00	3.940E+00
1.100E-08	2.000E+00	3.000E+00	4.000E+00
1.200E-08	2.000E+00	3.000E+00	4.000E+00
1.300E-08	2.000E+00	3.000E+00	4.000E+00
1.400E-08	2.000E+00	3.000E+00	4.000E+00
1.500E-08	2.000E+00	3.000E+00	4.000E+00
1.600E-08	2.000E+00	3.000E+00	4.000E+00
1.700E-08	2.000E+00	3.000E+00	4.000E+00
1.800E-08	2.000E+00	3.000E+00	4.000E+00
1.900E-08	2.000E+00	3.000E+00	4.000E+00
2.000E-08	2.000E+00	3.000E+00	4.000E+00

TIME	V(69)	V(73)	V(77)
0.000E+00	-4.392E-02	-2.021E-03	-6.022E-03
1.000E-09	-7.794E-02	3.154E-02	9.335E-02
2.000E-09	3.220E-01	1.315E-01	2.933E-01
3.000E-09	7.219E-01	2.315E-01	4.933E-01
4.000E-09	1.123E+00	3.316E-01	6.935E-01
5.000E-09	1.523E+00	4.316E-01	8.935E-01
6.000E-09	1.922E+00	5.315E-01	1.093E+00
7.000E-09	2.322E+00	6.315E-01	1.293E+00
8.000E-09	2.723E+00	7.316E-01	1.493E+00
9.000E-09	3.123E+00	8.316E-01	1.693E+00
1.000E-08	3.521E+00	9.315E-01	1.893E+00
1.100E-08	4.000E+00	1.000E+00	2.000E+00
1.200E-08	4.000E+00	1.000E+00	2.000E+00
1.300E-08	4.000E+00	1.000E+00	2.000E+00
1.400E-08	4.000E+00	1.000E+00	2.000E+00
1.500E-08	4.000E+00	1.000E+00	2.000E+00
1.600E-08	4.000E+00	1.000E+00	2.000E+00
1.700E-08	4.000E+00	1.000E+00	2.000E+00
1.800E-08	4.000E+00	1.000E+00	2.000E+00
1.900E-08	4.000E+00	1.000E+00	2.000E+00
2.000E-08	4.000E+00	1.000E+00	2.000E+00

JOB CONCLUDED

TIME	PAGE	DIRECT	BUFFERED
CPU	ELAPSED	FAULTS	I/O
0: 3:22.25	0: 3:27.79	380	8 3

TOTAL JOB TIME 202.24

## Appendix D: "Turbo C" Program for Generating SPICE Input File

```

/* This is a program to generate a SPICE input file which will solve*/
/* a system of linear equations using Neural Networks. */
#include <stdio.h>
main()
{
/****Declarations****/
  int cn,c_orig,vn,rn,vxn; /* nodes */
  int gn,i,j,k,l; /* indices */
  int neq; /* number of equations */
  int eq;
  float c[100][100]; /* coefficients of variables */
  float c_temp[100][100]; /* coefficients if negative */
  float b[100]; /* constants of equations */
  float a[100]; /* lagrange mult; addition of coeff of a variable */
  char filename[8];
  char redo[1];
  FILE *fp;

/****Input from keyboard****/
  clrscr ();
  printf("This program generates a SPICE input file to\n");
  printf("simulate the solution of a system of equations.\n");
  printf("\n");
  printf("Enter name of SPICE file to create.\n");
  scanf("%s", &filename);
  printf("\n");
  if ((fp=fopen(filename,"w"))==NULL)
  {
    printf("cannot open file\n");
    exit(1);
  }

  printf("Enter the number of equations in the system.\n");
  scanf("%d",&neq);
  printf("\n");
  for (i=1;i<=neq;i++)
  {
    printf("Equation %d\n",i);
    for (j=1;j<=neq;j++)
    {
      printf("Enter the coefficient of variable %d: ",j);
      scanf("%f",&c[i][j]);
    }
    printf("Enter the constant of equality: ");
    scanf("%f",&b[i]);
  }
/**** Compute the coefficients of the minimization function (A's) ****/
  for (i=1;i<=neq;i++)
  {
    a[i]=0;
  }
  for (j=1;j<=neq;j++)
  {
    for (i=1;i<=neq;i++)
    {
      a[j]=a[j] + c[i][j];
    }
  }
}

```

```

/****Calculations****/
/**** "-C" Eqns ****/
fprintf(fp,"*** -C Eqns ***\n");
cn=2;vn=5;
gn=1;
for (i=1;i<=neq;i++)
    {
        for (j=1;j<=neq;j++)
            {
                if (c[i][j]>0)
                    {
                        fprintf(fp,"G%d %d 0 %d 0 -%f\n",gn,cn,vn,c[i][j]);
                    }
                if (c[i][j]<0)
                    {
                        c_temp[i][j]=0 - c[i][j];
                        fprintf(fp,"G%d %d 0 %d 0 %f\n",gn,cn,vn,c_temp[i][j]);
                    }
                vn=vn+4;
                gn++;
            }
        cn=cn+2;
        vn=5;
    }
/**** "B" Eqns ****/
fprintf(fp,"*** B Eqns ***\n");
gn=neq*neq+1;cn=2;
for (i=1;i<=neq;i++)
    {
        fprintf(fp,"G%d %d 0 1 0 %f\n",gn,cn,b[i]);
        cn=cn+2;gn++;
    }
/**** "C" Eqns ****/
fprintf(fp,"*** C Eqns ***\n");
/* gn=neq*neq+neq +1; */
c_orig=cn;
vn=2;
for (i=1;i<=neq;i++)
    {
        for (j=1;j<=neq;j++)
            {
                if (c[i][j] != 0)
                    {
                        fprintf(fp,"G%d %d 0 %d 0 %f\n",gn,cn,vn,c[i][j]);
                    }
                cn=cn+2;
                gn++;
            }
        vn=vn+2;
        cn=c_orig;
    }
/**** "A" Eqns ****/
fprintf(fp,"*** A Eqns ***\n");
cn=c_orig;
for (i=1;i<=neq;i++)
    {
        fprintf(fp,"G%d %d 0 1 0 %f\n",gn,cn,a[i]);
        cn=cn+2;
        gn++;
    }

```

```

/** Other Info */
fprintf(fp,"*** Other Info *** \n");
fprintf(fp,"VS 1 0 FWL(0 0 10NS 1)\n");
fprintf(fp,"RS 1 0 1\n");
/* Diodes */
fprintf(fp,"** DIODES AND CAP'S ***\n");
cn=2;vn=5;
for (i=1;i<=2*neq;i++)
{
    fprintf(fp,"D%d %d 3 DIODE\n",i,cn);
    fprintf(fp,"C%d %d 0 1E-11\n",i,cn);
    cn=cn+2;
}
fprintf(fp,"VOFF 3 0 DC -.7\n");
cn=c_orig;
for (i=1;i<=neq;i++)
{
    vxn=2*neq +cn;
    rn=2*neq +i;
    fprintf(fp,"VX%d %d %d AC 0\n",i,cn,vxn);
    fprintf(fp,"R%d %d 0 1\n",i,vxn);
    fprintf(fp,"H%d %d 0 VX%d 1E11\n",i,vn,i);
    fprintf(fp,"R%d %d 0 1\n",rn,vn);
    vn=vn+4;
    cn=cn+2;
}
vn=5;
fprintf(fp,".MODEL DIODE D\n");
fprintf(fp,".WIDTH OUT=80\n");
fprintf(fp,".TRAN 1NS 30NS\n");
fprintf(fp,".PRINT TRAN ");
for (i=1;i<=neq;i++)
{
    fprintf(fp,"V(%d) ",vn);
    vn=vn+4;
}
fprintf(fp,"V(%d) \n",vn);
fprintf(fp,".END\n");
fclose(fp);
clrscr ();
printf("The SPICE file %s has been created. \n",filenm);
delay (3200);
}

```

Appendix E: SPICE File Generated by "Turbo C" Program for 2x2 Example

```
*** -C Eqns ***
G1 2 0 5 0 -1.000000
G2 2 0 9 0 -2.000000
G3 4 0 5 0 -4.000000
G4 4 0 9 0 -3.000000
*** B Eqns ***
G5 2 0 1 0 5.000000
G6 4 0 1 0 10.000000
*** C Eqns ***
G7 6 0 2 0 1.000000
G8 8 0 2 0 2.000000
G9 6 0 4 0 4.000000
G10 8 0 4 0 3.000000
*** A Eqns ***
G11 6 0 1 0 5.000000
G12 8 0 1 0 5.000000
*** Other Info ***
VS 1 0 PWL(0 0 10NS 1)
RS 1 0 1
** DIODES AND CAP'S ***
D1 2 3 DIODE
C1 2 0 1E-11
D2 4 3 DIODE
C2 4 0 1E-11
D3 6 3 DIODE
C3 6 0 1E-11
D4 8 3 DIODE
C4 8 0 1E-11
VOFF 3 0 DC -.7
VX1 6 10 AC 0
R1 10 0 1
H1 5 0 VX1 1E11
R5 5 0 1
VX2 8 12 AC 0
R2 12 0 1
H2 9 0 VX2 1E11
R6 9 0 1
.MODEL DIODE D
.WIDTH OUT=80
.TRAN 1NS 30NS
.PRINT TRAN V(5) V(9) V(13)
.END
```

## Appendix F: Mapping Equations to Circuit of Figure 7.1

In Figure 7.1 the main vertical lines represent voltages and the main horizontal lines represent a summation of currents. The first two horizontal lines correspond to the constraint equations in the two-variable example.  $f_1(V)$  is the first horizontal line.

$$f_1(V) = 1V_1 + 2V_2 - 5$$

The  $1V_1$  current is supplied by the VCCS which multiplies the  $V_1$  voltage by the gain of 1 and outputs the current. The  $2V_2$  current is supplied by the VCCS which multiplies the  $V_2$  voltage by the gain of 2 and outputs the current. The  $-5$  current results from the connection of the VCCS, with a gain of  $-5$ , to the  $1V_1$  vertical line. The diode function is shown in Figure 7.2 and  $U(f_1)$  is the voltage across the diode.  $f_2(V)$  is represented in the same way.

To illustrate the mapping procedure for the second pair of horizontal lines, consider the  $-\frac{dE}{dV_1}$  equation for Figure 7.1:

$$-\frac{dE}{dV_1} = -5 - U(f_1) \cdot 1 - U(f_2) \cdot 4 - \frac{1}{R}g^{-1}(V_1)$$

Recall

$$-\frac{dE}{dV_1} = C_1 g^{-1}(V_1) \frac{dV_1}{dt}$$

Solving for  $V_1$

$$V_1 = \frac{1}{C_1} \int g(V_1) \cdot \left(-\frac{dE}{dV}\right)$$

or

$$V_1 = \frac{1}{C_1} \int (g(V_1) \cdot (-5 - U(f_1) \cdot 1 - U(f_2) \cdot 4) - \frac{1}{R})$$

The  $-\frac{dE}{dV_1}$  equation is the summation of currents and thus is represented as the third horizontal line. The  $-5$  current is supplied by the VCCS connected to the  $1$  volt vertical line and multiplies by the gain of  $-5$ .  $-U(f_1) \cdot 1$  is the current supplied by the VCCS connected to the  $U(f_1)$  vertical voltage line and multiplies by a gain of  $-1$ . Likewise for  $-U(f_2) \cdot 4$ . The function  $g(V_1)$  is performed by the diode. Finally,  $\frac{1}{R}$  is the current through the resistor. To find the voltage,  $V_1$ , the summation of currents must be integrated (function of the capacitor) and multiplied by a constant (function of the amplifier).  $V_2$  is represented in the same manner.