A NEURAL NETWORK APPROACH TO HIGH PERFORMANCE ANALOG CIRCUIT DESIGN

Terry L. Sculley and Martin A. Brooke
Georgia Institute of Technology, School of Electrical Engineering, Atlanta, GA 30332
Telephone: (404)894-3304

Abstract: **In order to gain new insight into the design of high precision, high speed analog circuits, several possible network implementations of an A/D converter are presented. These networks are marked by programmability and parallelism, which can be used to maintain circuit precision without the use of feedback. This removes design constraints on closed loop stability and may lead to faster circuit performance. On-chip training or calibration is likely necessary, but can be done in an "off line" mode and thus may not hinder circuit speed significantly.**

The recent surge of interest in neural networks has led to analog circuit implementations of a variety of network architectures, most of which have been used to perform functions not typically done with analog circuits, such as combinatorial optimization or associative memory recall. However, a neural network can also be configured to perform functions more common to analog circuit design, such as linear amplification or A/D or D/A conversion. A careful look at the differences between a neural network approach and a conventional approach to an analog design can yield some useful ideas. The high levels of programmability and parallelism in neural networks give them the potential for obtaining high speed, high precision circuit operation. The programmability allows the circuit to be "trimmed" electronically after fabrication to overcome device mismatch and process variations, while the parallelism can increase the circuit speed. Although parallel circuit architectures are utilized extensively in conventional analog design, programmable circuit topologies have only been used to a limited extent.

An investigation of possible neural network implementations of a common analog building block, an A/D converter, yields a variety of network topologies. One possibility is a fully interconnected Hopfield network, as proposed by John Hopfield [1] and others. However, the large amounts of feedback present in these networks makes stability of the circuit a major concern and limits the speed of conversion. An obvious alternative is a multilayer perceptron network, which has a completely feedforward interconnect structure. In addition, perceptron networks are well suited for mapping input vectors to target output vectors or classifying inputs into regions in the input space. This corresponds exactly to the A/D conversion problem, in which a one-dimensional input is classified into 2^N possible regions, for an N bit converter, defined in the input space. Several possible implementations using a multilayer perceptron are discussed below.

Two Layer Perceptron A/D Converter

A two bit A/D converter using a two layer perceptron network with hard limiting activations amplifiers is shown in figure 1. The second layer activation amplifiers can be eliminated, simplifying the network to that shown in figure 2. This architecture is very similar to that of a flash A/D converter, shown in figure 3. Here, a resistor string is used to weight the reference voltage, and a digital decoding network is used in place of a linear network to decode the amplifier outputs into the digital output code. Extending the two layer perceptron to N bits, as shown in figure 4, is easily done by using more amplifiers and generalizing the interconnection matrices.

An analog circuit implementation of this two layer converter should be fast and accurate. Making each weight circuit with fine adjustability allows the gain to be adjusted precisely to account for any nonlinearity or offset in the corresponding summers and activation amplifiers. Since the input to each weight is either a 1 or 0, the only possible outputs are a precise analog output or zero. This is easily implemented with a D/A converter and a switch, allowing digital adjustment of the perceived weight value. Another requirement of this converter is an activation amplifier with gain on the order of 2^N. This, however, is easily attainable using conventional design techniques for N up to 20 or more. Unfortunately, like the conventional flash converter, this two layer converter can only be implemented in integrated circuitry for a low number of bits, since the number of weight circuits and activation amplifiers increases exponentially with the number of bits and quickly fills the available chip area.

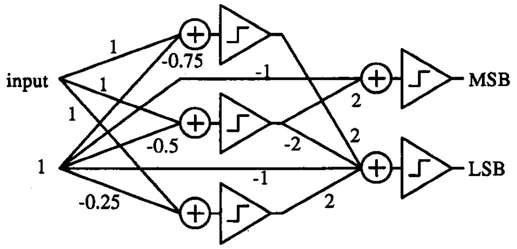


Figure 1. A Two Layer Perceptron A/D Converter

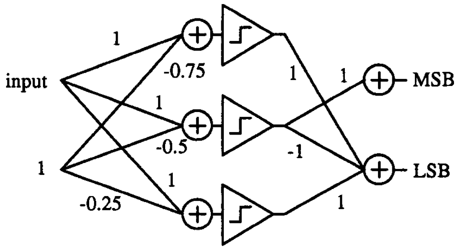


Figure 2. A Simplified Two Layer Perceptron A/D Converter

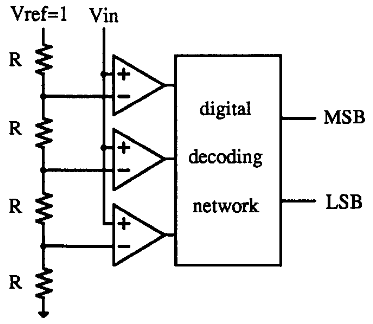


Figure 3. A Two Bit Flash Converter

89CH2785-4/90/0000-0497$01.00 © 1990 IEEE

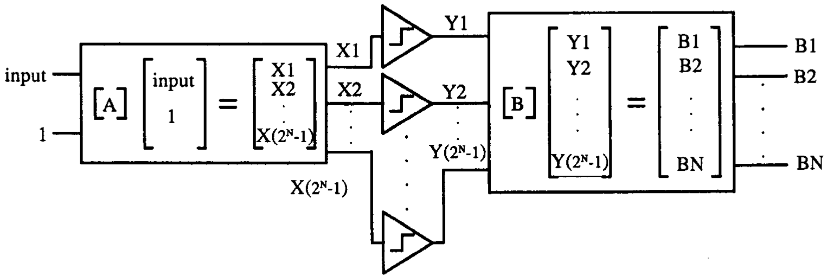


Figure 4. An N bit Two Layer Perceptron A/D Converter

N Layer Perceptron A/D Converter

A second possible perceptron A/D converter uses N layers for an N bit converter and one activation amplifier per layer. The diagram for this network is shown in figure 5, and a simplified network, which does not strictly adhere to a perceptron topology but has fewer weight circuits, is shown in figure 6. This design still has the advantage that the weight circuits only have two possible inputs, a constant or zero, so that implementation is simple. It also allows a much higher bit converter to be built in the same chip area than the two layer converter. However, now the summers used in the network must be linear to the desired number of bits. In addition, the activation amplifiers still require gain on the order of 2^N, as in the two layer case.

The N layer converter compares to two conventional A/D designs - a successive approximation converter, shown in figure 7, and a pipeline converter, shown in figure 8. The successive approximation converter forms the digital output one bit at a time, from the MSB to the LSB, and uses previously determined bits to form an approximate analog signal to compare the input to at each layer. The conversion speed is reduced as the number of bits increases, since each conversion must be completed before another can begin. The serial nature of the design, however, allows only one comparator and D/A converter to be used and the computation clocked around it in cycles. One type of successive approximation converter, the self calibrated converter, is in essence an implementation of figure 6 with on-chip training of the weights and has been used in high resolution converters. The pipeline converter has a very similar topology except that some of the summers and multipliers have been shifted from one side of each comparator to the other side, but each comparator still evaluates the same signal difference as the corresponding comparator in the successive approximation converter. The conversion rate of the pipeline converter stays approximately constant as the number of bits increases, because each layer can be operating on a different input sample. However, the actual delay from sampling to digital output does increase with the number of bits. Generally, pipeline converters are not built to a high number of bits due to the need for highly linear weight circuits, which are difficult to implement at high speed.

M+1 Layer Perceptron A/D Converter

A third possible perceptron converter is a compromise between the two layer and the N layer versions. An M+1 layer converter can be built, where 2<M<N, and each layer after the first may convert any number of bits. An example network for converting 6 bits using 4 layers is shown in figure 9. This variety of converter has the same performance restrictions on its various components that the N layer

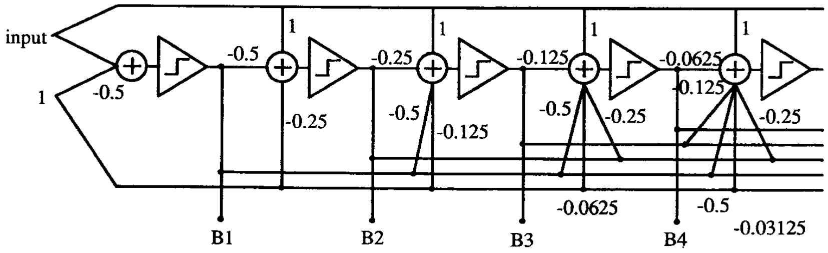


Figure 5. An N Layer Perceptron A/D Converter

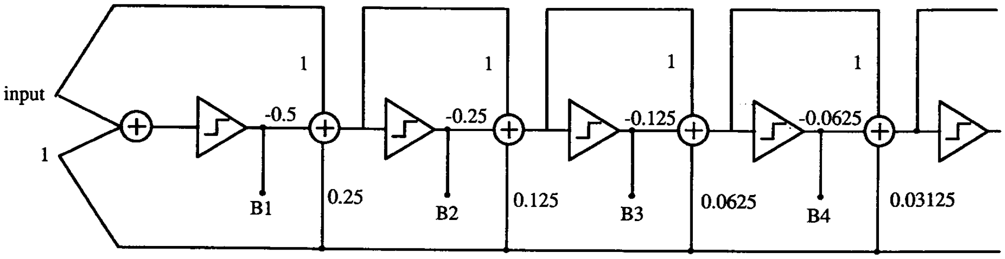


Figure 6. A Simplified N Layer Perceptron A/D Converter

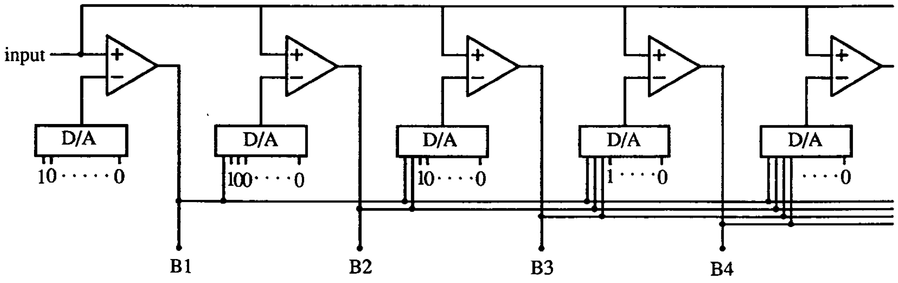


Figure 7. A Successive Approximation A/D Converter

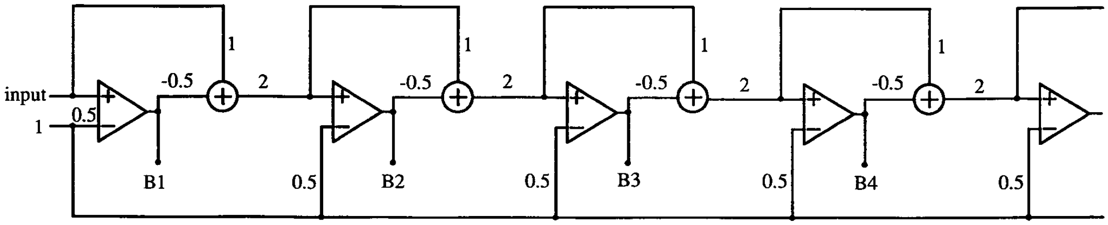


Figure 8. A Pipeline A/D Converter

converter has, but it trades an increased number of weights and activation amplifiers for fewer layers and thus higher speed.

The corresponding conventional A/D converter design in this case is the M step flash converter, which is shown in figure 10 for the case of M=3 and 2 bits/step. Most implementations of these converters are with M=2 and N=8 to 12 and use little or no correction, but they are usually high speed. A neural converter of this type can potentially achieve higher resolution than the conventional design and still maintain high speed. However, as the number of bits converted per layer increases, restrictions on chip area will become a limiting factor, as in the case of the two layer converter.

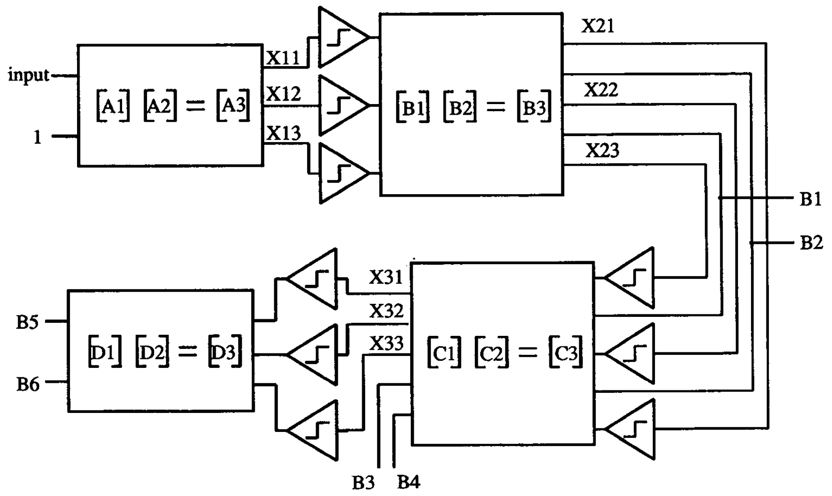
Conclusions

Neural A/D converter configurations are useful in giving a new perspective on how a high resolution A/D converter can be trained or calibrated for high accuracy. Conventional analog circuit design uses high gain blocks with feedback to trade gain for accuracy. This requires closed loop stability and generally results in lowered circuit speeds due to compensation techniques. The neural approach, however, utilizes adjustable circuit components to achieve precision and only requires feedback to train the circuit - a process normally done in an "off line" state and not during normal circuit operation. The next issue then becomes whether feedback circuits are necessary to build the network components. The answer depends on several factors, including the network topology, but slight modifications to some of the networks discussed in this paper yield topologies which ideally can achieve extremely high resolution without feedback circuitry. The limiting factor on the resolution of these converters then becomes chip area and electrical noise in the circuit.

One major problem associated with the use of circuit adjustment to obtain high precision analog circuit performance is the variation of circuit characteristics over time due to noise, thermal variations, and aging effects. In order to prevent these effects from destroying the circuit performance, the network must be retrained periodically, with the frequency of retraining dependent on how fast the circuit characteristics change and what resolution must be maintained. This means on-chip training is probably necessary. Training these networks is not as much of a problem as it can be for a general perceptron network, though, and appears entirely feasible, as demonstrated by the self calibrated successive approximation A/D converter.

References

[1] J. J. Hopfield and D. W. Tank, "Simple 'Neural'Optimization Networks: An A/D Converter, Signal Decision Circuit, and a Linear Programming Circuit," *IEEE Trans. Circuits Syst.*, vol. CAS-33, pp. 533-541, May 1988.



NOTE: The input and a 1 are fed to each layer, but are not shown in the diagram after the first layer

$$A1 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 1 & -3/4 \\ 1 & -2/4 \\ 1 & -1/4 \end{bmatrix} \quad A2 = \begin{bmatrix} \text{input} \\ 1 \end{bmatrix} \quad A3 = \begin{bmatrix} \text{input} \\ 1 \\ X11 \\ X12 \\ X13 \end{bmatrix} \quad B1 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & -1 & 1 \\ 1 & -3/16 & -1/4 & -1/4 & -1/4 \\ 1 & -2/16 & -1/4 & -1/4 & -1/4 \\ 1 & -1/16 & -1/4 & -1/4 & -1/4 \end{bmatrix} \quad B2 = \begin{bmatrix} \text{input} \\ 1 \\ Y11 \\ Y12 \\ Y13 \end{bmatrix}$$

$$B3 = \begin{bmatrix} \text{input} \\ 1 \\ B1 \\ B2 \\ X21 \\ X22 \\ X23 \end{bmatrix} \quad C1 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & -1 & 1 \\ 1 & -3/64 & -1/2 & -1/4 & -1/16 & -1/16 & -1/16 \\ 1 & -2/64 & -1/2 & -1/4 & -1/16 & -1/16 & -1/16 \\ 1 & -1/64 & -1/2 & -1/4 & -1/16 & -1/16 & -1/16 \end{bmatrix} \quad C2 = \begin{bmatrix} \text{input} \\ 1 \\ B1 \\ B2 \\ Y21 \\ Y22 \\ Y23 \end{bmatrix} \quad C3 = \begin{bmatrix} B3 \\ B4 \\ X31 \\ X32 \\ X33 \end{bmatrix}$$

$$D1 = \begin{bmatrix} 0 & 1 & 0 \\ 1 & -1 & 1 \end{bmatrix} \quad D2 = \begin{bmatrix} Y31 \\ Y32 \\ Y33 \end{bmatrix} \quad D3 = \begin{bmatrix} B4 \\ B5 \end{bmatrix}$$

with Yij being the output of an activation amplifier with input Xij

Figure 9. A Four Layer Perceptron A/D Converter

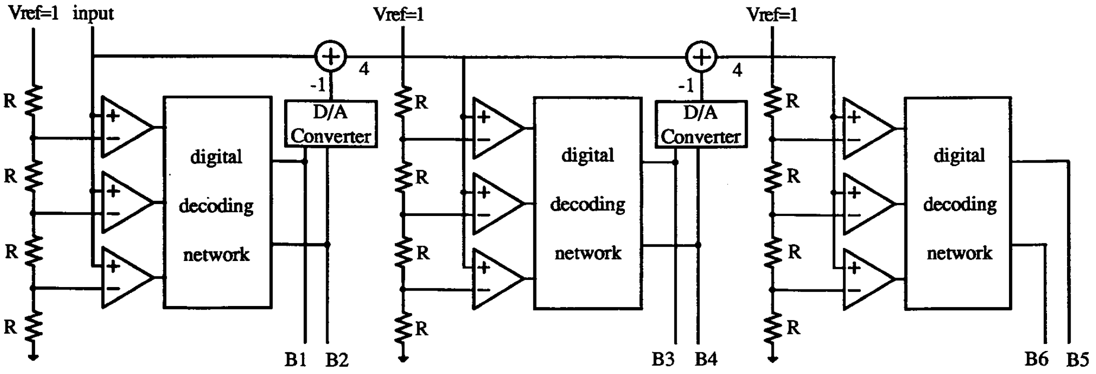


Figure 10. A Three Step Flash A/D Converter