

OBTAINING HIGH PRECISION OPERATION FROM NONIDEAL NEURAL NETWORKS

Terry L. Sculley and Martin A. Brooke

School of Electrical Engineering, Georgia Institute of Technology, Atlanta, Georgia 30332

Telephone: (404) 894-3304

Abstract: This paper examines several potential neural architectures for an A/D converter and discusses the level of nonidealities that can be present in the network components and still achieve high precision operation through training. Behavioral level simulations on sample converter networks with nonidealities modeled revealed a strong interrelationship between the network architectures and their tolerance to nonidealities.

When designing a neural network structure, it is possible to develop a network that, ideally, can be trained to perform high precision computation, such as A/D conversion. When implemented in electronic circuitry, however, nonidealities may occur in the operation of the various network components. These nonidealities can affect network performance significantly and may make a network impossible to train successfully, depending on the overall network architecture. Several possible neural or neural-like topologies for A/D converters are explored below, and the effects of these nonidealities on their precision of conversion are investigated.

Probably the first neural A/D converter that comes to mind is a fully interconnected Hopfield network [1]. However, the large amounts of feedback present in this network make stability of a circuit implementation a major concern, and compensation necessary to ensure stability may limit the circuit speed. An alternative structure uses a multilayer perceptron network, which has a completely feedforward interconnect structure. Perceptron networks have been demonstrated useful for classifying inputs into regions in the input space or for mapping input vectors to target output vectors. The problem of A/D conversion can be viewed as classifying a one-dimensional input into 2^N possible regions, for an N bit converter, defined in the input space. Several different A/D converter architectures are possible using a multilayer perceptron [2], some of which are discussed below.

A two layer perceptron implementation of a two bit A/D converter using hard limiting activation amplifiers is shown in figure 1. This architecture closely resembles that of

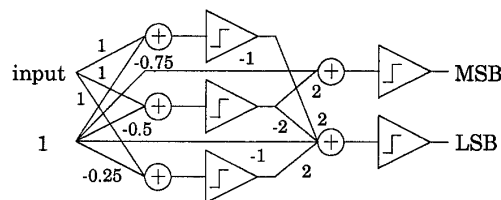


Figure 1. A two layer perceptron A/D converter

a conventional flash A/D converter, in which the output of the first stage amplifiers is a thermometer code (all outputs below a certain amplifier are high, while those above it are low). The major differences in the flash converter are the use of a nonprogrammable resistor string to create the weighted reference inputs and the use of digital logic to decode the thermometer code output of the first stage into the final digital output bits. Although the two layer converter shares the flash converter's advantage of potential high speed operation, it also shares its disadvantage of requiring approximately 2^N first stage amplifiers. Due to this exponential relationship between the number of bits and the number of amplifiers, the resolution of a two layer converter implemented in circuitry is limited by the large size circuit needed for a high number of bits.

A second possible perceptron A/D converter is shown in figure 2 using one activation amplifier per layer in four layers for a four bit converter. A simplified version of this network requiring fewer weights is shown in figure 3. This converter architecture is very similar to a conventional successive approximation or pipeline converter and shares many of their same characteristics. Like a pipeline converter, the number of amplifiers in the architecture of figure 3 increases linearly with the number of bits, yielding an N layer converter. However, the speed of this simplified N layer converter will be inversely proportional to the number of bits,

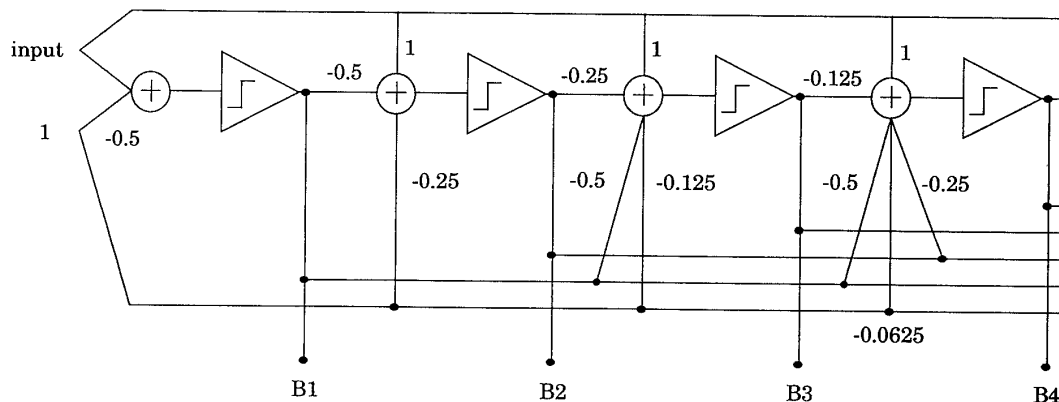


Figure 2. A four layer perceptron A/D converter

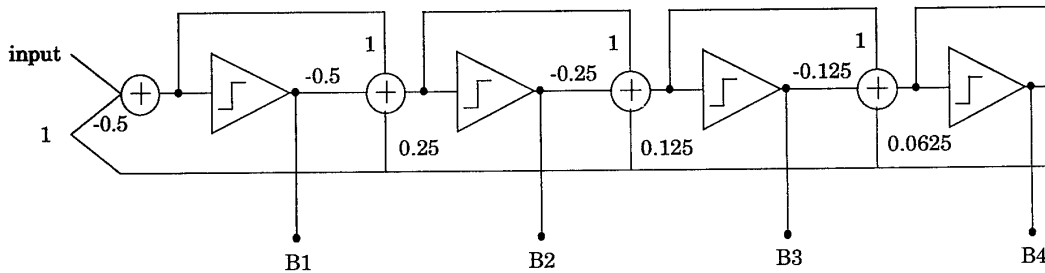


Figure 3. A simplified four layer A/D converter

giving a tradeoff between circuit size and speed when compared to the two layer architecture.

Many other architectures for perceptron A/D converters are possible using between two and N layers and allowing a varying tradeoff between speed and circuit size. The two converters which have been presented thus far represent the extremes in each direction, with the two layer converter having large size and very high speed and the N layer converter having small size and much lower speed.

Although both of these networks in figures 1 and 3 will ideally implement A/D converters, their actual resolution when implemented in electronic circuitry will depend on both the particular network topology and the precision with which the individual network components can be built. In micro-electronic circuitry, the hard limiting activation amplifiers can be implemented using comparators, which typically have very high gain but may have a significant input offset signal. The summers can be difficult to build with high precision, as are weighting amplifiers with a highly linear characteristic and an adjustable gain. However, these two network topologies avoid the need for highly linear amplifiers because no fully analog signal passes through any weight circuit that is not unity gain. All weight circuits that have non-unity gain have only one or two possible input values (with the second value being zero). This allows the weight circuits to be implemented by letting the weight input control a switch that connects the weight output to a preset value corresponding to the weight input times the weight value. A digitally adjustable weight circuit can be implemented in this fashion using a D/A converter and a switch controlled by the weight input.

Another extra component necessary for an A/D converter is a sample-and-hold, which is placed at the input to the converter. Its output is then fed to the converter network. Circuit implementations of sample-and-holds often suffer from offsets, gain errors, and nonlinearities, which can affect the network precision.

In order to determine the effects of these nonideal network components on converter precision, the two converter architectures of figures 1 and 3 were investigated analytically and then behaviorally modeled and simulated in a computer program. The analytical results indicated that the two layer converter is fairly tolerant to offsets, gain errors, and nonlinearities in the sample-and-hold, summers, D/A converters, and comparators. The main requirements are that the D/A converters have adequately fine adjustability (i.e., no large gaps in the analog output space) and that the comparators have high gain. Some generous limits exist on the amount of nonideality tolerable in the second layer components, but these were disregarded since a practical implementation of this converter would likely use a digital

decoding network in place of the second layer.

The analysis of the N layer converter architecture in figure 3 revealed a much lower tolerance to nonidealities than the two layer architecture. Here, the sample-and-hold and summers must be linear to the total converter precision, although offsets and gain errors are acceptable. Fine adjustability in the D/A converters and high gain comparators are still necessary, as in the previous case.

In order to verify the analytical results, the two converter architectures were modeled behaviorally and simulated in a computer program with nonidealities selectively included. The sample-and-hold was modeled by a nonlinear transfer function given by

$$S/H \text{ out} = sh_err0 + (sh_err1 \cdot input) + (sh_err2 \cdot input^3)$$

Of the weight circuits, those that were unity gain were assumed ideal, and the others were modeled as a D/A converter with a switch. A 20 bit D/A converter was modeled with differential nonlinearity of 20 bits but a much larger integral nonlinearity. A plot of the D/A transfer characteristic is shown in figure 4. All summers except for those in the second layer of the first converter were modeled by the following transfer function:

$$sum \text{ out} = sum_err0 + (sum_err1 \cdot input_1) + (sum_err2 \cdot input_2) + (sum_err3 \cdot input_1^3) + (sum_err4 \cdot input_2^3)$$

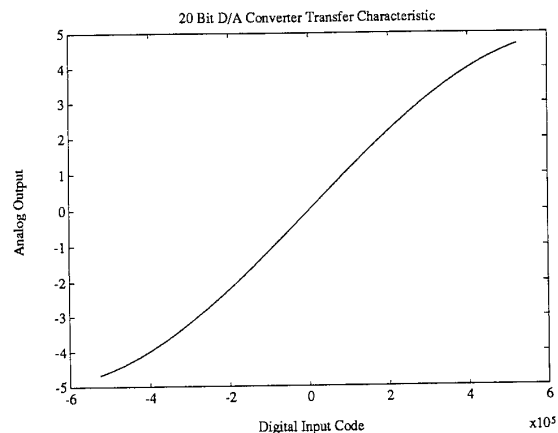


Figure 4. 20 bit D/A converter transfer characteristic

The activation amplifiers were modeled as comparators with their activation inputs connected to the positive terminals and the negative terminals grounded. Offset and common-mode error were modeled by forming an intermediate signal x , where

$$x = vin_plus - vin_minus + comp_err0 + (comp_err1 \cdot 0.5 \cdot (vin_plus + vin_minus))$$

This value was then passed through the nonlinearity shown in figure 5 (but with higher gain) to get the activation output.

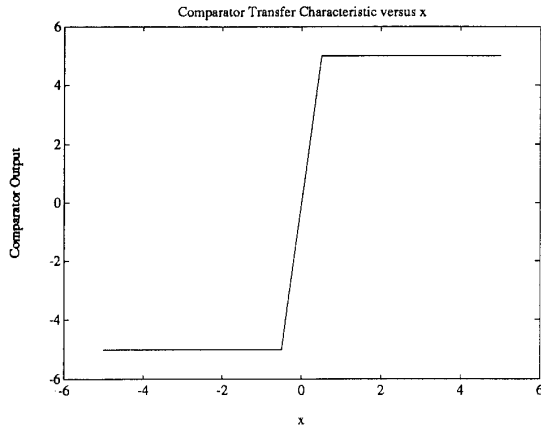


Figure 5. Comparator transfer characteristic versus x

In the case of the two layer converter, a four bit converter network was simulated and trained to yield greater than 16 bits precision. An actual circuit with these characteristics would likely be used in the first stage of a four bit/stage, 16 bit pipeline converter. When simulating the converter with full nonlinearity included, the D/A converters were first loaded with the ideal digital values that would result in proper operation if all network components were ideal. The A/D converter network was then simulated with the nonidealities included. A plot of the resulting differential nonlinearity is shown in figure 6, revealing that the converter is only about three bits accurate at this point. Next, the A/D converter was trained by applying analog signals at the input correspond-

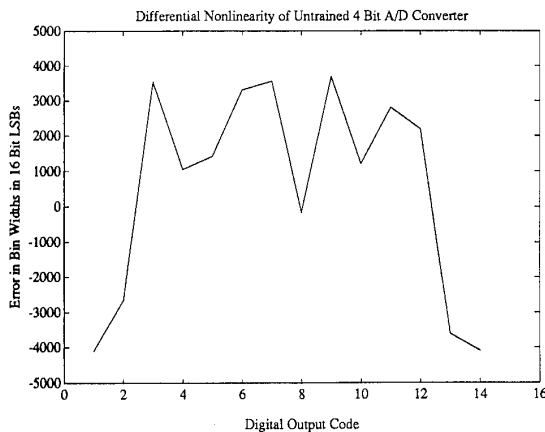


Figure 6. Differential nonlinearity of untrained four bit A/D converter

ing to the desired transition points between levels and performing a binary search on the D/A converters' inputs to set the transition points. A plot of the new differential nonlinearity is shown in figure 7. As can be seen, the converter is now approximately 17 bits accurate. Figure 8 shows a plot of the integral nonlinearity of the trained converter, which is also 17 bits accurate.

The N layer converter was simulated using $N=16$ for three different cases: (1) full nonideality in all components, (2) only offset and gain errors included, and (3) only offset errors included. The same nonlinear D/A converter model was used in all three cases. The converter was trained by applying 32 particular inputs to the converter and performing a binary search on each of the 32 D/A converter inputs. In the first case, the converter exhibited approximately four bits of both differential and integral nonlinearity after training. For both the second and third cases, however, the simulations resulted in a differential nonlinearity of 17 bits and an integral nonlinearity of 16 bits. These simulations are consistent with the analytical results. Figures 9 and 10 show plots of the integral nonlinearity for cases (1) and (2).

Although the N layer converter architecture in figure 3 is not tolerant to nonlinearities in the summers and sample-and-hold, the architecture can be modified to make it tolerant

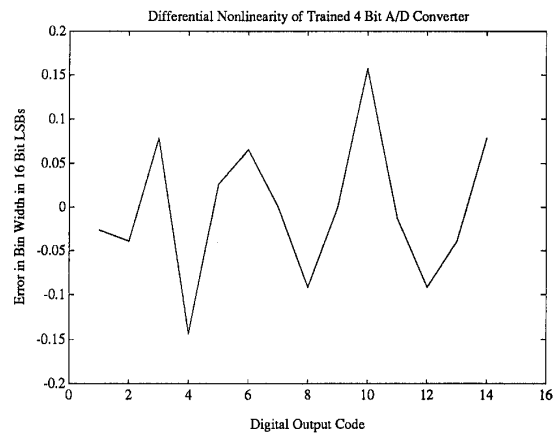


Figure 7. Differential nonlinearity of trained four bit A/D converter

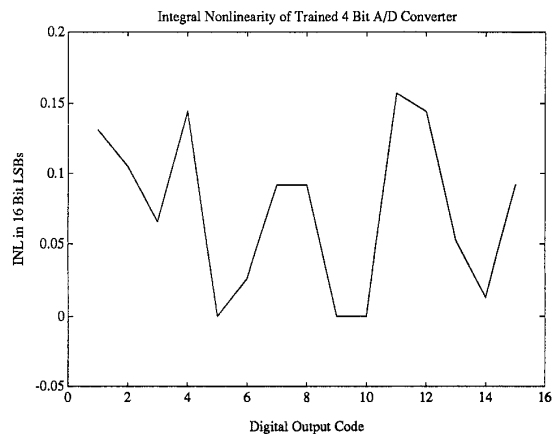


Figure 8. Integral nonlinearity of trained four bit A/D converter

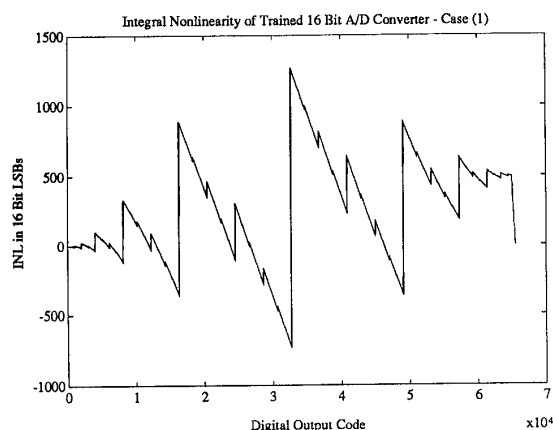


Figure 9. Integral nonlinearity of trained 16 bit A/D converter - Case (1)

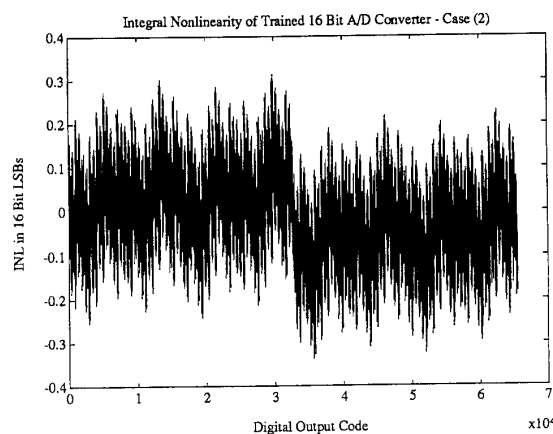


Figure 10. Integral nonlinearity of trained 16 bit A/D converter - Case (2)

by using the previously determined bits to select among different digital inputs to apply to the D/A converters. This modified architecture, shown in figure 11, now has a tolerance to nonlinearities similar to the two layer network but uses fewer network components. Simulations of a 16 bit converter with this modified architecture and including full nonidealities yielded integral and differential nonlinearities of 17 bits.

Conclusions

Of the A/D architectures presented here, both the two layer and the modified N layer converters are tolerant of nonlinearities in the network components. This tolerance is achieved in the two layer converter through the use of parallel neuron circuits and programmability. The same tolerance is achieved in the modified N layer converter through the use of parallel information (i.e., weight values) and fewer neuron circuits, although this compromises other aspects of performance, such as speed and fault tolerance. The N layer converter shown in figure 3, however, relies on the exact characteristics of the network components for obtaining high precision operation. This enables fewer neurons and fewer weights to be necessary in an ideal network, but it also makes the network incapable of being trained

successfully if component characteristics vary significantly. These results indicate that the particular network architecture used for a specific application will affect whether the network can be trained successfully if component performance is nonideal. However, this highlights the fact that tradeoffs are possible between various aspects of the network, such as tolerance to component variations, speed of operation, physical size of network, and tolerance to component failure. These tradeoffs can be exploited and may allow more complex or more precise networks to be implemented than would be possible otherwise.

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.
- [2] T. L. Sculley and M. A. Brooke, "A Neural Network Approach to High Performance Analog Circuit Design," *Proc. 1989 IEEE Midwest Symp. Circuits Syst.*, Aug. 14-16, 1989.

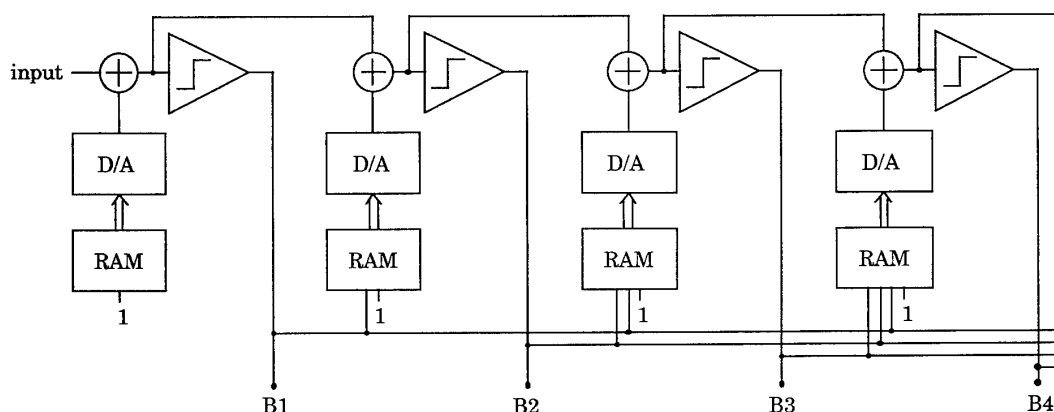


Figure 11. A modified four layer four bit A/D converter