CELLULAR NEURAL NETWORK A/D CONVERSION

by

Lin Yang and Leon O. Chua

Memorandum No. UCB/ERL M91/20

6 March 1991

CELLULAR NEURAL NETWORK
A/D CONVERSION

by

Lin Yang and Leon O. Chua

Memorandum No. UCB/ERL M91/20

6 March 1991

ELECTRONICS RESEARCH LABORATORY

College of Engineering
University of California, Berkeley
94720

CELLULAR NEURAL NETWORK
A/D CONVERSION

by

Lin Yang and Leon O. Chua

Memorandum No. UCB/ERL M91/20

6 March 1991

ELECTRONICS RESEARCH LABORATORY

College of Engineering
University of California, Berkeley
94720

Abstract

In this paper, we develop a design theory of A/D conversion using a cellular neural network (CNN) architecture [3, 2]. We consider the A/D conversion issue as an optimization problem, and develop a systematic method to design a CNN A/D converter. In spite of its *parallel* architecture, we observed an intrinsic *serial* computing phenomenon during the analysis of the dynamics of the CNN A/D converter. By studying the steady state behavior of the circuit, we developed an algorithm to design an N-bit CNN A/D converter for any required conversion error. We summarize our results into two theorems, thereby providing the theoretical foundation for CNN A/D conversion.

This work was supported in part by the U.S. Office of Naval Research under Grant no. N00014-89-J-1402, and the National Science Foundation under Grant no. MIP 8912639.

1 Introduction

In the paper [3], we introduced an analog computing architecture called *cellular neural networks*(CNN). In that paper, the nearest neighbor interconnection features of cellular neural networks have been emphasized. However, as a general computing architecture, cellular neural networks can be also used to model other neural networks, such as that due to Hopfield. Since the nonlinearity of the sigmoid function in a cellular neural network is piecewise-linear, it is much easier to analyze the dynamics of these networks.

An analog-to-digital (A/D) converter of Hopfield neural network has been proposed in [5, 8]. Nevertheless, because of the lack of a rigorous analysis, the Hopfield's A/D converter is very difficult to implement. Also, some implementation difficulties for Hopfield's A/D converter have been pointed out recently [7]. In this paper, we design an A/D converter circuit using a cellular neural network architecture and analyze its dynamical behavior under some conditions. In spite of the *parallel* computing architecture, the *serial* computing mechanism has been observed during the analysis of the dynamics of our CNN A/D converter. Furthermore, we explore this *serial* computing mechanism to develop an optimization method so that we can design a CNN A/D converter for any given conversion error.

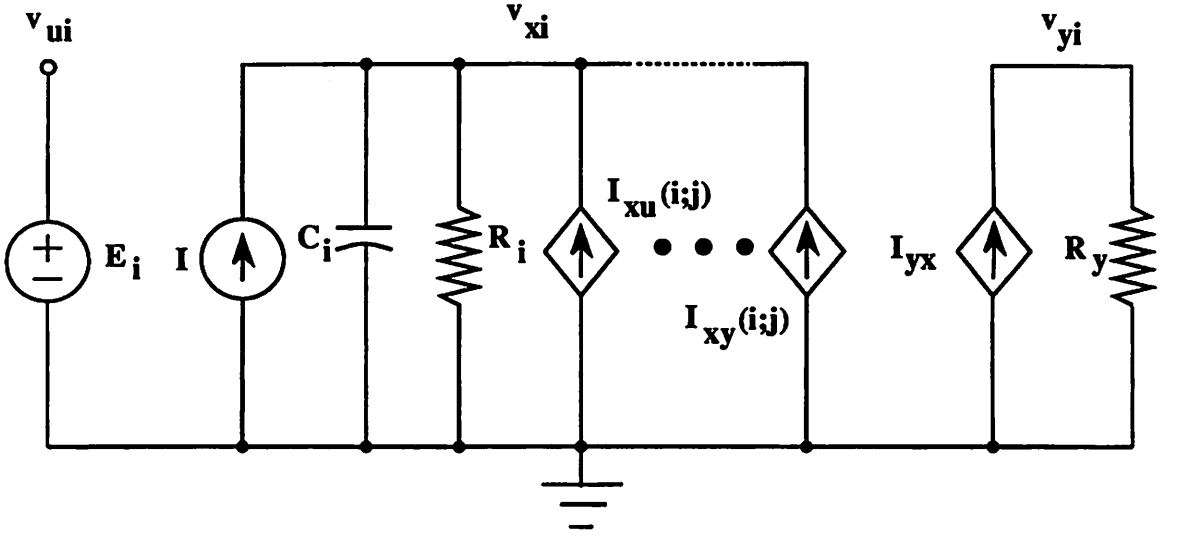
The paper is organized as follows. In *Section* 2, we specify the CNN architecture and give its related theoretical results for designing the A/D converter. In *Section* 3, we describe a digital representation system which is more suitable for cellular neural networks. In *Section* 4, without loss of generality, we use a 4-bit CNN A/D converter as an example to illustrate our design procedure. In *Section* 5, we analyze the dynamics of a *relaxed* CNN A/D converter by using linear system theory and computer simulations, and demonstrate some computer simulation results. In *Section* 6, we study the steady states issues. In *Section* 7, we provide an optimization method for designing an *unrelaxed* CNN A/D converter with any prescribed A/D conversion error. Finally, in *Section* 8, we summarize the results presented in this paper.

2 The CNN Architecture for A/D Conversion

As discussed in [3], we can design and characterize a cellular neural network by specifying its cell circuit and the corresponding cell circuit equations. In this section, we will specify the circuit parameters presented in order to design a CNN A/D converter.

An N-bit A/D converter can be considered simply as N cells, namely; an $N \times 1$ cellular neural network. For this one-dimensional cellular neural network, let us use the simpler notation $C(i)$ to denote the cell "i".

The cell circuit of the one-dimensional CNN is shown in Figure 1. There is a small difference between the parameter values in this cell circuit and the one



$$I_{xu}(i;j) = B(i;j)\ v_{uj}\ ; \qquad I_{xy}(i;j) = A(i;j)\ v_{yj}\ ;$$

$$I_{yx} = \frac{1}{2R_y}\ (|\ v_{xi} + 1\ | - |\ v_{xi} - 1\ |)$$

Figure 1: Cell circuit of CNN A/D converters

in Figure 3 of [3]; namely, the values of the cell capacitor C_i and the resistor R_i in the CNN cell circuit of Figure 1 depend on the ith cell $C(i)$, indexed by the subscript i, but those in [3] do not.

The cell circuit equations are as follows,

$$C_i \frac{dv_{xi}(t)}{dt} = \frac{-1}{R_i} v_{xi}(t) + \sum_{j=0}^{N-1} A_{ij} v_{yj}(t) + b_i v_u \tag{1}$$

$$v_{yi}(t) = 0.5\,(\mid v_{xi}(t) + v_c \mid - \mid v_{xi}(t) - v_c \mid) \tag{2}$$

for $i, j = 0, 1, 2, \ldots$ *and* $N-1$, where $A_{ij} = A_{ji}$ is the interactive parameter, b_i is the control parameter, v_{xi} is the state voltage of cell $C(i)$, and v_{yi} is the output voltage of cell $C(i)$ which corresponds to the output value of bit i of an N-bit A/D converter. Comparing the above equations with (2a) and (2b) of [3], we see that the circuit parameters defined in [3] now assume the values $I = 0$ and $v_{ui} = v_u$ which is the analog input voltage. Notice too that in the output equation (2) we allow an arbitrary cutoff voltage v_c, instead of $v_c = 1.0$, as in (2b) of [3]. This modifications is useful for scaling the physical parameters in the implementation of the circuit. Nevertheless, we alway use $v_c = 1$ in our theoretical development.

We can easily prove that the main results presented in [3] still hold for our present CNN, by using the same methods presented in [3]. For convenience, let us rewrite some of these results without proofs.

The Lyapunov function, $E(t)$, for our CNN A/D converter is given by

$$\begin{aligned} E(t) = \; & - \frac{1}{2} \sum_{(i)} \sum_{j=0}^{N-1} A_{ij} v_{yi}(t) v_{yj}(t) + \sum_{i=0}^{N-1} \frac{1}{2R_i} v_{yi}^2(t) \\ & - \sum_{i=0}^{N-1} b_i v_{yi}(t) v_u. \end{aligned} \tag{3}$$

If the circuit parameters satisfy

$$A_{ii} > \frac{1}{R_i}, \tag{4}$$

then each cell of our cellular neural network must settle to a *stable equilibrium point* after the *transient* has decayed to zero. Moreover, the magnitude of all *stable equilibrium points* is greater than 1. In other words, when a CNN settle to its steady state, the following properties are true:

$$\frac{d\, v_{xi}(t)}{d\, t} = 0, \tag{5}$$

$$\mid v_{xi}(t) \mid \geq 1 \tag{6}$$

$$v_{yi}(t) = \pm 1. \tag{7}$$

N_{out}	$v_{ys3}(2^3)$	$v_{ys2}(2^2)$	$v_{ys1}(2^1)$	$v_{ys0}(2^0)$
-15	-1	-1	-1	-1
-13	-1	-1	-1	1
-11	-1	-1	1	-1
-9	-1	-1	1	1
-7	-1	1	-1	-1
-5	-1	1	-1	1
-3	-1	1	1	-1
-1	-1	1	1	1
1	1	-1	-1	-1
3	1	-1	-1	1
5	1	-1	1	-1
7	1	-1	1	1
9	1	1	-1	-1
11	1	1	-1	1
13	1	1	1	-1
15	1	1	1	1

Table 1: A digital representation system

3 The Digital Representation System for CNN A/D Conversion

From the previous section, we know that for a given analog value v_u, the output voltage v_{yi} of each cell C_i of a CNN will have a binary value; namely, ± 1 according to (7). This implies that we should use -1 and 1 as our bit values in the digital representation. For an N-bit CNN A/D converter, a digital number represented by the circuit steady outputs can be expressed as

$$N_{out} = \sum_{i=0}^{N-1} 2^i v_{ysi} \tag{8}$$

where $v_{ysi} = \pm 1$. As an example, for a 4-bit CNN A/D converter, N_{out} represents 16 distinct numbers, which are all odd numbers, and are listed in *Table* 1. The *ideal* input-output analog-to-digital transfer function is shown in Figure 2(a). Here, the adjective *ideal* means that the maximum A/D conversion error is minimized. If we define

$$\epsilon_{ideal} \equiv \max_{v_u} \min_{N_{out}} \mid v_u - N_{out} \mid, \tag{9}$$

then we will have

N'_{out}	$v_{y3}(2^3)$	$v_{y2}(2^2)$	$v_{y1}(2^1)$	$v_{y0}(2^0)$
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1
10	1	0	1	0
11	1	0	1	1
12	1	1	0	0
13	1	1	0	1
14	1	1	1	0
15	1	1	1	1

Table 2: Conventional binary digital representation

$$\epsilon_{ideal} = 1 \tag{10}$$

for the ideal input-output A/D conversion. We will discuss the A/D conversion error in more detail in the following sections. Observe that the *lengths* of the horizontal steps in the staircase graph in Figure 2 are *uniform* for *ideal* conversion to occur.

Ovserve that the digital representation of (8) can be transformed into any other digital representation system. For instance, if we assume that

$$v_u = 2v'_u - 15 \qquad \forall v'_u \in [-0.5, 15.5] \tag{11}$$

and

$$N'_{out} = 0.5(N_{out} + 15), \tag{12}$$

then we would obtain the conventional binary digital representation, whose bit values are represented by 0 and 1, as shown in *Table* 2. The corresponding input-output relationship between the analog value v'_u and the digital number N'_{out} is shown in Figure 2(b), which was used in [5, 8]. One advantage of our digital representation (8) is that it can represent both positive and negative values. For convenience, we use the representation system (8) in the following discussion.

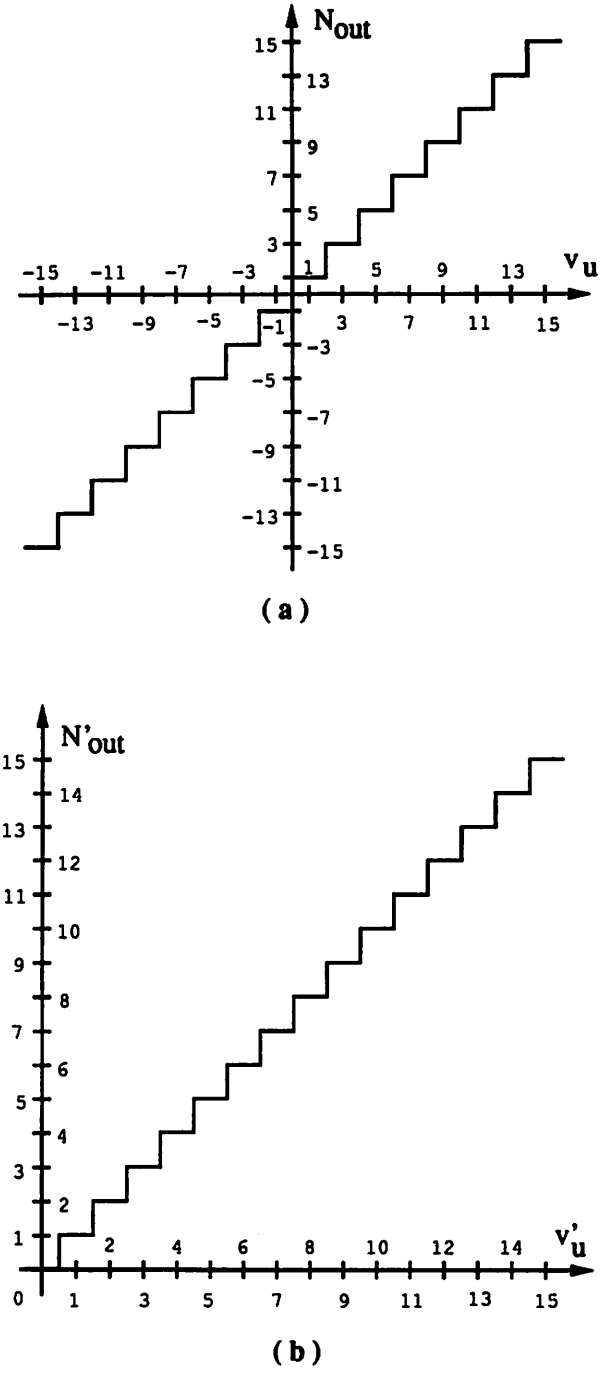


Figure 2: The A/D transfer functions. (a) for our CNN A/D converter; (b) for Tank and Hopfield's A/D converter.

An immediate requirement for our A/D converter from the digital representation (8) is that the output digital N_{out} should be a function of the input analog voltage v_u as indicated in Figure 2(a). Let us define the conversion error, ε, of an N-bit CNN A/D converter characterized by (1) and (2) as

$$\varepsilon = v_u - N_{out}, \tag{13}$$

where $v_c = 1$ is assumed as mentioned before.

It will be shown in the following sections that our CNN A/D converter seeks to minimize the conversion error, ε. For a good A/D converter, the A/D conversion error ε should be less than $\Delta N_{out} = 2$, where ΔN_{out} is the difference between two adjacent steps in Figure 2(a). We will show that we can design a CNN A/D converter for a maximum conversion error less than 2.

Another requirement of our digital representation (8) is the symmetry property of our CNN A/D converter. For instance, if a digital output value N'_{out} corresponds to an analog input value v'_u, then it is reasonable to require that the digital output value should be equal to $-N'_{out}$ for an analog input value of $-v'_u$. If we define a map from the input, v_u, to the output, N_{out}, as

$$\mathcal{G}: \quad N_{out} = \mathcal{G}(v_u), \tag{14}$$

then, the symmetry property can be represented analytically as

$$-N_{out} = -\mathcal{G}(v_u) = \mathcal{G}(-v_u). \tag{15}$$

We claim that the symmetry property is true for a *relaxed* CNN A/D converter. Here, a *relaxed* CNN A/D converter is defined by adding *zero* initial conditions to (1) and (2); namely,

$$v_{xi}(0) = 0, \tag{16}$$

where $1 \leq i \leq N$ for an N-bit converter.

The proof of this symmetry property can be divided into two steps. First, we will prove that if $\mathbf{v}^*_x(t)$ is a solution of (1) and (2) for an input v^*_u, then $-\mathbf{v}^*_x(t)$ will be a solution of (1) and (2) for the input $-v^*_u$. Since the system (1) and (2) satisfies the *Lipschitz* condition, the uniqueness of its solution for any given input v_u follows from the *fundamental existence and uniqueness theorem* for differential equations [4]. Therefore $-v^*_x$ is the only solution of (1) and (2) for the input $-v^*_u$. Then, by considering the obvious odd symmetry properties of equations (2) and (8), the symmetry property follows immediately.

Now, let us prove the first step. By multiplying the equations in (1) and (2) by -1, we obtain the following equivalent equations,

$$C_i \frac{d[-v_{xi}(t)]}{dt} = \frac{-1}{R_i}[-v_{xi}(t)] + \sum_{j=0}^{3} A_{ij}[-v_{yj}(t)] + b_i[-v_u] \tag{17}$$

$$-v_{yi}(t) = 0.5\,(-\,|\,-v_{xi}(t) - v_c\,| + |\,-v_{xi}(t) + v_c\,|) \tag{18}$$

$$-v_{xi}(0) = 0. \tag{19}$$

Let

$$v'_u = -v_u, \qquad v'_{xi}(t) = -v_{xi}(t), \qquad v'_{yi}(t) = -v_{yi}(t), \tag{20}$$

then it follows that

$$C_i \frac{dv'_{xi}(t)}{dt} = \frac{-1}{R_i} v'_{xi}(t) + \sum_{j=0}^{3} A_{ij} v'_{yj}(t) + b_i v'_u \tag{21}$$

$$v'_{yi}(t) = 0.5\,(\mid v'_{xi}(t) + v_c \mid - \mid v'_{xi}(t) - v_c \mid) \tag{22}$$

$$v'_{xi}(0) = 0. \tag{23}$$

Comparing (1) and (2) with (21) and (22) and assuming zero initial conditions, we can claim that if $\mathbf{v}^*_x(t)$ is a solution of (1) and (2) for an input v^*_u, then $-\mathbf{v}^*_x(t)$ is a solution of (1) and (2) for the input $-v^*_u$.

By using this symmetry property, we need only present our results for positive v_u in the following disccusions.

4 How to Design a CNN A/D Converter

In the previous sections, we have described our CNN A/D converter architecture, and defined a digital representation system corresponding to it. In this section, we will illustrate how to design the circuit parameters for our CNN A/D converter. Similar to the method used in [8], we consider the analog-to-digital conversion problem as a dynamic optimization issue; namely, given an analog input v_u, after the CNN A/D converter settles to its steady state, the error between the converted digital number N_{out} and the analog input v_u will be minimized.

For simplicity, we will design a 4-bit CNN A/D converter and use it as a vehicle to illustrate our design procedure. Let us assume the input value satisfies

$$v_u \in [-16, 16], \tag{24}$$

and

$$N_{out} = \sum_{i=0}^{3} 2^i v_{ysi}, \tag{25}$$

where $v_{ysi} = \pm 1$ is the corresponding discrete number resulting from the A/D conversion. Let us also define the A/D conversion error at time, t, as

$$\varepsilon(t) = v_u - \sum_{i=0}^{3} 2^i v_{yi}(t). \tag{26}$$

We can use the square of the A/D conversion error function

$$\varepsilon^2(t) = (v_u - \sum_{i=0}^{3} 2^i v_{yi}(t))^2 \tag{27}$$

as a part of the cost function of the associated analog-to-digital conversion optimization problem. In order to guarantee a binary-value output; namely, $v_{yi}(\infty) = \pm 1$, we need to introduce an additional cost function

$$-\sum_{i=0}^{3} a_i v_{yi}^2(t) \tag{28}$$

where a_i is an arbitrary positive weighting coefficient. It will be shown that condition (4), which guarantees binary outputs of CNN, introduces a term like (28) in the Lyapunov function of out CNN A/D converters automatically. Combining (27) and (28) together, we can define the following cost function for our analog-to-digital conversion problem as

$$f(t) = (v_u - \sum_{i=0}^{3} 2^i v_{yi}(t))^2 - \sum_{i=0}^{3} a_i v_{yi}^2(t). \tag{29}$$

In order to compare the cost function (29) with the Lyapunov function (3), let us rewrite (29) in the following form

$$f(t) = v_u^2 - 2v_u \sum_{i=0}^{3} 2^i v_{yi}(t) + \sum_{i=0}^{3}\sum_{j=0}^{3} 2^{i+j} v_{yi}(t) v_{yj}(t) \; - \sum_{i=0}^{3} a_i v_{yi}^2(t). \tag{30}$$

Comparing the second term of (30) with the third item of (3), we can determine the circuit parameter b_i as

$$b_i = 2^{i+1}. \tag{31}$$

To determine the circuit parameter A_{ij} and R_i, let us partition the third item of (30) into two parts as follows

$$\sum_{i=0}^{3}\sum_{j=0}^{3} 2^{i+j} v_{yi}(t) v_{yj}(t) = \sum_{i=0}^{3} 2^{2i} v_{yi}(t) + \sum_{i=0}^{3} \sum_{\substack{j=0 \\ i \neq j}}^{3} 2^{i+j} v_{yi}(t) v_{yj}(t). \tag{32}$$

Comparing (30) (with N = 4) and (3), and making use of (32), we obtain

$$A_{ij} = -2^{i+j+1}, \qquad for i \neq j. \tag{33}$$

From (30) and (3), we also have

$$-\frac{1}{2} A_{ii} + \frac{1}{2R_i} = 2^{2i} - a_i. \tag{34}$$

To satisfy condition (4) and $R_i > 0$, let us intuitively, for now, choose

$$R_i = 2^{-2i} \tag{35}$$

and

$$A_{ii} = 2^{2i+1}. \tag{36}$$

We will describe a method for an optimal design of A_{ii} and R_i later.

Solving equation (34) for a_i, we obtain

$$a_i = 3 \times 2^{2i-1} > 0, \tag{37}$$

which meets the requirement of the cost function (28). As mentioned before, this shows that the stability condition (4) will guarantee that (28) is included in the cost function.

Substituting all of the circuit parameters we have determined into the Lyapunov function (3) for a 4-bit CNN A/D converter, we obtain

$$\begin{aligned} E(t) &= \frac{-1}{2}\sum_{i=0}^{3} 2^{2i+1} v_{yi}^2(t) + \frac{1}{2}\sum_{i=0}^{3}\sum_{\substack{j=0 \\ i \neq j}}^{3} 2^{i+j+1} v_{yi}(t) v_{yj}(t) \\ &+ \frac{1}{2}\sum_{i=0}^{3} 2^{2i} v_{yi}^2(t) - \sum_{i=0}^{3} 2^{i+1} v_{yi}(t) v_u \\ &= \left[v_u - \sum_{i=0}^{3} 2^i v_{yi}(t) \right]^2 - \frac{3}{2}\sum_{i=0}^{3} 2^{2i} v_{yi}^2(t) - v_u^2. \end{aligned} \tag{38}$$

The first two terms in the above function is exactly the cost function, $f(t)$, defined in (29). The last term of (38) is a constant for a given value of v_u, which has no other effects on the optimization problem except that of normalizing the Lyapunov function to zero for a *relaxed* CNN A/D converter. We will analyze first the dynamic behavior of a *relaxed* CNN A/D converter in the following section, before considering the general CNN A/D converter having arbitrary initial conditions.

5 Dynamics of a Relaxed CNN A/D Converter

In this section, we analyze the 4-bit CNN A/D converter circuit designed in the previous section using linear system theory. We study the transient behavior of the circuit, that is, the dynamics of the A/D converter from its *zero initial state* to the steady state. We refer to a CNN A/D converter having *zero initial states* as *relaxed* because its Lyapunov function is zero at the initial time $t = 0$. Based on both theoretical analysis and computer simulation, we will present some

qualitative and quantitative results for our CNN A/D converter, which are also very informative for understanding other neural network A/D converters, such as Hopfield's.

The main result in this section is that although neural network A/D converters, such as Hopfield's circuit and ours, are thought to be *parallel processing* circuits, they actually perform *serial* computation internally without external control signals.

5.1 Analysis of the dynamic behaviors of CNN A/D converters

The circuit parameters of the 4-bit CNN A/D converter designed in the preceding section are given by

$$A_{ii} = 2^{2i+1} \tag{39}$$

$$A_{ij} = -2^{i+j+1} \qquad for \quad i \neq j; \tag{40}$$

$$b_i = 2^{i+1} \tag{41}$$

$$R_i = 2^{-2i} \tag{42}$$

$$C_i = C = 1, \tag{43}$$

where $0 \leq i \leq 3$.

Substituting these parameters into (1) and (2), and using the assumptions $v_c = 1$, $\mathbf{v_X}(0) = \mathbf{0}$, and $\mid v_u \mid \leq 16$, we can write the system equation of the 4-bit CNN A/D converter as follows:

State equation:

$$\begin{aligned}
\frac{d}{dt}\begin{pmatrix} v_{x0} \\ v_{x1} \\ v_{x2} \\ v_{x3} \end{pmatrix}(t) &= -\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 4 & 0 & 0 \\ 0 & 0 & 16 & 0 \\ 0 & 0 & 0 & 64 \end{pmatrix}\begin{pmatrix} v_{x0} \\ v_{x1} \\ v_{x2} \\ v_{x3} \end{pmatrix}(t) + \\
&+ \begin{pmatrix} 2 & -4 & -8 & -16 \\ -4 & 8 & -16 & -32 \\ -8 & -16 & 32 & -64 \\ -16 & -32 & -64 & 128 \end{pmatrix}\begin{pmatrix} v_{y0} \\ v_{y1} \\ v_{y2} \\ v_{y3} \end{pmatrix}(t) + \\
&+ \begin{pmatrix} 2 \\ 4 \\ 8 \\ 16 \end{pmatrix} v_u
\end{aligned} \tag{44}$$

Output equation:

$$v_{yi}(t) = 0.5\,(\mid v_{xi}(t) + 1 \mid - \mid v_{xi}(t) - 1 \mid) \tag{45}$$

Initial condition:

$$v_{xi}(0) = 0 \tag{46}$$

Input dynamical range:

$$| v_u | \leq 16, \tag{47}$$

where $0 \leq i \leq 3$.

First, let us examine the dynamic behavior of the CNN A/D converter over a short time interval $[0, \Delta t)$ where Δt is sufficiently small. From equations (46), the state voltage v_{xi} equals zero at the initial time $t = 0$ for all four cells. By considering the piecewise-linear output function in (45), we see that the nonlinear differential equations (44) – (47) are in fact linear at the starting time, since $v_{xi} = v_{yi}$ for $|v_{xi}| \leq 1$. Let us define

$$T_1 = \min \{t \ : \ | v_{xi}(t) | \geq 1, \quad for\ i = 0, 1, 2, and\ 3 \}. \tag{48}$$

Observe that $T_1 > 0$ exists in view of (6). Therefore, for $t \leq T_1$, equations (44) – (47) can be rewritten as

$$\begin{aligned}
\frac{d}{dt}\mathbf{v}_x(t) &= -\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 4 & 0 & 0 \\ 0 & 0 & 16 & 0 \\ 0 & 0 & 0 & 32 \end{pmatrix} \mathbf{v}_x(t) + \\
&+ \begin{pmatrix} 2 & -4 & -8 & -16 \\ -4 & 8 & -16 & -32 \\ -8 & -16 & 32 & -64 \\ -16 & -32 & -64 & 128 \end{pmatrix} \mathbf{v}_x(t) + \begin{pmatrix} 2 \\ 4 \\ 8 \\ 16 \end{pmatrix} v_u \\
&= \begin{pmatrix} 1 & -4 & -8 & -16 \\ -4 & 4 & -16 & -32 \\ -8 & -16 & 16 & -64 \\ -16 & -32 & -64 & 64 \end{pmatrix} \mathbf{v}_x(t) + \begin{pmatrix} 2 \\ 4 \\ 8 \\ 16 \end{pmatrix} v_u \\
&\equiv \mathbf{G}\mathbf{v}_x(t) + \mathbf{b}v_u.
\end{aligned} \tag{49}$$

$$\mathbf{v}_x(0) = \mathbf{0}, \qquad and \qquad |v_u| \leq 16. \tag{50}$$

This is a linear differential equation and can be solved analytically. The *zero-state* response of (49) is given by

$$\begin{aligned}
\mathbf{v}_x(t) &= \int_0^t e^{\mathbf{G}(t-\tau)}\mathbf{b}v_u d\tau \\
&= \int_0^t e^{\mathbf{G}(t-\tau)}d\tau\mathbf{b}v_u \\
&= \left[e^{\mathbf{G}(t-\tau)}\right]_0^t(-\mathbf{G}^{-1})\mathbf{b}v_u \\
&= \left[e^{\mathbf{G}t} - \mathbf{I}\right]\mathbf{G}^{-1}\mathbf{b}v_u
\end{aligned} \tag{51}$$

where $\mathbf{G}$ is a time invariant, symmetrical and nonsingular matrix; and $\mathbf{G}^{-1}$ is the inverse matrix of $\mathbf{G}$.

In order to analyze the dynamical behavior of the solution $\mathbf{v}_x$, let us study $\mathbf{G}$ further. Since $\mathbf{G}$ is a symmetric matrix, it can be decomposed into the form [6]

$$\mathbf{G} = \mathbf{Q}\mathbf{\Lambda}\mathbf{Q}^T, \quad \textit{and} \quad \mathbf{G}^{-1} = \mathbf{Q}\mathbf{\Lambda}^{-1}\mathbf{Q}^T, \tag{52}$$

where $\mathbf{Q}^T$ is the transpose matrix of $\mathbf{Q}$, and $\mathbf{Q}^T\mathbf{Q} = \mathbf{I}$; $\mathbf{\Lambda}$ is a diagonal matrix with the eigenvalues of $\mathbf{G}$ as its components.

$\mathbf{G}$ can be decomposed numerically by using EISPACK, a popular numerical analysis package for solving eigenvalue problems. The result of the decomposition of $\mathbf{G}$ is

$$\mathbf{\Lambda} = \begin{pmatrix} -52.8 & & & \\ & 112 & & \\ & & 4.37 & \\ & & & 21.4 \end{pmatrix}; \tag{53}$$

$$\mathbf{Q} = \begin{pmatrix} 0.286 & 0.0765 & 0.934 & -0.201 \\ 0.493 & 0.167 & -0.334 & -0.786 \\ 0.634 & 0.522 & -0.117 & 0.559 \\ 0.522 & -0.833 & -0.0544 & 0.174 \end{pmatrix}. \tag{54}$$

Now, the result in (52) allows us to recast (51) as follows [6]:

$$\mathbf{v}_x(t) = \mathbf{Q}\left(e^{\mathbf{\Lambda}t} - \mathbf{I}\right)\mathbf{\Lambda}^{-1}\mathbf{Q}^T\mathbf{b}v_u. \tag{55}$$

Before we analyze (55) further, let us examine first an approximate solution for very small t. If $t \ll \| \mathbf{\Lambda}^{-1} \|$, then

$$e^{\mathbf{\Lambda}t} \approx \mathbf{I} + \mathbf{\Lambda}t. \tag{56}$$

Therefore,

$$\mathbf{v}_x \approx \mathbf{Q}(\mathbf{\Lambda}t)\mathbf{\Lambda}^{-1}\mathbf{Q}^T\mathbf{b}v_u = \mathbf{b}v_u t. \tag{57}$$

Note that the above approximation can also be obtained directly by assuming $\mathbf{v}_x = 0$ in (49). From (57), we have the following observations:

(a) the sign of v_{x3} is the same as that of v_u;

(b) the maximum absolute value of the components of $\mathbf{v}_x$ is $|v_{x3}|$, because the maximum component of $\mathbf{b}$ is b_3.

We claim that the above observations are true at time T_1, that is,

$$v_{x3}(T_1) = \text{sgn}(v_u). \tag{58}$$

To see this, substitute the parameters from (53) - (54) into (55), and obtain

$$\begin{pmatrix} v_{x0} \\ v_{x1} \\ v_{x2} \\ v_{x3} \end{pmatrix}(t) = \begin{pmatrix} 0.400 - 0.00570e^{112t} - 0.0348e^{21.4t} - 0.273e^{4.37t} - 0.0867e^{-52.8t} \\ 0.200 - 0.0124e^{112t} - 0.136e^{21.4t} + 0.0977e^{4.37t} - 0.149e^{-52.8t} \\ 0.100 - 0.0389e^{112t} + 0.0967e^{21.4t} + 0.0342e^{4.37t} - 0.192e^{-52.8t} \\ -0.0922 + 0.0621e^{112t} + 0.0301e^{21.4t} + 0.0159e^{4.37t} - 0.0158e^{-52.8t} \end{pmatrix} v_u \quad (59)$$

Examining the above *zero-state* response, we see that the state variables v_{xi} is dominated by the term e^{112t}. Observe that v_{x3} has the largest weight, 0.0621, for the dominant term. This implies that the value of v_{x3} will change faster than the other state variables. Therefore, our claim (58) follows.

From the above discussion, we see that the most significant bit, namely bit-3, is first determined, and its value depends on the sign of the analog input v_u.

Now, let us continue to analyze the dynamics of the circuit. Suppose that

$$|v_{x3}(t)| \geq 1, \qquad \forall t \geq T_1, \quad (60)$$

then we will have

$$|v_{y3}| = 1, \qquad \forall t \geq T_1. \quad (61)$$

Since v_{y3} is a constant value for $t \geq T_1$, the first three equations in (44) can be rewritten as

$$\begin{aligned} \frac{d}{dt}\begin{pmatrix} v_{x0} \\ v_{x1} \\ v_{x2} \end{pmatrix}(t) = \; &= -\begin{pmatrix} 1 & 0 & 0 \\ 0 & 4 & 0 \\ 0 & 0 & 16 \end{pmatrix}\begin{pmatrix} v_{x0} \\ v_{x1} \\ v_{x2} \end{pmatrix}(t) + \\ &+ \begin{pmatrix} 2 & -4 & -8 \\ -4 & 8 & -16 \\ -8 & -16 & 32 \end{pmatrix}\begin{pmatrix} v_{y0} \\ v_{y1} \\ v_{y2} \end{pmatrix}(t) + \\ &+ \begin{pmatrix} 2 \\ 4 \\ 8 \end{pmatrix} v_u - \begin{pmatrix} 16 \\ 32 \\ 64 \end{pmatrix} v_{y3} \\ &= \begin{pmatrix} 1 & 0 & 0 \\ 0 & 4 & 0 \\ 0 & 0 & 16 \end{pmatrix}\begin{pmatrix} v_{x0} \\ v_{x1} \\ v_{x2} \end{pmatrix}(t) + \\ &+ \begin{pmatrix} 2 & -4 & -8 \\ -4 & 8 & -16 \\ -8 & -16 & 32 \end{pmatrix}\begin{pmatrix} v_{y0} \\ v_{y1} \\ v_{y2} \end{pmatrix}(t) + \\ &+ \begin{pmatrix} 2 \\ 4 \\ 8 \end{pmatrix} \mathrm{sgn}(v_u) \mid v_u - 8 \mid, \end{aligned} \quad (62)$$

where we have substituted $v_{y3} = \mathrm{sgn}(v_u)$. As before, let us define

$$T_2 = \min\{t \; : \; t > T_1, \; \textit{and} \; |v_{xi}(t)| \geq 1, \quad \textit{for } i = 0, 1, \textit{and } 2\}. \quad (63)$$

Over the time interval $T_1 \leq t \leq T_2$, equation (62) is once again a linear differential equation and can be written as

$$\begin{aligned} \frac{d}{dt}\mathbf{v'}_x(t) &= \begin{pmatrix} 1 & -4 & -8 \\ -4 & 4 & -16 \\ -8 & -16 & 16 \end{pmatrix} \mathbf{v'}_x(t) + \begin{pmatrix} 2 \\ 4 \\ 8 \end{pmatrix} v'_u \\ &\equiv \mathbf{G'v'}_x(t) + \mathbf{b'}v'_u, \end{aligned} \tag{64}$$

where

$$v'_u \equiv \mathrm{sgn}(v_u)\,[|v_u| - 8]. \tag{65}$$

The solution of (63) is

$$\mathbf{v'}_x(t) = e^{\mathbf{G'}t}\mathbf{v'}_x(T_1) + \left[e^{\mathbf{G'}t} - \mathbf{I}\right] \mathbf{G'}^{-1}\mathbf{b}v'_u, \qquad for\ T_1 < t < T_2. \tag{66}$$

Equation (66) has the same form as (51), except for the additional first term which is due to the non-zero initial condition $\mathbf{v'}_x(T_1)$.

Again let us decompose $\mathbf{G'}$ into

$$\mathbf{G'} = \mathbf{Q'\Lambda'Q'}^T. \tag{67}$$

The numerical solution of (67) is

$$\mathbf{\Lambda'} = \begin{pmatrix} 5.18 & & \\ & -12 & \\ & & 27.8 \end{pmatrix}; \tag{68}$$

$$\mathbf{Q'} = \begin{pmatrix} 0.831 & -0.530 & -0.169 \\ -0.530 & -0.662 & -0.530 \\ -0.169 & -0.530 & 0.831 \end{pmatrix}. \tag{69}$$

It is clear that the solution (66) depends not only on the input v'_u but also on the non-zero initial condition $\mathbf{v'}_x(T_1)$. Therefore, we can not directly apply the previous results (58) for bit-3 to bit-2. Substituting (68) and (69) into (66), we obtain

$$\begin{pmatrix} v_{x0} \\ v_{x1} \\ v_{x2} \end{pmatrix}(t) = \begin{pmatrix} (0.667 - 0.0254e^{27.8t} - 0.29e^{5.18t} - 0.351e^{-12t})v'_u \\ -0.169c_1e^{27.8t} + 0.831c_2e^{5.18t} - 0.53c_3e^{-12t} \\ (0.334 - 0.0799e^{27.8t} + 0.185e^{5.18t} - 0.439e^{-12t})v'_u \\ -0.53c_1e^{27.8t} - 0.53c_2e^{5.18t} - 0.662c_3e^{-12t} \\ (0.167 + 0.125e^{27.8t} + 0.06e^{5.18t} - 0.351e^{-12t})v'_u \\ +0.831c_1e^{27.8t} - 0.169c_2e^{5.18t} - 0.53c_3e^{-12t} \end{pmatrix}, \tag{70}$$

where

$$c_1 = -0.169v_{x0}(T_1) - 0.53v_{x1}(T_1) + 0.831v_{x2}(T_1); \tag{71}$$

$$c_2 = 0.831v_{x0}(T_1) - 0.53v_{x1}(T_1) - 0.169v_{x2}(T_1); \tag{72}$$

$$c_3 = -0.53v_{x0}(T_1) - 0.662v_{x1}(T_1) - 0.53v_{x2}(T_1). \tag{73}$$

In order to uncover some useful results from the solution (70), let us assume that

$$|v_{xi}(T_1)| \ll 1, \quad for\ i = 0,\ 1,\ and\ 2. \tag{74}$$

In this case, the solution (70) can be approximated by

$$\begin{pmatrix} v_{x0} \\ v_{x1} \\ v_{x2} \end{pmatrix}(t) \approx \begin{pmatrix} (0.667 - 0.0254e^{27.8t} - 0.29e^{5.18t} - 0.351e^{-12t})v'_u \\ (0.334 - 0.0799e^{27.8t} + 0.185e^{5.18t} - 0.439e^{-12t})v'_u \\ (0.167 + 0.125e^{27.8t} + 0.06e^{5.18t} - 0.351e^{-12t})v'_u \end{pmatrix}. \tag{75}$$

The dominant term in (75) is $e^{27.8t}$. Applying the same reasoning as that used for (59), we can say that the state voltage of bit-2, v_{x2}, increases faster than that of bit-1 or bit-0, and has the same sign as v'_u. Therefore, under the assumption (74), we have

$$v_{x2}(T_2) = \mathrm{sgn}(v'_u). \tag{76}$$

It remains for us to analyze the last two bits. Suppose that for $t \geq T_2$, $|v_{x2}(t)| \geq 1$, and let

$$T_3 = \min\{t\ :\ t > T_2,\ and\ |v_{xi}(t)| \geq 1, \quad for\ i = 0\ and\ 1\}. \tag{77}$$

Then, we obtain a two-dimensional linear differential equation for $T_2 \leq t \leq T_3$, that is,

$$\begin{aligned} \frac{d}{dt}\begin{pmatrix} v_{x0} \\ v_{x1} \end{pmatrix}(t) &= \begin{pmatrix} 1 & -4 \\ -4 & 4 \end{pmatrix}\begin{pmatrix} v_{x0} \\ v_{x1} \end{pmatrix}(t) + \begin{pmatrix} 2 \\ 4 \end{pmatrix} v_u - \\ &- \begin{pmatrix} 16 \\ 32 \end{pmatrix} v_{y3} - \begin{pmatrix} 8 \\ 16 \end{pmatrix} v_{y2} \\ &= \begin{pmatrix} 1 & -4 \\ -4 & 4 \end{pmatrix}\begin{pmatrix} v_{x0} \\ v_{x1} \end{pmatrix}(t) + \\ &+ \begin{pmatrix} 2 \\ 4 \end{pmatrix}[v_u - 8\mathrm{sgn}(v_u) - 4\mathrm{sgn}(v'_u)], \\ &\equiv \mathbf{G}''\mathbf{v}''_x(t) + \mathbf{b}''v''_u, \end{aligned} \tag{78}$$

where

$$\begin{aligned} v''_u &\equiv v_u - 8\mathrm{sgn}(v_u) - 4\mathrm{sgn}(v'_u) \\ &= \mathrm{sgn}(v_u)\,[|v_u| - 8 - 4\mathrm{sgn}(|v_u| - 8)] \\ &= \mathrm{sgn}(v_u)\mathrm{sgn}(|v_u| - 8)(|\,|v_u| - 8\,| - 4). \end{aligned} \tag{79}$$

Also,

$$\mathbf{G}'' = \mathbf{Q}''\mathbf{\Lambda}''\mathbf{Q}''^T, \tag{80}$$

where

$$\mathbf{\Lambda}'' = \begin{pmatrix} -1.77 & \\ & 6.77 \end{pmatrix}; \quad \mathbf{Q}'' = \begin{pmatrix} -0.822 & 0.57 \\ -0.57 & -0.822 \end{pmatrix}. \tag{81}$$

The solution of (78) is

$$\begin{aligned} \mathbf{v}''_x(t) &= e^{\mathbf{G}''t}\mathbf{v}''_x(T_2) + \left[e^{\mathbf{G}''t} - \mathbf{I}\right]\mathbf{G}''^{-1}\mathbf{b}v''_u, \\ &= \begin{pmatrix} (2 - 0.181e^{6.77t} - 1.82e^{-1.77t})v''_u \\ +0.57c'_1e^{6.77t} - 0.822c'_2e^{-1.77t} \\ (1.01 + 0.261e^{6.77t} - 1.27e^{-1.77t})v''_u \\ -0.822c'_1e^{6.77t} - 0.57c'_2e^{-1.77t} \end{pmatrix}, \end{aligned} \tag{82}$$

where,

$$c'_1 = 0.57v_{x0}(T_2) - 0.822v_{x1}(T_2); \tag{83}$$

$$c'_2 = -0.822v_{x0}(T_2) - 0.57v_{x1}(T_2). \tag{84}$$

Again, if we suppose that

$$|v_{xi}(T_2)| \ll 1, \quad \textit{for } i = 0 \textit{ and } 1, \tag{85}$$

then, we have

$$\mathbf{v}''_x(t) \approx \begin{pmatrix} (2 - 0.181e^{6.77t} - 1.82e^{-1.77t})v''_u \\ (1.01 + 0.261e^{6.77t} - 1.27e^{-1.77t})v''_u \end{pmatrix}. \tag{86}$$

Therefore, under the assumption (85), we obtain

$$v_{x1}(T_3) = \mathrm{sgn}(v''_u). \tag{87}$$

Finally, we consider the last bit, bit-0. Suppose that for $t \geq T_3$, $|v_{x1}(t)| \geq 1$. It can easily be shown that, for $t \geq T_3$,

$$\frac{dv_{x0}(t)}{dt} = v_{x0}(t) + 2v'''_u \tag{88}$$

where

$$\begin{aligned} v'''_u &\equiv v_u - 8v_{y3} - 4v_{y2} - 2v_{y1} \\ &= \mathrm{sgn}(v_u)\mathrm{sgn}\left(|v_u| - 8\right)\mathrm{sgn}\left(|\,|v_u| - 8\,| - 4\right) \\ &\quad \left(|\,|\,|v_u| - 8\,| - 4\,| - 2\right). \end{aligned} \tag{89}$$

The solution of (88) can be obtained, giving

$$v_{x0}(t) = v_{x0}(T_3)e^t + (e^t - 1)v'''_u. \tag{90}$$

As before, if we assume that

$$|v_{x0}(T_3)| \ll 1, \tag{91}$$

then we obtain

$$v_{x0}(T_4) = \text{sgn}(v_u'''), \tag{92}$$

for some $T_4 > T_3$.

Our preceding analysis yields the following two important results. First, by considering *Table* 1, we observe the following relationship between the output voltages of the four bits and the output digital numbers.

$$v_{y3} = \text{sgn}(N_{out}); \tag{93}$$

$$v_{y2} = \text{sgn}(N_{out})\text{sgn}\left(|N_{out}| - 8\right); \tag{94}$$

$$v_{y1} = \text{sgn}\left(N_{out}\right)\text{sgn}\left(|N_{out}| - 8\right)\text{sgn}\left(|\left(|N_{out}| - 8\right)| - 4\right); \tag{95}$$

$$\begin{aligned} v_{y0} \;=\; & \text{sgn}\left(N_{out}\right)\text{sgn}\left(|N_{out}| - 8\right)\text{sgn}\left(|\left(|N_{out}| - 8\right)| - 4\right) \\ & \text{sgn}\left(|\left(|\left(|N_{out}| - 8\right)| - 4\right)| - 2\right). \end{aligned} \tag{96}$$

On the other hand, we have shown that, under the *assumptions* in (74), (85), and (91), the relationship between the output voltages v_{yi} of the four bits and the input analog voltage v_u has the same form as (93) – (96) if N_{out} is replaced by v_u as follows

$$v_{y3} = \text{sgn}(v_u); \tag{97}$$

$$v_{y2} = \text{sgn}(v_u)\text{sgn}\left(|v_u| - 8\right); \tag{98}$$

$$v_{y1} = \text{sgn}\left(v_u\right)\text{sgn}\left(|v_u| - 8\right)\text{sgn}\left(|\left(|v_u| - 8\right)| - 4\right); \tag{99}$$

$$\begin{aligned} v_{y0} \;=\; & \text{sgn}\left(v_u\right)\text{sgn}\left(|v_u| - 8\right)\text{sgn}\left(|\left(|v_u| - 8\right)| - 4\right) \\ & \text{sgn}\left(|\left(|\left(|v_u| - 8\right)| - 4\right)| - 2\right). \end{aligned} \tag{100}$$

Here, we have ignored the values of v_u such that v_{yi} is zero, because $v_{yi} = 0$ is an *unstable* value and can not be observed in a physical circuit as discussed in [3]. We can immediately see that (97) – (98) represent the ideal A/D conversion shown in Figure 2(a). Therefore, it follows that, under our above assumptions, the CNN A/D converter can linearly transform an analog input into a digital output via a dynamical transition. From the optimization view point, it follows that under the assumptions we have made, the Lyapunov or energy function $E(t)$ will settle down to its *global* minimum point.

Observe that the A/D conversion is performed in a *serial* manner, that is, the most significant bit comes out first, then the next most significant bit, and

so forth. This observation is of fundamental theoretical importance even though it is not a desirable property of A/D conversion, because our neural network A/D converter was originally thought to have a parallel processing capability. Although we discussed only a neural model with piecewise-linear sigmoidal characteristics here, we believe that the above observation holds for any neural network A/D converter with a finite gain transition sigmoidal function.

Unfortunately, the assumptions (74), (85), and (91) which we made in the above analysis are not always true. Violation of any of these assumptions will cause conversion errors, and therefore drive the energy function $E(t)$ to a stable *local* minimum.

An analysis of the general case without assumptions is very difficult, if not impossible. Instead, we will present a set of computer simulations in the following subsection and show that our above simplified analysis is consistent with computer simulations. Furthermore, the results of our simplified analysis will provide some guidelines on how to reduce the A/D conversion errors, which will also be presented in the following subsection. Finally, a constructive algorithm for designing an N-bit A/D converter having any prescribed conversion error will be presented and proved in *Section* 7.

5.2 Computer simulations of the A/D converter

In this subsection, we present a set of computer simulation results to verify our analysis of the previous subsection. Because of the symmetry property of our CNN A/D converter circuit, as proved in *Section* 3, we will choose only positive values for the analog input v_u in these simulations.

Again, the circuit simulator we used is PWLSPICE [1]. The simulated CNN A/D converter circuit is characterized by the system equation (44) – (47) except that we have used the following more practical resistor and capacitor values:

$$R_i = 2^{-2i}R = 2^{-2i}k\Omega \qquad for \quad i = 0, 1, 2, \; and \; 3 \tag{101}$$

$$C_i = C = 1pF \qquad for \quad i = 0, 1, 2, \; and \; 3 \tag{102}$$

where R stands for a constant resistance, and C stands for a constant capacitance. Let us define the time constant

$$\tau_{RC} = RC, \tag{103}$$

which has no effect on the steady state response but will affect the transient speed of the circuit.

The eight graphs in Figure 3 are simulation results of the *zero-state response* $\mathbf{v}_x(t)$ for analog inputs v_u corresponding to the eight positive numbers in *Table* 1.

Observe that the output digital number N_{out} is exactly the analog input value v_u coded according to the digital representation (8), or *Table* 1. In the parlance of A/D conversion, this means that our A/D converter performs a

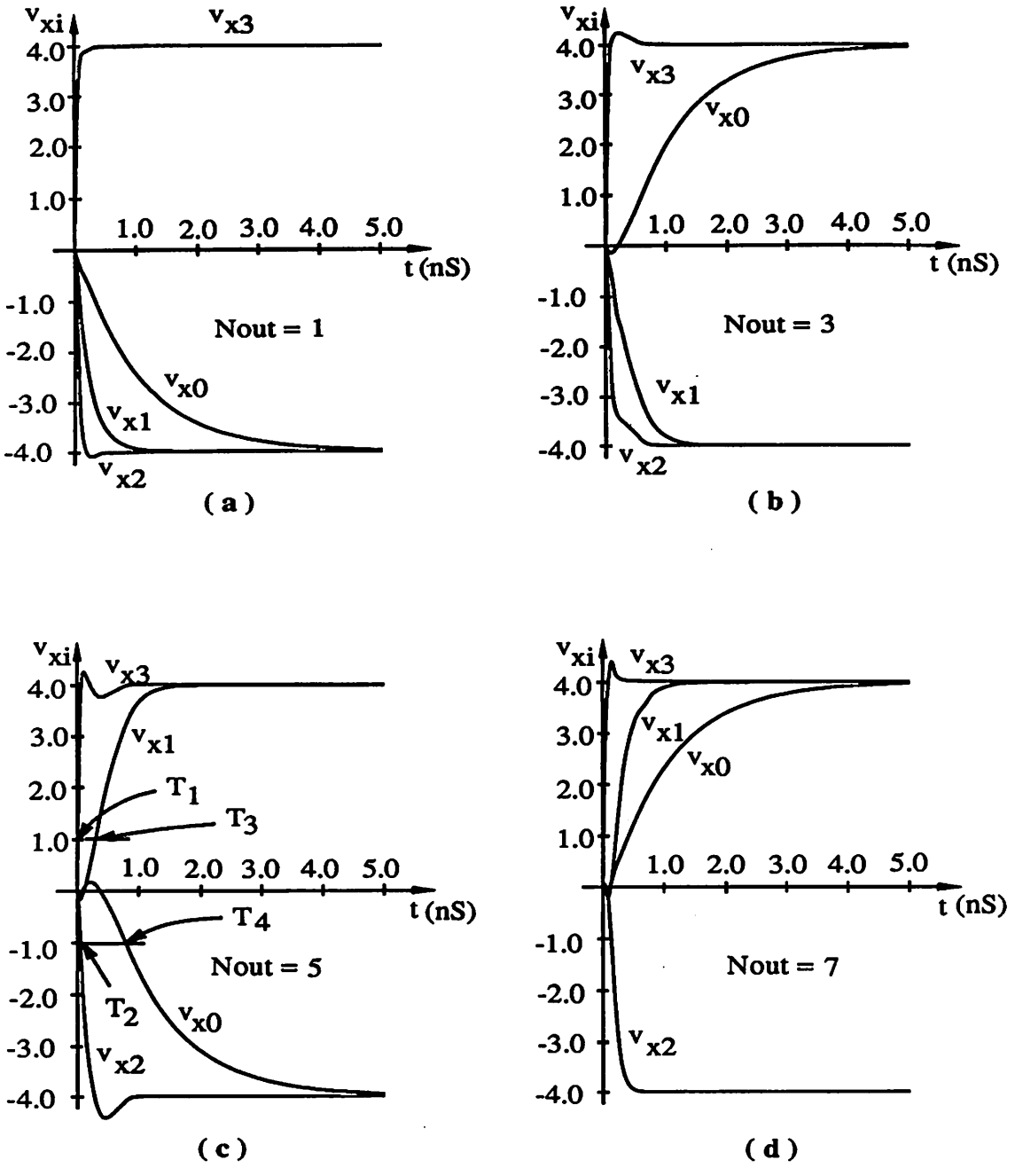
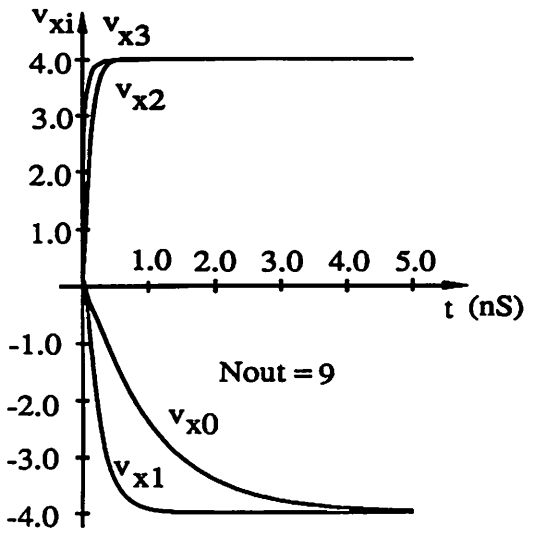
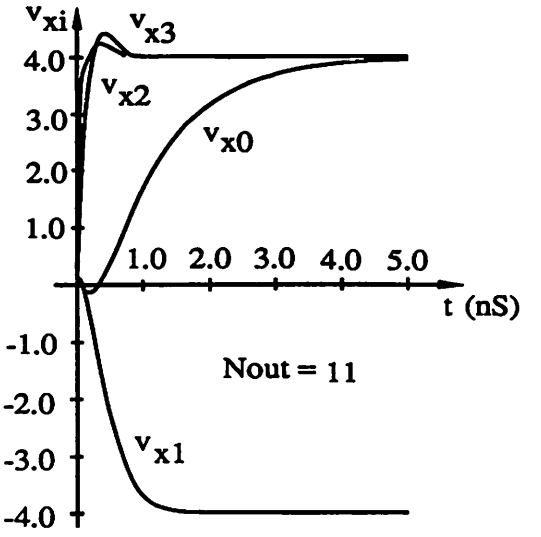


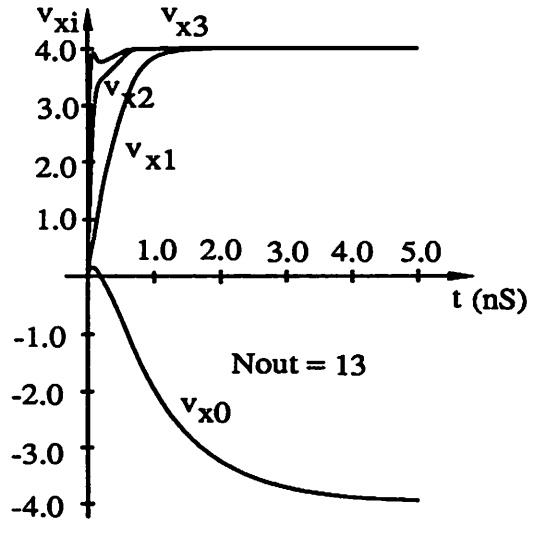
Figure 3: Simulation results for the CNN A/D converter.



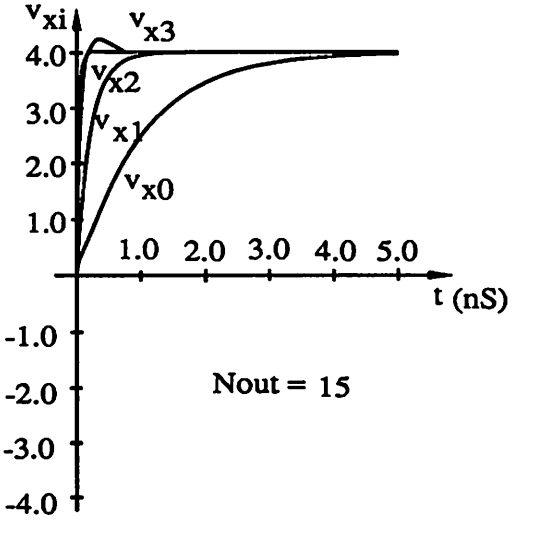
(e)



(f)



(g)



(h)

Figure 3: Simulation results for the CNN A/D converter.(Cont.)

linear transformation. Here, the term *linear transformation* means that if two analog inputs satisfy $v_{u1} \leq v_{u2}$ then the corresponding digital outputs satisfy $N_{out1} \leq N_{out2}$. It also shows that the A/D conversion error ε is less than 2.

Another observation from Figure 3 is that the transient behaviors of the eight cases are exactly what we have predicted in the previous subsection, that is, the state voltage v_{x3} of bit-3 has the fastest transition speed, and v_{x0} has the slowest speed. To be more specific, let us look at the case $u_u = 5.0$ in Figure 3(c). The critical time values, which are defined in *Section* 5.1, are:

$$T_1 = 0.012ns; \tag{104}$$

$$T_2 = 0.056ns; \tag{105}$$

$$T_3 = 0.38ns; \tag{106}$$

$$T_4 = 0.78ns. \tag{107}$$

Observe from Figure 3(c) that

$$v_{x3}(T_1) = 1 \ \textit{and} \ v_{x3}(t) > 1 \ \textit{for} \ t > T_1; \tag{108}$$

$$v_{x2}(T_2) = -1 \ \textit{and} \ v_{x2}(t) < -1 \ \textit{for} \ t > T_2; \tag{109}$$

$$v_{x1}(T_3) = 1 \ \textit{and} \ v_{x1}(t) > 1 \ \textit{for} \ t > T_3; \tag{110}$$

$$v_{x0}(T_4) = -1 \ \textit{and} \ v_{x0}(t) < -1 \ \textit{for} \ t > T_4 \tag{111}$$

These observations are consistent with our analysis in *Section* 5.1.

Table 3 lists 160 simulation results. From *Table* 3 we can construct the A/D transfer function of our A/D converter, as shown in Figure 4(a). The maximum conversion error ε_{max} calculated from *Table* 3 is

$$\varepsilon_{max} \equiv \max_{v_u} |\varepsilon| = 1.8. \tag{112}$$

This occurs when $v_u = 7.2$. For an ideal A/D converter, the maximum conversion error should be 1, which is half of ΔN_{out}.

How can we improve the accuracy of the A/D conversion ? In *Section* 7, we will discuss this issue in general. For now, let us restrict to the relaxed CNN A/D converter.

From the analysis in the previous subsection, we know that if conditions (74), (85), and (91) were satisfied, then the A/D conversion would attain the minimum error. This suggests that we should optimize the circuit parameters of our A/D converter in order to enhance the possibility of satisfying those conditions.

In order to see which circuit parameters affect these conditions, let us define the *bit transient time constant* τ_i for the ith bit as

V_u	N_{out}	$\mid \varepsilon \mid$	V_u	N_{out}	$\mid \varepsilon \mid$	V_u	N_{out}	$\mid \varepsilon \mid$	V_u	N_{out}	$\mid \varepsilon \mid$
0.1	1	0.9	4.1	3	1.1	8.1	9	0.9	12.1	13	0.9
0.2	1	0.8	4.2	3	1.2	8.2	9	0.8	12.2	13	0.8
0.3	1	0.7	4.3	3	1.3	8.3	9	0.7	12.3	13	0.7
0.4	1	0.6	4.4	3	1.4	8.4	9	0.6	12.4	13	0.6
0.5	1	0.5	4.5	5	0.5	8.5	9	0.5	12.5	13	0.5
0.6	1	0.4	4.6	5	0.4	8.6	9	0.4	12.6	13	0.4
0.7	1	0.3	4.7	5	0.3	8.7	9	0.3	12.7	13	0.3
0.8	1	0.2	4.8	5	0.2	8.8	9	0.2	12.8	13	0.2
0.9	1	0.1	4.9	5	0.1	8.9	9	0.1	12.9	13	0.1
1.0	1	0.0	5.0	5	0.0	9.0	9	0.0	13.0	13	0.0
1.1	1	0.1	5.1	5	0.1	9.1	9	0.1	13.1	13	0.1
1.2	1	0.2	5.2	5	0.2	9.2	9	0.2	13.2	13	0.2
1.3	1	0.3	4.3	5	0.3	9.3	9	0.3	13.3	13	0.3
1.4	1	0.4	5.4	5	0.4	9.4	9	0.4	13.4	13	0.4
1.5	1	0.5	5.5	5	0.5	9.5	9	0.5	13.5	13	0.5
1.6	1	0.6	5.6	5	0.6	9.6	9	0.6	13.6	13	0.6
1.7	1	0.7	5.7	5	0.7	9.7	9	0.7	13.7	13	0.7
1.8	1	0.8	5.8	5	0.8	9.8	9	0.8	13.8	13	0.8
1.9	1	0.9	5.9	7	1.1	9.9	9	0.9	13.9	15	1.1
2.0	1	1.0	6.0	7	1.0	10.0	9	1.0	14.0	15	1.0
2.1	1	1.1	6.1	7	0.9	10.1	9	1.1	14.1	15	0.9
2.2	3	0.8	6.2	7	0.8	10.2	11	0.8	14.2	15	0.8
2.3	3	0.7	6.3	7	0.7	10.3	11	0.7	14.3	15	0.7
2.4	3	0.6	6.4	7	0.6	10.4	11	0.6	14.4	15	0.6
2.5	3	0.5	6.5	7	0.5	10.5	11	0.5	14.5	15	0.5
2.6	3	0.4	6.6	7	0.4	10.6	11	0.4	14.6	15	0.4
2.7	3	0.3	6.7	7	0.3	10.7	11	0.3	14.7	15	0.3
2.8	3	0.2	6.8	7	0.2	10.8	11	0.2	14.8	15	0.2
2.9	3	0.1	6.9	7	0.1	10.9	11	0.1	14.9	15	0.1
3.0	3	0.0	7.0	7	0.0	11.0	11	0.0	15.0	15	0.0
3.1	3	0.1	7.1	7	0.1	11.1	11	0.1	15.1	15	0.1
3.2	3	0.2	7.2	9	1.8	11.2	11	0.2	15.2	15	0.2
3.3	3	0.3	7.3	9	1.7	11.3	11	0.3	15.3	15	0.3
3.4	3	0.4	7.4	9	1.6	11.4	11	0.4	15.4	15	0.4
3.5	3	0.5	7.5	9	1.5	11.5	11	0.5	15.5	15	0.5
3.6	3	0.6	7.6	9	1.4	11.6	13	1.4	15.6	15	0.6
3.7	3	0.7	7.7	9	1.3	11.7	13	1.3	15.7	15	0.7
3.8	3	0.8	7.8	9	1.2	11.8	13	1.2	15.8	15	0.8
3.9	3	0.9	7.9	9	1.1	11.9	13	1.1	15.9	15	0.9
4.0	3	1.0	8.0	9	1.0	12.0	13	1.0	16.0	15	1.0

Table 3: Input/output mapping list of A/D converter

$$\tau_i = R_i C_i. \tag{113}$$

For the parameters in (39) – (43), we have

$$\tau_0 = RC = \tau_{RC}; \tag{114}$$

$$\tau_1 = 2^{-2} RC = 2^{-2} \tau_{RC}; \tag{115}$$

$$\tau_2 = 2^{-4} RC = 2^{-4} \tau_{RC}; \tag{116}$$

$$\tau_3 = 2^{-8} RC = 2^{-8} \tau_{RC}. \tag{117}$$

From these time constants, it becomes clear why the four bits have different transient speeds. The ratio of the time constants between two contiguous bits is

$$r_\tau \equiv \frac{\tau_i}{\tau_{i+1}} = 2^2 = 4. \tag{118}$$

Since the transient speed of the state variable v_{xi} is proportional to its time constant, if the ratio r_τ is increased, then conditions (74), (85), and (91) will hold for more input analog values. Let us verify this by changing the capacitor values in (43). let

$$C_i = 2^{3-i} C \qquad for\ i\ =\ 0,\ 1,\ 2,\ 3. \tag{119}$$

Note that after this change, the matrix $\mathbf{G}$ in (49) is no longer symmetric. Nevertheless, it can still be decomposed as

$$\mathbf{G} = \mathbf{Q}\mathbf{\Lambda}\mathbf{P}, \qquad where \quad \mathbf{G}^{-1} = \mathbf{P}^{-1}\mathbf{\Lambda}^{-1}\mathbf{Q}^{-1}, \tag{120}$$

as long as its eigenvectors are linearly independent.

The ratio r_τ becomes

$$r_\tau = 2^3 = 8, \tag{121}$$

for this case.

The simplified theoretical analysis of this circuit is the same as in *Section* 5.1. To save space, we only present here the computer simulation results; see *Table* 4, or Figure 4(b). From this table, the maximum A/D conversion error ε_{max} is:

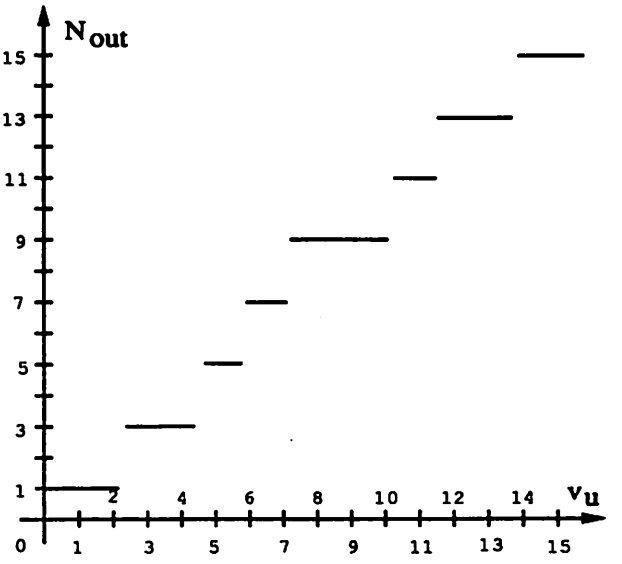
$$\varepsilon_{max} = 1.3, \tag{122}$$

which is much better than that in (112), and is very close to 1, the ideal conversion error. This significantly improved performance is manifested by the step lengths in Figure 4(b), which are now of *almost* uniform length. However, the transition or conversion speed of the latter circuit is slower than that of the former, because of the larger values of the capacitors. Thus, *we must trade accuracy for speed.*

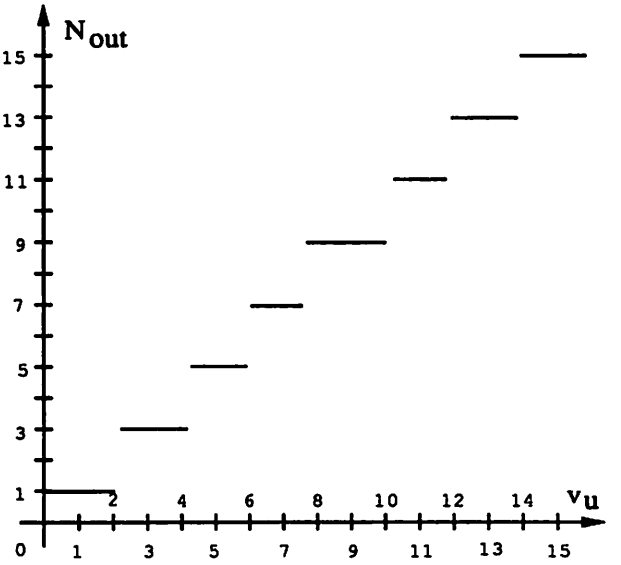
In this section, we have analyzed the dynamics of a relaxed 4-bit CNN A/D converter, and have shown that by increasing the ratio r_τ, we can reduce the analog-to-digital conversion error. In the following two sections, we will study the steady state behavior of the CNN A/D converter, and use a similar method to improve the performance of the circuit.

V_u	N_{out}	$\mid \varepsilon \mid$	V_u	N_{out}	$\mid \varepsilon \mid$	V_u	N_{out}	$\mid \varepsilon \mid$	V_u	N_{out}	$\mid \varepsilon \mid$
0.1	1	0.9	4.1	3	1.1	8.1	9	0.9	12.1	13	0.9
0.2	1	0.8	4.2	5	0.8	8.2	9	0.8	12.2	13	0.8
0.3	1	0.7	4.3	3	0.7	8.3	9	0.7	12.3	13	0.7
0.4	1	0.6	4.4	5	0.6	8.4	9	0.6	12.4	13	0.6
0.5	1	0.5	4.5	5	0.5	8.5	9	0.5	12.5	13	0.5
0.6	1	0.4	4.6	5	0.4	8.6	9	0.4	12.6	13	0.4
0.7	1	0.3	4.7	5	0.3	8.7	9	0.3	12.7	13	0.3
0.8	1	0.2	4.8	5	0.2	8.8	9	0.2	12.8	13	0.2
0.9	1	0.1	4.9	5	0.1	8.9	9	0.1	12.9	13	0.1
1.0	1	0.0	5.0	5	0.0	9.0	9	0.0	13.0	13	0.0
1.1	1	0.1	5.1	5	0.1	9.1	9	0.1	13.1	13	0.1
1.2	1	0.2	5.2	5	0.2	9.2	9	0.2	13.2	13	0.2
1.3	1	0.3	4.3	5	0.3	9.3	9	0.3	13.3	13	0.3
1.4	1	0.4	5.4	5	0.4	9.4	9	0.4	13.4	13	0.4
1.5	1	0.5	5.5	5	0.5	9.5	9	0.5	13.5	13	0.5
1.6	1	0.6	5.6	5	0.6	9.6	9	0.6	13.6	13	0.6
1.7	1	0.7	5.7	5	0.7	9.7	9	0.7	13.7	13	0.7
1.8	1	0.8	5.8	5	0.8	9.8	9	0.8	13.8	13	0.8
1.9	1	0.9	5.9	5	0.9	9.9	9	0.9	13.9	13	0.9
2.0	1	1.0	6.0	7	1.0	10.0	9	1.0	14.0	15	1.0
2.1	3	0.9	6.1	7	0.9	10.1	11	0.9	14.1	15	0.9
2.2	3	0.8	6.2	7	0.8	10.2	11	0.8	14.2	15	0.8
2.3	3	0.7	6.3	7	0.7	10.3	11	0.7	14.3	15	0.7
2.4	3	0.6	6.4	7	0.6	10.4	11	0.6	14.4	15	0.6
2.5	3	0.5	6.5	7	0.5	10.5	11	0.5	14.5	15	0.5
2.6	3	0.4	6.6	7	0.4	10.6	11	0.4	14.6	15	0.4
2.7	3	0.3	6.7	7	0.3	10.7	11	0.3	14.7	15	0.3
2.8	3	0.2	6.8	7	0.2	10.8	11	0.2	14.8	15	0.2
2.9	3	0.1	6.9	7	0.1	10.9	11	0.1	14.9	15	0.1
3.0	3	0.0	7.0	7	0.0	11.0	11	0.0	15.0	15	0.0
3.1	3	0.1	7.1	7	0.1	11.1	11	0.1	15.1	15	0.1
3.2	3	0.2	7.2	7	0.2	11.2	11	0.2	15.2	15	0.2
3.3	3	0.3	7.3	7	0.3	11.3	11	0.3	15.3	15	0.3
3.4	3	0.4	7.4	7	0.4	11.4	11	0.4	15.4	15	0.4
3.5	3	0.5	7.5	7	0.5	11.5	11	0.5	15.5	15	0.5
3.6	3	0.6	7.6	7	0.6	11.6	11	0.6	15.6	15	0.6
3.7	3	0.7	7.7	9	1.3	11.7	11	0.7	15.7	15	0.7
3.8	3	0.8	7.8	9	1.2	11.8	11	0.8	15.8	15	0.8
3.9	3	0.9	7.9	9	1.1	11.9	13	1.1	15.9	15	0.9
4.0	3	1.0	8.0	9	1.0	12.0	13	1.0	16.0	15	1.0

Table 4: Input/output mapping list of A/D converter



(a)



(b)

Figure 4: Input/output functions of the A/D converter

N_{out}	$E(v_u)$
1	$-2v_u - 126.5$
3	$-6v_u - 118.5$
5	$-10v_u - 102.5$
7	$-14v_u - 78.5$
9	$-18v_u - 46.5$
11	$-22v_u - 6.5$
13	$-26v_u + 41.5$
15	$-30v_u + 97.5$

Table 5: The steady state Lyapunov functions

6 Steady State Analysis of CNN A/D Converters

In *Section* 5, we have analyzed the *transient* dynamic behavior of the 4-bit CNN A/D converter under *zero* initial conditions, and have verified our results by computer simulations. In this section, we focus on the the *steady state* behavior of the CNN A/D converter under *arbitrary* initial conditions. Again without loss of generality, we still take the 4-bit CNN A/D converter as an example.

For a given analog input value,v_u, any output digital number, N_{out}, corresponds to a value of the Lyapunov function of the CNN A/D converter. The correct analog-to-digital conversion is the digital number N^*_{out} which has the minimum Lyapunov value for the given v_u. This N^*_{out} therefore corresponds to the *global* minimum of the Lyapunov function. All the other N_{out} corresponds to a *local* minimum of the Lyapunov function.

In *Section* 5.1, we have shown that if the assumptions in (74), (85), and (91) were satisfied, then the Lyapunov function would settle down to its *global* minimum. To prove this, let us examine first the Lyapunov function at steady state, which has the form as

$$\begin{aligned} E &= (v_u - N_{out})^2 - \frac{3}{2}\sum_{i=0}^{3} 2^{2i} - v_u^2 \\ &= -2v_u N_{out} + N_{out}^2 - 127.5. \end{aligned} \tag{123}$$

The steady-state Lyapunov function for all positive N_{out} are shown in *Table* 5. Based on this table we can determine the region of v_u in which an N^*_{out} coincides with the *global* minimum of the Lyapunov function. Since the functions in *Table* 5 are linear functions of v_u, we can determine the boundary points of the regions from each adjacent pair of functions. The result is shown in *Table* 6.

N^*_{out}	v_u
1	$v_u \in (0, 2.0)$
3	$v_u \in (2.0, 4.0)$
5	$v_u \in (4.0, 6.0)$
7	$v_u \in (6.0, 8.0)$
9	$v_u \in (8.0, 10.0)$
11	$v_u \in (10.0, 12.0)$
13	$v_u \in (12.0, 14.0)$
15	$v_u \in (14.0, 16.0)$

Table 6: The range of v_u for which N_{out} corresponds to the *global* minimum of the Lyapunov function.

Comparing *Table* 6 and Figure 2(a), we know that an ideal A/D conversion, which would be achieved under assumptions (74), (85), and (91), has all N_{out} corresponding to the *global* minimum of the Lyapunov function of the circuit.

From our simulation results in Figure 3, we have observed that some N_{out} (e.g., $N_{out} = 9$) correspond not to the *global* but to a *local* minimum of the Lyapunov function.

What are the necessary conditions for an N_{out} to be a *local* minimum of the Lyapunov function for a given input analog value v_u?

To answer this question, let us consider a particular case. For example, suppose $N_{out} = 3$; namely, $v_{sy0} = 1$, $v_{sy1} = -1$, $v_{sy2} = -1$ and $v_{sy3} = 1$, corresponding to a steady state of the circuit described by (44) – (47). Since the circuit is in the steady state, it follows from (5) that

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 4 & 0 & 0 \\ 0 & 0 & 16 & 0 \\ 0 & 0 & 0 & 64 \end{pmatrix} \begin{pmatrix} v_{x0}(\infty) \\ v_{x1}(\infty) \\ v_{x2}(\infty) \\ v_{x3}(\infty) \end{pmatrix}$$

$$= \begin{pmatrix} 2 & -4 & -8 & -16 \\ -4 & 8 & -16 & -32 \\ -8 & -16 & 32 & -64 \\ -16 & -32 & -64 & 128 \end{pmatrix} \begin{pmatrix} v_{sy0} \\ v_{sy1} \\ v_{sy2} \\ v_{sy3} \end{pmatrix} + \begin{pmatrix} 2 \\ 4 \\ 8 \\ 16 \end{pmatrix} v_u \qquad (124)$$

Substituting the above v_{syi} and using (6), we obtain the following inequalities

$$2 + 4 + 8 - 16 + 2v_u \geq 1, \qquad (125)$$

$$-4 - 8 + 16 - 32 + 4v_u \leq -4, \qquad (126)$$

$$-8 + 16 - 32 - 64 + 8v_u \leq -16, \qquad (127)$$

N_{out}	$v_u \in [-16, 16]$
1	$v_u \in [-16, 2.5]$
3	$v_u \in [1.5, 6.0]$
5	$v_u \in [2.0, 6.5]$
7	$v_u \in [5.5, 13.0]$
9	$v_u \in [3.0, 10.5]$
11	$v_u \in [9.5, 14.0]$
13	$v_u \in [10.0, 14.5]$
15	$v_u \in [13.5, 16.0]$

Table 7: The necessary condition region for N_{out} to be a local minimum of the Lyapunov function

$$-16 + 32 + 64 + 128 + 16v_u \geq 64. \tag{128}$$

Solving these inequalities, we obtain the following necessary condition of v_u for $N_{out} = 3$ to be a *local* minimum of the Lyapunov function:

$$1.5 \leq v_u \leq 6. \tag{129}$$

Following the same procedure, we can determine the necessary conditions for all other positive N_{out}, as shown in *Table* 7. Figure 5 displays the N_{out} and v_u relationship of *Table* 7. Observe that for any given analog value v_u, we can have 2 or 3 digital representations, as illustrated by the values v_{ua} and v_{ub} in Figure 5. This means that without the zero initial state condition (16), a CNN A/D converter will settle down to a local minimum of the Lyapunov function such that the A/D conversion error maybe as large as 6.0 (i.e. if we choose $N_{out} = 7$ and $v_u = 13.0$), which is 3 times larger than the digital representation value of the least significant bit.

The above problem is called *hysteresis* in the neural network A/D converter literatures [5, 8, 7]. In the implementations of neural network A/D converters, some researchers have used digital and analog mixed control methods to overcome this hysteresis problem. They attempt to use various techniques to control the A/D converter such that the conversion is accomplished one bit after another, starting from the most significant bit.

We have already pointed out in the previous section that a CNN A/D converter has a *serial* computing property. In the next section, we will exploit this property to minimize the conversion error.

Before going into the next section, let us try to extract some more information from Figure 5. Observe that Figure 5 exhibits some regular patterns. First, notice that the horizontal segments, which correspond to the region of v_u where an N_{out} exists, have some symmetry. If we fold the picture first along the

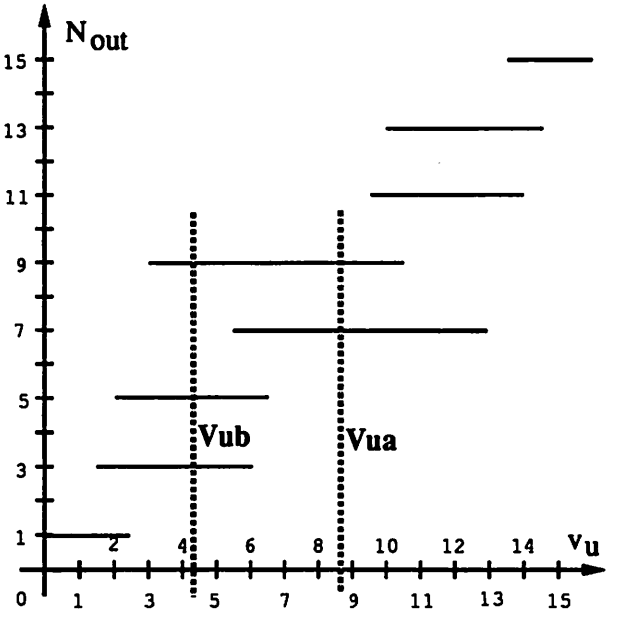


Figure 5: The necessary condition region for N_{out} to be a local minimum of the Lyapunov function

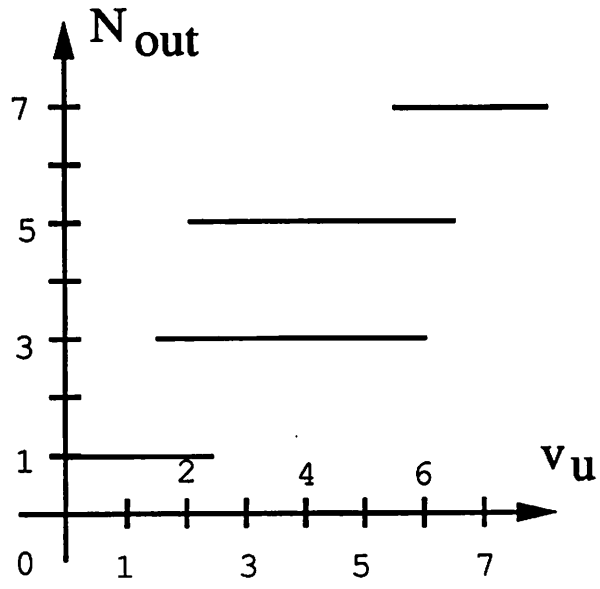


Figure 6: The necessary condition region for N_{out} to be a local minimum of the Lyapunov function (3-bit)

vertical line $v_u = 8$, and then along the horizontal line $N_{out} = 8$, we would see that all segments will overlap exactly. We can interpret this folded picture to be the same as that of a 3-bit CNN A/D converter. Figure 6 shows the picture corresponding to a 3-bit CNN A/D converter with the same circuit parameters, which confirms the above interpretation. If we fold the picture in Figure 6 again along $v_u = 4$ and $N_{out} = 4$, we would have a picture which is the same as that of a 2-bit CNN A/D converter with the same circuit parameters. This property is consistent with our analysis result in the preceding section, namely; if the most significant bit of an N-bit A/D converter is determined, then the conversion of the remaining bits will function similarly as an (n-1) bit A/D Converter. Secondly, noticed that the segments in Figure 5 have different lengths. This means that the maximum conversion errors are different for v_u in different regions. The segments for $N_{out} = 7$ and $N_{out} = 9$ are the longest, whereas those for $N_{out} = 1$ and $N_{out} = 15$ are the shortest. Can we design a CNN A/D converter such that the necessary condition regions for an N_{out} to be a local minimum of the Lyapunov function are the same? The answer is yes. In the next section, we will optimize our CNN A/D converter to force the segments in Figure 5 to have equal lengths.

7 Optimization of CNN A/D Converters

Based on the analysis and simulations in the previous sections, we have obtained a deep understanding of the dynamical and steady state behavior of the CNN A/D converters. In this section, we will present a design method which can be applied to a general CNN A/D converter without the *relaxed* condition (16). In our design, we will provide an algorithm which guarantees our CNN A/D converter will operate with any *prescribed maximum conversion error* ε_{max}, where $\varepsilon_{max} > \varepsilon_{ideal}$.

Similar to the procedure given in *Section* 3, we start our design by constructing an appropriate cost function for the optimization problem for analog-to-digital conversion.

Without loss of generality, we still continue to use a 4-bit CNN A/D converter as an example. We will summarize this design method in the forms of theorems for an N-bit CNN A/D converter at the end of this section.

Suppose that we just want to determine the most significant bit; namely, bit-3 in the conversion. In this case, we would include

$$(v_u - 2^3 v_{y3}(t))^2 \tag{130}$$

as a part of the cost function.

For the same reason, if we only want to determine the first two most significant bits, then we would include

$$(v_u - 2^3 v_{y3}(t) - 2^2 v_{y2}(t))^2 \tag{131}$$

into the cost function.

Following the same kind of reasoning, we define the cost function as follows

$$f(t) = \sum_{k=0}^{3} \lambda_k \left(v_u - \sum_{i=k}^{3} 2^i v_{yi}(t) \right)^2 - \sum_{i=0}^{3} a_i v_{yi}^2(t), \tag{132}$$

where, $\lambda_i \geq 0$ and $a_i \geq 0$.

Comparing (132) with (29), we see that in the cost function of (132) we consider not only the 4-bit conversion, but also a 3-bit conversion for the first three most significant digits, a 2-bit conversion for the first two most significant digits, and a 1-bit conversion for the most significant digit. The parameter $\lambda_k \geq 0$ serves as the weighing coefficient, which will be determined in the design. Clearly, (29) is a special case of (132) for $\lambda_0 = 1$, $\lambda_1 = 0$, $\lambda_2 = 0$ and $\lambda_3 = 0$.

If we choose $\lambda_0 = 0$, $\lambda_1 = 0$, $\lambda_2 = 0$ and $\lambda_3 = 1$, then we will have a cost function like (130), from which we can design a 1-bit CNN A/D converter. Furthermore, if we choose λ_i such that

$$\lambda_0 \ll \lambda_1 \ll \lambda_2 \ll \lambda_3, \tag{133}$$

then the most significant bit will be emphasized for correct conversion, and then followed, in decreasing emphasis, the next two most significant bits, and so on.

In order to compare the cost function (132) with the Lyapunov function (3), let us rewrite (132) into the following form

$$\begin{aligned} f(t) &= \sum_{k=0}^{3}\lambda_k\left(v_u^2 - 2v_u\sum_{i=k}^{3}2^i v_{yi}(t) + \sum_{i=k}^{3}\sum_{j=k}^{3}2^{i+j}v_{yi}(t)v_{yj}(t)\right) - \sum_{i=0}^{3}a_i v_{yi}^2(t) \\ &= v_u^2\sum_{k=0}^{3}\lambda_k - 2v_u\sum_{k=0}^{3}\lambda_k\sum_{i=k}^{3}2^i v_{yi}(t) + \sum_{k=0}^{3}\lambda_k\sum_{i=k}^{3}\sum_{j=k}^{3}2^{i+j}v_{yi}(t)v_{yj}(t) \\ &\quad - \sum_{i=0}^{3}a_i v_{yi}^2(t) \\ &= v_u^2\sum_{k=0}^{3}\lambda_k - v_u\sum_{i=0}^{3}2^{i+1}v_{yi}(t)\sum_{k=0}^{i}\lambda_k + \sum_{i=0}^{3}\sum_{j=0}^{3}2^{i+j}v_{yi}(t)v_{yj}(t)\sum_{k=0}^{\min(i,j)}\lambda_k \\ &\quad - \sum_{i=0}^{3}a_i v_{yi}^2(t). \end{aligned} \tag{134}$$

Comparing the corresponding terms in (134) and (3), we obtain:

$$b_i = 2^{i+1}\sum_{k=0}^{i}\lambda_k, \tag{135}$$

and

$$A_{ij} = -2^{i+j+1}\sum_{k=0}^{\min(i,j)}\lambda_k \tag{136}$$

for $i \neq j$.

We have some freedom in choosing the circuit parameters A_{ii} and R_i, as long as they satisfy the condition (4). Let us choose

$$R_i = 2^{-2i}R > 0 \tag{137}$$

and

$$A_{ii} = \frac{2}{R_i} = \frac{2^{2i+1}}{R}. \tag{138}$$

Using the same procedure as before, we can show that the results in *Table* 6 are still true for our latest 4-bit CNN A/D converter design with the cost function given by (132). In other words, for all A/D conversion results, if N_{out} correspond to the *global* minimum of the Lyapunov function (132), then the ideal conversion shown in Figure 2(a) will be achieved.

Now, let us derive the necessary conditions for an N_{out} to correspond to a local minimum of the Lyapunov function.

At steady state, we have

$$\frac{1}{R}\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 4 & 0 & 0 \\ 0 & 0 & 16 & 0 \\ 0 & 0 & 0 & 64 \end{pmatrix}\begin{pmatrix} v_{x0}(\infty) \\ v_{x1}(\infty) \\ v_{x2}(\infty) \\ v_{x3}(\infty) \end{pmatrix}$$

$$= \begin{pmatrix} 2/R & -4\lambda_0 & -8\lambda_0 & -16\lambda_0 \\ -4\lambda_0 & 8/R & -16(\lambda_0+\lambda_1) & -32(\lambda_0+\lambda_1) \\ -8\lambda_0 & -16(\lambda_0+\lambda_1) & 32/R & -64\sum_{k=0}^{2}\lambda_k \\ -16\lambda_0 & -32(\lambda_0+\lambda_1) & -64\sum_{k=0}^{2}\lambda_k & 128/R \end{pmatrix}\begin{pmatrix} v_{sy0} \\ v_{sy1} \\ v_{sy2} \\ v_{sy3} \end{pmatrix}$$

$$+ \begin{pmatrix} 2\lambda_0 \\ 4(\lambda_0+\lambda_1) \\ 8\sum_{k=0}^{2}\lambda_k \\ 16\sum_{k=0}^{3}\lambda_k \end{pmatrix} v_u \tag{139}$$

Since our goal is to minimize the A/D conversion error, let us rewrite (139) in order to introduce the *conversion error* ε as a variable by defining:

$$v_u = \sum_{i=0}^{3} 2^i v_{syi} - \varepsilon. \tag{140}$$

Substituting (140) into (139), we obtain the following equations

$$\frac{1}{R}v_{x0}(\infty) = 2\left(\frac{1}{R}+\lambda_0\right)v_{sy0} + 2\lambda_0\varepsilon, \tag{141}$$

$$\frac{4}{R}v_{x1}(\infty) = 4\lambda_1 v_{sy0} + 8\left(\frac{1}{R}+\sum_{k=0}^{1}\lambda_k\right)v_{sy1} + 4\sum_{k=0}^{1}\lambda_k\varepsilon, \tag{142}$$

$$\frac{16}{R}v_{x2}(\infty) = 8\sum_{k=1}^{2}\lambda_k v_{sy0} + 16\lambda_2 v_{sy1} + 32\left(\frac{1}{R}+\sum_{k=0}^{2}\lambda_k\right)v_{sy2} + 8\sum_{k=0}^{2}\lambda_k\varepsilon, \tag{143}$$

$$\begin{aligned}\frac{64}{R}v_{x3}(\infty) &= 16\sum_{k=1}^{3}\lambda_k v_{sy0} + 32\sum_{k=2}^{3}\lambda_k v_{sy1} + 64\lambda_3 v_{sy2} \\ &+ 128\left(\frac{1}{R}+\sum_{k=0}^{2}\lambda_k\right)v_{sy3} + 16\sum_{k=0}^{3}\lambda_k\varepsilon.\end{aligned} \tag{144}$$

In the following, we will show how to determine the parameters λ_i and R for *any prescribed* maximum A/D conversion error $\varepsilon_{max} > \varepsilon_{ideal} = 1$.

Let us first consider (141). Since v_{sy0} is a function of $v_{x0}(\infty)$, there is only one variable in (141). For $v_{sy0} = -1$, we will have $v_{x0}(\infty) \leq -1$, and hence

$$-\frac{1}{R} \geq -2(\frac{1}{R} + \lambda_0) + 2\lambda_0\varepsilon. \tag{145}$$

Therefore, we obtain the following upper-bound for ε :

$$\varepsilon \leq \frac{1 + 2R\lambda_0}{2R\lambda_0}. \tag{146}$$

For $v_{sy0} = 1$, we obtain the following lower-bound for ε :

$$\varepsilon \geq -\frac{1 + 2R\lambda_0}{2R\lambda_0}. \tag{147}$$

Therefore,

$$| \varepsilon | \leq \frac{1 + 2R\lambda_0}{2R\lambda_0}. \tag{148}$$

Let us consider next (142). Since there are two output variables in (142), we have four choices for the different combinations of v_{sy0} and v_{sy1}. We have found from the derivation of (146) and (147) that if v_{sy0} and v_{sy1} have the same value, then the conversion errors computed from (141) and (142) will be bounded in the same direction. For example, if $v_{sy0} = v_{sy1} = -1$ then ε will have two upper-bounds derived from (141) and (142). In this case, we should choose the minimum upper-bound as the upper-bound of ε. Unfortunately, we do not know which upper-bound is the minimum at this moment. Nevertheless, we have developed an algorithm to deal with this problem. We will provide a theorem later to justify the validity of our algorithm.

Algorithm 1

(1) Solve the equation involving the state variable v_{x0}, (141) in this case, for the upper-bound of $| \varepsilon |$ using $v_{sy0} = \pm 1$;

(2) Solve the equation involving the state variable v_{xi} $(i > 0)$, (142) - (144) in this case, for the upper-bound of $| \varepsilon |$ using $v_{syi} = \pm 1$, and $v_{syj} = -v_{syi}$ for $j < i$.

Applying the above algorithm to (142), there are only two cases to consider; namely, ($v_{sy0} = 1$, $v_{sy1} = -1$) and ($v_{sy0} = -1$, $v_{sy1} = 1$). For the case of $v_{sy0} = 1$ and $v_{sy1} = -1$, we found the following upper-bound :

$$\begin{aligned} \varepsilon &\leq \frac{1 + 2R(\lambda_0 + \lambda_1) - R\lambda_1}{R(\lambda_0 + \lambda_1)} \\ &\leq \frac{1 + 2R\lambda_0 + R\lambda_1}{R(\lambda_0 + \lambda_1)}. \end{aligned} \tag{149}$$

For the case of $v_{sy0} = -1$ and $v_{sy1} = 1$, we found the following lower-bound :

$$\varepsilon \geq -\frac{1+2R\lambda_0+R\lambda_1}{R(\lambda_0+\lambda_1)} \tag{150}$$

Therefore,

$$|\varepsilon| \leq \frac{1+R(2\lambda_0+\lambda_1)}{R(\lambda_0+\lambda_1)} \tag{151}$$

For (143) and (144), we can obtain the following inequalities

$$|\varepsilon| \leq \frac{2+R(4\lambda_0+3\lambda_1+\lambda_2)}{R(\lambda_0+\lambda_1+\lambda_2)} \tag{152}$$

and

$$|\varepsilon| \leq \frac{4+R(8\lambda_0+7\lambda_1+5\lambda_2+\lambda_3)}{R(\lambda_0+\lambda_1+\lambda_2+\lambda_3)}, \tag{153}$$

respectively, by using the same procedure as that for (141) and (142).

Now, let us examine these inequalities. Suppose that the following assumption

$$1 \ll \lambda_0 \ll \lambda_1 \ll \lambda_2 \ll \lambda_3 \tag{154}$$

is true. Then, by estimating (148), (151), (152) and (153), we found that the bound of $|\varepsilon|$ is very close to 1, which is the ideal A/D conversion error ε_{ideal}. This means that by choosing a sufficiently large value for the parameter λ_i we can design a CNN A/D converter arbitrarily close to the ideal case.

Now, let us determine λ_i and R for a given maximum A/D conversion error ε_{max}. Let us set the right hand side of the inequalities (148), (151), (152) and (153) to be equal to ε_{max}:

$$\varepsilon_{max} = \frac{1+2R\lambda_0}{2R\lambda_0}; \tag{155}$$

$$\varepsilon_{max} = \frac{1+R(2\lambda_0+\lambda_1)}{R(\lambda_0+\lambda_1)}; \tag{156}$$

$$\varepsilon_{max} = \frac{2+R(4\lambda_0+3\lambda_1+\lambda_2)}{R(\lambda_0+\lambda_1+\lambda_2)}; \tag{157}$$

$$\varepsilon_{max} = \frac{4+R(8\lambda_0+7\lambda_1+5\lambda_2+\lambda_3)}{R(\lambda_0+\lambda_1+\lambda_2+\lambda_3)}. \tag{158}$$

Since we have only four equations involving five variables, one of them can be arbitrarily valued. For convenience, we choose R as a variable to be determined later. Then we can solve the equations iteratively as follows:

$$\lambda_0 = \frac{1}{2R(\varepsilon_{max}-1)}, \tag{159}$$

$$\lambda_1 = \frac{1 + R\lambda_0(2 - \varepsilon_{max})}{R(\varepsilon_{max} - 1)}, \tag{160}$$

$$\lambda_2 = \frac{2 + R\lambda_0(4 - \varepsilon_{max}) + R\lambda_1(3 - \varepsilon_{max})}{R(\varepsilon_{max} - 1)}, \tag{161}$$

$$\lambda_3 = \frac{4 + R\lambda_0(8 - \varepsilon_{max}) + R\lambda_1(7 - \varepsilon_{max}) + R\lambda_2(5 - \varepsilon_{max})}{R(\varepsilon_{max} - 1)}, \tag{162}$$

We can also recast the above λ_i as a function of R and ε_{max} as follows:

$$\lambda_0 = \frac{1}{2R(\varepsilon_{max} - 1)}, \tag{163}$$

$$\lambda_1 = \frac{\varepsilon_{max}}{2R(\varepsilon_{max} - 1)^2}, \tag{164}$$

$$\lambda_2 = \frac{\varepsilon_{max}^2}{R(\varepsilon_{max} - 1)^3}, \tag{165}$$

$$\lambda_3 = \frac{2\varepsilon_{max}^2(\varepsilon_{max} + 1)}{R(\varepsilon_{max} - 1)^4}. \tag{166}$$

In order to verify the above derivations, let us work out an example. Suppose we want to design a 4-bit CNN A/D converter having a maximum conversion error of 1.5; namely,

$$\varepsilon_{max} = 1.5. \tag{167}$$

Using (163) – (166) and choosing $R = 1$ for convenience, we obtain:

$$R = 1; \tag{168}$$

$$\lambda_0 = 1; \tag{169}$$

$$\lambda_1 = 3; \tag{170}$$

$$\lambda_2 = 18; \tag{171}$$

$$\lambda_3 = 180. \tag{172}$$

Substituting these parameters into (141) – (144), we can determine the necessary condition shown in Figure 7 for N_{out} to be a *local* minimum of the Lyapunov function. It should not be surprising that all horizontal segments in Figure 7 have equal lengths. This is one of those rare situations where it is possible to design a nonlinear dynamical system such that the basins of attraction of all local minima are *uniformly* distributed!

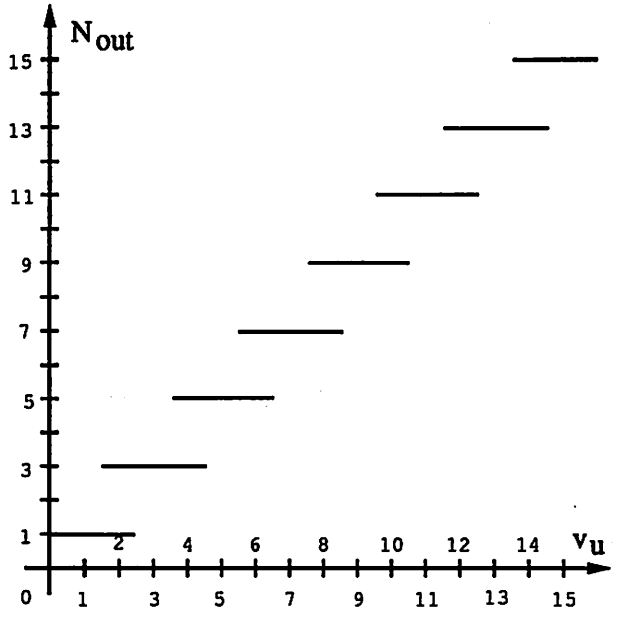


Figure 7: The necessary condition region for N_{out} to be a local minimum of the Lyapunov function (optimized)

Let us summarize the system equations describing the CNN A/D converter which we have just designed.

State equation:

$$\frac{d}{dt}\begin{pmatrix} v_{x0} \\ v_{x1} \\ v_{x2} \\ v_{x3} \end{pmatrix}(t) = -\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 4 & 0 & 0 \\ 0 & 0 & 16 & 0 \\ 0 & 0 & 0 & 64 \end{pmatrix}\begin{pmatrix} v_{x0} \\ v_{x1} \\ v_{x2} \\ v_{x3} \end{pmatrix}(t) + \\ + \begin{pmatrix} 2 & -4 & -8 & -16 \\ -4 & 8 & -64 & -128 \\ -8 & -64 & 32 & -1408 \\ -16 & -128 & -1408 & 128 \end{pmatrix}\begin{pmatrix} v_{y0} \\ v_{y1} \\ v_{y2} \\ v_{y3} \end{pmatrix}(t) + \\ + \begin{pmatrix} 2 \\ 16 \\ 176 \\ 3232 \end{pmatrix} v_u \tag{173}$$

Output equation:

$$v_{yi}(t) = 0.5\,(\mid v_{xi}(t) + 1 \mid - \mid v_{xi}(t) - 1 \mid) \tag{174}$$

Input dynamical range:

$$\mid v_u \mid \leq 16. \tag{175}$$

Observe that there are no longer any initial conditions stipulated in the above system equations. Indeed, no matter what the initial condition is, this CNN A/D converter is guaranteed settle down to a steady state such that the conversion error is less than or equal to $\varepsilon_{max} = 1.5$.

We have simulated this optimized CNN A/D converter using PWLSPICE. If we increase the input analog value v_u from 0 to 16, we would obtain the sampled results shown in *Table* 8. If we decrease the input analog value v_u from 16 to 0, the corresponding results is shown in *Table* 9. Figure 8 depicts the results listed in *Table* 8 and *Table* 9. From Figure 8 we observe a *hysteresis* phenomenon which has been restricted into a region having a width of $2(\varepsilon_{max} - \varepsilon_{ideal}) = 1$. From the simulation results in Figure 8, we see that the maximum A/D conversion error is 1.5 (corresponding to $v_u = 4.5$ and $v_u = 1.5$, for example, in *Table* 8 and *Table* 9, respectively). This confirms the validity of *Algorithm* 1.

From physical point of view, the *hysteresis* phenomenon is due to the basins of attraction of the local minima of CNN A/D converters. In order to transfer from one local minimum point of the Lyapunov function to another local minimum point, the system needs extra energy, which is provided by v_u in our case, from the outside of the circuit. If the system state is not in its local minimum point at the initial time, then it will be relatively easier to settle down to the global minimum point.

V_u	N_{out}	V_u	N_{out}	V_u	N_{out}	V_u	N_{out}
0.1	1	4.1	3	8.1	7	12.1	11
0.2	1	4.2	3	8.2	7	12.2	11
0.3	1	4.3	3	8.3	7	12.3	11
0.4	1	4.4	3	8.4	7	12.4	11
0.5	1	4.5	3	8.5	7	12.5	11
0.6	1	4.6	5	8.6	9	12.6	13
0.7	1	4.7	5	8.7	9	12.7	13
0.8	1	4.8	5	8.8	9	12.8	13
0.9	1	4.9	5	8.9	9	12.9	13
1.0	1	5.0	5	9.0	9	13.0	13
1.1	1	5.1	5	9.1	9	13.1	13
1.2	1	5.2	5	9.2	9	13.2	13
1.3	1	4.3	5	9.3	9	13.3	13
1.4	1	5.4	5	9.4	9	13.4	13
1.5	1	5.5	5	9.5	9	13.5	13
1.6	1	5.6	5	9.6	9	13.6	13
1.7	1	5.7	5	9.7	9	13.7	13
1.8	1	5.8	5	9.8	9	13.8	13
1.9	1	5.9	5	9.9	9	13.9	13
2.0	1	6.0	5	10.0	9	14.0	13
2.1	1	6.1	5	10.1	9	14.1	13
2.2	1	6.2	5	10.2	9	14.2	13
2.3	1	6.3	5	10.3	9	14.3	13
2.4	1	6.4	5	10.4	9	14.4	13
2.5	1	6.5	5	10.5	9	14.5	13
2.6	3	6.6	7	10.6	11	14.6	15
2.7	3	6.7	7	10.7	11	14.7	15
2.8	3	6.8	7	10.8	11	14.8	15
2.9	3	6.9	7	10.9	11	14.9	15
3.0	3	7.0	7	11.0	11	15.0	15
3.1	3	7.1	7	11.1	11	15.1	15
3.2	3	7.2	7	11.2	11	15.2	15
3.3	3	7.3	7	11.3	11	15.3	15
3.4	3	7.4	7	11.4	11	15.4	15
3.5	3	7.5	7	11.5	11	15.5	15
3.6	3	7.6	7	11.6	11	15.6	15
3.7	3	7.7	7	11.7	11	15.7	15
3.8	3	7.8	7	11.8	11	15.8	15
3.9	3	7.9	7	11.9	11	15.9	15
4.0	3	8.0	7	12.0	11	16.0	15

Table 8: Input/output mapping list of the optimized CNN A/D converter when v_u increases from 0 to 16

V_u	N_{out}	V_u	N_{out}	V_u	N_{out}	V_u	N_{out}
0.1	1	4.1	5	8.1	9	12.1	13
0.2	1	4.2	5	8.2	9	12.2	13
0.3	1	4.3	5	8.3	9	12.3	13
0.4	1	4.4	5	8.4	9	12.4	13
0.5	1	4.5	5	8.5	9	12.5	13
0.6	1	4.6	5	8.6	9	12.6	13
0.7	1	4.7	5	8.7	9	12.7	13
0.8	1	4.8	5	8.8	9	12.8	13
0.9	1	4.9	5	8.9	9	12.9	13
1.0	1	5.0	5	9.0	9	13.0	13
1.1	1	5.1	5	9.1	9	13.1	13
1.2	1	5.2	5	9.2	9	13.2	13
1.3	1	4.3	5	9.3	9	13.3	13
1.4	1	5.4	5	9.4	9	13.4	13
1.5	3	5.5	7	9.5	11	13.5	15
1.6	3	5.6	7	9.6	11	13.6	15
1.7	3	5.7	7	9.7	11	13.7	15
1.8	3	5.8	7	9.8	11	13.8	15
1.9	3	5.9	7	9.9	11	13.9	15
2.0	3	6.0	7	10.0	11	14.0	15
2.1	3	6.1	7	10.1	11	14.1	15
2.2	3	6.2	7	10.2	11	14.2	15
2.3	3	6.3	7	10.3	11	14.3	15
2.4	3	6.4	7	10.4	11	14.4	15
2.5	3	6.5	7	10.5	11	14.5	15
2.6	3	6.6	7	10.6	11	14.6	15
2.7	3	6.7	7	10.7	11	14.7	15
2.8	3	6.8	7	10.8	11	14.8	15
2.9	3	6.9	7	10.9	11	14.9	15
3.0	3	7.0	7	11.0	11	15.0	15
3.1	3	7.1	7	11.1	11	15.1	15
3.2	3	7.2	7	11.2	11	15.2	15
3.3	3	7.3	7	11.3	11	15.3	15
3.4	3	7.4	7	11.4	11	15.4	15
3.5	5	7.5	9	11.5	13	15.5	15
3.6	5	7.6	9	11.6	13	15.6	15
3.7	5	7.7	9	11.7	13	15.7	15
3.8	5	7.8	9	11.8	13	15.8	15
3.9	5	7.9	9	11.9	13	15.9	15
4.0	5	8.0	9	12.0	13	16.0	15

Table 9: Input/output mapping list of the optimized CNN A/D converter when v_u decreases from 16 to 0

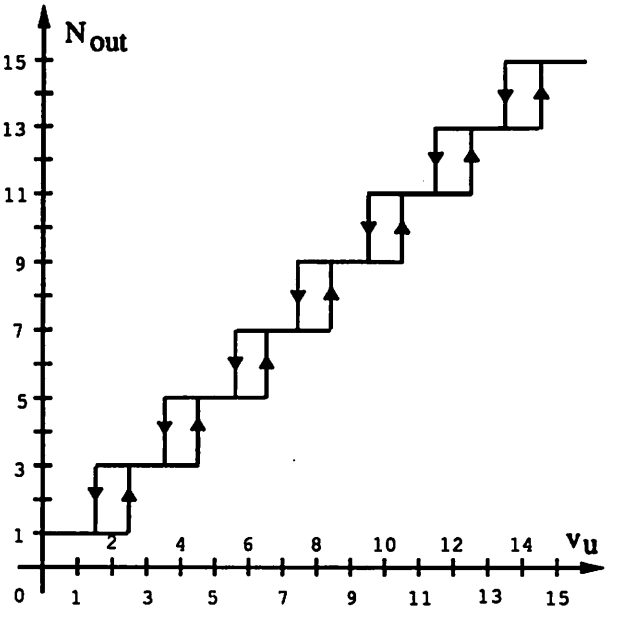


Figure 8: Input/output functions of the optimized CNN A/D converter

Table 10 and Figure 9 show the simulation results of the optimized CNN A/D converter under the *relaxed* condition. Here, the A/D conversion error is less than 1.1 (corresponding to $v_u = 2.0$ in *Table* 10. The improved conversion accuracy is due of course to the *relaxed* condition.

Let us now formalize our main results into the form of theorems. In the following, an N-bit CNN A/D converter means a circuit described by the system equations (1) and (2), and the digital representation (9). Also, when we say A/D conversion errors, we mean the errors based on the digital representation (9).

Theorem 1

For any prescribed maximum A/D conversion error $varepsilon_{max}$, where $1 < \varepsilon_{max} < 2$, and any integer N, there exists an N-bit CNN A/D converter with its conversion error absolutely bounded by ε_{max}.

Remark:

Note that for CNN A/D converters, the ideal and hence minimum A/D conversion error is 1, and that the linear A/D conversion requires the conversion error to be less than 2. We use a constructive method to prove this theorem.

Proof:

V_u	N_{out}	V_u	N_{out}	V_u	N_{out}	V_u	N_{out}
0.1	1	4.1	5	8.1	9	12.1	13
0.2	1	4.2	5	8.2	9	12.2	13
0.3	1	4.3	5	8.3	9	12.3	13
0.4	1	4.4	5	8.4	9	12.4	13
0.5	1	4.5	5	8.5	9	12.5	13
0.6	1	4.6	5	8.6	9	12.6	13
0.7	1	4.7	5	8.7	9	12.7	13
0.8	1	4.8	5	8.8	9	12.8	13
0.9	1	4.9	5	8.9	9	12.9	13
1.0	1	5.0	5	9.0	9	13.0	13
1.1	1	5.1	5	9.1	9	13.1	13
1.2	1	5.2	5	9.2	9	13.2	13
1.3	1	4.3	5	9.3	9	13.3	13
1.4	1	5.4	5	9.4	9	13.4	13
1.5	1	5.5	5	9.5	9	13.5	13
1.6	1	5.6	5	9.6	9	13.6	13
1.7	1	5.7	5	9.7	9	13.7	13
1.8	1	5.8	5	9.8	9	13.8	13
1.9	1	5.9	5	9.9	9	13.9	13
2.0	1	6.0	7	10.0	9	14.0	15
2.1	3	6.1	7	10.1	11	14.1	15
2.2	3	6.2	7	10.2	11	14.2	15
2.3	3	6.3	7	10.3	11	14.3	15
2.4	3	6.4	7	10.4	11	14.4	15
2.5	3	6.5	7	10.5	11	14.5	15
2.6	3	6.6	7	10.6	11	14.6	15
2.7	3	6.7	7	10.7	11	14.7	15
2.8	3	6.8	7	10.8	11	14.8	15
2.9	3	6.9	7	10.9	11	14.9	15
3.0	3	7.0	7	11.0	11	15.0	15
3.1	3	7.1	7	11.1	11	15.1	15
3.2	3	7.2	7	11.2	11	15.2	15
3.3	3	7.3	7	11.3	11	15.3	15
3.4	3	7.4	7	11.4	11	15.4	15
3.5	3	7.5	7	11.5	11	15.5	15
3.6	3	7.6	7	11.6	11	15.6	15
3.7	3	7.7	7	11.7	11	15.7	15
3.8	3	7.8	7	11.8	11	15.8	15
3.9	3	7.9	7	11.9	11	15.9	15
4.0	3	8.0	9	12.0	13	16.0	15

Table 10: Input/output mapping list of the relaxed optimized CNN A/D converter

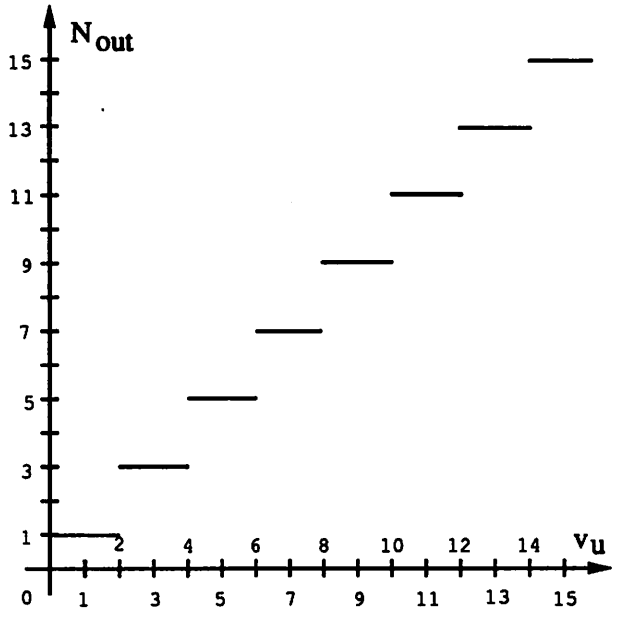


Figure 9: Input/output functions of the relaxed optimized CNN A/D converter

Our proof is a direct generalization of the 4-bit CNN A/D converter example. Let us choose the following circuit parameters for (1) and (2) :

$$A_{ii} = 2^{2i+1} \tag{176}$$

$$A_{ij} = -2^{i+j+1} \sum_{k=0}^{\min(i,j)} \lambda_k \qquad for \qquad i \neq j \tag{177}$$

$$b_i = 2^{i+1} \sum_{k=0}^{i} \lambda_k \tag{178}$$

$$R_i = 2^{-2i} \tag{179}$$

$$C_i = C = 1 \tag{180}$$

$$v_c = 1 \tag{181}$$

where $0 \leq i,\ j < N$, and $\lambda_k \geq 0$, which will be determined below.

Substituting the above parameters into (1) and (2), we obtain the following state equation :

$$\frac{dv_{xi}(t)}{dt} = - 2^{2i}v_{xi}(t) + 2^{2i+1}v_{yi}(t) - \sum_{\substack{j=0 \\ i \neq j}}^{N-1} 2^{i+j+1} \sum_{k=0}^{\min(i,j)} \lambda_k v_{yj}(t)$$

$$+ 2^{i+1} \sum_{k=0}^{i} \lambda_k v_u \tag{182}$$

where $0 \leq i < N$. In the steady state, we have

$$2^{2i}v_{sxi} = 2^{2i+1}v_{syi} - \sum_{\substack{j=0 \\ i \neq j}}^{N-1} 2^{i+j+1} \sum_{k=0}^{\min(i,j)} \lambda_k v_{syj} + 2^{i+1} \sum_{k=0}^{i} \lambda_k v_u \tag{183}$$

where $v_{sxi} \equiv v_{xi}(\infty)$, $v_{syi} \equiv v_{yi}(\infty) = \pm 1$, and $0 \leq i < N$. Substituting

$$v_u = \sum_{j=0}^{N-1} 2^j v_{syj} + \varepsilon \tag{184}$$

into (183) and rearranging terms, we obtain

$$2^i v_{sxi} = 2^{i+1}v_{syi} + 2^{i+1}v_{syi} \sum_{k=0}^{i} \lambda_k + \sum_{j=0}^{i-1} 2^{j+1} v_{syj} \sum_{k=j+1}^{i} \lambda_k + 2\varepsilon \sum_{k=0}^{i} \lambda_k, \tag{185}$$

where $0 \leq i < N$. Equation (185) is of crucial importance. First, it tells us that in the ith equation, only v_{syj} for $j \leq i$ and λ_k for $k \leq i$ are present. This means that we can solve the equations iteratively. Secondly, it shows us that for any N-bit CNN A/D converter, whose N is any integer, the equation corresponding to the ith bit, $i < N$, has the same form. This allows us to use an inductive method in the following proof.

For the case i = 0, we obtain

$$| \varepsilon | \leq \frac{1 + 2\lambda_0}{2\lambda_0}. \tag{186}$$

by following the same procedure as that for (148).

Defining

$$\varepsilon_{max} = \frac{1 + 2\lambda_0}{2\lambda_0}, \tag{187}$$

and solving for λ_0 we obtain

$$\lambda_0 = \frac{1}{2(\varepsilon_{max} - 1)} \tag{188}$$

which satisfies the condition $\lambda_0 > 0$ because $\varepsilon_{max} > 1$.

Suppose that we have determined $\lambda_j > 0$ for $j < i$. We must show how to calculate λ_i; where $i = j + 1$ and $i < N$. Using *Algorithm* 1, we obtain the general inequality :

$$|\varepsilon| \leq \frac{2^i + 2^{i+1}\sum_{k=0}^{i}\lambda_k - \sum_{j=0}^{i-1} 2^{j+1}\sum_{k=j+1}^{i}\lambda_k}{2\sum_{k=0}^{i}\lambda_k}. \tag{189}$$

Again, defining

$$\varepsilon_{max} = \frac{2^i + 2^{i+1}\sum_{k=0}^{i}\lambda_k - \sum_{j=0}^{i-1} 2^{j+1}\sum_{k=j+1}^{i}\lambda_k}{2\sum_{k=0}^{i}\lambda_k}. \tag{190}$$

and solving(190) for λ_i, we obtain

$$\lambda_i = \frac{2^{i-1} + 2^i\sum_{k=0}^{i-1}\lambda_k - \sum_{j=0}^{i-1} 2^j\sum_{k=j+1}^{i-1}\lambda_k - \varepsilon_{max}\sum_{k=0}^{i-1}\lambda_k}{\varepsilon_{max} - 1}. \tag{191}$$

Since

$$\begin{aligned}
\sum_{j=0}^{i-1} 2^j \sum_{k=j+1}^{i-1} \lambda_k &= \sum_{n=0}^{i-1}\sum_{k=n}^{i-1} 2^{n-1}\lambda_k - \sum_{n=0}^{i-1} 2^{-1}\lambda_k \\
&= \sum_{k=0}^{i-1}\sum_{n=0}^{k} 2^{n-1}\lambda_k - \sum_{n=0}^{i-1} 2^{-1}\lambda_k \\
&= \sum_{k=0}^{i-1}(2^k - 1)\lambda_k,
\end{aligned} \tag{192}$$

we have

$$\lambda_i = \frac{2^{i-1} + \sum_{k=0}^{i-1}\lambda_k\left(2^i - 2^k + 1 - \varepsilon_{max}\right)}{\varepsilon_{max} - 1}. \tag{193}$$

Because $\lambda_k > 0$ for $k < i$ and $1 < \varepsilon_{max} < 2$, we have

$$\lambda_i > 0, \tag{194}$$

where $i = j + 1$. It follows from our induction hypothesis that $\lambda_i > 0$ for all $i < N$. This completes the proof. □

Theorem 2

Algorithm 1 guarantees that the minimum upper-bound of $|\varepsilon|$ can be chosen.

Proof:
From (185), we have

$$| \varepsilon | \leq \frac{2^i + 2^{i+1} \sum_{k=0}^{i} \lambda_k - \sum_{j=0}^{i-1} 2^{j+1} v_{syj} \sum_{k=j+1}^{i} \lambda_k}{2 \sum_{k=0}^{i} \lambda_k}. \tag{195}$$

Here we do not apply any assumption on v_{syj} for $j < i$. The right hand side of (192) is obviously larger than or equals to ε_{max}, which is defined in *Algorithm* 1; namely,

$$\varepsilon_{max} = \frac{2^i + 2^{i+1} \sum_{k=0}^{i} \lambda_k - \sum_{j=0}^{i-1} 2^{j+1} \mid v_{syj} \mid \sum_{k=j+1}^{i} \lambda_k}{2 \sum_{k=0}^{i} \lambda_k}. \tag{196}$$

Since the above statement is true for all $0 \leq i < N$, the proof follows. □

Corollary 1
The preceding CNN A/D converter design is *optimal* in the sense that all the necessary condition regions of v_u, in which N_{out} is a *local* minimum of the Lyapunov function of the circuit, have equal lengths.

Although our CNN A/D converter design is optimized to achieve the minimum A/D conversion error, it is not optimized for implementation. As we have noticed that there are some freedom in choosing the circuit parameter R, C_i, and v_c. This kind of freedom allows us to transform the above CNN A/D converter into an *equivalent* circuit so that it is not only optimal for minimizing the conversion error, but also optimal for implementation. We will present the implementation issues of CNN A/D converters in a future paper.

8 Concluding Remarks

In this paper, we have presented an analog-to-digital conversion architecture using a cellular neural network. We have developed a systematic method to design CNN A/D converters. We have provided an algorithm to optimize the system performance. We have proved that we can design a CNN A/D converter for any prescribed and feasible A/D conversion error and any number of bits. We have verified our theoretical results using simple examples and computer simulations.

One of the main contributions of this paper is the uncovering of the intrinsic *serial* computing mechanism in this *parallel* computation architecture. *Theorem* 1 is another significant contribution. It is so general and practical that most of the important theoretical issues concerning CNN A/D conversion is solved. Furthermore, the same method may be used to design other neural network A/D converters, such as Hopfield's.

9 Figure Captions

Figure 1 : Cell circuit of CNN A/D converters.

Figure 2 : The A/D transfer functions. (a) for our CNN A/D converter; (b) for Tank and Hopfield's A/D converter.

Figure 3 : Simulation results for the CNN A/D converter.

Figure 3 : Simulation results for the CNN A/D converter.(Cont.)

Figure 4 : Input/output functions of the A/D converter.

Figure 5 : The necessary condition region for N_{out} to be a local minimum of the Lyapunov function.

Figure 6 : The necessary condition region for N_{out} to be a local minimum of the Lyapunov function (3-bit).

Figure 7 : The necessary condition region for N_{out} to be a local minimum of the Lyapunov function (optimized).

Figure 8 : Input/output functions of the optimized CNN A/D converter.

Figure 9 : Input/output functions of the relaxed optimized CNN A/D converter.

References

[1] L. O. Chua and Y. Liao. PWLSPICE: SPICE for piecewise-linear circuits. in preparation.

[2] L. O. Chua and L. Yang. Cellular neural networks: Applications. *IEEE Transactions on Circuits and Systems*, 35(10):1273–1290, Oct. 1988.

[3] L. O. Chua and L. Yang. Cellular neural networks: Theory. *IEEE Transactions on Circuits and Systems*, 35(10):1257–1272, Oct. 1988.

[4] M. W. Hirsch and S. Smale. *Differential equations, Dynamical systems, and Linear algebra.* Academic Press, New York, 1974.

[5] J. J. Hopfield. Neurons with graded response have collective computational properties like those of two-state neurons. *Proc. Natl. Acad. Sci. USA*, 81:3088–3092, 1984.

[6] T. Kailath. *Linear Systems.* Prentice-Hill, Englewood Cliffs, NJ, 1980.

[7] M. J. S. Smith and C. L. Portmann. Practical design and analysis of a simple 'neural' optimization circuit. *IEEE Transactions on Circuits and Systems*, 36(1):42–50, Jan. 1989.

[8] D. W. Tank and J. J. Hopfield. Simple 'neuron' optimization networks: An a/d converter, signal decision circuit, and a linear programming circuit. *IEEE Transactions on Circuits and Systems*, 33(5):533–541, May 1986.