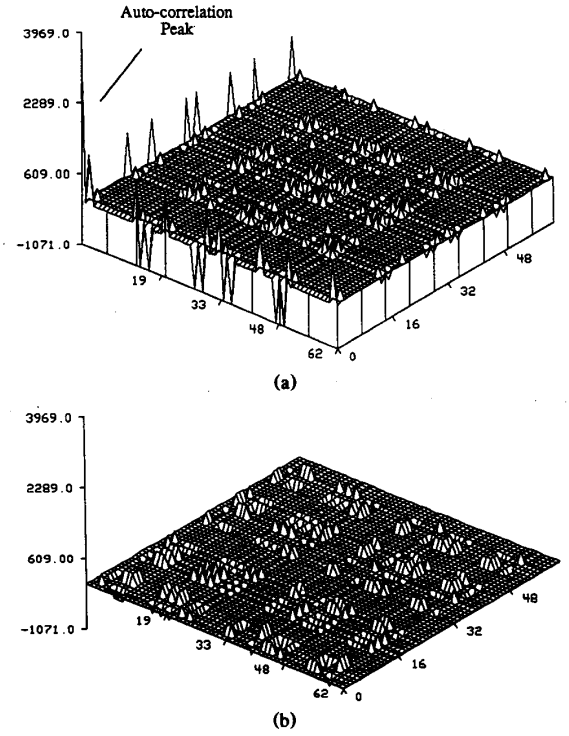

Fig. 2. Cyclic correlations of two 63×63 Gold code arrays. (a) Cyclic auto-correlation. (b) Cyclic cross-correlation.

N-dimensional quasi m-hypercubes and Gold code hypercubes can also be generated by the same construction method proposed in this correspondence. Quasi orthogonal property of these hypercubes is also expected. These n-dimensional hypercubes are much easier to be generated than the n-dimensional Welti codes [15].

Acknowledgment

The author acknowledges the valuable comments from anonymous reviewers.

References

[1] R. C. Dixon, *Spread Spectrum Systems*. New York: John Wiley, 1984, pp. 56–107.
[2] R. Skaug and J. F. Hjelmstad, *Spread Spectrum in Communication*. London: Peter Peregrinus, 1985, pp. 55–102.
[3] R. J. McEliece, *Finite Fields for Computer Scientists and Engineers*. Boston, MA: Kluwer Academic, 1987, pp. 123–200.
[4] J. G. Proakis, *Digital Communications*. New York: McGraw Hill, 1989, pp. 803–845.
[5] D. V. Sarwate and M. B. Pursley, "Cross-correlation properties of pseudorandom and related sequences: A review," *Proc. IEEE*, vol. 68, pp. 593–619, May 1980.
[6] T. Nomura, H. Miyakawa, H. Imai, and A. Fukuda, "A theory of two-dimensional linear recurring arrays," *IEEE Trans. Inform. Theory*, vol. IT-18, no. 6, pp. 775–785, Nov. 1972.
[7] F. J. MacWilliams and N. J. A. Sloane, "Pseudo-random sequences and arrays," *Proc. IEEE*, vol. 64, pp. 1715–1729, Dec. 1976.
[8] N. Ohyama, T. Honda, and J. Tsujiuchi, "An advanced coded imaging without sidelobes," *Optics Commun.*, vol. 27, pp. 339–344, Dec. 1978.
[9] W. Szepanski, "Compatibility problems in add-on data transmission for TV channels," *Proc. Second Symp. Electromag. Compat.*, pp. 263–268, June 1977.
[10] S. W. Golomb and H. Taylor, "Two-dimensional synchronization patterns for minimum ambiguity," *IEEE Trans. Inform. Theory*, vol. IT-28, no. 4, pp. 600–604, July 1982.
[11] C. J. Kuo and H. B. Rigaš, "Image multiplexing by code division technique," *SPIE Proc.*, vol. 1153, 1989.
[12] S. Lin and D. J. Costello, Jr., *Error Control Coding: Fundamentals and Applications*. Englewood Cliffs, NJ: Prentice-Hall, 1983, pp. 15–50.
[13] A. Rosenfeld and A. C. Kak, *Digital Picture Processing*, vol. 2. New York: Academic Press, 1982, pp. 75–82.
[14] B. Julesz and I. R. Bergen, "Textons, the fundamental elements in preattentive vision and perception of textures," *Bell Syst. Tech. J.*, vol. 62, pp. 1619–1645, July–Aug. 1983.
[15] H. D. Luke, "Sets of one and higher dimensional Welti codes and complementary codes," *IEEE Trans. Aerospace Electron Syst.*, vol. AES-21, pp. 170–179, Mar. 1985.

On the Maximum Tolerable Noise for Reliable Computation by Formulas

Bruce Hajek, *Fellow, IEEE,* and Timothy Weller, *Student Member, IEEE*

***Abstract*—It is shown that if formulas constructed from error-prone 3-input gates are used to compute Boolean functions, then a per-gate failure probability of 1/6 or more cannot be tolerated. The result is shown to be tight if the per-gate failure probability is constant and precisely known.**

***Index Terms*—Computation by unreliable components, reliable computing.**

I. Introduction

In a celebrated paper [6], von Neumann showed that arbitrary Boolean functions can be reliably computed by formulas built from noisy gates. For $\epsilon \geq 0$, the output of an ϵ-noisy, k-input gate is modeled as the result of applying a function $f: \{0,1\}^k \rightsquigarrow \{0,1\}$ to the inputs and then changing the value from 0 to 1 or vice versa with probability ϵ. If the output is changed we say the gate fails.

A *formula* is an interconnection of gates such that the output of each gate (except for one, which produces the output of the formula) is an input of one other gate. The inputs of the gates consist of Boolean variables that are the inputs for the formula, Boolean constants, or outputs of other gates. There are no loops in the sequence of interconnections. If a formula is composed of noisy gates, we assume that the gates fail independently of each other and of the inputs. Suppose we wish to evaluate a given deterministic Boolean function F of n variables, $F: \{0,1\}^n \rightsquigarrow \{0,1\}$, using a formula built from noisy gates. We allow each of the n Boolean arguments of F to be used, without error, as an input in multiple places in the formula. The maximum error probability of the formula is the maximum, over all 2^n possible

Manuscript received November 16, 1989; revised June 27, 1990. This material is based on work supported by a National Science Foundation Graduate Fellowship and the National Science Foundation under contract NSF ECS 83 52030.

The authors are with the Coordinate Science Laboratory, University of Illinois at Urbana-Champaign, 1101 W. Springfield Avenue, Urbana, Illinois 61801.

IEEE Log Number 9040741.

inputs, of the probability that the output of the formula differs from the corresponding value of the function F.

The goal of this paper is to identify precisely how large ϵ can be such that arbitrary Boolean functions can be computed with maximum error probability less than δ, for some δ depending only on ϵ with $\delta < 1/2$. In his paper, von Neumann [6, p. 69] used alternating stages of computation and error correction to show that reliable computation is possible for $\epsilon < 0.0073$. By "iterating the process of triplication at each step" as suggested by von Neumann, one finds that reliable computation is possible for $\epsilon < 0.09471$. In Section IV we show that reliable computation is in fact possible for $\epsilon < 1/6$. A caveat on these positive results is that the proofs use the assumption, implicit in our definition, that the ϵ-noisy gates fail with probability precisely ϵ—that is the gates are "reliably unreliable." Pippenger [3] showed that $\epsilon < 0.08415$ is sufficient even if the possibility that errors might not matter or cancel during a computation step is ignored, so $\epsilon < 0.08415$ is also sufficient under less restrictive error models, such as the one in [4].

On the negative side, Pippenger [5] showed that if $\epsilon \geq 1/3$ then such reliable computation is not possible. Moreover, for any ϵ in the range $0 < \epsilon < 1/3$, he obtained nontrivial lower bounds on the depth of formulas needed to compute certain functions with a given maximum error probability. Feder [1] improved Pippenger's bounds, and extended their domain of application to *networks* that compute Boolean functions. A network is similar to a formula except the output of any gate can be connected to the input of multiple gates. We prove (see Proposition 2) that if $\epsilon \geq 1/6$ then reliable computation is not possible, improving on the negative result of Pippenger. Although the negative results here and in [1], [5] are stated for reliably unreliable gates, they automatically apply for less restrictive failure models.

A survey paper of Pippenger [4] covers work related to several different aspects of von Neumann's paper.

II. Remembering and Computing with Noisy Gates—Negative Result

A problem related to von Neumann's result about computation is the problem of "remembering" a binary variable using noisy gates. Suppose X is a random variable with $P[X=0] = P[X=1] = 1/2$. Consider a formula built from ϵ-noisy gates such that each input is either X or a constant. For example, the formula might correspond to a balanced ternary tree with L-layers of ϵ-noisy 3-input majority logic gates and all 3^L inputs equal to X. For this example it is easy to show by induction that the output is equal to $1-X$ with probability $m_\epsilon^L(0)$, where m_ϵ^L is the L-fold composition of the function $m_\epsilon(a) = \epsilon + (1-2\epsilon)(3a^2 - 2a^3)$. As von Neumann noted, if $\epsilon < 1/6$ then $\lim_{L\to\infty} m_\epsilon^L(0) < 1/2$. Thus, for such ϵ, a single bit can be remembered with probability of error bounded away from $1/2$ for arbitrarily many layers by a formula composed of ϵ-noisy 3-input gates. On the other hand, if $\epsilon \geq 1/6$ then $\lim_{L\to\infty} m_\epsilon^L(0) = 1/2$. Thus, for such ϵ, it is not possible to so remember a bit by formulas built from ϵ-noisy 3-input majority logic gates. We will show that arbitrary ϵ-noisy 3-input gates can do no better, and that fact will imply a negative result about computation.

The joint distribution of X and an arbitrary binary random variable Y on the same probability space can be summarized by the parameter $\lambda^Y \in [0,1]^2$, where $\lambda^Y = (\lambda_0^Y, \lambda_1^Y)$ with $\lambda_i^Y = P[X \neq Y | X = i]$. For $0 \leq a \leq 1$ define $S(a)$ to be the closed, convex hull of $\{(a,a),(1,0),(0,1),(1-a,1-a)\}$. Note that X and Y are independent (i.e., $I(X;Y) = 0$) if and only if $\lambda^Y \in S(1/2)$, so we call $S(1/2)$ the *line of zero information*.

Consider once again a formula built from ϵ-noisy gates such that each input is either X or a constant. We can model a particular gate in the formula as shown in Fig. 1. The input

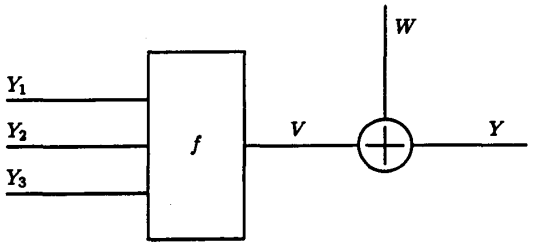

Fig. 1. Model of ϵ-noisy gate.

random variables Y_1, Y_2, Y_3 are conditionally independent given X. Setting $f(Y_1, Y_2, Y_3) = V$, we write the output of the noisy gate as $Y = V \oplus W$ where W is independent of X, Y_1, Y_2, Y_3 with $P[W=1] = \epsilon = 1 - P[W=0]$. Here f is an arbitrary 3-input Boolean function. For a given function f, λ^V is determined by $(\lambda^{Y_1}, \lambda^{Y_2}, \lambda^{Y_3})$. Furthermore, using vector addition,

$$\begin{aligned}\lambda^Y &= (1-\epsilon)\lambda^V + \epsilon(1-\lambda^V) \\ &= \epsilon + (1-2\epsilon)\lambda^V. \end{aligned} \tag{1}$$

The second form leads to the interpretation used by Feder [1], namely, an ϵ-noisy gate is equivalent to a gate that produces a random output with probability 2ϵ.

Proposition 1: Suppose $\epsilon \geq 1/6$ and $0 \leq a \leq 1/2$. If $\lambda^{Y_1}, \lambda^{Y_2}, \lambda^{Y_3} \in S(a)$ then $\lambda^Y \in S(m_\epsilon(a))$.

Proposition 1 will be proved in Section III.

Corollary 1 (Optimality of Majority Logic for Remembering a Bit): Suppose $\epsilon \geq 1/6$. Consider a formula built from ϵ-noisy 3-input gates such that all the inputs are constants or X, and such that X is not input into any gate less than L layers from the output Y_L. Then $P[Y_L \neq X] \geq m_\epsilon^L(0)$.

Proof: Note that $\lambda^X = (0,0) \in S(0)$, so Proposition 1 and induction on L imply that $\lambda^{Y_L} \in S(m_\epsilon^L(0))$. Thus $P[Y_L \neq X] = (\lambda_0^{Y_L} + \lambda_1^{Y_L})/2 \geq m_\epsilon^L(0)$. □

The main result of this correspondence is stated next.

Proposition 2: Let $\epsilon \geq 1/6$. Consider a Boolean function F of at least $1 + 3^{L-1}$ essential arguments, and consider any formula constructed of ϵ-noisy 3-input gates for computing F. Then the maximum error probability of the formula is greater than or equal to $m_\epsilon^L(0)$. (Recall that $\lim_{L\to\infty} m_\epsilon^L(0) = 1/2$).

Proof: At most 3^{L-1} variables can be inputs in the last $L-1$ layers of the formula, so at least one of the formula input variables, say the ith, is not an input for any of the gates in the last $L-1$ layers of the formula. We consider evaluating $F(c_1, c_2, \cdots, c_{i-1}, X, c_{i+1}, \cdots, c_n)$ using the formula where, as before, $P[X=0] = P[X=1] = 1/2$, and the binary constants $(c_j: j \neq i)$ are chosen so that $F(c_1, c_2, \cdots, c_{i-1}, X, c_{i+1}, \cdots, c_n)$ depends essentially on the value of X. By Corollary 1 the output of the formula is incorrect with probability at least $m_\epsilon^L(0)$. Thus, the probability the formula is incorrect for one of the two inputs $(c_1, c_2, \cdots, c_{i-1}, 0, c_{i+1}, \cdots, c_n)$ or $(c_1, c_2, \cdots, c_{i-1}, 1, c_{i+1}, \cdots, c_n)$ is also at least $m_\epsilon^L(0)$. □

III. Proof of Proposition 1

Proposition 1 will be proved after two lemmas are established.

Lemma 1: If Proposition 1 holds for the case where $\lambda^{Y_1} = \lambda^{Y_2} = \lambda^{Y_3} = (a,a)$, then it holds for any $(\lambda^{Y_1}, \lambda^{Y_2}, \lambda^{Y_3})$.

Proof: For any fixed noisy gate, the induced mapping of $(\lambda^{Y_1}, \lambda^{Y_2}, \lambda^{Y_3})$ to λ^Y is affine in each of $\lambda^{Y_1}, \lambda^{Y_2}, \lambda^{Y_3}$ when the other two are fixed. Hence the image of $S(a)^3 = S(a) \times S(a) \times$

$S(a)$ is the convex hull of the image of the set of extreme points of $S(a)^3$. The set $S(a)^3$ has 4^3 extreme points, where each extreme point is of the form $(\lambda^{Y_1}, \lambda^{Y_2}, \lambda^{Y_3})$ with $\lambda^{Y_i} \in \{(a,a),(0,1),(1,0),(1-a,1-a)\}$ for $1 \le i \le 3$. Therefore, it suffices to prove Proposition 1 for the case that $\lambda^{Y_i} \in \{(a,a),(0,1),(1,0),(1-a,1-a)\}$ for $1 \le i \le 3$. However, if $\lambda^{Y_i} \in \{(0,1),(1,0)\}$ then the same value of λ^Y can be obtained with $\lambda^{Y_i} = (a,a)$ by changing the function in the gate to one that does not depend on its ith input. Similarly, if $\lambda^{Y_i} = (1-a,1-a)$ then the same value of λ^Y can be obtained with $\lambda^{Y_i} = (a,a)$ by changing the function in the gate by complementing its ith input. The lemma follows. □

Define a 3-input Boolean function f to be a threshold function if for some integer k, f has the form $f(y_1,y_2,y_3) = I_{\{y_1+y_2+y_3 \ge k\}}$ or $f(y_1,y_2,y_3) = I_{\{y_1+y_2+y_3 \le k\}}$, where I_A denotes the indicator function of an event A.

Lemma 2: If Proposition 1 holds whenever the function f in the gate is a threshold function, then it holds in general.

Proof: We appeal to Lemma 1, and assume without loss of generality that $\lambda^{Y_1} = \lambda^{Y_2} = \lambda^{Y_3} = (a,a)$. We shall provide a self-contained proof although it is possible to deduce the Lemma from the Complete Class Theorem for Bayes decision rules [2]. Consider the output parameter $\lambda^Y = (\lambda_0^Y, \lambda_1^Y)$ as the function f varies, for a and ϵ fixed. Since $S(m_\epsilon(a))$ is a convex set, it suffices to prove that the convex hull of the set of λ^Y generated as f varies over the set of all 3-input Boolean functions is the same as the convex hull of the set of λ^Y generated as f varies only over the set of threshold functions. For that purpose, by the well-known separating hyperplane theorem [2], it suffices to prove that for any constants r and s the functional $C(f)$ defined by $C(f) = r\lambda_0^Y + s\lambda_1^Y$ achieves its minimum value at some threshold function f. Using (1) and the fact that $E[A] = E[E[A|B]]$ for random variables A and B yields that

$$\begin{aligned}
C(f) &= (r+s)\epsilon + (1-2\epsilon)\left(r\lambda_0^Y + s\lambda_1^Y\right) \\
&= (r+s)\epsilon + (1-2\epsilon)2\big(rE[V|X=0]P[X=0] \\
&\quad + sE[1-V|X=1]P[X=1]\big) \\
&= (r+s)\epsilon + (1-2\epsilon)2E\big[r(1-X)V + sX(1-V)\big] \\
&= (r+s)\epsilon + (1-2\epsilon)\big(s + 2E[V(r-(s+r)X)]\big) \\
&= (r+s)\epsilon + (1-2\epsilon) \\
&\quad \cdot \left(s + 2\sum_{y_1,y_2,y_3} U(y_1,y_2,y_3)p(y_1,y_2,y_3)\right),
\end{aligned}$$

where p is the joint probability mass function of (Y_1,Y_2,Y_3) and

$$\begin{aligned}
U(y_1,y_2,y_3) &= f(y_1,y_2,y_3) \\
&\quad \cdot \{r - (s+r)E[X|Y_1=y_1, Y_2=y_2, Y_3=y_3]\}. \quad (2)
\end{aligned}$$

Clearly the following function minimizes $C(f)$: $f(y_1,y_2,y_3) = 1$ precisely when the expression within braces in (2) is negative. The function f so defined is a threshold function since explicit computation readily shows that $E[X|Y_1=y_1, Y_2=y_2, Y_3=y_3]$ is a monotone function of the sum $y_1+y_2+y_3$. □

Proof of Proposition 1: By Lemmas 1 and 2, we assume without loss of generality that $\lambda^{Y_1} = \lambda^{Y_2} = \lambda^{Y_3} = (a,a)$ and that the function f is a threshold function. Furthermore, by symmetry, we need only consider the case where f has the form $f(y_1,y_2,y_3) = I_{\{y_1+y_2+y_3 \ge k\}}$ for some integer k. If $k \le 0$ or $k \ge 4$ then V is constant; so that $\lambda^Y \in S(1/2) \subset S(m_\epsilon(a))$. If $k = 2$ then f is the majority logic function; so that $\lambda^Y = (m_\epsilon(a), m_\epsilon(a)) \in S(m_\epsilon(a))$. Only the cases $k = 1$ and $k = 3$ remain, and by symmetry we need only consider one of these two cases. Hence, suppose $k = 3$ so that f is an AND function. We must show that $(\lambda_0^Y, \lambda_1^Y) \in S(m_\epsilon(a))$ where $\lambda_0^Y = \epsilon + (1-2\epsilon)a^3$ and $\lambda_1^Y = \epsilon + (1-2\epsilon)(a^3 - 3a^2 + 3a)$. We will show slightly more —namely that λ^Y is in the region bounded by the triangle with vertices $(m_\epsilon(a), m_\epsilon(a))$, $(0,1)$, and $(1/2,1/2)$. It is easy to verify that $\lambda_1^Y \ge \lambda_0^Y$ and that $\lambda_0^Y + \lambda_1^Y \le 1$, so it is enough to show that

$$\lambda_0^Y + m_\epsilon(a)\left(\lambda_1^Y - \lambda_0^Y\right) \ge m_\epsilon(a), \quad (3)$$

and it is sufficient to check the inequality for $\epsilon = 1/6$ because $m_\epsilon(a)$ is increasing in ϵ for $0 \le a \le 1/2$. Substituting $1/6$ for ϵ and expressing each variable in terms of a gives

$$\begin{aligned}
&\lambda_0^Y + m_{1/6}(a)\left(\lambda_1^Y - \lambda_0^Y\right) - m_{1/6}(a) \\
&\quad = \frac{1}{6} + \frac{2a^3}{3} + \frac{1}{6}\frac{2}{3}\left[1 + 4(3a^2 - 2a^3)\right](3a - 3a^2) \\
&\qquad - \frac{1}{6}\left[1 + 4(3a^2 - 2a^3)\right] \\
&\quad = \frac{8a^5 - 20a^4 + 18a^3 - 7a^2 + a}{3} = \frac{8\left(a - \frac{1}{2}\right)^3 (a-1)a}{3}.
\end{aligned}$$

The last expression is nonnegative for $0 \le a \le 1/2$, and Proposition 1 is proved. □

IV. Computing with (Precisely) ε-Noisy Gates—Positive Result

The following converse to Proposition 2 is proved in this section.

Proposition 3: Let $0 < \epsilon < 1/6$. There exists $\delta < 1/2$ such that any Boolean function can be computed by a formula built from ϵ-noisy 3-input gates with maximum error probability less than or equal to δ.

Fix ϵ with $0 < \epsilon < 1/6$ for the remainder of this section. Consider a formula corresponding to a balanced ternary tree with L layers of ϵ-noisy 3-input majority logic gates. Fix $x \in \{0,1\}$, let $U(1), \cdots, U(3^L)$ be independent (given x) inputs to the formula and let Y be the output. View the U's as noisy versions of x. If $P[U(i) \ne x] = r$ for all i, then $P[Y \ne x] = m_\epsilon^L(r)$, where m_ϵ^L was defined in Section II. In general, by induction on L, we see that $P[Y \ne x]$ is a monotone increasing function of $(P[U(i) \ne x]: 1 \le i \le 3^L)$ in the componentwise order. Thus, if for some $\delta < 1/2$ we have $P[U(i) \ne x] \in [0,\delta]$ for all i, then $P[Y \ne x] \in [m_\epsilon^L(0), m_\epsilon^L(\delta)]$. As L tends to infinity the interval $[m_\epsilon^L(0), m_\epsilon^L(\delta)]$ shrinks to the point η, where η is the unique point in $(0,1/2)$ with $\eta = m_\epsilon(\eta)$. Thus, after several stages of ϵ-noisy majority logic stages, the output error probability is close to η. This fact will be used in the proof of Proposition 3. The output error probability can be brought close to η, not just close-to-or-less-than η, because our gates are modeled as reliably unreliable as discussed in the Introduction.

Any Boolean function can be computed by a formula built using only noiseless 2-input NAND gates. Such gates can be obtained by ignoring entirely the third inputs of 3-input gates, but another idea in the proof of Proposition 3 is to use all three inputs to provide some error protection during a computation step. To this end, let XNAND be the 3-input Boolean function defined by XNAND = 1 for arguments (0,0,0), (1,0,0), (0,0,1) and (0,1,1), and XNAND = 0 otherwise. Consider computing NAND(x,y) where $x, y \in \{0,1\}$. Suppose that, for (x,y) fixed, X, Y_1 and Y_2 are independent Bernoulli random variables. Think of X as a noisy version of x, and Y_1 and Y_2 as noisy versions of y.

Lemma 3: There is a $\delta < 1/2$ and an open interval I with $\eta \in I \subset [0,1/2]$ so that the following is true. If $P[X \ne x]$, $P[Y_1 \ne y]$, $P[Y_2 \ne y] \in I$, and if Z is the output of an ϵ-noisy XNAND gate with input (X,Y_1,Y_2), then $P[Z \ne \text{NAND}(x,y)] < \delta$.

Proof: Using the fact that $\eta < 1/2$ and considering the four choices for (x, y) separately, the reader can readily check that $P[Z \neq \text{NAND}(x, y)] < 1/2$ if $P[X \neq x] = P[Y_1 \neq y] = P[Y_2 \neq y] = \eta$. The proof is finished using the fact that $P[Z \neq \text{NAND}(x, y)]$ is a continuous function of $(P[X \neq x], P[Y_1 \neq y], P[Y_2 \neq y])$ for each value of (x, y). □

Proof of Proposition 3: Use δ and I from Lemma 3, and choose L so large that $[m_\epsilon^L(0), m_\epsilon^L(\delta)] \subset I$. Given an arbitrary Boolean function, the network required for the proof is described as follows. Start with a formula for computing the function using only noiseless 2-input NAND gates. Replace the constituent NAND gates by XNAND gates one at a time, in order of nonincreasing depth from the final output, as follows.

Focus on the replacement of a particular NAND gate. Before replacement its inputs are provided by two smaller formulas, say formula A and formula B. Define an L-corrected version of a formula to be the new formula formed by feeding 3^L independent copies of the formula into a full L-layer formula of 3-input ϵ-noisy majority logic gates. The new ϵ-noisy XNAND gate needs three inputs. For the first, use the output of an L-corrected version of formula A. For the second and third, use the outputs of two independent L-corrected versions of formula B. The replacement of a particular NAND gate, and hence the complete modified formula, is specified.

Argument by induction, together with Lemma 3, yields the following: The output of any XNAND gate in the modified formula differs from the output of the corresponding NAND gate in the original formula, with probability at most δ. This is true in particular for the final output, and Proposition 3 is proved. □

V. Conclusion

Propositions 2 and 3 together establish that $\epsilon = 1/6$ is the critical value for computation by formulas of ϵ-noisy 3-input gates. Perhaps some sort of correlation inequality might yield an extension to networks. The positive result in [3] and Proposition 2 together establish that the critical value of ϵ for computation by formulas of "ϵ-noisy-or-better" 3-input gates lies somewhere in the interval $(0.08415, 1/6)$.

An interesting open problem is to close the gap between positive and negative results on the depth of reliable formulas for small ϵ. For ϵ close to zero, von Neumann [6] showed that computation by ϵ-noisy 3-input gates can be accomplished by adding to a network correction layers that account for a fraction of the total depth of the network that is asymptotic to $2/\log_3(1/\epsilon)$, whereas the results of Feder [1] show that for some functions the fraction must be at least $2\epsilon/\log_e 3$. For example, if $\epsilon = 0.001$, then we know that if roughly 31% of the stages are correction stages then reliable computation is possible, while from Feder's result we know that the additional stages due to error correction must be at least 0.2% of the whole. These bounds differ by more than a factor of 100. Unfortunately, it is not clear that our method can be used to close this gap.

Acknowledgment

The authors wish to thank an anonymous referee for Proposition 3.

References

[1] T. Feder, "Reliable computation by networks in the presence of noise," *IEEE Trans. Inform. Theory*, vol. 35, no. 3, pp. 569–572, May 1989.

[2] T. Ferguson, *Mathematical Statistics—A Decision Theoretic Approach*. New York: Academic Press, 1967.

[3] N. Pippenger, "Analysis of error correction by majority voting," in *Advances in Computing Research, Volume 5*, Silvio Micali, Ed. Greenwich, CT: JAI Press, 1989, pp. 171–198.

[4] ___, "Developments in the synthesis of reliable organisms from unreliable components," technical report RJ 6331 (62000) 7/11/88, IBM Watson Research Center, Distribution Services, P.O. Box 218, Yorktown Heights, NY 10598, 1988. To be published by the American Mathematical Society as part of a collection of papers regarding von Neumann's work.

[5] ___, "Reliable computation by formulas in the presence of noise," *IEEE Trans. Inform. Theory*, vol. 34, no. 2, pp. 194–197, Mar. 1988.

[6] J. von Neumann, "Probabalistic logics and the synthesis of reliable organisms from unreliable components," in *Automata Studies*, C. E. Shannon and J. McCarthy, Eds. Princeton, NJ: Princeton University Press, 1956, pp. 43–98.

On the Number of Points on Shells for Shifted Z^{4n} Lattices

Paul Fortier

***Abstract*—A conjecture is proven on the number of points on shells for the shifted Z^4 and the shifted Z^8 lattices. We also find an expression for the number of points on shells for any shifted Z^{4n} lattice.**

***Index Terms*—Integer lattice, theta series, multidimensional signal sets.**

I. Introduction

In [1, p. 107] Ruiz notes that not all shells of points on $Z^2 + (1/2, 1/2)$ have points on them. However, he conjectures that for the shifted Z^4 and Z^8 lattices, all shells have points on them. In this correspondence we prove, using arguments from number theory, that this conjecture is true. We also extend it to any shifted Z^{4n} lattice. This result has applications in the design of multidimensional signal sets.

II. Shells on the Shifted Z^n Lattice

In two dimensions, the squared radius of a shell of points is given by

$$r_k = 2(k-1) + 0.5, \qquad k = 1, 2, \cdots.$$

Thus the first shell has a squared radius of 0.5, the second shell has a squared radius of 2.5, and so on. Not all shells have points on them. For example, the sixth shell with squared radius 10.5 contains no points. In other words there are no pairs (x, y) of half integers such that $x^2 + y^2 = 10.5$.

In four dimensions the squared radius of a shell of points is given by

$$r_k = 2k - 1, \qquad k = 1, 2, \cdots, \tag{1}$$

and in eight dimensions it is given by

$$r_k = 2k, \qquad k = 1, 2, \cdots. \tag{2}$$

Ruiz [1, p. 107] has conjectured that all the shells in 4-D and 8-D contain points.

Manuscript received November 7, 1989; revised June 26, 1990. This work was supported in part by a Doctoral Fellowship from Laval University, Québec, Canada.

The author is with the Department of Electrical Engineering, Laval University, Québec, Canada G1K 7P4.

IEEE Log Number 9040909.