

Fluctuation and Noise Letters
© World Scientific Publishing Company

FAST MEASUREMENT OF HYPERSPACE VECTORS IN NOISE-BASED LOGIC

L.L. STACHÓ

*Bolyai Institute, Szeged University, Aradi Vértanúk tere 1
Szeged, H-6720, Hungary
stacho@math.u-szeged.hu*

Received (received date)

Revised (revised date)

We propose a method to determine single hyperspace vectors (product strings of noise-bits) by classical means with the same effectiveness as the results using time shifted noise-based logic. A system of binary linear equations based on the amplitudes of the hyperspace vector and the reference noise-bits is set up and solved after enough independent information is collected. The resulting error probability (the chance of getting no answer) has approximately an exponential decay with the time of measurement.

1. Introduction

Noise-based logic (NBL) [1-9], on the contrary of its name, is a deterministic logic scheme. The logic information is carried by independent stochastic processes, their products and their superpositions [1]. In non-squeezed NBL, $2N$ independent reference noises are needed to form N noise-bits: one for each L and another one for each H value and N -bit long products (product-strings) represent hyperspace vectors.

The identification of a single product-string (hyperspace vector) is necessary after a quantum-computing-type calculation with a simple output (such as the binary representation of a prime number) and it is one of the challenging questions. Recently, "time shifted noise-based logic" has been introduced [9], which is constructed by shifting each reference signal with a small time delay. This modification implies an exponential speedup of single hyperspace vector identification compared to the former case and it requires the same, $O(N)$ complexity as in quantum computing.

The goal of the present Letter is to show that the identification of such a single hyperspace vector can be done with similar complexity without the need of time shifted noise-based logic. Thus this task can be carried out within the original noise-based logic framework, which uses the same clock timing for each reference component.

2 *L.L. Stachó*

2. Problem setting

Let N denote the number of bits and for $k = 1, 2, \dots, N$ let

$$A_k, B_k : \{1, 2, \dots\} \rightarrow \{-1, 1\} \quad (1)$$

be random telegraph waves (RTW). As for the physical realization, each function A_k or B_k represents a sequence of electric impulses with intensity $A_k(t)$ (or $B_k(t)$) in every time interval $(t-1, t)$ between consecutive clock signals for $t = 1, 2, \dots$. We are given a binary string x_0, x_1, \dots, x_N consisting of terms 0 or 1 as usually which will be transformed into the *product wave*

$$W := X_1 X_2 \cdots X_N \quad \text{where} \quad X_k = A_k \text{ if } x_k = 1 \text{ and } X_k = B_k \text{ if } x_k = 0. \quad (2)$$

Problem 0. (Measurement problem). *Knowing only the product wave W , determine the 0-1 string x_1, \dots, x_N satisfying (2).*

Our purpose will be to develop an algorithm based on an infinite system of binary linear algebraic equations describing the relationships encoded in (2). Given the waves $W, A_1, B_1, A_2, B_2, \dots, A_N, B_N$, for $T = 1, 2, \dots$, our algorithm reads the impulses $W(T), A_1(T), B_1(T), \dots, A_N(T), B_N(T)$ and it determines if there is only a unique string x_1, \dots, x_N suitable for solving Problem 0 on the basis of the gathered data as far. Having found the first time

$$S = S(W, A_1, B_1, \dots, A_N, B_N) \quad (3)$$

with this uniqueness property, the algorithm provides the precise guess for the sequence x_0, x_1, \dots, x_N and then stops. We shall call S the *time requirement for complete measurement*. In Definition 3 below we formulate a precise algebraic construction for S . Notice that the possibility $S = \infty$ cannot be excluded as e.g. if $W(t) = A_k(t) = B_k(t) \equiv 1$. However, such algebraically extreme cases are practically improbable due to the random construction of the waves A_k, B_k . Actually, throughout the whole paper we regard the terms $A_k(t), B_k(t)$ ($k = 1, \dots, N; t \in \mathbf{Z}_+$) to be *completely independent random variables* such that

$$\Prb(A_k(t) = (-1)^s) = \Prb(B_k(t) = (-1)^s) = 1/2 \quad (s = 0, 1). \quad (4)$$

Definition 1. Introduce the below *transformed variables* resp. *product wave*:

$$C_k := A_k B_k \quad (k = 1, \dots, N), \quad U := W B_1 B_2 \cdots B_N. \quad (5)$$

Observation 1. *Due to the identities $A_k^2 = B_k^2 = 1$, we have $X_k = C_k^{x_k} B_k$, and hence $X_1 X_2 \cdots X_N = W \iff (X_1 B_1)(X_2 B_2) \cdots (X_N B_N) = U$ that is*

$$C_1^{x_1} C_2^{x_2} \cdots C_N^{x_N} = U. \quad (6)$$

Definition 2. Introduce the matrix $\mathbf{C} := [c_{tk}]_{t=1, k=1}^{\infty, N}$ respectively the column vector $\mathbf{u} := [u_t]_{t=1}^{\infty}$ as follows: we let

$$c_{tk} := [1 \text{ if } C_k(t) = -1, \quad 0 \text{ if } C_k(t) = 1], \quad (7)$$

$$u_t := [1 \text{ if } U(t) = -1, \quad 0 \text{ if } U(t) = 1]. \quad (8)$$

4 *L.L. Stachó*

To do so, we can proceed to find entries with indices $(t_1, k_1), (t_2, k_2), \dots, (t_N, k_N)$ in the following manner.

Algorithm.

Let $\mathbf{G}_0 := \mathbf{C}$ the matrix introduced in Definition 2, $K_0 := \{1, \dots, N\}$.

- Step 1) Let t_1 be the row index of the first nonzero row in \mathbf{G}_0 and let k_1 be the column index of the first non zero entry in this row. Then we set $K_1 := K_0 \setminus \{k_1\}$ and we let \mathbf{G}_1 to be the matrix obtained with elimination killing the subcolumn below the entry t_1, k_1 .
- \vdots
- Step d) Let t_d be the first row index $> t_{d-1}$ such that row t_d in \mathbf{G}_{d-1} does not vanish but all the rows of \mathbf{G}_{d-1} vanish between row t_{d-1} and t_d . Then let k_d be the minimal index in K_{d-1} such that the entry of \mathbf{G}_{d-1} with index (t_d, k_d) does not vanish. We define $K_d := K_{d-1} \setminus \{k_d\}$ and the matrix \mathbf{G}_d is obtained with elimination from \mathbf{G}_{d-1} by killing the subcolumn below the entry (t_d, k_d) .

Theorem 1. *Let $0 = t_0 < t_1 < t_2 < \dots < t_N$ be arbitrarily given integers and let $n_d := t_d - t_{d-1} - 1$ ($d = 1, \dots, N$). Then we have*

$$\mathbf{Prb}\left(\text{the Algorithm produces the row indices } t_1, \dots, t_N\right) = \prod_{d=0}^{N-1} \left[2^{n_d(N-d)} (1-2^{d-N}) \right].$$

Proof. $\mathbf{Prb}\left(\text{the Algorithm produces the row indices } t_1, \dots, t_N\right) =$

$$\begin{aligned} &= \left[\mathbf{Prb}\left(\text{the } Nn_0 \text{ entries of the first } n_0 \text{ rows vanish}\right) \times \right. \\ &\quad \times \left. \sum_{r=0}^{N-1} \mathbf{Prb}\left(r \text{ zeros stand before a term 1 in row } t_1\right) \right] \times \\ &\quad \times \left[\mathbf{Prb}\left(\text{the } (N-1)n_1 \text{ entries of the first } n_1 \text{ rows after the } t_1\text{th vanish}\right) \times \right. \\ &\quad \times \left. \sum_{r=0}^{N-2} \mathbf{Prb}\left(r \text{ zeros before a 1 in the part with column indices in } K_1 \text{ of row } t_2\right) \right] \times \\ &\quad \vdots \\ &\quad \times \left[\mathbf{Prb}\left(\text{the } (N-1)n_1 \text{ entries of the first } n_1 \text{ rows after the } t_1\text{th vanish}\right) \times \right. \\ &\quad \times \left. \sum_{r=0}^{N-2} \mathbf{Prb}\left(r \text{ zeros before a 1 in the part with column indices in } K_1 \text{ of row } t_2\right) \right]. \end{aligned}$$

Since the entries of \mathbf{C} assume the values 0-1 with probability $1/2$ independently, hence

$$\begin{aligned} & \Prb\left(\text{the Algorithm produces the row indices } t_1, \dots, t_N\right) = \\ &= \prod_{d=0}^{N-1} \left[2^{-n_d(N-d)} \left(2^{-1} + 2^{-2} + \dots + 2^{d-N} \right) \right] = \\ &= \prod_{d=0}^{N-1} \left[2^{-n_d(N-d)} \left(1 - 2^{d-N} \right) \right]. \end{aligned}$$

Corollary. For any $s \geq N$ we have $\pi_s \leq \binom{s}{N} 2^{N-s}$.

Proof.

We know that $\pi_s = \sum_{1 \leq t_1, \dots, t_N = s} \Prb\left(\text{the Algorithm produces the row indices } t_1, \dots, t_N\right)$.

Therefore

$$\begin{aligned} \pi_s &= \sum_{n_0 + \dots + n_{N-1} = s-N} \prod_{d=0}^{N-1} \left[2^{-n_d(N-d)} \left(1 - 2^{d-N} \right) \right] \leq \\ &\leq \sum_{n_0 + \dots + n_{N-1} = s-N} \exp_2 \left(- \sum_{d=0}^{N-1} n_d(N-d) \right) \leq \\ &\leq \sum_{n_0 + \dots + n_{N-1} = s-N} \exp_2 \left(- \sum_{d=0}^{N-1} n_d \right) = \binom{s}{N} 2^{N-s} \end{aligned}$$

due to the combinatorial fact that $\#\{(n_0, \dots, n_{N-1}) : n_0 + \dots + n_{N-1} = s-N\}$ coincides with the number of repeated combination of dividing $s-N$ elements into N parts.

Theorem 2. For $t \geq N$ we have $\bar{\pi}_t \leq 2^{N+1-t} \sum_{k=0}^N \binom{t}{k}$.

Proof. Observe that, in general, $\binom{s}{N} x^{N-s} = \frac{1}{N!} \frac{d^N}{dx^N} x^s$. Thus, by the Corollary,

$$\bar{\pi}_t = \sum_{s=t}^{\infty} \pi_s \leq \sum_{s=t}^{\infty} \binom{s}{N} \left(\frac{1}{2} \right)^{N-s} = \frac{1}{N!} \frac{d^N}{dx^N} x^t (1-x)^{-1} \Big|_{x=1/2}. \quad (11)$$

According to Leibniz' rule,

$$\begin{aligned} \frac{d^N}{dx^N} x^t (1-x)^{-1} &= \sum_{k=0}^N \binom{N}{k} \frac{d^k}{dx^k} x^t \frac{d^{N-k}}{dx^{N-k}} (1-x)^{-1} = \\ &= \sum_{k=0}^N \binom{N}{k} t(t-1) \dots (t-k+1) x^{t-k} (-1)^{N-k} (N-k)! (1-x)^{-(N-k+1)}. \end{aligned}$$

6 *L.L. Stachó*

Taking into account that $x^{t-k}(1-x)^{-(N-k+1)}|_{x=1/2} = 2^{k-t+N-k+1} = 2^{N+1-t}$, from (11) we get

$$\bar{\pi}_t \leq \frac{1}{N!} \sum_{k=0}^N \binom{N}{k} \frac{t!}{(t-k)!} (N-k)! 2^{N+1-t} = \sum_{k=0}^N \binom{t}{k} 2^{N+1-t}.$$

Corollary. For $t \geq N$ we have $\bar{\pi}_t \leq 2et^N/2^{t-N}$.

Proof. Putting the inequalities $\binom{t}{k} = t(t-1)\cdots(t-k+1)/k! \leq t^N/k!$ ($k \leq N$) in Theorem 2, we get $\bar{\pi}_t \leq 2t^N 2^{N-t} \sum_k k!^{-1}$.

We are now ready to answer Problem 2. Notice that as far we used an arbitrarily fixed number N of bits (and we suppressed the parameter N in the notation in the notation $\bar{\pi}_t = \bar{\pi}_t^{(N)}$). We are now interested in finding reasonably small function $N \mapsto t(N)$ such that $\bar{\pi}_t^{(N)}$ decreases exponentially to zero as $N \rightarrow \infty$, or which is the same, we have $\limsup_{N \rightarrow \infty} N^{-1} \log_2 \bar{\pi}_t^{(N)} < 0$ or even $= -\infty$.

Proposition 1. Given any small constant $\varepsilon > 0$, with the choice $t(N) := N(\log_2 N)^{1+\varepsilon}$ we have $\lim_{N \rightarrow \infty} N^{-1} \log_2 \bar{\pi}_{t(N)}^{(N)} = -\infty$.

Proof. In this case

$$\begin{aligned} \frac{1}{N} \log_2 \bar{\pi}_{t(N)}^{(N)} &\leq \frac{1}{N} \log_2 \left(2e \left[N(\log_2 N)^{1+\varepsilon} \right]^N 2^{N(\log_2 N)^{1+\varepsilon} - N} \right) = \\ &= \frac{1 + \log_2 e}{N} + 1 + (1 + \varepsilon) \log_2 \log_2 N + \log_2 N - (\log_2 N)^{1+\varepsilon} \rightarrow -\infty. \end{aligned}$$

Remark 3. In conclusion, whatever small $\varepsilon > 0$ we choose, the probability that the measurement algorithm fails in less than $t_\varepsilon = N(\log_2 N)^{1+\varepsilon}$ time steps is scaling as 2^{-N} in the limit of large N .

Acknowledgements

I would like to express my sincere gratitude to Prof. L. Kish for his valuable suggestions and the stimulating discussions on the topics. The author was supported partially by the project TAMOP-4.2.1/B-09/-1/KONV-2010-0005.

References

- [1] L. B. Kish, Noise-based logic: Binary, multi-valued, or fuzzy, with optional superposition of logic states, *Physics Letters A* **373** (2009) 911-918.
- [2] L. B. Kish, S. Khatri and S. Sethuraman, Noise-based logic hyperspace with the superposition of 2N states in a single wire, *Physics Letters A* **373** (2009) 1928-1934.
- [3] S. M. Bezrukov and L. B. Kish, Deterministic multivalued logic scheme for information processing and routing in the brain, *Physics Letters A* **373** (2009) 2338-2342.
- [4] L. B. Kish, S. Khatri and F. Peper, Instantaneous noise-based logic, *Fluctuation and Noise Letters* **9** (2010) 323-330.

- [5] F. Peper and L. B. Kish, Instantaneous, non-squeezed, noise-based logic, *Fluctuation and Noise Letters* **10** (2011) 231-237.
- [6] L. B. Kish, S. Khatri and T. Horvath, Computation using noise-based logic: efficient string verification over a slow communication channel, *European Physical Journal B* **79** (2011) 85-90.
- [7] Z. Gingl, S. Khatri, L.B. Kish, "Towards brain-inspired computing", *Fluctuation and Noise Letters* **9** (2010) 403-412.
- [8] L. B. Kish, S. M. Bezrukov, S. P. Khatri, Z. Gingl, S. Sethuraman, Noise-Based Logic and Computing: From Boolean Logic Gates to Brain Circuitry and Its Possible Hardware Realization, *Natural Computing* **2** (2010) 13-22.
- [9] H. Wen, L. B. KISH, A. Klappenecker and F. Peper, "New noise-based logic representations to avoid some problems with complexity", submitted to *Fluctuation and Noise Letters*, <http://arxiv.org/abs/1111.3859>.