Bayesian Estimation of Multi-Trap RTN Parameters Using Markov Chain Monte Carlo Method

Hiromitsu AWANO(a), Nonmember, Hiroshi TSUTSUI(b), Hiroyuki OCHI(c), and Takashi SATO(d), Members

SUMMARY Random telegraph noise (RTN) is a phenomenon that is considered to limit the reliability and performance of circuits using advanced devices. The time constants of carrier capture and emission and the associated change in the threshold voltage are important parameters commonly included in various models, but their extraction from time-domain observations has been a difficult task. In this study, we propose a statistical method for simultaneously estimating interrelated parameters: the time constants and magnitude of the threshold voltage shift. Our method is based on a graphical network representation, and the parameters are estimated using the Markov chain Monte Carlo method. Experimental application of the proposed method to synthetic and measured time-domain RTN signals was successful. The proposed method can handle interrelated parameters of multiple traps and thereby contributes to the construction of more accurate RTN models.

key words: random telegraph noise, Bayesian estimation, Markov chain Monte Carlo, device characterization, source separation, statistical machine learning

1. Introduction

Random telegraph noise (RTN) is the temporal change in threshold voltages observed in metal-oxide-semiconductor (MOS) transistors. Figure 1 shows the RTN generation mechanism. The capture of a carrier to an electrically active trap (defect) and its release cause a charge change in the dielectric, leading to shifts in the threshold voltage [1]. As device sizes shrink, the effects of RTN are expected to become more serious [2].

Nagumo et al. introduced a time lag plot (TLP) to investigate the magnitude of the threshold voltage shift caused by RTN [3]. They further analyzed other parameters such as trap position by measuring devices with a single observable trap [4]. Miki et al. [5] and Realov et al. [6] applied a hidden Markov model (HMM) for investigating the time constants of RTN. In the conventional studies, however, extraction of amplitudes has been limited to devices that have equal to or less than two traps. Since devices having more than two traps in measurement data are common, it is important to develop a method that is applicable to arbitrary number of traps. Furthermore, the extraction of time constants cannot be conducted through HMM. Hence, a separate process, such as application of TLP, is required. Application of two different analyses to extract interrelated parameters leads to insufficient accuracy in parameter extraction. In Table 1, characteristics of the proposed and the existing methods are compared. Figure 2 shows two examples of measured threshold voltage shifts due to RTN. In the case of devices with a single trap, we can estimate the magnitude of the threshold voltage shift simply by calculating the distance between two peaks in threshold voltage histogram. On the other hand, in the case of multiple traps, parameter extraction becomes much more complicated because parameters such as magnitudes, time constants, and trap states are strongly interrelated. Furthermore, it is an under-determined problem because it requires reconstruction of multi-trap time series sequences from only the threshold voltage fluctuation, which is superimposed with multi-trap activity. Hence, the direct use of HMM, as in conventional estimation methods, is inappropriate.

In this paper, we propose a novel method for extracting RTN parameters based on a machine learning method. We construct a statistical generation model that reflects the device physics of RTN, in which magnitudes, time constants, and trap states are random parameter variables. In our method, these parameters are estimated using Markov chain Monte Carlo methods (MCMC) so that they best fit the model to an observed RTN signal.

The proposed method has advantages over existing methods, which are summarized as follows.

1. There is no limit to the number of traps. More than two traps can be treated by the proposed method.
2. All RTN parameters are directly estimated. No post-processing is necessary.
Table 1  Summary of capabilities among the proposed and the existing methods.

<table>
<thead>
<tr>
<th>Proposed Method</th>
<th>Number of traps</th>
<th>Amplitudes extraction</th>
<th>Time constants extraction</th>
</tr>
</thead>
<tbody>
<tr>
<td>TLP [3]</td>
<td>Equal to or less than two traps</td>
<td>OK</td>
<td>NG</td>
</tr>
<tr>
<td>TLP+HMM [5,6]</td>
<td>Equal to or less than two traps</td>
<td>OK</td>
<td>OK (insufficient accuracy)</td>
</tr>
</tbody>
</table>

3. The RTN parameters are simultaneously estimated so that estimations of the interrelated parameters become consistent.

The rest of this paper is organized as follows. In Sect. 2, we will describe the proposed method for decomposing the statistics of traps. In Sect. 3, we will describe the experimental validation of the method using synthetic RTN and its results. We will also discuss the ability of the method. In Sect. 4, we will describe the results of a parameter extraction experiment using measured threshold voltages. Finally, Sect. 6 summarizes this paper.

2. Proposed Method

In this section, we describe the proposed model and parameter extraction method.

2.1 Problem Setting

The input and output of the proposed method is summarized as follows.

- Input: measured RTN and the maximum number of traps assumed.
- Output: estimated temporal sequence of trap states, magnitudes of threshold voltage shift, and time constants of the traps.

The inputs of the proposed method are the threshold voltages as a function of time and the maximum number of traps assumed. The determination of the number of traps will also be described in Sect. 2.3. Figure 3(a) shows an example of the measured threshold voltage time series. The outputs of the proposed method are the RTN parameters, i.e., the estimated trap states and magnitude of the threshold voltage shifts caused by the respective traps, as shown in Fig. 3(c). Figure 3(b) shows the reconstructed threshold voltage shift using the estimated magnitudes and trap states.

2.2 Proposed Statistical Generation Model

Figure 4 shows the graphical model defined in the proposed method. Each node corresponds to a random variable. The links among nodes represent the relationships among the random variables. The variable $w$ which represent magnitudes of threshold voltage shifts, for example, depend on $\alpha_w$ and $\beta_w$. The trap states represented by the binary latent variables $z_{t,i}$, where $t = 1, \cdots, N$ are the time steps, $i = 1, \cdots, K$ are the indices of the traps, $N$ is the number of observations, and $K$ is the number of traps. The threshold voltages at each time step are $X = \{x_1, x_2, \cdots, x_N\}$.

The total threshold voltage shift is modeled as a linear summation of the threshold voltage shifts caused by each trap, which in turn is modeled as the product of the trap state and corresponding magnitude. Here, we assume that the observation error follows a normal distribution whose standard deviation is $\sqrt{\gamma}$. The probability distribution of $x_t$ can be written as

$$p(x_t | z_{t,(1:K)}, w_{1:K}, \gamma) = \mathcal{N} \left( x_t \sum_{i=1}^{K} w_i z_{t,i}, \frac{1}{\gamma} \right),$$

where $w_{1:K}$ is the magnitude of the trap and $\mathcal{N}(x|\mu,\sigma)$ represents a probability distribution function of the normal di-
The proposed method automatically adjusts the model complexity based on Bayesian inference. Owing to the sparse assumption placed to the prior distribution, excess traps will be degenerated, i.e., amplitudes of those traps that are unnecessary to explain a given RTN waveform converge around zero. By comparing the estimated magnitude of noise and the amplitude of a trap, we can determine the most appropriate number of traps. Hence, when we apply the proposed method to an RTN waveform having unknown number of traps, we may give a large number, such as five, as the initial number of traps. Alternatively, we can also loosely estimate the number of traps using TLP.

2.4 Parameter Estimation Algorithm

To estimate the model parameters, we have to evaluate the posterior distribution

\[ P(w, 1/\tau, \gamma, Z, \alpha, \beta, \alpha_{1/\tau}, \beta_{1/\tau} | X) \]

(17)

which represents the model parameter probability after \( X \) is observed. However, the posterior distribution cannot be analytically calculated because this requires analytical integration of the posterior density function. Hence, we approximate the posterior distribution by samples generated from the distribution using the MCMC method.

2.4.1 Gibbs Sampling

Gibbs sampling is an example of an MCMC algorithm. The purpose of this algorithm is to generate a sequence of samples from a joint probability distribution of multivariate random variables [7]. Each step in the Gibbs sampling algorithm involves replacing the value of one random variable with a new sample generated from the distribution conditioned on the values of the remaining variables. Here, we outline how Gibbs sampling can be applied to a Monte Carlo approximation of Eq. (17). The following procedures are iterated after initialization of each random variables, and \( w^{(n)} \) indicates the samples drawn from the corresponding distribution in the \( n \)-th iteration of Gibbs sampling.

1. For \( i = 1, \cdots, K \), sample \( w_i \) from the following conditional posterior density,

\[ w_i^{(n+1)} \sim p \left( w_i | X_{(1:N)}, w_{(i+1)}^{(n+1)} \right) \]

(18)

2. Sample \( \gamma^{(n+1)} \) from the following conditional posterior density,

\[ \gamma^{(n+1)} \sim p \left( \gamma | X_{(1:N)}, \alpha_i^{(n+1)} \right) \]

(19)

Note that the new sample of \( w_i \) drawn at the previous step is used.

3. Sample \( z_i^{(n+1)} \) from the following posterior density,

\[ z_i^{(n+1)} \sim p \left( z_i | \alpha_i^{(n)}, \beta_i^{(n)} \right) \]

(20)

In this step, \( w \) and \( \gamma \) are replaced with the new samples...
4. For \( i = 1, \ldots, K \), sample \( \pi_i^{(n+1)} \) from the following posterior density,
\[
\pi_i^{(n+1)} \sim p \left( \frac{\pi_i^{(n+1)}}{\alpha_i^{(n)}}, a_i^{(n)}, b_i^{(n)} \right).
\]  
(21)

5. For \( t = 2, \ldots, N \), sample \( \zeta_i^{(n+1)} \) from the following posterior density,
\[
\zeta_i^{(n+1)} \sim p \left( \zeta_i | x_i^{(n+1)}, \gamma_i^{(n)}, 1/\tau_i^{(n)}, u_i^{(n)}, y_i^{(n+1)} \right).
\]  
(22)

6. For \( i = 1, \ldots, K \) and \( l = 0, 1 \), sample \( \frac{1}{\tau_i^{(n)}} \) from the following posterior density,
\[
1/\tau_i^{(n+1)} \sim p \left( \frac{1}{\tau_i^{(n,j)}} | \gamma_i^{(n)} \right) \sim \frac{1}{\gamma_i^{(n)}}, a_i^{(n)}, b_i^{(n)} \right).
\]  
(23)

7. For \( i = 1, \ldots, K \), sample \( \alpha_i^{(n+1)} \) and \( \beta_i^{(n+1)} \) from the following posterior densities, respectively,
\[
\alpha_i^{(n+1)} \sim p \left( \alpha_i | \mu_i^{(n)}, \gamma_i^{(n)} \right), \quad \text{and} \quad \beta_i^{(n+1)} \sim p \left( \beta_i | \mu_i^{(n)}, \gamma_i^{(n)} \right).
\]  
(24)

8. For \( i = 1, \ldots, K \) and \( l = 0, 1 \), sample \( \alpha_{1/\tau_i^{(n)}} \) and \( \beta_{1/\tau_i^{(n)}} \) in the same way as sampling \( \alpha_i \) and \( \beta_i \).

### 2.4.2 Conditional Posterior Density

A part of developing the posterior density is derived from a paper on sound source separation using MCMC [8].

1. The posterior density corresponding to \( u_i^{(n+1)} \) is
\[
p \left( u_i | x_i^{(n+1)}, u_i^{(n+1)}, y_i^{(n)}, \gamma_i^{(n)} \right) \sim p \left( x_i | N \right) \gamma_i^{(n)} \right), \alpha_i^{(n)}, b_i^{(n)} \right).
\]  
(26)

The first term on the right-hand side of Eq. (26) can be written as
\[
p \left( x_i | N \right) \gamma_i^{(n)} \right), \alpha_i^{(n)}, b_i^{(n)} \right),
\]  
(27)

where \( \gamma_i^{(n)}, \mu_i^{(n)}, \text{and} \; \mathcal{E}_i^{(-)} \) are defined as
\[
\gamma_i^{(n)} = \sqrt{\gamma_i^{(n)} \sum_{i=1}^{N} \left( \frac{l_{i,n}(i)}{\gamma_i^{(n)}} \right)^2},
\]  
(28)

\[
\mu_i^{(n)} = \frac{\sum_{i=1}^{N} l_{i,n}(i) \mathcal{E}_i^{(-)}(i)}{\sum_{i=1}^{N} l_{i,n}(i)^2}, \quad \text{and} \quad \mathcal{E}_i^{(-)}(i) = x_i - \sum_{j=1}^{K} l_{i,n}(j) u_i^{(n+1)}(j) - \sum_{j=1}^{K} l_{i,n}(j) u_i^{(n)}(j).
\]  
(29)

The second term on the right-hand side of Eq. (26) is the probability density function of the gamma distribution defined in Eq. (13). Therefore, the posterior density function can be written as
\[
p \left( u_i^{(n)} | X, u_i^{(n+1)}, u_i^{(n+1)}(i), y_i^{(n)}, \gamma_i^{(n)}, \alpha_i^{(n)}, b_i^{(n)} \right)
\]  
(31)

The shape of this distribution is too complex to analytically calculate the normalization constant, which requires analytical integration of the distribution. Hence, we sample \( u_i^{(n)} \) by using the Metropolis method [9]. To construct a proposal distribution, we approximate the posterior distribution by a Gaussian distribution whose variance and mode is same as the posterior distribution.

First, Eq. (31) is rewritten as follows,
\[
p \left( u_i^{(n)} | X, u_i^{(n+1)}, u_i^{(n+1)}(i), y_i^{(n)}, \gamma_i^{(n)}, \alpha_i^{(n)}, b_i^{(n)} \right)
\]  
(32)

where \( \gamma_i^{(n)} \) and \( \mu_i^{(n)} \) are defined as
\[
\gamma_i^{(n)} = \gamma_i^{(n)} \quad \text{and} \quad \mu_i^{(n)} = \mu_i^{(n)} - 1 \gamma_i^{(n)}.
\]  
(33)

The mode of the posterior density can be calculated as the solution of the following equation,
\[
u_i^{(n)} = \exp \left( \frac{-u_i^{(n)} - \mu_i^{(n)}}{2 \gamma_i^{(n)}} \right).
\]  
(35)

Therefore, the mode of the posterior distribution can be written as
\[
u_i^{(n)} = \begin{cases} 0 & \text{if } D < 0 \\ \max \left( \mu_i^{(n)} \pm \sqrt{D}, 0 \right) & \text{otherwise} \end{cases}
\]  
(36)

\[
D = \left( \mu_i^{(n)} \right)^2 + 4 \gamma_i^{(n)} - 1 \gamma_i^{(n)}.
\]  
(37)

Finally, we can obtain the following proposal distribution,
\[
q(u_i^{(n+1)}) = N \left( \frac{u_i^{(n+1)}}{\mu_i^{(n+1)}}, \sqrt{1/\gamma_i^{(n)}}, \nu_i^{(n+1)} \right).
\]  
(38)
The first term on the right-hand side of Eq. (39) can be written as

\[
p \left( x_{1:N}^{(n+1)}, \gamma^{(n+1)} \Big| \frac{1}{\tau_{(n+1)}}, \alpha_{\gamma}^{\text{prior}}, \beta_{\gamma}^{\text{prior}} \right) = \prod_{i=1}^{N} \mathcal{N} \left( x_i \Big| \frac{1}{\tau_{(n+1)}}, \gamma^{(n+1)} \right) \propto \gamma^{N/2} \exp \left\{ -\gamma \sum_{i=1}^{N} \left( x_i - \sum_{j=1}^{K} \gamma^{(n)} w_{ij}^{(n+1)} \right)^2 \right\}. \tag{40} \]

Because the prior distribution \( p \left( y | \alpha_{\gamma}^{\text{prior}}, \beta_{\gamma}^{\text{prior}} \right) \) is a gamma distribution, the posterior distribution can also be written as the following gamma distribution,

\[
p \left( \gamma^{(n+1)} | x_{1:N}, \alpha_{\gamma}^{\text{post}}, \beta_{\gamma}^{\text{post}} \right) = \mathcal{G} \left( \gamma^{(n+1)} | \alpha_{\gamma}^{\text{post}}, \beta_{\gamma}^{\text{post}} \right), \tag{41} \]

where

\[
\alpha_{\gamma}^{\text{post}} = \alpha_{\gamma}^{\text{prior}} + \frac{N}{2} \quad \text{and} \quad \beta_{\gamma}^{\text{post}} = \frac{1}{\beta_{\gamma}^{\text{prior}}} + \frac{1}{2} \sum_{i=1}^{N} \left( x_i - \sum_{j=1}^{K} \gamma^{(n)} w_{ij}^{(n+1)} \right)^2. \tag{42} \]

3. The posterior distribution corresponding to \( z_i^{(n+1)} \) is

\[
p \left( z_i | x_i, z_{i-1}^{(n)}, z_{i+1}^{(n)}, 1/\tau^{(n)}, w_{i(1:p)}, \gamma^{(n+1)} \right) \propto p \left( z_i | x_i, w_{i(1:p)}, z_{i-1}^{(n)}, z_{i+1}^{(n)}, 1/\tau^{(n)} \right) \times p \left( z_i^{(n)} | z_{i-1}^{(n)}, 1/\tau^{(n)} \right), \tag{43} \]

where \( p \left( x_i | z_i, w_{i(1:p)}, \gamma \right) \) and \( p \left( z_{i(0:l)}, 1/\tau^{(n)} \right) \) are defined in Eq. (1) and Eq. (3), respectively. To sample \( z_i \), we enumerate the possible combinations of \( z_i \) (number of combinations: \( 2^K \)), calculate the probability of each combination, and sample \( z_i \) according to the probability.

4. The posterior distribution corresponding to \( z_1^{(n+1)} \) is

\[
p \left( z_1 | x_1, z_2^{(n)}, 1/\tau^{(n)}, w_{1(1:p)}, \gamma^{(n+1)} \right) \propto p \left( x_1 | z_1, w_{1(1:p)}, z_2^{(n)}, 1/\tau^{(n)} \right) \times p \left( z_2^{(n)} | z_1, 1/\tau^{(n)} \right), \tag{44} \]

where \( p \left( x_1 | z_1, w_{1(1:p)}, \gamma \right) \) is defined as Eq. (4). Drawing samples from the posterior distribution is done in the same way as sampling \( z_i \) \((n + 1)\).

5. The posterior distribution corresponding to \( 1/\tau_{(n+1)} \) is

\[
p \left( \frac{1}{\tau_{(n+1)}}, \gamma^{(n+1)} | x_{1:N}, \alpha_{\gamma}^{\text{post}}, \beta_{\gamma}^{\text{post}} \right) \propto p \left( \frac{1}{\tau_{(n+1)}}, \gamma^{(n+1)} | x_{1:N}, \alpha_{\gamma}^{\text{post}}, \beta_{\gamma}^{\text{post}} \right). \tag{46} \]

The first term on the right-hand side of Eq. (46) can be written as

\[
p \left( \tau_{(n+1)}, \gamma^{(n+1)} | x_{1:N} \right) \propto \frac{1}{\tau_{(n+1)}} \left\{ 1 - \exp \left( -\frac{1}{\tau_{(n+1)}} \right) \right\}, \tag{47} \]

where \( n_{i-\ell} \) is the number of steps whose state is the same as the previous one \( z_{i(l)}^{(n+1)} = z_{i(l-1)}^{(n+1)} \), and \( n_{i-\ell} \) is the number of steps whose state is different from the previous one \( z_{i(l)}^{(n+1)} = l \) and \( z_{i(l-1)}^{(n+1)} = l \). The second term on the right-hand side of Eq. (46) is the gamma distribution. The posterior distribution does not belong to well-known distributions. Therefore, samples are drawn using the Metropolis method.

6. The posterior distribution corresponding to \( n_i^{(n+1)} \) is

\[
p \left( n_i | \tau_{(n+1)}, \alpha_{n}^{\text{prior}}, \beta_{n}^{\text{prior}} \right) \propto p \left( n_i | \alpha_{n(1)}, \beta_{n(1)} \right), \tag{48} \]

The first term on the right-hand side of Eq. (48) is defined in Eq. (4). The second term on the right-hand side of Eq. (48) is the beta distribution defined in Eq. (15). Hence, the posterior distribution also becomes the following beta distribution,

\[
p \left( n_i | \tau_{(n+1)}, \alpha_{n}^{\text{prior}}, \beta_{n}^{\text{prior}} \right) = \mathcal{B} \left( n_i | \alpha_{n}^{\text{post}}, \beta_{n}^{\text{post}} \right), \tag{49} \]

where

\[
\alpha_{n}^{\text{post}} = \alpha_{n}^{\text{prior}} - n_{i(0:l)}^{(n+1)} + 1 \quad \text{and} \quad \beta_{n}^{\text{post}} = \beta_{n}^{\text{prior}} + n_{i(0:l)}^{(n+1)}. \tag{50} \]

7. The posterior distribution corresponding to \( a^{(n+1)}_{w} \) is

\[
p \left( a^{(n+1)}_{w(i)}, \beta_{w(i)}, \lambda_{a(i)} \right) \propto p \left( a^{(n+1)}_{w(i)} | \beta_{w(i)}, \lambda_{a(i)} \right), \tag{52} \]

This posterior distribution also does not belong to well-known distributions. Therefore, the Metropolis method is used to draw samples from the posterior distribution.

8. The posterior distribution for sampling \( \beta_{w}, \alpha_{1(1:k)}, \beta_{1(1:k)} \) and \( \beta_{1(1:k,0)} \) is the same as the posterior distribution of \( \alpha_{w} \).

2.4.3 Proposed Parameter Estimation Algorithm

The proposed parameter estimation algorithm is defined as below.

1. Initialize the following random variables: \( \gamma^{(0)}, \nu^{(0)}, \alpha^{(0)}, \beta^{(0)}, \alpha_{1(1:k)}, \beta_{1(1:k)}, \beta_{1(1:k,0)} \), and

\[
\alpha \left( z_{i(1:k)}, t_{(n+1)} | \frac{1}{\tau_{(n+1)}}, \alpha_{\gamma}^{(n)}, \beta_{\gamma}^{(n)} \right). \tag{46} \]
AWANO et al.: BAYESIAN ESTIMATION OF MULTI-TRAP RTN PARAMETERS USING MARKOV CHAIN MONTE CARLO METHOD

The estimated trap states and reconstructed RTN using the proposed method successfully estimates the magnitude and time constants of the traps of the synthetic RTN signal within an error of 13.9% and 14.2%, respectively. From Fig. 5, we can see that the amplitude and trap states are extracted with good accuracy from the input data. They match the results obtained from a manual extraction.

3.2 Failure Analysis Using Synthetic RTN Data

To estimate the parameters of the measured RTN signal, it is important that they are validated because we cannot know the ground truth. Therefore, we conduct a Monte Carlo simulation experiment to evaluate the performance of the proposed method.
3.2.1 Experimental Procedure

Amplitudes \( w \) and time constants \( \tau \) are randomly sampled from the following distributions

\[
\begin{align*}
w &\sim \text{lognormal}(-1, 2) \\
\tau &\sim \text{lognormal}(7, 1),
\end{align*}
\]

where \( \text{lognormal}(w|\mu, \sigma) \) indicates log-normal distribution with the following probability density function,

\[
p(w|\mu, \sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp \left( -\frac{(\ln w - \mu)^2}{2\sigma^2} \right).
\]

In order to choose the distribution suitable for the following experiments, we examined existing studies though most of the analysis are limited to a single trap. One paper [6] reports that the amplitudes are well represented by log-normal distribution. On the other hand, it looks there is no consensus to the time constants distribution. In the following, based on [10],[11], we adopt log-normal distribution as the time constants model. We first generate a clean RTN signal by using \( w \) and \( \tau \) generated from the distributions above. Then, Gaussian noise that simulates measurement noise is injected. Here, the standard variation of the noise is set to 0.01. We then compare the estimated parameters with those used for generating the test sequence. Note here that the comparison is non-trivial because the trap permutation can be arbitrary. For example, the parameters of trap \#1 in Table 2 correspond to those of trap \#3 in Table 3. We need a unified approach to match the estimated traps to those in the generation.

In this experiment, we solve this permutation problem by maximizing the match between the temporal trap sequences of the generated and estimated signals. For this purpose, we introduce the Rogers-Tanimoto coefficient [12] to evaluate the similarity of trap states \( x \) and \( y \),

\[
D_{\text{Rogers–Tanimoto}}(x, y) = \frac{a_{00} + a_{11}}{a_{00} + a_{11} + 2(a_{01} + a_{10})}.
\]

Here, \( a_{00} \), \( a_{01} \), \( a_{10} \), and \( a_{11} \) are

\[
\begin{align*}
a_{00} &= \#(x_i = 0 \land y_i = 0), \\
a_{01} &= \#(x_i = 0 \land y_i = 1), \\
a_{10} &= \#(x_i = 1 \land y_i = 0), \text{ and} \\
a_{11} &= \#(x_i = 1 \land y_i = 1),
\end{align*}
\]

where \( x_i \) and \( y_i \) are the temporal sequences of the trap states for time-step \( i \). The coefficient increases when two traps are more similar. Then, the permutation that maximizes the above similarity is used for the evaluation. Once correspondence between traps is determined, we can define estimation errors corresponding to each trap.

3.2.2 Confidence Evaluation of Estimated Parameters

Because there is no way of knowing the ground truth of parameters, i.e., the amplitudes and time constants, a method to evaluate the confidence of the estimated parameters is definitely required. For that purpose, we propose the following criteria to recognize the accuracy of the decomposed multi-trap parameters in the proposed method.

**Estimated noise accuracy** \( \gamma \) Good noise accuracy, which can be observed as \( \gamma \) (the inverse of the noise variance), indicates good agreement between the input and reconstructed signals. This means that the proposed method successfully decomposed the trap properties.

**Similarities between estimated trap states** When multiple traps exhibit similar temporal trapping-and-emitting sequences, it is difficult to estimate the parameters of each trap separately. In these input signals, confidence of the estimated result becomes low. Here, the similarities between estimated trap sequences can again be evaluated using the Rogers-Tanimoto coefficient. The maximum similarity is used as a criterion among possible \( K(K + 1)/2 \) pairs for \( K \) traps.

**Difference in \( V_{\text{TH}} \) levels between states** There is an ambiguity in the amplitude if the threshold voltage shift caused by some trap combinations is similar to that caused by another trap combination (Fig. 6). The histogram peaks formed by projection onto the threshold voltage axis should be clearly distinguished from each other. We calculate the reconstructed signal and then calculate a voltage histogram to investigate the distances between states. When there are \( K \) traps, the number of \( V_{\text{TH}} \) levels to be distinguished is \( 2^K \). This means that the problem becomes inherently difficult as the number of states increases. The minimum distance is used as the criterion.

3.2.3 Experimental Result

The results of the Monte Carlo experiment are shown in Figs. 7 and 8. Here, two traps are used and 5,000 samples are generated. The parameters for the prior distributions are set to \( \alpha_\gamma = 10^3, \beta_\gamma = 10^{-1}, \lambda_{\alpha_\gamma} = 1.0, \lambda_{\beta_\gamma} = 0.1, \alpha_\sigma = 1.0, \beta_\sigma = 1.0, \lambda_{\alpha_{\sigma_1\gamma}} = 10, \text{ and } \lambda_{\beta_{\sigma_1\gamma}} = 1. \) Figures 7(a) and 8(a) show the maximum estimation error of the amplitudes and time constants (Z axis) versus the estimated noise accuracy \( \gamma \). Figures 7(b) and 8(b) show the maximum estimation error of the amplitudes and time constants (Y axis) versus the minimum distance of \( V_{\text{TH}} \) levels (X axis), respectively. The blue diamonds represent estimated results that fall into the trust region and those for which the maximum estimation error of the amplitudes and
Fig. 7 Estimation accuracy of the amplitudes. Samples inside the trust region are indicated by red crosses and blue diamonds. Red crosses indicate that the maximum estimation error of the amplitudes is larger than 10%. Blue diamonds indicate that the proposed method successfully estimated the amplitude with a 10% accuracy. Samples outside the trust region are indicated by the green triangles.

Fig. 8 Estimation accuracy of the time constants. Samples inside the trust region are indicated by red crosses and blue diamonds. Red crosses indicate the maximum estimation error of the time constants is larger than 50%. Blue diamonds indicate the proposed method successfully estimated the time constant with a 50% accuracy. Samples outside the trust region are indicated by the green triangles.

The minimum difference of the $V_{TH}$ levels is less than 10% and 50%, respectively. The red crosses indicate that estimated results fall into the trust region with at least one of the estimation errors of the amplitudes and time constants being greater than 10% and 50%, respectively. The green triangles indicate estimated results that fall outside of the trust region. The followings is a description of the trust region.

1. The estimated noise accuracy $\gamma$ is greater than $9.8 \times 10^3$.
2. The maximum similarity of estimated trap states is less than 0.9.
3. The minimum distance of the $V_{TH}$ levels between estimated states is greater than $10^{-2}$.

Because we assume that the standard variation of the measurement noise is 0.01, the estimated noise accuracy should be around $10^4$. By the same token, the distances between each state should be greater than the amplitude fluctuation caused by noise. The maximum similarity reaches 1.0 if and only if two traps exhibit the same temporal trapping-and-emitting sequences. To exclude such cases, we introduce a condition for the similarity.

Figure 9 shows the relationship between amplitudes and time constants of generated sequences (X axis) and estimation accuracy (Y axis). The green diamonds represent estimated results that fall into the trust region while the black crosses represent estimated results that fall outside of the trust region. We can see that estimation accuracy becomes low as the amplitudes become small (less than 0.01) and the time constants become large (greater than 10k). In this Monte Carlo simulation experiment, 3,571 out of 5,000 samples fall into the trust region. Among those, 3,499 samples succeeded in estimating amplitudes within 10% accuracy. Similarly, 2,833 samples in the trust region succeeded
Fig. 9 The relationship between generation parameters and estimation accuracy. Green diamonds indicate the samples inside the trust region and black crosses indicate the samples outside of the trust region, respectively. The estimation error of amplitudes tend to increase as amplitude becomes small (a) while the estimation error of time constants tend to increase as time constants become large (b).

in estimating time constants within 50% accuracy. 2,823 samples simultaneously satisfied the above amplitudes and time constants bounds. On the other hand, relative errors become large for those samples having traps with small amplitudes. In our experiment, the estimation accuracy is judged by the maximum estimation error of all traps. Thus, traps with small amplitudes have a low estimation accuracy, whereas other parameters are estimated with good accuracy. Furthermore, estimating time constants is difficult owing to the weak sensitivity of the transition probabilities to the time constants. When \( \tau \) is 100, for example, \( \exp(-1/\tau) \) is about 0.999, which increases by only 0.005 when \( \tau \) is increased to 200.

4. Experiments Using Measured RTN Data

We extract parameters from the measured threshold voltage shift of a pMOS transistor fabricated in a 65-nm process [13]. All parameters for the prior distribution remain fixed as those of the previous section. We have to know the ground truth of the noise variance of the measured RTN data so that we can apply the same trust region as we used in the previous section. To calculate the ground truth of the noise variance, measured signals unaffected by RTN are used. Then, measured signals with the RTN effects are scaled so that they have the same noise variance as the synthetic RTN signal in the previous section. Estimation results that fall outside the trust region defined in the previous section are omitted. These procedures are summarized as follows.

1. Find the trust region using the synthetic RTN data.
2. Apply a model estimation algorithm to the measured signal:
   a. Calculate the noise variance using the measured data without the effects of RTN.
   b. Scale the measured data that includes the RTN effects.
   c. Omit the estimation results that fall outside the trust region.

The estimated results of 46 out of 128 devices fall into the trust region, one of them is shown in Fig. 10. The channel length \( L \) and width \( W \) are \( W/L = 360 \text{ nm}/120 \text{ nm} \), respectively. The estimated amplitudes and time constants are listed in Table 4. The estimated noise accuracy \( \gamma \) is \( 1.27 \times 10^4 \). It takes about 15 minutes for estimating the parameters of a single device on Core i7 at 2.8 GHz.

From Fig. 10, we can see that the temporal sequences of two traps are clearly decomposed. The residue signal after subtraction of the reconstructed RTN shows that the effect of RTN is clearly removed. This means that the proposed method successfully captures the statistics of multitraps.

Table 4 Estimated parameters from the measured RTN signal (W/L=360/120).

<table>
<thead>
<tr>
<th>Trap</th>
<th>#1</th>
<th>#2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Amplitude [a.u.]</td>
<td>0.43×10^{-2}</td>
<td>1.12×10^{-2}</td>
</tr>
<tr>
<td>( \tau_0 ) [a.u.]</td>
<td>1.80×10^3</td>
<td>8.80×10^3</td>
</tr>
<tr>
<td>( \tau_1 ) [a.u.]</td>
<td>2.34×10^2</td>
<td>1.47×10^3</td>
</tr>
</tbody>
</table>
such as with parallel tempering, is also one topic of our future work.

5. Comparative Experiments with Existing Method

In this section, we compare our method with hidden Markov model (HMM) [14]. HMM is a popular and widely used method which is suitable to build a statistical model for time-domain sequences, such as voices. In the context of RTN analysis, HMM can be used to extract transition probabilities between discrete threshold voltage states. The main difference between the proposed method and HMM is that the proposed method takes into account the generation process of RTN while HMM does not. Large threshold voltage fluctuation is usually caused by combination of two or more trap activities. On the other hand, estimating the probability that a large threshold voltage shift occurs is difficult because such states that multi traps capture electrons will hardly be observed. In the following experiment, we show that extraction accuracy of such probability is improved over HMM by proposed method.

5.1 Experiment Setup

We show the estimation accuracy of the transition probability of proposed method and HMM using the artificially generated test sequence that is shown in Fig. 11(a). Here, we use the HMM of MATLAB toolkit [15]. Assuming that amplitudes are extracted accurately using other method such as TLP in advance to the transition probability estimation using HMM, the emission probabilities of each state are modeled as Gaussian distributions whose mean and variance are considered to be the ground truth. They are fixed during parameter update. Hence, in this experiment, HMM only estimates transition probabilities.

5.2 Results and Discussion

Figure 11(b) shows the decomposed trap states using the proposed method and Fig. 11(c) shows the reconstructed RTN waveform using Viterbi path of trained HMM [16]. The Viterbi path means the most probable sequence of hidden states for a given observation sequence. In the context of RTN analysis, the Viterbi path represent the most probable trapping-and-emitting sequence for a given threshold voltage sequence. The direct comparison between HMM and proposed method is difficult because HMM can only estimate transition probability matrix. Therefore, we create the transition probability matrix of the proposed method from the estimated time constants as shown in Fig. 12. For evaluation, we analyze the steady state of estimated Markov model. A steady state probability $\pi = (\pi_1, \cdots, \pi_K)$ of Markov model is a solution of the following linear equation

$$A\pi = \pi,$$  \hspace{1cm} (61)

where $A$ is the transition probability matrix. The calculated steady state probabilities using estimated transition probability of HMM and the proposed method are listed in Ta-

Acknowledgment

This research is partially supported by a Grant-in-Aid for Scientific Research (B) 22360143 and JST CREST. Numerical computations were carried out on the super computer system, ACCMS, Kyoto University.

References

Hiroshi Tsutsui received his B.E. degree in Electrical and Electronic Engineering and his master and Ph.D. degrees in Communications and Computer Engineering from Kyoto University in 2000, 2002, and 2005, respectively. He is currently an assistant professor in the Department of Communications and Computer Engineering, Kyoto University. His research interests include circuits and systems for image processing and VLSI design methodology. He is a member of IEEE, ACM, IPSJ, IEEJ, and IIEEJ.

Hiroyuki Ochi received the B.E., M.E., and Ph.D. degrees in Engineering from Kyoto University in 1989, 1991, and 1994, respectively. In 1994, he joined Department of Computer Engineering, Hiroshima City University as an associate professor. Since 2004, he has been an associate professor of Department of Communications and Computer Engineering, Kyoto University. His research interests include low-power/reliability-aware VLSI design and reconfigurable architectures. He is a member of IPSJ, IEEE, and ACM.

Takashi Sato received B.E. and M.E. degrees from Waseda University, Tokyo, Japan, and a Ph.D. degree from Kyoto University, Kyoto, Japan. He was with Hitachi, Ltd., Tokyo, Japan, from 1991 to 2003, with Renesas Technology Corp., Tokyo, Japan, from 2003 to 2006, and with the Tokyo Institute of Technology, Yokohama, Japan. In 2009, he joined the Graduate School of Informatics, Kyoto University, Kyoto, Japan, where he is currently a professor. He was a visiting industrial fellow at the University of California, Berkeley, from 1998 to 1999. His research interests include CAD for nanometer-scale LSI design, fabrication-aware design methodology, and performance optimization for variation tolerance. He received the Beatrice Winner Award at ISSCC 2000 and the Best Paper Award at ISQED 2003.