An Efficient Gaussian Mixture Reduction to Two Components

Naoya Yokoyama[†], Daiki Azuma, Shuji Tsukiyama

Dept. of Electric, Electronic, and Communication Eng.
Chuo University, Tokyo 112-8551, Japan
e-mail : {d_azuma@tsuki., tsuki@}elect.chuo-u.ac.jp
† currently, TDC Software Eng., Inc.

Masahiro Fukui

Dept. of Electronic and Computer Eng.
Ritsumeikan Univ.
Shiga, 525-8577 Japan
e-mail : mfukui@se.ritsumei.ac.jp

Abstract - In statistical methods, such as statistical static timing analysis, Gaussian mixture model (GMM) is a useful tool for representing a non-Gaussian distribution and handling correlation easily. In order to repeat various statistical operations such as summation and maximum for GMMs efficiently, the number of components should be restricted around two. In this paper, we propose a method for reducing the number of components of a given GMM to two (2-GMM) such that the mean and the variance of the 2-GMM are equal to those of original GMM and the normalized integral square error of 2-GMM PDF is minimized. In order to demonstrate the performance of the proposed methods, we show some experimental results.

I. Introduction

Due to the progress of nanometer process technologies, variability of circuit parameters has been increased and various statistical approaches to electronic design have been proposed[1]. Among them, the statistical static timing analysis has been studied intensively in the beginning of this century[2], and algorithms using Gaussian mixture model (GMM) to represent distribution have been proposed[3,4]. Since the probability density function (PDF) of GMM is a probability weighted sum of Gaussian PDFs (each of which is called a component), GMM can represent a non-Gaussian distribution and treat correlation easily.

Since GMM has such remarkable features, it is used in various fields, such as lifetime analysis of a battery pack[5], target tracking in acoustics[6], and so on. But, in order to use GMM efficiently, we need a good algorithm to reduce the number of components, since if we repeat statistical operations for GMMs, then the number of components may increase exponentially. For example, for a given two GMMs each of which has m components (m-GMM), the sum and the maximum are represented by an m^2-GMM and a $2m^2$-GMM, respectively[4]. Because, the joint PDF of the two GMMs may have m^2 components, and each of the components produces a distribution (a component) of the sum and two components of the maximum. Hence, if we add n 2-GMMs, the sum may contain 2^n components, unless the number of components is reduced.

So far, several methods[6-12] for reducing components have been proposed, which are called Gaussian mixture reduction. All of them exclusive of [9] repeat merging of two components selected by a certain measure corresponding to a distance between two components. In [11], performance comparisons of methods in [6], [7], and [8] are described, and the method in [8], which uses a distance based on K-L discrimination, is ranked as the best method from both viewpoints of accuracy and efficiency. The method in [9] finds a GMM with reduced number of components by increasing number of components from Gaussian by splitting a component. The method in [10] modifies the distribution of the merged component. Hence, the mean and the variance of GMM with the reduced number of components obtained by [9] or [10] may differ from those of the original GMM. The method in [12] uses weighted K-L divergence as a distance, and claims that weighted K-L divergence is more suitable than K-L divergence.

In this paper, we propose a new method of reducing the number of components of a given m-GMM to two[4,13]. The method finds a 2-GMM such that the mean and the variance of the 2-GMM are equal to those of m-GMM and the normalized integral square error of 2-GMM PDF is minimized. The method combines the methods in [4] and [8] by utilizing their advantages effectively so as to obtain better performance.

We also show some experimental results to evaluate the performance of the proposed method. Since the methods in [9] and [10] do not satisfy the condition for the mean and the variance of 2-GMM to be the same as those of a given m-GMM, we do not use them in the comparison. In [13], we showed that the method in [13] could find better 2-GMMs than the methods in [12], and hence we do not use it in the comparisons, either. Moreover, since we found by our experiments that the method in [8] is better than the method in [13], we incorporate the method in [8] into the proposed method, and do not use the method in [13] in comparison. Thus, we compare the proposed method with [4] and [8].

The rest of the paper is organized as follows. In Sections II and III, we introduce Gaussian mixture models and briefly explain the methods in [4] and [8], respectively. In Section IV, we describe the proposed method, and in Section V, we show some experimental results. Finally, in Section VI, we give conclusions.

II. Gaussian Mixture Model

Let D be a random variable (RV) whose probability density function (PDF) $f_m(D)$ is denoted by a weighted sum of m Gaussian distributions such that

$$f_{\mathrm{m}}(D) = \sum_{\mathrm{i}=1}^{\mathrm{m}} \mathrm{P_i} \cdot \frac{1}{\sigma_{\mathrm{i}}} \cdot \phi\left(\frac{D-\mu_{\mathrm{i}}}{\sigma_{\mathrm{i}}}\right), \qquad (1)$$

where $\sum_{\mathrm{i}=1}^{\mathrm{m}} \mathrm{P_i} = 1$, and $\phi(\bullet)$ is the PDF of the standard Gaussian distribution N(0,1),

$$\phi(x) = \frac{1}{\sqrt{2\pi}} \cdot \exp\left[-\frac{x^2}{2}\right]. \qquad (2)$$

Then, the distribution of RV *D* is said to be represented by m-GMM (Gaussian mixture model consisting of m Gaussian distributions). Each Gaussian distribution $\mathrm{N}(\mu_i,\sigma_i^2)$ ($1 \le i \le m$) is called the i-th component of m-GMM, and probability $\mathrm{P_i}$ is called the mixture proportion of the i-th component. Henceforth, if the distribution of RV *D* is represented by an m-GMM, then we say that *D* is an m-GMM, or denote simply m-GMM *D*.

The mean E[*D*], the 2nd moment E[D^2], and the 3rd moment $\mathrm{E}[D^3]$ of an m-GMM RV *D* are calculated by the following equations, respectively,

$$\mathrm{E}[D] = \sum_{\mathrm{i}=1}^{\mathrm{m}} \mathrm{P_i} \cdot \mathrm{E_i}[D] = \sum_{\mathrm{i}=1}^{\mathrm{m}} \mathrm{P_i} \cdot \mu_{\mathrm{i}}, \qquad (3)$$

$$\mathrm{E}[D^2] = \sum_{\mathrm{i}=1}^{\mathrm{m}} \mathrm{P_i} \cdot \mathrm{E_i}[D^2] = \sum_{\mathrm{i}=1}^{\mathrm{m}} \mathrm{P_i} \cdot (\mu_{\mathrm{i}}^2 + \sigma_{\mathrm{i}}^2), \qquad (4)$$

$$\mathrm{E}[D^3] = \sum_{\mathrm{i}=1}^{\mathrm{m}} \mathrm{P_i} \cdot \mathrm{E_i}[D^3] = \sum_{\mathrm{i}=1}^{\mathrm{m}} \mathrm{P_i} \cdot (\mu_{\mathrm{i}}^3 + \mu_{\mathrm{i}} \cdot \sigma_{\mathrm{i}}^2). \qquad (5)$$

Hence, the variance V[*D*] and the skewness $\mathrm{S}[D]$ of the m-GMM *D* are obtained as follows:

$$\mathrm{V}[D] = \mathrm{E}[D^2] - \mathrm{E}[D]^2. \qquad (6)$$

$$\mathrm{S}[D] = \mathrm{E}[D^3]/\mathrm{V}[D]^3. \qquad (7)$$

In the following, $\mathrm{E_i}[D] = \mu_{\mathrm{i}}$, $\mathrm{E_i}[D^2] = \mu_{\mathrm{i}}^2 + \sigma_{\mathrm{i}}^2$, and $\mathrm{V_i}[D] = \sigma_{\mathrm{i}}^2$ denote the mean, the 2nd moment, and the variance of the i-th component of *D*, respectively.

Let $Y \sim \mathrm{N}(\mu_Y,\sigma_Y^2)$ be an RV such that the correlation coefficient between *Y* and the i-th component of *D* is ρ_i, and the joint PDF (JPDF) of *D* and *Y* is the following:

$$g(D,Y) = \sum_{\mathrm{i}=1}^{\mathrm{m}} \mathrm{P_i} \cdot \frac{1}{\sigma_{\mathrm{i}} \cdot \sigma_{\mathrm{Y}}} \cdot \phi_2\left(\frac{D-\mu_{\mathrm{i}}}{\sigma_{\mathrm{i}}}, \frac{Y-\mu_{\mathrm{Y}}}{\sigma_{\mathrm{Y}}} : \rho_{\mathrm{i}}\right), \qquad (8)$$

where $\phi_2(x,y{:}\rho)$ is the standard Gaussian JPDF of two RVs *x* and *y* with correlation coefficient ρ:

$$\phi_2(x,y{:}\,\rho) = \frac{1}{2\pi \cdot \sqrt{1-\rho^2}} \cdot \exp\left[-\frac{x^2 - 2\rho \cdot x \cdot y + y^2}{2 \cdot (1-\rho^2)}\right]. \qquad (9)$$

Then, the covariance between *D* and *Y* can be obtained by the following equation,

$$\mathrm{C}[D,Y] = \sum_{\mathrm{i}=1}^{\mathrm{m}} \mathrm{P_i} \cdot \sigma_{\mathrm{i}} \cdot \sigma_{\mathrm{Y}} \cdot \rho_{\mathrm{i}} = \sum_{\mathrm{i}=1}^{\mathrm{m}} \mathrm{P_i} \cdot \mathrm{C_i}[D,Y], \qquad (10)$$

where $\mathrm{C_i}[D,Y] = \sigma_i \cdot \sigma_Y \cdot \rho_i$ is the covariance of the i-th component of *D* and *Y*.

In this paper, we assume that m-GMM *D* varies depending on n+1 factors denoted by n+1 explanatory RVs x_D and r_g ($1 \le g \le n$), such that all of them are N(0,1) and independent each other. Among them, x_D is the local factor proper to *D*, and r_g ($1 \le g \le n$) is a common factor used for RVs other than *D*. Moreover, we assume that the distribution of each component of *D* is represented by a linear combination of x_D and r_g ($1 \le g \le n$). Namely, we assume that D_i representing the i-th component of *D* is denoted by

$$D_{\mathrm{i}} = \mu_{\mathrm{i}} + \mathrm{s_{x,i}}[D] \cdot x_{\mathrm{D}} + \sum_{\mathrm{g}=1}^{\mathrm{n}} \mathrm{s_{g,i}}[D] \cdot r_{\mathrm{g}}, \qquad (11)$$

where $\mathrm{s_{x,i}}[D] = \mathrm{C}[D_{\mathrm{i}}, x_{\mathrm{D}}]$ and $\mathrm{s_{g,i}}[D] = \mathrm{C}[D_{\mathrm{i}}, r_{\mathrm{g}}]$ are sensitivities of D_{i} to x_{D} and r_{g}, respectively. This equation means that fluctuations caused by the factors are small enough to ignore higher order terms[1]. Since any two of x_D and r_g ($1 \le g \le n$) are independent each other, the variance V[D_i] = $\mathrm{V_i}$[*D*] can be obtained by

$$\mathrm{V_i}[D] = \mathrm{s_{x,i}}[D]^2 + \sum_{\mathrm{g}=1}^{\mathrm{n}} \mathrm{s_{g,i}}[D]^2. \qquad (12)$$

Note here that RV D_i is introduced for representing the distribution of the i-th component of *D* and does not have any specific meaning.

In this paper, we consider a problem to approximate a given m-GMM *D* (m > 2) by a 2-GMM *M* such that the PDF is given by

$$f_2(M) = \sum_{\mathrm{k}=1}^{2} \mathrm{P_{Mk}} \cdot \frac{1}{\sigma_{\mathrm{Mk}}} \cdot \phi\left(\frac{M-\mu_{\mathrm{Mk}}}{\sigma_{\mathrm{Mk}}}\right), \qquad (13)$$

and the first and second moments match, that is, E[*M*] = E[*D*] and E[M^2] = E[D^2]. Moreover, in order to evaluate the approximation, we use NISE (Normalized Integral Square Error) ε_{M} defined by

$$\varepsilon_{\mathrm{M}} = \frac{\int_{-\infty}^{\infty} \left(f_2(x) - f_{\mathrm{m}}(x)\right)^2 dx}{\int_{-\infty}^{\infty} {f_2}^2(x)\, dx + \int_{-\infty}^{\infty} {f_{\mathrm{m}}}^2(x)\, dx}. \qquad (14)$$

In the case of GMM, this value is calculated by

$$\varepsilon_{\mathrm{M}} = \frac{J_{22} + J_{\mathrm{mm}} - 2 \cdot J_{2\mathrm{m}}}{J_{22} + J_{\mathrm{mm}}}, \qquad (15)$$

where J_{22}, $J_{2\mathrm{m}}$, and J_{mm} are the following equations[11],

$$J_{22} = \sum_{\mathrm{k,h} \in \{1,2\}} \mathrm{P_{Mk}} \cdot \mathrm{P_{Mh}} \cdot \frac{1}{\sqrt{\sigma_{\mathrm{Mk}}^2 + \sigma_{\mathrm{Mh}}^2}} \cdot \phi\left(\frac{\mu_{\mathrm{Mk}} - \mu_{\mathrm{Mh}}}{\sqrt{\sigma_{\mathrm{Mk}}^2 + \sigma_{\mathrm{Mh}}^2}}\right),$$

$$J_{2\mathrm{m}} = \sum_{\mathrm{k} \in \{1,2\}} \sum_{\mathrm{i} \in \{1,\cdots,\mathrm{m}\}} \mathrm{P_{Mk}} \cdot \mathrm{P_i} \cdot \frac{1}{\sqrt{\sigma_{\mathrm{Mk}}^2 + \sigma_{\mathrm{i}}^2}} \cdot \phi\left(\frac{\mu_{\mathrm{Mk}} - \mu_{\mathrm{i}}}{\sqrt{\sigma_{\mathrm{Mk}}^2 + \sigma_{\mathrm{i}}^2}}\right),$$

$$J_{\mathrm{mm}} = \sum_{\mathrm{i,j} \in \{1,\cdots,\mathrm{m}\}} \mathrm{P_i} \cdot \mathrm{P_j} \cdot \frac{1}{\sqrt{\sigma_{\mathrm{i}}^2 + \sigma_{\mathrm{j}}^2}} \cdot \phi\left(\frac{\mu_{\mathrm{i}} - \mu_{\mathrm{j}}}{\sqrt{\sigma_{\mathrm{i}}^2 + \sigma_{\mathrm{j}}^2}}\right).$$

Note here that if two PDFs $f_{\mathrm{m}}(x)$ and $f_2(x)$ are the same, then $\varepsilon_{\mathrm{M}} = 0$, and if two PDFs do not have overlap, then $\varepsilon_{\mathrm{M}} = 1$. Moreover, note that NISE is different from the total absolute PDF difference $\int_{-\infty}^{\infty} |f_2(x) - f_{\mathrm{m}}(x)|\, dx$ used in [4]. Hence, the optimum approximation with respect to NISE is not always optimum with respect to the total absolute PDF difference. However, we use NISE for evaluation in this paper, since it is easily calculated and seems to be used commonly in the field of GMM reduction[9-11].

III. Previous Algorithms

The method in [4] fits two lines with one joint to the shape of CDF curve of m-GMM *D*, so as to find a 2-GMM *M* whose CDF is close to that of m-GMM *D*. Then, by using the value D_J of the joint, it partitions the set U = { i | 1 ≤ i ≤ m } of component numbers into three disjoint subsets U1 = { i∈U | $\mu_i < D_J$ }, U2 = { i∈U | $\mu_i > D_J$ }, and U3 = { i∈U | $\mu_i = D_J$ }, and determine $\mathrm{P_{Mk}}$, μ_{Mk}, and σ_{Mk} (k=1,2) by the following equations:

$$\mathrm{P_{M1}} = \sum_{\mathrm{i} \in \mathrm{U1}} \mathrm{P_i} + \frac{1}{2} \cdot \sum_{\mathrm{i} \in \mathrm{U3}} \mathrm{P_i}, \quad \mathrm{P_{M2}} = 1 - \mathrm{P_{M1}}, \qquad (16)$$

$$\mathrm{P_{Mk}} \cdot \mu_{\mathrm{Mk}} = \sum_{\mathrm{i} \in \mathrm{Uk}} \mathrm{P_i} \cdot \mu_{\mathrm{i}} + \frac{1}{2} \cdot \sum_{\mathrm{i} \in \mathrm{U3}} \mathrm{P_i} \cdot \mu_{\mathrm{i}}, \qquad (17)$$

$$\mathrm{P_{Mk}} \cdot (\sigma_{\mathrm{Mk}}^2 + \mu_{\mathrm{Mk}}^2) = \sum_{\mathrm{i} \in \mathrm{Uk}} \mathrm{P_i} \cdot (\sigma_{\mathrm{i}}^2 + \mu_{\mathrm{i}}^2) + \frac{1}{2} \cdot \sum_{\mathrm{i} \in \mathrm{U3}} \mathrm{P_i} \cdot (\sigma_{\mathrm{i}}^2 + \mu_{\mathrm{i}}^2). \qquad (18)$$

Namely, the k-th component of *M* is composed from components in Uk and a half of each component in U3. Therefore, the method in [4], which we call the algorithm “Method-Cdf” in the following, solves our problem by finding a partition of U with small NISE.

From Eqs.(16) through (18), we can see that if U3 is nonempty or both U1 and U2 are nonempty, then we can generate a 2-GMM. Since the number of partitions dividing U into two non-empty subsets is $2^{m-1}-1$, and since the number of ways to select h elements from U for U3 is ${}_{m}C_{h}$, the total number of partitions which generate a 2-GMM is

$$2^{m-1}-1+\sum_{h=1}^{m-1}{}_{m}C_{h}\cdot 2^{m-h-1}. \qquad (19)$$

Henceforth, we call a brute force algorithm “Method-Opt,” which checks all such partitions and selects the optimum partition with the minimum NISE. We use Method-Opt to evaluate solutions obtained by algorithms.

Although Method-Cdf is efficient, it cannot find an optimum 2-GMM in the case where the distribution of *D* is composed of more than two major parts, since the algorithm tries to approximate the distribution by two parts[4]. In Fig.1, two PDFs of such distributions are shown. Hence, we incorporate a technique used in Gaussian mixture reduction into Method-CDF in order to improve Method-Cdf.

The commonly used methods in Gaussian mixture reduction repeat merging of components[8,12,13], which selects two nearest components with the use of a distance between two components, and merges them first. The K-L discrimination[8] $\mathrm{kld}(g_1|g_2)$ of PDF g_1 from PDF g_2 is used as such a distance, and is defined by

$$\mathrm{kld}(g_1|g_2)=\int_{-\infty}^{\infty} g_1(x)\cdot\log_e\frac{g_1(x)}{g_2(x)}\,dx \qquad (20)$$

If each PDF g_i (i=1,2) is a Gaussian $N(\mu_i,\sigma_i^2)$, then it is calculated by

$$\mathrm{kld}(g_1|g_2)=\frac{1}{2}\cdot\left\{\frac{{\sigma_1}^2+(\mu_1-\mu_2)^2}{{\sigma_2}^2}-1\right\}. \qquad (21)$$

Since $\mathrm{kld}(g_1|g_2)\neq\mathrm{kld}(g_2|g_1)$, $\mathrm{kld}(g_1|g_2)$ cannot be used directly as a distance between g_1 and g_2. Hence, K-L divergence (or K-L distance) is used as a distance. The K-L divergence $d_{kl}(i,j)$ between the i-th and j-th components is defined by

$$d_{kl}(i,j)=\mathrm{kld}(g_i|g_j)+\mathrm{kld}(g_j|g_i). \qquad (22)$$

If the i-th and j-th components are merged, then the mixture proportion P_{ij}, the mean μ_{ij}, and the variance ${\sigma_{ij}}^2$ of the merged component are determined by moment matching similarly to Eqs.(16) through (18).

Since K-L divergence $d_{kl}(i,j)$ does not take the mixture proportion into consideration, the method in [12] used weighted K-L divergence, which is defined by

$$d_{klw}(i,j)=\mathrm{kld}(P_i\cdot g_i|P_i\cdot g_j)+\mathrm{kld}(P_j\cdot g_j|P_j\cdot g_i). \qquad (23)$$

This value is also calculated easily if each PDF is Gaussian. However, the mixture proportion is not sufficient to represent the shape of distribution of each component. Hence, we proposed a method in [13] which takes the distribution shape of each component into account, and we showed that it is superior to the method in [12]. However, since we found that the method in [8] showed better performance than the method in [13], we use the method in [8] in our proposed algorithm.

The method in [8] defines the distance $L_m(i,j)$ between the i-th and j-th components with the use of the merged distribution, that is, the distance $L_m(i,j)$ is defined by

$$L_m(i,j)=P_i\cdot\mathrm{kld}(g_i|g_{ij})+P_j\cdot\mathrm{kld}(g_j|g_{ij}). \qquad (24)$$

where g_{ij} is the PDF of the distribution obtained by merging the i-th and j-th components (g_i and g_j). If each g_i (i=1,2) is a Gaussian $N(\mu_i,\sigma_i^2)$, then it is calculated by

$$L_m(i,j)=\frac{1}{2}\cdot\left(P_{ij}\cdot\log_e{\sigma_{ij}}^2-P_i\cdot\log_e{\sigma_i}^2-P_j\cdot\log_e{\sigma_j}^2\right) \qquad (25)$$

where mixture proportion P_{ij} and variance ${\sigma_{ij}}^2$ are calculated by similar equations to Eqs.(16) through (18). We call the algorithm using this distance to select two components for merging “Method-Mergd.”

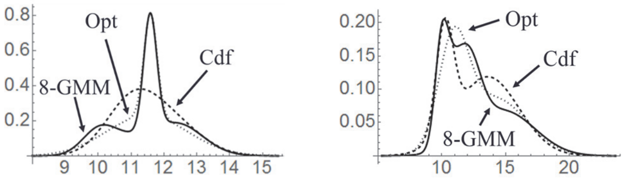
IV. Proposed Algorithm

As shown in [13], the method in [13] using Gaussian mixture reduction is not always superior to Method-Cdf, and needs longer CPU time than Method-Cdf. Similarly, Method-Mergd has the same characteristics. For example, it cannot find a better 2-GMM than Method-Cdf for the 8-GMMs shown in Fig.2. Therefore, we combine Method-Cdf and Method-Mergd so as to take advantages of both methods. Before describing the algorithm, we introduce an algorithm to check whether or not the distribution of a given m-GMM is similar to the shape of Fig.1(a), which we denote by <PDF-Check>.

<PDF-Check>

1°: Find the component $1\leq i\leq m$, whose probability density value at the mean μ_i (we call it vp value and denote it by vp_i) is maximum, and let μ_{max}, σ_{max}, and vp_{max} be the mean, the standard deviation, and the vp value of the component, respectively.

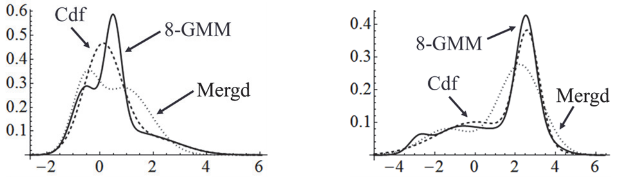
2°: Check if there exists a component i such that μ_i



(a) Shape-I

(b) Shape-II

Fig.1 – PDFs of 8-GMM for which method-Cdf cannot find an appropriate 2-GMM.



(a) Shape-III

(b) Shape-IV

Fig.2 – PDFs of 8-GMM for which method-Mergd cannot find an appropriate 2-GMM.

satisfies $\mu_i < \mu_{max} - \sigma_{max}$ or $\mu_{max} + \sigma_{max} < \mu_i$, and vp_i satisfies $vp_i > T_{pk} \cdot vp_{max}$, where T_{pk} is the threshold for the vp value and we set $T_{pk} = 1/4$ in the experiments in Section V.

3°: If there exists such a component i, then the shape of distribution of a given m-GMM is regarded as a shape different from the one shown in Fig.1(a), and finish the algorithm.

4°: Otherwise, put the component whose vp value vp_i satisfies $vp_i \leq T_{pk} \cdot vp_{max}$ into U1, and put remaining components into U2.

Now, we describe the proposed algorithm denoted by <2-GMM approximation> as follows:

<2-GMM Approximation>

1°: Conduct <PDF-Check>, and if it produced U1 and U2, then generate 2-GMM by using them, and terminate the algorithm. Otherwise go to 2°.

2°: By using Method-Cdf, generate a 2-GMM, and compute NISE ε_M with the use of Eq.(15). If ε_M is not greater than threshold Tε, then terminate the algorithm. Otherwise, discard this 2-GMM, and go to 3°.

3°: By using Method-Mergd, generate a 2-GMM.

Since we have no strict algorithm to determine Tε, we use Tε = 0.01 which is around one third of the maximum NISE ε_{Opt} obtained from 2-GMMs generated by Method-Opt for four hundreds 8-GMMs used in the experiment in [4]. This value ε_{Opt} indicates the maximum of inevitable NISEs in the 2-GMM approximation obtained by a partition of U into U1, U2, and U3.

As shown in Eq.(11), we assume that the distribution of each component is represented by a linear combination of local RV and n common RVs. Hence, the distribution of each component of 2-GMM M must be represented by such a linear combination in order to repeat statistical operations for GMMs. Namely, the RV M_k representing the distribution of the k-th component of M must be represented by

$$M_k = \mu_{Mk} + s_{x,k}[M] \cdot x_M + \sum_{g=1}^{n} s_{g,k}[M] \cdot r_g. \qquad (26)$$

To do so, we must determine the sensitivities $s_{x,k}[M] = C[M_k, x_M]$ and $s_{g,k}[M] = C[M_k, r_g]$.

If M_k is represented by Eq.(26), then the variance $V[M_k]$ must satisfy Eq.(12). However, $V[M_k] = \sigma_{Mk}{}^2$ is already determined by Eq.(18) so as to match the moments of D and M. Hence, sensitivities must satisfy

$$V[M_k] = s_{x,k}[M]^2 + \sum_{g=1}^{n} s_{g,k}[M]^2 = \sigma_{Mk}{}^2, \qquad (27)$$

from which we have the following inequalities for k = 1,2,

$$\sum_{g=1}^{n} s_{g,k}[M]^2 \leq \sigma_{Mk}{}^2. \qquad (28)$$

Moreover, since the covariance values $C_i[D, r_g] = s_{g,i}[D]$ (1≤i≤m, 1≤g≤n) of m-GMM D are given, we would like to match covariance values for 1≤g≤n as follows:

$$\sum_{k=1}^{2} P_{Mk} \cdot s_{g,k}[M] = C[D, r_g] = \sum_{i=1}^{m} P_i \cdot s_{g,i}[D]. \qquad (29)$$

However, it is not always possible to satisfy both Eqs.(28) and (29). A necessary and sufficient condition for sensitivities $s_{g,k}[M]$ to satisfy both Eqs.(28) and (29) is to satisfy Eq.(30) for k = 1,2, and Eq.(31):

$$\left(\sum_{g=1}^{n} C[D, r_g]\right)^2 + n \cdot \left(\sigma_{Mk}{}^2 - \sum_{g=1}^{n} C[D, r_g]^2\right) \geq 0, \qquad (30)$$

$$\sum_{g=1}^{n} C[D, r_g] \leq P_{M1} \cdot \sqrt{DSC_1} + P_{M2} \cdot \sqrt{DSC_2}, \qquad (31)$$

where DSC_k is the left hand side of Eq.(30). If this condition is not satisfied, then we cannot find sensitivities satisfying both Eqs.(29) and (30). If it is impossible, we change value of $C[D, r_g]$ to $\hat{C}\langle D, r_g\rangle$ so as to satisfy Eq.(29). Note here that since we keep the values $\sigma_{Mk}{}^2$ (k=1,2), the distribution shape of M does not change, and so does not NISE of M.

Since it is not easy to find an appropriate value $\hat{C}\langle D, r_g\rangle$ for each r_g, we use a sufficient value $\hat{C}\langle D, r_g\rangle$ by multiplying modification factor γ as shown in the following.

<Calculation of Sensitivities>

1°: Compute the following modification factor γ:

$$\gamma = \mathrm{Min}\left[\sqrt{\frac{\sigma_{M1}{}^2}{\sum_g C[D, r_g]^2}}, \sqrt{\frac{\sigma_{M2}{}^2}{\sum_g C[D, r_g]^2}}\right]. \qquad (32)$$

2°: For each r_g, if $\gamma < 1$, set $\hat{C}\langle D, r_g\rangle = \gamma \cdot C[D, r_g]$, otherwise, set $\hat{C}\langle D, r_g\rangle = C[D, r_g]$.

3°: For k = 1 and 2, compute the upper limit $UL_k \geq 1$ of $s_{g,k}[M]$ by

$$UL_k = \sqrt{\frac{\sigma_{Mk}{}^2}{\sum_g \hat{C}\langle D, r_g\rangle^2}}. \qquad (33)$$

4°: For each r_g, compute the target value $Trgt_g$ of the ratio $s_{g,1}[M]/C[M, r_g]$ of the covariance of the 1st component of M to the total covariance by

$$Trgt_g = \frac{C[D_{U13}, r_g]}{C[D, r_g]}, \qquad (34)$$

where D_{U13} is the RV of distribution generated from the components of D contained in U1 and a half of each component in U3.

5°: If $P_{M1} \cdot Trgt_g \leq 1 - P_{M2} \cdot UL_2$, then set weight factor $w1_g = 1 - P_{M2} \cdot UL_2$; if $P_{M1} \cdot UL_1 < P_{M1} \cdot Trgt_g$, then set $w1_g = P_{M1} \cdot UL_1$; otherwise (that is, if $1 - P_{M2} \cdot UL_2 \leq P_{M1} \cdot Trgt_g \leq P_{M1} \cdot UL_1$), then set $w1_g = P_{M1} \cdot Trgt_g$.

6°: For each r_g, determine sensitivities by the following equation:

$$\begin{cases} s_{g,1}[M] = \frac{w1_g}{P_{M1}} \cdot \hat{C}\langle D, r_g\rangle \\ s_{g,2}[M] = \frac{1 - w1_g}{P_{M2}} \cdot \hat{C}\langle D, r_g\rangle \end{cases}. \qquad (35)$$

7°: For k = 1 and 2, determine sensitivities of local RV by the following equation:

$$s_{x,k}[M] = \sigma_{Mk}{}^2 - \sum_{g=1}^{n} s_{g,k}[M]^2. \qquad (36)$$

V. Experimental Results

In order to evaluate the performance of the proposed algorithm consisting of <2-GMM Approximation>, <PDF-Check>, and <Calculation of Sensitivities>, we

TABLE 1: Parameters of D_A.

	D_{A1}	D_{A2}
P_{Aj}	$P_{A1} = 0.25, 0.5, 0.75$	$P_{A2} = 1 - P_{A1}$
μ_{Aj}	$\mu_{A1} = -1.5$	$\mu_{A2} = 3 \cdot \sigma_{A1} - \mu_{A1} - 3 \cdot \sigma_{A2}$
σ_{Aj}	$\sigma_{A1} = \sqrt{2}$	$\sigma_{A2} = 1.0, \quad \sqrt{2}$
s_g	$s_g[D_{A1}] = 1$	$s_g[D_{A2}] = \sqrt{\sigma_{A2}{}^2/(a_A{}^2 + 1)}$
s_x	$s_x[D_{A1}] = 1$	$s_x[D_{A2}] = a_A \cdot s_g[D_{A2}]$

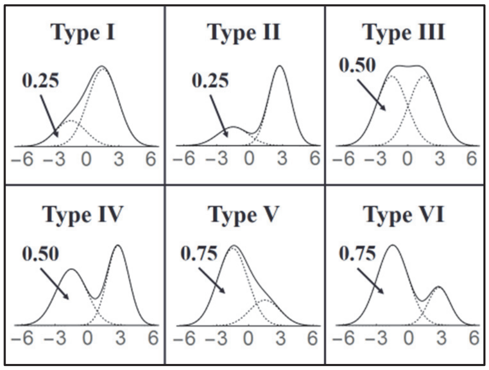


Fig.3 – Six types of PDF shape of D_A.

generated 8-GMM $D_M = \mathrm{Max}[D_A, D_B]$ from 2-GMM D_A and D_B such that

$$D_{Aj} = \mu_{Aj} + s_x[D_{Aj}] \cdot x_A + s_g[D_{Aj}] \cdot r, \tag{37}$$

$$D_{Bh} = \mu_{Bh} + s_x[D_{Bh}] \cdot x_B + s_g[D_{Bh}] \cdot r, \tag{38}$$

where $j, h \in \{1,2\}$. The 8-GMM is generated by the method in [4], and each component of the 8-GMM corresponds to a part of a component of JPDF of D_A and D_B contained in the area of $D_A \geq D_B$ or $D_A < D_B$.

The parameters of D_A are shown in TABLE 1, where the parameters of the 1st component exclusive of the mixture proportion are fixed. By setting μ_{A2} to the value shown in the table, D_A distributes around 0. In order to generate various distribution shapes, we used 3 values as the mixture proportion P_{A1} and 2 values as the standard deviation σ_{A2} of the 2nd component, which are shown in the table. Hence, we have 6 different distributions of D_A, as shown in Fig.3.

We introduced the ratio a_A of sensitivities of the 2nd components, whose value does not affect the shape of distribution, but change the correlation between D_A and D_B. We used the following 5 values as a_A, and hence we have totally 30 different distributions of D_A:

$$a_A = \quad 0.1, \quad 0.5, \quad 1.0, \quad 2.0, \quad 10.0 \tag{39}$$

TABLE 2: Parameters of D_B.

<table>
<tr><td>P_{B1}</td><td colspan="20">0.3</td><td colspan="20">0.5</td><td colspan="20">0.7</td></tr>
<tr><td>μ_{B1}</td><td colspan="20">−2.5</td><td colspan="20">−1.5</td><td colspan="20">−0.5</td></tr>
<tr><td>σ_{B1}</td><td colspan="15">2.0</td><td colspan="15">1.5</td><td colspan="15">1.0</td><td colspan="15">0.5</td></tr>
<tr><td>b</td><td colspan="10">0.6</td><td colspan="10">0.8</td><td colspan="10">1.0</td><td colspan="10">1.2</td><td colspan="10">1.4</td><td colspan="10">1.6</td></tr>
<tr><td>a_{B1}, a_{B2}</td><td colspan="12">0.1</td><td colspan="12">0.5</td><td colspan="12">1.0</td><td colspan="12">2.0</td><td colspan="12">10.0</td></tr>
</table>

Since D_A distributes around 0, we set $E[D_B] = 0$ by setting μ_{B2} to

$$\mu_{B2} = \mu_{B1} \cdot P_{B1}/(P_{B1} - 1). \tag{40}$$

Moreover, in order to generate various distributions for D_B, we introduced ratio b of standard deviations of the 1st and the 2nd components, and ratio a_{Bh} (h = 1,2) of sensitivities of the h-th component. Then, we have

$$b = \frac{\sigma_{B1}}{\sigma_{B2}} = \frac{s_g[D_{B1}] \cdot \sqrt{a_{B1}{}^2 + 1}}{s_g[D_{B2}] \cdot \sqrt{a_{B2}{}^2 + 1}}. \tag{41}$$

Therefore, by setting the values of the mixture proportion P_{B1}, the mean μ_{B1} and the standard deviation σ_{B1} of the 1st component, and the ratios b and a_{Bh} (h = 1,2), we can determine the distribution of D_B. The values that we used for these parameters are shown in TABLE 2. Therefore, we have totally 5,400 different distributions of D_B, and the correlation coefficient between D_A and D_B varies between 0.00173 and 0.668.

From these D_A and D_B, we generated totally 162,000 distributions of 8-GMM D_M, and applied Method-Cdf, Method-Mergd, Method-Opt, and the proposed method. TABLE 3 shows the average and the maximum of NISE ε_M of 2-GMM M and the percentage N_{bad} and N_{good} of cases when $\varepsilon_M > 0.1$ and $\varepsilon_M \leq 1.1 \cdot \varepsilon_{Opt}$, respectively, in 162,000 total cases, where ε_{Opt} is NISE of 2-GMM obtained by Method-Opt. Since the maximum of NISE obtained by Method-Opt is smaller than 0.1, N_{bad} can be regarded as the rate of bad approximation. On the other hand, N_{good} is regarded as the rate of good approximation.

From the table, we can see that the worst-case error (Maximum ε_M) of the proposed method is improved by 46.1% from Method-Cdf, and the average error of ε_M by 43.2%. Moreover, N_{bad} decreased by 3.44 points and N_{good} increased by 5.8 points. Hence, we can see that the proposed method could cover the defect of Method-Cdf by introducing Gaussian reduction technique in Method-Mergd.

The improvements of the proposed method from Method-Mergd is not large, and N_{good} decreased by 4.1 points from Method-Mergd. However, the average error of ε_M is improved by 3.36%, which means that the proposed method can take advantage of Method-Cdf. Namely, as shown in Fig. 2, Method-Cdf can find a better 2-GMM than Method-Mergd, and the proposed method uses this advantage. However, in 16,330 cases (10.1%), the proposed algorithm changed good 2-GMMs obtained by Method-Cdf by worse 2-GMMs by conducting Method-Mergd. This means that the way to select Method-Mergd in the proposed method is not sufficient and there still exists room for improvement in the proposed method.

TABLE 3: NISE ε_M 2-GMM M.

	Cdf	Mergd	proposed	Opt
Average $\varepsilon_M \cdot 10^3$	16.2	9.52	9.20	6.03
Maximum ε_M	0.271	0.146	0.146	0.083
N_{bad} [%]	3.80	0.357	0.357	0
N_{good} [%]	31.7	41.5	37.4	---

TABLE 4: CPU times [μs]

Cdf	Mergd	proposed			
		Path 1°	Path 2°	Path 3°	average
1.10	3.30	0.115	2.18	5.50	2.60

TABLE 5: Percentage of Path with respect to type of D_{A} [%].

Type	I	II	III	IV	V	VI	total
Path 1°	4.60	7.03	0.795	0.059	3.60	1.29	17.4
Path 2°	9.27	6.53	12.4	10.2	9.60	10.3	58.2
Path 3°	2.80	3.11	3.50	6.42	3.47	5.09	24.4

TABLE 4 shows the CPU times to find a 2-GMM by Method-Cdf, Method-Mergd, and the proposed method. In the table, columns Path 1°, 2°, and 3° show the CPU times for a 2-GMM to be obtained at Step 1°, 2°, and 3° in <2-GMM Approximation>, respectively, and column average shows average CPU time of 162,000 cases. Although the CPU times needed for a single 2-GMM approximation of these methods are small, if the approximation is repeated many times in applications such as the statistical static timing analysis of LSI, then the difference of the times becomes important. The machine used for experiments has Intel Xeon processor E3-1270 v2 with 3.5GHz clock frequency and 16GB memory.

TABLE 5 shows the percentage of each Path in which 2-GMMs are generated. Moreover, it shows the percentages of each Path with respect to the distribution type of D_{A} shown in Fig.3. Since the ratio of Path 1° is 17.4%, the average CPU time of the proposed method is reduced from Method-Mergd by 21.2%.

VI. Conclusions

In this paper, we proposed a new algorithm for approximating a given m-GMM by a 2-GMM with small NISE (normalized integral square error) of PDF. The algorithm combines three methods so as to minimize NISE and CPU time, and keeps the shape of distribution (the variance of each component) for not changing NISE when determining sensitivities. From the experiments, we found that the proposed method can take the advantages of the previous algorithm; Method-Cdf and Method-Mergd, and shows better performance from the viewpoints of NISE and CPU time.

In the proposed algorithm, Method-Mergd is selected for producing better approximation by using threshold $T\varepsilon$ and NISE of 2-GMM PDF obtained by Method-Cdf. However, this selection method is not perfect, and hence our future work is to find a better selection method, which invokes Method-Mergd only when it finds better solution than Method-Cdf, and overhead of CPU time for selection is less than the CPU time of Method-Cdf.

Moreover, the proposed algorithm does not change variance of each component when determining sensitivities, so as to keep NISE of 2-GMM. However, there exist applications in which such objective function may be inappropriate. For example, in the lifetime analysis of a battery pack, minimum operation may be repeated many times[5], and in such a case, minimizing NISE may be inappropriate. Therefore, devising an algorithm suitable to such an application and considering an appropriate objective function are another future work.

Acknowledgements

This research was partially supported by Grant-in-Aid for Scientific Research (C) 15K06031.

References

[1] A.Srivastava, D.Sylvester, and D.Blaauw, Statistical Analysis and Optimization for VLSI: Timing and Power, Springer, 2005.

[2] D.Blaauw, K.Chopra, A.Srivastave, L.Scheffer, “Statistical timing analysis: From basic principles to state of the art,” IEEE Trans. CAD/ICAS, vol.27, no.4, pp.589-607, 2008.

[3] S.Takahashi and S.Tsukiyama, “A new statistical timing analysis using Gaussian mixture models for delay and slew propagated together,” IEICE Trans. Fundamentals, vol.E92-A, no.3, pp.900-911, 2009.

[4] S.Tsukiyama and M.Fukui, “A statistical maximum algorithm for Gaussian mixture models considering the cumulative function curve,” IEICE Trans. Fundamentals, vol. E94-A, no.12, pp.2528-2536, 2011.

[5] D.Sasaki, S.Tsukiyama, M.Matsunaga, O.Ishibashi, S.Takahashi, “A statistical method for analyzing lifetime of a battery pack,” IEEE Power & Energy Society General Meeting, paper no. PESGM2015-001311, 2015.

[6] D.J. Salmond, “Mixture reduction algorithms for target tracking in clutter,” Proc. Signal and Data Processing of Small Targets, vol.1305, pp.434–445, 1990.

[7] J.L. Williams and P.S. Maybeck, “Cost-function-based Gaussian mixture reduction for target tracking,” Proc. 6th Int. Conf. on Information Fusion, pp.1047–1054, 2003.

[8] A. R. Runnalls, “Kullback-Leibler approach to Gaussian mixture reduction,” IEEE Trans. Aerosp. Electron. Syst., vol.43, no.3, pp.989–999, 2007.

[9] M.F. Huber and U.D. Hanebeck, “Progressive Gaussian mixture reduction,” Proc. 11th Int. Conf. on Information Fusion, 2008.

[10] D. Schieferdecker and M.F. Huber, “Gaussian mixture reduction via clustering,” Proc. 12th Int. Conf. on information Fusion, pp.1536–1543, 2009.

[11] G. Valverde, J.Q. Tortos, and V. Terzija, “Comparison of Gaussian mixture reductions for probabilistic studies in power systems,” IEEE Power and Energy Society General Meeting, 2012.

[12] A. Ogawa and S. Takahashi, “Weighted distance measures for efficient reduction of Gaussian mixture components in HMM-based acoustic model,” Proc. Int. Conf. on Acoustics, Speech, and Signal Processing, pp. 4173–4176, 2008

[13] N.Yokoyama, S.Tsukiyama, and M.Fukui, “An algorithm to reduce components of a Gaussian mixture model considering distribution shape of each component,” IEICE Tech. Report, CPSY2014-170, vol.114, no.506, pp.49-54, 2015 (in Japanese).