Worst-Case Eye Analysis of High-Speed Channels Based on Bayesian Optimization

Majid Ahadi Dolatsara, *Student Member, IEEE*, Jose Ale Hejase, *Member, IEEE*, Wiren Dale Becker, *Fellow, IEEE*, Jinwoo Kim, *Student Member, IEEE*, Sung Kyu Lim, *Senior Member, IEEE*, and Madhavan Swaminathan, *Fellow, IEEE*

***Abstract*—One of the favorable tools for signal integrity evaluation is eye diagram analysis. This is traditionally performed with a lengthy transient simulation, which can be prohibitively time consuming for complex high-speed channels with a low bit error rate. Methods for eye estimation exist; however, they are either only applicable to linear time-invariant systems or have lack in accuracy or efficiency. In this article, an optimization-based approach is proposed to quickly obtain the worst-case eye diagram characteristics. This approach focuses on the inter-symbol interference since its effect can span over many symbols and include crosstalk, making it challenging to model. In this article, the data patterns leading to the lowest voltage corresponding to a high symbol, the highest voltage corresponding to a low symbol, and the times of minimum and maximum level crossing points are calculated. Then, eye height, eye width, and the worst-case eye opening are estimated using these points. To reduce complexity, the proposed approach includes a mapping algorithm that exploits the Gray code. Additionally, Bayesian optimization is used because of its efficiency and good performance on non-linear and non-convex problems. Finally, the application of the proposed approach to high-speed SerDes channels, and channels in system-on-package designs is evaluated with numerical examples, where the results show its accuracy and efficiency.**

***Index Terms*—Bayesian optimization (BO), crosstalk, eye diagram, gray code, high-speed channels, intersymbol interference, machine learning, SerDes, signal integrity.**

I. Introduction

WHEN a signal passes through a high-speed channel it encounters timing jitter and amplitude noise. With the exponential increase of bitrate in recent years, it has become considerably harder to avoid communication failure due to such interferences. Therefore, designers rely on rigorous modeling and simulation in the early stages of design of high-speed channels to predict the jitter and noise. A common analysis to evaluate the quality of the signal is the eye diagram analysis. However, this analysis requires lengthy transient simulation, leading to exorbitantly high computation time and large memory. For instance, for a channel that has a bit error rate (BER) equal to 10^{-12}, simulating transmission of 10^{12} bits is required to determine if the transmission of one bit fails on average. Therefore, in this article, we suggest a novel approach that exploits the recent advancements in machine learning and applies an optimization algorithm to quickly estimate the worst-case eye characteristics. Here the worst-case eye represents the closest eye that can be achieved by a lengthy transient simulation. For the rest of the article, the method presented is referred to as Worst-eye.

Jitter and noise are complex signals and are caused by various sources. Hence, their different components need to be considered for modeling. Generally, jitter and noise can be classified into random and deterministic types. The former can be caused by intrinsic sources in the device and modeled with Gaussian distributions. Cause of the latter can be design-related sources such as crosstalk, reflection, ground bounce, etc. The deterministic category includes the data-dependent noise and jitter (DDN and DDJ), which can be caused by the inter-symbol interference (ISI) [1]. For modern high-speed channels with high bitrates, it can be challenging to estimate the effect of ISI since the pulse response can span over several unit intervals (UIs). Therefore, in this article, we focus on finding the bit patterns causing the worst ISI signals by searching the random space of several previous bits. In addition, the effect of crosstalk on the final eye diagram is predicted. It is worth mentioning that the proposed approach can be integrated with established methods of estimating other types of jitter and noise (e.g., random jitter and noise) during post processing.

In the literature several eye estimation methods have been suggested [2]–[18]. Eye estimation methods have been proposed that attempt to analytically estimate a channel's eye characteristics [2]–[4]. However, analytical methods are hard to generalize and require simplifications and expert knowledge. Furthermore, to reduce the computational costs, statistical eye analysis methods have been developed. In the peak distortion analysis (PDA) method [5], first a single pulse response is obtained. Then the

Manuscript received March 30, 2020; revised June 12, 2020; accepted July 21, 2020. This work was supported in part by the DARPA CHIPS Project under Award N00014-17-1-2950, and in part by the National Science Foundation under Grant CNS 16-24731—Center for Advanced Electronics through Machine Learning and its industry members. *(Corresponding author: Majid Ahadi Dolatsara.)*

Majid Ahadi Dolatsara and Madhavan Swaminathan are with the School of Electrical and Computer Engineering, 3D Systems Packaging Research Center, Georgia Institute of Technology, Atlanta, GA 30332 USA (e-mail: mad6@gatech.edu; madhavan.swaminathan@ece.gatech.edu).

Jose Ale Hejase is with IBM, Austin, TX 78758 USA (e-mail: jhejase@ieee.org).

Wiren Dale Becker is with IBM, New York, NY 10022 USA (e-mail: wbecker@us.ibm.com).

Jinwoo Kim and Sung Kyu Lim are with the School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA 30332 USA (e-mail: jinwookim@gatech.edu; limsk@ece.gatech.edu).

Color versions of one or more of the figures in this article are available online at https://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TEMC.2020.3012960

lowest high and the highest low received symbols are found by superposition of shifted copies of the pulse response. This procedure can be repeated at multiple sampling time points to estimate the worst-case eye. Moreover, StatEye [6] extends the superposition idea by statistically calculating the PDF of noise and jitter on the eye, caused by all possible combinations of data. Then, BER and the bathtub curve are derived from the PDFs. Although PDA and StatEye are fast, they are only applicable to linear time-invariant (LTI) systems because they estimate the response of a pulse sequence by superposition of the shifted single pulse responses. Unfortunately, high-speed channels can be non-LTI due to the presence of nonlinear IO drivers, nonlinear receivers, low compression point of the receiver, and use of single-ended signaling. Particularly, single-ended signaling appears in DDR memory systems, and it can cause asymmetric rising and falling edges, showing that the system is non-LTI. To address this issue, Tsuk *et al*. [7] find distributions of noise and jitter from the superposition of a single rising edge, a single falling edge, and their shifted duplicates. It is worth noting that the edge responses are not independent of each other; hence, Tsuk *et al*. [7] develop a relatively complex inductive technique to estimate the distribution of the receiver voltage. Although using the two edge responses increases the accuracy, it is not always enough because different variations of rising and falling edges might exist based on their previous bits. Therefore, Ren and Oh [8] suggest using all possible edge responses that show considerable differences. For instance, if M previous bits are considered, 2^M edge responses are generated and used to obtain the statistical eye. Using multiple edge responses improves the accuracy of the statistical eye diagram for systems with asymmetric rising and falling edges; nevertheless, the improvements are limited since many edge responses might be required for complex cases. Besides, [7] and [8] are still expansions of superposition and can fail for systems with other sources of nonlinearity. Finally, expert knowledge is needed to find the cause of being non-LTI and the edge responses.

To reduce the computational costs, researchers have developed surrogate models of high-speed channels' components, which can be used for eye analysis. A popular candidate for this task is neural networks (NNs). In [9] and [10], recurrent NNs are used to model nonlinear I/O drivers, and in [11] recurrent NNs are used to model SerDes channels. However, training NNs can be complicated and time consuming. Moreover, they can accumulate error over time. Also, they can become significantly large for channels with a long memory. Alternatively, we therefore recently suggested lighter surrogate models [12] based on the Polynomial Chaos theory [13]. These models reduce the computational cost, and as a bonus, they provide the statistical moments with closed-form equations as well. Nevertheless, surrogate models can introduce inaccuracy as a result of simplifications and still be computationally expensive. Therefore, the development of more advanced methods is necessary.

Moreover, Evolutionary optimization algorithms have been used to find the bit patterns that correspond to the worst-case eye, where Ulrich *et al*. [16] and Singh *et al*. [17] use the genetic algorithm and the particle swarm optimization, respectively. However, evolution-based optimization methods have shown to have low convergence rates; hence, the physical simulation costs can be cumbersome.

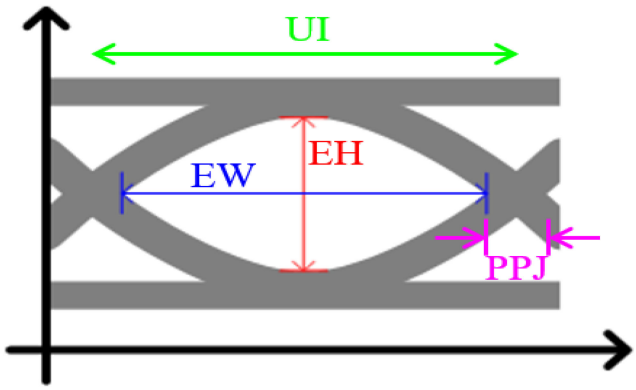


Fig. 1. EH, EW, PPJ, and UI shown on an eye diagram.

To alleviate the high computational costs, in this article, we propose the worst-eye approach. In this method, initially, the search space is mapped to a low-dimensional space using the reflected binary codes (Gray codes) [19] to reduce the dimensionality and sparsity. Then, after preparations based on the domain knowledge, Bayesian optimization (BO) [20] is applied to find the bit patterns resulting in the waveforms that pass through the points defining eye height (EH) and eye width (EW) as shown in Fig. 1. BO is selected for this approach because the objective functions are nonlinear and nonconvex. Therefore, the classic optimization algorithms are not suitable. The worst-case eye is estimated by overlaying the resulting waveforms. The proposed worst-eye is advantageous compared to earlier works that search for the worst-case eye opening by means of the evolutionary algorithms [16], [17] since BO's convergence rate is proven to be faster. For instance, a comparison of BO and the genetic algorithm has been previously done in [21], where the genetic algorithm needed more than 26× samples compared to BO for optimization of a complex function. In addition, worst-eye includes a mapping algorithm and domain knowledge considerations that further improve the convergence rate compared to [16] and [17].

It is worth noting that BO-based methods have been previously used in the optimization of electronic designs [21]–[23]. In [21], BO is used to minimize the clock skew rate in 3-D systems by co-optimization of thermal and electrical performance. The input parameters in this article are geometrical and physical characteristics of the materials used in the 3-D system such as thermal conductivity of the PCB, and thermal conductivity and thickness of the thermal interface material. In [21], the BO algorithm finds the optimal input values that result in the minimum clock skew rate. However, in this article, we efficiently estimate the eye diagram for a fixed design. Here, the input parameters are the data patterns, and the proposed approach finds the patterns that result in the worst signal degradation to estimate the worst-case eye characteristics. In addition, Torun *et al*. [22] and Torun and Swaminathan [23] suggest variations of the BO algorithm to optimize design characteristics by finding optimal physical parameters, similar to [21]. Moreover, Torun

TABLE I
BINARY NUMBERS WITH 3 BITS AND THEIR CORRESPONDING GRAY CODES

	Binary	**Gray**
0	000	000
1	001	001
2	010	011
3	011	010
4	100	110
5	101	111
6	110	101
7	111	100

et al. [24] suggest a BO-based approach to determine physical variables that produce the closest eye, where the closest eye refers to the worst signal quality caused by changing the physical variables. The novelty of the proposed approach is in finding the data patterns and waveforms causing the worst-case eye characteristics. In other words, the goal of this approach is to reduce the computational costs of finding the eye diagram for a set of fixed physical parameters. In [25], we briefly discussed this idea for the worst-case EH, which did not include the Gray code in its mapping. However, in this article, the new mapping scheme is introduced, which shows better performance. In addition, sampling is done more efficiently to cover a larger area of the search space. Moreover, the worst-case EW and worst-case eye opening are calculated. Finally, the proposed approach is expanded to include crosstalk.

The remainder of this article is organized as follows. In Section II, background information regarding the Gray code and BO are reviewed. In Section III, the proposed worst-eye approach is discussed in detail. Next, in Section IV, worst-eye is evaluated by three numerical examples. Finally, Section V concludes this article.

II. BACKGROUND REVIEW

A. Reflected Binary Code (Gray Code)

The reflected binary code or the Gray code, originally introduced by Frank Gray in 1947 [26], refers to a reordering of the binary numbers where each two subsequent values only differ in a single binary bit [19]. Gray code is particularly helpful in digital communication since it reduces the potential errors in reading the successive binary values. For instance, in binary numbers "0111" is followed by "1000," which means every bit is inverted. However, in the Gray code sequence, only one bit switches at a time, reducing the chances of error. The binary numbers with 3 b and their corresponding Gray code are shown in Table I. Gray codes can be generated recursively from the list of Gray codes with one bit, which is $\{0, 1\}$. To find the Gray codes with k bits, the Gray codes with $k-1$ bits are listed. Then, the list is mirrored resulting in a second list in the reverse order. Next, members of the first list are prefixed with a 0, and members of the second list are prefixed with a 1. Finally, the two lists are concatenated. Alternatively, the Gray codes can be generated directly. The decimal number corresponding to the *k*th Gray code is generated as [27]

$$k \wedge (k \gg 1) \tag{1}$$

where $\wedge$ and $\gg$ are bitwise exclusive OR and shift-right operators, respectively. In this work, we have used (1) since it is more efficient than the recursive method.

B. Bayesian Optimization

BO [20], [28] is an active learning optimization algorithm that has shown a good performance in optimizing complex nonconvex and nonlinear functions. Here, we discuss maximizing a black-box function $f(x)$; nevertheless, minimizing is done similarly. This technique is inspired by the Bayes' theorem since an estimated prior distribution is assigned to $f(x)$; then the system is evaluated at a new sample point, and a posterior distribution is determined as

$$P\left(f\left(x\right)|D_{1:t}\right) \propto P\left(D_{1:t}|f\left(x\right)\right)P\left(f\left(x\right)\right) \tag{2}$$

where $D_{1:t} = \{x_{1:t},\ f_{1:t}\}$ is the set of t samples and evaluations. $P(f(x))$ and $P(f(x)|D_{1:t})$ are prior and posterior distributions, respectively. Moreover, $P(D_{1:t}|f(x))$ is the likelihood of observing $D_{1:t}$, given our prior belief of $P(f(x))$. In other words, BO corrects our belief on the distribution of the objective function. BO is an iterative algorithm, which adds a new sample point at each iteration and updates the posterior distribution. The algorithm continues until it converges to the true value of $\max(f(x))$. To find the next sample point, BO estimates mean $(\mu(x))$, and standard deviation $(\sigma(x))$ of the objective function using $D_{1:t}$. Then, the next sample is selected by maximizing an acquisition function, which is a function of $\mu(x)$ and $\sigma(x)$. The acquisition function is responsible for balancing exploitation and exploration. Exploitation is setting the next point where mean is high to reach the maximum in that area. On the other hand, exploration is probing areas where the variance is high to find the global maximum. A popular choice for determining the posterior is the Gaussian process (GP). In BO, the system is initially sampled at a number of points, and the prior is set as a normal distribution: $f_t \sim \mathcal{N}(\mu, \mathbf{K})$, with μ and $\mathbf{K}$ representing the mean and covariance matrix, respectively. For convenience, μ is initially set to zero. Moreover, $\mathbf{K}_{i,j} = k(x_i, x_j)$, with k being the kernel function, and $1 \leq i, j \leq t$. A suitable choice for the kernel which we have used in this article is the Matérn function with smoothness factor of 2.5 [29]

$$k\ (r) = \left(1 + \frac{\sqrt{5}r}{\lambda} + \frac{5r^2}{3\lambda^2}\right)\ \exp\left(-\frac{\sqrt{5}r}{\lambda}\right) \tag{3}$$

where $r\ = \|x_i - x_j\|$ and λ is an optimizable length factor (see the documentation in [30]). Using this kernel, posterior distribution of f at $t+1$ is calculated as

$$\begin{aligned} &P\ \left(f_{t+1}|D_{1:t}, x_{t+1}\right) = \mathcal{N}\left(\mu_{t+1}, \sigma^2_{t+1}\right) \\ &\mu_{t+1} = \vec{K}^T \boldsymbol{K}^{-1} f_{1:t}, \quad \sigma^2_{t+1} = k\left(x_{t+1}, x_{t+1}\right) - \vec{K}^T \boldsymbol{K}^{-1} \vec{K} \end{aligned} \tag{4}$$

where $\vec{K} = [k(x_1, x_{t+1}),\ k(x_2, x_{t+1}), \ldots, k(x_t, x_{t+1})]$. We can assume GP is a surrogate model of $f(x)$ that yields its

mean and variance. Additionally, these values are used in the acquisition function, to determine the next sample point (i.e., x_{t+1}). Common acquisition functions are the probability of improvement (PI), expected improvement (EI), and upper confidence bound (UCB), which are defined as

$$\begin{aligned} \mathrm{PI}\ (x) &= P\ \left(f\left(x\right) \geq f\left(x^{+}\right) + \varepsilon\right) \\ &= \Phi\left(\frac{\mu\left(x\right) - f\left(x^{+}\right) - \varepsilon}{\sigma\left(x\right)}\right) \end{aligned} \tag{5}$$

$$\mathrm{EI}\ (x) = \begin{cases} \left(\mu\left(x\right) - f\left(x^{+}\right) - \varepsilon\right)\Phi\left(Z\right) \\ +\sigma\left(x\right)\phi\left(Z\right) & \text{if } \sigma\left(x\right) > 0 \\ 0 & \text{Otherwise} \end{cases}$$

$$Z = \frac{\mu\left(x\right) - f\left(x^{+}\right) - \varepsilon}{\sigma\left(x\right)} \tag{6}$$

$$\mathrm{UCB}\ (x) = \mu\left(x\right) + \kappa\sigma\left(x\right) \tag{7}$$

where $\Phi(.)$ and $\phi(.)$ show CDF and PDF of the standard normal distribution, respectively. $f(x^{+})$ is the maximum value observed so far. In addition, $\varepsilon \geq 0$ and $\kappa \geq 0$ are hyperparameters to balance exploration and exploitation. x_{t+1} is found as $x_{t+1} = \mathrm{argmax}_x\, u(x|\ D_{1:t})$, with $u(.)$ being one of the acquisition functions. To further improve the convergence of BO, we use the GP_Hedge acquisition function, which probabilistically selects one of the above three functions at each iteration [31]. It is worth mentioning that BO is a repetitive approach, meaning it reproduces the same results given the same starting random points. This has been tested in the numerical examples in this article.

III. Proposed Worst-Eye Approach

A. Intuition

Although for a comprehensive examination of the signal, the distributions of noise and jitter is required, engineers often use EH and EW for a quick evaluation of the channel. Therefore, it would be immensely helpful to find these values without simulating all possible bit pattern combinations. As shown in Fig. 1, EH is the difference between the lowest high and the highest low received symbols at the sampling time. Moreover, EW is equal to a UI minus the peak to peak jitter (PPJ). Note that the PPJ is difference of the rightmost and the leftmost level-crossing time points, caused by the rising and falling edges. Since in this article, we are only concerned with DDN, DDJ, ISI, and crosstalk, one can use optimization techniques to find the sequences of symbols that result in the waveforms passing through the mentioned four points. Next, the worst-case eye is estimated by overlaying the four obtained waveforms. However, the estimated worst-case eye is reliable around the four worst-case points. To increase the accuracy, the number of the sampling time points, for evaluating the received symbol, is simply increased. Then, the additional lowest high and highest low points at the new sampling points are calculated, and the waveforms that pass through them are added on top of the previous waveforms to provide a better estimate of the worst-case eye. It is worth noting that the proposed approach is designed for nonreturn-to-zero (NRZ) pulses; however, the idea can be expanded to other signaling methods.

We acknowledge that BER is the ultimate measure for the overall evaluation of high-speed channels. However, the motivation for this work was a stage in the design process where the circuit is not fixed yet. Sometimes in the design process, we need to have an intermediate objective, which the EH and EW fulfills in this case. The channel designer needs to select the settings that maximize the eye opening to provide the maximum flexibility for the circuit designer.

Moreover, the focus of this article is on data dependent jitter and noise (DDJ and DDN). In the numerical examples, we show that DDJ and DDN do not necessarily cause a bit transmission to fail. The RJ and RN are a key part of calculating the BER. The worst-case BER can be calculated by superimposing the distribution of RJ and RN on the worst-case eye diagram reported by the proposed approach. By worst-case BER, we mean BER if we consider RJ and RN and only transmit the worst-case data pattern found in this article. Therefore, the final BER directly depends on the worst-case EH and EW. Hence, we need to maximize the eye opening when designing the channel to achieve a lower BER.

B. Mapping Scheme

The optimization suggested in the previous section is in fact a challenging problem because the search space is high-dimensional, discrete, and sparse. The cause of high-dimensionality is that the single pulse response of modern channels can span through several UIs (e.g., > 20). In addition, the search space is discrete and sparse because the value of each dimension can only be either zero or one.

To address these issues, we suggest reformulating the problem based on our knowledge of high-speed channels and ISI. We know that ISI is caused by a sequence of symbols that can be shown as

$$\boldsymbol{\lambda} = [\lambda_{-n}, \lambda_{-n+1}, \ldots, \lambda_{-1},\ \lambda_0,\ \lambda_1,\ \ldots,\ \lambda_m\ ] \tag{8}$$

where n precursors and m postcursors are considered. In addition, the size of the sample space $(\Omega_{\boldsymbol{\lambda}})$ is equal to 2^{n+m+1}. The goal of worst-eye is to determine such sequences resulting in the worst-case voltage and timing values. For simplicity, we assume that $m = 1$, which means the effect of the rest of the postcursors is negligible. In addition, the state of λ_1 is given based on the pulse response. For instance, the lowest high is always followed by a low symbol for the cases considered in this article. Intuitively, it can be said that the following low symbol pulls the high symbol down. Furthermore, it is known that the ISI from each symbol decreases as it gets further from the current symbol. Hence, the state of λ_{-n} has the least impact on noise and jitter, and the state of λ_{-1} has the most impact. By setting λ_{-n} as the least important bit and λ_{-1} as the most important bit, we obtain a unique binary index number, $I(\boldsymbol{\lambda})$, for each possible bit pattern. Note that $0 \leq I(\boldsymbol{\lambda}) < 2^n$. This format is shown in Fig. 2.

The next step would be to find the $I(\boldsymbol{\lambda})$ that minimizes or maximizes the objective functions. This technique maps

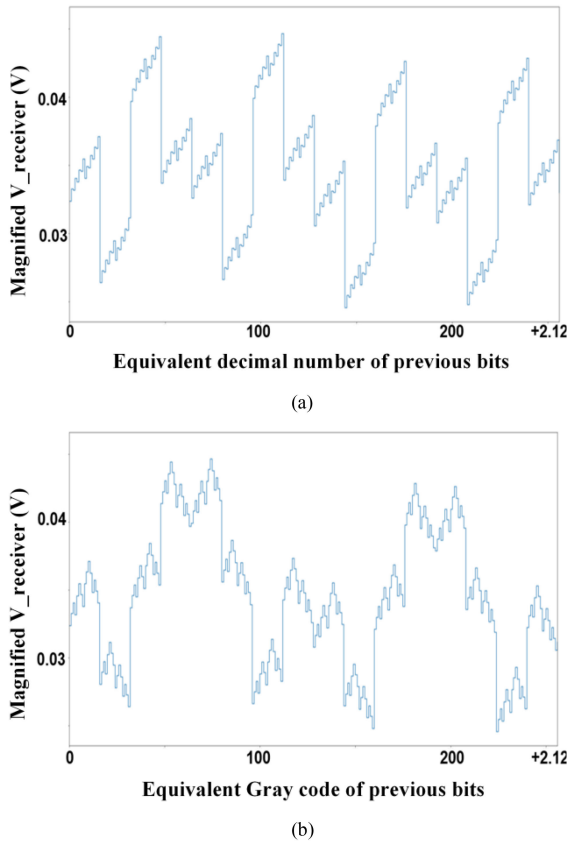


Fig. 2. Receiver voltage as a function of the index values corresponding to the previous bit patterns. (a) Binary order. (b) Gray code order.

Fig. 3. Indexing of the bit patterns for $n = 4$ in the binary format.

the problem from *n*-dimensions to one. However, as stated in Section II-A, there are sudden changes, including total inversions, in successive binary numbers. Subsequently, this causes sudden changes in the objective functions. To alleviate this issue, we suggest reordering the index numbers as Gray codes and generating a new index, labeled as $I_g(\boldsymbol{\lambda})$. For example, the values shown in Fig. 2 are reordered as {0000, 0001, 0011, 0010, 0110, …}. For comparison, an example of corresponding sections of $V_r(t_s, I(\boldsymbol{\lambda})|_{\lambda_0=1,\ \lambda_1=0})$ and $V_r(t_s, I_g(\boldsymbol{\lambda})|_{\lambda_0=1,\ \lambda_1=0})$ are illustrated in Fig. 3(a) and (b), respectively. Note that $V_r(t_s, i)$ is the receiver voltage caused by the bit pattern with index *i,* at the sampling time t_s. It is seen that the Gray code order leads to smaller jumps in the objective function. Additionally, our numerical tests have confirmed that the Gray code indices results in a faster convergence, which is a result of subsequent values being different in only one symbol.

Finally, it is worth noting that variations of the Gray code [19] and other coding schemes exist, which have characteristics similar to the ones discussed here. However, in this article, we have settled for the original Gray code [26] and did not experiment with other coding mechanisms.

C. Optimization

The mapping scheme results in objective functions that are suitable for BO. Therefore, we use the BO algorithm, reviewed in Section II-B, to find the lowest high symbol (V_{LH}), the highest low symbol (V_{HL}), the leftmost level crossing point (t_{LX}), and the rightmost level crossing point (t_{RX}). It is worth noting, when calculating V_{LH} and V_{HL}, we can further improve the convergence rate by manually setting value of λ_{-1} in the same manner that we assigned λ_1. In other words

$$V_{LH} = \min_{I_g} \ V_r\left(t_s, I_g\left(\boldsymbol{\lambda}\right)|_{\lambda_{-1}=0,\lambda_0=1,\ \lambda_1=0}\right) \tag{9}$$

$$V_{HL} = \max_{I_g} \ V_r\left(t_s, I_g\left(\boldsymbol{\lambda}\right)|_{\lambda_{-1}=1,\lambda_0=0,\ \lambda_1=1}\right). \tag{10}$$

On the other hand, t_{LX} and t_{RX} are the result of falling edges or rising edges. Therefore, both types of the edges need to be considered for either of them

$$t_{LX} = \min\left\{\begin{array}{l}\min_{I_g} \ t\left(V_0, I_g\left(\boldsymbol{\lambda}\right)|_{\lambda_{-1}=0,\ \lambda_0=1,\ \ \lambda_1=1}\right), \\ \min_{I_g} \ t\left(V_0, I_g\left(\boldsymbol{\lambda}\right)|_{\lambda_{-1}=1,\ \lambda_0=0,\ \ \lambda_1=0}\right)\end{array}\right\} \tag{11}$$

$$t_{RX} = \max\left\{\begin{array}{l}\max_{I_g} \ t\left(V_0, I_g\left(\boldsymbol{\lambda}\right)|_{\lambda_{-1}=0,\ \lambda_0=1,\ \ \lambda_1=0}\right), \\ \max_{I_g} \ t\left(V_0, I_g\left(\boldsymbol{\lambda}\right)|_{\lambda_{-1}=1,\ \lambda_0=0,\ \ \lambda_1=1}\right)\end{array}\right\} \tag{12}$$

where $0 \le t(V_0, i) < \mathrm{UI}$, and $t(V_0, i)$ is the time when the receiver voltage crosses the threshold voltage V_0, when applying the bit pattern marked by the index *i*. Additionally, λ_1 is determined intuitively, similar to the V_{LH} and V_{HL} cases. Furthermore, instead of performing two rounds of BO in (11) or in (12), we prefer to add a new symbol to the right side of the active symbols in $\boldsymbol{\lambda}$, which means it has the highest impact on the output. We name this symbol λ_j. If λ_j is low the first term in the brackets is chosen; otherwise, the second term is selected. It is worth mentioning that the minimums in (9) and (11) can be calculated by finding the maximum of the negative objective functions using BO.

Next, we introduce another technique to further improve the efficiency of worst-eye. It is observed that when *n* is too large, determination of the final *k* symbols (i.e., $-n$ to $-n + k - 1$) becomes much harder. Note that, the last *k* symbols represent 2^k patterns that share the remaining $(n + 2 - k)$ symbols. When *k* is small (e.g., $k \le 10$), often sweeping over these 2^k patterns can be quickly done in a single transient simulation by concatenating all the 2^k patterns. This simulation can be more efficient than including the final *k* symbols in BO. Therefore, in the proposed approach a BO sample does not include the final *k* symbols. The objective function receives this sample and sweeps over the final *k* symbols that can be added to the sample; then it reports the one yielding the maximum output. That is to say, in the finalized

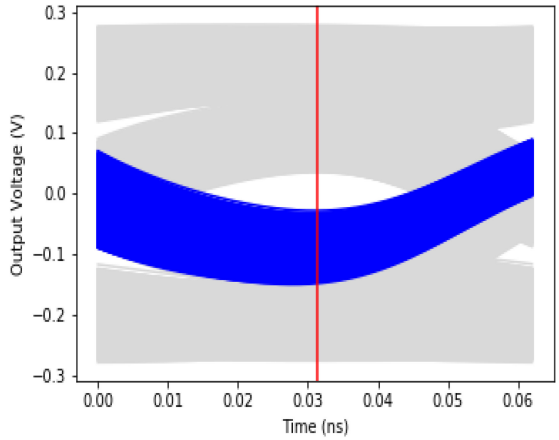


Fig. 4. Example eye diagram and the waveforms corresponding to the bit patterns ending in $\{\lambda_{-1}=1, \lambda_0=0, \lambda_1=1\}$, shown in blue.

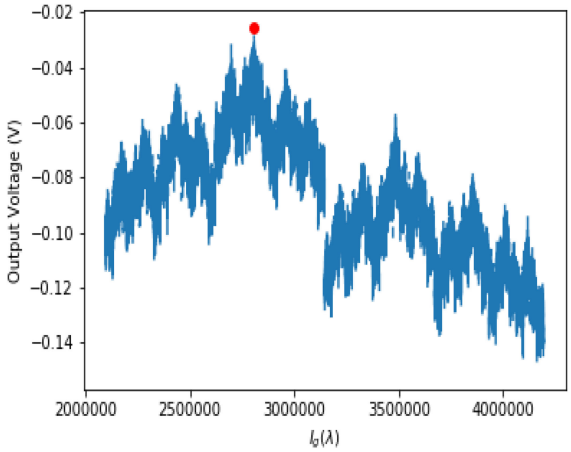


Fig. 5. Objective function which is passed to BO to find V_{HL} in Fig. 4.

approach, $\boldsymbol{\lambda}$ and size of the sample space of BO in (9) and (10) are as follows:

$$\boldsymbol{\lambda} = [\lambda_{-n+k}, \ldots, \lambda_{-2}, \lambda_{-1}, \lambda_0, \lambda_1], \quad \|\Omega_{\boldsymbol{\lambda}}\| = 2^{n-k-1}. \tag{13}$$

Moreover, $\boldsymbol{\lambda}$ and size of the sample space of BO in (11) and (12) are as follows:

$$\boldsymbol{\lambda} = [\lambda_{-n+k}, \ldots, \lambda_{-2}, \lambda_j, \lambda_{-1}, \lambda_0, \lambda_1], \quad \|\Omega_{\boldsymbol{\lambda}}\| = 2^{n-k}. \tag{14}$$

For illustration, in an example eye diagram, the waveforms that correspond to the bit patterns ending in $\{\lambda_{-1}=1, \quad \lambda_0=0, \quad \lambda_1=1\}$ are shown in blue color in Fig. 4. Worst-eye determines V_{HL} by finding the highest voltage of such waveforms at the sampling time point, shown by a red line in this figure. In addition, Fig. 5 illustrates the objective function passed to BO, which corresponds to theses waveforms. In this example, it is assumed $n = 30$ and $k = 8$. Furthermore, maximum of this function is marked with a red circle in Fig. 5.

Next, EH and EW are calculated as

$$\text{EH} = V_{HL} - V_{LH} \tag{15}$$

$$\text{EW} = \text{UI} - (t_{RX} - t_{LX}). \tag{16}$$

It is worth noting that BO has been extensively studied and used for continuous variables, while its use for discrete variables has been limited. In worst-eye, although the variables are discrete, they take subsequent integer values after mapping with Gray codes; therefore, adjacent samples are equidistant. For integer numbers, we followed the common practice of rounding the next BO sample to the nearest integer number. This strategy is used in popular developed modules for BO [32], [33]. More advanced methods for BO with discrete variables exist in the literature [34]–[36], which were not used in this study since satisfactory results were achieved with the rounding approach.

Finally, the worst-case eye is estimated by overlaying the four waveforms that pass through the points determined by (9)–(12). If higher accuracy for the worst-case eye is desired, additional t_s sampling points are defined. Then, (9) and (10) are solved at the new sampling points, and the corresponding worst-case waveforms are added on top of the previous waveforms to provide a better estimate of the worst-case eye. To decide how many and where the extra sampling points should be selected, we suggest the following strategy.

1) Start with one sampling point at the center of the eye and carry out the proposed approach.
2) Superimpose all waveforms found by the proposed approach in addition to the worst-case waveforms.
3) See if the worst-case waveforms are suboptimal at other time points.
4) If there are time points with a considerable difference, find the lowest high and highest low at these points using the proposed approach. Then go back to step 2. Otherwise, end the algorithm.

There is no need to compare the results with Monte Carlo to find new sampling points in step 2. The comparison is done with already existing simulations in the proposed approach. Generally, the signal integrity is worse for these waveforms compared to randomly selected ones because in optimization we have searched toward the worst-cases.

D. Crosstalk

Crosstalk is one of the major issues in signal integrity, which needs to be carefully modeled when designing a high-speed channel. Therefore, in this section, we expand worst-eye to find the worst-case eye in the presence of the crosstalk.

To do so, one variable per aggressor line is added to the optimization problems in (9)–(12). These variables represent new index numbers for the possible patterns of symbols on each aggressor line. We show symbols of each line as $\boldsymbol{\xi}^j$, where j is an integer number, and $1 \le j \le q$, with q being the number of aggressor lines. The new indices are shown as $J_g(\boldsymbol{\xi}^j)$. Similar to the index $I_g(\boldsymbol{\lambda})$, $J_g(\boldsymbol{\xi}^j)$ is determined using the Gray code. However, we do not manually assign values of the first few symbols since their values are not intuitively clear, and it is better to leave their determination to the optimization algorithm. In addition, no postcursor is taken into account, and the number of the considered precursors is small because the effect of further symbols is negligible on the victim line. Therefore, the additional computational costs due to estimating the worst-case

crosstalk are not cumbersome. Considering crosstalk, the new optimization problems are as follows:

$$V_{LH} = \min_{I_g, J_g^1, J_g^2, \ldots, J_g^q} V_r\left(t_s, I_g|_{\lambda_{-1}=0, \lambda_0=1,\ \lambda_1=0}, J_g^1, J_g^2, \ldots, J_g^q\right) \tag{17}$$

$$V_{HL} = \max_{I_g, J_g^1, J_g^2, \ldots, J_g^q} V_r\left(t_s, I_g|_{\lambda_{-1}=1, \lambda_0=0,\ \lambda_1=1}, J_g^1, J_g^2, \ldots, J_g^q\right) \tag{18}$$

$$t_{LX} = \min \left\{ \begin{array}{l} \min\limits_{I_g, J_g^1, J_g^2, \ldots, J_g^q} t\left(V_0, I_g|_{\lambda_{-1}=0, \lambda_0=1,\ \lambda_1=1}, J_g^1, J_g^2, \ldots, J_g^q\right), \\ \min\limits_{I_g, J_g^1, J_g^2, \ldots, J_g^q} t\left(V_0, I_g|_{\lambda_{-1}=1, \lambda_0=0, \lambda_1=0}, J_g^1, J_g^2, \ldots, J_g^q\right) \end{array} \right\} \tag{19}$$

$$t_{RX} = \max \left\{ \begin{array}{l} \max\limits_{I, J_g^1, J_g^2, \ldots, J_g^q} t\left(V_0, I_g|_{\lambda_{-1}=0, \lambda_0=1,\ \lambda_1=0}, J_g^1, J_g^2, \ldots, J_g^q\right), \\ \max\limits_{I, J_g^1, J_g^2, \ldots, J_g^q} t\left(V_0, I_g|_{\lambda_{-1}=1, \lambda_0=0,\ \lambda_1=1}, J_g^1, J_g^2, \ldots, J_g^q\right) \end{array} \right\} \tag{20}$$

where for simplicity $I_g(\boldsymbol{\lambda})$ and $J_g(\boldsymbol{\xi}^j)$ are shown as I_g and J_g^j, respectively. Here, $\boldsymbol{\lambda}$ and size of its sample space is similar to (9)–(12). Moreover, $\boldsymbol{\xi}^j$ and size of its sample space is as follows:

$$\boldsymbol{\xi}^j = \left[\xi_{-h^j-1}^j, \xi_{-h^j-2}^j, \ldots, \xi_0^j\right], \quad \Omega_{\boldsymbol{\xi}^j} = 2^{h^j} \tag{21}$$

where h^j is the number of effective symbols considered for the jth aggressor line. Next, EH and EW are calculated using (15) and (16), respectively. Finally, the worst-case eye with crosstalk is estimated by overlaying the four waveforms that pass through the points determined by (17)–(20). To increase accuracy of the worst-case eye, additional waveforms are generated by changing t_s and repeating (17) and (18).

It is possible to extend the proposed approach to PAM-4 signaling. The main difference is that instead of the 4 critical values $\{V_{LH}, V_{HL}, t_{LX}, t_{RX}\}$, 12 critical values need to be calculated, which can be shown as $\{V_{LH\,i}, V_{HL\,i}, t_{LX\,i}, t_{RX\,i}$, for i $\in$ [1, 2, 3]$\}$. Similar to the mapping of NRZ, first the bit pattern samples of optimization need to be ordered in the Gray code representation. Then, every two subsequent bits of each sample are converted to PAM-4 signaling. Using Gray codes for PAM-4 is a standard practice [37]. For instance, for two bits 00, 01, 11, and 10 correspond to -1. -1/3, +1/3, and +1, respectively. The advantage of using this mapping for PAM-4 is that each two subsequent samples in the design space are only different in one-third of the amplitude. Therefore, the objective function becomes smoother than the binary mapping.

IV. Numerical Examples

In this section, the application of the proposed worst-eye approach is demonstrated on three examples to evaluate its performance and compare it with the conventional transient eye analysis. Worst-eye is coded in Python 3, and the Scikit-Optimize library [32] is used for BO, which is based on [30] and [38]. In this library, BO is coded in the minimization format; hence, maximizing is done by finding the minimum of negative objective functions. Moreover, for the covariance kernel in BO, we used the Matern function with a smoothness factor of 2.5, which is the recommended value in Scikit-Optimize. We suggest using this kernel for the general use of worst-eye since it has shown a satisfactory performance for BO in the literature [35], [39]. Nevertheless, we acknowledge that using lower smoothness factors (0.5 and 1.5) can be helpful since they result in less smooth approximated functions (see the documentation in [30]), and the worst-eye functions can show such behavior.

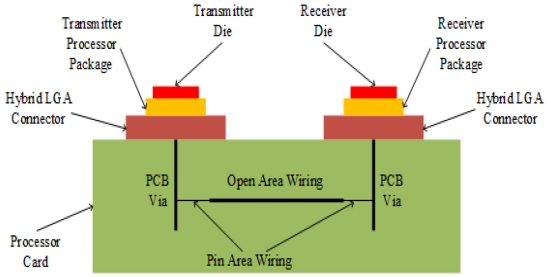


Fig. 6. High-speed SerDes channel in Example 1, comprised of two packages, vias, and differential wiring.

A. Example 1

The purpose of this example is to evaluate worst-eye when ISI is dominant. Therefore, the SerDes channel in Fig. 6 is utilized, which does not have any crosstalk. This channel includes two processor packages connected to a board with a hybrid land grid array connectors. The passive channel loss is ~11 dB at 8 GHz. An 85-Ω differential stripline wiring is used in the transmitter and receiver processor packages with 31 mm and 34 mm lengths, respectively. These striplines are embedded in the GZ41 material which has Dk $\sim$ 3.31 and Df $\sim$ 0.0092 at 1 GHz. In addition, the board includes two differential PCB vias with an active via length of 150 mil, and stub length of 20 mil. It also contains 1 in of necked down pin area wiring in the shadow of each processor, and 2 in of 85 Ω differential wiring in the open area, which amounts to 4 in of wiring on the board. Moreover, the dielectric of the board is a low loss material with Dk $\sim$ 3.95 and Df $\sim$ 0.0084 at 1 GHz. This channel is simulated in a custom-build solver named HSSCDR [40], [41], which is developed by IBM and used for fast simulation of SerDes channels. For comparison, a conventional transient eye analysis with 10 million random bits is performed. This long simulation is done in portions of one million bits each for better handling of the data. The transient eye includes no more than 10 million bits because variations in the results are negligible after this point, and we were limited by the computational costs.

In this example, the channel operates at 16 Gb/s; hence, the UI is equal to 62.5 ps. The pulse response of the channel is shown in Fig. 7. From this figure, it is observed that the channel response is about 32 UIs. Therefore, the total number of symbols with effective ISI is set to 32 (i.e., $n + 2 = 32$). Furthermore, k is set to 8; thus, each sample includes 256 b patterns, where the length of each pattern is 32. This results in a simulating

TABLE II
TRANSIENT EYE AND WORST-EYE ANALYSIS RESULTS IN EXAMPLE 1

	Lowest high (mV)	Highest low (mV)	Leftmost crossing (ps)	Rightmost crossing (ps)	Eye height (mV)	Eye width (ps)	Number of bits
Worst-eye, binary mapping	25.6	-25.6	12.0	47.1	51.2	27.4	1,720,320
Worst-eye, Gray code mapping	23.2	-25.7	11.5	47.1	48.9	26.9	2,547,712
Transient Eye	24.4	-25.3	11.8	47.3	49.7	27	10,000,000

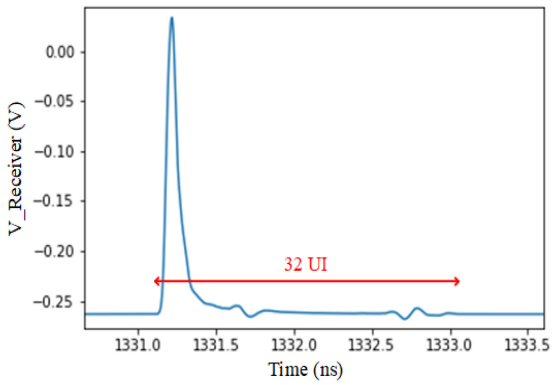


Fig. 7. Pulse response of the channel in Fig. 6 when pulsewidth = 62.5 ps.

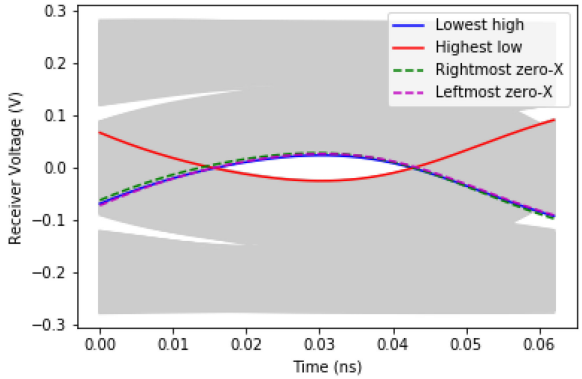


Fig. 8. Transient eye and the worst-eye waveforms in Example 1.

transmission of 8192 b for each sample. It is worth noting that transient simulation of this many bits can be done in negligible time using HSSCDR. Next, worst-eye is used to find the values of $V_{LH}, V_{HL}, t_{LX}, t_{RX}$, and subsequently EH and EW. For V_{LH} or V_{HL}, BO determines value of 21 b, while for t_{LX} or t_{RX} BO determines value of 22 b. However, for each of them, only a 1-D optimization is performed since the bit patterns are mapped to a single dimension using the Gray code. The algorithm is stopped after 100 iterations for each objective function. For comparison, in this example, we have also used the simple binary mapping instead of the Gray code mapping and kept everything else the same. The results are shown in Table II, where they are compared with the transient eye analysis as well. The comparison shows that worst-eye with Gray code mapping provides slightly smaller EH and EW values than the transient eye. Nevertheless, since V_{LH}, V_{HL}, t_{LX}, and t_{RX} correspond to real simulation results, and they are not estimated, this means in this example, worst-eye with Gray code mapping is more accurate than the transient eye with 10 million bits. In other words, a longer transient eye analysis can eventually capture the waveforms found by worst-eye with Gray code mapping and provide the same or better accuracy; however, it would be even more time consuming than simulating transmission of 10 million bits. Furthermore, it is seen from the results in Table II that using the binary mapping instead of the Gray code mapping in worst-eye can make the proposed approach converge to suboptimal values. Moreover, the number of simulated bits in HSSCDR for each approach is shown in the last column of Table II. The number of bits for either of the worst-eyes is less than the number of bits per sample multiplied by the total number of iterations since this algorithm saves simulation results and reuses them if BO selects a point that has been sampled before for any of the objective functions. This strategy leads to further reduction of the computational costs.

Although an equal number of BO iterations were used with both mappings, it is observed that the total number of simulated bits with binary mapping is about 800 000 less than the Gray code method. This means the binary method has repeated the sampling at more points and has used the saved data. Adding this observation to the fact that binary mapping leads to less accurate results, makes us believe it is more prone to getting stuck in local optima. In the remainder of the article, only the Gray code mapping is considered, and worst-eye refers to the proposed approach with this type of mapping.

In this example, evaluation of additional LH and HL at new sampling points is not necessary, because overlaying only the waveforms from the original four worst-case points provides a good estimate of the worst-case eye. The transient eye and the waveforms that pass through the four worst-case points of worst-eye are illustrated in Fig. 8. For aesthetic reasons, we have shifted the eye diagram and waveforms found by worst-eye. Hence, there is some difference between the level-crossing time points in Table II and Fig. 8; nevertheless, both show the same EH and EW. Next, the convergence curves of BO for the four objective functions are presented in Fig. 9. The curves show that the minimum value decreases drastically which is due to the fast convergence rate of BO. Nevertheless, Fig. 9(b) suggests that achieving even lower $-V_{HL}$ values might be possible by increasing number of the iterations.

Finally, to compare the computational costs, the number of simulated bits in the last two rows of Table II is considered. The HSSCDR simulation for the transient eye and the worst-eye roughly takes 34 and 9 min, respectively. Moreover, the overhead optimization cost of the proposed worst-eye approach is about

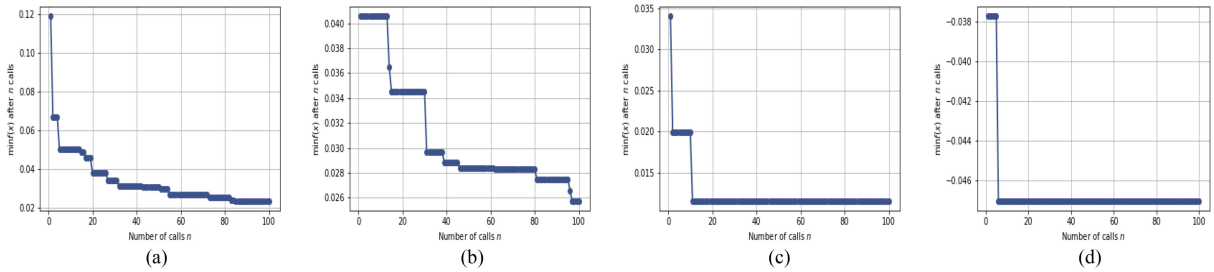


Fig. 9. Convergence plots of the four worst-case variables in Example 1. (a) V_{LH}. (b) $-V_{HL}$. (c) t_{LX}. (d) $-t_{RX}$.

TABLE III
TRANSIENT EYE AND WORST-EYE ANALYSIS RESULTS IN EXAMPLE 2

	Lowest high (mV)	Highest low (mV)	Leftmost crossing (ps)	Rightmost crossing (ps)	Eye height (mV)	Eye width (ps)	Number of bits
Worst-eye	20.6	-20.7	11.5	47.1	41.3	26.9	2,310,144
Transient Eye	20.1	-20.5	11.8	47.3	40.6	27	10,000,000

6 min, leading to an overall speedup of greater than 2 times. It is worth noting, this speedup can significantly increase based on the example and the worst-eye's settings. We recommend using the worst-eye approach when the optimization cost is negligible compared to the channel's simulation time. HSSCDR is a relatively fast solver since it is custom-build for this type of channels, and it takes advantage of several preexisting models.

B. Example 2

In this example, the performance of the proposed approach is studied when it is applied to a nonlinear system. For this purpose, once more the SerDes channel in Example 1 is considered; however, the compression point of the receiver is reduced far enough to affect the EH. This change results in a nonlinear decrease in the received voltages. Other configurations of the channel and the input pulse are the same as Example 1.

For the proposed worst-eye algorithm, n is set to 32, k is set to 8, and the BO algorithm is stopped after 100 iterations for each objective function, similar to Example 1. In addition, hyperparameters of BO are the same as Example 1. In this example, we show that accurate results can be achieved without a major effort to tune the parameters after modifying the channel in the design process. For comparison, a transient eye analysis with 10 million random bits is performed, which was done in portions of one million bits each for better handling of the data. The results are presented in Table III, where it is observed that the results of the proposed worst-eye approach closely match results of the transient eye analysis.

Similar to Example 1, additional sampling points are not necessary since the accuracy of the estimated worst-case eye is adequate. The shifted transient eye and the waveforms that pass through the V_{LH}, V_{HL}, t_{LX}, and t_{RX} points are illustrated in Fig. 10, showing a good match between the predicted worst-case eye and the eye opening of the transient eye. Furthermore, the nonlinear compression in this example can be observed by comparing Figs. 8 and 10, which show the eye diagram before and after reducing the compression point.

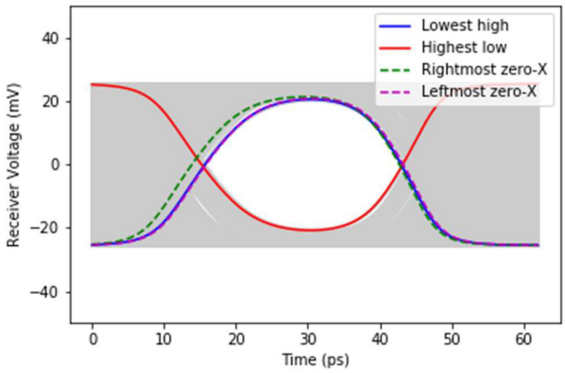


Fig. 10. Transient eye and the worst-eye waveforms in Example 2.

Moreover, the convergence curves of BO for the four objective functions is presented in Fig. 11, showing a fast convergence rate. By comparing Figs. 9 and 11, we draw the following conclusions. Fig. 11(c) and (d) is almost the same as Fig. 9(c) and (d), respectively, and they converge to the same final values. This results in equal EWs for example 1 and 2. This observation means that the low compression point only affects the EH and not the EW, which is expected since the low compression does not affect the signals near center of the eye; hence, their zero crossing points stays the same. Furthermore, it is observed that Fig. 11(c) and (d) shows faster convergence compared to Fig. 9(c) and (d). The reason might be that the lowest high and the highest low values are compressed in Example 2 since they are close to the receiver's compression point. Therefore, the voltage variations reduce, and the corresponding functions can have close or equal values for different inputs.

The number of bits simulated in HSSCDR for each approach is shown in the last column of Table III. Although the number of the BO iterations and bits per sample is the same as the previous example, the total number of bits for worst-eye is less because more samples have been repeated in this example. Furthermore, the HSSCDR simulation for the transient eye and the worst-eye roughly takes 34 and 8 min, respectively. Additionally, the

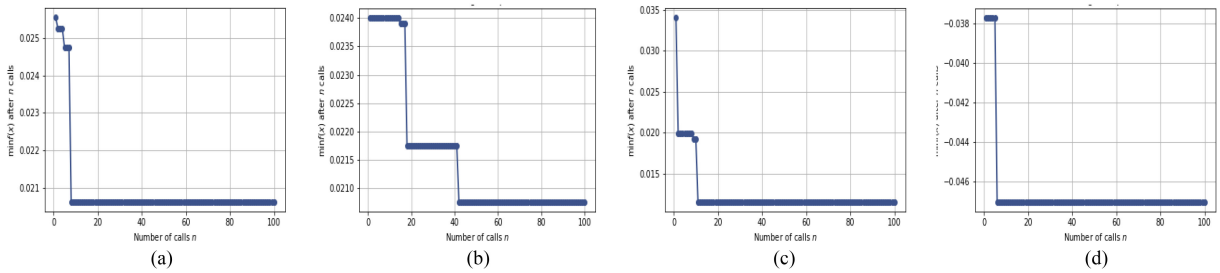


Fig. 11. Convergence plots of the four worst-case variables in Example 2. (a) V_{LH}. (b) $-V_{HL}$. (c) t_{LX}. (d) $-t_{RX}$.

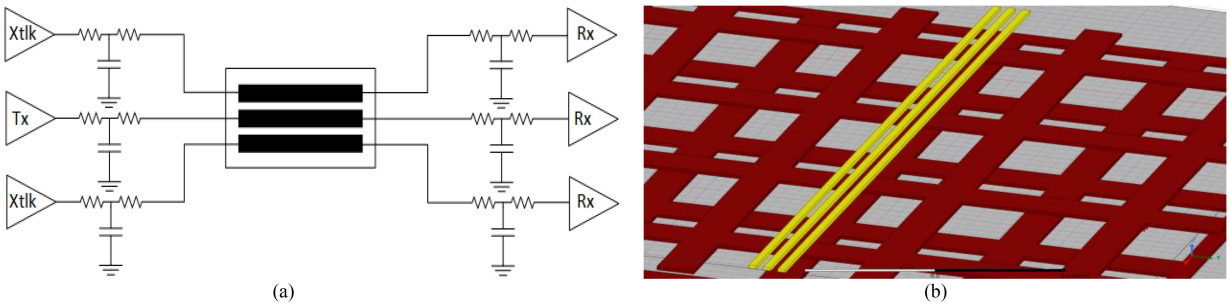


Fig. 12. The high-speed channel of Example 3. (a) Schematics. (b) Physical design of the embedded microstrip lines.

TABLE IV
TRANSIENT EYE AND WORST-EYE ANALYSIS RESULTS IN EXAMPLE 3

	Lowest high (mV)	Highest low (mV)	Leftmost crossing (ps)	Rightmost crossing (ps)	Eye height (mV)	Eye width (ps)	Number of bits
Worst-eye (1st round)	775	136	19	274	639	745	5,760
Transient Eye	775	136	19	275	639	744	1,000,000

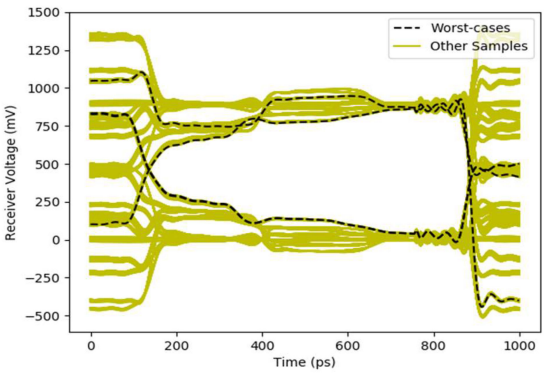


Fig. 13. Comparison of the worst-case waveforms and other waveforms.

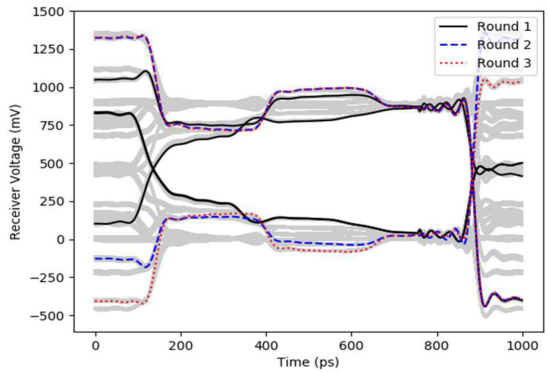
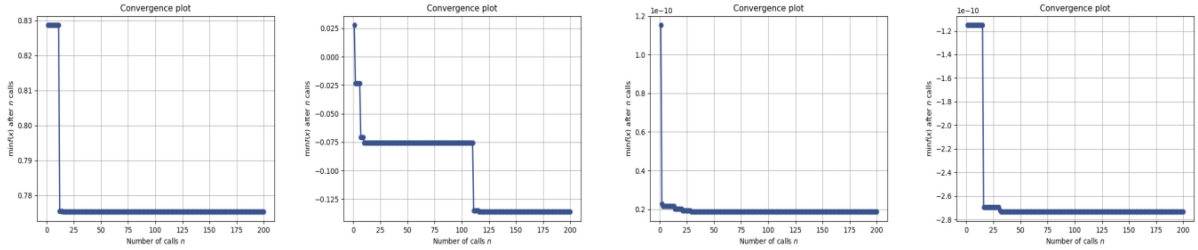


Fig. 14. Transient eye and 3 rounds of worst-case waveforms in found in the first round of the proposed approach in Example 3.

overhead optimization cost of the proposed approach is about 6 min, leading to an overall speedup of greater than two times. As mentioned before, the speedup can significantly increase based on the example. For instance, a much higher speedup is achieved in the next example.

C. Example 3

The purpose of this example is a demonstration of a higher speedup and evaluation of the proposed approach in the presence of crosstalk. Therefore, the high-speed channel illustrated in Fig. 12(a) is considered. In this channel, the middle line is the victim, and the output is observed before the receiver of this line. The value of each capacitor and resistor is 936 mF and about 22 Ω, respectively. The channel is from a system-on-package design, with single-ended signaling. It is formed of three coupled embedded microstrip lines over two meshed PDN layers, as illustrated in Fig. 12(b). In addition, width, height,



(a) (b) (c) (d)

Fig. 15. Convergence plots of finding the four worst-case variables in Example 3. (a) V_{LH}. (b) $-V_{HL}$. (c) t_{LX}. (d) $-t_{RX}$.

distance, and total length of the lines are 12, 9, 12, and 19 200 μm, respectively. The structure is embedded in liquid crystal polymer, with a thickness of 25 μm. Moreover, the network is simulated in HSPICE O-2018. It is worth noting, that the embedded microstrip lines, transmitters, and receivers in this example have realistic and complex models, which results in increased circuit simulation times. Furthermore, the data rate is 1 Gb/s, and the low and high logics are 0 and 0.9 V, respectively. For comparison, a conventional transient eye analysis with 1 million bits is performed. HSPICE does not perform well with simulation of millions of bits. To get around this issue, we limited each HSPICE simulation to only 1000 b after reaching the steady state. A thousand of such simulations were performed and put together to obtain the results of the transient eye simulation. The transient eye includes no more than 1 million bits because variations in the results are negligible after this point, and we were limited by the computational costs.

A total of 10 precursors on the victim line is considered; thus, $n + 2 = 12$. k is set to 0 since the number of the precursors is already small and manageable. In addition, q is equal to 2 because two aggressor lines are present. It is assumed that the state of the last four symbols on each aggressor line has nontrivial crosstalk effects; hence, $h = 4$. Next, worst-eye is used to find the values of $V_{LH}, V_{HL}, t_{LX}, t_{RX}$, and subsequently EH and EW. Note that, the optimization algorithm solves a 3-D problem per objective function since the patterns are mapped using the Gray code scheme. For V_{LH} and V_{HL}, the optimization algorithm determines values of 9 b on the victim line and 4 b on each aggressor line, while for t_{LX} and t_{RX} the optimization algorithm determines values of 10 b on the victim line and 4 b on each aggressor line. The algorithm is stopped after 200 iterations for each objective function. The number of iterations is the only hyperparameter of BO which is different from the previous examples. However, this example is significantly different from the previous examples, and some effort to tune the parameters is expected. The results and the total number of simulated bits are shown in Table IV, where it is compared with the transient eye analysis. The results show that Worst-eye closely matches the EH and EW results of the transient eye.

In this example, at some time points, it is observed that the worst-case waveforms are suboptimal when all the waveforms found by the proposed approach are superimposed. This comparison is presented in Fig. 13, where the worst mismatch is seen at $t = 371$ ps and $t = 843$ ps. The mismatches appeared when we introduced the crosstalk to this example. Therefore, we believe that the mismatch is caused by the crosstalk. In addition, the reason for having two mismatches can be that crosstalk is strongest near the rising and falling edges or the switchings. The mismatches happen near $t = 371$ ps and 843 ps; thus, it should be where the switching noise reaches the victim line. Next, we find the lowest high, the highest low, and the corresponding waveforms at these two points. Superimposing, all the worst-case waveforms on the transient eye of this example is shown in Fig. 14. It is observed that superimposing the worst-case waveforms results in a good approximation of the inner opening of the transient eye. It is worth noting that in rounds 2 and 3 only 2412 and 1476 additional bits have been simulated, respectively, because the proposed approach takes advantage of the previously saved data.

Furthermore, the convergence curves of BO for the four objective functions are presented in Fig. 15, showing a fast convergence rate.

As presented in Table IV, the number of the simulated bits in the first round of the proposed algorithm is 5760, which is orders of magnitude smaller than the one million bits in the transient eye analysis. The HSPICE simulation time of the transient eye and the first round of worst-eye are roughly 1356 min and 8 min, respectively. In addition, the overhead optimization cost of the first round of worst-eye is about 21 min. Hence, the proposed approach provides a speedup of roughly 47 times for calculating the EH and EW. Furthermore, in the second round, transient simulation and overhead of worst-eye are roughly 3 min and 13 min, respectively. In the third round, transient simulation and overhead of worst-eye are roughly 2 min and 13 min, respectively. The overhead is higher in the first round since t_{LX} and t_{RX} are only calculated in this round. In total, the three rounds of worst-eye take about 60 min; therefore, it provides a speedup of roughly 23 times for finding the worst-case eye opening.

V. Conclusion

In this article, an optimization-based algorithm for quick evaluation of the eye diagram is suggested. Traditionally, eye diagram analysis is performed using a lengthy transient simulation, which can be prohibitive for modern complex channels with low BER. To alleviate the computational costs, many estimation methods have been suggested in the literature; however, they can be limited in their accuracy, efficiency, or applications. Hence, developing a new method is necessary. The proposed approach, dubbed worst-eye, focuses on DDJ, DDN, ISI, and crosstalk,

for an NRZ pulse sequence. This approach finds data patterns that result in V_{LH}, V_{HL}, t_{LX}, and t_{RX}, which are the boundary points on the worst-case eye opening. Using these points, EH and EW are calculated, and the waveforms passing through the worst-case points are overlaid to estimate the worst-case eye opening. Worst-eye takes advantage of a mapping scheme based on the Gray code to reduce complexity. In addition, after necessary considerations based on the domain knowledge, BO is used to find the worst-case points and waveforms. Finally, worst-eye is evaluated by its application on a high-speed SerDes channel on PCB with high- and low-receiver compression point, and a channel in a system-on-package design. Numerical results show that the proposed approach can accurately find the EW and EH with up to 47 times speedup, and the worst-case eye opening with up to 23 times speedup, when compared with the transient eye.

REFERENCES

[1] M. P. Li, *Jitter Noise and Signal Integrity at High-Speed*. Upper Saddle River, NJ, USA: Prentice Hall, 2007.

[2] J. Cho, E. Song, J. Shim, J. Kim, and J. Kim, "A precise analytical eye-diagram estimation method for non-ideal high-speed channels," in *Proc. 18th IEEE Conf. Electr. Performan. Electron. Packag. Syst.*, 2009, pp. 159–162.

[3] W. Yao, Y. Shi, L. He, S. Pamarti, and Y. Hu, "Worst case timing jitter and amplitude noise in differential signaling," in *Proc. IEEE 10th Int. Symp. Qual. Electron. Des.*, 2009, pp. 40–46.

[4] J. F. Buckwalter, "Predicting microwave digital signal integrity," *IEEE Trans. Adv. Packag.*, vol. 32, no. 2, pp. 280–289, May 2009.

[5] B. K. Casper, M. Haycock, and R. Mooney, "An accurate and efficient analysis method for multi-Gb/s chip-to-chip signaling schemes," in *Proc. IEEE Symp. Very Large Scale Integr. Circuits*, 2002, pp. 54–57.

[6] A. Sanders, M. Resoo, and D. Ambrosia, "Channel compliance testing using novel statistical eye methodology," in Design Con., Santa Clara, CA, USA, 2004.

[7] M. Tsuk, D. Dvorscak, C. S. Ong, and J. White, "An electrical-level superposed-edge approach to statistical serial link simulation," in *Proc. IEEE/ACM Int. Conf. Comput.-Aided Des. Dig. Tech. Papers*, Nov. 2009, pp. 717–724.

[8] J. Ren and K. S. Oh, "Multiple edge responses for fast and accurate system simulations," *IEEE Trans. Adv. Packag.*, vol. 31, no. 4, pp. 741–748, Nov. 2008.

[9] B. Mutnury, M. Swaminathan, and J. P. Libous, "Macromodeling of nonlinear digital I/O drivers," *IEEE Trans. Adv. Packag.*, vol. 29, no. 1, pp. 102–113, Feb. 2006.

[10] H. Yu, H. Chalamalasetty, and M. Swaminathan, "Behavioral modeling of steady-state oscillators with buffers using neural networks," in *Proc. 27th IEEE Conf. Elect. Perform. Electron. Packag. Syst.*, Oct. 2018, pp. 307–309.

[11] T. Nguyen, T. Lu, J. Sun, Q. Le, K. Wu and J. Schut-Aine, "Transient simulation for high-speed channels with recurrent neural network," in *Proc. 27th IEEE Conf. Elect. Perform. Electron. Packag. Syst.*, 2018, pp. 303–305.

[12] M. Ahadi Dolatsara, J. Hejase, W. D. Becker, and M. Swaminathan, "A hybrid methodology for jitter and eye estimation in high-speed serial channels using polynomial chaos surrogate models," *IEEE Access*, vol. 7, pp. 53629–53640, 2019.

[13] D. Xiu and G. E. Karniadakis, "The Wiener-Askey polynomial chaos for stochastic differential equations," *SIAM J. Sci. Comput.*, vol. 24, no. 2, pp. 619–644, 2002.

[14] Z. Chen and G. Katopis, "Searching for the worst-case eye diagram of a signal channel in electronic packaging system including the effects of the nonlinear I/O devices and the crosstalk from adjacent channels," in *Proc. IEEE 59th Electron. Compon. Technol. Conf.*, 2009, pp. 1106–1113.

[15] Z. Chen, W. D. Becker, and G. Katopis, "A new approach to deriving packaging system statistical eye diagram based on parallel non-linear transient simulations using multiple short signal bit patterns," in *Proc. IEEE 62nd Electron. Components Technol. Conf.*, 2012.

[16] S. M. Ulrich, A. M. Wirick, D. de Araujo, N. Pham, and M. Cases, "The Nittany Genome Project: A genetic algorithm approach to optimize a worst case bitstream for package simulation," in *Proc. IEEE Elect. Perform. Elect. Packag.*, 2003, pp. 303–306.

[17] N. Singh, B. Mutnury, N. Pham, M. Cases, and C. Wesley, "Bit-pattern optimization for accurate analysis of complex high-speed interfaces," in *Proc. IEEE 58th Electron. Compon. Technol. Conf.*, 2008, pp. 669–675.

[18] S. N. Ahmadyan, C. Gu, S. Natarajan, E. Chiprout, and S. Vasudevan, "Fast eye diagram analysis for high-speed CMOS circuits," in *Proc. EDA Consortium Des. Automat. Test Eur. Conf. Exhib.*, 2015, pp. 1377–1382.

[19] H. S. Wilf, *Combinatorial Algorithms: An Update*, vol. 55, Philadelphia, PA, USA: SIAM, 1989.

[20] E. Brochu, V. M. Cora, and N. De Freitas, "A tutorial on Bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning," 2010, *arXiv:1012.2599*.

[21] S. J. Park, B. Bae, J. Kim, and M. Swaminathan, "Application of machine learning for optimization of 3-D integrated circuits and systems," *IEEE Trans. Very Large Scale Integr. Syst.*, vol. 25, no. 6, pp. 1856–1865, Jun. 2017.

[22] H. M. Torun, M. Swaminathan, A. K. Davis, and M. L. F. Bellaredj, "A global Bayesian optimization algorithm and its application to integrated system design," *IEEE Trans. Very Large Scale Integr. Syst.*, vol. 26, no. 4, pp. 792–802, Apr. 2018.

[23] H. M. Torun and M. Swaminathan, "High-dimensional global optimization method for high-frequency electronic design," *IEEE Trans. Microw. Theory Techn.*, vol. 67, no. 6, pp. 2128–2141, Jun. 2019.

[24] H. Torun, J. A. Hejase, J. Tang, W. D. Becker, and M. Swaminathan, "Bayesian active learning for uncertainty quantification of high speed channel signaling," in *Proc. IEEE 27th Conf. Elect. Perform. Electron. Packag. Syst.*, 2018, pp. 311–313.

[25] M. Ahadi Dolatsara and M. Swaminathan, "Determining worst-case eye height in low BER channels using Bayesian optimization," in *Proc. 11th IEEE Latin Amer. Symp. Circuits Syst.*, 2020, pp. 1–4.

[26] F. Gray, "Pulse code communication," U. S. Patent 2 632 058, 17 Mar. 1953.

[27] H. G. Dietz, "The aggregate magic algorithms," Technical report, University of Kentucky, 2011. [Online]. Available: http://aggregate.org/MAGIC/. Accessed Mar. 2020

[28] P. I. Frazier, "A tutorial on Bayesian optimization," 2018, *arXiv:1807.02811*.

[29] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning*. Cambridge, MA, USA: MIT Press, pp. 84–85, 2006.

[30] "Scikit-learn," Source code. [Online]. Available: https://scikit-learn.org/stable/. Accessed Jun. 2020

[31] M. D. Hoffman, E. Brochu, and N. de Freitas, "Portfolio allocation for Bayesian optimization," in *Proc. 27th Conf. Uncertainty Artif. Intell.*, 2011, pp. 327–336.

[32] "Scikit-optimize," Source code. [Online]. Available: https://scikit-optimize.github.io/stable/. Accessed Oct. 2019

[33] "Spearmint," Source code. [Online]. Available: https://github.com/HIPS/Spearmint. Accessed Jun. 2020

[34] E. C. Garrido-Merchán and D. Hernández-Lobato, "Dealing with integer-valued variables in Bayesian optimization with Gaussian processes," 2017, *arXiv:1706.03673*.

[35] E. C. Garrido-Merchán and D. Hernández-Lobato, "Dealing with categorical and integer-valued variables in Bayesian optimization with Gaussian processes," *Neurocomputing*, vol. 380, pp. 20–35, 2020.

[36] R. Baptista and M. Poloczek, "Bayesian optimization of combinatorial structures," 2018, *arXiv:1806.08838*.

[37] H. Zhang, B. Jiao, Y. Liao, and G. Zhang, "PAM4 signaling for 56G serial link applications—A tutorial," in Design Con, 2016.

[38] F. Pedregosa *et al.*, "Scikit-learn: Machine learning in Python," *J. Mach. Learn. Res.*, vol. 12, pp. 2825–2830, 2011.

[39] J. Snoek, H. Larochelle, and R. P. Adams, "Practical Bayesian optimization of machine learning algorithms," in *Proc. 25th Int. Conf. Neural Inf. Process. Syst.*, 2012, pp. 2951–2959.

[40] S. Chun *et al.*, "Package and printed circuit board design of a 19.2 Gb/s data link for high-performance computing," in *Proc. IEEE 67th Electron. Compon. Technol. Conf.*, 2017, pp. 1701–1707.

[41] T. Beukema, "Topics in design and analysis of high data rate SERDES systems," in *IEEE SSCS Denver Section Semin*. IBM T. J. Watson Res. Center, Fort Collins, CO, USA, Sep. 2009.

Majid Ahadi Dolatsara (Student Member, IEEE) received the B.Sc. degree from the K.N.Toosi University of Technology, Tehran, Iran, in 2013, and the M.Sc. degree from Colorado State University, Fort Collins, CO, USA, in 2016, both in electrical engineering. He is currently working toward the Ph.D. degree in electrical engineering with Georgia Institute of Technology, Atlanta, GA, USA.

His current research interests include electronic design automation and machine learning for signal and power integrity and packaging.

Mr. Ahadi Dolatsara was the recipient of the Best Poster Paper Award at the 23rd IEEE International Conference on Electrical Performance of Electronic Packaging and Systems in 2014.

Jose Ale Hejase (Member, IEEE) received the B.S. degree in electrical engineering (*cum laude*) from Oakland University, Rochester Hills, MI, USA, in 2006, and the M.S. and Ph.D. degrees in electrical and computer engineering—specializing in applied electromagnetics—from Michigan State University, East Lansing, MI, USA, in 2009 and 2012, respectively.

Since 2012, he has been with IBM Corporation. He is currently a Senior Engineer with IBM working with the POWER servers hardware development organization. His responsibilities revolve around signal integrity design for the latest and highest performance high-speed computer sever bus links. His current research interests include electromagnetic material characterization, electromagnetic nondestructive evaluation, millimeter-wave guiding and high-speed link channel design/signal integrity.

Dr. Hejase is a member of the technical program committees of the IEEE EPEPS and IEEE SPI conferences and a member of the IEEE EPS technical committee on Electrical Modeling, Design and Simulation. He was the recipient of the IBM Early Tenure Inventor Award in 2013, IBM Outstanding Technical Achievement Awards in 2017 and 2019 for his work on developing POWER9 servers high-speed buses and his fourth IBM Invention Plateau in 2020.

Wiren Dale Becker (Fellow, IEEE) received the B.E.E. degree from the University of Minnesota, the M.S.E.E. degree from Syracuse University, Syracuse, NY, USA, and the Ph.D. degree in electrical engineering from University of Illinois at Urbana-Champaign, Champaign, IL, USA.

He is a Chief Engineer in electronic packaging integration with IBM Systems. His responsibility is the electrical system packaging architecture of IBM Systems including the design of high-speed channels to enable the computer system performance and the power distribution networks for reliable operation of the integrated circuits that make up the processor subsystem.

Dr. Becker has chaired the IEEE EPEPS Conference and the SIPI Embedded Conference of the EMC Symposium. He currently chairs the IEEE EPS Technical Committee on electrical modeling, design, and simulation and is a Senior Area Editor for the IEEE Transactions on Components, Packaging and Manufacturing Technology. He is a Cochair of the High-Performance Computing Working Group of the Heterogeneous Integration Roadmap. He has chaired the iNEMI PEG on high-end systems including the chapter on the high-end systems roadmap. He is an iNEMI Technical Committee member and a member of SWE.

Jinwoo Kim (Student Member, IEEE) received the B.S. degree in electrical and computer engineering and the M.S. degree in electrical engineering and computer science from Seoul National University, Seoul, South Korea, in 2011 and 2013, respectively. He is currently working toward the Ph.D. degree in electrical and computer engineering with the School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA, USA.

He was an Analog Circuit Designer with Samsung Electronics, Hwaseong, South Korea, from 2013 to 2017. His current research interests include interposer-based 2.5-D IC design and coanalysis, and 3-D IC design and methodology.

Sung Kyu Lim (Senior Member, IEEE) received the B.S., M.S., and Ph.D. degrees in computer science from the Computer Science Department, University of California, Los Angeles (UCLA), Los Angeles, CA, USA, in 1994, 1997, and 2000, respectively.

He is currently a Professor with the School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA, USA. He joined the School in 2001. His research is featured as Research Highlight in the Communication of the ACM in January, 2014. He is the author of Practical Problems in VLSI Physical Design Automation (Springer, 2008) and Design for High Performance, Low Power, and Reliable 3D Integrated Circuits (Springer, 2013). He has authored or coauthored more than 350 papers on 2.5-D and 3-D ICs. His current research focus is on the architecture, design, test, and EDA solutions for 2.5-D and 3-D ICs.

Prof. Lim was the recipient of the National Science Foundation Faculty Early Career Development (CAREER) Award in 2006 and the ACM SIGDA Distinguished Service Award in 2008. He was also the recipient of the Best Paper Award from ATS 2012, ITC 2014, and EDAP 2017. His works have been nominated for the Best Paper Award at several top venues in EDA and circuit/packaging design. He was an Associate Editor of the IEEE TRANSACTIONS ON VERY LARGE SCALE INTEGRATION SYSTEMS during 2007–2009 and the IEEE TRANSACTIONS ON COMPUTER-AIDED DESIGN OF INTEGRATED CIRCUITS AND SYSTEMS during 2013–2018.

Madhavan Swaminathan (Fellow, IEEE) received the M.S./Ph.D. degrees in electrical engineering from Syracuse University, Syracuse, NY, USA, in 1989 and 1991, respectively.

He is currently the John Pippin Chair in microsystems packaging and electromagnetics with the School of Electrical and Computer Engineering (ECE), Professor ECE with a joint appointment with the School of Materials Science and Engineering (MSE), and Director of the 3-D Systems Packaging Research Center (PRC), GT. He also serves as the Site Director for the NSF Center for Advanced Electronics through Machine Learning (CAEML) and Theme Leader for Heterogeneous Integration, SRC JUMP ASCENT Center. He formerly held the position of Founding Director, Center for Co-Design of Chip, Package, System (C3PS), Joseph M. Pettit, Professor in Electronics in ECE and Deputy Director of the Packaging Research Center (NSF ERC), GT. Prior to joining GT, he was with IBM working on packaging for supercomputers. He has authored 500+ refereed technical publications, holds 30 patents, primary author and coeditor of three books, Founder and Co-Founder of two start-up companies, and founder of the IEEE Conference Electrical Design of Advanced Packaging and Systems (EDAPS), a premier conference sponsored by the EPS society.

Prof. Swaminathan's research has been recognized with 22 best paper and best student paper awards. In addition, his most recent awards include the D. Scott Wills ECE Distinguished Mentor Award (2018), Georgia Tech Outstanding Achievement in Research Program Development Award (2017), Distinguished Alumnus Award from the National Institute of Technology Tiruchirappalli (NITT) in India (2014), and Outstanding Sustained Technical Contribution Award from the IEEE Components, Packaging, and Manufacturing Technology Society (2014). He has served as the Distinguished Lecturer for the IEEE EMC society.