# DesignCon 2019

# Eye Diagram and Jitter Estimation in SerDes Designs using Surrogate Models Generated with the Polynomial Chaos Theory

Majid Ahadi Dolatsara, School of Electrical and Computer Engineering, Center for Co-Design of Chip, Package, System (C3PS), Georgia Institute of Technology
mad6@gatech.edu

Jose Hejase, IBM
jahejase@us.ibm.com

Wiren Becker, IBM
wbecker@us.ibm.com

Madhavan Swaminathan, School of Electrical and Computer Engineering, Center for Co-Design of Chip, Package, System (C3PS), Georgia Institute of Technology
madhavan@ece.gatech.edu

## Abstract

In modern SerDes channels, calculation of data dependent jitter and ISI is challenging since the traditional transient eye analysis has a slow convergence, and statistical methods have limited use. This paper suggests a methodology to train an end-to-end surrogate model using an expansion of orthogonal polynomials which is based on the generalized Polynomial Chaos theory. This model can be used for estimation of the receiver voltage and statistical analysis for jitter and eye calculations. A significant speedup compared to the transient eye analysis is observed since the length of simulation to train the model is much smaller, and the overhead is negligible.

## Authors Biography

**Majid Ahadi Dolatsara** received his B.Sc. degree in electrical engineering from K.N.Toosi University of Technology, Tehran, Iran, in 2013, and M.Sc. degree in electrical engineering from Colorado State University, Fort Collins, Colorado in 2016. Currently he is doing his PhD degree in electrical engineering at Georgia Institute of Technology, Atlanta, Georgia. His research interest includes EDA tools and Machine Learning.

**Jose Hejase** Jose Ale Hejase received a B.S. degree in electrical engineering (cum laude) from Oakland University, MI, in 2006 and an M.S. and Ph.D. degrees in electrical and computer engineering—specializing in applied electromagnetics—from Michigan State University in 2009 and 2012, respectively. He subsequently joined IBM in 2012, where he is currently a senior engineer and has worked in the high-speed bus signal integrity team on developing and pioneering the latest high speed signal computer sever bus designs. He received the IBM Early Tenure Inventor Award in 2013, an IBM Outstanding Technical Achievement Award in 2017 for his work on developing POWER9 servers high speed buses, and his third IBM Invention Plateau in 2018. He is author or coauthor of 8 patents and more than 50 peer reviewed technical journals and conference papers.

**Wiren Dale Becker** received the B.E.E degree from the U. of MN, M.S.E.E. from Syracuse U. and the Ph.D. from UIUC. He is a Chief Engineer of Electronic Packaging Integration in 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 Transactions on CPMT. He has chaired the iNEMI PEG on High-End Systems including the chapter on the High-End Systems Roadmap. He is an IEEE Fellow, an iNEMI Technical Committee member, and a member of SWE.

**Madhavan Swaminathan** is the John Pippin Chair in Microsystems Packaging & Electromagnetics in the School of Electrical and Computer Engineering (ECE) and Founding Director of the Center for Co-Design of Chip, Package, System (C3PS), Georgia

Tech. He formerly held the position of Joseph M. Pettit Professor in Electronics in ECE and Deputy Director of the NSF Microsystems Packaging Research Center, Georgia Tech. Prior to joining Georgia Tech, he was with IBM working on packaging for supercomputers. He is the author of 450+ refereed technical publications, holds 30 patents, primary author and co-editor of 3 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 in the Asian region. His research work has been recognized through several awards including 22 best paper and best student paper awards and 2014 Outstanding Sustained Technical Contribution Award from IEEE EPS Society. He is an IEEE Fellow and has served as the Distinguished Lecturer for the IEEE EMC society. He received his MS/PhD degrees in Electrical Engineering from Syracuse University in 1989 and 1991, respectively.

# 1- Introduction

In the past few decades, speed of high speed channels has increased exponentially [1], which has resulted in significant advancements in high speed systems. However, signal integrity issues that used to be trivial, cannot be ignored as the frequency increases. Degradation of transmitted signal is seen in amplitude and timing, which are called amplitude noise and timing jitter, respectively. In this paper we focus on jitter; nevertheless, a parallel discussion can be done for amplitude noise. In general, jitter is divided to deterministic and random. Random jitter is unbounded, and it is mainly caused by physical properties of materials. Moreover, it can be modeled with Gaussian distributions, and it is common to overlay random jitter on the signal after simulation of the system. On the other hand, deterministic jitter is bounded, which includes Data Dependent Jitter (DDJ), Bounded Uncorrelated Jitter, and Periodic Jitter. This classification is shown in Fig. 1. A source of DDJ is InterSymbol Interference (ISI), which is the deviation added to a pulse response from the response of its neighboring pulses [2]. Since ISI spans through several pulse responses, it is challenging to model. Therefore, in this paper a new approach for modeling DDJ caused by ISI is presented, which can be combined with modeling techniques for other types of jitter to simulate the final form of the received signal.

In digital communication, a signal is often evaluated with an eye diagram. In simulation, the traditional method to draw the eye diagram is with running a very long transient simulation. Then, results are overlaid in one Unit Interval (UI), which resembles an eye shape. An open eye means a better quality for signal, and when its height and width decreases, the quality degrades. We refer to this method as transient eye. This method is straightforward, and provides a great accuracy for lower speeds. However, as the Bit Error Rate (BER) of the channel decreases, the transient simulation needs to be longer, which can be prohibitive for modern channels. For instance, a BER of $10^{-12}$, which has become common in the industry, needs to be simulated for $10^{12}$ bits on average to experience one failure. Obviously, running a simulation with such length is not usually possible. Hence, designers do a shorter simulation up to $10^{10}$ bits, and find the error rate with approximation and extrapolation techniques, which reduces the accuracy. Moreover, this method requires a large amount of memory to store the transient simulation.

To address the issues with transient eye, statistical eye analysis methods have been developed. These methods can quickly predict the eye diagram from impulse response of the channel, when the system is Linear Time Invariant (LTI). One of these approaches is the StatEye method [3], which was introduced earlier than most of the similar approaches. In this method, initially the pulse response is calculated:

$$r(t) = p(t) * h(t), \tag{1}$$

where , $*$ represents the convolution, $r(t)$ is the pulse response at the receiver, $h(t)$ is the channel's impulse response, and $p(t)$ is the input pulse, which is determined with respect to the baud rate. Due to ISI, $r(t)$ spans over multiple unit intervals; however, the response to a sequence of pulses can be found with superposition of shifted pulse responses because of the LTI condition. Therefore the receiver voltage is approximated as:

$$\hat{r}(t) = \beta_0 r(t) + \sum_{\substack{k=-\infty \\ k\neq 0}}^{\infty} \beta_k r(t - kT), \tag{2}$$

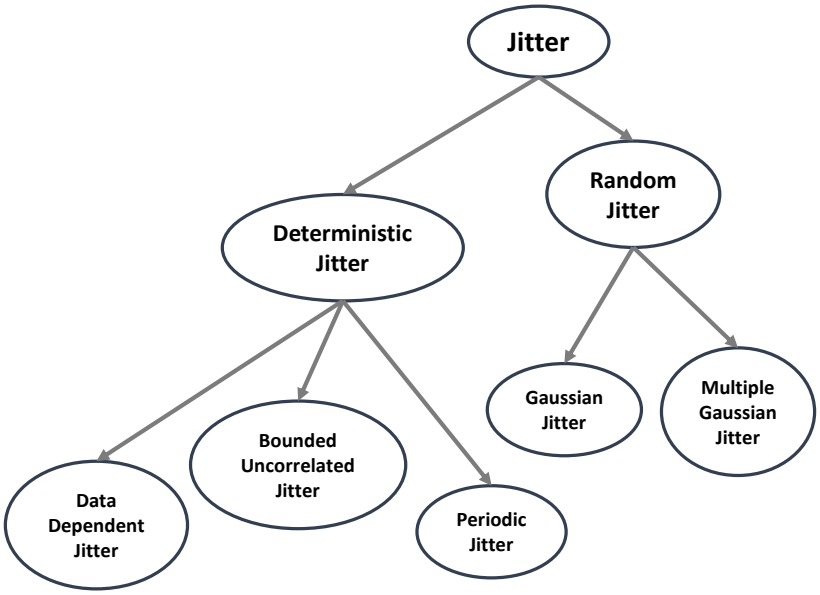



Fig. 1. Classification of jitter types.

where *t* is the sampling time point, T shows length of one UI, and $\beta_k$ represents value of the k-th transmitted symbol, which can be zero or one. In other words, $\beta_0 r(t)$ is response of the current pulse, and $\beta_k r(t - kT)$ is the ISI effect from pre-cursors and post-cursors of neighboring pulses. Equation (2) is the idea behind the StatEye method, and it can predict the receiver voltage for any combination of input bits; however, it is often time and memory consuming to calculate (2) for all possible combinations. Therefore, [3] suggests a statistical approach, that considers probability of different ISI combinations, and finds the distribution of ISI. Using this approach, an array of Probability Distribution Functions (PDFs) of ISI is produced for different sampling points through the UI. By tracking these distributions and connecting the points with same probability, the statistical eye is created. Moreover, results can be used to find BER. Knowing distribution of receiver voltage, BER is found as ratio of the distribution that crosses time and voltage thresholds, causing error. Furthermore, a bathtub curve is calculated by finding BER over one UI [1].

StatEye provides a quick estimation of the eye diagram; however, it is only applicable to LTI systems because of using superposition in (2). Unfortunately, this issue can be a problem in simulation of high speed links since sources of nonlinearity, such as I/O ports, can be present and constitute to jitter. Moreover, asymmetrical rising and falling edges are another sign of non-LTI systems, and can be present in signaling systems. Finally, StatEye has difficulties in modeling transmitter jitter [4]. Because of these issues, StatEye produces inaccurate results in some practical examples.

Recent works have tried to expand the idea in StatEye to non-linear cases. In [4], it is suggested to estimate the receiver voltage with responses of a rising edge and a falling edge. This is similar to (2); however, the difference between rising edge and falling edge responses is taken into account. The receiver voltage is calculated as:

$$\hat{r}(t) = \sum_{k=-\infty}^{\infty} \alpha_k v_r(t - kT) + \beta_k v_f(t - kT), \qquad (3)$$

where, $v_r(t)$ and $v_f(t)$ are the responses to rising and falling edges, respectively. Moreover, $\alpha_k$ and $\beta_k$ are zero or one factors that show the occurrence of rising and falling edges. $\alpha_k$ is one when a rising edge happens, and it is zero otherwise. Similarly, $\beta_k$ is one when a falling edge happens, and it is zero otherwise. The receiver voltage is found with (3); however, finding the worst-case eye or the statistical eye is more challenging than the StatEye method since the edge responses are not independent. For instance, there should be a rising edge between two falling edges. Therefore, [4] develops an inductive technique to find the distribution of receiver voltage. This method starts with the response to a rising edge and a falling edge, which are followed by *n* high and low bits, respectively, where *n* is the number of bits with effective ISI. Distribution of receiver voltage after *n* bits is two impulses and known since ISI has disappeared. Then, this distribution is used to find distribution of the receiver voltage after *n*-1 bits with convolution equations. This process is continued for *n* steps to find distribution of the receiver voltage after zero bits, or in other words distribution of the receiver voltage. Finally, the receiver voltage distribution is used to find the statistical eye diagram, BER, and bathtub curve similar to the StatEye approach.

Although the edge response method improves the accuracy of statistical eye for non-LTI systems, it is not comprehensive. It requires some internal knowledge of the system, and needs to know the source of nonlinearity. In other words, it does not work if nonlinearity is caused by a source other than the asymmetric rising and falling edges. Moreover, it is an extension of the superposition method, and might fail based on the nonlinearity. Finally, statistical methods might still need a large amount of memory based on their implementation.

Since the transient eye analysis is time and memory consuming, and statistical eye analysis methods have limited use, a new approach is proposed in this paper, based on the work in [5]. This approach takes an uncertainty quantification point of view, where the propagation of randomness from input to output random variables is studied. This relates to the problem studied in this paper since we are looking to find the pattern between data and the resulted jitter, where both have a random nature. The traditional method for uncertainty quantification is Monte Carlo (MC) analysis; however, where high accuracy is needed, MC would be extremely time consuming. Therefore, in recent decades several efficient uncertainty quantification approaches have been developed [6] – [19]. A popular concept in developing such methods, is the generalized Polynomial Chaos (PC) theory [9], where random variables are estimated as sum of a series of orthogonal polynomials. This work is inspired by the similarities between the transient eye analysis and MC, and proposes a modified uncertainty quantification approach, based on the PC expansion, to drastically decrease the computation cost. The PC expansion is selected since it provides statistics of jitter and eye analysis prior to calculating the eye diagram, and with using significantly less memory. Furthermore, the PC expansion is used to develop surrogate models for receiver voltage and drawing the actual eye diagram. It is worth noting, that the proposed surrogate models do not have a restriction on nonlinear functions, and can be used for non-LTI systems.

Table. 1. The first five normalized Hermite polynomials

| | Normalized Hermite Polynomials |
|---|---|
| $H_0(\lambda)$ | 1 |
| $H_1(\lambda)$ | $\lambda$ |
| $H_2(\lambda)$ | $(\lambda^2-1)/\sqrt{2}$ |
| $H_3(\lambda)$ | $(\lambda^3-3\lambda)/\sqrt{6}$ |
| $H_4(\lambda)$ | $(\lambda^4-6\lambda^2+3)/(2\sqrt{6})$ |

This work trains a surrogate model from a small number of samples to imitate behavior of a system. However, in machine learning many approaches have been suggested for developing surrogate models. One of the most popular approaches in this area is modeling with Neural Networks (NN). Previously, nonlinear drivers have been modeled in [20], where recurrent NNs were used to develop models for the nonlinear components. However, in general training NNs is a more complicated and time consuming task than PC models, and their size quickly increases with memory of the channel. Moreover, NN models need to be evaluated at numerous samples, similar to MC, to provide statistical results. This takes a large amount of memory; however, PC models provide the statistical results at no extra cost after training. For these reasons, PC theory is used to develop the surrogate model of the channel.
The rest of this work is organized as follows. In section 2, the PC theory is discussed. Next, the proposed approach is explained in section 3. Section 4 represents an example for evaluation of this method. Finally, the paper is concluded in section 5.

# 2- Polynomial Chaos theory

Polynomial Chaos expansion [9] is commonly used in uncertainty quantification approaches. In this section, its details are discussed.

## 2-1- PC expansion

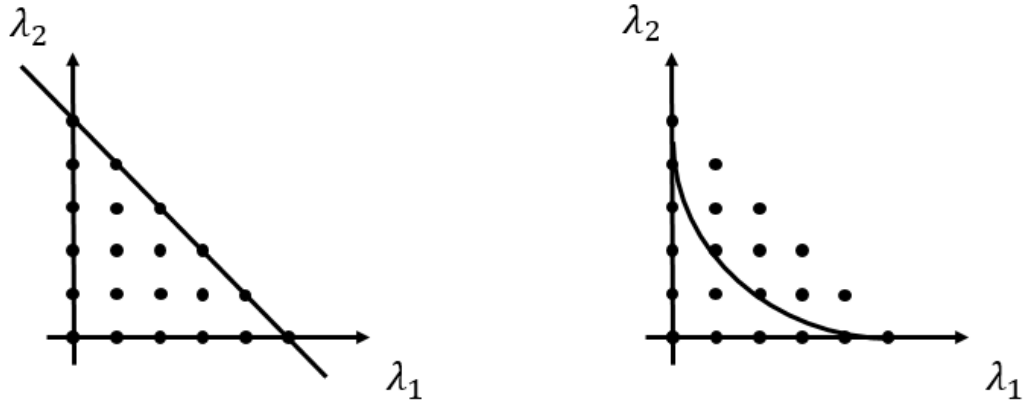
PC expansion suggest using sum of a series of orthogonal polynomials for approximating smooth functions of random variables. In other word, for one dimension a function of random variable $\lambda$ is estimated as:

$$f(\lambda) = \sum_{i=0}^{\infty} c_i \phi_i(\lambda), \tag{4}$$

where $c_i$ shows a series of unknown coefficients, and $\phi_i(\lambda)$ represents the polynomials, which are orthogonal with respect to the distribution of $\lambda$. Orthogonality condition is defined as:

$$\langle \phi_i(\lambda), \phi_j(\lambda) \rangle = \int_{\Omega} \phi_i(\lambda)\phi_j(\lambda)\rho(\lambda)d\lambda = \alpha_i^2 \delta_{i,j}, \tag{5}$$

with $\langle , \rangle$ being the sign for inner product, and $\Omega$ being the random space. Moreover, $\rho$ is the distribution of $\lambda$, $\delta_{i,j}$ is the Dirac function, and $\alpha_i$ is a scalar, which can be calculated with a closed-form equation. For simplification, we always normalize the polynomials; thus, $\alpha_i$ would always be equal to one. Furthermore, for common distributions there are known polynomials that are orthogonal with respect to them. For instance, Legendre and Hermite polynomials are orthogonal with respect to uniform and normal distributions,




(a) (b)

Fig. 2. Illustration of restrictions for choosing polynomials for a case with two dimensions. a) Linear PC condition. b) Hyperbolic HPC condition.

respectively. Random Non-Return-to-Zero (NRZ) pulses, considered in this paper, have zero and one values with equal probability. Hermite polynomials are orthogonal to this distribution as well; hence, they are used for developing the proposed PC models. Hermite polynomials are generated analytically using the following equation:

$$H_i(\lambda) = (-1)^i e^{\frac{\lambda^2}{2}} \frac{d^i}{d\lambda^i} e^{-\frac{\lambda^2}{2}}, \tag{6}$$

with $H_i(\lambda)$ being the *i*-th degree Hermite polynomial. Moreover, the normalization factor is calculated as $\alpha_i^2 = i!$. As an example, $H_0(\lambda)$ to $H_4(\lambda)$ are presented in Table. 1.
In the multidimensional case, the PC expansion is extended for a vector of *n* variables, $\boldsymbol{\lambda} = [\lambda_1, \lambda_2, \dots, \lambda_n]$, as:

$$f(\boldsymbol{\lambda}) \approx \sum_{i=0}^{P} c_i \phi_i(\boldsymbol{\lambda}), \tag{7}$$

where the expansion is truncated after *P* + 1 terms for practical reasons. This length is calculated as:

$$P + 1 = \binom{m+n}{m} = \frac{(m+n)!}{m!n!}, \tag{8}$$

where, *m* is the maximum order used in the polynomials, which usually does not need to be greater than 2 or 3 for smooth functions. Furthermore, the multidimensional polynomials are generated using a subset of the tensor product of one-dimensional polynomials:

$$\phi_i(\boldsymbol{\lambda}) = \prod_{j=1}^{n} \phi_{\boldsymbol{d}_j}(\lambda_j), \tag{9}$$

where $\boldsymbol{d} = [d_1, d_2, \dots, d_n]$ is a vector showing the index of selected one-dimensional polynomials to generate $\phi_i$, and it is determined using the following condition:

$$\|\boldsymbol{d}\|_1 = d_1 + d_2 + \cdots + d_n \leq m. \tag{10}$$

This condition represents a linear restriction, which is shown in Fig. 2 (a) for two dimensions.

It is worth noting, that (7) can be used as a surrogate model to estimate $f(\boldsymbol{\lambda})$ at arbitrary samples. However, its size, shown in (8), increases with a rate of $O(n^m)$, which limits the use of PC expansion to a relatively low number of variables. This problem is called curse of dimensionality, and to address it a modified PC expansion, called Hyperbolic Polynomial Chaos (HPC), is developed [17]. In this approach, it is noted that not all the polynomials in (7) are equally important. In fact, based on sparsity of effects [21], it is known that polynomials with a lower degree of interaction between the random variables have a higher impact on the output. Alternatively, it means these terms have a higher coefficient in the PC expansion. Therefore, a new expansion is developed:

$$f(\boldsymbol{\lambda}) \approx \sum_{i=0}^{\hat{P}} c_i \hat{\phi}_i(\boldsymbol{\lambda}), \tag{11}$$

where $\hat{\phi}_i(\boldsymbol{\lambda})$ shows the nontrivial polynomials; which are selected with (9) and the following condition:

$$\|\boldsymbol{d}\|_u^u = d_1^u + d_2^u + \cdots + d_n^u \leq m^u, \tag{12}$$

with $0 < u < 1$ being a factor for setting the level of reducing the expansion. This condition represents a hyperbolic restriction, shown in Fig. 2. (b) for two dimensions. HPC limits the polynomials to the ones with a higher impact, which are located closer to the multidimensional axis. Moreover, it is proven that the polynomials in both (7) and (11) follow the orthogonality condition in the multidimensional space. In this paper, the proposed approach uses (11) for modeling the random variables.

## 2-2 Training the PC models

Training the PC models is done by finding the coefficients in (11). Several methods have been introduced to find the unknown coefficients in the PC expansion [6]-[18]. In general, these methods are divided to intrusive and nonintrusive. Intrusive methods, such as Stochastic Galerkin [12], replace all the random variables in the equations governing the system (e.g. Kirchhoff's laws, Telegrapher's equations, etc.) with their PC expansion. The replacement results in development of an augmented and more complex set of equations. Then, PC coefficients are found by solving these augmented equations. Although intrusive approaches are relatively more accurate, commercial solvers do not support this type of analysis, and these methods need to develop new stochastic solvers or augmented systems to find the coefficients. Moreover, size of the augmented equations grows drastically with number of random variables, and the CPU cost is often prohibitive when there are more than a handful of random variables. On the other hand, non-intrusive approaches are based on running simulations for a number of sample points. Hence, these methods do not need to develop new solvers or systems, and work with deterministic commercial solvers. For instance, Monte Carlo is a non-intrusive approach. Although non-intrusive PC approaches are less accurate compared to intrusive approaches, their computational cost scales at a lower rate with number of random variables. Therefore, they can be applied to a higher number of random variables. In this category, a straightforward technique to find the PC coefficient is linear regression [10], [11].
The linear regression method is used in the proposed approach, and it is discussed next. In this method, initially the system is sampled at *N* points, with respect to the distribution of random variables. The sample are labeled as $\boldsymbol{\lambda}^1, \boldsymbol{\lambda}^2, \dots, \boldsymbol{\lambda}^N$. Moreover, $N \geq kL$, where *L* is the length of PC expansion, and *k* is equal or greater than 2 or 3. By using a

simulation tool and evaluating the system at these samples, the outputs $f(\boldsymbol{\lambda}^1), f(\boldsymbol{\lambda}^2), \dots, f(\boldsymbol{\lambda}^N)$ are obtained. Samples and the corresponding outputs form the training data, and by plugging them in the PC expansion, results are written in the matrix form:

$$\boldsymbol{A}\boldsymbol{\Gamma} = \boldsymbol{E}, \tag{13}$$

where

$$\boldsymbol{A} = \begin{bmatrix} \phi_0(\boldsymbol{\lambda}^1) & \dots & \phi_P(\boldsymbol{\lambda}^1) \\ \vdots & \ddots & \vdots \\ \phi_0(\boldsymbol{\lambda}^N) & \dots & \phi_P(\boldsymbol{\lambda}^N) \end{bmatrix}, \boldsymbol{\Gamma} = \begin{bmatrix} c_0 \\ \vdots \\ c_P \end{bmatrix}, \mathbf{E} = \begin{bmatrix} f(\boldsymbol{\lambda}^1) \\ \vdots \\ f(\boldsymbol{\lambda}^N) \end{bmatrix}. \tag{14}$$

The system shown in (13) is oversampled since $N > P + 1$; thus, it does not have a perfect solution. Nevertheless, the error can be minimized by using the least square method:

$$\hat{\boldsymbol{\Gamma}} = \underset{\boldsymbol{\Gamma}}{Argmin} \|\boldsymbol{E} - \boldsymbol{A}\boldsymbol{\Gamma}\|_2^2, \tag{15}$$

It is proven that the error is minimized when:

$$\hat{\boldsymbol{\Gamma}} = (\boldsymbol{A}^\tau \boldsymbol{A})^{-1} \boldsymbol{A}^\tau \boldsymbol{E}, \tag{16}$$

where, $\tau$ is the transpose sign.
After finding the coefficients, they can be used to provide mean and variance of the output at no extra cost. Mean of $f(\boldsymbol{\lambda})$ is defined as:

$$E\big(f(\boldsymbol{\lambda})\big) = \int_\Omega f(\boldsymbol{\lambda})\rho(\boldsymbol{\lambda})d\boldsymbol{\lambda}. \tag{17}$$

Calculation of this multidimensional integral can be challenging; however, by plugging the PC expansion of $f(\boldsymbol{\lambda})$ in (17), and considering the similarities between (5) and (17), it can be written as:

$$E[f(\boldsymbol{\lambda})] \approx \sum_{i=0}^{P} \langle c_i \phi_i(\boldsymbol{\lambda}), \phi_0(\boldsymbol{\lambda}) \rangle = c_0. \tag{18}$$

In other words, mean of a random variable is equal to the first coefficient in its PC expansion. Furthermore, Variance of $f(\boldsymbol{\lambda})$ is defined as:

$$var[f(\boldsymbol{\lambda})] = \sigma^2 = E[(f(\boldsymbol{\lambda}) - E[f(\boldsymbol{\lambda})])^2]. \tag{19}$$

Like (18), by replacing the PC expansion of $f(\boldsymbol{\lambda})$, and using the definition of inner product, (19) is simplified to:

$$Var[f(\boldsymbol{\lambda})] = \sigma^2 = \sum_{i=1}^{P} c_i^2. \tag{20}$$

Alternatively, it can be said that variance of a random variable is equal to sum of square of all the coefficients, except the first one, in its PC expansion.
To have a better evaluation of random variables, it is necessary to see their distribution. Therefore, PC surrogate models are used to find PDFs of random variables. This is done by evaluating (11) at several sample points similar to MC. However, this evaluation is much faster since it is analytical, and no additional simulation is needed.

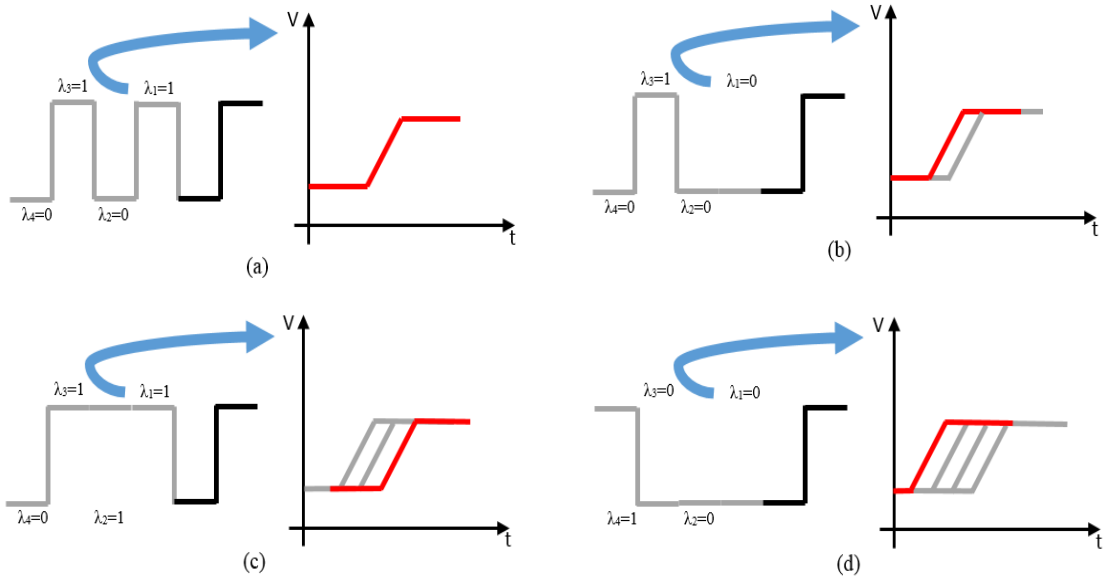



Fig. 3. Simplified illustration of different input pulse sequences and the corresponding rising edge.

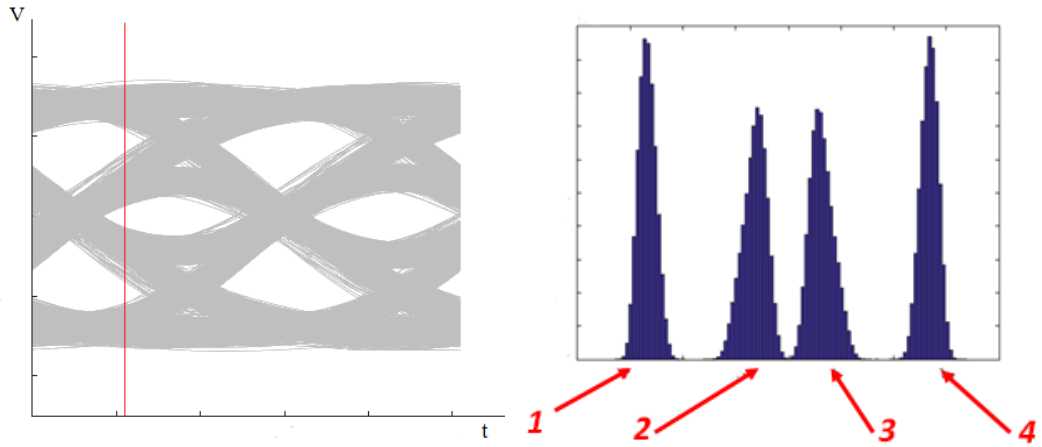
# 3- The proposed modeling approach

In this section, the idea behind the proposed approach and details of its implementation are discussed.

## 3-1- Intuition behind the proposed approach

The intuition behind this work is similarities between the transient eye and Monte Carlo analysis. We observed that eye parameters such as jitter are random variables and a function of random input bit sequences. In fact, transient eye is a method that finds the behavior of such parameters with running a tedious time domain simulation, while it tries numerous data patterns. On the other hand, in MC analysis randomness in the system is evaluated with running simulations at numerous samples, which are selected randomly. Therefore, by treating each data pattern as a sample, transient eye resembles the MC analysis. Moreover, both methods provide a straightforward implementation with high accuracy; however, their computation cost can be prohibitive for complex systems since they need to consider many possibilities. In uncertainty quantification studies, several approaches have been developed to provide results as accurate as MC, while the computation cost is kept low. Our goal is to develop a similar methodology with high efficiency for the eye analysis.

This idea is illustrated in Fig. 3, where the simplified response to a sample pulse train, ending in a rising edge, is shown in Fig. 3 (a). Next, the bits before the rising edge are slightly altered in Fig. 3 (b), which changes the ISI effect. The effect of the new ISI is seen in this figure as movement of the rising edge, while the old rising edge is shown in a gray shade. The input is altered again in Fig. 3 (c) and (d), which has resulted in further changes in the rising edge, and it is seen that the rising edge might move to left or right based on the previous bits. In the proposed approach, variations in the input pulse train and the output are considered as random variables, and the relation between them is




(a) (b)

Fig. 4. Demonstrating a typical eye diagram and possible transitions of the output. a) Eye diagram. b) Distribution of receiver voltage at a sampling time point.

captured with the PC expansion. In other words, the proposed approach learns the pattern between data and ISI from a short simulation. Then, it is generalized by training a surrogate model, and it is used to estimate the output for any possible combination of input bits. It is worth noting that this approach is applicable to non-LTI systems.

## 3-2- Implementation of the proposed approach

In this section, implementation of the proposed PC approach for estimation of jitter and eye diagram, is discussed. First, we need to note that the output signal covers a wide range of variation, when it switches between low and high. This is shown in a typical eye diagram in Fig. 4 (a). Using conventional PC to model a variable with large variance, usually does not provide accurate results. To address this issue, we exploit the fact that eye diagram is comprised from four distinct transitions, which are caused by the last two bits, and are zero to zero, zero to one, one to zero, and one to one. Bits prior to these two cause ISI, and make each transition to deviate from its average value. This is shown by drawing the distribution of receiver voltage at the red line in Fig. 4 (a), which is represented in Fig. 4 (b). In this figure, four distinct distributions are observed, which correspond to the four transitions. It is seen that each distribution has a much smaller variation compared to the original eye; therefore, we develop a separate sub-model for each transition. Inputs to each model are the values of $n$ previous bits except the last two, which are used to select the sub-model. Hence, the input bits are labeled as $\boldsymbol{\lambda} = [\lambda_1, \lambda_2, \dots, \lambda_{n-2}]$. Since the input pulse train is ideal, these variables are either zero or one, and set randomly with equal chance. Next, two types of output are defined. The first type is eye diagram parameters such as jitter, which is directly and efficiently modeled. The second type is the receiver voltage through one UI, which is used for drawing the actual eye-diagram.

To develop PC models, first the training data needs to be collected, which is done by sampling different combinations of previous bits. However, the CPU cost would be exorbitant if a separate simulation is run for each sample. Therefore, all the samples are extracted from a single simulation, which is significantly shorter than the transient eye simulation and much longer than $\hat{P} + 1$ UIs. To extract the samples, we move backward

from the end of simulation, and at every UI record the outputs and their corresponding previous *n* bits. Next, the samples are divided to four groups based on transition of the last two bits to be used in relevant sub-models.

In modern high-speed channels, number of bits causing ISI can span through a large number of bits (e.g., 50). Therefore, developing a conventional PC model, shown in (7), would be extremely large and prohibitive. To address this issue, we suggest using the HPC expansion, represented in (11). In fact, through numerical examples it was observed that a $u$ factor close to zero could provide accurate results. This low factor removes the interactions between dimensions, and only leaves $\hat{P} = m * n + 1$ polynomials, which are located on the axis of the multidimensional space. Therefore, size of the model increases linearly with number of considered previous bits.

To directly model jitter, only two of the transitions, which are rising and falling edges, need to be considered. The corresponding models are:

$$\begin{aligned} J_r(\boldsymbol{\lambda}) &\approx \sum_{i=0}^{\hat{P}} C_{r\_i}\hat{\phi}_i(\boldsymbol{\lambda}) \\ J_f(\boldsymbol{\lambda}) &\approx \sum_{i=0}^{\hat{P}} C_{f\_i}\hat{\phi}_i(\boldsymbol{\lambda}), \end{aligned} \tag{21}$$

where $C_{r\_i}$ and $C_{f\_i}$ are the coefficients of rising and falling edge sub-models, respectively. Equation (21) can be used as a surrogate model to estimate jitter for any combination of input data. Furthermore, we calculate jitter of rising and falling edges separately. Since their average is zero, their RMS value is equal to standard deviation, which is directly calculated in (20). It is worth noting, that the RMS value is found without calculating the full eye diagram, or using an approach like MC to evaluate the surrogate model at numerous samples. Therefore, significant savings in computational cost and memory is observed.

Although jitter provides a measure for degradation of signal, we generally prefer to see the actual eye diagram too. To do so, PC surrogate models are developed for the receiver voltage at each sampling time point on a UI:

$$\begin{aligned} V_{00}(t,\boldsymbol{\lambda}) &\approx \sum_{i=0}^{\hat{P}} C_{00\_i}(t)\hat{\phi}_i(\boldsymbol{\lambda}) \\ V_{01}(t,\boldsymbol{\lambda}) &\approx \sum_{i=0}^{\hat{P}} C_{01\_i}(t)\hat{\phi}_i(\boldsymbol{\lambda}) \\ V_{10}(t,\boldsymbol{\lambda}) &\approx \sum_{i=0}^{\hat{P}} C_{10\_i}(t)\hat{\phi}_i(\boldsymbol{\lambda}) \\ V_{11}(t,\boldsymbol{\lambda}) &\approx \sum_{i=0}^{\hat{P}} C_{11\_i}(t)\hat{\phi}_i(\boldsymbol{\lambda}), \end{aligned} \tag{22}$$

where, $C_{00\_i}$, $C_{01\_i}$, $C_{10\_i}$, and $C_{11\_i}$ show the coefficients of steady zero, rising edge, falling edge, and steady one sub-models, respectively. Equation (22) is used to estimate the receiver voltage for any combination of data, additionally statistics of each transition are found with (18) and (20) at no extra cost. Nevertheless, to draw the eye diagram, (22) needs to be evaluated at numerous number of points. There are $2^n$ possible combinations for the input bits; however, evaluating the model for all of them is not possible when *n* is large. Hence, we choose a random subset of them, which includes about one million samples, to draw the eye diagram.

Advanced regression techniques exist in the literature [15], [18]; however, they require complex sampling techniques, which is not possible in the setting of the problem considered in this paper. Therefore, the linear regression and least square method,

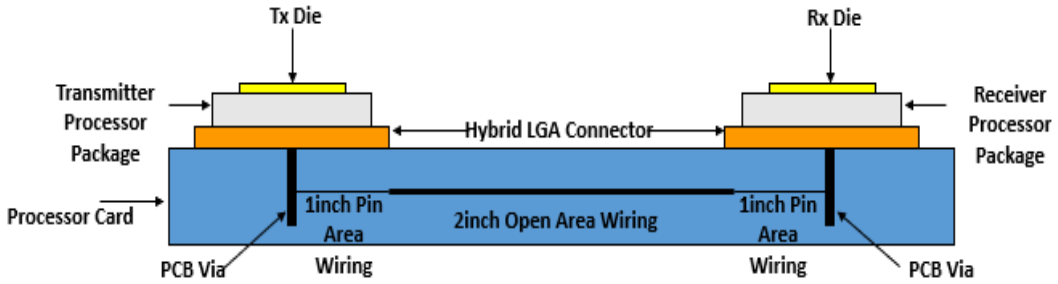



Fig. 5. A graphical representation of the SerDes channel in the example.

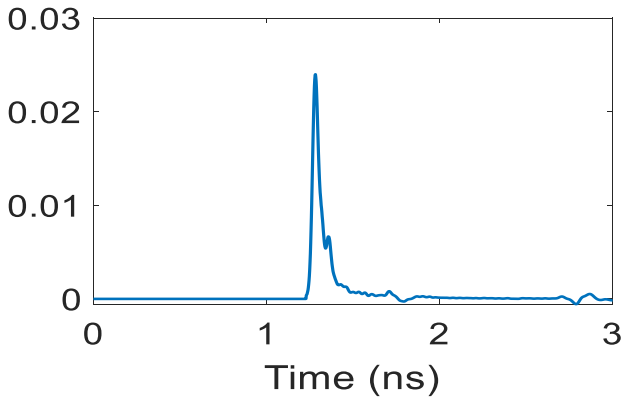



Fig. 6. Impulse response of the SerDes channel.

described in 2-2, is used to find the coefficients in (21) and (22). First, $N$ training samples are extracted from a single simulation, and are labeled as $\boldsymbol{\lambda}^1$ to $\boldsymbol{\lambda}^N$. Then, the coefficients are found using (16). For instance, regarding the $V_{01}(t, \boldsymbol{\lambda})$ sub-model in (22):

$$\widehat{\boldsymbol{\Gamma_{01}}}(\mathrm{t}) = (\boldsymbol{A}^\tau \boldsymbol{A})^{-1} \boldsymbol{A}^\tau \boldsymbol{E}(t), \tag{23}$$

where

$$\widehat{\boldsymbol{\Gamma_{01}}}(\mathrm{t}) = \left[C_{01\_0}(t), C_{01\_1}(t), \dots, C_{01\_P}(t)\right]^\tau. \tag{24}$$

A few edits are done to make the matrix inversion in (22) possible. First, if a random variable in $\boldsymbol{\lambda}$ is zero, it is converted to -1, otherwise several polynomials would be zero and equal. Moreover, matrix **A** needs to be full rank. Because all the variables are either -1 or 1 this might not be true. Therefore, columns that are a linear conversion of other columns are removed. This does not decrease the accuracy since these columns do not provide any additional information. Moreover, note that since the inputs are the same for sub-models in (21) and (22), the same matrix **A** is used for all the model. To further increase the efficiency, this matrix is calculated and stored beforehand, and a known sequence of bits is used for training. This method covers the DDJ estimation; however, it can be combined with other types of jitter estimation methods, such as random jitter, to provide a more general estimation approach.

Finally, it is worth noting that this approach is different from other HPC calculation methods [17]. Since, it modifies the process to be efficient and applicable to jitter and eye estimation for high speed channels. Modifications include developing different sub-models for different transitions, and the sample extraction method.

Table. 2. RMS values of jitter in rising and falling edges.

| | Rising edge jitter RMS | Falling edge jitter RMS |
|---|---|---|
| Transient eye analysis | 7.1 ps | 7.1 ps |
| Proposed PC approach | 7.1 ps | 7.0 ps |

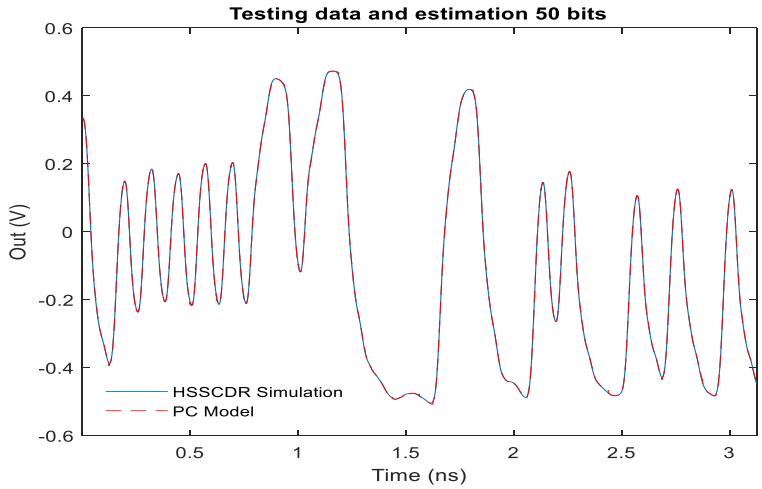



Fig. 7. Estimation of the output for 50 bits using the voltage surrogate models.

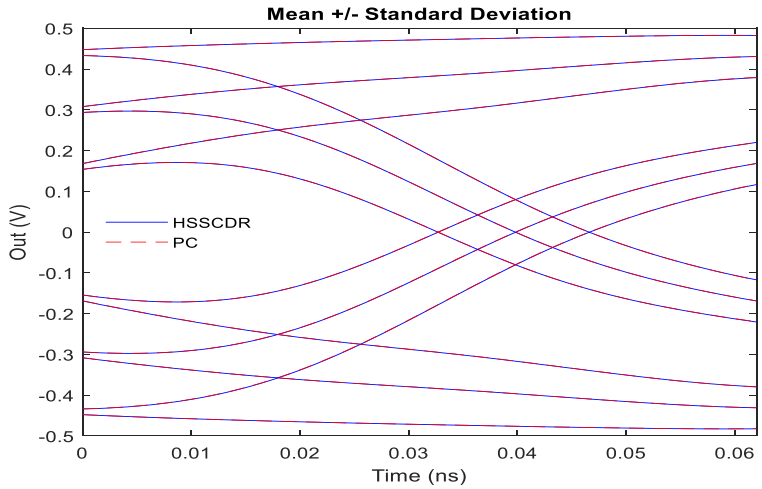
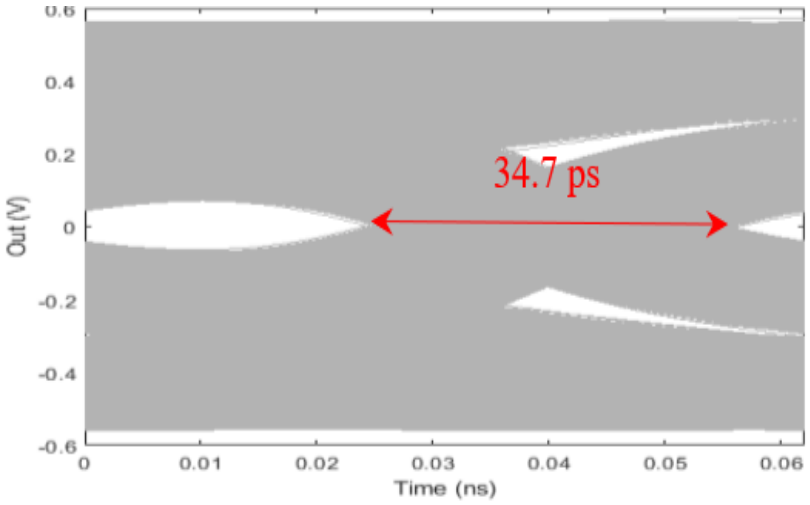


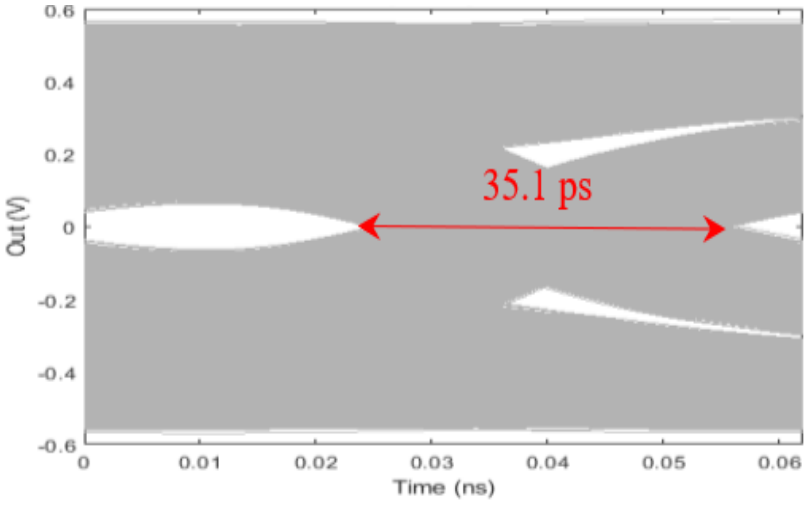

Fig. 8. Mean +/- standard deviation of the receiver voltage for steady zero, rising edge, falling edge, and steady one transitions through one UI.

# 4- Numerical example

To show performance of the proposed approach the SerDes channel, pictured in Fig. 5, is considered. This channel is provided by IBM, and it includes two processor packages. These packages communicate with each other and are interfaced to a board with two hybrid Land Grid Array (LGA) connectors. Moreover, the transmitter and receiver processor packages contain 85 Ohm differential stripline wiring in GZ41 material (Dk ~




(a)




(b)

Fig. 9. Eye diagram and peak to peak jitter of the receiver voltage. a) Calculated with the traditional transient eye. b) Estimation with the proposed approach.

3.31 and Df ~ 0.0092 at 1 GHz), and have 31 mm and 34 mm length, respectively. Furthermore, the board contains two differential PCB vias with an active via length of 150 mil, and stub length of 20 mil. It contains 4 inches of total wiring, including 1 inch of necked down pin area wiring in the shadow of each processor, and 2 inches of 85 Ohm differential open area wiring. Additionally, the dielectric material utilized for the board is a low loss material, having a Dk ~ 3.95 and Df ~ 0.0084 at 1GHz. The passive channel loss at 8 GHz is ~11 dB. Finally, the input signal is an NRZ pulse train with a bit rate of 16 Gb/s.

The impulse response of the channel is shown in Fig. 6. Moreover, this channel is simulated with HSSCDR channel simulator [22], provided by IBM, and the rest of computations, including development of HPC models, is done in MATLAB R2015a. Note that HSSCDR is a linearized simulator; however, the proposed approach does not take advantage of the linearity condition. Hence, it can be applied to similar commercial software with nonlinear components.

In this example, about $n$ = 50 bits show nontrivial ISI effect. Therefore, each of the four-sub-models has 48 dimensions, which means the conventional PC models would be too large. Therefore, the HPC expansion, with $u$ near zero, is used. Maximum order of expansion, $m$ = 3, is selected for developing the models; thus, length of expansion $\hat{P}$ is equal to 145.

For comparing the accuracy of PC models, a transient eye analysis with a sequence of one million bits is done. On the other hand, for training the PC models a simulation with only 10000 bits is used; thus, the proposed approach has decreased the length of actual simulation 100 times. For developing jitter sub-models, 2000 samples for each sub-model is extracted from the short simulation, and by using the linear regression technique, the coefficients are found. Next, jitter RMS values are directly found from the coefficients, and presented in Table. 2. Additionally, they are compared with the results from transient eye, and it is observed that the proposed approach shows a great accuracy in this estimation.

Next, to develop the receiver voltage sub-models, 2000 samples per transition are extracted from the same short simulation, and they are used to find the PC coefficients. The voltage sub-models are put together to form a surrogate model, which predicts the receiver voltage for any combination of input bits. This surrogate model is used to predict the response to a new sequence of 50 bits, which is represented in Fig. 7, and compared with simulation results from HSSCDR. Moreover, the coefficients from receiver voltage sub-models are used to find the mean and standard deviation of possible voltage transitions. These statistics are represented in Fig. 8, where for each transition mean +/- one standard deviation is shown, and compared with the same statistics from the one million bits simulation. It is observed that both figures show a perfect match between the proposed approach and the actual simulation.

Finally, the eye diagram is generated by evaluating the surrogate model of the receiver voltage at one million new samples, which are randomly selected. Eye diagrams found with the original transient eye and this method are shown in Fig. 9 (a) and (b), respectively. For technical reasons, only 30000 out of one million responses are shown in these figures. Furthermore, the peak to peak jitter, found from the eye diagrams, are shown in Fig. 9. It is observed that the estimated eye and peak to peak jitter are a good approximation of the actual values.

## 5- Conclusion

Calculation of data dependent jitter and eye diagram can be challenging for modern high-speed channels since the traditional transient eye is extremely time consuming, and the statistical methods are only applicable to LTI systems. Hence, in this paper an uncertainty quantification approach is suggested. This approach uses a hyperbolic Polynomial Chaos expansion to develop a surrogate model for a case with a large number of random variables. Then, the surrogate model is used to find estimations and statistics of jitter and eye diagram. Furthermore, the surrogate model is comprised from multiple sub-models to limit the variance of output. Models are trained with linear regression and a single short transient simulation, which can be 100 times shorter than the simulation necessary for transient eye. Additionally, the overhead cost for developing the surrogate model is negligible in comparison to the cost of transient simulation of complex channels. Moreover, this method can handle non-LTI systems. Finally, a numerical example is

provided which shows the accuracy and efficiency of the proposed approach, when applied to a commercial SerDes channel.

## Acknowledgement

This material is based upon work supported by the National Science Foundation under Grant NO CNS 16-24810 – Center for Advanced Electronics through Machine Learning (CAEML).

## References

[1] T. Wu, F. Buesink, and F. Canavero. "Overview of signal integrity and EMC design technologies on PCB: Fundamentals and latest progress." *IEEE transactions on electromagnetic compatibility*, vol. 55, no. 4, pp. 624-638, 2013.
[2] M. P. Li, *Jitter Noise and Signal Integrity at High-Speed*, Upper Saddle River, NJ, Prentice Hall, 2007, pp. 1-17, pp. 119-128.
[3] A. Sanders, M. Resso, J. D. Ambrosia, "Channel Compliance Testing Using Novel Sta-tistical Eye Methodology", DesignCon, 2004.
[4] M. Tsuk, D. Dvorscak, C. S. Ong, J. White, "An Electrical-Level Superposed-Edge Approach to Statistical Serial Link Simulation," International Conference on Computer-Aided Design, November 2009, ACM.
[5] M. Ahadi, J. Hejase, W. Becker, M. Swaminathan, "Jitter and Eye Estimation in SerDes Channels using Modified Polynomial Chaos Surrogate Models", in Proc. 27rd IEEE Conf. on Electrical Peroformance and of Electronic Packaging and Systems, Oct. 2018.
[6] R. G. Ghanem, and P. D. Spanos, *Stochastic Finite Elements: A Spectral Approach*, NY, Springer, 1991.
[7] P. Manfredi, "High-speed interconnect models with stochastic parameter variability," Ph.D. dissertation, Informat. Comm. Tech., Politecnico di Torino, Turin, Italy, 2013
[8] D. Xiu, "Fast numerical methods for stochastic computations: a review", Communications in computational physics, vol. 5, no. 2-4, pp. 242-272, 2009
[9] D. Xiu, G. E. Karniadakis. "The Wiener-Askey polynomial chaos for stochastic differential equations." *SIAM journal on scientific computing*, vol. 24, no. 2, pp. 619-644, 2002.
[10] D. Spina, F. Ferranti, G. Antonini, T. Dhaene and L. Knockaert, "Non-intrusive polynomial chaos-based stochastic macromodeling of multiport systems", in Proc. IEEE 18th Workshop on Signal and Power Integrity, pp. 1-4, May 2014
[11] D. Spina, D. De Jonghe, D. Deschrijver, G. Gielen, L. Knockaert, and T. Dhaene, "Stochastic macromodeling of nonlinear systems via polynomial-chaos expansion and transfer function trajectories", IEEE Trans. Microwave Theory Tech, vol. 62, no. 7, pp. 1454-1460, July 2014
[12] P. Manfredi, D. Vande Ginste, D. De Zutter and F. Canavero, "Stochastic modeling of nonlinear circuits via SPICE-compatible spectral equivalents", IEEE Transactions on Circuits and Systems, vol. 61, no. 7, pp. 2057-2065, July 2014
[13] Z. Zhang, T. A. El-Moselhy, I. M. Elfadel and L. Daniel, "Stochastic testing method of transistor level uncertainty quantification based on generalized polynomial chaos", IEEE Transactions on Computer Aided Design, vol. 32, no. 10, pp. 1533-1545, Oct. 2013

[14] P. Manfredi, D. Vande Ginste, D. De Zutter and F. Canavero, “Generalized decoupled polynomial chaos for nonlinear circuits with many random parameters”, IEEE Microwave and Wireless Components Letters, vol. 25, no. 8, pp. 505-507, Aug. 2015
[15] P. Manfredi and F. Canavero, “Efficient statistical simulation of microwave devices via stochastic testing-based circuit equivalents of nonlinear components”, IEEE Transactions on Microwave Theory and Techniques, vol. 63, no. 5, pp. 1502-1511, May 2015
[16] A. K. Prasad, A. Roy. "A novel dimension fusion based polynomial chaos approach for mixed aleatory-epistemic uncertainty quantification of carbon nanotube interconnects." *IEEE International Symposium on Electromagnetic Compatibility & Signal/Power Integrity (EMCSI)*, 2017.
[17] M. Ahadi, A. K. Prasad and S. Roy, “Hyperbolic polynomial chaos expansion (HPCE) and its application to statistical analysis of nonlinear circuits”, IEEE 20th Workshop on Signal and Power Integrity (SPI), pp. 1-4, May 2016
[18] M. Ahadi and S. Roy, “Sparse linear regression (SPLINER) approach for efficient multidimensional uncertainty quantification of high-speed circuits”, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 35, no. 10, pp. 1640-1652, 2016
[19] C. Chauviere, H. S. Jan, C. W. Lucas, “Efficient computation of RCS from scatterers of uncertain shapes”, IEEE Transaction on Antennas and Propagation, 55.5 : 1437-1448, 2007
[20] B. Mutnury, M. Swaminathan, and J. P. Libous. "Macromodeling of nonlinear digital I/O drivers", IEEE Transactions on Advanced Packaging, vol. 29, no.1, pp. 102-113, 2006.
[21] D. C. Montgomery, *Design and Analysis of Experiment*, 8th ed. John Wiley, NY, 2012.
[22] C. Sungjun, J. Hejase, et al., "Package and Printed Circuit Board Design of a 19.2 Gb/s Data Link for High-Performance Computing." In 67th IEEE Electronic Components and Technology Conference (ECTC), 2017