HIGH-SPEED CHANNEL ANALYSIS AND DESIGN USING POLYNOMIAL CHAOS THEORY AND MACHINE LEARNING

A Dissertation
Presented to
The Academic Faculty

by

Majid Ahadi Dolatsara

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in the
School of Electrical and Computer Engineering

Georgia Institute of Technology
May 2021

HIGH-SPEED CHANNEL ANALYSIS AND DESIGN USING POLYNOMIAL CHAOS THEORY MACHINE LEARNING

Approved by:

Dr. Madhavan Swaminathan, Advisor
School of Electrical and Computer Engineering
Georgia Institute of Technology

Dr. Arijit Raychowdhury
School of Electrical and Computer Engineering
Georgia Institute of Technology

Dr. Sung-Kyu Lim
School of Electrical and Computer Engineering
Georgia Institute of Technology

Dr. Suresh K Sitaraman
School of Mechanical Engineering
Georgia Institute of Technology

Dr. Saibal Mukhopadhyay
School of Electrical and Computer Engineering
Georgia Institute of Technology

Dr. Duen Horng Chau
School of Computational Science and Engineering
Georgia Institute of Technology

Date Approved: 01/27/2021

This work is dedicated to my parents for their endless love and support.

ACKNOWLEDGEMENTS

I would like to express my gratitude to my advisor, Dr. Madhavan Swaminathan, and my PhD committee members, Dr. Sung-Kyu Lim, Dr. Saibal Mukhopadhyay, Dr. Arijit Raychowdhury, Dr. Suresh K Sitaraman, and Dr. Duen Horng Chau. In addition, I thank my collaborators and coauthors, including but now limited to Dr. Jose Ale Hejase and Dr. Wiren Dale Becker from IBM, since this work would not be possible without their collaboration. Finally, I appreciate the support of my family, friends, and my colleagues at the 3D Systems Packaging Research Center (PRC) at the Georgia Institute of Technology and the NSF IUCRC center for Advanced Electronics through Machine Learning (CAEML).

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS AND ABBREVIATIONS

LTI	linear time invariant
BER	bit error rate
BO	Bayesian optimization
CDF	cumulative distribution function
CTLE	continuous time linear equalizer
DCD	duty cycle distortion
DDJ	data dependent jitter
DDN	data dependent noise
DFE	decision feedback equalizer
EH	eye height
EI	expected improvement
EW	eye width
FFE	feed forward equalizer
FIR	finite impulse response
GP	Gaussian process
HPC	hyperbolic Polynomial Chaos
i.i.d	independently and identically distributed
INN	invertible neural network
ISI	intersymbol interference
LCP	liquid crystal polymer
LGA	land grid array
LIB	least important bit

MC	Monte Carlo
MIB	most important bit
ML	machine learning
MLP	multilayer perceptron
MMD	maximum mean discrepancy
NN	neural network
NRZ	non-return-to-zero
P	probability
PC	Polynomial Chaos
PDA	peak distortion analysis
PDF	probability distribution function
PI	probability of improvement
RJ	random jitter
RN	random noise
s	Frequency parameter
t	time
UCB	upper confidence bound
UI	unit interval
λ	vector of random variables
μ	mean
σ	standard deviation
ϕ_i	*i*-th polynomial function

SUMMARY

With the exponential increase in the data rate of high-speed serial channels, their efficient and accurate analysis and design has become of crucial importance. Signal integrity analysis of these channels is often done with the eye diagram analysis, which demonstrates jitter and noise of the channel. Conventional methods for this type of analysis are either exorbitantly time and memory consuming, or only applicable to linear time invariant (LTI) systems. On the other hand, recently advancements in numerical methods and machine learning has shown a great potential for analysis and design of high-speed electronics. Therefore, in this dissertation we introduce two novel approaches for efficient eye analysis, based on machine learning and numerical techniques. These methods are focused on the data dependent jitter and noise, and the intersymbol interference. In the first approach, a complete surrogate model of the channel is trained using a short transient simulation. This model is based on the Polynomial Chaos theory. It can directly and quickly provide distribution of the jitter and other statistics of the eye diagram. In addition, it provides an estimation of the full eye diagram. The second analysis method is for faster analysis when we are interested in finding the worst-case eye width, eye height, and inner eye opening, which would be achieved by the conventional eye analysis if its transient simulation is continued for an arbitrary amount of time. The proposed approach quickly finds the data patterns resulting in the worst signal integrity; hence, in the closest eye. This method is based on the Bayesian optimization. Although majority of the contributions of this dissertation are on the analysis part, for the sake of completeness the final portion of this work is dedicated to design of high-speed channels with machine learning since the interference and complex interactions in modern channels has made their design challenging and time consuming too. The proposed design approach focuses on inverse design of CTLE, where the desired eye height and eye width are given, and the algorithm finds the corresponding peaking and DC gain of CTLE. This approach is based on the invertible neural networks. Main advantage of this network is the possibility to provide multiple solutions for cases where the answer to the inverse problem is not unique. Numerical examples are provided to evaluate efficiency and accuracy of the proposed approaches. The results show up to 11.5X speedup for direct estimation of the jitter distribution using the PC surrogate model approach. In addition, up to 23X speedup using the worst-case eye analysis approach is achieved, and the inverse design of CTLE shows promising results.

CHAPTER 1. INTRODUCTION

1.1 Objectives

The problems under consideration in this dissertation are discussed in this chapter. Subsequently, the developed solutions are introduced, which includes two novel eye analysis methods and an inverse design approach. Next, the prior art is discussed, which is followed by the main challenges and contributions of this dissertation. The chapter ends with a section on organization of this document.

1.2 Description of the Problem

In the past decades the bandwidth of high-speed channels has increased exponentially. As frequency of systems increase, it becomes more challenging to avoid undesired deviations in the transmitted signal due to amplitude noise and timing jitter. Figure 1 shows an example of such deviations. In this figure, the noise shown as Δv causes a failure in data transmission if it pushes the signal beyond the threshold voltage of the receiver. Besides, the jitter shown as Δt, causes an error if it moves the rising or falling edge over the sampling point τ. Error caused by jitter and noise is often measured by bit error rate (BER). Additionally, the eye diagram, bathtub curve, and probability distribution functions (PDFs) are used to evaluate quality of the signal [1]. In this section we focus on jitter; however, a parallel discussion can be applied to noise as well. To study jitter, its various causes need to be considered. In general, jitter sources are divided into intrinsic

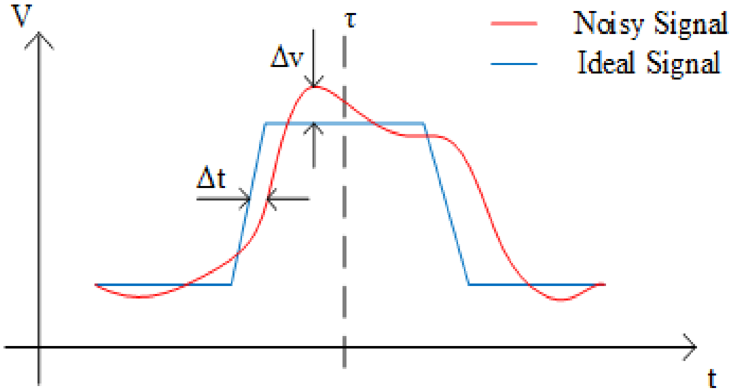


Figure 1 – Ideal and noisy signal, showing voltage noise Δv, jitter Δt, and sampling point τ.

and non-intrinsic. The intrinsic jitter sources are based on physical properties of the materials, and the non-intrinsic sources are related to the design parameters. The latter includes duty cycle distortion (DCD), crosstalk and intersymbol interference (ISI). Amongst them, ISI is a result of memory and capacitive effects in circuits, which is challenging to model. On the other hand, jitter can be divided into random and deterministic types. Random jitter is mainly caused by intrinsic sources, which can be added to the signal in post-simulation with some considerations. The root cause of deterministic jitter is effects such as reflection, crosstalk, electromagnetic interference, etc. In other words, it is generated by design of the channel. Furthermore, deterministic jitter is divided into periodic jitter, bounded uncorrelated jitter, and data dependent jitter (DDJ). DDJ depends on the data pattern, and it is caused by DCD and ISI. Different jitter types are demonstrated in Figure 2 [1].

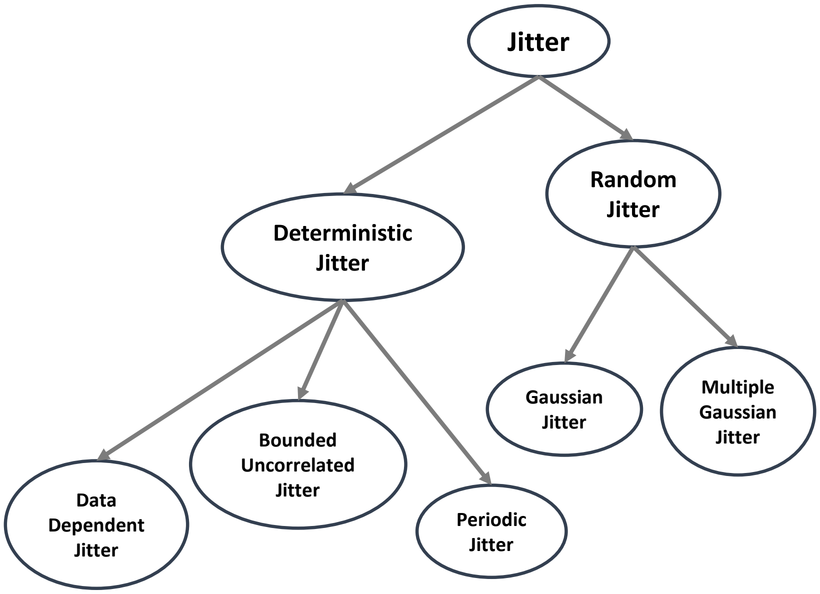


Figure 2 – Classification of jitter types.

Current simulation methods are not always accurate in capturing the ISI and DDJ; thus, in the analysis part of this work we focus on estimating this type of jitter. The proposed approaches can be combined with other estimation methods to efficiently predict BER. Jitter is often evaluated by deriving an eye diagram. Traditionally, it is done with a lengthy transient simulation (transient eye). Although it is straightforward, generating the transient eye requires high computational time and memory storage. For instance, if the BER is 10^{-12}, it is expected that data transmission of 10^{12} bits be simulated to observe one bit failure. Such a long simulation is not practical for complex circuits. Hence, methods based on extrapolation and approximation from a shorter transient simulation (e.g., 10^{10} bits) are used, which reduce the accuracy. Furthermore, statistical methods have been developed to quickly estimate the eye diagram [2], [3]; however, these methods are only applicable to LTI systems. To address these issues, we suggest two new efficient eye analysis methods, which are introduced in 1.3 and 1.4, respectively.

The final portion of this dissertation is dedicated to inverse design of high-speed channels. Since data rate of the channels has increased exponentially, their design has become more challenging. Designers need to set many design parameters and consider several constraints to satisfy the performance criteria, including a low BER. One of the critical tasks in this process is design of the equalization, which is often done with feed forward equalizer (FFE), decision feedback equalizer (DFE), and continuous time linear equalizer (CTLE). FFE and DFE coefficients are often calculated adaptively. Although the frequency response of CTLE can be found theoretically, in reality it does not always provide a satisfactory result. Therefore, engineers depend on simulating the channel with all practical CTLE settings. Unfortunately, this method can be very time consuming. It is possible to utilize human tuning or optimization methods; however, the best solution may still not be achieved, or several satisfactory possibilities could be ignored due to the nature of these algorithms. To address these issues, in 1.5 we introduce an inverse design approach to find the CTLE settings, which includes the peaking and DC gain value of CTLE's frequency response.

1.3 Jitter and Eye Estimation with Polynomial Chaos Surrogate Models

The first analysis approach is partially based on uncertainty quantification, which can be defined as the study of propagation of random variables in a system [4]. This method can be applied to estimation of the data dependent jitter and noise which are a function of random input pulses. The most well-known uncertainty quantification method is Monte Carlo (MC) analysis. MC can be prohibitively time consuming; hence, more efficient methods have been developed [4], [5], [6], [7], [8], [9]. A popular concept used in such methods is the generalized Polynomial Chaos (PC) theory [10], where random variables are approximated as sum of an expansion of orthogonal polynomials. In the proposed approach, based on PC theory and uncertainty quantification, we propose a methodology to predict jitter, eye diagram, and statistics of high-speed channels' output. This is done by training a surrogate model of the system using a transient simulation, which is significantly shorter than the simulation required for the transient eye analysis. In addition, this approach is applicable to non-LTI systems.

1.4 Worst-case Eye Analysis with Bayesian Optimization

Although the PC surrogate models reduce the computational costs of eye analysis and simplify the statistical equations, building the surrogate models can introduce some inaccuracy as a result of the simplifications. In addition, it can still be computationally expensive due to the overhead cost of training the models. Therefore, development of more advanced methods is appreciated.

To further alleviate the computational costs and provide a higher accuracy, we propose a second approach, named Worst-eye. In this method, the bit patterns resulting in the worst-case signal integrity are identified with an optimization algorithm. Then, the eye

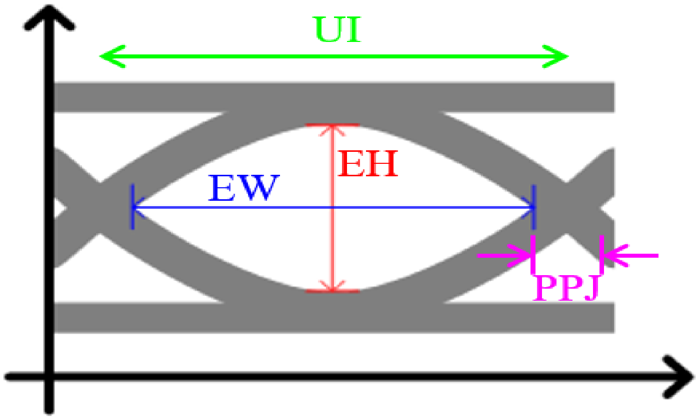


Figure 3 – Eye height (EH), eye width (EW), peak to peak jitter (PPJ), and unit interval (UI) shown on an eye diagram.

height, eye width, and the inner eye opening (worst-case eye) are found from the resulting signals. In this method, initially the search space is mapped to a low dimensional space using the reflected binary code (Gray code) [11] to reduce the dimensionality and sparsity. Then, after preparations based on the domain knowledge, Bayesian optimization (BO) [12] is applied to find the bit patterns resulting in the worst-case waveforms which pass through the points defining eye height and eye width as shown in Figure 3. 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. Furthermore, this approach is extended to find the worst-case bit patterns of crosstalk resulting in the worst-case eye. Note that the advantage of the PC surrogate modeling approach over the worst-case eye method is providing the full eye diagram, statistics of the signal, and the jitter distribution, while the worst-case eye only reports the eye height, eye width, and the inner eye opening. On the other hand, the worst-case eye method is faster, and since it derives the results from actual transient simulations it can provide more accurate results.

It is worth noting that, BO-based methods have been previously used in optimization of electronic designs [13], [14], [15]. In [13], BO is used to minimize the clock skew rate in 3D systems by co-optimization of thermal and electrical performance. The input parameters in this paper are geometrical and physical characteristics of the materials used in the 3D system such as thermal conductivity of the PCB, and thermal conductivity and thickness of the thermal interface material. In [13] the BO algorithm finds the optimal input values that result in the minimum clock skew rate. However, in the proposed approach 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, [14] and [15] suggest variations of the BO algorithm to optimize design characteristics by finding optimal physical parameters, similar to [13]. Moreover, [16] suggests 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. Novelty of the proposed approach is in finding the data patterns and waveforms causing the worst-case eye characteristics. In other words, the goal in this approach is to reduce the computational costs of finding the worst-case eye diagram for a set of fixed physical parameters.

1.5 Inverse Design of CTLE with Invertible Neural Networks

In the traditional design and modeling process, from a combination of design parameters (inputs) the output of the system is found. In contrast, in the inverse approach we start from the output objectives and derive the corresponding input parameters that satisfy these objectives. The inverse problem has been a popular concept for decades. However, recently with the advancements in machine learning (ML), there has been several attempts for inverse design of high-speed electronics by exploiting the ML techniques [17], [18], [19], [20], [21], [22]. From these techniques, invertible neural networks (INNs) [23] have shown a great potential. A main advantage of these networks is providing distribution of the design parameters instead of deterministic values. This advantage can be used to deal with the non-uniqueness of the solution issue, which can be a major problem in inverse design with traditional approaches. Using INNs, we can derive several possible combinations of the design parameters instead of one. Then a satisfactory design can be selected based on other constraints. Previously INNs have been used for design of SIW filters in D-band [22]. In this work, we investigate to see if a similar approach can be used to derive the peaking and DC gain of CTLE from the desired eye height and eye width. This is a challenging problem because the considered CTLE settings are discrete. In addition, the outputs have a nonlinear relationship with the design parameters.

1.6 Prior Art

In this section various existing methods for the problems considered in this dissertation, namely full eye analysis, worst-case eye analysis, and inverse design of electronics, are reviewed. In addition, shortcomings of these methods and the reason to develop a new approach is discussed.

1.6.1 Transient Eye

Transient eye is the conventional method to derive the eye diagram [1]. In this method, first a lengthy transient simulation is performed. Next, the received signal is divided to sections with length of one unit interval (UI), which are then superimposed. The resulting diagram has an eye shape; hence the name. A smaller eye opening implies higher jitter and noise, while a larger eye opening shows better quality of the signal. As discussed before, this method can be prohibitively time and memory consuming; therefore, various techniques are developed to increase the efficiency of eye analysis.

1.6.2 Statistical Eye Estimation Methods

To address limitations of the transient eye analysis, statistical methods have been developed. One of these methods is peak distortion analysis (PDA), which is used to find the worst-case eye opening of an LTI system [2], [24]. Initially in this method, the response to a single pulse is determined. The pulse response can be longer than one unit interval (UI), and it can overlap with response of another symbol, causing ISI. Therefore, the response of a sequence of pulses is found by superposition of shifted pulse responses, which is applicable if the system is LTI. The lowest logical one and the highest logical zero are calculated at the sampling point, and their difference is reported as the worst-case vertical eye opening. The worst-case eye shape can also be found by repeating this analysis at multiple sampling points and superimposing the results [25].

To extend the worst-case eye analysis, the StatEye method has been developed [3]. This method suggests that the same superposition technique can be used to find the response to any combination of input pulses. Based on this idea, a statistical method has been developed to find the distribution of receiver voltages. This is done statistically to avoid substantial CPU run times and high memory requirements. The StatEye method considers probability of different pulse combinations, and finds the distribution of ISI at an arbitrary sampling point. By connecting different sampling points, the statistical eye is formed. Furthermore, BER is calculated as the ratio of area in the jitter's/noise's PDF that crosses the time/voltage threshold. Finally, bathtub curve is determined by drawing the cumulative distribution function (CDF) of BER over one UI [1].

Although efficient, both PDA and Stat-Eye are only applicable to LTI systems since the result is based on linear superposition of the pulse responses. However, non-LTI components, such as I/O drivers and receivers, can be present in high-speed channels and can contribute to jitter and noise. Notably, this is observed in single-ended signaling, such as DDR signaling in memory systems. In these systems rising and falling edge responses can be asymmetric, which shows that the system is non-LTI. Recent work has tried to extend statistical methods to non-LTI cases. In [26], it has been suggested to estimate the receiver voltage using the superposition of responses of a rising edge and a falling edge.

This method improves the result since the difference between the rising and falling edge is considered. However, calculating the statistical eye is challenging since edge responses are not independent of each other. Therefore, [26] develops an inductive technique to find the distribution of receiver voltage, which is derived from steady state responses step by step. However, using this approach, improvements in accuracy of non-LTI systems is not always sufficient. Therefore, [25] expands this method by using various edge responses, which are different in the bits prior to the rising or falling edge. For instance, if M bits are considered, 2^M responses are generated. Nevertheless, it is hard to generalize the edge response-based methods since their accuracy depends on number of considered responses, which increases exponentially with M. Moreover, they are an extension of the superposition method; thus, they fail for more complicated non-LTI systems.

1.6.3 Analytical Eye Diagram Estimation

If the analytical formula describing a system is known, it would be easy to estimate the eye diagram since such functions are often evaluated quickly for arbitrary inputs. In addition, if the derived transfer function is LTI, statistical methods can be applied. Therefore, there has been several attempts to develop mathematical models to describe high-speed channels [27], [28], [29], [30], [31], [32], [33]. In [27], the voltage transfer function of a channel with a single pair of differential microstrip lines on PCB is obtained from the ABCD parameters of its components. Then, a modified PDA method is proposed, that utilizes multiple pulse responses and the transfer function to address the asymmetrical rise/ fall times. In [28], based on RLGC models, Telegrapher’s equations, and a finite impulse response (FIR) equalization, a deterministic formula for crosstalk and ISI is derived. This formula describes noise and jitter, and it is used to find the bit patterns that cause worst-case noise and jitter, by using a binary search algorithm. Furthermore, assuming a linear channel, [29] does an extensive analysis to develop closed form equations for eye height and eye width from ISI and DDJ, respectively. This paper takes into account the effects of reflection, attenuation, shunt capacitance, and passive equalization.

Unfortunately, analytical methods cannot be generalized since they use mathematical formulations that are limited to the considered systems. In addition, simplifications and assumptions are often made to make the derivation of analytical equations possible. Finally, designers need to have expert knowledge on nature of the system to derive these equations. Therefore, development of more general approaches is needed.

1.6.4 Using Surrogate Models for Eye Estimation

There are many candidates in machine learning for developing surrogate models. Using neural networks (NNs) has been previously suggested to generate surrogate models of nonlinear circuits. In [34] and [35], recurrent NNs is used to model nonlinear I/O drivers, and in [36] recurrent NNs is used to model SerDes channels. This is similar to our proposed approach where PC theory is exploited to develop surrogate model of the system. However, training NNS is generally more complicated and time consuming compared to PC models.

Moreover, NNs can accumulate error over time, and can become significantly large for channels with large memory. In addition, PC provides statistical moments of the output as a byproduct of the training process with no additional costs, while numerous evaluations using the NN model are necessary to determine the statistics.

1.6.5 Eye Analysis with Simulating All Possible Waveforms

As our computing resources increases, it becomes more enticing to simulate all possible combinations of the previous bits to obtain the eye diagram. In fact, in transient eye analysis we hope to achieve all the combinations in our random sequence. In [37] and [38], authors have developed methods that take advantage of simulating all possible ISI combinations. One advantage of this method is that it can confidently report the worst-case eye. Additionally, it handles non-linear systems because it is based on transient simulation. However, this method is reasonable when number of symbols with effective ISI is small since 2^n simulations are needed for *n* symbols. Moreover, unlike transient eye, simulations are done separately. The overhead cost of calling the simulation software 2^n times, and waiting for the channel to achieve steady state before simulating the intended sequence, can be prohibitive. In, fact it might be faster to run the transient eye simulation.

1.6.6 Worst-case Eye Diagram with Evolution Based Optimization

To improve the previous approach, one can apply evolution-based optimization algorithms, where each transient simulation is a sample, to find the worst bit pattern which corresponds to the worst-case eye. [39] and [40] use the genetic algorithm and particle swarm optimization, respectively, to find the worst bit pattern. However, evolution-based optimization methods are proved to have a low convergence rate. Besides, as the number of bits with effective ISI increases it would be more challenging to run enough simulations for these algorithms due to curse of dimensionality.

1.6.7 Worst-case Eye Diagram with Random Tree Based Optimization

A random tree optimization approach to find the eye diagram of high-speed CMOS circuits is suggested in [41]. However, this method only considers a very short sequence of deterministic input symbols. Then, it introduces random processes as perturbations on timing of signals, and voltages of different nodes of the circuit. The perturbation parameters are simply modeled as truncated Gaussian random processes. Finally, this method finds impact of the introduced random processes on the output eye diagram. Applying this method on high-speed channels can be problematic because considering a very short input bit sequence results in missing certain ISI patterns. Moreover, assuming Gaussian distributions for voltage and timing variables on input of the receiver is oversimplification and can result in inaccurate results.

1.6.8 Inverse Design with Neural Networks

Neural networks have been used for the inverse design of electromagnetic systems in several works in the past decades. In the following some of these papers are discussed.

In [42], an inverse problem approach is suggested, which is used to reconstruct the permittivity and conductivity of a medium from its measured scattered electromagnetic fields. This approach trains a Hopfield neural network, which is connected to a linear programming network, to find the solution of inverse problem.

In [43], a simple multilayer perceptron (MLP), with one hidden layer, is trained in the forward direction to model the relationship of the geometrical variables and electromagnetic fields in sample test cases. Although this network is not inherently invertible, [43] tackles the inverse problem by defining an error threshold for the objective functions and iteratively updating the outputs and inputs of the hidden layer, and subsequently the input design parameters, until achieving a viable solution.

An inverse design approach for determination of the geometrical variables of microstrip bandpass filters from the desired filter response is suggested in [17]. In this method, two neural networks, each of them including only one hidden layer, are trained for the forward and inverse mapping of the design parameters and the output objective functions. The inverse network receives the target response and yields an initial guess of the design parameters. Then, the forward mapping network along with an optimization algorithm is used iteratively to tune these variables in order to achieve the desired response.

A similar approach is developed in [18] for inverse design of the Raman amplifier by deriving the design parameters from the desired gain values. In this method, multi-layer NNs are used for inverse and forward mapping. Furthermore, approximation result of the inverse network is fine-adjusted using the forward network and the gradient descent algorithm.

It is observed that the methods described above make an approximation of the target variables using relatively simple NNs, which is then fine-tuned with different methods. Although this approach can be successful for relatively simple cases, it does not provide a general solution for complicated systems where multiple designs can result in the same response. Hence, development of more advanced inverse design methods is necessary.

Larger NNs have been used in [44] for inverse design of plasmonic nanostructures, where the required nanostructure geometry is found from the target optical response spectrum. The idea behind this method is similar to [17] and [18]; however, in addition to the difference in the size of NNs, [44] suggests simultaneous training of the cascaded inverse and forward mapping networks instead of using an optimization algorithm for tuning the inverse response. Note that, larger networks require more training samples (the example in [44] takes 15000 samples), and it still does not provide a general solution for the inverse problems.

Finally, we like to mention the application of generative adversarial networks (GANs) for inverse design of electronics. GAN is a neural network comprised of a generative network and a discriminative network. The generative network seeks to generate samples similar to the training data, while the discriminative network is tasked to distinguish the original samples and the ones fabricated by the generative model. The competition between the two result in generation of realistic samples by GAN which have never been seen before [45]. In [46] authors have exploited this structure for inverse design of metasurface structures. This method receives the desired optical spectra and generates the candidate patterns matching that spectra. Nevertheless, GANs are not an ideal candidate for inverse design because of the possible instability in their training process. In addition, they cannot systematically solve the non-uniqueness problem because the model needs to estimate the multi-modal distribution of the design parameters in such problems, which can be problematic for GANs [22].

1.7 Challenges and Contributions

As mentioned earlier, the rapid increase in bandwidth of high-speed channels and the decrease in the required BER has made channels' analysis and design more complicated and time consuming. In this section, the challenges in this area that are addressed by the proposed methods and the main contributions of this dissertation are discussed.

Traditional eye analysis of high-speed channels can be memory and time consuming. Fast statistical approaches exist; however, they are generally only applicable to LTI systems. Our first analysis method alleviates these issues by introducing a quick surrogate modeling approach, based on the Polynomial Chaos theory, for DDJ, DDN, and ISI analysis. We have focused on this type of jitter and noise because it is challenging to estimate them by traditional methods. This approach is trained with a much shorter transient simulation, and it is applicable to LTI systems. Additionally, since the surrogate model is comprised of orthogonal polynomials, it simplifies the statistical equations of the received signal, which will be further explained in CHAPTER 2. After further examination it was observed that size of the surrogate model increases near exponentially with delay of the channel. This is because a channel with a larger delay needs to consider a higher number of previous bits for determining ISI, which increases the size of the surrogate model. Therefore, we use a modified PC expansion, called the hyperbolic Polynomial Chaos (HPC) expansion, and develop a smaller model for channels with large delays. The modified model significantly reduces computational costs with minimal decline of accuracy. This work provides the instructions to choose between the PC and HPC models to balance accuracy and efficiency. Other considerations for channels with large delays are provided as well. Furthermore, we have suggested use of the ridge regression for training the proposed surrogate models to provide more stable results, which to our knowledge has not been used before for PC surrogate models. Another contribution of this work is providing a direct jitter model to estimate distribution and statistics of the jitter. This model is much faster than finding the complete eye diagram, and it provides a comprehensive evaluation of jitter. Moreover, the computational cost analysis of this approach is provided, and its performance is evaluated on complex examples.

The second approach provides a higher speedup for cases where we are primarily interested in the eye height, eye width, and the inner eye opening (worst-case eye) caused by DDJ and DDN. For this task, the proposed approach finds the bit patterns causing the worst ISI signals by searching the random space of several previous bits. To find such bit patterns, we have taken advantage of the Bayesian optimization. The proposed approach is advantageous compared to earlier works that search for the worst-case eye opening by means of the evolutionary algorithms [39], [40] 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 [13], where the genetic algorithm needed more than 26 times samples compared to BO for optimization of a complex function. Nevertheless, finding the target bit patterns is still a challenging problem because the objective functions are nonconvex and nonlinear. In addition, the search space is high-dimensional and sparse. Therefore, we have developed a mapping algorithm based on the domain knowledge and the Gray code to simplify the optimization. Another contribution of this work is incorporating the crosstalk in this problem and predicting its effect on the worst-case eye. Furthermore, an iterative algorithm is suggested to ensure accuracy of the worst-case eye by repeating the proposed approach at different sampling points. Finally, numerical examples along with their computational costs and convergence rates are provided to show performance of the proposed approach.

On the design side, we have proposed an inverse design approach to configure the CTLE. We have focused on CTLE since its design can be complicated and time consuming and require sweeping over all the possible designs. In inverse design, design parameters are derived from the desired outputs, which is moving in the opposite direction of the traditional forward design approach. The main challenge in inverse design is the possibility of existence of multiple designs that result in the same output, which is called non-uniqueness of the solution. In these cases, a regular regression approach would be incapable of finding a solution since the system would be ill-posed. To address this issue, we have used the invertible neural network. This network provides distribution of the input variables based on the desired outputs. Therefore, it can facilitate prediction of multiple solutions for a single objective. This approach is advantageous compared to forward optimization techniques since it does not become trapped in local minima. In addition, providing multiple solutions is advantageous for the designer since it provides the flexibility to choose from the possible solutions based on other design constraints. Nevertheless, the CTLE design is not a straightforward problem, and the proposed approach needs to address other challenges including the nonlinear relationship between the CTLE parameters and the output signal. Additionally, in the past INN has been studied for continuous input variables, while the CTLE parameters are discrete. In this work, we evaluate performance of the INN for such variables with a numerical example.

1.8 Organization of This Dissertation

In this dissertation two novel eye analysis methods, and an inverse design approach for CTLE are proposed. Each approach is preceded by a review of the main required knowledge and techniques for that approach. Organization of the rest of this dissertation is as follows. In CHAPTER 2, the PC theory and its training process is reviewed, which is followed by the proposed PC surrogate modeling approach, its computational cost analysis, and numerical examples. Next, the Gray Code and Bayesian optimization are reviewed in CHAPTER 3. Then, the proposed worst-case eye analysis approach with examples is presented in the same chapter. INN is discussed at the beginning of CHAPTER 4. Afterwards, the basics of CTLE is reviewed in this chapter. Then, the proposed CTLE inverse design approach is introduced and its performance is evaluated on an example. Finally, the dissertation is concluded in CHAPTER 5, which includes a discussion on the proposed methods, possible future works, and list of the publications resulted from this work.

1.9 Summary

This chapter covered an introduction on the problems addressed in this dissertation, which includes improvements to efficiency of the analysis and design of high-speed channels using numerical and machine learning approaches. The proposed approaches are based on Polynomial Chaos theory, Bayesian optimization, and invertible neural networks. Additionally, several state-of-the-art methods in this area and their pitfalls were reviewed. Challenges and contributions of this work were discussed, and the chapter ended with describing the organization of this document.

CHAPTER 2. POLYNOMIAL CHAOS SURROGATE MODELING APPROACH FOR HIGH-SPEED CHANNELS

2.1 Objectives

In this chapter our first efficient eye diagram analysis approach is presented. This approach develops a surrogate model for the high-speed channel to quickly perform the eye analysis. In addition, it directly provides the jitter distribution and other statistics of the eye diagram with significantly less computational costs. The surrogate model is based on the Polynomial Chaos (PC) theory. Therefore, the PC theory is discussed at the beginning of this chapter. Afterwards, the proposed approach is discussed in detail. Finally, the chapter ends with numerical examples, demonstrating the performance of this approach.

2.2 Polynomial Chaos Theory

In this section, PC surrogate models, and a training method are discussed. Generalized PC theory suggests approximating a smooth function of random variables as:

$$f(\boldsymbol{\lambda}) \approx \sum_{i=0}^{P} c_i \phi_i(\boldsymbol{\lambda}) \tag{1}$$

where $\boldsymbol{\lambda} = [\lambda_1, \lambda_2, \ldots, \lambda_n]$ represents a vector of random variables, c_i are unknown coefficients, P is truncated length of the expansion, and $\phi_i(\boldsymbol{\lambda})$ are multidimensional polynomials, which are orthonormal with respect to the distribution of $\boldsymbol{\lambda}$ [4], [10]. Using the orthogonality condition, mean and variance are found analytically:

$$E\big(f(\boldsymbol{\lambda})\big) \approx c_0 \tag{2}$$

$$var\big(f(\boldsymbol{\lambda})\big) = \sigma^2 = \sum_{i=1,2,\ldots} c_i^2, \tag{3}$$

where σ is the standard deviation. It is worth noting, that mean and variance are found directly from coefficients of the PC expansion at no extra cost. When analyzing jitter, we use (3) to find Jitter RMS value since it has zero mean. Finally, to determine the PDF from the PC expansion, an approach similar to MC analysis is taken, by evaluating the PC expansion of (1) at numerous sample points, which is relatively quick since it is done analytically and without the need for circuit simulations. Furthermore, the multidimensional polynomials, $\phi_i(\boldsymbol{\lambda})$, are found as:

$$\phi_i(\boldsymbol{\lambda}) = \prod_{j=1}^{n} \phi_{\alpha_j}(\lambda_j), \tag{4}$$

where $\phi_{\alpha_j}(\lambda_j)$ is a 1-D polynomial at the *j*-th dimension, and $\boldsymbol{\alpha} = [\alpha_1, \alpha_2, \ldots, \alpha_n]$ shows index of selected 1-D polynomials. Traditionally, α_j are selected with a linear constraint:

$$\|\boldsymbol{\alpha}\|_1 = \alpha_1 + \alpha_2 + \cdots + \alpha_n \leq m. \tag{5}$$

Number of terms selected in (5) determines length of the expansion in (1):

$$P + 1 = \binom{m+n}{m} = \frac{(m+n)!}{m!\,n!}, \tag{6}$$

where *m* shows the maximum order of each polynomial term in the expansion. For smooth functions, good approximation is achieved with *m* set to 2 or 3. Unfortunately, *P* scales near exponentially with *n*, and size of the PC expansion becomes prohibitive when *n* is greater than 20. For the problems addressed in this work, expansions with *n* as large as 50 are required; hence, development of a modified approach is necessary. We propose using the hyperbolic Polynomial Chaos (HPC) expansion [9], when it is necessary. HPC uses, the following constraint instead of (5), to select the multidimensional polynomial basis:

$$\|\boldsymbol{\alpha}\|_u = \sqrt[u]{\sum_{j=1}^{n} \alpha_j^u} \leq m. \tag{7}$$

Here the first norm of $\boldsymbol{\alpha}$ is replaced by its *u*-norm, where $u < 1$ to provide a concise selection of the basis. This process drastically reduces length of the expansion in (1).

The next step is finding the c_i coefficients in (1), which alternatively can be called training the model. For this purpose, a non-intrusive linear regression method can be used [6], [7]. In this method, *N* random training samples are generated from distribution of $\boldsymbol{\lambda}$, where $N =$

$k(P+1)$, and $k \geq 2$. We show these samples as $\boldsymbol{\lambda}^1$ to $\boldsymbol{\lambda}^N$. By placing them in the PC expansion of (1) and writing them in matrix form, we have:

$$\boldsymbol{A}\boldsymbol{\Gamma} = \boldsymbol{E}, \tag{8}$$

$$\boldsymbol{A} = \begin{bmatrix} \phi_0(\boldsymbol{\lambda}^1) & \dots & \phi_P(\boldsymbol{\lambda}^1) \\ \vdots & \ddots & \vdots \\ \phi_0(\boldsymbol{\lambda}^N) & \dots & \phi_P(\boldsymbol{\lambda}^N) \end{bmatrix}, \boldsymbol{\Gamma} = \begin{bmatrix} c_0 \\ \vdots \\ c_P \end{bmatrix}, \mathbf{E} = \begin{bmatrix} f(\boldsymbol{\lambda}^1) \\ \vdots \\ f(\boldsymbol{\lambda}^N) \end{bmatrix}. \tag{9}$$

Next, the solution is found by using the least squares method. Although this approach minimizes the error for training data, it does not always provide the best result for testing data. In other words, the solution does not always generalize well. Therefore, we suggest improving the model by using Ridge Regression, which is a regularized regression method [47]. This approach suggests adding a regularization term to the least squares method:

$$\hat{\boldsymbol{\Gamma}} = \underset{\boldsymbol{\Gamma}}{Argmin} \|\boldsymbol{E} - \boldsymbol{A}\boldsymbol{\Gamma}\|_2^2 + \|\mathbf{B}\boldsymbol{\Gamma}\|_2^2, \tag{10}$$

with **B** being equal to $\sqrt{\beta}\mathbf{I}$, except that $\mathbf{B}(0,0)$ is kept at zero. Moreover, **I** is the unity matrix, and β is a constant that determines the weight of regularization. With optimization, it can be shown that (10) is minimized when:

$$\hat{\boldsymbol{\Gamma}} = (\boldsymbol{A}^\tau \boldsymbol{A} + \boldsymbol{B}^\tau \boldsymbol{B})^{-1} \boldsymbol{A}^\tau \boldsymbol{E}. \tag{11}$$

2.3 Modified PC Modeling Approach

Jitter and noise parameters in the eye diagram have stochastic characteristics, which enables using uncertainty quantification approaches to calculate them. In fact, this work is inspired by the similarities between the transient eye [48] and the MC [49] analysis. In the MC approach, random variables in the system are sampled at numerous points. Then, the system is simulated at the samples, and the results are used to estimate distribution of output random variables. Similarly, in the transient eye, a long physical simulation is performed to calculate the uncertainty of output. This simulation is the response to a long sequence of random pulses, and it is equivalent to finding the output at various sample points in the MC analysis. In this case the outputs are noise and jitter. Both methods are straightforward and accurate if given enough time to complete; however, their computation costs can be prohibitive for complex systems. On the other hand, modern uncertainty quantification methods, such as PC theory, are substantially more efficient than MC. The idea behind this approach is to use PC to develop surrogate models for estimating jitter and eye diagram, as an efficient alternative to the transient eye.

The simplified process is illustrated in Figure 4, where random input patterns, and a falling edge in the output are shown. Intuitively, it can be said that different previous bits cause different jitter. Figure 4 (a) shows the response to a falling edge after a sample signal. However, the response can be shifted from its original place by changing the previous bits, as shown in Figure 4 (b). Moreover, Figure 4 (c) shows all possible combinations of previous bits and the corresponding outputs. Both previous bits and jitter are random and lead to the memory effect described in Figure 4. The goal in this work is therefore to find

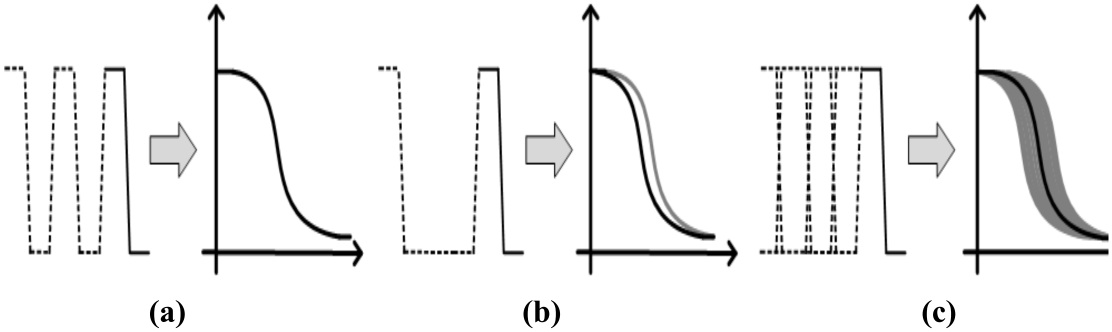


Figure 4 – Simplified input and output signals of a channel for illustrating cause of data dependent jitter in a falling edge. a) A sample signal. b) Small change in previous bits of (a). c) All possible combinations of previous bits.

the relationship between them, and determine how the randomness propagates from data to jitter. Unlike statistical methods, in this approach the system does not have to be LTI.

As mentioned earlier, PC methods are suitable for smooth variations in the output. However, the output signal of a channel can show a wide variation, between logical zero and one, as shown in Figure 5. Therefore, directly applying the existing PC approaches to this problem can yield inaccurate results. To address this issue, we suggest breaking the problem to four smaller cases and developing four sub-models. Figure 5 (a) shows a typical eye diagram of an NRZ signal, where at an arbitrary point on the X-axis, shown with a blue line, four possible transitions can be seen. The transitions are zero to zero, zero to one, one to zero, and one to one. The response to each transition is disturbed from its average value by variations in the previous bits. Moreover, Figure 5 (b) shows distribution of the signal at the blue line in Figure 5 (a). Each Gaussian curve in this figure shows distribution of one of the possible transitions. It is observed that, by modeling each transition separately the

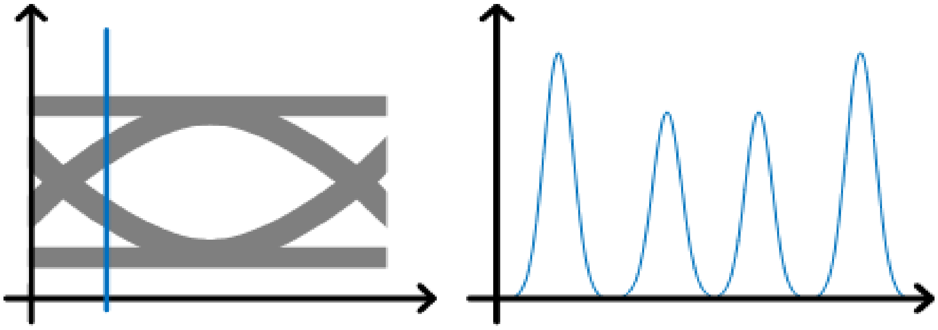

(a) (b)

Figure 5 – A typical eye diagram with distributions of possible transitions at a sample time point. a) Eye diagram. b) Distribution of transitions.

variation of output is significantly limited, which makes the PC methods applicable. Similarly, this approach can be extended to other coding schemes such as PAM4.

In this approach, inputs are the previous bits that cause effective ISI, while the last two bits show the transition. These two bits are used to divide the training and testing data into four categories, and show which sub-model needs to be used for each category. The remainder of the previous bits disturbs output of each transition from its average value, and they are therefore inputs to the four sub-models. In the remainder of this chapter, we label these bits as $\boldsymbol{\lambda} = [\lambda_1, \lambda_2, \dots, \lambda_{n-2}]$. Furthermore, we define two types of outputs for two different ways of evaluating the degraded signal. The first method is a quick and direct method to find jitter, which can also be applied to other eye parameters such as eye height and eye width. Training data is obtained directly by measuring jitter at each instance. Moreover, only rising and falling transition sub-models are needed because steady one and steady zero do not have a zero crossing. Using (1), the surrogate models for rising and falling edge jitter, $J_r(\boldsymbol{\lambda})$ and $J_f(\boldsymbol{\lambda})$, are defined as:

$$J_r(\boldsymbol{\lambda}) \approx \sum_{i=0}^{P} C_{\mathrm{r\_i}}\phi_i(\boldsymbol{\lambda}),\ \ J_f(\boldsymbol{\lambda}) \approx \sum_{i=0}^{P} C_{f\_i}\phi_i(\boldsymbol{\lambda}), \qquad (12)$$

where C_{r_i} and C_{f_i} are unknown coefficients for the rising edge and falling edge models, respectively. Using these coefficients, the RMS value of jitter for rising and falling edges, and the jitter distribution are calculated.

To create the full eye, we define a second type of output, which is the voltage of each time point on the X-axis of the eye diagram. In other words, $V(t, \boldsymbol{\lambda})$ is modeled over one UI. Training data is obtained by measuring the voltage of each time point over a number of intervals, and the surrogate models is found as:

$$V_{\mathrm{rs}}(t,\boldsymbol{\lambda}) \approx \sum_{i=0}^{P} C_{rs\_i}(t)\phi_i(\boldsymbol{\lambda}), \qquad (13)$$

where $0 \leq t \leq T$, and $r, s \in \{0,1\}$, which represent the four sub-models needed for modeling the receiver voltage. Thus, C_{00_i}, C_{01_i}, C_{10_i}, and C_{11_i}, are the unknown coefficients for steady zero, rising edge, falling edge, and steady one models, respectively. With these coefficients, mean and variance of the receiver voltage are found using (2) and (3), respectively. Furthermore, to draw the eye diagram, (13) is evaluated at an arbitrary number of possible samples, then the results are overlaid over one UI.

There are 2^{n-2} possible combinations of $\boldsymbol{\lambda}$ in (12) and (13). If $n \leq 20$, all possible combinations are calculated for drawing the jitter distribution and the eye diagram. Note that using more samples does not improve the results since all the possible values are

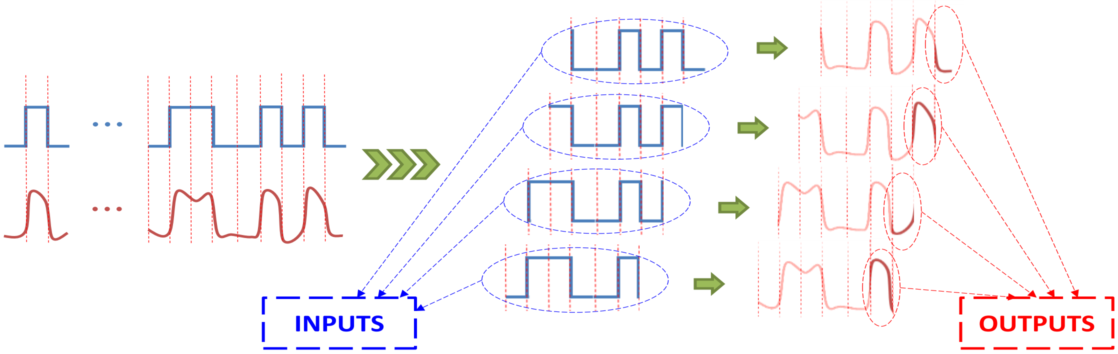


Figure 6 – Extraction of training samples from a single transient simulation.

included. If $n > 20$, it is not practical to estimate all possible outputs. Hence, a large enough subset of the samples is selected randomly.

Next, we discuss the training process. It is not efficient to run a separate simulation for each training sample. Hence, all necessary information is extracted from a single short transient simulation, with length of the simulation being much greater than P+1 and much smaller than the traditional eye. This process is illustrated in Figure 6. In this figure, input and output signals of the channel are shown. Imagine a box, with length of n bits, moving from end of the input signal to the left, with n being number of bits with effective ISI. Similarly, imagine a box with length of one bit, moving from end of the output signal to the left. We crop the signals at each instance, then boxes are moved one bit to the left. Each pair of corresponding inputs and outputs shows one sample or experiment. Number of input bits, n, is calculated based on the channel, using some heuristics. However, we use a rule of thumb, which states that $n * T \approx 3 * D$, where T is length of one unit interval, and D is delay of the channel. This formula considers effective ISI for a signal traveling through a channel and its two reflections. For $n \leq 20$, size and cost of creating the regular PC model is justifiable; however, for channels with a longer memory the computational cost can be prohibitive. Therefore, we suggest a hybrid methodology where the regular PC expansion is used for channels with $n \leq 20$ to obtain more accurate results, and the HPC expansion is applied for channels with a longer memory to reduce CPU costs. We show PC and HPC models with the same formulation; nevertheless, polynomials ϕ_i, and length of the expansions, shown with *P*, are different.

To find the coefficients with the regularized regression approach, suggested in 2.2, *N* samples are extracted from a short transient simulation, which provides $\boldsymbol{\lambda}^1$ to $\boldsymbol{\lambda}^N$ and $f(\boldsymbol{\lambda}^1)$ to $f(\boldsymbol{\lambda}^N)$. Using these values, matrices $\boldsymbol{A}$ and $\boldsymbol{E}$ in (8) are filled. Matrix $\boldsymbol{A}$ needs to be full rank in order to do the matrix inversion in (11). Initially, this might not be true because all the input variables are 0 or 1, which makes some of the columns a linear combination of other columns. Therefore, we remove the repeated columns and their corresponding polynomials ϕ_i since they do not provide any additional information. This would not cause a problem for estimation of new test samples because they are a combination of zeros and ones as well. After this modification, matrix $\boldsymbol{A}$ becomes full rank, and the unknown coefficients are found with the matrix inversion in (11). For example, coefficients in the surrogate model of the rising edge are found as:

$$\widehat{\boldsymbol{\Gamma}_{01}}(\mathrm{t}) = (\boldsymbol{A}^T\boldsymbol{A} + \boldsymbol{B}^{\tau}\boldsymbol{B})^{-1}\boldsymbol{A}^T\boldsymbol{E}(t), \tag{14}$$

where $\widehat{\boldsymbol{\Gamma}_{01}}(\mathrm{t})$ is a vector of C_{rs_i} coefficients. After training the models, we use the surrogate models of (13) for a validation step, where they are used to estimate the receiver voltage for a short sequence of random input pulses, not used for model training. If this matches with the actual transient simulation, we proceed to find the statistics and the estimated eye. Otherwise, the parameters are further tuned to provide a more accurate surrogate model.

It is worth noting, that random jitter can be added to this analysis later as single or multiple Gaussian distributions. Adding random jitter in post-processing is a common practice in commercial solvers such as HSSCDR, a specialized channel simulator, used by IBM [50]. Therefore, we suggest the same technique to integrate the random jitter with results of the proposed approach.

2.4 Computational Cost Analysis

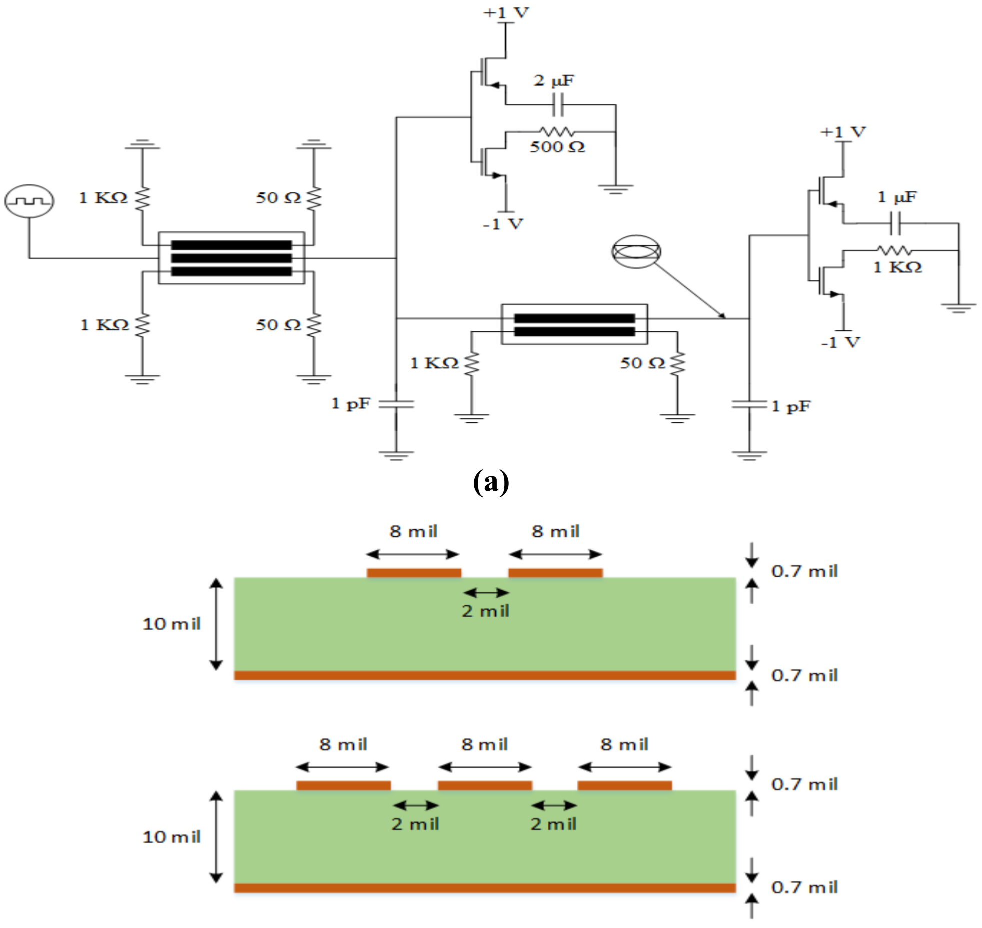
The proposed approach can reduce cost of the physical transient simulation significantly. We suggest using this approach for problems where overhead cost of developing the surrogate models is negligible compared to the transient simulation. Note that, ϕ_i polynomials are known beforehand, and by keeping the same sequence of random training data, all the parameters of $(\boldsymbol{A}^{\tau}\boldsymbol{A} + \boldsymbol{B}^{\tau}\boldsymbol{B})^{-1}\boldsymbol{A}^{\tau}$ in (11) can be calculated and stored in advance. Therefore, the main overhead cost in the training process is the remaining matrix vector multiplication in (11). The computations cost of this multiplication is $O((P+1)^2)$, and it needs to be calculated for every sub-model in (13). There are four sub-models, and assuming there are Ψ time points in one UI, the corresponding overhead cost is $O(4\Psi(P+1)^2) = O(\Psi P^2)$. Cost of the direct method for training the jitter models in (12) is negligible in comparison since the matrix vector multiplication is done only twice. Furthermore, the testing process and calculating the full eye diagram involves evaluating (13) at ν samples, where the $\phi_i(\boldsymbol{\lambda})$ values can be calculated beforehand. In the matrix form, this process is similar to (8), where $\boldsymbol{A}$ is computed in advance and $\boldsymbol{\Gamma}$ is known. Hence, the computational cost to find $\boldsymbol{E}$ is of a matrix vector multiplication, which is $O\ (\nu(P+1))$. With Ψ time points in one UI, the corresponding cost for evaluating the voltage models is $O(\Psi\nu(P+1)) = O(\Psi\nu P)$. Again, the CPU cost of directly finding jitter for ν samples is negligible in comparison. Therefore, the total overhead cost is $O(\Psi\nu P + \Psi P^2)$. It is worth noting, this approach is highly parallelizable; hence, its computation time can significantly decrease with parallelization

2.5 Numerical Examples

In this section the proposed approach is applied to numerical examples to evaluate the performance of the PC based surrogate models. PC calculations and modeling are done in MATLAB R2015a, while transient simulation is performed using available commercial software.

2.5.1 Example 1

The circuit in Figure 7 (a) is considered for observing performance of the proposed approach, when memory of the channel is relatively small, and a full PC expansion can be



(a)

(b)

Figure 7 – Example 1: A single-ended signaling high-speed link with coupled transmission lines and nonlinear terminations. a) Circuit schematics. b) Cross section of the transmission lines.

used. This example shows a single-ended signaling system, which includes two sets of coupled microstrip lines and nonlinear terminations. The cross section of microstrip lines is shown in Figure 7 (b). The first set of lines are three coupled lines that are 4 inches long, while the second set consists of two coupled lines with length of 2 inches. All conductors are copper; dielectric permittivity is 4.5, and $\tan\delta = 0.02$. The PMOS and NMOS transistors in the terminations are Schicman-Hodges models. Supply voltage is +/- 1 V, and the circuit is fed through a voltage source with 50 Ω resistance, using a random trapezoidal pulse with $V_{high} = 1$ V and $V_{Low} = -1$ V. Bit rate is 1 Gb/s; rise time and fall time are both 100 ps, and output voltage is observed at the shown eye probe. Finally, the transient simulation is done using ANSYS Electronics Desktop 17.2 circuit simulator [51].

In this example, 20 bits are considered to capture the effective ISI. Hence, we use the regular PC expansion. In addition, there are 2^{20} possible combinations of these bits; therefore, all possible outputs are estimated. Note that using more samples does not change the result since all the possible combinations are included. The results are compared with an actual transient simulation with one million random bits. We did not use more than one million samples for the transient eye since variations in the outcome were minimal, and we were limited by the memory of our machine. Moreover, maximum order of expansion, m, is set to 3. For training the models, first a short transient simulation is performed, with its length being more than 60000 bits. From this simulation, four sets of training data are extracted. Each set shows a different transition and has about 15000 samples. Using them and the proposed method, jitter and voltage surrogate models are generated. For validation, the receiver voltage for a short sequence of random pulses is estimated. The result is shown in Figure 8, and compared with the actual circuit simulation. Since the model shows perfect accuracy, we continue to calculate the statistical results and the eye diagram. Next, the jitter RMS is directly calculated, and it is presented in Table 1. Additionally, distribution of jitter

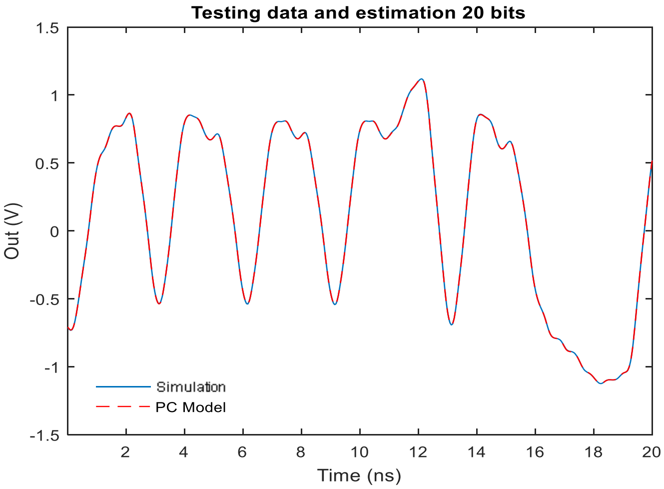


Figure 8 – Estimation of the output voltage for validation of the surrogate models in example 1.

Table 1 – RMS jitter values of Example 1.

	Low to high RMS jitter	High to low RMS jitter
Transient eye analysis	87.2 ps	87.9 ps
Proposed PC approach	87.3 ps	87.8 ps

is calculated and shown in Figure 9 and compared with transient simulation, showing a good agreement.

In the next step, the mean and standard deviation of receiver voltage is calculated over one UI, and the results are shown in Figure 10. In this figure, the mean and mean +/- standard deviation of the four possible transitions are shown, and compared with the transient simulation. Figure 10 gives us an idea about quality of the signal. The mean and standard deviation in this figure are found using only coefficients of the voltage model;

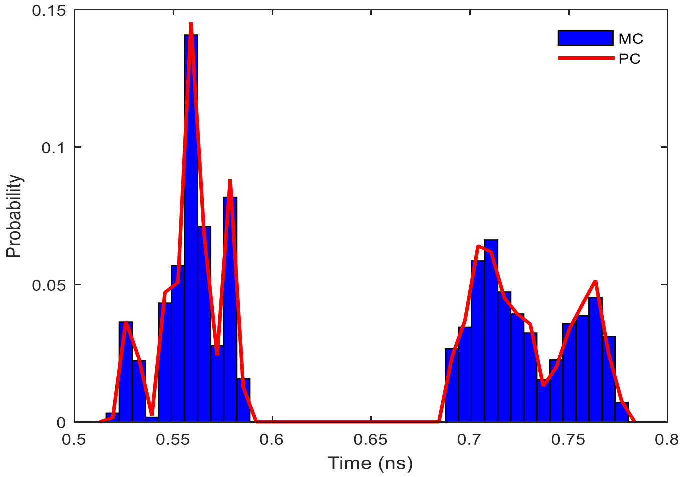


Figure 9 – Distribution of jitter seen in the output of Example 1.

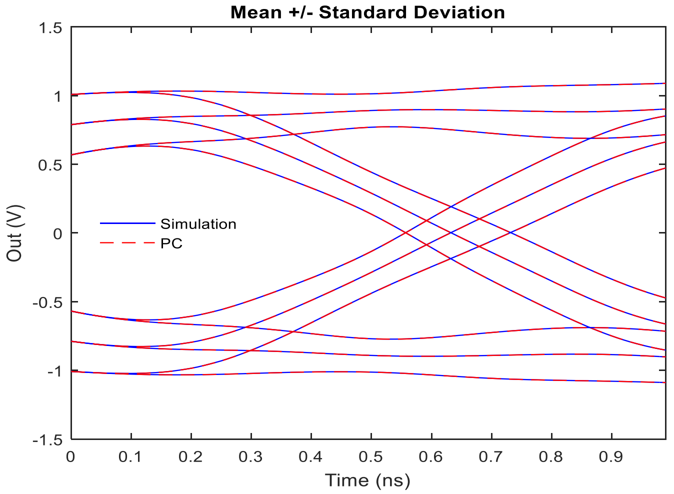
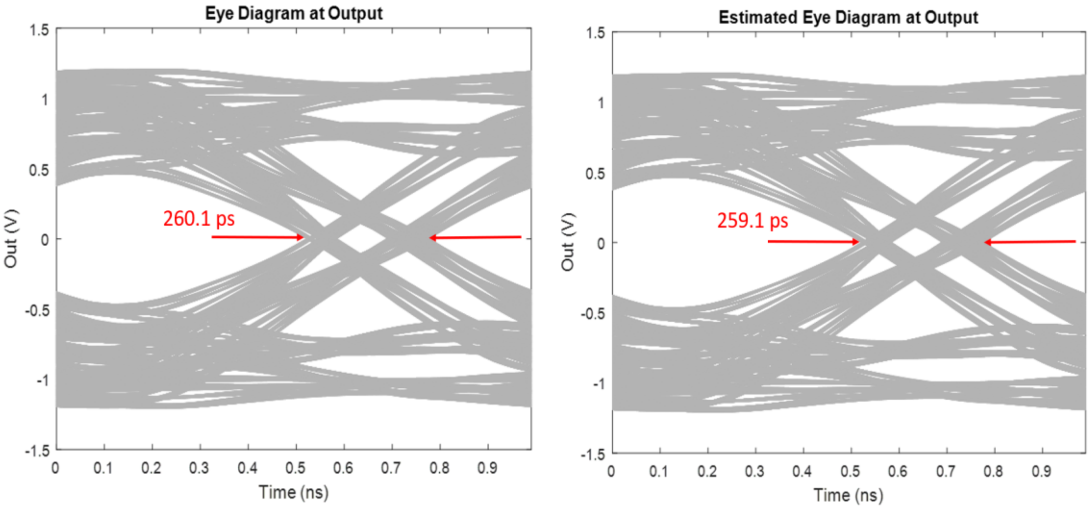


Figure 10 – Mean +/- Standard deviation of the four possible transitions of the output in Example 1.

hence, it reduces time and memory. Finally, the eye diagram is evaluated, and shown in Figure 11 (b). The eye diagram, from the transient simulation with one million bits, is shown in Figure 11 (a). Peak to peak jitter is also displayed in the Figure 11 (a) and (b), which is calculated from the final eye diagram. In all the results it is observed that the



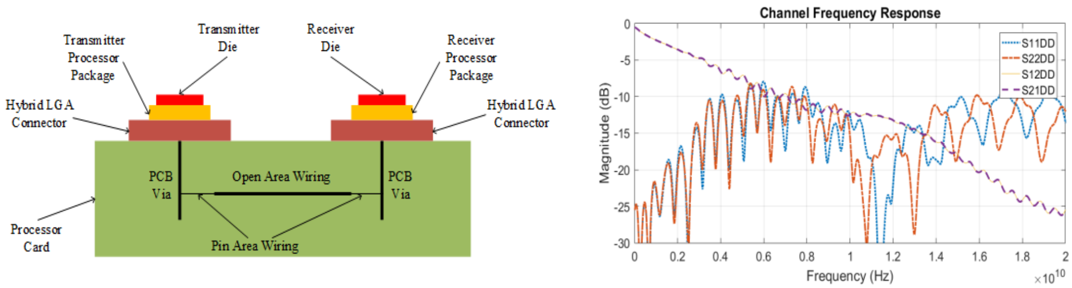
(a) **(b)**

Figure 11 – Eye diagram and peak to peak jitter of the output in Example 1, showing 10000 out of one million random bits. a) Transient eye. b) Proposed approach.

Table 2 – Computation times of different steps in Example 1.

	Time
60000 bits sim. with Ansys	663 s
Training jitter and voltage models	102 s
Jitter Distribution	196 s
1 million bits estimation with PC	3827 s
1 million bits sim. with Ansys	11055 s

proposed approach matches well with the transient simulation. Moreover, computation time of the main steps, except the initialization, are reported in Table 2. It is observed that the CPU cost for determining jitter and statistics is negligible. The full eye diagram



(a) **(b)**

Figure 12 – Example 2: A SerDes channel including packages, vias and differential wiring. a) Schematics of the channel. b) Frequency response.

evaluation is more time consuming; however, it is about 3 times faster than the transient eye. Note that this time can be significantly reduced by using a shorter expansion. Moreover, the required memory for finding mean and variance values is reduced by about 35%.

2.5.2 Example 2

The purpose of this example is to show performance of the proposed approach for a channel with long memory, where the Hyperbolic PC expansion needs to be used. Therefore, a commercial SerDes channel with differential signaling is considered, provided by IBM. This channel is shown in Figure 12 (a), and its frequency response is presented in Figure 12 (b). Output of the system is the receiver voltage. Moreover, the channel is comprised of two processor packages, communicating with each other and interfaced to a board with two hybrid land grid array (LGA) connectors. The transmitter and receiver processor packages contain 85 Ohm differential stripline wiring in GZ41 material (Dk ~ 3.31 and Df ~ 0.0092 at 1 GHz), and have 31 mm and 34 mm lengths, respectively. The

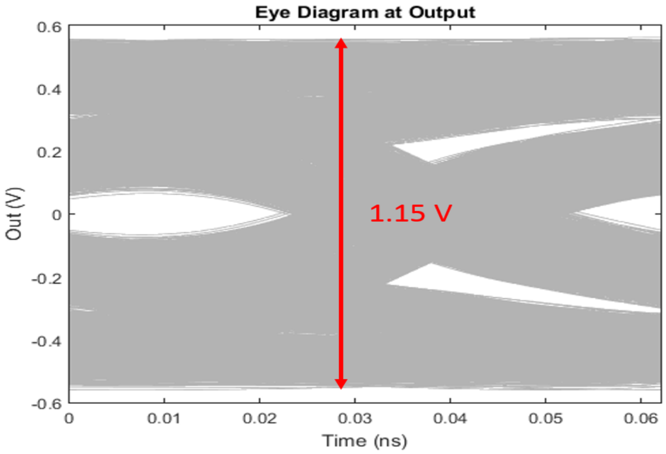


Figure 13 – Transient eye diagram of the output in example 2 before decreasing the compression point of the receiver, showing 10000 out of one million random bits.

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. The dielectric material utilized for the board is a low loss material, having a Dk ~ 3.95 and Df ~ 0.0084 at 1GHz. This channel is simplified and does not include crosstalk. The passive channel loss at 8 GHz is ~11 dB. The high-speed link, including the channel, transmitter and receiver is tested with a bit rate of 16 Gb/s. For transient simulations the channel is simulated with HSSCDR [50]. In addition, to ensure nonlinearity a low compression point for the receiver is considered. To show the impact, eye diagrams of the output before and after decreasing the compression point, are shown in Figure 13 and Figure 17 (a), respectively. Moreover, the difference of maximum and minimum of the receiver voltage is indicated in both figures. Comparison of the figures shows that eye

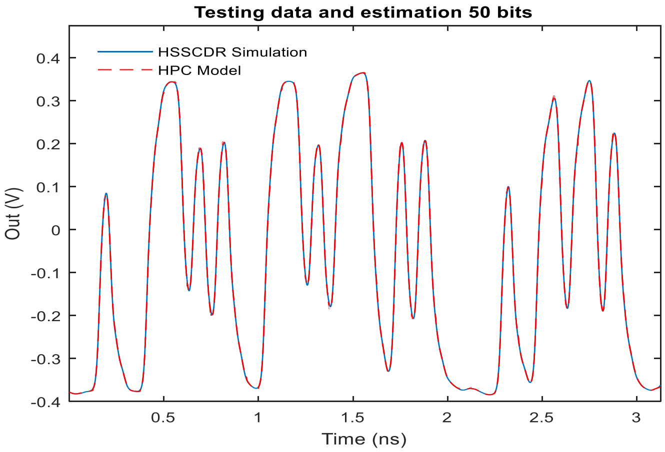


Figure 14 – Estimation of the receiver voltage for validation of the surrogate models in example 2.

diagram is considerably affected by the low compression point. Note that the compression is nonlinear.

In this example, due to long delay of the channel, 50 bits are considered for capturing the ISI effect. Since size of the regular PC expansion would be prohibitive, we use the HPC expansion. The maximum order of polynomials m is set to 3, and the constraining factor u is set to 0.7. Moreover, there are 2^{50} possible combinations of input bits; therefore, we select a random subset from them with one million samples to estimate the outputs. Furthermore, a transient simulation with a different one million random bits is simulated using HSSCDR for comparison. We did not use more than one million samples since variations in the outcome were minimal, and we were limited by the memory of our machine.

Table 3 – RMS jitter values of Example 2.

	Low to high RMS jitter	High to low RMS jitter
Transient eye analysis	7.1 ps	7.1 ps
Proposed PC approach	7.0 ps	7.0 ps

For training the models, first a short transient simulation is performed, with its length being more than 50000 bits. From this simulation, four sets of training data are extracted. Each set shows a different transition and has 12500 samples. Next, jitter and voltage surrogate models are generated. For validation, a short random sequence of receiver voltage is estimated and shown in Figure 14. Since the model shows perfect accuracy, we continue to calculate the statistical results and the eye diagram.

The jitter RMS is found directly, and presented in Table 3. Distribution of jitter is calculated and shown in Figure 15. Next, mean and standard deviation of the receiver voltage is found over one UI, and results are shown in Figure 16. Finally, the eye diagram is evaluated, and shown in Figure 17 (b). The eye diagram from the transient simulation with one million bits, is shown in Figure 17 (a). The difference of maximum and minimum of the receiver voltage is also displayed in Figure 17 (a) and (b). Peak to peak jitter values calculated with the transient eye and the proposed approach are 34.7 ps and 35.1 ps, respectively. In this example, it is observed that replacing the PC expansion with an HPC expansion provides for a good estimation. All training and testing steps, including obtaining the eye diagram, were completed in about 50 minutes. This includes calculation

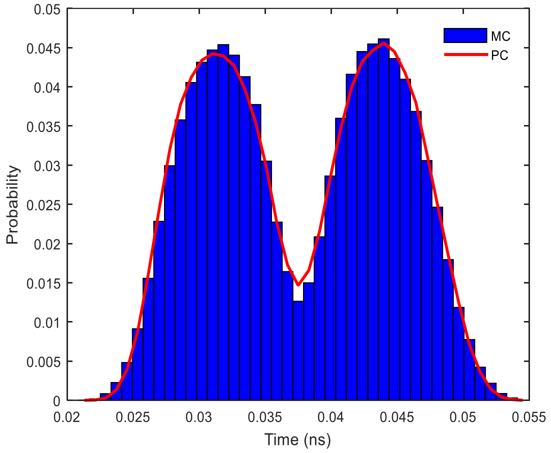


Figure 15 – Distribution of jitter seen in the receiver voltage of example 2.

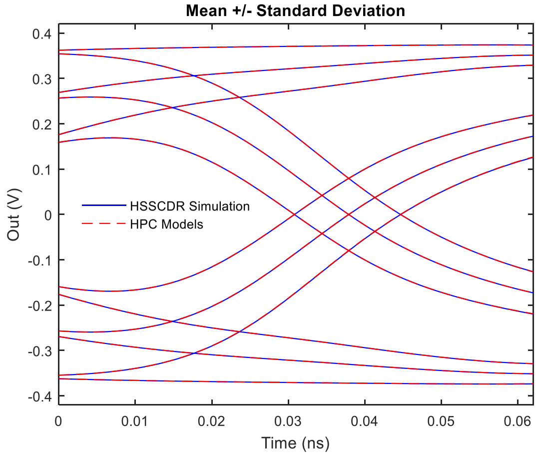
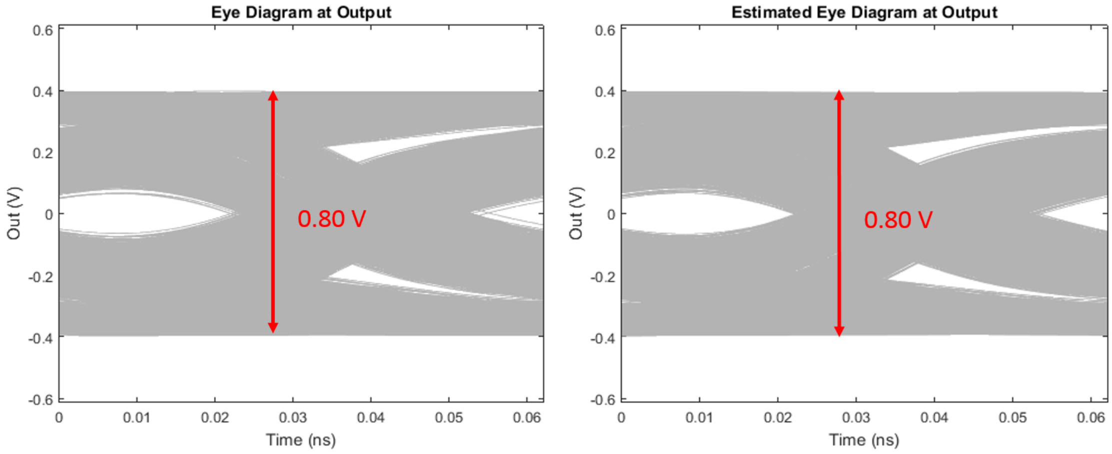


Figure 16 – Mean +/- σ of four possible transitions in example 2.

of mean and variance values of the receiver voltage and jitter, which takes negligible time. Besides, the required memory for finding mean and variance values was reduced by about 50%. We have published the work presented in this chapter in [52].

2.6 Summary



(a) **(b)**

Figure 17 – Eye diagrams of the output in example 2 after decreasing compression point of the receiver, showing 10000 out of one million random bits. a) Transient eye. b) Proposed approach.

This chapter covers the background knowledge needed for the first proposed approach for efficient eye analysis. Moreover, development steps of this approach are discussed. Numerical examples demonstrate that the proposed approach provides the full eye diagram faster than the traditional transient eye analysis. In addition, the jitter distribution and other eye diagram statistics are provided with a significantly higher speedup.

CHAPTER 3. WORST-CASE EYE ANALYSIS OF HIGH-SPEED CHANNELS BASED ON BAYESIAN OPTIMIZATION

3.1 Objectives

In this chapter our second eye diagram analysis approach is presented. This approach takes an optimization perspective to find the input waveforms that result in the inner opening of the eye diagram. We refer to the inner eye opening as the worst-case eye. This approach provides a higher speedup compared to the previous approach since the full eye diagram is not calculated. However, the eye height, eye width, and the worst-case eye which are the desired measurements in many applications are found with a significantly higher speedup. Initially the theoretical background needed for this approach is reviewed, which includes the Gray Code and the Bayesian Optimization. Then, the proposed approach is discussed in detail. Finally, the chapter ends with numerical examples, demonstrating the performance of this approach.

3.2 Background Review

3.2.1 Reflected Binary Code (Gray Code)

The reflected binary code or the Gray code, originally introduced by Frank Gray in 1947 [53], refers to a reordering of the binary numbers where each two subsequent values only differ in a single binary bit [11]. Gray code is particularly helpful in digital

Table 4 – 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

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 bits, and their corresponding Gray code are shown in Table 4. 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 [54]:

$$k \hat{} (k >> 1), \tag{15}$$

where, ^ and >> are bitwise exclusive OR and shift-right operators, respectively. In this work we have used (15) since it is more efficient than the recursive method.

3.2.2 Bayesian Optimization (BO)

BO [12], [55] is an active learning optimization algorithm which has shown a good performance in optimizing nonconvex and nonlinear complex 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:

$$P(f(x)|D_{1:t}) \propto P\big(D_{1:t}|f(x)\big)P\big(f(x)\big), \tag{16}$$

where $D_{1:t} = \{x_{1:t}, f_{1:t}\}$ is the set of t samples and evaluations. $P\big(f(x)\big)$ and $P(f(x)|D_{1:t})$ are prior and posterior distributions, respectively. Moreover, $P\big(D_{1:t}|f(x)\big)$ is the likelihood of observing $D_{1:t}$, given our prior belief of $P\big(f(x)\big)$. In other words, BO corrects our belief of 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 $max\big(f(x)\big)$. 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 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 work is the Matérn function with smoothness factor of 2.5 [56]:

$$k(r) = \left(1 + \frac{\sqrt{5}r}{\lambda} + \frac{5r^2}{3\lambda^2}\right) \exp\left(-\frac{\sqrt{5}r}{\lambda}\right), \tag{17}$$

where $r = \|x_i - x_j\|$, and λ is an optimizable length factor (see the documentation in [57]). Using this kernel, posterior distribution of f at $t + 1$ is calculated as:

$$\begin{gathered} P(f_{t+1} | D_{1:t}, x_{t+1}) = \mathcal{N}(\mu_{t+1}, \sigma_{t+1}^2), \\ \mu_{t+1} = \vec{K}^T \boldsymbol{K}^{-1} f_{1:t}\,, \qquad \sigma_{t+1}^2 = k(x_{t+1}, x_{t+1}) - \vec{K}^T \boldsymbol{K}^{-1} \vec{K}, \end{gathered} \tag{18}$$

where $\vec{K} = [k(x_1, x_{t+1}), k(x_2, x_{t+1}), \dots, k(x_t, x_{t+1})]$. 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 probability of improvement (*PI*), expected improvement (*EI*), and upper confidence bound (*UCB*), which are defined as:

$$PI(x) = P(f(x) \geq f(x^{+}) + \varepsilon) = \Phi\left(\frac{\mu(x) - f(x^{+}) - \varepsilon}{\sigma(x)}\right), \tag{19}$$

$$\begin{gathered} EI(x) = \begin{cases} (\mu(x) - f(x^{+}) - \varepsilon)\Phi(Z) + \sigma(x)\phi(Z) & if\ \sigma(x) > 0 \\ 0 & Otherwise \end{cases}, \\ Z = \frac{\mu(x) - f(x^{+}) - \varepsilon}{\sigma(x)}, \end{gathered} \tag{20}$$

$$UCB(x) = \mu(x) + \kappa\sigma(x), \tag{21}$$

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 $x_{t+1} = \underset{x}{\operatorname{argmax}}\, u(x|\, D_{1:t})$, with $u(.)$ being one of the acquisition functions. To further improve convergence of BO, we use the GP_Hedge acquisition function, which probabilistically selects one of the above three functions at each iteration [58]. 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 paper.

3.3 Proposed Worst-eye Approach

3.3.1 Intuition

Although for a comprehensive examination of the signal, the distributions of noise and jitter is required, engineers often use eye height and eye width 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 Figure 3, eye height (EH) is the difference between the lowest high and the highest low received symbols at the sampling time. Moreover, eye width (EW) is equal to a unit interval minus the peak to peak jitter. Note that the peak to peak jitter is difference of the rightmost and the leftmost level-crossing time points, caused by the rising and falling edges. Since in this paper we are only concerned with DDN, DDJ, ISI and crosstalk, one can use optimization techniques to find the sequences of symbols that results 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, number of the sampling time points, for evaluating the received symbol, is simply increased. Then, 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 non-return-to-zero (NRZ) pulses; however, the idea can be expanded to other signaling methods. Hereinafter, the proposed approach is referred to as Worst-eye.

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 eye height and eye width fulfills in this case. The channel designer needs to select the settings that maximizes the eye opening to provide the maximum flexibility for the circuit designer.

Moreover, the focus of this paper 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 random jitter (RJ) and random noise (RN) are a key part in calculating the BER. The worst-case BER can be calculated by superimposing 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 paper. Therefore, the final BER directly depends on the worst-case eye height and eye width. Hence, we need to maximize the eye opening when designing the channel to achieve a lower BER.

3.3.2 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 unit intervals (e.g. > 20). In addition, the search space is discrete and sparse because 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:

$$\boldsymbol{\lambda} = [\lambda_{-n}, \lambda_{-n+1}, \dots, \lambda_{-1}, \lambda_0, \lambda_1, \dots, \lambda_m\ ], \tag{22}$$

where *n* precursors and *m* postcursors are considered. In addition, 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 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 paper. 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 (LIB) and λ_{-1} as the most important bit (MIB), we obtain a unique binary index number, $I(\boldsymbol{\lambda})$, for each possible bit pattern. Note that $0 \le I(\boldsymbol{\lambda}) < 2^n$. This format is shown in Figure 18.

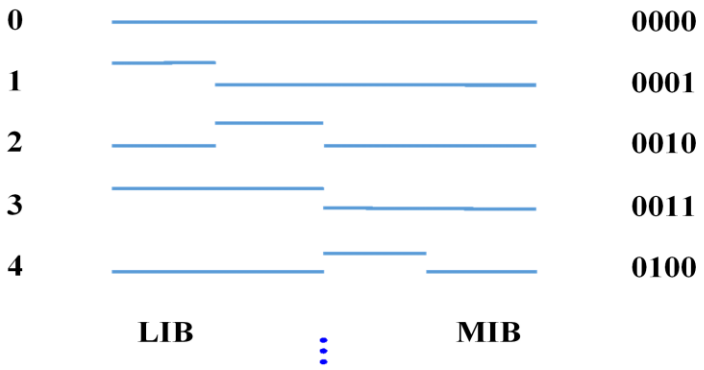
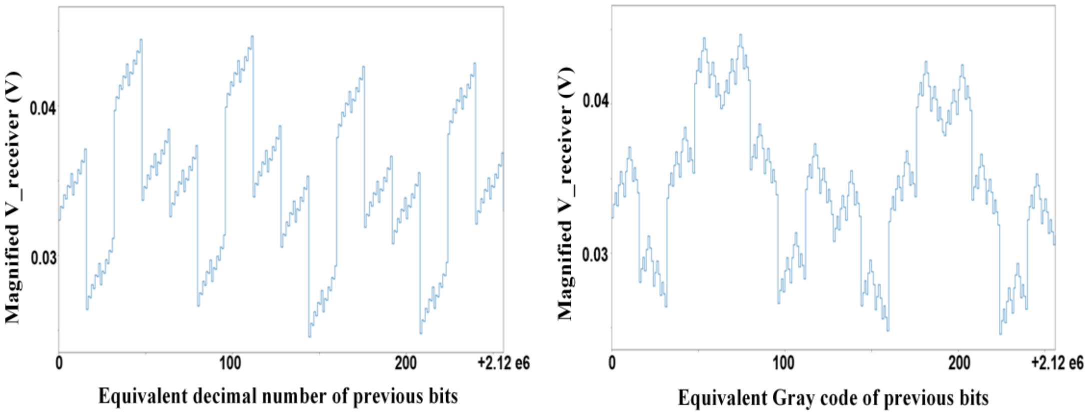


Figure 18 – Indexing of the bit patterns for n=4 in the binary format.

Next step would be to find the $I(\boldsymbol{\lambda})$ that minimizes or maximizes the objective functions. This technique maps the problem from n-dimensions to one. However, as stated in section 3.2.1, 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 Figure 18 are reordered as {0000, 0001, 0011, 0010, 0110, …}. For comparison, an example of corresponding sections of $V_r\left(t_s, I(\boldsymbol{\lambda})|_{\lambda_0=1,\, \lambda_1=0}\right)$ and $V_r\left(t_s, I_g(\boldsymbol{\lambda})|_{\lambda_0=1,\, \lambda_1=0}\right)$ is illustrated in Figure 19 (a) and Figure 19 (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.



(a) **(b)**

Figure 19 – Receiver voltage as a function of the index values corresponding to the previous bit patterns. a) Binary order. b) Gray code order.

Finally, it is worth noting that variations of the Gray code [11] and other coding schemes exist, which have characteristics similar to the ones discussed here. However, in this paper we have settled for the original Gray code [53], and did not experiment with other coding mechanisms.

3.3.3 Optimization

The mapping scheme results in objective functions that are suitable for BO. Therefore, we use the BO algorithm, reviewed in section 3.2.2, 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:

$$V_{LH} = \min_{I_g} \; V_r\big(t_s, I_g(\boldsymbol{\lambda})|_{\lambda_{-1}=0,\lambda_0=1,\,\lambda_1=0}\big), \tag{23}$$

$$V_{HL} = \max_{I_g} \; V_r\big(t_s, I_g(\boldsymbol{\lambda})|_{\lambda_{-1}=1,\lambda_0=0,\,\lambda_1=1}\big), \tag{24}$$

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:

$$t_{LX} = \min \begin{Bmatrix} \min\limits_{I_g} \; \mathrm{t}\big(V_0, I_g(\boldsymbol{\lambda})|_{\lambda_{-1}=0,\lambda_0=1,\,\lambda_1=1}\big), \\ \min\limits_{I_g} \; \mathrm{t}\big(V_0, I_g(\boldsymbol{\lambda})|_{\lambda_{-1}=1,\lambda_0=0,\,\lambda_1=0}\big) \end{Bmatrix}, \tag{25}$$

$$t_{RX} = \max \begin{Bmatrix} \max\limits_{I_g} \; \mathrm{t}\big(V_0, I_g(\boldsymbol{\lambda})|_{\lambda_{-1}=0,\lambda_0=1,\,\lambda_1=0}\big), \\ \max\limits_{I_g} \; \mathrm{t}\big(V_0, I_g(\boldsymbol{\lambda})|_{\lambda_{-1}=1,\lambda_0=0,\,\lambda_1=1}\big) \end{Bmatrix}, \tag{26}$$

where $0 \le \mathrm{t}(V_0, \mathrm{i}) < UI$, and $\mathrm{t}(V_0, \mathrm{i})$ 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 (25) or in (26), 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 (23) and (25) 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. $k \leq 10$), 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 approach, $\boldsymbol{\lambda}$ and size of the sample space of BO in (23) and (24) are:

$$\boldsymbol{\lambda} = [\lambda_{-n+k}, \dots, \lambda_{-2}, \lambda_{-1}, \lambda_0, \lambda_1\,], \;\; \|\Omega_{\boldsymbol{\lambda}}\| = 2^{n-k-1}. \tag{27}$$

Moreover, $\boldsymbol{\lambda}$ and size of the sample space of BO in (25) and (26) are:

$$\boldsymbol{\lambda} = \left[\lambda_{-n+k}, \dots, \lambda_{-2}, \lambda_j, \lambda_{-1}, \lambda_0, \lambda_1\,\right], \;\; \|\Omega_{\boldsymbol{\lambda}}\| = 2^{n-k}. \tag{28}$$

For illustration, in an example eye diagram the waveforms that correspond to the bit patterns ending in $\{\lambda_{-1} = 1, \lambda_0 = 0,\ \lambda_1 = 1\}$ are shown in blue color in Figure 20. 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, Figure 21 illustrates the objective

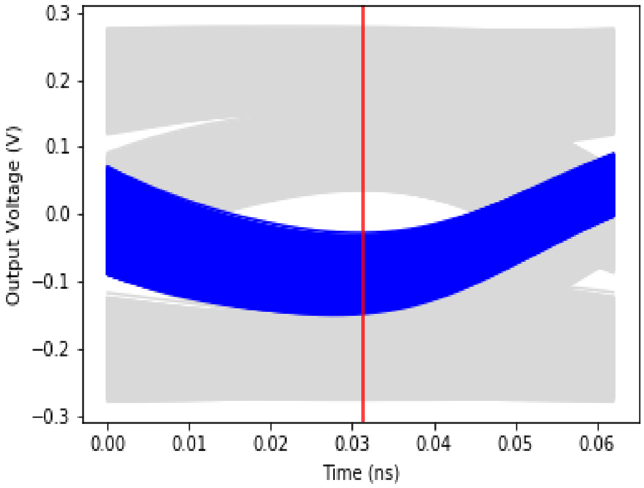


Figure 20 – Example eye diagram and the waveforms corresponding to the bit patterns ending in $\{\lambda_{-1} = 1, \lambda_0 = 0,\ \lambda_1 = 1\}$, shown in blue.

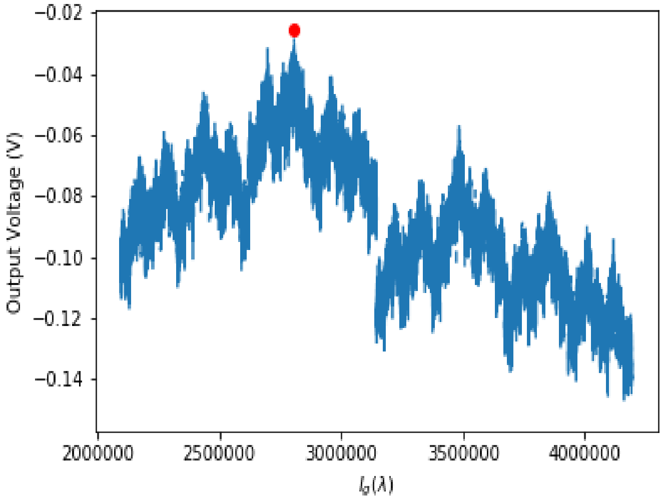


Figure 21 – The objective function which is passed to BO to find V_{HL} in Figure 20.

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 Figure 21.

Next, eye height and eye width are calculated as:

$$EH = V_{HL} - V_{LH}, \tag{29}$$

$$EW = UI - (t_{RX} - t_{LX}). \tag{30}$$

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 [59], [60]. More advanced methods for BO with discrete variables exist in the literature [61], [62], [63], 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 (23) to (26). If higher accuracy for the worst-case eye is desired, additional t_s sampling points are defined. Then, (23) and (24) 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 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 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.

3.3.4 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 presence of the crosstalk.

In order to do so, one variable per aggressor line is added to the optimization problems in (23) to (26). 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 $1 \leq j \leq q$, 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 is not intuitively clear, and it is better to leave their determination to the optimization algorithm. In addition, no postcursor is taken into account, and 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 is not cumbersome. Considering crosstalk, the new optimization problems are:

$$V_{LH} = \min_{I_g,J_g^1,J_g^2,\dots,J_g^q} V_r\left(t_s, I_g|_{\lambda_{-1}=0,\lambda_0=1,\,\lambda_1=0}, J_g^1, J_g^2, \dots, J_g^q\right), \tag{31}$$

$$V_{HL} = \max_{I_g,J_g^1,J_g^2,\dots,J_g^q} V_r\left(t_s, I_g|_{\lambda_{-1}=1,\lambda_0=0,\,\lambda_1=1}, J_g^1, J_g^2, \dots, J_g^q\right), \tag{32}$$

$$t_{LX} = \min\left\{\begin{matrix} \min\limits_{I_g,J_g^1,J_g^2,\dots,J_g^q} \mathrm{t}\left(V_0, I_g|_{\lambda_{-1}=0,\lambda_0=1,\,\lambda_1=1}, J_g^1, J_g^2, \dots, J_g^q\right), \\ \min\limits_{I_g,J_g^1,J_g^2,\dots,J_g^q} \mathrm{t}\left(V_0, I_g|_{\lambda_{-1}=1,\lambda_0=0,\,\lambda_1=0}, J_g^1, J_g^2, \dots, J_g^q\right) \end{matrix}\right\}, \tag{33}$$

$$t_{RX} = \max\left\{\begin{matrix} \max\limits_{I,J_g^1,J_g^2,\dots,J_g^q} \mathrm{t}\left(V_0, I_g|_{\lambda_{-1}=0,\lambda_0=1,\,\lambda_1=0}, J_g^1, J_g^2, \dots, J_g^q\right), \\ \max\limits_{I,J_g^1,J_g^2,\dots,J_g^q} \mathrm{t}\left(V_0, I_g|_{\lambda_{-1}=1,\lambda_0=0,\,\lambda_1=1}, J_g^1, J_g^2, \dots, J_g^q\right) \end{matrix}\right\}, \tag{34}$$

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 (23) to (26). Moreover, $\boldsymbol{\xi}^j$and size of its sample space is:

$$\boldsymbol{\xi}^j = \left[\xi^j_{-h^j-1}, \xi^j_{-h^j-2}, \dots, \xi^j_0\right], \ \left\|\Omega_{\xi^j}\right\| = 2^{h^j}, \tag{35}$$

where h^j is the number of effective symbols considered for the *j*-th aggressor line. Next, EH and EW are calculated using (29) and (30), respectively. Finally, the worst-case eye with crosstalk is estimated by overlaying the four waveforms that pass through the points determined by (31) to (34). To increase accuracy of the worst-case eye, similar to the case with no crosstalk, additional waveforms are generated by changing t_s and repeating (31) and (32).

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 is converted to PAM-4 signaling. Using Gray codes for PAM-4 is a standard practice [64]. For instance, for two bits 00, 01, 11, and 10 correspond to -1. -1/3, +1/3, and +1, respectively. 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.

3.4 Numerical Examples

In this section, 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 [59] is used for BO, which is based on [57], [65]. 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 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 [62], [66]. 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 [57]), and the Worst-eye functions can show such behavior.

3.4.1 Example 1

The purpose of this example is to evaluate Worst-eye when ISI is dominant. Therefore, the SerDes channel in Figure 22, which was introduced in the previous Chapter, is utilized. As a reminder, this channel does not have any crosstalk, and it includes two

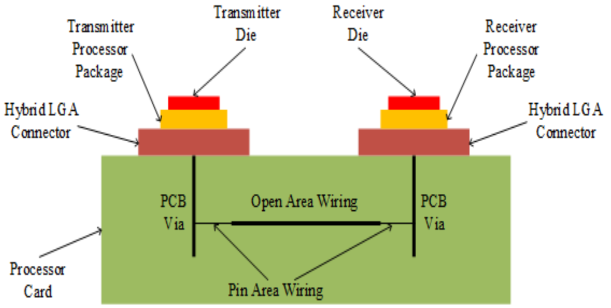


Figure 22 – The high-speed SerDes channel in Example 1, comprised of two packages, vias, and differential wiring.

processor packages connected to a board with hybrid land grid array (LGA) connectors. The passive channel loss is ~11 dB at 8 GHz. 85 Ohm 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 ~ 3.31 and Df ~ 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 inch of necked down pin area wiring in the shadow of each processor, and 2 inches of 85 Ohm differential wiring in the open area, which amounts to 4 inches of wiring on the board.

Moreover, the dielectric of the board is a low loss material with Dk ~ 3.95 and Df ~ 0.0084 at 1GHz. This channel is simulated in a custom-build solver named HSSCDR [50], [67], 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. We increased the number of bits in the transient eye, compared to the

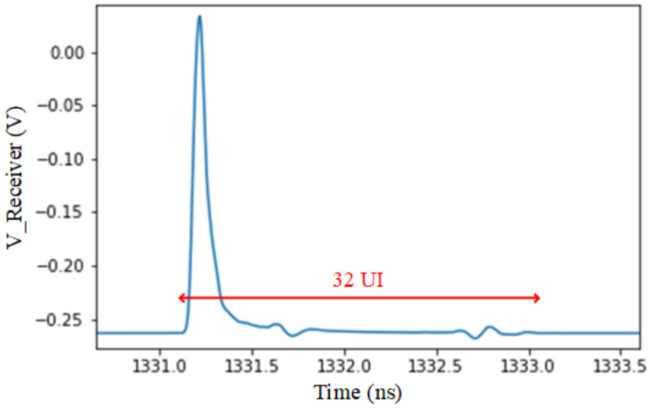


Figure 23 – Pulse response of the channel in Figure 22 when pulse width = 62.5 ps.

previous chapter, to ensure capturing the worst-case bit patterns. 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 16Gb/s; hence, the unit interval is equal to 62.5 ps. Pulse response of the channel is shown in Figure 23. 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 bit patterns, where length of each pattern is 32. This results in simulating transmission of $256 * 32 = 8{,}192$ bits 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 bits, while for t_{LX} or t_{RX} BO determines value of 22 bits. However, for each of them only a one-dimensional optimization is performed since

Table 5 – 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

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 5, where they are compared with the transient eye analysis as well. The comparison shows that Worst-eye with Gray code mapping provides slightly smaller eye height and eye width 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 5 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 5. Number of bits for either of the Worst-eyes is less than 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 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 paper 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 Figure 24. For aesthetic reasons we have shifted the eye diagram and waveforms found by Worst-eye.

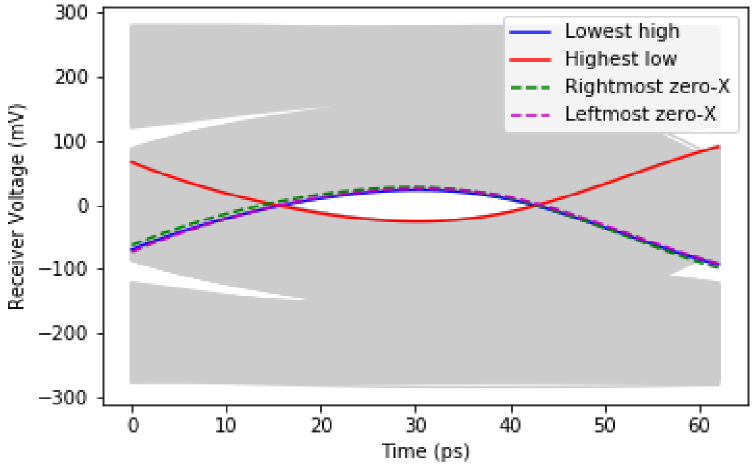
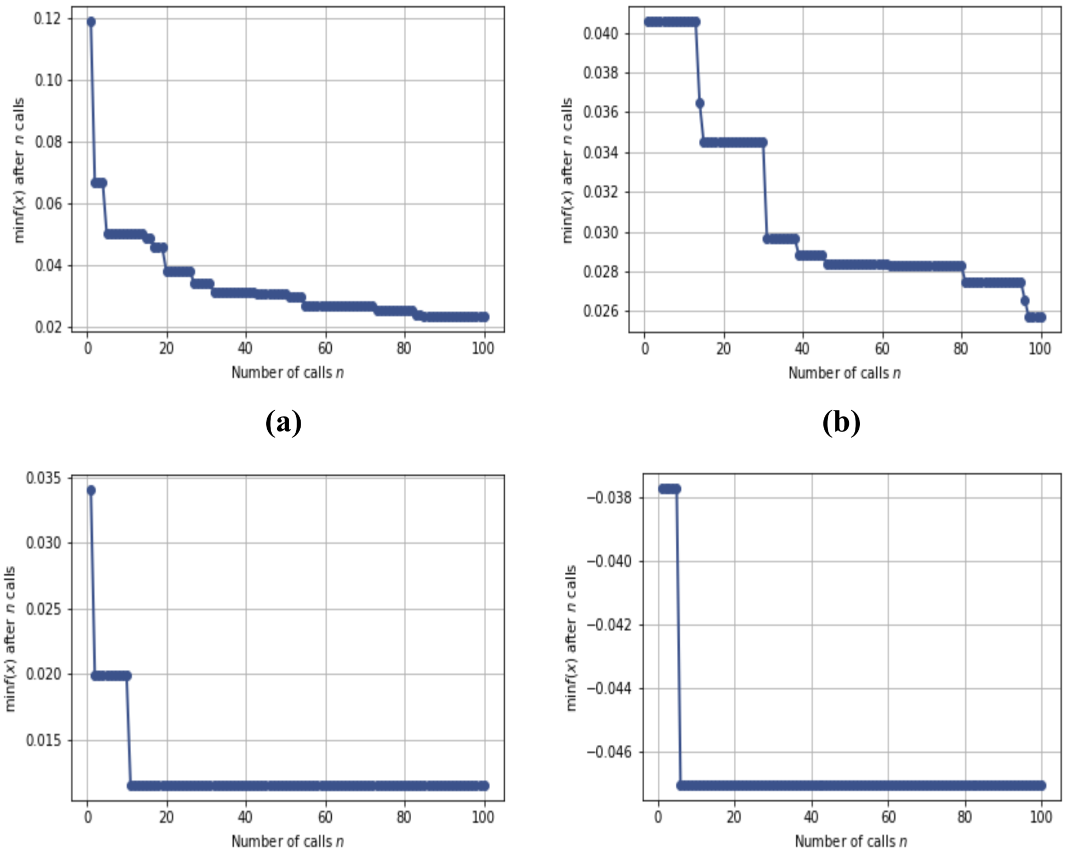


Figure 24 – Transient eye and the Worst-eye waveforms in Example 1.

Hence, there is some difference between the level-crossing time points in Table 5 and Figure 24; nevertheless, both show the same EH and EW. Next, the convergence curves of BO for the four objective functions are presented in Figure 25. The curves show that the minimum value decreases drastically which is due to the fast convergence rate of BO. Nevertheless, Figure 25 (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 5 is considered. The HSSCDR simulation for the transient eye and the Worst-eye roughly takes 34 and 9 minutes, respectively. Moreover, the overhead optimization cost of the proposed Worst-eye approach is about 6 minutes, 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



(a) **(b)**

(c) **(d)**

Figure 25 – Convergence plots of the four worst-case variables in Example 1. a) V_{LH}. b) $-V_{HL}$. c) t_{LX}. d) $-t_{RX}$.

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.

3.4.2 Example 2

In this example, 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

Table 6 – 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

considered; however, the compression point of the receiver is reduced far enough to affect the eye height. This change results in a nonlinear decrease of the received voltages. Other configurations of the channel and the input pulse are 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 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. Results are presented in Table 6, where it is observed that results of the proposed Worst-eye approach closely match results of the transient eye analysis.

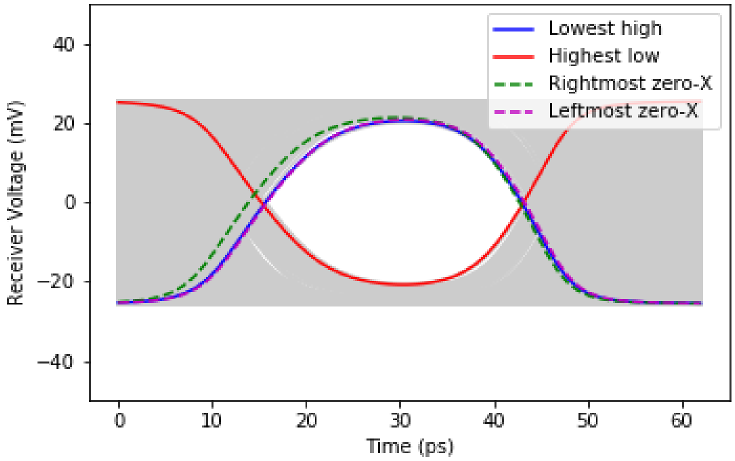
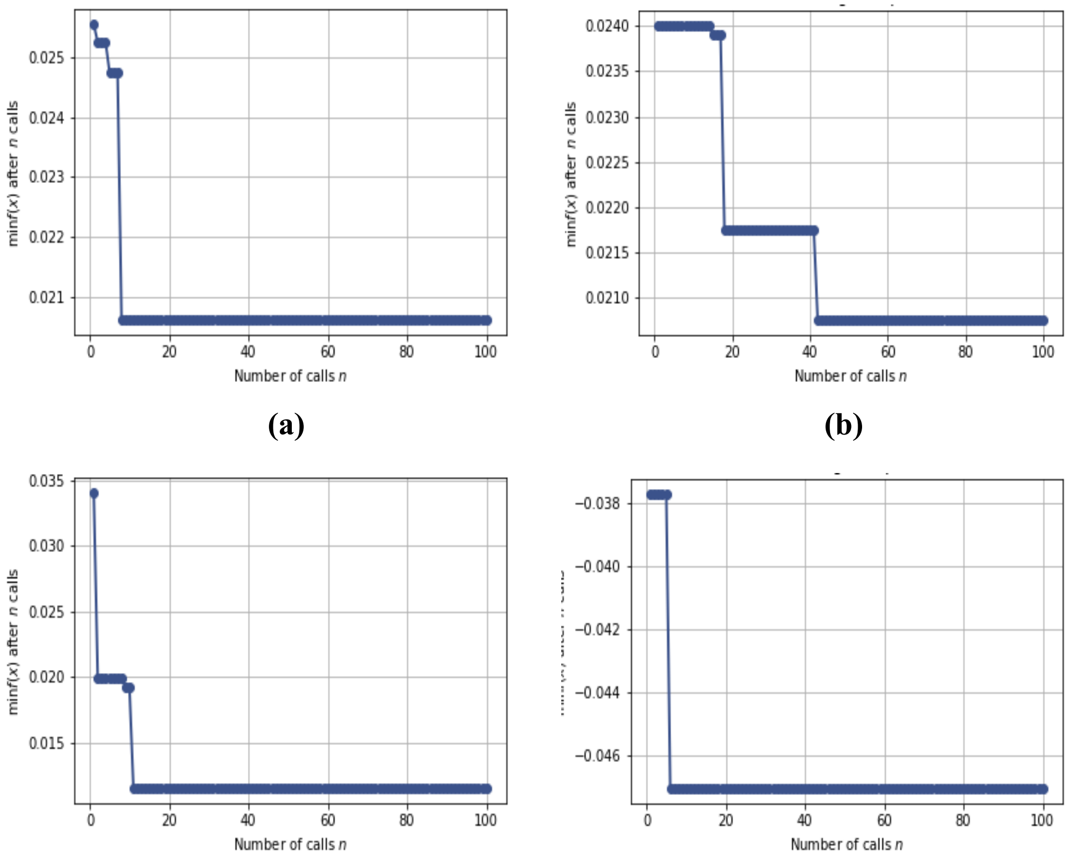


Figure 26 – Transient eye and the Worst-eye waveforms in Example 2.

Similar to Example 1, additional sampling points are not necessary since 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 Figure 26, 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 Figure 24 and Figure 26, which show the eye diagram before and after reducing the compression point.

Moreover, the convergence curves of BO of the four objective functions is presented in Figure 27, showing a fast convergence rate. By comparing Figure 25 and Figure 27 we draw the following conclusions. Figure 27 c) and d) are almost the same as Figure 25 c) and d), respectively, and they converge to the same final values. This results in equal eye widths for example 1 and 2. This observation means that the low compression

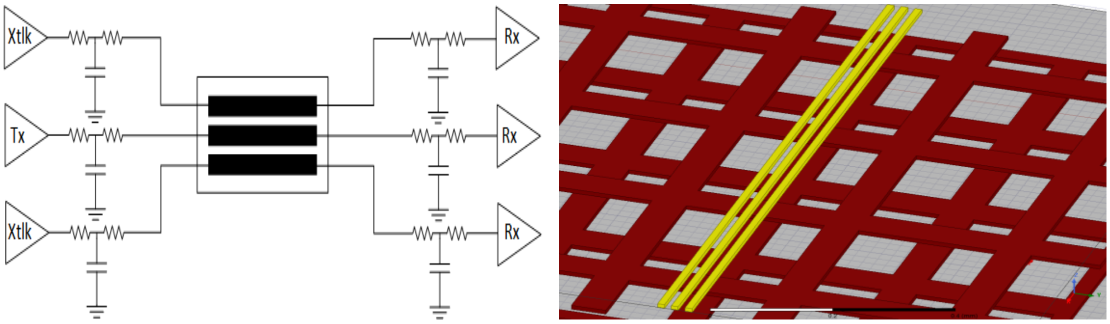


(a) **(b)**

(c) **(d)**

Figure 27 – Convergence plots of the four worst-case variables in Example 2. a) V_{LH}. b) $-V_{HL}$. c) t_{LX}. d) $-t_{RX}$.

point only affects the eye height and not the eye width, 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 Figure 27 a) and b) show faster convergence compared to Figure 25 a) and b). 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.



(a) **(b)**

Figure 28 – The high-speed channel of Example 3. a) Schematics. b) Physical design of the embedded microstrip lines.

Number of bits simulated in HSSCDR for each approach is shown in the last column of Table 6. Although number of the BO iterations and bits per sample is 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 minutes, respectively. Additionally, the overhead optimization cost of the proposed approach is about 6 minutes, leading to an overall speedup of greater than 2 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.

3.4.3 Example 3

Purpose of this example is demonstration of a higher speedup and evaluation of the proposed approach in presence of crosstalk. Therefore, the high-speed channel illustrated in Figure 28 (a) is considered. In this channel, the middle line is the victim, and the output is observed before the receiver of this line. 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 3 coupled embedded microstrip lines over two meshed PDN layers, as illustrated in Figure 28 (b). In addition, width, height, distance, and total length of the lines are 12 µm, 9 µm, 12 µm, and 19,200 µm, respectively. The structure is embedded in liquid crystal polymer (LCP), 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 0V and 0.9V, 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 bits 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 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 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

Table 7 – 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

optimization algorithm solves a three-dimensional 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 bits on the victim line and 4 bits on each aggressor line, while for t_{LX} and t_{RX} the optimization algorithm determines values of 10 bits on the victim line and 4 bits on each aggressor line. The algorithm is stopped after 200 iterations for each objective function. 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 7, where it is compared with the transient eye analysis. The results show that Worst-eye closely matches the eye height and eye width 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.

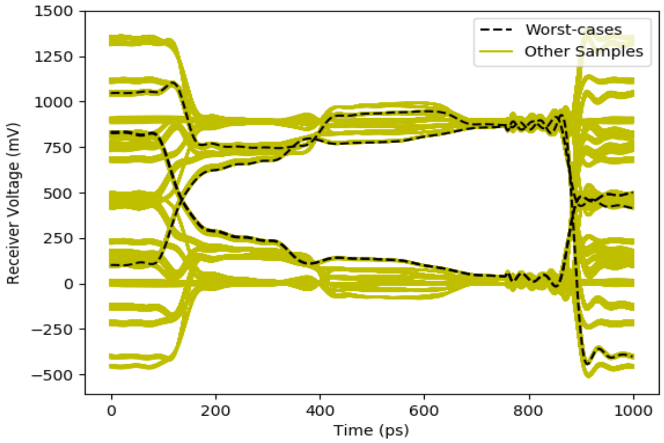


Figure 29 – Comparison of the worst-case waveforms and other waveforms found in the first round of the proposed approach in Example 3.

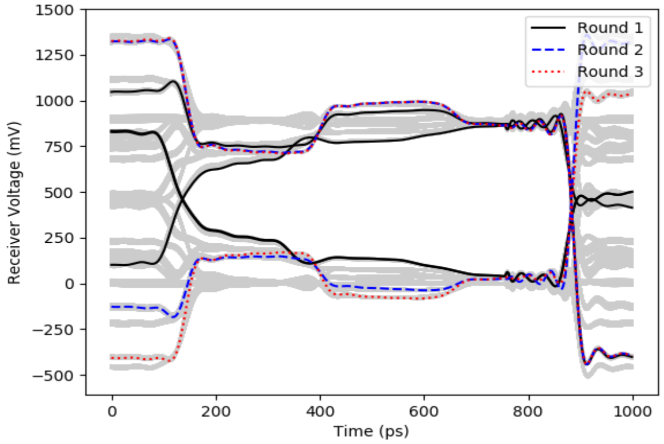
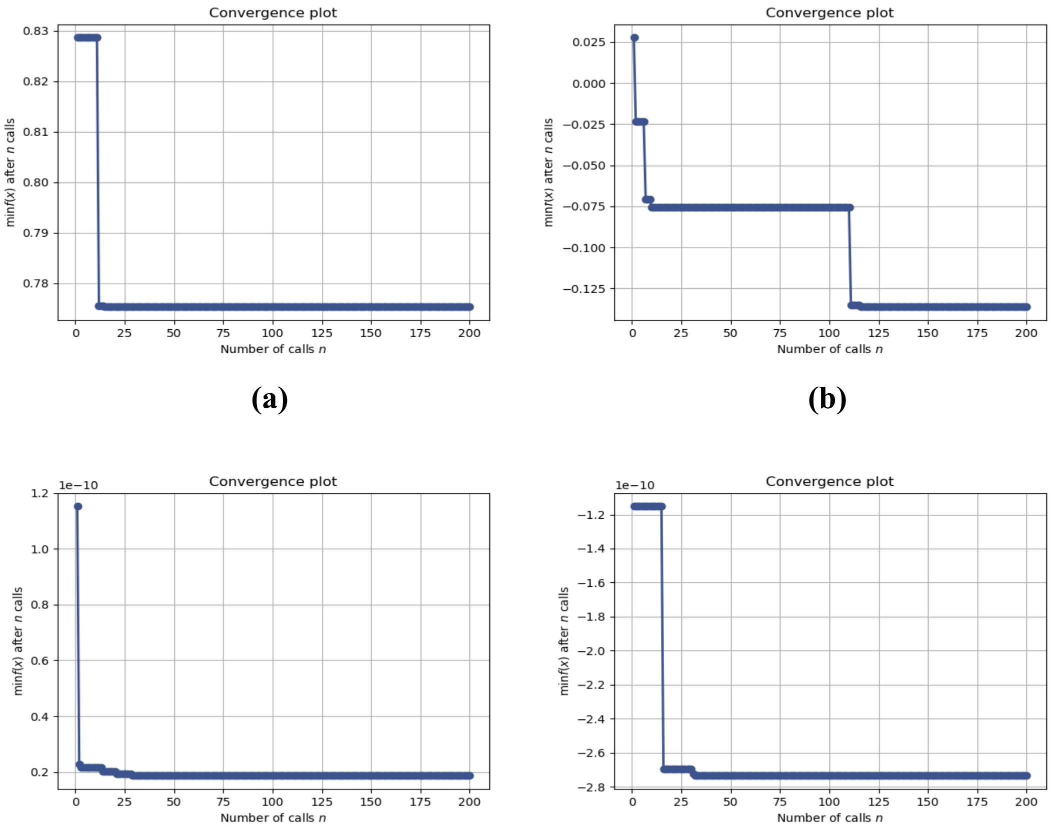


Figure 30 – Transient eye and 3 rounds of worst-case waveforms in Example 3.

This comparison is presented in Figure 29, where the worst mismatches 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 were 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 Figure 30. 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 Figure 31, showing a fast convergence rate. As presented in Table 7, number of the simulated bits in the first round of the proposed algorithm is 5,760, 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 1,356 minutes and 8 minutes, respectively. In addition, the overhead optimization cost of the first round of Worst-eye is about 21 minutes. Hence, the proposed approach provides a speedup of roughly 47 times for calculating the eye height and eye width. Furthermore, in the second round, transient simulation and overhead of Worst-eye are roughly 3 minutes and 13 minutes, respectively. In the third round, transient simulation and overhead of Worst-eye are roughly 2 minutes and 13 minutes, 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 minutes; therefore, it provides a speedup of roughly 23 times for



(a) **(b)**

(c) **(d)**

Figure 31 – Convergence plots of the four worst-case variables in Example 3. a) V_{LH}. b) $-V_{HL}$. c) t_{LX}. d) $-t_{RX}$.

finding the worst-case eye opening. We have published the work presented in this chapter in [68].

3.5 Summary

This chapter covers a novel approach to efficiently predict the worst-case eye opening based on the input waveforms, which results in the fast estimation of the eye height, eye width, and the inner eye opening. Necessary background knowledge for development of this approach, including the Gray code and Bayesian optimization, are reviewed. Afterwards, the development steps of this approach are discussed in detail. Numerical examples show that the proposed approach can provide a significant speedup compared to the traditional transient eye analysis.

CHAPTER 4. INVERTIBLE NEURAL NETWORKS FOR INVERSE DESIGN OF CTLE

4.1 Objectives

Finding the parameters that shape the frequency response of the continuous time linear equalization (CTLE) can be challenging and time-consuming. Therefore, in this chapter a novel approach for inverse design of CTLE with the invertible neural network (INN) is proposed. In this inverse approach, the desired eye height and eye width are given, and the algorithm derives the corresponding design parameters. This chapter starts with a brief description of the INN. Next, the CTLE circuit is reviewed. Then, inverse design of CTLE with INN is presented. Finally, a numerical example for evaluation of this approach is presented. The proposed approach is a preliminary work, and further improvements and modifications are encouraged.

4.2 Invertible Neural Networks

4.2.1 INN Structure

In the proposed inverse design approach INNs [23] are used. The general INN network is illustrated in Figure 32, which shows it is comprised of several reversible blocks. This reversible structure permits bidirectional training of the network. In this figure, **X** shows the input parameters, and **Y** is the output. **Z** is a set of latent variables with normal distribution that do not exist in the actual system. These variables are added in the output

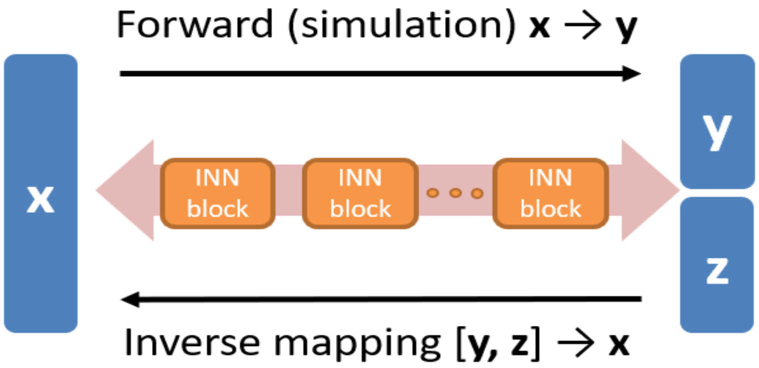


Figure 32 – Structure of the invertible neural networks.

to store the lost information in the forward mapping from **X** to **Y**. In the training process, a supervised loss function, such as the mean square error, is used for **Y** since it represents deterministic variables. On the other hand, because **X** and **Z** are stochastic variables and represent distributions, the maximum mean discrepancy (MMD) is used as their loss function. Here, MMD serves as an unsupervised loss function, and it only needs samples from two distributions to compare them. This function is discussed in the next section. Relative weights of the loss functions in INN are tuned as hyperparameters in order to make their effect about equal.

4.2.2 Maximum Mean Discrepancy

The *Two-Sample Test* in statistics is defined as follows. Given a random variable x with probability distribution p, and a random variable y with probability distribution q, it is intended to find if $p \neq q$, only from observations $X \coloneqq \{x_1, \dots, x_m\}$ and $Y \coloneqq \{y_1, \dots, y_n\}$, which are independently and identically distributed (i.i.d) from p and q, respectively. The maximum mean discrepancy is a method to conduct this test. MMD can be formulated as:

$$MMD^2(p,q) = \boldsymbol{E}_{x,x'}[k(x,x')] - 2\boldsymbol{E}_{x,y}[k(x,y)] + \boldsymbol{E}_{y,y'}[k(y,y')], \tag{36}$$

where x' and y' are independent copies of x and y, respectively, with the same distributions. A larger MMD means a larger difference between p and q [69], [70]. Moreover, k is a kernel function required for this method. When using MMD as a loss function for INN, the Inverse Multiquadratic function is selected as the kernel, which is defined as [23]:

$$k(x,x') = \frac{1}{1 + \|(x-x')/h\|_2^2}. \tag{37}$$

4.2.3 Reversible Blocks in INN

Inner structure of the reversible blocks in INN is illustrated in Figure 33 (inspired by [71]). Each block needs an even and equal number of inputs and outputs, which is enforced with zero padding if needed. Inputs and outputs are divided into two halves, which are shown as $[\boldsymbol{X_1}, \boldsymbol{X_2}]$ and $[\boldsymbol{Y_1}, \boldsymbol{Y_2}]$, respectively. The forward path of each block is equivalent to:

$$\boldsymbol{y_1} = \boldsymbol{x_1} \,.\, exp\big(S_2(\boldsymbol{x_2})\big) + t_2(\boldsymbol{x_2}), \tag{38}$$

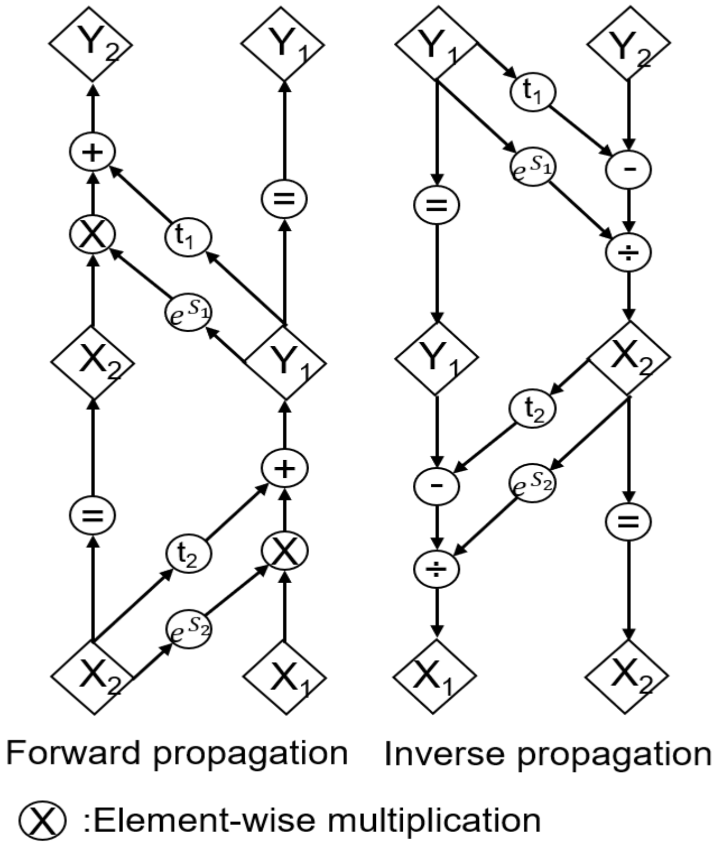


Figure 33 – Structure of a single INN block.

$$\boldsymbol{y_2} = \boldsymbol{x_2} \, . \; exp\big(S_1(\boldsymbol{y_1})\big) + t_1(\boldsymbol{y_1}).$$

And the reverse path of each block is equivalent to:

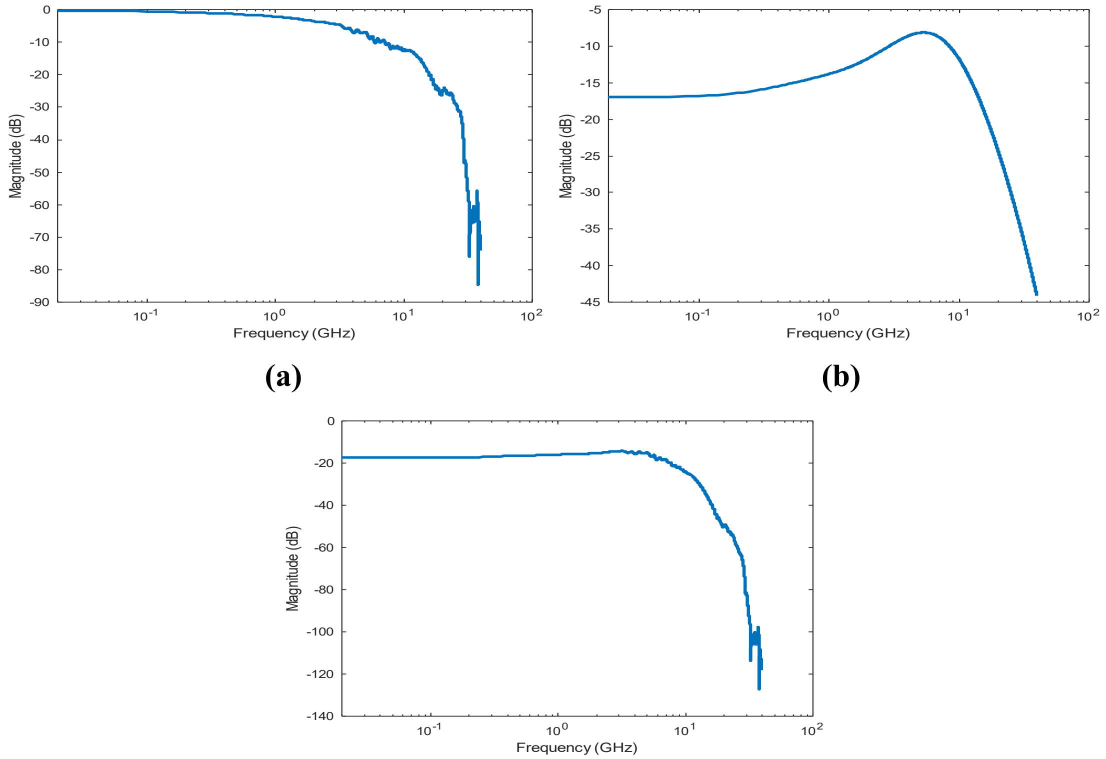
$$\begin{aligned} \boldsymbol{x_2} &= \big(\boldsymbol{y_2} - t_1(\boldsymbol{y_1})\big) \, . \, exp\big(-S_1(\boldsymbol{y_1})\big), \\ \boldsymbol{x_1} &= \big(\boldsymbol{y_1} - t_2(\boldsymbol{x_2})\big) \, . \, exp\big(-S_2(\boldsymbol{x_2})\big), \end{aligned} \tag{39}$$

where, $S_1, S_2, t_1,$ and t_2 are neural networks themselves. Note that although these subnetworks are not invertible, (38) and (39) are always invertible. The individual blocks are connected with shuffling layers. For additional details refer to [23].

4.3 Continuous Time Linear Equalizer

The high-speed channels have a limited bandwidth, which causes a higher attenuation of the signal at high frequencies. In other words, the channel behaves as a low pass filter making the data pattern lose its sharp corners. Thus, the transitions take longer to complete, which results in a higher ISI, closer eye diagram, and subsequently a higher BER. Therefore, equalization techniques have been developed to compensate the low-pass filtering effect of the channel. A common equalization circuit used in high-speed channels is the continuous time linear equalizer (CTLE) [72], [73]. CTLE's frequency response has a peaking at high frequencies. This results in boosting the overall frequency response of the channel at high frequencies when CTLE is connected in series at the receiver side. Advantages of CTLE includes its relatively low power, small area, and low complexity.

As an example, frequency response of a sample channel is shown in Figure 34 (a), which shows the low pass behavior of the channel. In addition, frequency response of a sample CTLE is presented in Figure 34 (b) showing the peaking at high frequencies. These frequency responses are derived from realistic component models provided by IBM, which are used in their custom-build channel simulator called HSSCDR [50]. The frequency response of cascaded channel and CTLE is found by calculating product of their frequency



(a) **(b)**

(c)

Figure 34 – Frequency responses of a sample channel and CTLE. a) Channel. b) CTLE. c) Cascaded channel and CTLE.

responses, which is shown in Figure 34 (c). It is observed in this figure that CTLE flattens response of the channel and increases its bandwidth, resulting in an equalized channel.

Transfer function of a simple channel can be approximated with one pole as:

$$H_{ch}(s) = \frac{k_{ch}}{s + p_{ch}}, \tag{40}$$

where p_{ch} is the dominant pole, and k_{ch} is a constant. Moreover, transfer function of a simple CTLE is described as:

$$H_{CTLE}(s) = \frac{k_{CTLE}(s + z_1)}{(s + p_1)(s + p_2)}, \tag{41}$$

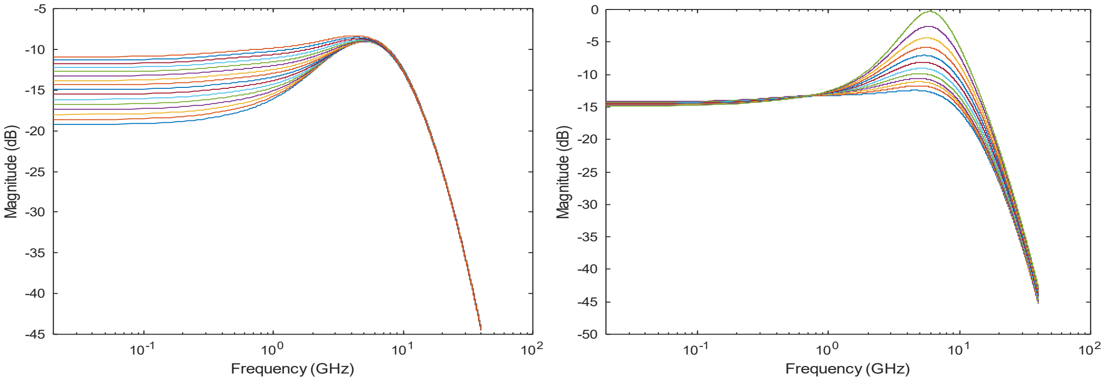
where $p_1$and p_2 are the poles. $z_1$is the single zero of CTLE, and k_{CTLE} is a constant. An equalized channel can be achieved by matching p_{ch} and $z_1$since they cancel one another and the resulting transfer function would be:

$$H_{eq}(s) = \frac{k_{eq}}{(s + p_1)(s + p_2)}, \tag{42}$$

where $k_{eq} = k_{ch}k_1$. Note that the bandwidth can be increased by setting the first pole of CTLE higher than the dominant pole of the channel [73].

4.4 Inverse Design of CTLE

The CTLE implementations in this work are characterized by a DC gain value and peaking parameter instead of poles and zeros. Although gain and peaking are continuous variables, in reality a limited number of CTLE hardware designs are available to the designer. Therefore, the possible DC gain and peaking values are discrete, and we show them as $[p_1, p_2, \ldots p_N]$ and $[g_1, g_2, \ldots g_M]$, respectively, where, *N* and *M* are the number of possible implementations. In total we have the $N * M$ combinations for the CTLE design. For illustration, a set of IBM's CTLE frequency responses are shown in Figure 35. In Figure 35 (a), the peaking is kept constant and the DC gain is varied. However, in Figure 35 (b), the DC gain is kept constant and the peaking is varied.



(a) **(b)**

Figure 35 – Variations of CTLE by DC gain and peaking. a) Constant peaking and variable DC gain. b) Constant DC gain and variable peaking.

Goal of the proposed approach is to derive the suitable p_i and g_j for the desired eye height (EH) and eye width (EW) values, using the INN. The inverse problem is equivalent to:

$$[p_i, g_j] = f^{-1}(EH, EW, \boldsymbol{Z}), \tag{43}$$

where, $\boldsymbol{Z}$ is the latent variables, and f^{-1} is the inverse mapping, which is found by training the INN. After training the network, (43) is evaluated numerously to derive the joint distribution of p_i and g_j. Note that in these evaluations *EH* and *EW* are fixed while $\boldsymbol{Z}$ is sampled from its normal distribution. Afterwards, the closest available input parameters to the most likely point from distribution of p_i and g_j is selected as peaking and DC gain values. If the distribution is multi-modal, multiple candidate designs are produced. Finally, the eye diagram is simulated for the design(s) to evaluate the resulting eye height and eye width, and compare with the target values. The proposed approach is implemented in

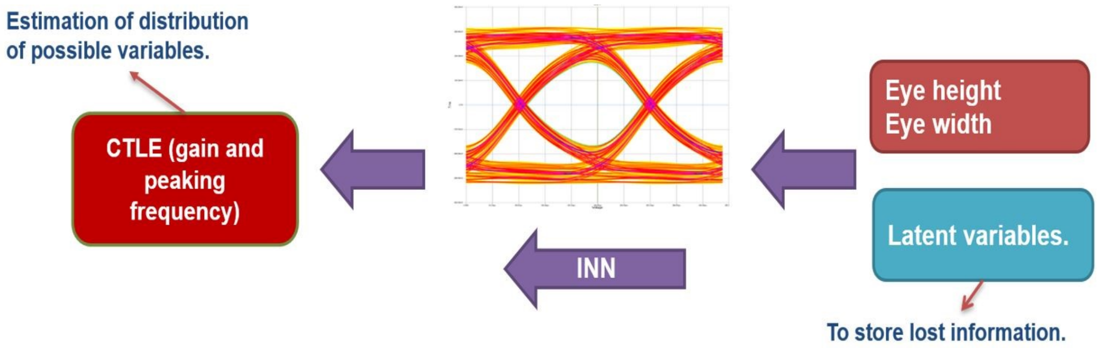


Figure 36 – Flow of the algorithm for inverse design of CTLE.

Python 3.7 using the INN source code published in [74]. Flow of the inverse algorithm is represented in Figure 36.

4.5 Numerical Example

To investigate effectiveness of the proposed approach for inverse design of CTLE, once again the SerDes channel, used in the previous chapters and pictured in Figure 37, is considered. To recap, this channel includes two processor packages, connected to the board with hybrid land grid array connectors. The board contains 4 inches of differential wiring in total, which is connected to the connectors with differential vias. No crosstalk is considered. This channel is simulated with HSSCDR. Furthermore, the channel operates at 16 Gb/s; resulting in a unit interval of 62.5 ps. For the CTLE design, 10 possible peaking parameters and 16 possible DC gain values are considered. These values are shown symbolically as $[p_1, p_2, \ldots p_{10}]$ and $[g_1, g_2, \ldots g_{16}]$, which are ordered sequentially. Frequency response of the channel is the one shown in Figure 34 (a), and some of the CTLE responses are shown in Figure 35.

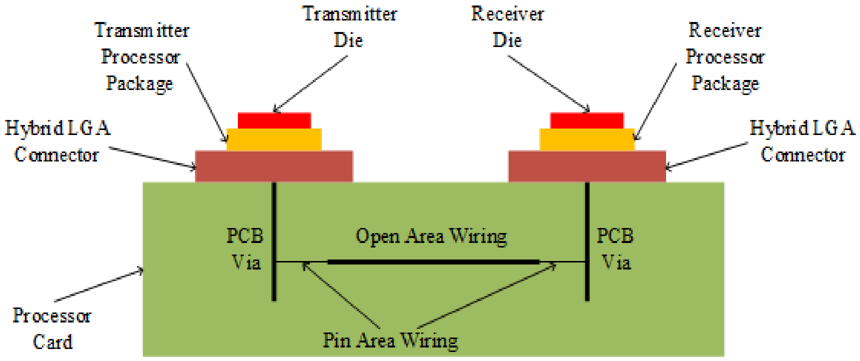


Figure 37 – High-speed SerDes channel in the numerical example.

As a rule of thumb, often about 80% of the total number of samples is selected for training and validation, which is subsequently divided to 80% and 20% sections for training and validation, respectively. The remaining 20% of the total number of samples is used for testing. Therefore, from the 160 combinations of p_i and g_j, 102 samples are randomly selected for training the network. Another 25 random samples are used for validation and tuning the hyperparameters of the network. After tuning, number of the latent variables in $\boldsymbol{Z}$ is set to 2. INN is comprised of 4 reversible blocks. S_1, S_2, t_1, and t_2 are fully connected neural networks, and each of them has 1 hidden layer with 100 nodes and the ReLU activation function. Number of dimensions in the input and output is increased to 16 with zero-padding, and the training takes 200 epochs.

The remaining 33 samples are used for testing. The resulting EH and EW values, reported by the inverse design, show a wide range of accuracy. For some test samples a

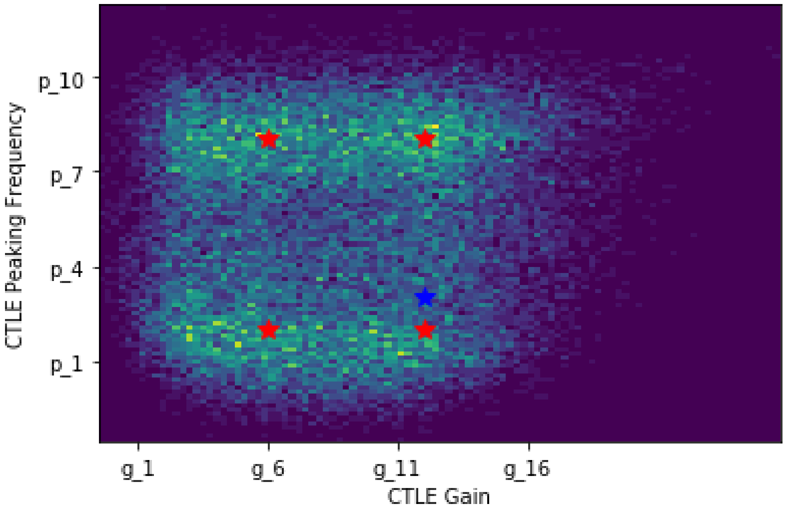
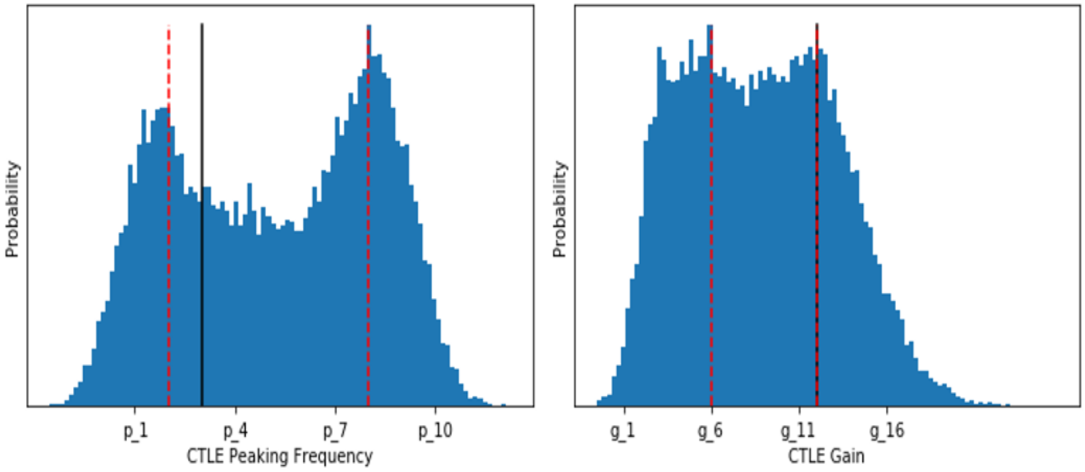


Figure 38 – Joint distribution of CTLE peaking (p_i) and CTLE DC gain (g_j) in the numerical example. Candidate designs are shown with red stars and the accurate design is shown with a blue star.



(a) **(b)**

Figure 39 – Marginal distributions of the CTLE parameters in the numerical example. Candidate designs are shown with red dashed lines and the accurate design is shown with a black line. a) CTLE peaking (p_i). b) CTLE DC gain (g_j).

good match with the desired EH and EW is achieved. While, for some others the results

Table 8 – Accurate and candidate CTLE designs and their corresponding EH and EW.

Design	Peaking parameter	Gain value	EH (mV)	EW (ps)
Accurate	p_3	g_{12}	175	44.6
INN1	p_2	g_6	186	49.9
INN2	p_2	g_{12}	153	42.4
INN3	p_8	g_6	235	37.1
INN4	p_8	g_{12}	292	44.3

were not satisfactory. Next, we show the results for a case with moderate accuracy, and discuss the other test cases afterwards. In this test case the desired EH and EW are 175 mV and 44.6 ps, respectively. We sampled (43) for 30,000 times to derive distribution of p_i and g_j. Note that this evaluation is almost instant because INN translates to an analytical calculation. Joint distribution of p_i and g_j, and their marginal distributions are shown in Figure 38 and Figure 39, respectively. It is seen that the distribution is multi-modal. In other words, the proposed approach suggests four possible solutions which are $(p_2, g_6), (p_2, g_{12}), (p_8, g_6),$ and (p_8, g_{12}). These solutions and their corresponding EH and EW values are presented in Table 8. In addition, the accurate selection of p_i and g_j for the desired EH and EW is included in Figure 38, Figure 39, and Table 8. We know the accurate p_i and g_j since we had swept over all of their possible values. From the results in Table 8 it is observed that the INN approach has achieved a design (INN2) which is only one step

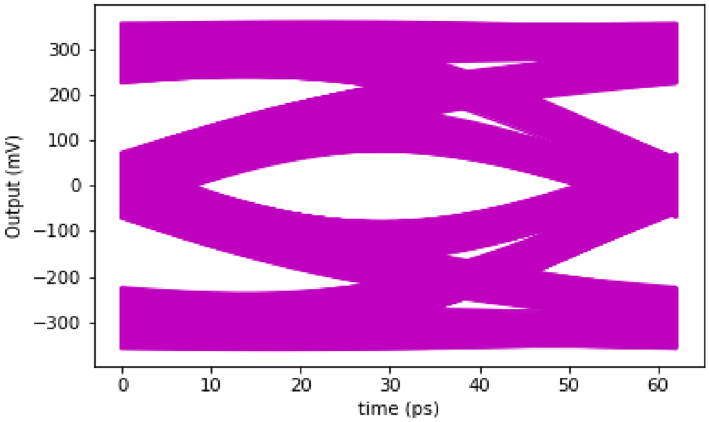


Figure 40 – Eye diagram of the channel in the numerical example when the INN2 design is used for CTLE.

away from the accurate design in the peaking parameter, and it has the same DC gain value. EH and EW of this design are close to the target values. On the other hand, results of the INN1 design are also close to the target values, while its gain is not close to the accurate gain. This design can be selected if INN2 is not possible due to other constraints, and it shows that the INN approach can find multiple solutions for a single target objective. The eye diagram obtained from the INN2 design is illustrated in Figure 40.

Although the results achieved in this test case are close to the target values, they are not a perfect match. In the 33 test cases, more accurate results were observed; however, some other test cases had a higher mismatch rate, including some unacceptable results. Overall, we conclude that the INN structure is not a universal solution in its current state and needs improvements. One of the issues that can cause the mismatch is handling of discrete variables. The proposed approach derives the CTLE variables by selecting the closest possible values to the candidate points taken from the distribution provided by INN. However, it is seen in Table 8 that even one step mismatch in the peaking parameter can result in nontrivial mismatch with the target values. Addressing this issues is left for future work. We have published the work presented in this chapter in [75].

4.6 Summary

This chapter covers a novel inverse design approach to find the possible CTLE parameters from the given eye height and eye width to significantly reduce the design time. This approach is based on an invertible neural network (INN) structure. Initially the INN structure, and the traditional CTLE design are discussed. Afterwards, development of the proposed approach is described. Finally, a numerical example is provided showing the application of the proposed approach. Promising results are achieved in this preliminary work, and further improvements are left for future endeavors.

CHAPTER 5. CONCLUSION

5.1 Discussion

With the rapid increase in the bandwidth and complexity of high-speed serial channels, their design and analysis has become more challenging and time consuming. On the other hand, recent advancements in machine learning and numerical methods has shown a great potential to be used in design and analysis of high-speed electronics. Therefore, in this dissertation novel methods, based on these advancements, are introduced to accelerate design and analysis of the high-speed channels.

Our focus is mainly on the eye-diagram analysis of high-speed channels with an NRZ pulse sequence. The first analysis approach is a methodology for estimation of data dependent jitter and the corresponding eye diagram. Using a short transient simulation, the proposed approach trains surrogate models to efficiently estimate jitter, eye diagram, and statistics of the output signal. The surrogate models are trained with a Polynomial Chaos method to model behavior of the system. Moreover, the expansion is altered based on memory of the channel to either increase the accuracy or decrease the computation costs. Finally, numerical examples are provided, showing a good match between results of the proposed approach and the traditional transient eye simulation, with savings in memory and CPU time. In the examples up to 2.4 times speedup for the full eye diagram and up to 11.5 times speedup for the jitter distribution were observed.

Next, an optimization-based algorithm for quick evaluation of eye height, eye width, and the inner eye opening (worst-case eye) is suggested. This approach, dubbed

Worst-eye, focuses on DDJ, DDN, ISI, and crosstalk. Worst-eye 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, eye height and eye width 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, Bayesian optimization 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, and a channel in a system-on-package design. Numerical results show that the proposed approach can accurately find the eye width and eye height with up to 47 times speedup, and the worst-case eye opening with up to 23 times speedup, when compared with the transient eye.

The final section of this dissertation is focused on the design part, where an approach for inverse design of CTLE of high-speed channels is proposed in order to decrease the time taken for design of such system. The algorithm receives the desired eye height and eye width, and it derives the required peaking and DC gain of CTLE. This approach is based on invertible neural networks, which can be trained and used in both directions. An example with moderate accuracy is provided. However, it is observed that the algorithm can produce inaccurate results in some other test cases. Therefore, improvements to the algorithm are needed.

5.2 Future Work

The eye analysis methods, proposed in this dissertation, have focused on DDJ, DDN and ISI. Also, the data dependent crosstalk is included in the Worst-eye approach. Thus, the next logical step would be inclusion of other sources of noise and jitter, including RJ, RN, and DCD. Subsequently, the final BER and the bathtub curve could be derived. Although, similar to some other eye analysis methods we have suggested integration of RJ and RN in post analysis, it is not completely clear how the nonlinear systems would affect the impact of RJ and RN. Therefore, further studies on these variables is suggested.

For the PC surrogate modeling approach, inclusion of crosstalk can be included in future work. Similar to Worst-eye, symbols on the aggressor lines can be included in the PC models as additional random variables. However, this approach might prohibitively increase the computational costs due to the curse of dimensionality. Therefore, decoupling the transmission lines might be a better approach. Regarding the Worst-eye, improvements on BO for discrete variables is suggested. In this work, we settled for the practice of rounding the next BO sample to the nearest integer number since satisfactory results were achieved. However, a better convergence might be achieved with more advanced discrete BO as suggested in [62].

As discussed in CHAPTER 4, the CTLE design with INN does not always provide accurate results. Therefore, improvements on this approach are needed. One possible way can be involving additional eye measurements such as jitter RMS, noise RMS, average rise/fall time, etc. on the output side of INN to preserve more of the lost information in the forward mapping. However, distribution of these variables would be considered instead of their deterministic values, similar to the latent variables. Furthermore, as mentioned at the end of CHAPTER 4 a better handling of the discrete design parameters needs to be developed since the INN structure is originally developed for continuous variables. Moreover, in this work we performed inverse design for target values that have at least one existing solution. If the solution does not exist, the algorithm needs to provide the solution resulting in the closest possible response. Moreover, for this study it would be interesting to examine lossier channels as well. It would be interesting to see how the inverse CTLE design works for a lossy channel which has a closed eye without equalization. These channels should also be examined to see if the multi-modal behavior of the inverse design persists for them.

5.3 State of ML for electronics design and analysis

Similar to this work, in recent years many studies have been performed to develop machine learning (ML) methods for design and analysis of high-speed electronics. Many of these studies have been referenced in this dissertation. In general, this is a promising field, which seems to continue to provide more opportunities in the future. Although, many ML algorithms have been around for many years, and we had seen excitements and expectations around them, the situation seems to be different this time for applications in electronics design and analysis due to the following reasons. First, the computing power has significantly increased, which makes application of ML algorithms possible when working with costly EM simulations. Moreover, open-source ML algorithms and codes are now abundant and easily accessible, which has encouraged many engineers to start working in this area. Finally, for many applications, the amount of data has significantly increased, making application of more data-hungry ML algorithms possible. Nevertheless, the industry still seems reluctant to adapt these new algorithms, which can be due to the fundamental differences with the stablished algorithms and in some cases reliability of the ML algorithms. However, this indicates that there are more opportunities for research in this developing field to make it more reliable for commercial use.

It also should be noted that, due to the abundance and accessibility of ML algorithms, some of new research papers have been blindly applying off-the-shelf algorithms to electronics design and analysis problems. Of course, this approach is not helpful, and the researcher needs to understand which algorithms are more suitable for their applications. In other words, domain knowledge and modifications to the existing algorithms is necessary.

5.4 List of Publications

Journal Papers:

- **M. Ahadi Dolatsara**, J. Hejase, W. Becker, M. Swaminathan, “Methods for Jitter and Eye Diagram Estimation in High-Speed Serial Channels Using Polynomial Chaos Surrogate Models”, *IEEE ACCESS*, April 2019.

- **M. Ahadi Dolatsara**, J. Hejase, W. Becker, J. Kim, S. Lim, M. Swaminathan, "Worst-case Eye Analysis of High-speed Channels Based on Bayesian Optimization," *IEEE Transactions on Electromagnetic Compatibility*, 2020.
- H. Park, J. Kim, V. C. K. Chekuri, **M. Ahadi Dolatsara**, M. Nabeel, A. Bojesomo, S. Patnaik, O. Sinanoglu, M. Swaminathan, S. Mukhopadhyay, J. Knechtel, S. K. Lim, **"**Design Flow for Active Interposer-Based 2.5D ICs and Study of RISC-V Architecture with Secure NoC," *IEEE Transactions on Components, Packaging and Manufacturing Technology*, vol. 10, no. 12, pp. 2047-2060, 2020.
- G. Murali, H. Park, E. Qin, H. Torun, **M. Ahadi Dolatsara**, M. Swaminathan, T. Krishna, S. K. Lim, "Clock Delivery Network Design and Analysis for Interposer-Based 2.5-D Heterogeneous Systems", *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 29, no. 4, pp. 605-616, 2021.

Conference Papers:

- **M. Ahadi Dolatsara**, J. Hejase, W. Becker, M. Swaminathan "Development of Polynomial Chaos based Surrogate Models for Channel Simulation", *IEEE Conference on Electromagnetic Compatibility, Signal & Power Integrity EMC+SIPI,* July 2018.
- **M. Ahadi Dolatsara**, H. Yu, J. Hejase, W. Becker, M. Swaminathan, "Polynomial Chaos modeling for jitter estimation in high-speed links", International Test Conference, October 2018.

- **M. Ahadi Dolatsara**, J. Hejase, W. Becker, M. Swaminathan "Jitter and Eye Estimation in SerDes Channels using Modified Polynomial Chaos Surrogate Models", *IEEE Conference on Electrical Performance of Electronic Packaging and Systems,* October 2018.
- **M. Ahadi Dolatsara**, A. Varma, K. Keshavan, M. Swaminathan, "Design Space Exploration with Polynomial Chaos Surrogate Models for Analyzing Large System Designs", *DesignCon 2019*, Jan. 2019.
- **M. Ahadi Dolatsara**, J. Hejase, W. Becker, M. Swaminathan, "Eye Diagram and Jitter Estimation in SerDes Designs using Surrogate Models Generated with the Polynomial Chaos Theory", *DesignCon 2019*, Jan. 2019.
- **M. Ahadi Dolatsara**, A. Varma, K. Keshavan, M. Swaminathan, "A Modified Polynomial Chaos Modeling Approach for Uncertainty Quantification*", International Applied Computational Electromagnetics Society (ACES) Symposium*, April 2019.
- K. Roy, **M. Ahadi Dolatsara**, H. Torun, R. Trinchero, M. Swaminathan, "Inverse Design of Transmission Lines with Deep Learning", *IEEE Conference on Electrical Performance of Electronic Packaging and Systems,* October 2019.
- R. Trinchero, **M. Ahadi Dolatsara**, K. Roy, M. Swaminathan, F. G. Canavero, "Design of High-Speed Links via a Machine Learning Surrogate Model for the Inverse Problem", *IEEE Electrical Design of Advanced Packaging and Systems (EDAPS)*, Kaohsiung, Taiwan, 2019.

- **M. Ahadi Dolatsara**, M. Swaminathan, “Determining worst-case eye height in low BER channels using Bayesian optimization.”, LASCAS 2020 - *11th IEEE Latin American Symposium on Circuits and Systems*, San José, Costa Rica, 2020.
- J. Kim, V. C. K. Chekuri, N. M. Rahman, **M. Ahadi Dolatsara**, H. Torun, M. Swaminathan, S. Mukhopadhyay, S. K. Lim, "Silicon vs. Organic Interposer: PPA and Reliability Tradeoffs in Heterogeneous 2.5D Chiplet Integration*," IEEE International Conference on Computer Design (ICCD*), Oct. 2020
- **M. Ahadi Dolatsara**, H. Yu, J. Hejase, W. Becker, M. Swaminathan, “Invertible Neural Networks for Inverse Design of CTLE in High-speed Channels”, *IEEE Electrical Design of Advanced Packaging and Systems (EDAPS)*, Virtual, 2020.

REFERENCES

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

[2] B. K. Casper, M. Haycock and R. Mooney, "An accurate and efficient analysis method for multi-Gb/s chip-to-chip signaling schemes," in *IEEE Symp. Very Large Scale Integr. (VLSI) Circuits*, 2002.

[3] A. Sanders, M. Resoo and D. Ambrosia, "Channel Compliance Testing Using Novel Statistical Eye Methodology," in *DesignCon*, Santa Clara, CA, 2004.

[4] P. Manfredi, *High-speed interconnect models with stochastic parameter variability,* Turin, Italy: Ph.D. dissertation, Informat. Comm. Tech., Politecnico di Torino, 2013.

[5] D. Xiu, "Fast numerical methods for stochastic computations: A review," *Commun. Comput. Phys.,* vol. 5, no. 2-4, pp. 242-272, 2009.

[6] D. Spina, F. Ferranti, G. Antonini, T. Dhaene and L. Knockaert, "Non intrusive polynomial chaos-based stochastic macromodeling of multiport systems," in *IEEE 18th Workshop Signal Power Integrity*, 2014.

[7] D. Spina and e. al., "Stochastic macromodeling of nonlinear systems via poly-nomial chaos expansion and transfer function trajectories," *IEEE Trans. Microw. Theory Tech.,* vol. 62, no. 7, pp. 1454-1460, 2014.

[8] P. Manfredi and F. G. Canavero, "`Efficient statistical simulation of microwave devices via stochastic testing-based circuit equivalents of nonlinear components," *IEEE Trans. Microw. Theory Techn.,* vol. 63, no. 5, pp. 1502-1511, 2015.

[9] M. Ahadi, K. Prasad and S. Roy, "Hyperbolic polynomial chaos expansion (HPCE) and its application to statistical analysis of nonlinear circuits," in *IEEE 20th Workshop Signal Power Integrity (SPI)*, May, 2016.

[10] D. Xiu and 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.

[11] H. S. Wilf, Combinatorial algorithms: an update. Vol. 55., Philadelphia, PA : SIAM, 1989.

[12] 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," *arXiv preprint,* arXiv:1012.2599, 2010.

[13] S. J. Park, B. Bae, J. Kim and M. Swaminathan, "Application of machine learning for optimization of 3-D integrated circuits and systems," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems,* vol. 25, no. 6, pp. 1856-1865, 2017.

[14] 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 Transactions on Very Large Scale Integration (VLSI) Systems,* vol. 26, no. 4, p. 792.802, 2018.

[15] H. M. Torun and M. Swaminathan, "High-Dimensional Global Optimization Method for High-Frequency Electronic Design," *IEEE Transactions on Microwave Theory and Techniques,* vol. 67, no. 6, pp. 2128-2141, 2019.

[16] 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 *2018 IEEE 27th Conference on Electrical Performance of Electronic Packaging and Systems (EPEPS)*, San Jose, 2018.

[17] M. Ohira, A. Yamashita, Z. Ma and X. Wang, "Automated Microstrip Bandpass Filter Design Using Feedforward and Inverse Models of Neural Network," in *Asia-Pacific Microwave Conference (APMC)*, 2018.

[18] D. Zibar, A. M. R. Brusin, U. C. Moura, F. D. Ros, V. Curri and A. Carena, "Inverse System Design Using Machine Learning: The Raman Amplifier Case," *Journal of Lightwave Technology,* vol. 38, no. 4, pp. 736-753, 2020.

[19] H. Ma, E. Li, A. C. Cangellaris and X. Chen, "High-Speed Link Design Optimization Using Machine Learning SVR-AS Method," in *IEEE 29th Conference on Electrical Performance of Electronic Packaging and Systems (EPEPS)*, 2020.

[20] K. Roy, M. Ahadi Dolatsara, H. M. Torun, R. Trinchero and M. Swaminathan, "Inverse Design of Transmission Lines with Deep Learning," in *IEEE 28th Conference on Electrical Performance of Electronic Packaging and Systems (EPEPS)*, 2019.

[21] R. Trinchero, M. Ahadi Dolatsara, K. Roy, M. Swaminathan and F. G. Canavero, "Design of high-speed links via a machine learning surrogate model for the inverse problem," in *Electrical Design of Advanced Packaging and Systems (EDAPS)*, 2019.

[22] H. Yu, H. M. Torun, M. U. Rehman and M. Swaminathan, "Design of SIW Filters in D-band Using Invertible Neural Nets," in *IEEE/MTT-S International Microwave Symposium (IMS)*, 2020.

[23] L. Ardizzone, J. Kruse, S. Wirkert, D. Rahner, E. W. Pellegrini, R. S. Klessen, L. Maier-Hein, C. Rother and U. Köthe, "Analyzing Inverse Problems with Invertible Neural Networks," in *International Conference on Learning Representations*, 2019.

[24] J. G. Proakis, Digital Communication, 4th ed., Singapore: McGraw-Hill, 2001, pp. 617-618.

[25] 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.

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

[27] 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 *18th IEEE Conference on Electrical Performance of Electronic Packaging and Systems*, 2009.

[28] W. Yao, Y. Shi, L. He, S. Pamarti and Y. Hu, "Worst case timing jitter and amplitude noise in differential signaling," in *IEEE 10th International Symposium on Quality Electronic Design*, 2009.

[29] J. F. Buckwalter, "Predicting microwave digital signal integrity," *IEEE Transactions on Advanced Packaging,* vol. 32, no. 2, pp. 280-289, 2009.

[30] D. Kim, H. Kim and Y. Eo, "Efficient eye diagram determination of strongly coupled lines for differential signals," in *IEEE 2010 International SoC Design Conf.*, 2010.

[31] J. Lee and Y. Eo, "An efficient eye-diagram determination technique for multi-coupled interconnect lines," in *IEEE 23rd International Workshop on Power and Timing Modeling, Optimization and Simulation (PATMOS)*, 2013.

[32] W. D. Guo, J. H. Lin, C. M. Lin, T. W. Huang and R. B. Wu, "Fast methodology for determining eye diagram characteristics of lossy transmission lines," *IEEE Transactions on Advanced Packaging,* vol. 32, no. 1, pp. 175-183, 2009.

[33] H. Kim and e. al., "A fast and accurate statistical eye-diagram estimation method for high-speed channel including non-linear receiver buffer circuit," in *IEEE Asia-Pacific Symposium on Electromagnetic Compatibility (APEMC)*, 2015.

[34] 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.

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

[36] T. Nguyen and e. al., "Transient simulation for high-speed channels with recurrent neural network," in *27th IEEE Conf. Elect. Perform. Electron. Packag. Syst.*, 2018.

[37] 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 *IEEE 59th Electronic Components and Technology Conference*, 2009.

[38] 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 *IEEE 62nd Electronic Components and Technology Conference*, 2012.

[39] 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 *IEEE Electrical Performance of Electrical Packaging*, 2003.

[40] N. Singh, B. Mutnury, N. Pham, M. Cases and C. Wesley, "Bit-pattern optimization for accurate analysis of complex high-speed interfaces," in *IEEE 58th Electronic Components and Technology Conference*, 2008.

[41] S. N. Ahmadyan, C. Gu, S. Natarajan, E. Chiprout and S. Vasudevan, "Fast eye diagram analysis for high-speed CMOS circuits," in *EDA Consortium Design, Automation & Test in Europe Conference & Exhibition*, 2015.

[42] I. Elshafiey, L. Udpa and S. S. Udpa, "Application of neural networks to inverse problems in electromagnetics," *IEEE transactions on magnetics,* vol. 30, no. 5, pp. 3629-3632, 1994.

[43] D. Cherubini, A. Fanni, A. Montisci and P. Testoni, "Inversion of MLP neural networks for direct solution of inverse problems," *IEEE transactions on magnetics,* vol. 41, no. 5, pp. 1784-1787, 2005.

[44] I. Malkiel, M. Mrejen, A. Nagler, U. Arieli, L. Wolf and H. Suchowski, "Plasmonic nanostructure design and characterization via Deep Learning," *Light: Science & Applications,* vol. 7, no. 1, pp. 1-8, 2018.

[45] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville and Y. Bengio, "Generative adversarial networks," in *Advances in Neural Information Processing Systems*, 2014.

[46] Z. Liu, D. Zhu, S. P. Rodrigues, K. T. Lee and W. Cai, "Generative Model for the Inverse Design of Metasurfaces," *Nano letters,* vol. 18, no. 10, pp. 6570-6576, 2018.

[47] M. Mohri, A. Rostamizadeh and A. Talwakar, Foundations of Machine Learning, Cambridge, MA: MIT Press, 2012.

[48] S. H. Hall and H. L. Heck, Advanced Signal Integrity for High-Speed Digital Designs, Hoboken, NJ: Wiley, 2009, Ch. 13.

[49] G. Fishman, Monte Carlo: Concepts, Algorithms, and Applications, Newyok, NY: Springer-Verlag, 1996.

[50] S. Chun, J. Hejase, J. Tang, J. Audet, D. Becker, D. Dreps, G. Wiedemeier, M. Nguyen, L. Walls, F. Preda and D. Douriet, "Package and printed circuit board design of a 19.2 Gb/s data link for high-performance computing," in *IEEE 67th Electronic Components and Technology Conference (ECTC)*, 2017.

[51] ANSYS Inc., "ANSYS Electronics Desktop," [Online]. Available: https://www.ansys.com/products/electronics/ansys-electronics-desktop. [Accessed 25 Feb. 2019].

[52] 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.

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

[54] H. G. Dietz, "The Aggregate Magic Algorithms," University of Kentucky, 2011. [Online]. Available: http://aggregate.org/MAGIC/. [Accessed March 2020].

[55] P. I. Frazier, "A tutorial on bayesian optimization," *arXiv preprint,* arXiv:1807.02811, 2018.

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

[57] "Scikit-learn," [Online]. Available: https://scikit-learn.org/stable/. [Accessed June 2020].

[58] M. D. Hoffman, E. Brochu and N. de Freitas, "Portfolio Allocation for Bayesian Optimization," in *27th Conference on Uncertainty in Artificial Intelligence*, Barcelona, Spain, 2011.

[59] "scikit-optimize," [Online]. Available: https://scikit-optimize.github.io/stable/. [Accessed Oct. 2019].

[60] "Spearmint," [Online]. Available: https://github.com/HIPS/Spearmint. [Accessed June 2020].

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

[62] 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.

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

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

[65] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, O. Grisel and et. al., "Scikit-learn: Machine Learning in Python," *the Journal of machine learning research,* vol. 12, pp. 2825-2830, 2011.

[66] J. Snoek, H. Larochelle and R. P. Adams, "Practical Bayesian Optimization of machine learning algorithms," in *Advances in Neural Information Processing Systems, pp. 2951-2959*, 2012.

[67] T. Beukema, "Topics in Design and Analysis of High Data Rate SERDES Systems," in *IEEE SSCS Denver Section Seminar*, Fort Collins, CO, September, 2009.

[68] M. Ahadi Dolatsara, J. Hejase, W. Becker, J. Kim, S. Lim and M. Swaminathan, "Worst-Case Eye Analysis of High-Speed Channels Based on Bayesian Optimization," *IEEE Transactions on Electromagnetic Compatibility,* 2020.

[69] A. Gretton, K. M. Borgwardt, M. J. Rasch, B. Schölkopf and A. Smola, "A kernel two-sample test," *The Journal of Machine Learning Research,* vol. 13, no. 1, pp. 723-773, 2012.

[70] C. L. Li, W. C. Chang, Y. Cheng, Y. Yang and B. Póczos, "MMD GAN: Towards deeper understanding of moment matching network," in *Advances in neural information processing systems (NIPS)*, 2017.

[71] L. Dinh, J. Sohl-Dickstein and S. Bengio, "Density estimation using real NVP," *arXiv preprint arXiv:1605.08803,* 2016.

[72] C. G. Gasca, S. C. Pueyo and C. A. Chagoyen, CMOS continuous-time adaptive equalizers for high-speed serial links, New York, USA: Springer International Publishing, 2015.

[73] W. T. Beyene, "The design of continuous-time linear equalizers using model order reduction techniques," in *IEEE-EPEP Electrical Performance of Electronic Packaging*, 2008.

[74] L. Ardizzone, J. Kruse and et al., "Framework for Easily Invertible Architectures (FrEIA), Source code," [Online]. Available: https://github.com/VLL-HD/FrEIA. [Accessed June 2020].

[75] M. Ahadi Dolatsara, H. Yu, J. Hejase, W. Becker and M. Swaminathan, "Invertible Neural Networks for Inverse Design of CTLE in High-speed Channels," in *IEEE Electrical Design of Advanced Packaging and Systems (EDAPS) Symposium*, 2020.

[76] S. Chun and e. al., "Package and printed circuit board design of a 19.2 Gb/s data link for high-performance computing," in *IEEE 67th Electronic Components and Technology Conference (ECTC)*, 2017.