A Bayesian Framework for Optimizing Interconnects in High-Speed Channels

Hakki M. Torun*, Mourad Larbi, and Madhavan Swaminathan
School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA, 30332-0250
Center for Co-design of Chip, Package and System (C3PS)
Email: *htorun3@gatech.edu

***Abstract*—Increasing demand in higher bandwidth chip-to chip communications have resulted in challenges related to modelling and optimization of their electrical performance due to CPU intensive simulations arising from multiscale structures. Conventional approaches use various approximations to either reduce the design complexity or reduce the simulation time, however, this can lead to inaccurate models and sub-optimal designs. In this paper, we address this problem by using machine learning based techniques and propose a Bayesian framework to model and optimize interconnects in high-speed channels in an accurate yet efficient fashion.**

I. Introduction

One of the key challenges in designing modern electronic systems is to ensure reliability and performance of chip-to-chip interconnect channels. The bandwidth of these channels needs to be sufficiently high so that they can support higher data rates. However, communicating at higher speeds degrades signal integrity due to increased insertion loss, return loss, delay and cross-talk related intersymbol interference (ISI), causing higher bit error rates (BER). In order to reduce these effects and improve signal integrity, the interconnects must be optimized for their electrical performance so that they can support the high bandwidth requirement.

Optimization of such high speed channels is a non-trivial task given that the system has a non-linear response to its control parameters such as the geometrical parameters of the interconnects and the material properties. Furthermore, a single optimization loop usually consists of two CPU extensive stages, namely frequency response generation and eye diagram simulation. In the first stage, the channel is commonly modelled as coupled, lossy transmission lines. The impedance and coupling profile, represented as multiconductor RLGC matrices or scattering matrix, of the channel is extracted using a 2D or 3D full-wave EM simulation over a large bandwidth. This is then exported to a time domain circuit simulator to be combined with the driver and receiver models and the eye diagram is simulated using several million pseudorandom bit sequence (PRBS) as input to capture the eye characteristics at very low BER contours.

Conventional approaches to high-speed channel optimization involves various approximations to reduce the CPU time required to characterize the jitter and eye opening. For the frequency response generation stage, W-element transmission line models are being utilized to parametrize the frequency response of the channel assuming constant inductance and capacitance over the frequency band and approximating the resistance of the transmission line as a function of square root of the frequency to model the skin effect [1]. For the eye diagram simulation stage, statistical methods are being utilized to bypass bit-by-bit simulation to generate the eye characteristics, but the non-linearity of the driver and receiver circuits can not be included with statistical approaches, which can lead to inaccurate results.

In order to optimize the eye diagram using an accurate simulation framework where no approximations are being made, machine learning based techniques that are designed to operate in CPU intensive simulation frameworks, namely Bayesian Optimization (BO), can be utilized. However, for the application to high-speed channels, directly using BO to optimize the eye diagram can still be a CPU exhaustive procedure as each system query corresponds to two very expensive simulations.

In this paper, we extend the conventional approach of surrogate based optimization and propose a new, Bayesian based framework that uses an additional surrogate model to directly optimize the eye opening of interconnects in high-speed channels in an accurate yet efficient way. Usually, the cost of full-wave EM simulations to generate the frequency response of these channels are very high due to performing a frequency sweep from DC to high GHz regimes. If BO is applied directly, at every iteration, this frequency sweep has to be performed from scratch along with expensive eye diagram simulations. However, single-frequency point simulations require orders of magnitude less CPU time. Here, we leverage this fact to derive a surrogate model of the frequency response of the channel by treating frequency as another input parameter to the predictive model, hence, eliminate the requirement of high-bandwidth simulations. This greatly reduces computational overhead of collecting training data and allows for creating the model with full-wave EM accuracy in a very efficient fashion. We use additive Gaussian Process (ADD-GP) [2] as our predictive model and represent the frequency response of the channel by multi-conductor RLGC matrices without enforcing strict frequency dependence on any of its elements. Then, we use Two-Stage Bayesian Optimization (TSBO) [3] algorithm to optimize the eye opening by using the derived ADD-GP model to generate frequency response of the channel and use it in a commercial circuit solver to perform bit-by-bit time domain simulation to generate the eye diagram. As an example of the high speed channel, we consider three single-ended microstrip transmission lines on a silicon dioxide substrate.

II. Gaussian Process and Bayesian Optimization

GPs are powerful and flexible predictors that are being widely used in both machine learning and electronics design (EDA) community due to its theoretical and practical advantages. From the theoretical point of view, it has been shown that neural networks with one hidden layer converges to a GP when the number of hidden units tend to infinity [4]. Further, GPs with certain type of kernels have the universality property [5], meaning they can model any function when there is sufficient

978-1-5386-5204-6/18/$31.00 ©2018 IEEE

data. From the practical point of view, being a non-parametric method, GPs eliminate the need for determining problem specific model structures encountered in neural networks such as network architecture, number of hidden layers and hidden units, learning rate or activation function to be used.

In GP, the model is derived according to the Bayes' Theorem. Here, the prior used is a joint, multivariate GP with a mean (μ) and covariance matrix (K), given by:

$$f_{1:t} = \mathcal{N}(\mu(x_{1:t}), K(x_{1:t})) \tag{1}$$

where $x_{1:t}$ is the multidimensional input vector at time t. The K matrix, constructed by a predefined kernel function $k(x, x')$, of a GP solely determines types of interactions between different variables that can be captured in the final model. In an additive GP model, the underlying function is modelled as summation over all possible interactions between input variables. This is realized via defining a sub-kernel, $k_i(x_i, x_i')$, to every input variable of the problem and parametrizing the overall kernel function as follows [2] :

$$\begin{aligned} k(x,x') =& \sigma_{f_1}^2 \sum_{i=1}^{D} k_i(x_i,x_i') + \sigma_{f_2}^2 \sum_{1<i<j<D} k_i(x_i,x_i')k_j(x_j,x_j') \\ &... + \sigma_{f_D}^2 \sum_{1<i_1<...<i_n<D} \left[\prod_{d=1}^{n} k_{i_d}(x_{i_d},x_{i_d}')\right] \end{aligned} \tag{2}$$

where (σ_{f_n}) is the variance hyperparameter associated to each order of interaction; n denotes the maximum allowed order of interactions among D possible orders. By assigning a hyperparameter to each order of interaction, the final ADD-GP model enables to interpret which order contributes more to the final response. It should also be noted that in this paper, we choose $n = D$, where D^{th} order of interaction corresponds to a standard GP model. Hence, ADD-GP becomes a superset of the standard GP where all the variables interact together to form the prediction. Further, we use the same sub-kernel for every input variable as *squared-exponential* function with unit variance, defined as:

$$k_i(x_i, x_i') = exp\left(-\frac{1}{2}\frac{(x_i - x_i')^2}{\sigma_{l_i}^2}\right) \tag{3}$$

where σ_{l_i} is known as the length-scale parameter. As each variable has its own sub-kernel, a separate length-scale is assigned to every parameter, effectively implementing the automatic relevance determination (ARD).

After defining the kernel function as in (2), the covariance matrix for the GP prior is constructed as:

$$K(x) = \begin{bmatrix} k(x_1,x_1) & \dots & k(x_1,x_t) \\ \vdots & \ddots & \vdots \\ k(x_t,x_1) & \dots & k(x_t,x_t) \end{bmatrix} \tag{4}$$

The training of a GP is performed in a Bayesian framework to maximize the marginal likelihood [6], corresponding to finding $\sigma_{f_{1:D}}$ and $\sigma_{l_{1:D}}$ in (2) and (3) along with the standard deviation of the predicted noise associated with the data, σ_n. This is done by using a quasi-newton based gradient descent method to minimize negative of the log marginal likelihood function, given by:

$$\begin{aligned} \log p(y_{1:t} \mid x_{1:t}) =& -\frac{1}{2} y_{1:t}^T (K + \sigma_n^2 I)^{-1} y_{1:t} ... \\ &... - \frac{1}{2}\log|K + \sigma_n^2 I| - \frac{t}{2}\log 2\pi \end{aligned} \tag{5}$$

where I is the identity matrix of size t. Finally, the predictive mean and standard deviation at test points, x^*, are found by:

$$\mu(x^*) = k^T (K + \sigma_n^2 I)^{-1} f_{1:t} \tag{6}$$

$$\sigma^2(x^*) = k^* - k^T (K + \sigma_n^2 I)^{-1} k \tag{7}$$

where $k^* = k(x^*, x^*)$ and $k = k(x^*, x_{1:t})$ are as in (2); K is given by (4) and σ_n is determined from the training.

A. Two-Stage Bayesian Optimization

The Bayesian framework supplied by the GP model can be used to develop a global optimization framework for black-box systems. This is called Bayesian Optimization (BO) [7]. In BO, the point-wise predictive mean and standard deviation in (6) and (7) are used to construct an acquisition function, $u(x)$. In conventional BO, an acquisition function is selected apriori to the optimization problem. Then, the next set of parameters to be simulated with the goal finding the global optimizer is selected as the parameters maximizing $u(x)$. The most popular strategies to construct $u(x)$ are probability of improvement (PI), expected improvement (EI) and upper confidence bound (UCB) criteria, given as:

$$u_{PI} = \Phi\left((\mu(x) - \widetilde{f}^* - \zeta)/\sigma(x)\right) \tag{8}$$

$$u_{EI} = (\mu(x) - \widetilde{f}^* - \zeta)\Phi(Z) + \sigma(x)\phi(Z) \tag{9}$$

$$u_{UCB} = \mu(x) + K\sigma(x), \quad K = \sqrt{2ln(2\pi M^2/(12\eta))} \tag{10}$$

where $\widetilde{f}^*$ is the best point observed so far, ζ is a hyper parameter for u_{PI} and u_{EI}, M is number of calls made to UCB so far, $(1-\eta)$ is the probability of zero regret for UCB, $Z = (\mu(x) - \widetilde{f}^* - \zeta)/\sigma(x)$, $\Phi(.)$ and $\phi(.)$ are the CDF and PDF of normal distribution respectively.

In TSBO, the $u(x)$ that best suits the problem is learned in an adaptive fashion. This is realized via using each $u(x)$ in (8-10) sequentially and monitoring the contribution of each function to the problem of finding the global optimum. After enough observations have been made, the $u(x)$ with the highest contribution is selected and used for subsequent iterations. Further, TSBO uses a hierarchical partitioning tree to perform fast exploration of the underlying sample space and identifies a tight region that contains the global optimum. Then, this tight region is fully exploited to find the set of parameters that maximizes the underlying function, i.e. $x^* = \arg\max_{x \in \mathbb{X}^D} f(x)$. Note that TSBO does not use the additive kernel in (2) for constructing the GP used for optimization, but uses ARD Matern 5/2 function, given as:

$$k(x_i, x_j) = \sigma_f^2(1 + \sqrt{5}r + \frac{5}{3}r^2)e^{-\sqrt{5}r} \tag{11}$$

where $r = \left(\sum_{d=1}^{D} \frac{(x_{i,d} - x_{j,d})^2}{\sigma_d^2}\right)^{1/2}$; σ_d is the length scale of each input parameter and σ_f is the signal standard deviation.

III. High-Speed Channel Model

The structure of the high-speed channel considered in this work is three single-ended and coupled microstrip transmission lines on a silicon dioxide substrate as in Fig. 1. The input parameters, along with their corresponding bounds, are given in Table I.

In order to create the surrogate model of the microstrip channel using ADD-GP, 500 samples based on uniform Latin

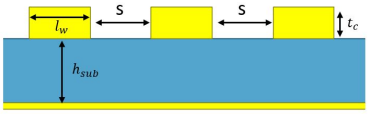

Fig. 1. Cross-section of the single ended microstrip channel.

TABLE I. CONTROL PARAMETERS OF THE MICROSTRIP STRUCTURE

Parameter		Unit	Min	Max
Line Width	l_w	μm	0.4	3
Line Thickness	t_c	μm	0.4	3
Spacing	s	μm	0.4	3
Substrate Height	h_{sub}	μm	1	5
Frequency	f	GHz	0.1	20

Hypercube Sampling (LHS) are determined. Then, a full-wave EM solver, Ansys HFSS [8], is used to extract the coupled RLGC matrices using lines of length $100\mu m$. As we consider frequency as an input to the surrogate model, simulations are done at single frequency points rather than a sweep in the entire bandwidth. Hence, the total amount of CPU time required to collect training data is greatly reduced. The dimensionality of the output space defined by the full rank RLGC matrices is 36. Since the microstrip channel is a reciprocal and symmetric network, it is sufficient to only consider 11, 12, 13 and 22 elements of the RLGC matrices to represent the complete impedance and coupling profile, effectively reducing the number of outputs to 16.

After the training data is collected, the samples are standardized to have zero mean and unit standard deviation as following:

$$\widetilde{x_{1:N}} = \frac{x_{1:N} - \mu(x_{1:N})}{\sigma(x_{1:N})}, \quad \widetilde{y_{1:N}} = \frac{y_{1:N} - \mu(y_{1:N})}{\sigma(y_{1:N})} \tag{12}$$

where μ_x, μ_y, σ_x and σ_y denotes the mean and standard deviation of the input and output training data respectively. For GP models using a zero mean multivariate normal distribution as the prior, standardizing the data before training is significant to relate the training data more to the prior distribution.

The standardized input and output vectors in (12) are used to perform the training of the ADD-GP model as explained in Section II. Since GP models can only model a single output at a time, 16 independent ADD-GP models have been trained to predict RLGC matrices.

The model quality is assessed using k-fold cross validation (CV) method with $k = 5$. Here, the training data is randomly divided into k folds, each of which containing (N/k) samples. Then, the ADD-GP model is trained using $(k-1)$ folds and k^{th} fold is used as the validation set. The process is repeated k times so that each fold is used both in training and validation sets. The quality of the model is then measured using the mean-squared error (MSE) of the k-fold CV, calculated as:

$$\epsilon_{CV\text{-}MSE} = \frac{1}{K}\sum_{k=1}^{K}\frac{1}{N}\sum_{n=1}^{N}\left(\hat{y}^{(n)}_{(X\setminus X_k)} - \tilde{y}^{(n)}_k\right)^2 \tag{13}$$

where $\hat{y}^{(n)}_{(X\setminus X_k)}$ is the prediction of the n^{th} sample when the model is trained using (k-1) folds, i.e. $(X \setminus X_k)$; $\tilde{y}_k$ is the standardized output vector in k^{th} fold and N is the total number of samples in a single fold. As can be seen from Table II, the CV-MSE error for standardized RLGC matrices is kept less than 2.9%, showing the high quality of the predictive model. A further test of the ADD-GP model is performed by converting the RLGC matrices to S-Parameters and comparing it to full-wave simulation of a channel with a length of 15 mm. For a random parameter assignment, the comparison for insertion loss (IL), return loss (IL), near-end (NEXT) and far-end (FEXT) crosstalk over the entire bandwidth can be seen in Fig. 2.

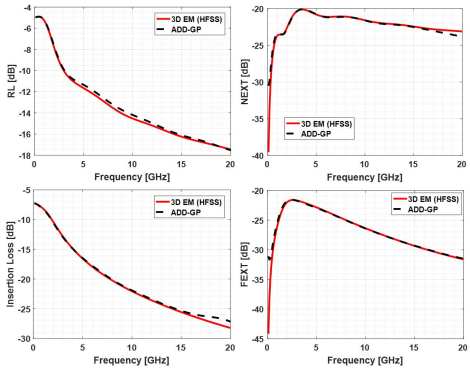

Fig. 2. Comparison of the ADD-GP model with full-wave simulations for a channel of length 15mm.

TABLE II. CV MSE VALUES OF THE ADD-GP MODEL

Parameter	ADD-GP CV MSE	Parameter	ADD-GP CV MSE	Parameter	ADD-GP CV MSE	Parameter	ADD-GP CV MSE
R_{11}	0.001	L_{11}	0.006	G_{11}	0.008	C_{11}	0.027
R_{12}	0.005	L_{12}	0.003	G_{12}	0.025	C_{12}	0.019
R_{13}	0.018	L_{13}	0.007	G_{13}	0.010	C_{13}	0.017
R_{22}	0.005	L_{22}	0.004	G_{22}	0.029	C_{22}	0.022

IV. OPTIMIZING HIGH SPEED CHANNEL

The conventional approach to optimize a high-speed channel is finding the geometrical parameters of the interconnects that would minimize IL, NEXT and FEXT while matching the input impedance of the channel to output impedance of the driver to minimize reflections. We call this approach as *frequency domain optimization* (FDO) and summarize its framework in Fig. 3(a). Usually, a set of target specifications are determined and the cost function to be minimized via optimization is defined as:

$$f(x) = \sum_{i=1}^{4} w_i(|y_i - y_{i_t}|) \tag{14}$$

where y_i denotes the objectives, i.e. RL, IL, FEXT and NEXT; y_{i_t} denotes the target specifications and w_i is the weighting constant to transform multi-objective optimization problem into a single-objective one. However, FDO approach can often result in sub-optimal designs in terms of signal integrity. The trade-offs regarding IL, RL, NEXT and FEXT are determined prior to the optimization since it is not clear which objective affects the eye opening more. For instance, crosstalk related terms may be affecting the eye opening more than IL or RL for a particular channel and vice versa for another.

The proposed framework in this paper directly optimizes the eye opening, which results in automatic determination of frequency domain trade-offs of a particular structure. Compared to directly using BO for the same objective as in Fig. 3(b), the framework used in this paper eliminates the need for high-bandwidth frequency domain simulations in the optimization loop by using the ADD-GP model derived using only single-frequency simulations. This reduces the system simulation time in the optimization loop by orders of

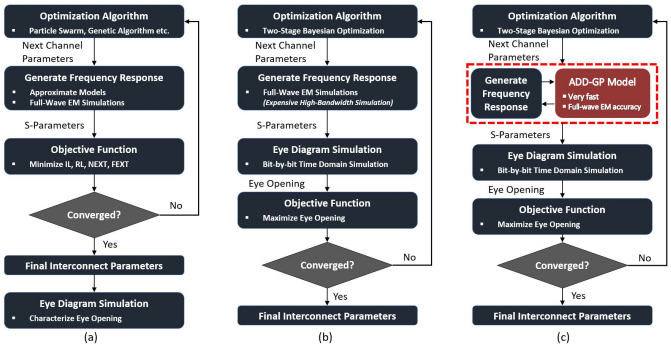

Fig. 3. Comparison of optimization setups used for different approaches. (a) FDO. (b) Inefficient Direct Eye Optimization. (c) Proposed Framework.

magnitude. For direct eye optimization, the objective function is defined as:

$$f(x) = W_E H_E \tag{15}$$

where W_E and H_E denotes width and height of the eye diagram at a particular BER contour, chosen as 10^{-12} for this work. The automated flow of the proposed framework is summarized in Fig 3(c). Here, geometrical parameters of the microstrip channel is chosen by TSBO and fed into the ADD-GP model to generate the S-Parameters of the microstrip channel with a length of 10mm. The channel S-Parameters are then used by a commercial circuit solver, Keysight ADS [9], to perform bit-by-bit simulation to generate the eye height and eye width at a data rate of 16 Gbps. The output impedance of the driver is fixed to 50Ω and the load at the receiver is fixed as a 50Ω resistor in parallel with a shunt capacitor of 1pF to represent pad parasitics. The eye width and height generated by ADS is then combined in (15) and fed back into TSBO to proceed into next iteration.

TABLE III. CHARACTERISTICS OF EYE DIAGRAMS

	FDO	Direct Eye Optimized
Eye Width	26.6 ps	46.2 ps
Eye Height	90 mV	167 mV
Pk-Pk Jitter	25.9 ps	11.8 ps

The optimization results are summarized in Table III and in Fig. 4. For the FDO approach, we choose $w = 1$ for every objective in (14) and choose the target specifications as *zero* for NEXT, FEXT, IL and RL. Compared to FDO, direct eye optimized channel has resulted in 42.4% and 46.1% increase in eye height and width along with 54.4% reduction in peak-to-peak jitter. This is due to automatically adjusting the trade-offs regarding RL, IL, NEXT and FEXT by directly optimizing the eye opening. As can be seen in Fig. 4(a), the direct eye optimization approach have favored parameters that reduce FEXT more instead of IL and RL and achieved a better overall channel performance.

V. CONCLUSION

In this work, we have proposed a accurate yet efficient framework for optimizing signal integrity of high-speed channels. In order to reduce the CPU time of channel simulation, we created a surrogate model of RLGC matrices using ADD-GP. The training data used for this step is collected using single frequency point simulations, which enabled an inexpensive way to derive the model without losing accuracy. The

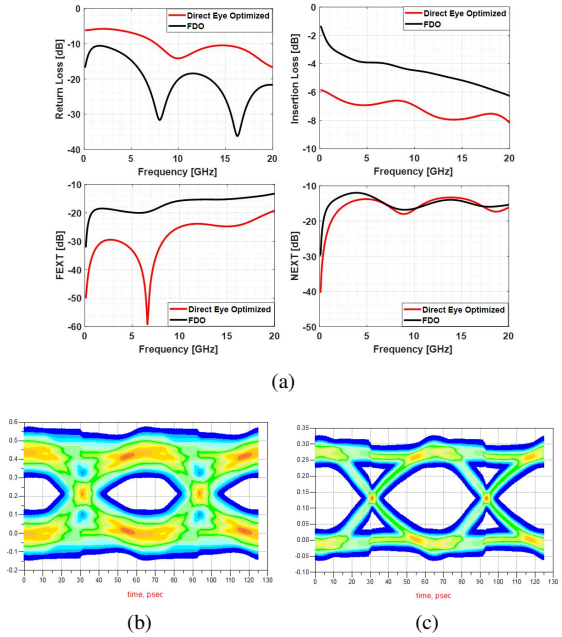

Fig. 4. Performance comparison of FDO to direct eye optimization. (a) Frequency response of resulting channels. (b) Resulting eye diagram for FDO. (c) Resulting eye diagram for direct eye optimization.

resulting ADD-GP model is shown to agree well with full-wave simulations, having a maximum of 2.9% CV-MSE on all elements of RLGC matrices. This ADD-GP model is then used in the optimization loop driven by TSBO to directly maximize eye opening rather than conventional approach of tuning the frequency response of the channel. Compared to conventional approach, direct eye optimization resulted in 42.4% and 46.1% increase in eye height and width along with 54.4% reduction in peak-to-peak jitter at a data of 16 Gbps.

ACKNOWLEDGEMENT

This research is funded by the DARPA CHIPS project under Award N00014-17-1-2950.

REFERENCES

[1] H. Kim, C. Sui, K. Cai, B. Sen, and J. Fan, "Fast and precise high-speed channel modeling and optimization technique based on machine learning," *IEEE Transactions on Electromagnetic Compatibility*, 2017.

[2] D. K. Duvenaud, H. Nickisch, and C. E. Rasmussen, "Additive gaussian processes," in *Advances in neural information processing systems*, 2011.

[3] 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*, 2018.

[4] R. M. Neal, *Bayesian learning for neural networks*. Springer, 1996, vol. Lecutre Nots in Statistics 118.

[5] C. A. Micchelli, Y. Xu, and H. Zhang, "Universal kernels," *Journal of Machine Learning Research*, vol. 7, no. Dec, pp. 2651–2667, 2006.

[6] C. E. Rasmussen and C. K. Williams, *Gaussian processes for machine learning*. MIT press Cambridge, 2006, vol. 1.

[7] 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.org, eprint arXiv:1012.2599, December 2010.

[8] ANSYS, "Ansys hfss ver. 2015.2." [Online]. Available: http://www.ansys.com

[9] Keysight Technologies, "Advanced design systems (ads) ver. 2016.01." [Online]. Available: https://www.keysight.com