This is the author's version of an article that has been published in this journal. Changes were made to this version by the publisher prior to publication. The final version is available at http://dx.doi.org/10.1109/TEMC.2020.2973946

Surrogate-based Analysis and Design Optimization of Power Delivery Networks

Felipe de Jesús Leal-Romo, José E. Rayas-Sánchez, and José L. Chávez-Hurtado

***Abstract*— As microprocessor architectures continue to increase computing performance under low-energy consumption, the combination of signal integrity, electromagnetic interference, and power delivery is becoming crucial in the computer industry. In this context, power delivery engineers make use of complex and computationally expensive models that impose time-consuming industrial practices to reach an adequate power delivery design. In this paper, we propose a general surrogate-based methodology for fast and reliable analysis and design optimization of power delivery networks (PDN). We first formulate a generic surrogate model methodology exploiting passive lumped models optimized by parameter extraction to fit PDN impedance profiles. This PDN modeling formulation is illustrated with industrial laboratory measurements of a 4th generation server CPU motherboard. We next propose a black box PDN surrogate modeling methodology for efficient and reliable power delivery design optimization. To build our black box PDN surrogate, we compare four metamodeling techniques: support vector machines, polynomial surrogate modeling, generalized regression neural networks, and Kriging. The resultant best metamodel is then used to enable fast and accurate optimization of the PDN performance. Two examples validate our surrogate-based optimization approach: a voltage regulator with dual power rail remote sensing intended for communications and storage applications, by finding optimal sensing resistors and loading conditions; and a multiphase voltage regulator from a 6th generation Intel® server motherboard, by finding optimal compensation settings to reduce the number of bulk capacitors without losing CPU performance.**

***Index Terms*— impedance profile, IP protection, metamodels, microprocessor, motherboard, optimization, power delivery network, power integrity, surrogate modeling, voltage regulator.**

I. INTRODUCTION

As computer architectures continue to increase the number of cores while keeping frugal power consumption [1],[2], the combination of signal integrity (SI), electromagnetic interference (EMI), and power delivery (PD) is becoming crucial in the industrial design of computer platforms to satisfy stringent performance requirements within time-to-market commitments. In this context, industry is paying more attention to PD by emphasizing power integrity (PI) engineering to overcome cost and performance targets.

In a typical industrial PD analysis, a physical power delivery network (PDN) is proposed for which a large and complex circuital model (SPICE-like network), including distributed passive components, is built. This circuital model is enabled to include detailed models for external decoupling capacitance aimed at compensating events of sudden change of load (di/dt), from the silicon (CPU) up to the voltage regulator (VR). These circuital models allow PI engineers to assess the performance of the PDN, for instance, by looking into the impedance profile [3]-[5] for which specified target impedances are defined [6],[7]. In addition to the impedance profile, other performance metrics, such as voltage drop analyses, are needed to ensure the minimum voltage level of operation allowed by the VR and other components. Other metrics include simultaneous switching noise (SSN) [8],[9], also known as ground bounce, voltage regulator's power losses [10],[11], plane current density for copper's reliability, etc.

PI engineers rely on several CAD tools to design and characterize a PDN; they frequently employ 2.5-D and 3-D full-wave EM simulators to extract accurate circuital models able to emulate most intrinsic PDN parasitic effects. As mentioned before, these circuital models are typically implemented as large distributed SPICE networks representing the whole PDN [9],[12]-[14], whose simulation can last from a couple of minutes up to several hours or even days. To overcome this high computational cost, some approaches to speed up PDN network development and simulation have been proposed. Extraction of the PDN equivalent circuital model can be accelerated by using the partial element equivalent circuit (PEEC) method [14],[15]. PDN metamodeling approaches for accurate package prediction of bump inductance is proposed in [16] by using design of experiments (DoE) and machine learning techniques, *e.g.*, artificial neural networks (ANN), support vector machines (SVM), nonlinear regression, and combined ANN-piecewise-linear (PWL) modeling [16]. Efficient PDN Bayesian optimization is proposed in [17] to minimize clock skew and maximize voltage regulators efficiency.

In this paper, a general methodology for surrogate-based analysis and design optimization of power delivery networks is proposed. We first formulate a generic surrogate model methodology for accurate and fast prediction of PDN performance. Our modeling methodology exploits fast passive lumped models optimized by parameter extraction (PE) to fit PDN impedance profile from industrial laboratory measurements. This is illustrated by modeling a PDN of a 4th generation Intel® Xeon® CPU server. Secondly, we propose a black box surrogate modeling approach for efficient and reliable PDN design optimization [18]-[20]. We compare

F. J. Leal-Romo is with Intel Corp., Zapopan, 45109 Mexico (e-mail: felipe.de.jesus.leal.romo@intel.com), J. E. Rayas-Sánchez, and J. L. Chávez-Hurtado are with the Department of Electronics, Systems, and Informatics, ITESO – The Jesuit University of Guadalajara, Tlaquepaque, Jalisco, 45604 Mexico. F. J. Leal-Romo is funded through a CONACYT scholarship (*Consejo Nacional de Ciencia y Tecnología,* Mexican Government).

several PDN metamodels, including support vector machines (SVM), polynomial surrogate modeling (PSM), generalized regression neural networks (GRNN), and Kriging. The end goal of these PDN metamodels is to enable accurate and fast optimization of PDN performance. We illustrate our metamodel-based design optimization approach by two examples: 1) a PDN with dual sensing voltage regulator for communications and storage applications, to find optimal sensing resistors and loading conditions; and 2) a PDN motherboard of a 6th generation Intel® Xeon® computer, to find optimal settings of a multiphase switching VR controller, reducing by 30% the number of decoupling capacitors without losing CPU performance.

The rest of the paper is organized as follows. Section II illustrates the development of PDN passive lumped models by fitting the impedance profile, intended to enable customers to efficiently build their customized PD board. Section III describes a generic formulation to develop PD black box surrogate models for enabling PDN performance assessment at low computational cost. Section IV illustrates an example to build a PD surrogate model of a dual sensing VR to come up with an optimized sense resistors' recipe and achieve the best power consumption without impacting silicon's threshold voltage. In Section V, a second example is described by addressing components cost reduction of a 6th generation Intel® Xeon® CPU from motherboard and VR point of view. Finally, in Section VI, our conclusions are presented.

II. SURROGATE LUMPED MODELS BY PARAMETER EXTRACTION FOR CUSTOMIZED ANALYSIS AND DESIGN

A. Custom PDN Design

Customers and vendors usually want to customize their own motherboard design to offer different CPU performance features or cheaper platform costs. CPU manufacturers can enable customers by providing a set of reference PD design guidelines, as well as coarse surrogate models to estimate impedance PDN. This is done with two goals: 1) to protect manufacturer's CPU and chipset intellectual property; and 2) to provide an easy way to simulate PD performance for further customers' analyses.

B. Surrogate Lumped Models for PDN Intellectual Property Protection

As mentioned before, the PDN impedance profile can be approximated by equivalent SPICE-like lumped models valid over a limited frequency band. A typical frequency response measured from a realistic PDN is illustrated in Fig. 1, where some frequency resonances caused by each decoupling state are shown.

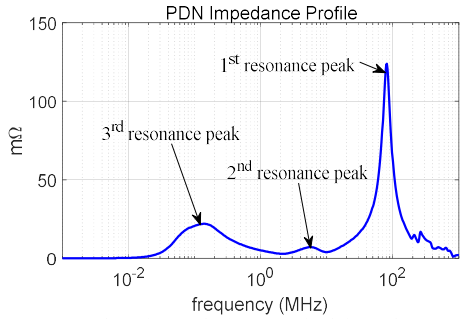

Fig. 1. Frequency domain analysis: a typical impedance profile showing PDN behavior from the voltage regulator module (VRM) to the silicon.

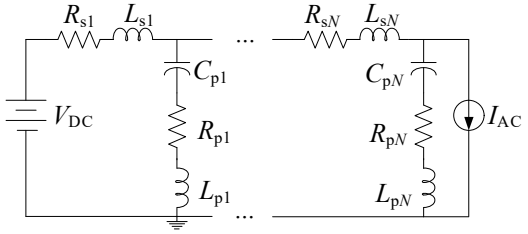

Fig. 2. Single-sided equivalent RLC lumped model with ideal ground connection to fit the impedance profile.

A fast PDN lumped model can be implemented by a series of single-sided RLC sections with an ideal ground return path, as shown in Fig. 2. This lumped model approximation is built incrementally, where N corresponds to a number of RLC sections included, which mainly depends on the number of resonant points in the measured impedance profile. The series resistors R_{s1} … R_{sN} account for the resistivity of copper from parallel power and ground layers. Series inductances L_{s1} … L_{sN} are associated mainly to the vertical transitions of vias interconnecting layers as well as the distributed parasitic inductance of power and ground planes. Finally, parallel capacitances C_{p1} … C_{pN} , along with their corresponding intrinsic parasites R_{p1} … R_{pN} and L_{p1} … L_{pN}, represent each capacitive decoupling stage placed along the entire PDN, from the VR to the silicon. A simplified cross-sectional view of a typical PDN with different decoupling stages is shown in Fig. 3, where it is illustrated the path from the voltage regulator module (VRM) up to the chip (silicon). As it is observed, the PDN is conformed of several components acting as decoupling stages to store energy until the VRM reacts. Some of the most important decoupling stages are the die side capacitors (DSC), close to the silicon for fast reaction, next the land side capacitors (LSC), which act as a secondary reserve of energy, followed by the interconnection from the package to the motherboard with the socket, which typically is very resistive, and finally the last energy reserves implemented by decoupling and bulk capacitors.

C. Optimizing Surrogate Lumped Models by Parameter Extraction

Suitable parameter values for a given a number of RLC sections (N) in the proposed PDN lumped model (see Fig. 2) should be determined by performing a curve fitting of the PDN impedance profile obtained from actual laboratory measurements. Performing laboratory measurements on the physical PDN and VR can be implemented with a voltage

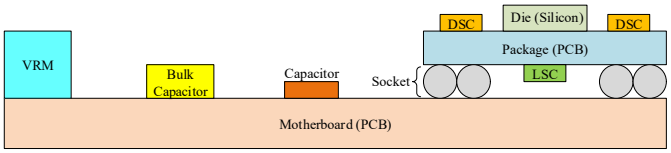

Fig. 3. Cross-sectional view of a PDN including different decoupling stages along the VR up to the silicon.

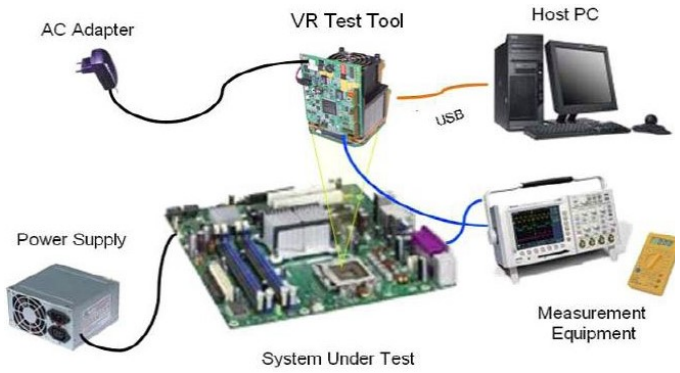

Fig. 4. Voltage regulator test tool (VRTT) to measure frequency and time-domain PDN and VR features. Taken from [21].

regulator test tool (VRTT) that makes use of an interposer to emulate CPU loading conditions [21], as illustrated in Fig. 4. After collecting the PDN impedance profile from the VRTT, we propose finding optimal surrogate lumped model parameter values by solving the following parameter extraction (PE) problem [22]:

$$\boldsymbol{x}^* = \arg\min_{\boldsymbol{x}} \left\| \boldsymbol{R}(\boldsymbol{x}) - \boldsymbol{R}^{\mathrm{t}} \right\|_2^2 \tag{1}$$

where optimization variables are in $\boldsymbol{x} = [R_{\mathrm{s}1}\ \ L_{\mathrm{s}1}\ \ \ldots\ \ R_{\mathrm{s}N}\ \ L_{\mathrm{s}N}\ \ C_{\mathrm{p}1}\ \ R_{\mathrm{p}1}\ \ L_{\mathrm{p}1}\ \ \ldots\ \ C_{\mathrm{p}n}\ \ R_{\mathrm{p}N}\ \ L_{\mathrm{p}N}]^{\mathrm{T}} \in \Re^n$, with $n = 5N$, $\boldsymbol{x}^*$ contains the extracted optimal parameter values that make the lumped circuit response $\boldsymbol{R}(\boldsymbol{x}^*)$ as close as possible to the target response $\boldsymbol{R}^{\mathrm{t}} \in \Re^r$; in our case, $\boldsymbol{R}^{\mathrm{t}}$ contains the measured PDN impedance profile. Once $\boldsymbol{R}^{\mathrm{t}}$ is available, solving (1) is computationally very fast and can be repeated by gradually increasing N until a good match is found at $\boldsymbol{x}^*$.

D. PDN Lumped Model PE Example

To exemplify the usage of this technique, we measured the impedance profile of a 4th generation Intel® Xeon® CPU server. The SPICE lumped model uses the same topology as in Fig. 2, with $N = 8$ sections (a total of $n = 40$ optimization variables). The PDN impedance profile measured from 100 Hz to 7 MHz is shown in Fig. 5. This particular frequency range was selected considering the intended PDN application (motherboard). We solved (1) by using the trust-region interior point nonlinear optimization method available in MATLAB[1]. A comparison between the measured and surrogate lumped model impedance is also depicted in Fig. 5. It is observed that the lumped model closely follows the impedance profile up to 6 MHz, with a slight difference in the 30-200 KHz frequency range due to the circuit topology employed. The time to generate this optimized lumped model was 1 minute and 36 seconds on a laptop computer with 8 GB RAM and a 7th generation i5 processor.

III. A METHODOLOGY FOR EFFICIENT AND ACCURATE PDN DESIGN OPTIMIZATION

It is possible to use PDN metamodels to optimize a PD design, as long as these metamodels have enough accuracy in the design region of interest. To obtain our PDN metamodels, we generate training and testing data within a desired design region and apply the corresponding metamodeling techniques. Then, we select the PDN metamodel with the best generalization performance and use it as a vehicle for direct PD design optimization. Our end goal is to accelerate PD design process, by reducing the time that engineers spend on assessing several tradeoff scenarios during PD design flow.

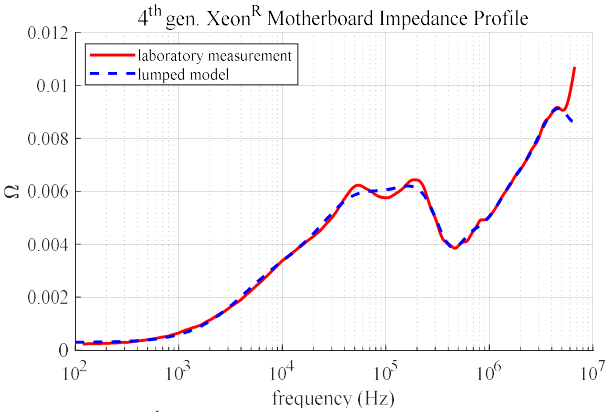

Fig. 5. Intel's 4th CPU generation motherboard's impedance profile comparison: equivalent lumped model (dashed line) *vs* laboratory measurement (solid line).

A. High Fidelity Simulation of Power Delivery Networks

Since most of the PDN building blocks are based on physical structures, engineers use CAD tools to predict the PD performance for specific power domains. Most of these CAD tools are either full-wave EM simulators or quasi-static simulation tools. Some of the commercial tools most widely used in this area include ADS[2] (Momentum, Power EM, etc.), ANSYS[3] (HFSS, SIwave, etc.), Cadence[4] (Power DC, Power SI, etc.), CST[5], among others. They are in general regarded as highly accurate, however, they are computational expensive given the complexity of the simulated structures. In this paper, we will refer to them as fine models.

During the simulation stage, PD engineers typically assess the effects of changing, for instance, the voltage regulator bandwidth, the number of phases needed by the regulator, the copper thickness, etc., based on their expertise. Hence, these tradeoff scenarios imply a significant amount of time redoing fine model simulations, and as a consequence, there is a significant engineering cost to provide a robust PD solution.

B. PDN Surrogate Modeling Flow

We propose developing a computationally cheap but sufficiently accurate surrogate model of the PDN, which later can be used for fast parametric studies or for efficient direct design optimization. A simplified surrogate modelling flow is shown in Fig. 6. It essentially consists of the following steps:

[1] MATLAB, Version R2010a, The MathWorks, Inc., 3 Apple Hill Drive, Natick MA 01760-2098, 2006.

[2] ADS – Advanced Design System, Keysight Technologies, Inc., 1400 Fountain Grove Pkwy, Santa Rosa, CA 95403-1738, USA, 2019.

[3] ANSYS Inc., Southpointe 2600 Ansys Dr., Canonsburg, PA 15317, USA, 2019.

[4] Cadence Design Systems, Inc., 2655 Seely Avenue, San Jose, CA 95134, USA, 2019.

[5] CST – Computer Simulation Technology AG, CST of America, LLC. Dassault Systemes, 175 Wyman St., Walman, MA 02451, USA., 2019.

1) Select a suitable PDN fine model (see Sub-section III.A), define the n input design parameters ($\boldsymbol{x} \in \Re^n$) and the r responses of interest ($\boldsymbol{R}_\mathrm{f} \in \Re^r$).

2) Generate training and testing data in the design space of interest. Given the high computational cost of each fine model simulation, frugal sets of data should be collected. We can exploit design of experiments (DoE) techniques to improve coverage while keeping a small amount of fine model simulations. Among the most frugal distributions of points, the star and box distributions are preferred. The star distribution requires only $2n + 1$ training base points [23], while the box distribution requires $2^n + 1$ training base points [24]. Central composite design (CCD) [25] can be exploited to define upper and lower limits, while the confusion technique with fractional factorial design (FFD) [25] can further decrease the amount of training data in box distributions with high dimensionality. For testing, we use random points inside the training region, in the range of 20-25% of the total number of training data.

3) Train the PDN surrogate model $\boldsymbol{R}_\mathrm{s}(\boldsymbol{x})$. Here we actually train several surrogate models with the available training data generated in the previous step. We use the following machine learning approaches: support vector machines (SVM), polynomial surrogate modeling (PSM), generalized regression neural networks (GRNN), and Kriging. These particular metamodeling techniques have the common characteristic of an easy regularization process, reducing the risks of over-training [26]. They are briefly described in Sub-section III.C

4) Test the PDN surrogate models. Here we test the generalization performance of the resultant surrogate models after training. This is done by measuring the relative error of each $\boldsymbol{R}_\mathrm{s}(\boldsymbol{x})$ with respect to $\boldsymbol{R}_\mathrm{f}(\boldsymbol{x})$ at all the random testing base points not seen during training.

5) Select the best PDN surrogate model. After measuring the testing errors of all the surrogate models trained, we select that one with the smallest maximum relative testing error. If the best generalization performance obtained is acceptable, we end the modeling flow. If not, we either reduce the region of interest and retrain the surrogate models, or add more training data in the same region and retrain the surrogates (for cases where the training region size must be kept fixed).

Having available the best PDN surrogate model, with an acceptable generalization performance, we can optimize it to find the best PDN design, as described in Sub-section III.D.

C. *Surrogate Modeling Techniques for Fast PDN Simulation*

As mentioned before, we evaluated four different surrogate modeling techniques: polynomial surrogate modeling (PSM), generalized regression neural networks (GRNN), support vector machines (SVM), and Kriging. A brief description of each technique follows.

Polynomial Surrogate Model (PSM). Polynomial functional surrogates have been efficiently exploited to approximate complex microwave structures [27], even by enforcing low-order polynomials [24]. Here we use the PSM formulation presented in [18], where the multinomial theorem is exploited with automated regularization. It is in essence very similar to

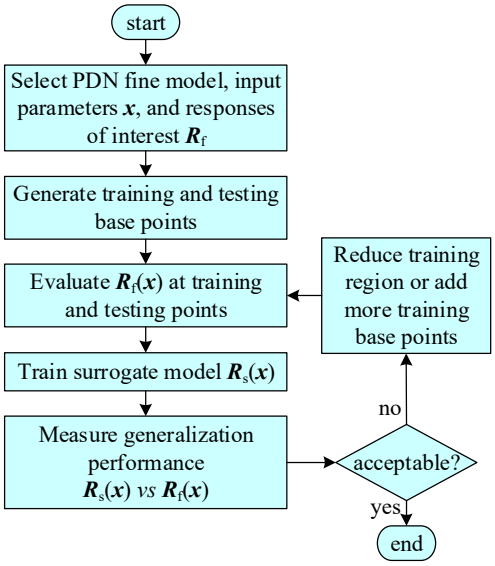

Fig. 6. Flow diagram to train a PDN surrogate model and test its generalization performance.

RSM, although PSM is not limited to second order polynomials. The m-th order functional approximation used in scalar form (per frequency point) is

$$R_\mathrm{s}^{(m)}(\boldsymbol{x}) = R_\mathrm{s}^{(m-1)}(\boldsymbol{x}) + \boldsymbol{w}^{(m)\mathrm{T}}\boldsymbol{q}^{(m)}(\boldsymbol{x}) \tag{2}$$

where $R_\mathrm{s}^{(m-1)}(\boldsymbol{x})$ is the previous $m-1$ order polynomial surrogate model function, $\boldsymbol{w}^{(m)}$ is the corresponding vector of weighting factors, and $\boldsymbol{q}^{(m)}(\boldsymbol{x}_\mathrm{f})$ contains the m-th order multinomial terms [18]. This technique exhibits good generalization performance when applied to relatively small training regions [18].

Generalized Regression Neural Network (GRNN). GRNN is a special kind of ANN that does not require an iterative training procedure [28]. Its number of hidden neurons is equal to the amount of learning data [28]. As the number of learning samples becomes large, the GRNN exhibits a fast learning and convergence to the optimal regression surface [29]. GRNN uses a special type of radial basis functions as nonlinearity; for our purposes, we use GRNN default settings of the neural network toolbox available in MATLAB.

Support Vector Machines (SVM). While ANNs are trained using the empirical risk minimization principle, SVMs use the structural risk minimization, allowing them to obtain a good trade-off between model complexity and generalization performance [30]. In order to find the optimal model parameters, the SVM technique solves a constrained quadratic optimization problem by exploiting the use of kernel functions [31],[32]. For our implementation, we use the SVM regression available in MATLAB with default linear kernel functions and sequential minimal optimization solver.

Kriging. Kriging is a type of kernel-based probabilistic model that is based on space filling experiments aiming at covering the experimental area [33]. Kriging minimizes the prediction variance by exploiting the best linear unbiased estimator (BLUE) of the output value for a given input [33]. If there are not sufficient training samples, the predictions of the resultant Kriging models may become inaccurate [33]. For our work, we use default settings of the algorithm as implemented

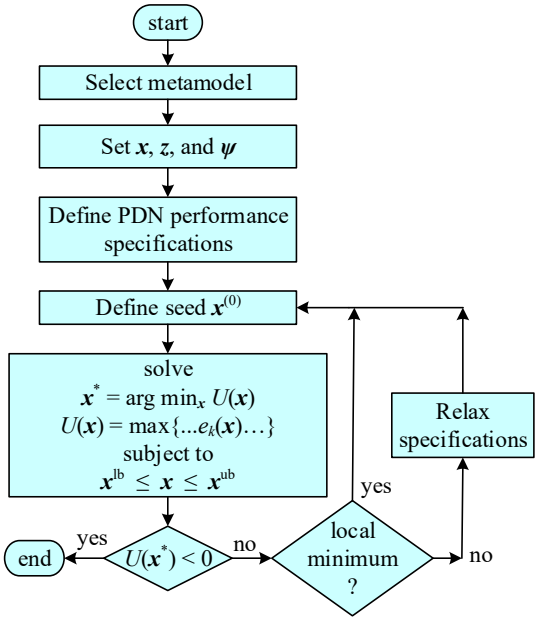

Fig. 7. Flow diagram for direct constrained design optimization of the PDN metamodel.

in MATLAB by exploiting its Gaussian regression process.

D. PDN Surrogate-Based Optimization

Once the metamodels have been developed, the next step is to select the best one and use it for direct optimization to fulfil PDN performance specifications. A flow diagram showing the proposed optimization methodology is shown in Fig. 7, which essentially consist of the following steps:

1) Select the best PDN surrogate model according to their generalization performance.

2) Set PDN surrogate model input variables ($\boldsymbol{x}$) as well as its constant pre-assigned input parameters $\boldsymbol{z}$ and independent variables $\boldsymbol{\psi}$, if any.

3) Define PDN performance specifications. Regularly, the PDN specifications fall in the category of dc resistance (R_{dc}), also called as R-path, minimum functional threshold voltage (V_{min}), maximum voltage (V_{max}) allowed for functional reliability, target impedance (Z_t), peak to peak voltage noise (V_{p2p}), power consumption (P_D), power loss (P_{loss}), etc.

4) Set an initial design $\boldsymbol{x}^{(0)}$ to start the optimization algorithm. Usually, this seed is determined from PD engineer's expertise.

5) Formulate the objective function $U(\boldsymbol{x})$ to solve the following constrained minimax optimization problem:

$$\boldsymbol{x}^* = \arg\min_{\boldsymbol{x}} U(\boldsymbol{x}) \tag{3}$$

$$U(\boldsymbol{x}) = \max\{\ldots e_k(\boldsymbol{x})\ldots\} \tag{4}$$

subject to

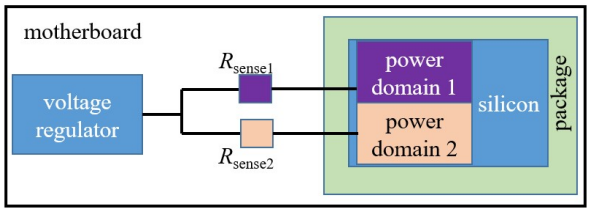

Fig. 8. PDN with a single voltage regulator and two remote sense resistors for two different power domains at silicon level. Taken from [34].

$$\boldsymbol{x}^{lb} \le \boldsymbol{x} \le \boldsymbol{x}^{ub} \tag{5}$$

where $\boldsymbol{x}^*$ is the optimal design found, $\boldsymbol{x}^{lb}$ and $\boldsymbol{x}^{ub}$ are the lower and upper limits of the input variables $\boldsymbol{x}$, respectively, and $e_k(\boldsymbol{x})$ is the k-th error function defined as:

$$e_k(\boldsymbol{x}) = \begin{cases} \dfrac{R_k(\boldsymbol{x})}{S_k^{ub}} - 1 & \text{for all } k \in I^{ub} \\ 1 - \dfrac{R_k(\boldsymbol{x})}{S_k^{lb}} & \text{for all } k \in I^{lb} \end{cases} \tag{6}$$

where $R_k(\boldsymbol{x})$ is the k-th model response at point $\boldsymbol{x}$, S_k^{ub} and S_k^{lb} are the upper and lower bound specifications ($\neq 0$), and I^{ub} and I^{lb} are the corresponding index sets.

6) Check if the optimum response $\boldsymbol{R}(\boldsymbol{x}^*)$ complies with the required performance specifications. For this assessment, two conditions can occur: a) $U(\boldsymbol{x}^*) < 0$, which implies the optimal solution found satisfies all the performance specifications, ending the algorithm; b) $U(\boldsymbol{x}^*) \ge 0$, which implies that at least one of the performance specifications is violated. If the second condition is found, it might be due to a local minimum, in which case we perturbed the starting point and optimize again the surrogate model. If it is not a local minimum, then the PDN performance specifications might be too demanding, in which case we relax them and re-optimize the surrogate (see Fig. 7).

IV. EXAMPLE I: PDN OPTIMIZATION FOR DUAL SENSING VOLTAGE REGULATOR

Consider the monolithic CPU processor proposed in [34], which consists of two power domains sharing a single VR using a dual sensing scheme, as shown in Fig. 8. Each power domain has its own individual minimum voltages (contained in vector $\boldsymbol{V}_{min} \in \Re^2$) and power consumptions ($\boldsymbol{P}_D \in \Re^2$) specifications. By varying the sense resistors ($\boldsymbol{R}_{sense} \in \Re^2$) and limiting the maximum current of each power domain ($\boldsymbol{I}_{max} \in \Re^2$), the purpose of the optimization problem is to ensure the best performance of the circuitry while reducing the power consumed from both power domains. Our PDN fine model was simulated with Synopsys®-HSPICE[6], where each simulation took an average of 5 minutes using a laptop computer with 8 GB RAM and a 7th generation i5 core processor. The design variables are $\boldsymbol{x} = [R_{sense1}\ \ R_{sense2}\ \ I_{max1}\ \ I_{max2}]^T \in \Re^4$ and the system responses are $\boldsymbol{R}(\boldsymbol{x}) = [P_{D1}\ \ P_{D2}\ \ V_{min1}\ \ V_{min2}]^T \in \Re^4$.

A. Training and Testing Data Generation

Training data was generated using a box distribution exploiting CCD and FFD techniques using 2 central points, resulting in 28 training base points. For generalization measurement, we simulated 6 random testing points within the design region of interest, which was delimited by the following lower and upper bounds: 15 Ω to 85 Ω for R_{sense1} and R_{sense2}; 0.9 A to 2.2 A for I_{max1}; and 12 A to 28 A for I_{max2}. In addition, we normalized all the input data to train the surrogate models. The entire process of generating and collecting the training and testing data took 2.83 hours.

[6] Hspui for Windows, G-2012.06, Synopsys®, 690 East Middlefield Road, Mountain View, CA 94043.

TABLE I
SUMMARY OF RELATIVE TRAINING ERRORS (%) USING DIFFERENT SURROGATE MODELS FOR TWO POWER DOMAINS

output	e_{rGRNN}	e_{rPSM}	e_{rSVM}	$e_{rKriging}$
P_{D1}	46.59	18.40	18.92	18.05
P_{D2}	47.97	2.03	6.34	1.160
V_{min1}	35.81	28.91	13.08	11.26
V_{min2}	37.02	43.30	7.25	6.22

TABLE II
SUMMARY OF RELATIVE TESTING ERRORS (%) USING DIFFERENT SURROGATE MODELS FOR TWO POWER DOMAINS

output	e_{rGRNN}	e_{rPSM}	e_{rSVM}	$e_{rKriging}$
P_{D1}	44.32	10.46	9.45	13.44
P_{D2}	35.04	3.90	4.63	3.47
V_{min1}	38.70	32.72	25.10	23.40
V_{min2}	50.54	36.15	20.78	19.42

B. Surrogate Modeling

Four different surrogate modeling techniques were implemented by using the above training data: PSM, Kriging, GRNN, and SVM. The corresponding training errors are shown in Table I. By using the testing data, the corresponding generalization errors are shown in Table II. From these results, it is observed that in this case the Kriging surrogate model exhibits the overall best training and generalization performance, selecting it to perform the direct optimization procedure.

C. Surrogate-Based Optimization

We use the following performance specifications:

a) Maximum transient power for P_{D1}, P_{D1lim} = 2.86 W
b) Maximum transient power for P_{D2}, P_{D2lim} = 36.45 W
c) Minimum transient voltage for V_{min1}, $V_{min1lim}$ = 0.71 V
d) Minimum transient voltage for V_{min2}, $V_{min2lim}$ = 0.70 V.

By using the Kriging surrogate model, we perform its direct optimization using a starting point $\boldsymbol{x}^{(0)} = [50 \quad 50 \quad 1.55 \quad 20]^T$. The corresponding initial system response is $\boldsymbol{R}_s(\boldsymbol{x}^{(0)}) = [1.6368 \quad 19.6409 \quad 0.710 \quad 0.691]^T$ which yields an objective function value $U(\boldsymbol{x}^{(0)}) = 0.0143$. By using the Nelder-Mead optimization algorithm available in MATLAB, the optimal design was found after 26 iterations and 136 surrogate model evaluations and corresponds to $\boldsymbol{x}^* = [50 \quad 49.97 \quad 0.8099 \quad 7.6522]^T$. The corresponding surrogate system response is $\boldsymbol{R}_s(\boldsymbol{x}^*) = [1.6168 \quad 19.9141 \quad 0.721 \quad 0.703]^T$, which leads to a surrogate objective function $U(\boldsymbol{R}_s(\boldsymbol{x}^*)) = -0.0043$. The corresponding system fine response for the optimal metamodel design is $\boldsymbol{R}_f(\boldsymbol{x}^*) = [1.62 \quad 19.91 \quad 0.721 \quad 0.704]^T$ yielding a fine model objective function $U(\boldsymbol{R}_f(\boldsymbol{x}^*)) = -0.0043$.

According to these results, the surrogate optimal designs resulted in a fine model that achieves the desired specifications. A total of 35 system simulations were required (34 for collecting training and testing data and one simulation to test the optimal design), instead of the hundreds of fine model system evaluations that a direct optimization methodology could require, speeding up the system design process.

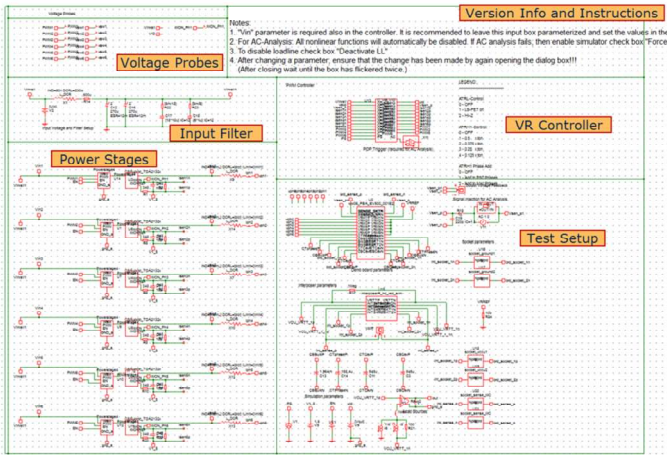

Fig. 9. SIMPLIS schematic of a multiphase VR controller including output inductors and all decoupling capacitors, as well as a 6th generation Xeon® CPU board and VRTT interposer parasites PD model.

V. EXAMPLE II: CAPACITORS REDUCTION ADJUSTING VOLTAGE REGULATOR'S PID COMPENSATION

Now consider a motherboard of a 6th generation Intel® Xeon® server. The compensation parameters of a multiphase VR controller are included in the design variables to optimize the motherboard performance, as discussed in [35] and [36]. The VR controller is simulated in SIMetrix/SIMPLIS[7], including the PDN parasites and the VRTT interposer, as mentioned in [35] and illustrated in Fig. 9.

The VR controller PID compensation parameters (k_p, k_d, k_i) and the number of bulk capacitors used in the motherboard layout area (N_{Cblk}) are selected as design variables, $\boldsymbol{x} = [k_d \quad k_p \quad k_i \quad N_{Cblk}]^T \in \Re^4$. The responses of interest correspond to the maximum transient voltage V_{max} and the minimum transient voltage V_{min}, represented as $\boldsymbol{R}(\boldsymbol{x}) = [V_{max} \quad V_{min}]^T \in \Re^2$. Fig. 10 illustrates a typical transient waveform, where V_{max} is measured as the maximum peak voltage and V_{min} is measured as the lowest valley, both occurring during a 3 ms time window.

A. Training and Testing Data Generation

Training data was generated using a star distribution, resulting in 9 training points. For generalization measurement, 3 random testing points were generated. The design region was constrained by the following lower and upper bounds: 100 to 250 for k_p and k_i; 250 to 450 for k_d; and 3 to 7 for N_{Cblk}. For training purposes, input data was normalized. Collecting this set of data from simulations took approximately 48 hours.

B. Surrogate Modeling

Once again, four different surrogate modeling techniques were implemented: PSM, Kriging, GRNN, and SVM. A summary of the resulting training and testing (generalization) errors are shown in Tables III and IV, respectively. From these results, it is observed that the Kriging model exhibits the best

[7] SIMetrix/Simplis 7.20e (x64), Copyright © 2014 Simplis Technologies Ltd, 78 Chapel Street, Thatcham, Berkshire, RG18 4QN, UK, http://www.simetrix.co.uk/site/index.html

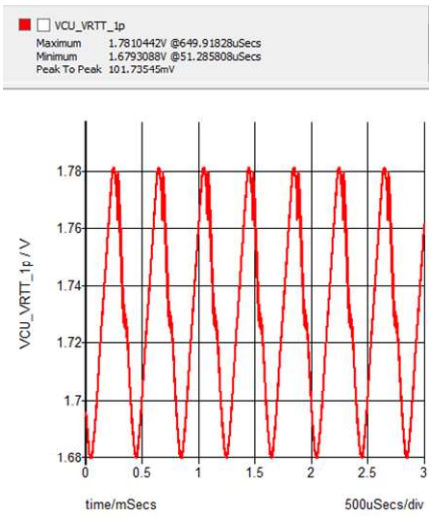

Fig. 10. SIMPLIS time domain simulation from 0 to 3 ms. In this case, V_{max} = 1.78 V, V_{min} = 1.68V, and peak to peak ripple noise is V_{p2p} = 101.73 mV, measured under a specific slew rate scenario.

generalization performance, selecting it as the PDN metamodel to perform direct optimization.

C. Surrogate-Based Optimization

The desired performance specifications are:

a) Lower bound for min. transient voltage, $V^{lb}_{min} = 1.65$ V
b) Upper bound for max. transient voltage, $V^{ub}_{max} = 1.85$ V
c) Peak to peak noise ripple, $V_{p2p} \leq 200$ mV.

For the surrogate-based optimization we used the initial design $\boldsymbol{x}^{(0)} = [175\ 350\ 175\ 5]^T$, which yields an initial surrogate system response $\boldsymbol{R}_s(\boldsymbol{x}^{(0)}) = [1.8270\ \ 1.6560]^T$ and an initial surrogate objective function value $U(\boldsymbol{R}_s(\boldsymbol{x}^{(0)})) = 0.0148$. By using the Nelder-Mead algorithm, the optimal design was found after 20 iterations and 105 surrogate model evaluations and corresponds to $\boldsymbol{x}^* = [162.7846\ \ 350.3494\ \ 189.9643\ \ 5]^T$. The optimal surrogate system response is $\boldsymbol{R}_s(\boldsymbol{x}^*) = [1.7810\ \ 1.6761]^T$ which leads to an optimal surrogate objective function $U(\boldsymbol{R}_s(\boldsymbol{x}^*)) = -0.01067$. The fine model system response at the optimal surrogate design is $\boldsymbol{R}_f(\boldsymbol{x}^*) = [1.782\ \ 1.678]^T$, yielding a fine model objective function $U(\boldsymbol{R}_f(\boldsymbol{x}^*)) = -0.0101$. These results confirm that the proposed methodology can be exploited to substantially speed up the design process by reducing the amount of fine model system simulations, finding an optimal design that achieves the performance specifications. The whole process required 13 system fine model simulations (12 for collecting training and testing data, plus one simulation to test the optimal design), instead of the hundreds of fine model evaluations that a direct optimization methodology could require. The engineer's expertise is used to define the metamodel design region as well as a suitable starting point. Once the metamodel is developed, the optimization process is conducted automatically with the proposed methodology, allowing engineers to focus their efforts on other aspects of the system design.

VI. CONCLUSIONS

We described a general methodology for surrogate-based analysis and design optimization of power delivery networks. We first propose a formulation for accurate and fast prediction of PDN performance by exploiting passive lumped models optimized by parameter extraction (PE) to fit PDN impedance profiles obtained from industrial laboratory measurements. This formulation allows protecting chip makers intellectual property. Next, we propose a metamodeling approach for efficient and reliable PDN design optimization exploiting support vector machines (SVM), polynomial surrogate modeling (PSM), generalized regression neural networks (GRNN), and Kriging. Our methodology allows accurate and fast optimization of PDN performance. Realistic industrial cases illustrate our general PDN analysis and design optimization methodology.

TABLE III
SUMMARY OF RELATIVE TRAINING ERRORS (%) USING DIFFERENT SURROGATE MODELS FOR PDN

output	e_{rGRNN}	e_{rPSM}	e_{rSVM}	$e_{rKriging}$
V_{max}	0.81	8.21	0.88	0.66
V_{min}	0.71	8.97	0.76	0.01

TABLE IV
SUMMARY OF RELATIVE TESTING ERRORS (%) USING DIFFERENT SURROGATE MODELS FOR PDN

output	e_{rGRNN}	e_{rPSM}	e_{rSVM}	$e_{rKriging}$
V_{max}	4.53	18.16	4.75	4.52
V_{min}	4.12	17.86	4.27	3.96

REFERENCES

[1] K. Rupp (2019). *42 Years of Microprocessor Trend Data* [Online]. Available: https://www.karlrupp.net/2018/02/42-years-of-microprocessor-trend-data/.

[2] A. Gonzalez - Universitat Politècnica de Catalunya, (2019). *Trends in Processor Arquitecture* [Online]. Avaliable: https://arxiv.org/ftp/arxiv/papers/1801/1801.05215.pdf.

[3] M. Swaminathan, J. Kim, I. Novak, and J. Libous, "Power distribution networks for system-on-package: status and challenges," *IEEE Trans. on Advanced Packaging*, vol. 27, no. 2, pp. 286–300, May 2004.

[4] Altera Corporation (2019), *AN 574-1.0: Printed Circuit Board (PCB) Power Delivery Network (PDN) Design Methodology* [Online]. Available: https://www.intel.com/content/dam/www/programmable/us/en/pdfs/literature/an/an574.pdf.

[5] H. Y. Loo, B. H. Oh, P. T. Oh, and E. K. Lee, "CPU package design optimization for performance improvement and package cost reduction," in *Int. Conf. on Electronic Materials and Packaging*, Kowloon, China, Dec. 2006, pp. 1-5.

[6] L. D. Smith, R. E. Anderson, D. W. Forehand, T. J. Pelc, T. Roy, "Power distribution system design methodology and capacitor selection for modern CMOS technology," *IEEE Trans. on Advanced Packaging*, vol. 22, no. 3, pp. 284-291, Aug. 1999.

[7] Y. Kim, K. Kim, J. Cho, J. Kim, K Kang, T. Yang, Y. Ra, and W. Paik, "Power distribution network design and optimization based on frequency dependent target impedance," in *IEEE Electrical Design of Advanced Packaging and Systems Symposium (EDAPS)*, Seoul, South Korea, Dec. 2015, pp. 89-92.

[8] M. Ramdani, E. Sicard, A. Boyer, S. B. Dhia, J. J. Whalen, T. H. Hubing, M. Coenen, and O. Wada, "The electromagnetic compatibility of integrated circuits—past, present, and future," *IEEE Trans. on Electromagnetic Compatibility*, vol. 51, no. 1, pp. 78-100, Feb. 2009.

[9] K. Bharath, E. Engin, M. Swaminathan, K. Uriu, T. Yamada, "Computationally efficient power integrity simulation for system-on-package applications," in *IEEE Design Automation Conf.*, San Diego, CA, Jun. 2007, pp. 612-617.

[10] R. J. R. Kumar and A. Jain, "Power loss and voltage regulation calculation in a radial system with distributed generations and voltage regulators," in *Int. Conf. on Energy, Power and Environment: Towards Sustainable Growth (ICEPE)*, Shillong, India, Jun. 2015, pp. 1-5.

[11] N. I. Tsygulev, V. K. Khlebnikov, and V. A. Shelest, "Algorithm for selection of automatic voltage regulator setting to reduce power losses," in *Int. Conf. on Industrial Eng., Applications and Manufacturing (ICIEAM)*, St. Petersburg, Russia, May 2017, pp. 1-4.

[12] B. Goral, C. Gautier and A. Amedeo, "Power delivery network simulation methodology including integrated circuit behavior," in *IEEE Workshop on Signal and Power Integrity (SPI)*, Turin, Italy, May 2016, pp. 1-4.

[13] A. E. Engin, K. Bharath, and M. Swaminathan, "Analysis for signal and power integrity using the multilayered finite difference method," in *IEEE Int. Symp. on Circuits and Systems*, New Orleans, LA, Jun. 2007, pp. 1493-1496.

[14] B. Zhao, S. Bai, S. Connor, M. Cocchini, D. Becker, M, Cracraft, A. Ruehli, B. Archambeault, and L. Drewniak, "System level power integrity analysis with physics-based modeling methodology," in *IEEE Symp. on Electromagnetic Comp., Signal Integrity and Power Integrity (EMC, SI & PI)*, Long Beach, CA, Aug. 2018, pp. 379-384.

[15] S. Kim, K. J. Han, S. Kang, and Y. Kim, "Analysis and reduction of voltage noise of multi-layer 3D IC with PEEC-based PDN and frequency-dependent TSV models," in *Int. SoC Design Conf. (ISOCC)*, Jeju, South Korea, Nov. 2014, pp. 124-125.

[16] Y. Cao, A. B. Kahng, J. Li, A. Roy, V. Srinivas, and B. Xu, "Learning-based prediction of package power delivery network quality," in *Asia and South Pacific Design Autom. Conf. (ASP-DAC)*, Tokyo, Japan, Jan. 2019, pp. 1-7.

[17] H. M. Torun, M. Swaminathan, A. Kavungal Davis and M. L. F. Bellaredj, "A global Bayesian optimization algorithm and its application to integrated system design," *IEEE Trans. on Very Large Scale Integration (VLSI) Systems*, vol. 26, no. 4, pp. 792-802, Apr. 2018.

[18] J. L. Chavez-Hurtado and J. E. Rayas-Sánchez, "Polynomial-based surrogate modeling of RF and microwave circuits in frequency domain exploiting the multinomial theorem," *IEEE Trans. Microwave Theory Techn.*, vol. 64, no. 12, pp. 4371-4381, Dec. 2016.

[19] F. E. Rangel-Patiño, J. L. Chávez-Hurtado, A. Viveros-Wacher, J. E. Rayas-Sánchez, and N. Hakim, "System margining surrogate-based optimization in post-silicon validation," *IEEE Trans. Microwave Theory Techn.*, vol. 65, no. 9, pp. 3109-3115, Sep. 2017.

[20] S. Koziel and L. Leifsson, *Surrogate-Based Modeling and Optimization: Applications in Engineering*, New York, NY: Springer, 2013.

[21] W. Xu, J. He, and D. Zhong, "Power delivery modeling for full switching voltage regulator on high performance computing system," in *IEEE Int. Symp. on Electromagnetic Compatibility*, Denver, CO, Aug. 2013, pp. 599-603.

[22] S. Koziel, X. S. Yang, and Q. J. Zhang, *Simulation-driven Design Optimization and Modeling for Microwave Engineering*, Covent Garden, London: Imperial College Press, 2013.

[23] J. E. Rayas-Sánchez and V. Gutiérrez-Ayala, "EM-based Monte Carlo analysis and yield prediction of microwave circuits using linear-input neural-output space mapping," *IEEE Trans. Microwave Theory Techn.*, vol. 54, no. 12, pp. 4528-4537, Dec. 2006.

[24] J. E. Rayas-Sánchez, J. L. Chavez-Hurtado, and Z. Brito-Brito, "Design optimization of full-wave EM models by low-order low-dimension polynomial surrogate functionals," *Int. J. Numerical Modelling: Electron. Networks, Dev. Fields.*, vol. 30, no. 3-4, e2094, May-Aug. 2017.

[25] D. C. Montgomery, *Design and Analysis of Experiments*, Hoboken, NJ: John Wiley & Sons, 2012.

[26] S. Haykin, *Neural Networks: A Comprehensive Foundation*. New Jersey, MA: Prentice Hall, 1999.

[27] J. E. Rayas-Sánchez, J. Aguilar-Torrentera and J. A. Jasso-Urzúa, "Surrogate modeling of microwave circuits using polynomial functional interpolants," in *IEEE MTT-S Int. Microwave Symp. Dig.*, Anaheim, CA, May 2010, pp. 197-200.

[28] D. F. Specht, "A general regression neural network," *IEEE Trans. Neural Netw.*, vol. 2, no. 6, pp. 568–576, Nov. 1991.

[29] B. N. Panda, M. V. A. R. Bahubalendruni, and B. B. Biswal, "Optimization of resistance spot welding parameters using differential evolution algorithm and GRNN," in *Proc. IEEE Int. Conf. Intell. Syst. Control (ISCO)*, vol. 2. Coimbatore, India, Jan. 2014, no. 6, pp. 50–55.

[30] N. T. Tokan and F. Günes, "Analysis and synthesis of the microstrip lines based on support vector regression," in *Proc. IEEE Eur. Microw. Conf. (EuMC)*, Amsterdam, The Netherlands, Oct. 2008, pp. 1473–1476.

[31] L. Xia, J. Meng, R. Xu, B. Yan, and Y. Guo, "Modeling of 3-D vertical interconnect using support vector machine regression," *IEEE Microw. Wireless Compon. Lett.*, vol. 16, no. 12, pp. 639–641, Dec. 2006.

[32] G. Angiulli, M. Cacciola, and M. Versaci, "Microwave devices and antennas modelling by support vector regression machines," *IEEE Trans. Magn.*, vol. 43, no. 4, pp. 1589–1592, Apr. 2007.

[33] W. C. M. Van Beers, "Kriging metamodeling in discrete-event simulation: An overview," in *Proc. Simulation Conf.*, Dec. 2005, pp. 202–208.

[34] F. J. Leal-Romo, J. L. Chavez-Hurtado, and J. E. Rayas-Sanchez, "Selecting surrogate-based modeling techniques for power integrity analysis," in *IEEE MTT-S Latin American Microwave Conf. (LAMC-2018)*, Arequipa, Peru, Dec. 2018, pp. 1-3.

[35] W. Xu, J. Fang, J. He, and T. Kim, "Switching voltage regulator modeling and its applications in power delivery design," in *IEEE Int. Symp. on Electromag. Comp.*, Raleigh, NC, Aug. 2014, pp. 855-860.

[36] A. Singh and Q. Zhou, "Server platform power design optimization using switching voltage regulator modeling techniques," in *Int. Conf. on Electronic Packaging Technology (ICEPT)*, Wuhan, China, Aug. 2016, pp. 1497-1502.