Design of Multi-Output Switched-Capacitor Voltage Regulator via Machine Learning

Zhiyuan Zhou*, Syrine Belakaria*, Aryan Deshwal, Wookpyo Hong
Janardhan Rao Doppa, Partha Pratim Pande and Deukhyoun Heo
School of EECS, Washington State University, Pullman, WA, U.S.A.

Abstract— **Efficiency of power management system (PMS) is one of the key performance metrics for highly integrated system on chips (SoCs). Towards the goal of improving power efficiency of SoCs, we make two key technical contributions in this paper. First, we develop a multi-output switched-capacitor voltage regulator (SCVR) with a new flying capacitor crossing technique (FCCT) and cloud-capacitor method. Second, to optimize the design parameters of SCVR, we introduce a novel machine-learning (ML)-inspired optimization framework to reduce the number of expensive design simulations. Simulation shows that power loss of the multi-output SCVR with FCCT is reduced by more than 40% compared to conventional multiple single-output SCVRs. Our ML-based design optimization framework is able to achieve more than 90% reduction in the number of simulations needed to uncover optimized circuit parameters of the proposed SCVR.**

Keywords— Power management system, Electronic design automation, Machine Learning, Multi-objective Optimization

I. INTRODUCTION

Modern system on chips (SoCs) integrate multiple diverse sub-blocks, and pose significant challenges in designing efficient and compact power management system (PMS) for them. To overcome these challenges, on-chip voltage regulators are proposed as a promising solution due to their ability to provide good voltage switching and power delivery [1-5]. However, fully integrated switched-inductor voltage regulators (SIVRs) suffer from the limitation of on-chip inductors, e.g., poor quality factor, and low power density of SIVRs [1]. Low-dropout regulators (LDOs) can achieve high power conversion efficiency only within a narrow output voltage range [1], which limits their applicability in the wide output voltage range.

Recent on-chip capacitors provide better power density when compared to on-chip inductors [1,5]. Fully integrated switched-capacitor voltage regulators (SCVRs) have gained popularity as they can generate wide output voltage range for a specific input [1-3]. In order to resolve performance-efficiency trade-off, various sub-blocks in SoCs, e.g., digital, analog, and mixed-signal blocks, are required to achieve optimal performance under different voltages [4]. Conventional SCVR-based PMSs that provide multiple supply voltages are composed of several *independent* single-output SCVRs [2-4]. However, this is not an optimized solution to design PMS with compact size and high power efficiency. Some recent works proposed dual-output SCVRs (DOSCVRs) with the benefits of reducing power electronic components [2], wide input and output voltage range [3], and increased power efficiency [4]. However, these DOSCVRs can only improve the power efficiency when two load currents are different and cannot handle wide range of output voltages and large load currents. To improve the efficiency of SCVR for the same and different load current conditions of multiple outputs, we propose a novel *flying capacitor crossing technique (FCCT)* in this paper. The new technique can improve efficiency without increasing the area of the chip, when compared with multiple conventional single-output SCVRs. Thus, this improved efficiency comes with no additional cost. Besides, the *on-chip flying capacitors are shared "in the cloud"* and will be assigned appropriately to fulfill the time-varying power demands of load tied to a specific output. Hence, this mechanism guarantees the most efficient utilization of limited on-chip flying capacitors.

The design and optimization of PMS pose significant challenges in finding SCVR circuit parameters (e.g., size of capacitors, power of MOSFETs) to optimize the overall performance [6,7,8]. First, the design space of circuit parameters is very large to explore via trial-and-error. Second, evaluation of candidate circuit configurations in terms of multiple optimization objectives involves performing computationally-expensive simulations. Third, circuit designs should satisfy some constraints (e.g., output ripple, output voltage, and efficiency) that cannot be verified without simulating the circuit. Fourth, the optimal design parameters of the PMS circuit vary depending on the target output requirements. Prior work on PMS circuit optimization is lacking on many fronts. These methods primarily focus on PMS system-level optimization [6]. Experimental data used by these approaches come from past PMS design analysis in the literature. They ignore the performance variation of the PMS circuit. For example, the data from existing PMS design analysis is accurate only for the same design environment such as technology process, input/output voltage, and load conditions, etc. Also, the optimization procedure needs to be performed from scratch in the traditional circuit design methodology if the specification of the PMS changes [1], which incurs huge design cost.

* Equally-contributing first authors.

This work is supported by gift funding from Intel Corporation and by National Science Foundation grant IIS-1845922

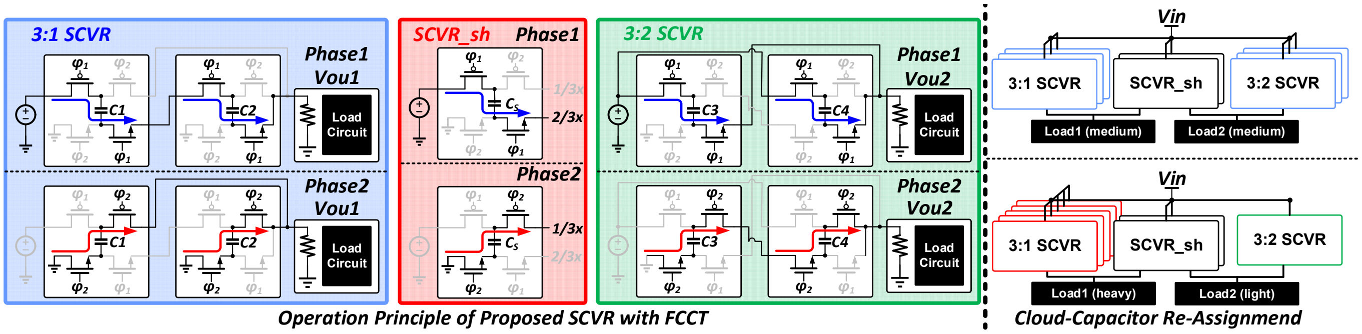

Fig. 1. Power stage of the proposed multi-output SCVR and an example of two-output SCVR design with cloud-capacitor technique.

To overcome the above-mentioned challenges for PMS circuit design optimization, this paper introduces a novel machine learning (ML) algorithm referred as **U**ncertainty **R**eduction via **M**ultiple **A**cquisition **C**onstrained (URMAC) for *multi-objective* (MO) circuit design optimization algorithm with *constraints*. Output voltage ripple and efficiency are two main objectives for PMS circuits. We need to find the Pareto optimal set of circuit designs that satisfy the specified constraints. A design is called *Pareto optimal* if it cannot be improved in any of the objectives without compromising some other objective [12,13]. The overall goal is to approximate the true constrained Pareto set by minimizing the number of expensive circuit simulations. The key idea behind URMAC is to build a cheap surrogate statistical model (e.g., Gaussian Process [10]) using the real circuit simulations data; and employ it to intelligently select the sequence of candidate circuit designs for undertaking thorough simulations. URMAC employs a two-stage search procedure to improve the accuracy and computational-efficiency of sequential decision-making under uncertainty for selecting candidate circuit designs for evaluation. First, it solves a cheap constrained MO optimization problem defined using the statistical models (one for each unknown objective) to identify a list of promising candidates. Second, it selects the best candidate from this list based on a measure of uncertainty.

II. Proposed Multi-Output SCVR

A. *Power Stage with FCCT and Cloud-Capacitor*

In order to fully utilize the limited on-chip flying capacitors, the FCCT technique is proposed in this paper. The power stage with FCCT is shown in Fig. 1. The SCVR in Fig.1 can simultaneously generate two output voltages with 1/3x and 2/3x of the input voltage. The power stage in the proposed approach with FCCT is composed of three sub-blocks – (i) a 3:1 SCVR, (ii) a 3:2 SCVR, and (iii) an SCVR_sh, while the conventional SCVR only consists of the 3:1 SCVR and the 3:2 SCVR. The 3:1 SCVR generates V_{out1} of 1/3x of input voltage and the 3:2 SCVR generates V_{out2} of 2/3x of the input voltage. The FCCT is achieved by the additional sub-SCVR of SCVR_sh, which is composed of four power switches and one flying capacitor, C_{s}. The SCVR works in two phases as illustrated in Fig. 1. The 3:1 SCVR and 3:2 SCVR work similar to the conventional SCVR: the flying capacitors are charged in one phase and discharged in the other phase [1]. For the SCVR_sh, the two terminals of the flying capacitor C_{s} are connected to the input voltage source and V_{out2} in phase1, and connected to Gnd and V_{out1} in phase 2. So that C_{s} is charged in phase1 and discharged in phase2. The voltage across C_{s} is regulated at 1/3x of the input voltage. The blue arrows in Fig. 1 illustrate the charge flow in phase1, and the red arrows illustrate the charge flow in phase2. The additional SCVR_sh provides a parallel current path to both outputs. This path helps in reducing the portion of charges that each flying capacitor carries to the output. The reduction in the flowing charges of the flying capacitors, in turn, decreases the power loss related to the charging and the discharging of the capacitors, i.e., charge redistribution loss. Moreover, the charge passing through the resistive elements also decreases. Hence, resistive conduction loss is reduced. Compared with the conventional approach, the proposed FCCT uses an additional capacitor, C_{s}, but the total capacitance is kept constant by reducing the other capacitance values to compensate for the additional C_{s}. Thus, there will be no additional area overhead. We also provide the analysis of the proposed SCVR with FCCT to demonstrate that the power loss reduction increases the efficiency without additional penalty.

Another issue of the conventional multi-output SCVR is the wasting of flying capacitor resource when the load condition of SCVR differs from that of the SCVR's highest efficiency point. This is because of the power stage in the conventional SCVR cannot adjust to the load variation. In this work, a cloud-capacitor technique is proposed to address the issue. The on-chip flying capacitors are shared "in the cloud": the flying capacitors will be assigned to meet time-varying power demands of load tied to a specific output. Hence, this mechanism guarantees the most efficient utilization of limited on-chip flying capacitors [1,4]. For example, right part of the Fig. 1 stands for the flying-capacitor assignment under different loads. The total on-chip flying capacitors consist of eight units of flying capacitor. The above portion of the right part of Fig. 1 shows the case that load1 and load2 are under medium load condition. Both of the loads have three units of flying capacitors to support the loads' power requirement. The bottom part shows the case of load1 is under heavy load, while load2 is under light load conditions. Thus, five units of flying capacitors are used for powering load1 and only one unit of the flying capacitor is used for powering load2. The below analysis will illustrate the value of the capacitor in SCVR_sh also needs to be re-assigned based on varying load conditions. This complicated fine-tuning of various parameters increases the challenge of PMS design, resulting in sub-optimal PMS hardware.

B. Analysis of Power Loss

Output impedance of the SCVR closely affects power loss and power efficiency. A reduction in output impedance decreases power loss, thus increasing efficiency. For the proposed SCVR, the power loss based on a calculated output impedance is compared with the power loss of the conventional approach. The trans-impedance model [5] is used to calculate the slow-switching limit (SSL) impedance and fast-switching limit (FSL) impedance of the SCVR with FCCT in Fig. 2. SSL impedance (Z_{SSL}) represents the charge redistribution loss and is obtained by the power loss of flying capacitors. FSL impedance (Z_{FSL}) indicates the conduction loss on the resistance of the switches and is obtained by the losses produced in all switches. These two losses contribute to the majority portion of the power loss of the SCVR.

$$P_{loss(C_1-C_S)} = \frac{fsw a^2}{C_1} + \frac{fsw a^2}{C_2} + \frac{fsw b^2}{C_3} + \frac{fsw b^2}{C_4} + \frac{fsw s^2}{C_S} \quad (1)$$

$$P_{loss(S)} = \frac{1}{2} \cdot \sum R_i (a \cdot 2 f_{SW})^2 + \frac{1}{2} \cdot \sum R_i (b \cdot 2 f_{SW})^2 + \frac{1}{2} \cdot \sum R_i (s \cdot 2 f_{SW})^2 \quad (2)$$

where $P_{loss(C_1-C_S)}$ is the loss due to Z_{SSL} and $P_{loss(S)}$ is the loss due to Z_{FSL}. The f_{SW} presents the switching frequency; R_i is the turn-on resistance of switch *i*; a represents charge flowing through the flying capacitors C_1 and C_2; b represents charge flowing through the flying capacitors C_3 and C_4 ; and s represents charge flowing through the flying capacitors C_S.

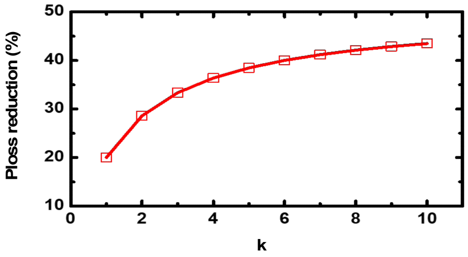

Fig. 2. Relationship between the power loss ($P_{loss(C1-CS)}$) reduction and *k*.

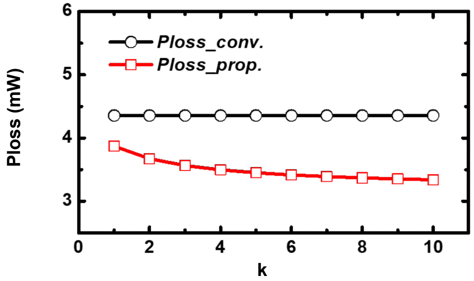

Fig. 3. Relationship between the power loss ($P_{loss(S)}$) and *k*. Comparison between the conventional and the proposed approach when I_1=I_2.

Based on the calculated Z_{SSL}, the power loss $P_{loss(C_1-C_S)}$ of the proposed SCVR is compared with the power loss of the conventional approach by simulation. The maximum power loss reduction versus k is illustrated in Fig. 2, where k is a multiplication factor to represent the size of C_S compared with C_1. Here, we set the same value to C_1, C_2, C_3, and C_4. For fair comparison, all of the simulations are performed under the condition of same total flying capacitor value. The total value of C_1, C_2, C_3, and C_4 in conventional SCVR is the same as the total value of C_1, C_2, C_3, C_4, and C_S in the SCVR with FCCT. As the value of k increases, we can reduce more power loss when compared with the conventional approach. This power loss reduction is maximized when both output currents are the same due to C_S. Based on the calculated Z_{FSL}, the power loss $P_{loss(S)}$ of the proposed approach is also presented in Fig. 3, when $I_1 = I_2$. As shown in the Fig. 2 and 3, the power loss of the multi-output SCVR is reduced by more than 40% compared to conventional multiple single-output SCVRs.

III. Machine Learning Algorithm for Multi-Objective Circuit Design Optimization

A. Circuit Design Space of the Proposed SCVR

We choose a four-output SCVR to optimize the parameters of the proposed multi-output SCVR with FCCT and cloud-capacitor method. This SCVR circuit generates two outputs of V_{out1} and V_{out2} at 1/3x of the input voltage, and the other two outputs of V_{out3} and V_{out4} at 2/3x of the input voltage. There will be eight sub-SCVRs: two 3:1 SCVRs for V_{out1} and V_{out2}, two 3:2 SCVRs for V_{out3} and V_{out4} , and four SCVR_shs (connected between V_{out1} and V_{out3}, V_{out1} and V_{out4}, V_{out2} and V_{out3}, V_{out2} and V_{out4} respectively). Our proposed ML algorithm aims to optimize the performance of PMS by finding the optimized values of flying capacitors in these sub-SCVRs based on the load requirements (output voltages and output currents) by keeping the total capacitor value constant using the cloud-capacitor method. Let $C_1, C_2, C_3, C_4, C_{13}, C_{14}, C_{23}, C_{24}$ stand for the capacitor values in the eight sub-SCVRs. Since our aim is to optimize the circuit under real-conditions, the SCVR is implemented at transistor-level in the TSMC 180-nm CMOS technology. Each value of those metal insulator metal (MIM) capacitors is dependent on three variables: unit capacitor width W_i, unit capacitor length L_i, and the number of unit capacitors M_i , for each $i \in \{1,2,3,4,13,14,23,24\}$. W_i and L_i are set within the range of 5um to 30um; and M_i is set to within the range of 1 to 999. The power switches and the driver buffers are also sized according to the geometry of MIM capacitors. Additionally, our optimization involves the four reference voltages ref_v and four load resistances r, which are the most important design variables. The reference voltages are within the range of 500 mV to 630 mV and 1.05 V to 1.23 V due to the conversion ratios of 3:1 and 3:2 under the input voltage of 1.8V. The load resistance is set within 10 Ohms to 2000 Ohms for the maximum output power of the SCVR to be around 300 mW. Since we use output voltages and load currents as design variables, the trained model can generate the optimized SCVR circuit for any output voltages and load currents conditions. Thus, there are 32 design variables in total.

B. Multi-Objective Constrained Optimization Problem

Given the circuit design space $\mathcal{X}$, a circuit simulator that can evaluate candidate designs, and a set of constraints *CS,* our goal is to closely approximate the optimal Pareto set $\mathcal{X}^* \subset \mathcal{X}$ by minimizing the number of circuit simulations. A solution is called *Pareto optimal* if it cannot be improved in any of the objectives without compromising some other objective [12,13]. In our specific case, each circuit simulation with a candidate

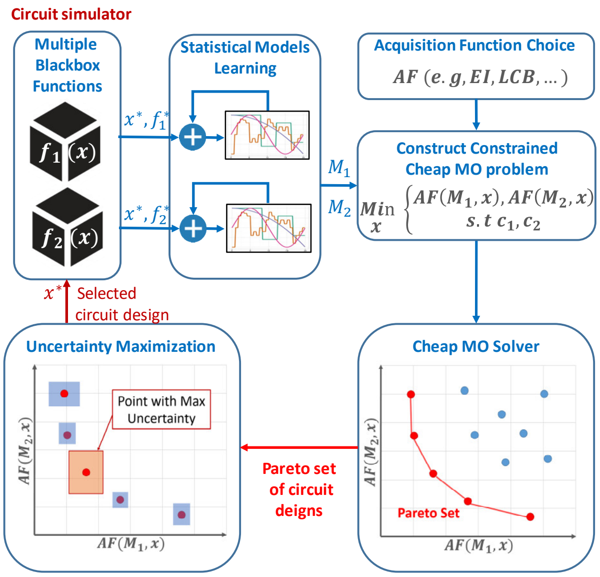
Circuit simulator
Multiple Blackbox Functions
$f_1(x)$
$f_2(x)$
x^*, f_1^*
x^*, f_2^*
Statistical Models Learning
Acquisition Function Choice
$AF\ (e.g, EI, LCB, \dots)$
Construct Constrained Cheap MO problem
M_1
M_2
$\underset{x}{Min} \begin{cases} AF(M_1, x), AF(M_2, x) \\ s.t\ c_1, c_2 \end{cases}$
x^* Selected circuit design
Uncertainty Maximization
Point with Max Uncertainty
$AF(M_2, x)$
$AF(M_1, x)$
Pareto set of circuit deigns
Cheap MO Solver
Pareto Set
$AF(M_2, x)$
$AF(M_1, x)$

Fig 4. Example illustration of URMAC framework for two objectives (e.g., efficiency and voltage ripple).

Algorithm 1 : URMAC Framework

```
Input: $\mathcal{X}$ input design space; $F_1(x), F_2(x), \cdots, F_k(x)$, $k$ blackbox objective functions; $AF$ acquisition function; $T_{max}$ maximum no. of iterations; $CS$ set of constraints
Initialize training data of function evaluations $\mathcal{D}$
Initialize statistical models $M_1, M_2, \cdots, M_k$ from $\mathcal{D}$
for each iteration $t = 1$ to $T_{max}$
    //Solve constrained cheap MO problem with objectives to get candidate circuit designs
    $\mathcal{X}_p \leftarrow min_{x \in \mathcal{X}} \ (AF(M_1, x), \cdots, AF(M_k, x))$
        $s.t\ cs_0, \cdots, cs_9$
    // Pick the candidate design with maximum uncertainty
    Select $x_{t+1} \leftarrow \arg max_{x \in \mathcal{X}_p} U_{\beta_t}(x)$
    Evaluate $x_{t+1}$: $Y_{t+1} \leftarrow (F_1(x_{t+1}), \cdots, F_k(x_{t+1}))$
    Aggregate data: $\mathcal{D} \leftarrow \mathcal{D} \cup \{(x_{t+1}, Y_{t+1})\}$
    Update models $M_1, \cdots, M_k$ using $\mathcal{D}$
    $t \leftarrow t + 1$
end for
return Pareto set and Pareto front of $\mathcal{D}$
```

combination of the design variables will result in the following objective function evaluations: four output ripples, four output voltages, and efficiency of the circuit. Our goal is to find the circuit designs that will maximize the output voltages and efficiency, and minimize the output ripples while satisfying the following set of constraints CS.

Constraint cs_0: $C_{total} \simeq 20\ uF$
Constraint $cs_1$to cs_4: $Ov_i \geq ref_{v_i}$ for all $i \in \{1 \cdots 4\}$
Constraint cs_5 to cs_8 : $Or_{lb} \leq Or_i \leq Or_{ub}$ for all $i \in \{1 \cdots 4\}$
Constraint cs_9: $efficiency \leq 100\%$

where Ov is the output voltage, ref_v is the associated reference voltage, the optimized SCVR will be able to generate output voltages higher than the reference voltages; Or is the output ripple, Or_{ub} and Or_{lb} are the output ripple upper bound and lower bound respectively. With the requirement of high-performance computing cores as loads of SCVR, we set the constraint for output voltage ripples within the range of 0-100 mV. We set the value of total on-chip flying capacitors to 20 uF due to the limited chip area in practice.

C. Uncertainty Reduction via Multiple Acquisition Constrained (URMAC) Optimization Algorithm

We propose a novel constrained multi-objective optimization approach referred as URMAC to optimize the parameters of our novel SCVR circuit. The main challenge in solving this optimization problem is that the evaluation of each candidate circuit design involves performing computationally-expensive simulation. The key insight behind our URMAC algorithm is to build statistical models [10] using the circuit designs evaluated in the past and employ them to intelligently select the next circuit design for evaluation to be able to reduce the uncertainty of our models. A simple illustration of URMAC with two objectives, namely, efficiency and voltage ripple, is shown in Fig. 4. URMAC is an iterative algorithm, which involves four key components (steps) as described below. For the example in figure, we used efficiency and ripple as two black-box objective functions to be optimized over the design space of SCVR circuit parameters. It is noteworthy that two objectives are used here for example, the real framework will include nine objectives that were defined in the problem definition section earlier.

1. Learning statistical models: we build statistical models (e.g., Gaussian Processes [10]) $M_1, \cdots, M_9$ for the nine objective functions corresponding to efficiency, output voltages, and output ripples from the training data in the form of past circuit design evaluations. These statistical models can predict the output of unknown circuit designs and also quantify their uncertainty for those predictions. In each iteration of URMAC, the learned statistical models are employed to select the next candidate circuit design for evaluation via the circuit simulator.

2. Constructing a constrained cheap MO problem: We select a set of promising candidate circuit designs for evaluation by solving a constrained cheap multi-objective (MO) optimization problem defined using the statistical models $M_1, \cdots, M_9$. The objective functions in this optimization problem are *cheap* because they rely on the learned statistical models instead of expensive simulations. We can employ a wide variety of acquisition functions from the literature to construct this cheap MO problem. An acquisition function (*AF*) parameterized by the statistical model is a cheap scoring function that measures the utility of evaluating a candidate circuit design for optimizing the corresponding black-box objective function. Some example acquisition functions include upper confidence bound (UCB) [11], expected improvement (EI), and Thompson sampling (TS). In the example illustrated in Fig. 4, we will have two cheap objective functions for a specified acquisition function *AF*: *AF(M₁)* and *AF(M₂)*.

$$UCB(x) = \mu(x) + \beta^{1/2}\sigma(x) \quad (3)$$

$$LCB(x) = \mu(x) - \beta^{1/2}\sigma(x) \quad (4)$$

$$EI(x) = \sigma(x)\big(\alpha\Phi(\alpha) + \phi(\alpha)\big) \quad (5)$$

where $\alpha = \frac{\tau - \mu(x)}{\sigma(x)}$, $\mu(x)$, and $\sigma(x)$ correspond to the mean and standard deviation of the prediction from the statistical model, and represent exploitation and exploration scores respectively; β is a parameter that balances exploration and exploitation which is automatically specified depending on the iteration number t based on the theoretical analysis of convergence from [11]; τ is the best-uncovered function value; and Φ and ϕ are the CDF and PDF of normal distribution respectively.

3. Solving the constrained cheap MO problem: We use the popular NSGA-II algorithm [13] in its constrained version to solve the cheap MO problem to obtain the Pareto set representing the most promising candidate circuit designs for evaluation that satisfies the constraints. The constraints for our optimization problem are defined as follows:

Constraint $cs_1 \cdots cs_4$: $\mu_{Ov_i}(x) \leq ref_{vi}$, for all $i \in \{1 \cdots 4\}$
Constraint $cs_5 \cdots cs_8 : Or_{lb} \leq \mu_{Or_i}(x) \leq Or_{ub}$, for all $i \in \{1 \cdots 4\}$
Constraint cs_9 : $\mu_{efficiency}(x) \leq 100\%$

The acquisition functions for each black-box objective function tells us the utility of evaluating a candidate circuit design. For example, a design might have a high utility for efficiency but may have a lower utility for optimizing the output ripple. Therefore, the Pareto set found by solving the constrained cheap MO problem captures the trade-off between the utility for all objective functions by satisfying the constraints.

$$\mathcal{X}_{\wp} \leftarrow \min_{x \in \mathcal{X}} \left(AF(M_1, x), \cdots, AF(M_k, x)\right)$$
$$s.t\ cs_0, \cdots, cs_9$$

4. Uncertainty maximization: From the promising candidate designs obtained by solving the constrained cheap MO problem, we need to select the best circuit design such that it will guide the overall search towards the goal of quickly approximating the true Pareto set. We employ an uncertainty measure defined in terms of the statistical models to select the most promising candidate circuit design for evaluation. We define the uncertainty measure as the volume of the uncertainty hyper-rectangle (via lower-confidence bound and upper-confidence bound obtained from statistical models). We compute the uncertainty measure for each promising candidate circuit design obtained from step 3 as follows.

$$U_{\beta_t}(x) = VOL\left(LCB(M_i, x), UCB(M_i, x)_{i=1}^{k}\right)$$

We select the one with maximum uncertainty for evaluation via circuit simulation.

$$x_{t+1} = \arg m\, ax_{x \in \mathcal{X}_{\wp}}\ U_{\beta_t}(x)$$

Finally, the selected circuit design is used for evaluation via simulator to get the corresponding evaluations for different objective functions. The next iteration starts after the statistical models are updated using the new training example (input is the circuit design and output is a vector of objective evaluations).

IV. Experimental Results

A. *Effectiveness of ML-based Circuit Design Optimization*

CADENCE Simulation Setup. We build the circuit schematic in Cadence simulation tool in TMSC 180-nm CMOS technology. The schematic consists of the SCVR power stage, controller, and test bench schematic. All design variables are set as the parameters of the circuit component symbols. All the circuit performance objectives are plotted, measured, and calculated from Cadence to ensure the accuracy of the results.

Baselines. We compare our URMAC algorithm with the state-of-the-art multi-objective evolutionary algorithms *NSGA-II* [13] and *MOEA/D* [14]. We also compare to an evaluation over a *grid* sampled uniformly at random from the given design space. NSGA-II evaluates the objective functions at several input designs and sorts them into a hierarchy of sub-groups based on the ordering of Pareto dominance. The similarity between members of each sub-group and their Pareto dominance is used by the algorithm to move towards more promising parts of the input space. MOEA/D decomposes a multi-objective optimization problem into a number of scalar optimization sub-problems and optimizes them simultaneously. Each sub-problem is optimized by only using the information from its neighboring sub-problems. We employ the NSGA-II and MOEA/D code from the known python library *Platypus*.

Prior work has proposed surrogate models based optimization methods in the context of circuit optimization [13,14]. However, none of these algorithms consider constrained optimization setting. Consequently, we cannot compare URMAC with these methods in a fair manner.

Setup for URMAC. We employ a Gaussian process (GP) based statistical model with squared exponential (SE) kernel in all our experiments. The SE kernel is defined as $\kappa(x, x') = s \cdot exp\left(\frac{-|x-x'|^2}{2\sigma^2}\right)$, where s and σ correspond to scale and bandwidth parameters. These hyper-parameters are estimated after every 10 function evaluations. We initialize the GP model using five inputs chosen randomly.

Evaluation Metrics. To measure the performance of baselines and URMAC, we employ two different metrics, one measuring the accuracy of solutions and another one measuring the efficiency in terms of the number of simulations. *1) Pareto hypervolume (PHV)* is a commonly employed metric to measure the quality of a given Pareto front [12]. PHV is defined as the volume between a reference point and the given Pareto front. After each iteration t (or the number of circuit simulations), we measure the PHV for all algorithms. We evaluate all algorithms for 100 circuit simulations.

2) Percentage gain in simulations is the fraction of simulations our ML-based optimization algorithm (URMAC) is saving to reach the PHV accuracy of solutions at the convergence point of baseline algorithm employed for comparison.

	Grid	MOEAD	NSGA-II
% gain in simulations	94.8%	90.7%	93.3%

Table 1: Percentage gain in simulations achieved by our URMAC algorithm when compared with each baseline.

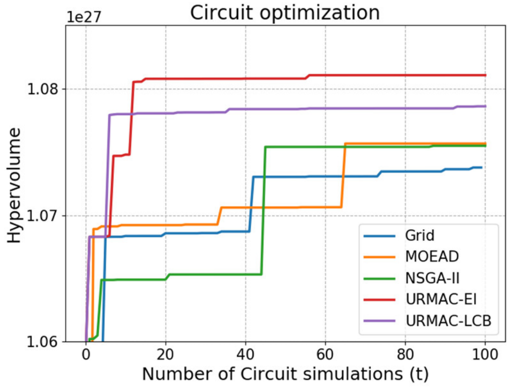

Fig 5: Results of different multi-objective algorithms including URMAC. The hypervolume metric is a function of the number of circuit design simulations.

Results and Discussion. We evaluate the performance of URMAC with two different acquisition functions (EI and LCB) to show the generality and robustness of our approach. We also provide results for the percentage gain in simulations achieved by URMAC when compared to each baseline method in Table 1. Fig. 5 shows the PHV metric achieved by different multi-objective methods including URMAC as a function of the number of circuit simulations. We make the following observations: 1) URMAC with both EI and LCB acquisition functions perform significantly better than all baseline methods. 2) URMAC produces better quality Pareto designs than all baselines using less number of circuit simulations. 3) URMAC is able to uncover the best Pareto solutions from baselines using significantly less number of circuit simulations. This result shows the efficiency of our ML based optimization approach. URMAC achieves percentage gain in simulations w.r.t baseline methods ranging from 90% to 95%. The SCVR is implemented in the industry-provided process design kit (PDK) and shows better efficiency and output ripples. Due to the huge number of parameters and design specs in the analog circuit design optimization, traditional methods will be very expensive. Our results show huge practical benefits in terms of faster convergence and better quality Pareto designs. Since MOEAD is the best performing baseline optimization method, we use it for the rest of the experimental analysis.

B. *Quality of Optimized SCVR Circuits via ML-based Circuit Design Optimization*

Table 2 illustrates the simulated performance of four-output SCVR with FCCT and cloud-capacitor method optimized by MOEAD (best baseline) and URMAC-EI (best variant of our proposed algorithm). $V_{ref(1-4)}$ and $R_{(1-4)}$ are the reference voltages and load resistances of the four outputs. $V_{(o1-o4)}$ and OR(1-4) are simulated output voltages and ripples. "*Eff*" stands for the overall efficiency of the four-output SCVR (total output power vs. input power). Results of both algorithms meet the voltage reference and ripple requirements (100mV) . Compared to MOEAD, the optimized SCVR with URMAC-EI can achieve a higher conversion efficiency of 76.2% (5.25% higher than MOEAD, highlight in red color) with similar output ripples. The optimized SCVRs can generate the target output voltages within the range of 0.52V-0.61V (1/3x ratio) and 1.07V-1.12V (2/3x ratio) under the loads varying from 14Ohms to 1697Ohms (highlight in black and green colors). Thus, the capability of URMAC to optimize the parameters of SCVR under different output voltage/current conditions is clearly verified. Importantly, our ML-based optimization framework combined with the proposed SCVR provides a scalable solution to find optimized on-chip PMS designs for complex high-performance computing systems.

V. Conclusions

This paper studied a novel multi-output SCVR combined with a flying capacitor crossing technique (FCCT), cloud-capacitor method, and a novel ML-based circuit design optimization framework towards the goal of improving the efficiency of PMS for highly integrated SoCs. Results show that power loss of the proposed SCVR is reduced by more than 40% when compared to conventional multiple single-output SCVRs. Our ML-based circuit design optimization framework is able to achieve more than 90% reduction in the number of simulations needed to find optimized circuit parameters of the proposed SCVR, and is also able to uncover significantly efficient circuit designs when compared to baseline optimization algorithms.

SPECS	MOEAD			URMAC-EI (Proposed)		
V_{ref1}(V)	0.53	0.6	0.6	*0.52*	0.53	0.56
V_{ref2}(V)	0.55	0.51	0.59	*0.55*	0.61	0.57
V_{ref3}(V)	1.14	1.06	1.07	*1.07*	1.12	1.11
V_{ref4}(V)	1.22	1.16	1.14	*1.09*	1.06	1.1
R_1(Ohm)	144	1668	1012	**207**	**1198**	**619**
R_2(Ohm)	758	620	559	**306**	**1697**	**89**
R_3(Ohm)	247	66	10	**67**	**1379**	**70**
R_4(Ohm)	222	144	1830	**42**	**14**	**301**
V_{o1}(mV)	551.5	702.18	775.01	**677.10**	**760.60**	**656.9**
V_{o2} (mV)	612.2	671.01	912.22	**690.70**	**725.70**	**569.4**
V_{o3} (V)	1.17	1.09	867.96	***1.08***	**1.15**	**1.14**
V_{o4} (V)	1.117	1.12	1.12	***1.08***	0.99	**1.13**
OR1(mV)	52.69	11.39	0.496	2.50	4.20	11.30
OR2(mV)	9.32	4.52	0.984	3.50	5.00	15.90
OR3(mV)	64.96	96.57	28.199	58.9	87.1	75.7
OR4(mV)	4.75	4.80	1.05	80.7	25.1	74.3
Eff (%)	70.95	65.94	64.61	***76.2003***	***74.82***	***73.71***

Table 2: Comparison table of optimized four-output SCVR parameters obtained by MOEAD and URMAC-EI implemented in TSMC 180nm CMOS technology. (designs are selected from the Pareto set prioritized by efficiency)

References

[1] H. Le *et al.*, "Design Techniques for Fully Integrated Switched-Capacitor DC-DC Converters," in *ISSCC*, pp.2120-2131, 2011.

[2] Z. Hua *et al.*, "A reconfigurable dual-output switched-capacitor DC-DC regulator with sub-harmonic adaptive-on-time control for low-power applications," in *IEEE JSSC*, vol. 50, no. 3, pp. 724–736, March. 2015.

[3] C. K. Teh *et al.*, "A 2-Output Step-Up/Step-Down Switched-Capacitor DC-DC Converter with 95.8% Peak Efficiency and 0.85-to-3.6V Input Voltage Range," in *ISSCC*, pp. 222-223, 2016.

[4] J. Jiang *et al.*, "20.5 A dual-symmetrical-output switched-capacitor converter with dynamic power cells and minimized cross regulation for application processors in 28nm CMOS," in *ISSCC*, pp. 344-345, 2017.

[5] J. Delos *et al.*, "On the modeling of switched capacitor converters with multiple outputs,"in *APEC*, pp. 2796-2803, 2014.

[6] L. Wang *et al.*, "Optimizing the Energy Efficiency of Power Supply in Heterogeneous Multicore Chips with Integrated Switched-Capacitor Converters," *in DATE*,pp. 836-841, 2019.

[7] X. Mi *et al.*, "Flying and decoupling capacitance optimization for area-constrained on-chip switched-capacitor voltage regulators," *in DATE*, pp. 1269-1272, 2017.

[8] P. A. Beerel *et al.*, "Opportunities for Machine Learning in Electronic Design Automation*," in IEEE ISCAS*, Florence, 2018, pp. 1-5.

[9] Platypus: A free and open source Python library for multi-objective optimization. https://github.com/Project-Platypus/Platypus

[10] C. E. Rasmussen *et al.*, "Gaussian processes for machine learning," in *MIT Press*, 2006.

[11] N. Srinivas *et al.*, "Gaussian process optimization in the bandit setting: No regret and experimental design," in *ICML*, 2009.

[12] E. Zitzler. "Evolutionary algorithms for multiobjective optimization: Methods and applications," in *PhD Thesis, University of Zurich*, 1999.

[13] K. Deb *et al.*, "NSGA-II," in *IEEE TEC,* pp.182–197, 2002.

[14] F. Passos *et al.*, "Accurate Synthesis of Integrated RF Passive Components Using Surrogate Models," in DATE, pp. 397–402, 2016.

[15] Lyu, Wenlong, et al. "Multi-objective bayesian optimization for analog/RF circuit synthesis." in DAC, 2018.

[16] Q. Zhang *et al.*, "MOEA/D: A multiobjective evolutionary algorithm based on decomposition," in *IEEE TEC,* pp.712-731, 2007.