Microelectronics Journal 44 (2013) 631–641

Contents lists available at SciVerse ScienceDirect

Microelectronics Journal

journal homepage: www.elsevier.com/locate/mejo

Fast optimization of nano-CMOS voltage-controlled oscillator using polynomial regression and genetic algorithm

Dhruva Ghai [a,*], Saraju P. Mohanty [b], Garima Thakral [c]

[a] Department of Electronics and Communication Engineering, Oriental University, Indore, India
[b] Department of Computer Science and Engineering, University of North Texas, USA
[c] Department of Computer Science, Oriental University, Indore, India

ARTICLE INFO

Article history:
Received 15 January 2013
Received in revised form
20 April 2013
Accepted 25 April 2013
Available online 17 May 2013

Keywords:
Circuit optimization
Design optimization
Polynomial regression
Genetic algorithm
Nanoscale CMOS (Nano-CMOS)
Voltage-controlled oscillator (VCO)

ABSTRACT

Fast optimization of CMOS circuits is needed to reduce design cycle time and chip cost and to enhance yield. Mature electronic design automation (EDA) tools and well-defined abstraction-levels for digital circuits have largely automated the digital design process. However, analog circuit design and optimization is still not automated. Custom design of analog circuits and slow analog in SPICE has always needed maximum efforts, skills and design cycle time. In this paper, two novel design flows are presented for fast multiobjective optimization of nano-CMOS circuits: actual-value optimization and normalized-value optimization. The design flows consider two characteristics for optimization i.e. power and frequency in a current-starved 50 nm voltage-controlled oscillator (VCO). Accurate polynomial-regression based models have been developed for power (including leakage) and frequency of the VCO to speedup the design optimization. In the actual-value optimization flow, the power model is minimized using genetic algorithm, while treating frequency ≥100 MHz as a constraint. The actual-value optimization flow achieved 21.67% power savings, while maintaining a frequency ≥100 MHz. In the normalized-value optimization flow, the normalized form of these models are subjected to a weighted optimization using genetic algorithm. The normalized-value optimization flow achieved 16.67% power savings, with frequency ≥100 MHz. It is observed that while the actual-value optimization approach provides a better exploration of the design space, the normalized-value optimization approach provides a ≈5× speedup in the computation time.

1. Introduction

Digital design exploration and optimization is highly automated due to availability of large number of electronic design automation (EDA) or computer-aided design (CAD) tools. The digital design automation is aided by the availability of well-defined abstractions for digital circuits (such as system, architecture, and logic levels). However, analog design optimization is still a difficult and time intensive process [1]. For example, the analog simulation time for a nano-CMOS phase-locked loop is a matter of several days. So, debugging such a design is time intensive and costly. This results in high-cost and longer design cycle time. If such analog design are performed at nano-CMOS technology, the issues are further complicated due to leakage and process variation resulting in yield loss.

Most analog integrated circuit (IC) optimization problems involve minimizing a cost function subject to certain constraints. Due to the increasing complexity of modern analog integrated circuits, analog sizing has evolved into a "simulation and optimization" based approach from a "paper and pencil" based approach [2]. Analog sizing problems often require handling multiple conflicting goals, such as power consumption and frequency of a VCO [3]. Novel design/optimization flows are needed, to help the circuit designers [4]. Multi-objective optimization [5] is the process of simultaneously optimizing two or more conflicting design objectives while subjecting the design variables to constraints. During optimization, the baseline design is iteratively tuned by adjusting a large number of design parameters to vast amounts of different design possibilities of the circuit to meet the target design objectives, making it very tedious to do exhaustive design space exploration for complex nano-CMOS circuits to find an optimal solution. Also, the use of compact models [6] with hundreds of parameters in nano-CMOS technology further aggravates the situation.

Polynomial regression model is an abstract model of the netlist which enables a fast design space search. Polynomial regression

* Corresponding author. Tel.: +91 9589563210.
E-mail addresses: dhruvaghai@gmail.com, dhruvaghai@orientaluniversity.in (D. Ghai), saraju.mohanty@unt.edu (S.P. Mohanty), garimathakral@oriental.ac.in (G. Thakral).

0026-2692/$ - see front matter © 2013 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.mejo.2013.04.010

models are useful for relative functions to unknown and very complex non-linear relationship [7,8]. This model is a mathematical predictive equation which may be used as a substitute for the actual circuit, leading to easier and faster simulations with multiple iterations during optimization. For example, as reported in current literature, simulated annealing used on a circuit netlist in a simulator gives convergence in order of minutes as compared to milliseconds, when used over a polynomial regression model [9]. Hence, it can be used as an alternative to the exhaustive search of the design space of the actual circuits. The model can also be used in a variety of tools, such as MATLAB, and is language independent and can be used in a flexible fashion.

To give an overview, the notations and definitions for various terminologies used in this paper are given in Table 1. The paper is organized in the following manner: Contributions of this paper are summarized in Section 2. Section 3 presents the prior related research. Section 4 discusses the proposed novel design flows. The design and analysis of the 50 nm VCO is presented in Section 5. Polynomial models for actual-value and normalized-value optimization are presented in Sections 6 and 7, respectively. Sections 8 and 9 highlight the optimization step of the optimization flows, using genetic algorithm. This is followed by conclusions and future research in Section 10.

2. Contributions of this paper

This paper advances the state-of-the-art in design optimization of analog and mixed-signal circuit where optimization over netlist is time consuming. The *novel contributions* of this paper are as follows:

1. Two novel fast design flows for multiobjective cost function optimization over actual-value and normalized-value in nano-CMOS analog circuits are proposed. The speed up in the design flows is achieved by the use of polynomial regression models and genetic algorithm based algorithms.
2. A method for polynomial regression based modeling has been used for analog circuits. The goodness-of-fit of the polynomial regression models is measured using *SSE*, *RMSE* and R^2.
3. A genetic algorithm based optimization approach is presented that considers power consumption as objective and frequency as constraint for the actual-value optimization approach, and a weighted fitness function for the normalized-value optimization approach.
4. A 50 nm CMOS based current starved VCO is subjected to the proposed design methodology. We report 21.67% power savings and frequency ≥100 MHz using the actual-value optimization flow, and 16.67% power savings and frequency ≥100 MHz in the VCO using the normalized-value optimization flow.

Table 1
Notations and definitions used in this paper.

pwr	Power consumption of VCO
$freq$	Oscillation frequency of VCO
$\widehat{f_{pwr}}$	Normalized polynomial model for power consumption of VCO
$\widehat{f_{freq}}$	Normalized polynomial model for oscillation frequency of VCO
f_{pwr}	Polynomial model for power consumption of VCO
f_{freq}	Polynomial model for oscillation frequency of VCO
RMSE	Root of mean square error
R^2	Coefficient of determination
FF	Fitness function of VCO to be optimized
GA	Genetic algorithm
V_{dd}	Supply voltage of nano-CMOS circuit
V_{in}	Input voltage to nano-CMOS circuit
V_{out}	Output voltage to nano-CMOS circuit
W_p	Width of the PMOS transistor
W_n	Width of the NMOS transistor
$g(x)$	Cost function
$h(x)$	Non-linear inequality constraint

3. Related prior research

Multiobjective optimization problems [10,11] are used wherever optimal decisions need to be taken in the presence of trade-offs between two or more conflicting objectives. For non-trivial multiobjective problems, one cannot identify a single solution that simultaneously optimizes each objective. While searching for solutions, one reaches points such that, when attempting to improve an objective further, other objectives suffer as a result. A comprehensive survey of related works on modeling for analog design has been provided in [12]. The technique for generation of posynomial equation based performance estimation models for analog circuits like multistage amplifiers is described in [13]. An automatic procedure for generation of posynomial models using fitting technique is described in [14]. Other modeling techniques, like Pareto surfaces [15] suffer from the issue of scalability. In order to accommodate a larger design space with a higher number of variables, we may use techniques such as artificial neural networks [16], Takagi–Sugeno neuro-fuzzy logic systems [17], support vector machines based regression [18], Kriging [19] in the proposed design flow instead of polynomial regression. Support vector machine (SVM) has been used for modeling of performance parameters for RF and analog circuits [18,20].

A number of global optimization algorithms are available in current literature for optimization of analog circuits, such as genetic algorithm [10], simulated annealing [19], particle-swarm optimization [21], and artificial bee-colony optimization [22]. These algorithms are particularly effective in finding global optimal or near-optimal solutions, as compared to local optimization techniques like conjugate-gradient. Convex Optimization has been explored in [13] where circuit designs are expressed as posynomial models.

Table 2 shows a comparative perspective among some of the existing optimization flows and proposed optimization flows in the paper. The execution time is highly dependent on the complexity of the circuits involved, and also the algorithm being used. The execution times reported in this paper are comparable with

Table 2
Execution time comparison with existing optimization flows.

Reference	Design	Algorithm	Execution time (s)
Garitselov [23]	LC-VCO	Simulated annealing	1.22
Okobiah [24]	Sense amplifier	Ant colony	1.36
Zheng [25]	Folded cascode Op-amp	Cuckoo search	2.6
This paper—actual-value optimization	21-stage current starved VCO	Genetic algorithm	2.99
This paper—normalized-value optimization	21-stage current starved VCO	Genetic algorithm	0.6

current literature involving fast optimization. It is not possible to have a fair comparison as the circuits, models, and algorithms are different in each case. The actual-value optimization tends to take longer, as it involves constraints.

4. Proposed fast optimization flows for nano-CMOS VCO

In this section, we discuss the two optimization flows for multi-objective optimization in nano-CMOS circuits. A VCO is used as an example circuit for reducing power dissipation, and increasing frequency. The design flows are depicted in Fig. 1.

4.1. Actual-value optimization flow

This section presents the proposed flow for actual-value optimization in nano-CMOS based analog circuits. Although, a VCO is used as a case study for reducing power consumption and increasing frequency, the proposed optimization flow can be generalized for other nano-CMOS analog circuits (like operational amplifiers, filters, and sense amplifiers) as well. The proposed optimization flow over actual-value of characteristics of the VCO is presented in Fig. 1(a). The input to the proposed design flow is a baseline design of circuit. This is one time manual design step. At this stage a netlist is sufficient for the design flow. For a schematic

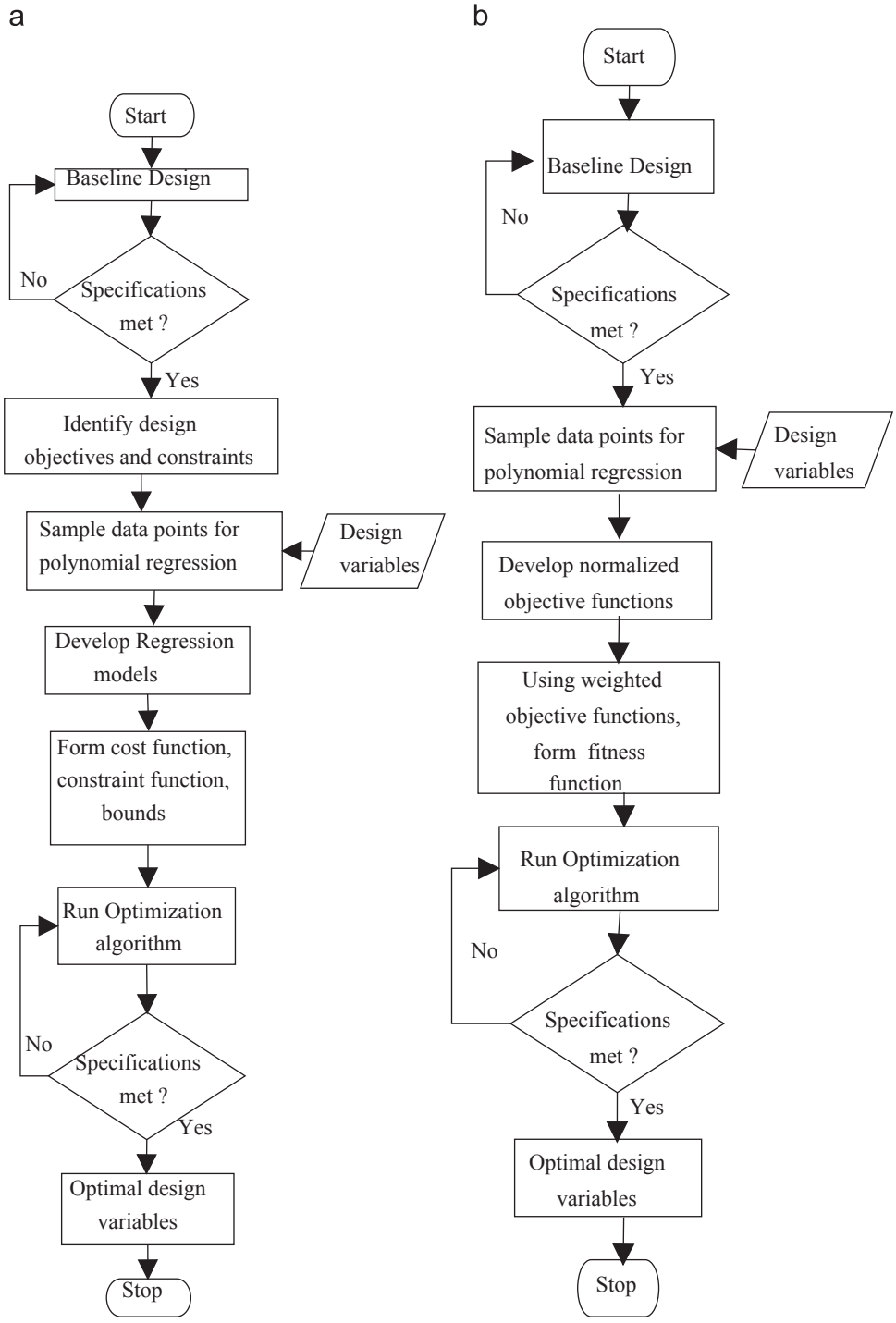

Fig. 1. The proposed design optimization flows: (a) actual-value optimization flow and (b) normalized-value optimization flow.

design this netlist will have only active devices whereas for a layout design full-blown parasitics (RCLK) are included. In this paper, we have used a 50 nm CMOS based circuit of a VCO. A baseline design is carried out as per the specifications. The design objectives and constraints are identified and measured for this baseline VCO (power and frequency). In order to collect data for building the regression-based models, exhaustive simulations are performed to obtain the sample data points in the space defined by the bounded design variables. For this study, we consider two design variables i.e. W_p: width of the PMOS transistors, and W_n: width of the NMOS transistors in the current starved VCO. Thereafter, a polynomial regression based model is developed for each design objective (i.e. power and frequency) using these sample data points. Polynomial regression is efficient, reliable and allows for very fast design exploration [26] in a smaller design space.

At this stage of the design flow, the analog circuit design problem is formulated as an optimization problem. For an example the following problem:

$$\begin{aligned} &\underset{x}{\text{minimize}} \quad g(x) \\ &\text{subject to} \quad h(x) \le 0, \\ &\quad X_L < x < X_H \end{aligned} \tag{1}$$

where $g(x)$ is the cost function to be minimized and the vector $h(x)$ is the non-linear inequality constraint. Vector x corresponds to the design variable set ($x=[W_n, W_p]^T$), and X_L and X_H are their lower and upper bounds, respectively. $g(x)$ and $h(x)$ are formed using the models, which are developed in the regression step of the flow. We have considered power of the VCO as the design objective and frequency of the VCO as the design constraint. However, other VCO specifications like phase noise and tuning linearity may also be considered in the set of design objectives: $g(x)$ or design constraints: $h(x)$. Cost function, constraints and bounds are formed for design variables which are the inputs to the optimization algorithm. This formulation is then fed to the optimization algorithm.

We have used the genetic algorithm (GA) to solve our optimization problem, as it is easily transferred to existing simulations and models. GA is a programming technique that mimics biological evolution as a problem-solving strategy. GA generates solutions to optimization problems using techniques inspired by natural evolution, such as inheritance, mutation, selection, and crossover [5]. GA can handle arbitrary constraints and objectives unlike other commonly used optimization methodologies. The output of the algorithm is the optimal values of the design variables. The circuit is then re-simulated using these design variable values for obtaining the design objectives. For example, power and frequency for the VCO.

4.2. Normalized-value optimization flow

In this section, we present the normalized-value optimization flow for multi-objective optimization in nano-CMOS circuits. The proposed design flow which optimizes over normalized-value of VCO characteristics is shown in Fig. 1(b). The important differences between actual-value optimization (Section 4.1), and the normalized-value optimization flow are as follows:

1. The actual-value optimization flow works with actual values of figures-of-merit (FoMs), while the normalized-value flow works with normalized values. Once the normalized-value flow is complete, the values of the design variables are denormalized and fed to the circuit simulator.
2. In the actual-value optimization flow, the optimization problem involves developing a cost function, a constraint function and the bounds for design variables. For the optimization problem in normalized-value design flow, a cost function over normalized-value (we call it fitness function) and the bounds are developed.

Using the data points obtained in Section 4.1, a normalized polynomial regression [26] based model is developed for each design objective (power and frequency). The models have been normalized by division with the maximum value in its range, so the maximum value that any design objective (or variable) can take is 1. The units of frequency are Hertz (Hz), and the units of power are Watts (W). It is not possible to perform meaningful arithmetic on the two quantities while retaining their units. Normalization renders the models unit-less, so arithmetic operations may be performed, in order to develop a weighted fitness function for multi-objective optimization. Using a weighted version of these normalized objective functions, we develop a fitness function to be subjected to an optimization algorithm. We have applied the genetic algorithm (GA) to our fitness function. The output of the algorithm is the optimal values of the design variables. The circuit is then re-simulated using these design variable values for obtaining the design objectives (power and frequency for the VCO). We report 16.67% power savings, and frequency ≥100 MHz using the design flow.

5. Design and modeling of 50 nm CMOS VCO

The baseline 50 nm CMOS current-starved VCO is shown in Fig. 2. The supply voltage (V_{dd}) is 1 V. The devices P1 and N1 form an inverter, while P2 and N2 operate as current sources. The current sources (P2 and N2) limit the current available to the inverter (P1 and N1), hence starving the inverter for current. The drain currents in the devices P11 and N11 are the same, set by the input voltage. The currents in P11 and N11 are mirrored in each inverter/current source stage.

The oscillation frequency of the current-starved VCO is given by the expression (2) [27]

$$freq_{VCO} = \frac{I_D}{N \times C_{tot} \times V_{dd}} \tag{2}$$

where I_D, drain current; N, number of stages; C_{tot}, total capacitance on the drains of P1 and N1 $= \frac{5}{2} \times C_{ox'} \times (W_p \times L + W_n \times L)$ and V_{dd}, supply voltage. We have chosen $N=21$, $I_D = 10\ \mu A$ and $C_{tot}=4.7fF$ for a target frequency of 100 MHz. Frequency is measured as 111.4 MHz for the baseline design. So, there is a 11.4% error between the prediction from Eq. (2) and simulation. This long-channel equation is an approximation at best and is useful only for baseline design. Hence, Eq. (2) cannot be directly used for

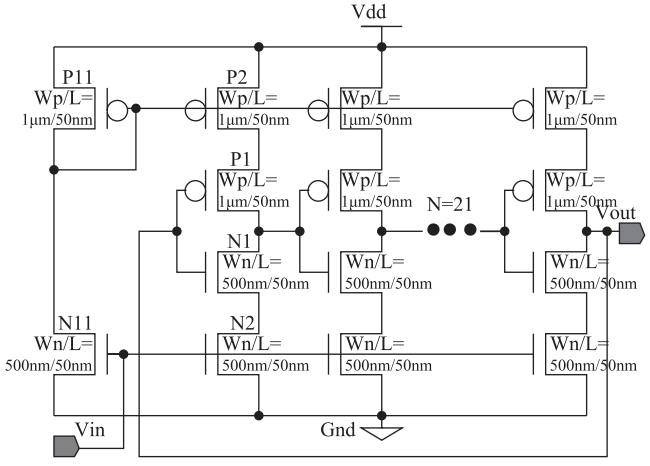

Fig. 2. Logical diagram for VCO with sizes for baseline circuit.

Table 3
Baseline design VCO range characterization.

Parameter	Value
$freq_{VCO}$-min	18.25 MHz
$freq_{VCO}$-max	418.52 MHz
VCO range	400.27 MHz
V_{in}	0–1 V

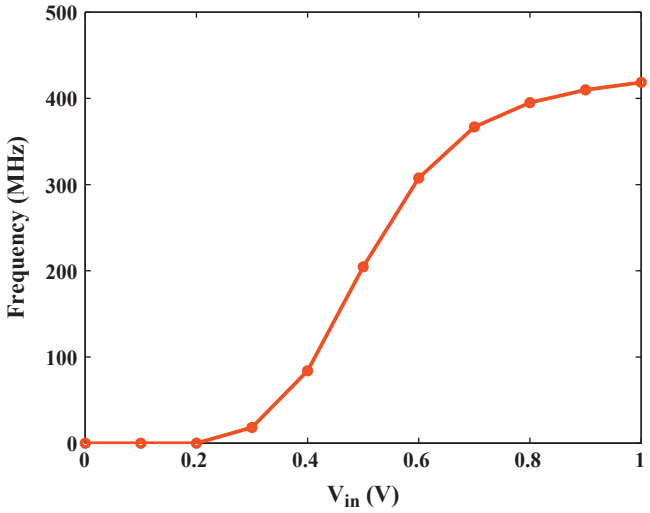

Fig. 3. Frequency–voltage characteristics of 50 nm CMOS based baseline VCO.

optimization in an algorithm. For accurate design optimization, we need a baseline design using which we obtain our polynomial regression model which defines these relationships more accurately. Also, I_D and C_{tot} are both dependent on C_{ox}, (oxide capacitance per unit area). As per Eq. (2) (I_D in the numerator, and C_{tot} in the denominator) their capacitive components get canceled, leaving $freq_{VCO}$ mainly dependent on device geometry and overdrive voltages. This happens, irrespective of whether the device is in saturation or linear region of operation. The VCO range characterization is presented in Table 3, where $freq_{VCO}$-min and $freq_{VCO}$-max are the minimum and maximum frequencies. Fig. 3 shows the frequency–voltage characteristics.

The average power consumption by the nano-CMOS VCO is given by the following expression:

$$pwr_{VCO} = P_{\text{dynamic}} + P_{\text{subthreshold}} + P_{\text{gate-oxide}} \tag{3}$$

$$\begin{aligned} &= NC_{tot}V_{dd}^2 \text{freq}_{VCO} \\ &+ C\exp\left(\frac{V_{gs}-V_{Th}}{Sv_{therm}}\right)\left(1-\exp\left(\frac{-V_{ds}}{v_{therm}}\right)\right) \\ &+ \alpha WL\left(\frac{V_{ox}}{T_{ox}}\right)^2 \exp\left(\frac{-\beta\left(1-\left(1-\frac{V_{ox}}{\phi_{ox}}\right)^{3/2}\right)}{\left(\frac{V_{ox}}{T_{ox}}\right)}\right) \end{aligned} \tag{4}$$

where $C = (\mu_0 \times (\epsilon_{ox}W/T_{ox}L_{eff}) \times v_{therm}^2 \times e^{1.8})$, T_{ox} is the oxide thickness, ϕ_{ox} is the barrier height for the tunneling particle (hole or electron), and α and β are physical parameters used in modeling. The $P_{dynamic}$ component of pwr_{VCO} can be expressed as a function of our design variable set $x = [W_p, W_n]^T$ (using the relationship between C_{tot} and $x = [W_p, W_n]^T$). However, mathematical relationship between $P_{subthreshold}$, P_{gate} and $x = [W_p, W_n]^T$ is not clear, hence not usable. Hence we need a baseline design for obtaining our polynomial regression model which defines these relationships more accurately. We report an average power consumption of 60 μW for the baseline VCO design.

For modeling power dissipation (including leakage) and center frequency as a function of the design variables $[W_p, W_n]^T$, we use polynomial regression. The models are expressed as follows:

$$\widehat{f_X} = \sum_{i,j=0}^{n} p_{ij} \times W_n^i \times W_p^j \tag{5}$$

where X is *freq* or *pwr*, n is the order of regression and p_{ij} is the matrix of coefficients obtained during polynomial regression. Polynomial regression based models have the following advantages:

1. Polynomial models are well known and have a simple form.
2. They are a closed family. Changes of location and scale in the raw data result in a polynomial model being mapped to a polynomial model. That is, polynomial models are not dependent on the underlying metric.
3. Polynomial models are computationally easy to use.
4. Lower degree polynomials have good interpolatory properties. Provide good fits within the range of data.

Polynomial models may have scalability issue. If one wants to maintain the same density (accuracy) of the model when more dimensions are added, then the sample size increases enormously (usually exponentially with the number of dimensions). The model becomes difficult to interpret as more explanatory variables are included. Graphing the fitted values helps, but with more than two predictors we cannot see the complete regression surface. Despite these limitations, multiple regression can be useful as long as a regression surface can be built.

6. Polynomial models for actual-value optimization

This section presents the polynomial models used for actual-value optimization. The design space is explored through parametric simulations for the values of W_n and W_p ranging from the lower bound (100 nm) to upper bound (1 μm) and the data points are used to obtain least squares fit polynomial. The regression surfaces for power and frequency are shown in Fig. 4(a) and (b), respectively. The polynomial coefficients in a matrix format are provided in Eqs. (6) and (7). A polynomial of the order 2 has been used, hence we obtain a 3×3 matrix:

$$p_{ij}(f_{pwr}) = \begin{Bmatrix} 5.409\times10^{-5} & 6.809\times10^{-6} & -1.348\times10^{-6} \\ 6.569\times10^{-6} & 2.718\times10^{-6} & 0 \\ -1.191\times10^{-6} & 0 & 0 \end{Bmatrix} \tag{6}$$

$$p_{ij}(f_{freq}) = \begin{Bmatrix} 1.095\times10^{8} & 1.613\times10^{7} & -1.022\times10^{7} \\ 1.17\times10^{7} & 8.572\times10^{6} & 0 \\ -8.409\times10^{6} & 0 & 0 \end{Bmatrix} \tag{7}$$

For goodness-of-fit of the polynomial model, we analyze the sum of squared error (SSE), root of Mean Square Error (RMSE), and coefficient of determination (R^2) [8]. SSE is a measure of the discrepancy between the data and an estimation model. If the estimation model is well fitted, it results in predictive data values close to observed values. A small SSE indicates a tight fit of the model to the data. The formula used for calculating SSE is the following:

$$SSE = \sum_{i=0}^{N_{samp}} (f(x_i) - \widehat{f(x_i)})^2 \tag{8}$$

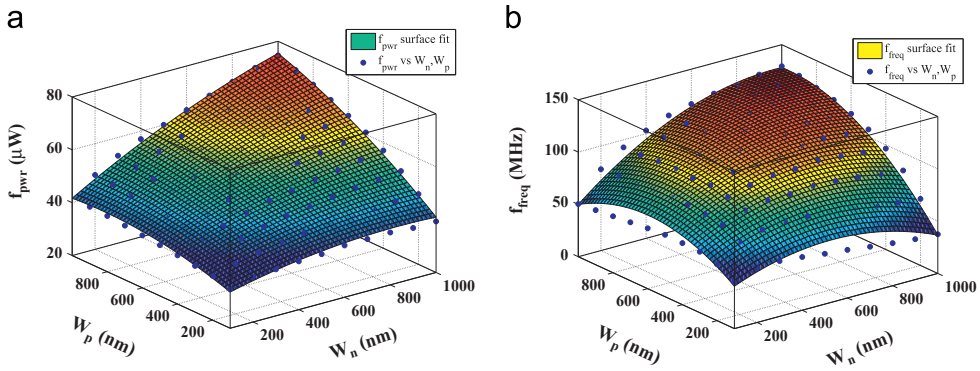

Fig. 4. Surface plots for power and frequency during actual-value optimization: (a) average power (leakage) and (b) center frequency.

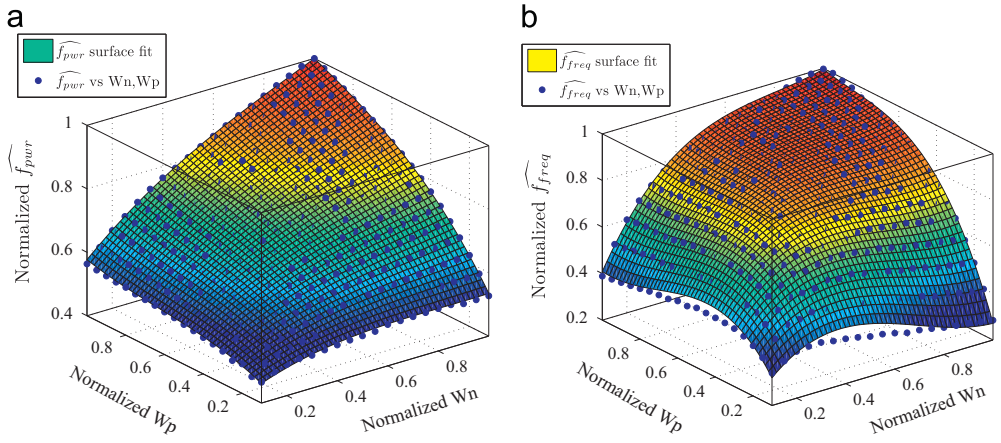

Fig. 5. Surface plots for normalized power and frequency during normalized-value optimization: (a) normalized power dissipation (including leakage) and (b) normalized center frequency.

where $N_{samp} = 10 \times 10 = 100$ are uniformly distributed points of the parameters selected in the design domain (x). Current literature [28,9] shows that uniform sampling results in an even distribution, produces more effective coverage than random sampling, provides superior accuracy to random sampling, can deal with a large number of runs and input variables and are computationally cheap to generate. Hence, we have decided to go for uniform sampling. We report an SSE of 56.55 pW for the power model and 4.76×10^{15} Hz for the frequency model. $f(x_i)$ and $\widehat{f(x_i)}$ are the responses at point (x_i) of the data point observations and the regression based model, respectively. One way to calculate RMSE is obtained by substituting Eq. (8) in Eq. (9). Alternatively, RMSE is directly computed as shown in Eq. (10):

$$RMSE = \sqrt{\frac{SSE}{N_{samp}}} \tag{9}$$

$$RMSE = \sqrt{\frac{1}{N_{samp}} \sum_{i=0}^{N} (f(x_i) - \widehat{f(x_i)})^2}. \tag{10}$$

The RMSE estimates the difference between the observed data points from simulations and the polynomial regression model. A smaller RMSE value indicates an accurate polynomial regression model [8]. We report an RMSE of 0.2378 μW for the power model and 2.184 MHz for the frequency model. Since we have used a smaller number of samples (100) and lower order polynomial regression (order=2) for the actual-value optimization, a larger RMSE error (2.18% for frequency model) is obtained [28]. The proposed flow can produce results with further low RMSE if number of sample points and order of polynomial regressions increased.

The coefficient of determination (R^2) measures the proportion of the variation of the data point observations around the mean that is explained by the fitted regression model. Advantage of using R^2 is that its scale is intuitive, and an improvement in the regression model results in proportional increase in R^2. The closer R^2 is to 1, the greater the degree of association between variables x and the response. The expression used for calculating R^2 is the following:

$$R^2 = 1 - \left(\sum_{i=0}^{N_{samp}} \frac{(f(x_i) - \widehat{f(x_i)})^2}{\sum_{i=0}^{N_{samp}} (f(x_i) - \overline{f(x_i)})^2} \right) \tag{11}$$

where $\overline{f(x_i)}$ is the mean of the response at point (x_i) of the data point observations. We report an R^2 value of 0.9943 for the power model and 0.9953 for the frequency model.

7. Polynomial models for normalized-value optimization

For normalized-value optimization, we work with normalized data. The data obtained in Section 6 is normalized (both the design variables and FoMs). The normalization is carried out by dividing the entire data range by the maximum value within that range. This is followed by developing the normalized polynomial model for power and frequency. We use $\widehat{f_{pwr}}$ and $\widehat{f_{freq}}$ to denote normalized vales instead of actual values. The regression plots for normalized power and frequency are shown in Fig. 5(a) and (b), respectively. The coefficient matrix is presented in Eqs. (12) and (13). A polynomial of the order 3 has been used, hence we obtain a 4×4 matrix:

$p_{ij}(\widehat{f_{pwr}})$

$$= \begin{Bmatrix} 0.4028 & 0.2884 & -0.4526 & 0.2592 \\ 0.3025 & 0.8811 & -0.3421 & 0 \\ -0.4608 & -0.06591 & 0 & 0 \\ 0.187 & 0 & 0 & 0 \end{Bmatrix} \quad (12)$$

$$p_{ij}(\widehat{f_{freq}}) = \begin{Bmatrix} 0.06758 & 1.538 & -2.808 & 1.376 \\ 1.294 & 2.08 & -0.7208 & 0 \\ -2.666 & -0.4581 & 0 & 0 \\ 1.294 & 0 & 0 & 0 \end{Bmatrix} \quad (13)$$

In the above specific case, $N_{samp} = 19 \times 19 = 361$ and order=3 for polynomial regression. The goodness-of-fit is measured using RMSE and R^2 described in Eqs. (10) and (11), respectively. We report an RMSE of 0.003559 and R^2 value of 0.9992 for the power model. RMSE of 0.0151, and R^2 of 0.9929 is reported for the center-frequency model. Here, as the number of samples and the order of polynomial regression increases as compared to the actual-value optimization, the RMSE error is lesser (1.51%) [28].

8. Actual-value optimization using genetic algorithm (GA)

This section discusses the development of the cost function, constraint function and genetic algorithm (GA) used for optimization. The Pareto-front is presented in Fig. 6 [29]. The feasible region where Power_{VCO} will be minimized while keeping $\text{Frequency}_{VCO} \geq 100$ MHz is encircled. From the feasible region we can see that the minimum power consumption would be in the neighborhood of 50 μW.

We now formulate the optimization problem for power minimization as follows:

$$\begin{aligned} &\text{minimize} \quad \text{Power}_{VCO} \\ &\text{such that} \quad \text{Frequency}_{VCO} \geq 100 \text{ MHz}, \\ &100 \text{ nm} \leq [W_p, W_n]^T \leq 1 \text{ μm}. \end{aligned} \quad (14)$$

In order to make Eq. (14) the same format as Eq. (1), we formulate the optimization problem as follows:

$$\begin{aligned} &\text{minimize} \quad g(x) = \text{Power}_{VCO} \\ &\text{such that} \quad 100 \times 10^6 - \text{Frequency}_{VCO} \leq 0, \\ &100 \text{ nm} \leq x \leq 1 \text{ μm} \end{aligned} \quad (15)$$

where cost function is $g(x) = \text{Power}_{VCO}$ and constraint function is $h(x) = 100 \times 10^6 - \text{Frequency}_{VCO}$. The lower and upper bounds for the design variable set $x = [W_p, W_n]^T$ are 100 nm and 1 μm, respectively. The cost function of a target VCO is minimized through a genetic algorithm which is shown in Algorithm 1.

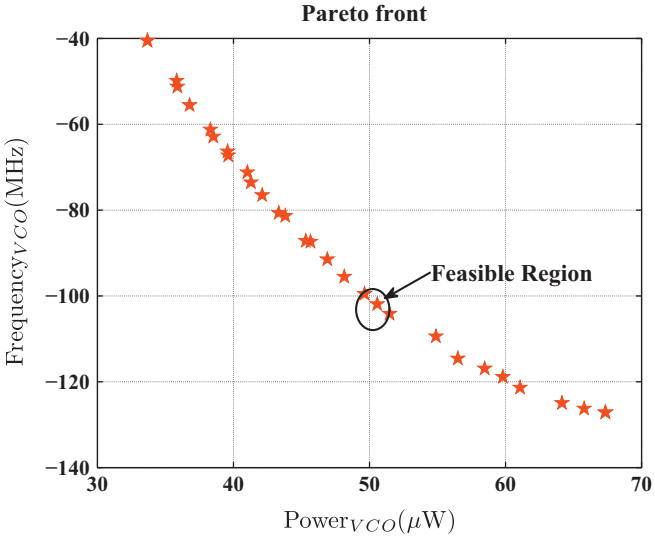

Fig. 6. Pareto-front obtained using Power_{VCO} and Frequency_{VCO}.

Algorithm 1. Proposed genetic algorithm (GA) for actual-value optimization.

1: **Input:** Cost function $g(x)$, constraint function $h(x)$, 100 nm$\leq x \leq$1 μm and design solution set x.
2: **Output:** Optimal design solution x_{opt}.
3: Generate initial population design variable x.
4: Initialize the number of iterations, gen=0.
5: **while** gen < maxgen−1 **do**
6: Select mating pool from the initial population as $x' \subset x$.
7: Initialize set of children $x'' = \varnothing$.
8: **for** i=0 to population size-1 **do**
9: Select individuals x'_a at random from x'.
10: Apply crossover to x'_a to produce child x'_{child}.
11: Randomly mutate produced child x'_{child}.
12: $x'' = x'' \cup x'_{child}$.
13: **end for**
14: $x''' = x'' \cup x'$.
15: Evaluate fitness using $g(x''')$, $h(x''')$.
16: Increment the counter as gen=gen+1.
17: **end while**
18: The optimal solution is obtained: $x_{opt} = x'''$.
19: Assign x_{opt} to transistors in VCO and recreate the design using the new parameters.
20: Re-simulate VCO to characterize for *freq* and *pwr*.

The genetic algorithm has an advantage over most of other techniques presented in current literature as it helps in formulating the problem as a non-linear optimization with equality and inequality constraints [30]. The inputs to the Algorithm 1 are the cost function $g(x)$, the non-linear inequality constraint function $h(x)$ and the lower (X_L) and upper (X_H) bounds to the design solution set x. New candidates (children) for the design solution set are generated with a mechanism called crossover (rate=0.8) which combines part of the genetic material of each parent x' and then applies a random mutation. If the child x'_{child} inherits good characteristics from its parents x', it will have a higher probability to survive. The values of child x'_{child} are stored in the set of children x''. The fitness of the child x'' and parent x' population is then evaluated using $g(x)$, $h(x)$ and the survivors can be formed either by the fittest from x''∪ the fittest from x'.

Genetic algorithm first accepts a set of design solution set (statements 6 and 7 in Algorithm 1), and then constructs a set of child design solution set (statements 9–14 in Algorithm 1). The stopping criterion is provided by the number of generations (maxgen) which as fixed at maxgen=100. The algorithm converges before the maxgen is reached (generations (iterations)=20) as shown in Fig. 7(a). The final values of design variables (best individuals) as shown in Fig. 7(b). 30 independent trials were run for the actual-value optimization, and the best solution from the pool of feasible candidates is reported. The execution time for the best solution is also reported as ≈2.99 s. This is higher compared to current literature of VCOs because this is optimization with constraints and a more complex circuit (VCO having N=21 stages) is involved. Table 5 shows the final design solution set obtained from proposed genetic algorithm based optimization. The VCO circuit with optimized sizes for actual-value optimization is presented in Fig. 8(a). The corresponding frequency–voltage characteristic is shown in Fig. 8(b). The VCO range characterization is presented in Table 4.

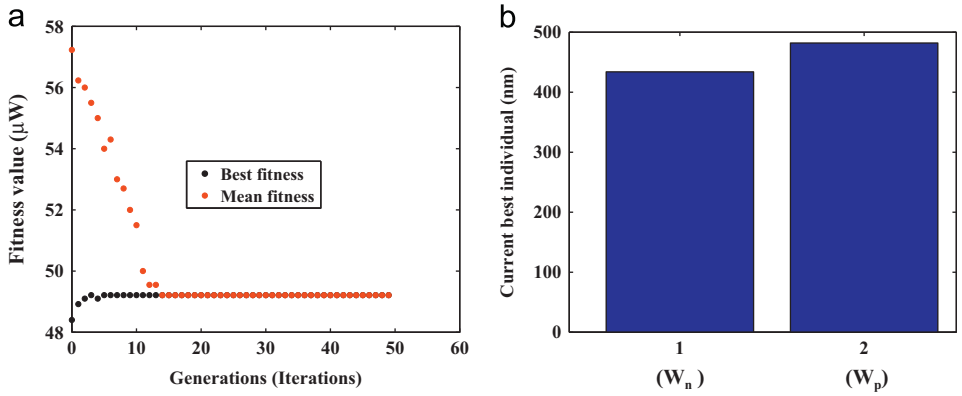

Fig. 7. GA algorithm converging for actual-value optimization and the optimal design variables (best individual): (a) GA convergence and (b) best individual.

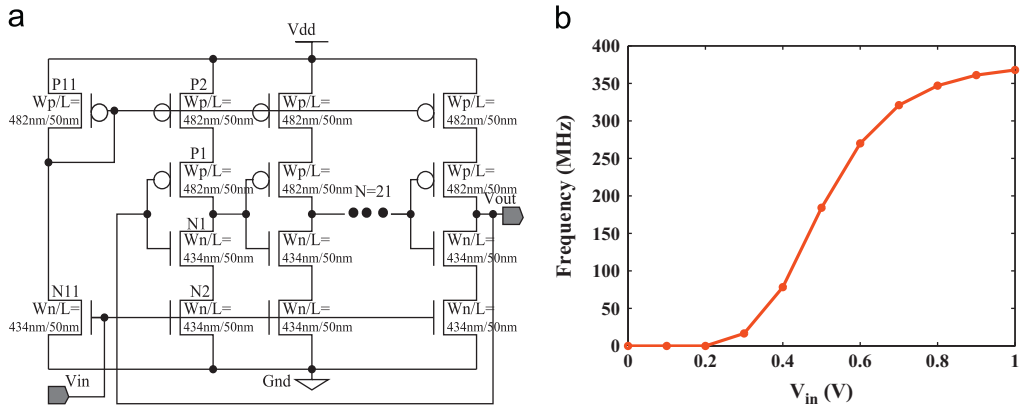

Fig. 8. Optimized VCO using actual-value optimization algorithm: (a) optimized VCO circuit with sizes and (b) optimal VCO characteristic.

Table 4
VCO range characterization of the actual-value optimized design.

Parameter	Value
$freq_{VCO}$-min	16.63 MHz
$freq_{VCO}$-max	367.84 MHz
VCO range	351.21 MHz
V_{in}	0–1 V

9. Normalized-value optimization using weighted genetic algorithm (GA)

This section discusses the development of the fitness function, and GA used for optimization. We formulate the optimization problem as

$$\begin{aligned} &\text{minimize} \quad FF = \sum_{i=0}^{1} w_i \times F(i) \\ &\text{where} \quad F = (\widehat{f_{pwr}}, \widehat{f_{freq}}) \\ &\text{such that} \quad 0.1 \le [\widehat{W_p}, \widehat{W_n}] \le 1, \ \sum_{i=0}^{1} w_i = 1 \end{aligned} \tag{16}$$

where FF is the fitness function (also called objective set), $F(i)$ is the ith objective function in the objective set FF and w_i is the weight assigned to the ith objective function. Since we are handling two objective functions, $F(0) = \widehat{f_{pwr}}$, $F(1) = \widehat{f_{freq}}$. We assign a weight of 0.5 to each objective function, in order to maintain equal priority. The lower and upper bounded constraints for the design variables $[W_p, W_n]$ are 100 nm and 1 μm, respectively, which translate to 0.1 and 1 by normalization. The fitness function is minimized through GA. Since we wish to minimize power and maximize frequency, we assign a negative sign to $\widehat{f_{freq}}$. Optimization problem of Eq. (16) can be re-formulated as a new optimization problem shown in the following:

$$\begin{aligned} &\text{minimize} \quad FF = 0.5 \times \widehat{f_{pwr}} - 0.5 \times \widehat{f_{freq}} \\ &\text{such that} \quad 0.1 \le [\widehat{W_p}, \widehat{W_n}] \le 1. \end{aligned} \tag{17}$$

The coefficient matrix for FF as presented in the following:

$$p_{ij}(FF) = \begin{Bmatrix} 0.1676 & -0.6248 & 1.1777 & -0.5584 \\ -0.4958 & -0.5995 & 0.1893 & 0 \\ 1.1026 & 0.1961 & 0 & 0 \\ -0.5535 & 0 & 0 & 0 \end{Bmatrix} \tag{18}$$

The normalized-value optimization basically formulates the multi-objective optimization problem as a single-objective optimization problem, where the different performance objectives are combined to form a single scalar objective (called fitness function), which produces only one (and unique) solution for multiple independent trials [17,31].

The pseudocode for the GA applied to VCO is shown in Algorithm 2. The inputs to the algorithm are the fitness function FF and the lower and upper bounds to the design solution set $[W_p, W_n]$. Children for the design solution set are generated with crossover (rate = 0.8) which combines part of the genetic material of each parent $[W_{pa}, W_{nb}]'$ and then applies a random mutation. The child $[W_{pchild}, W_{nchild}]'$ which inherits good characteristics from its parents $[W_{pa}, W_{nb}]'$ has a higher probability to survive. The values of child $[W_{pchild}, W_{nchild}]'$ are stored in the set of children $[W_p, W_n]''$. The fitness of the child $[W_p, W_n]''$ and parent $[W_p, W_n]'$ population is evaluated using FF and the survivors can be formed

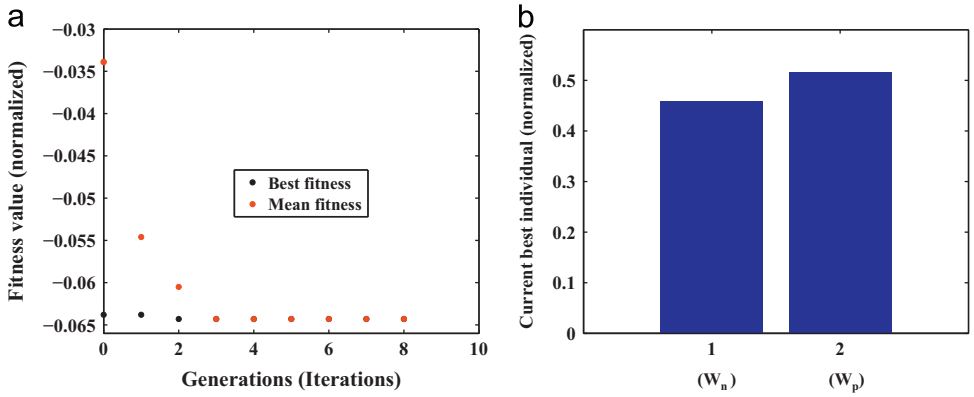

Fig. 9. GA algorithm converging for normalized-value optimization and the optimal design variables (best individual): (a) GA convergence and (b) best individual.

Table 5
Comparison of design objectives in baseline and optimized VCO for actual-value optimization.

Design	W_p	W_n	*pwr*	*freq*	Convergence	Execution time
Baseline	1 μm	500 nm	60 μW	111.4 MHz	–	–
Optimized	482 nm	434 nm	47 μW	105.4 MHz		
Prediction from algorithm	482 nm	434 nm	48.89 μW Error = 4.02%	103.3 MHz Error = 1.99%	20 iterations	2.99 s

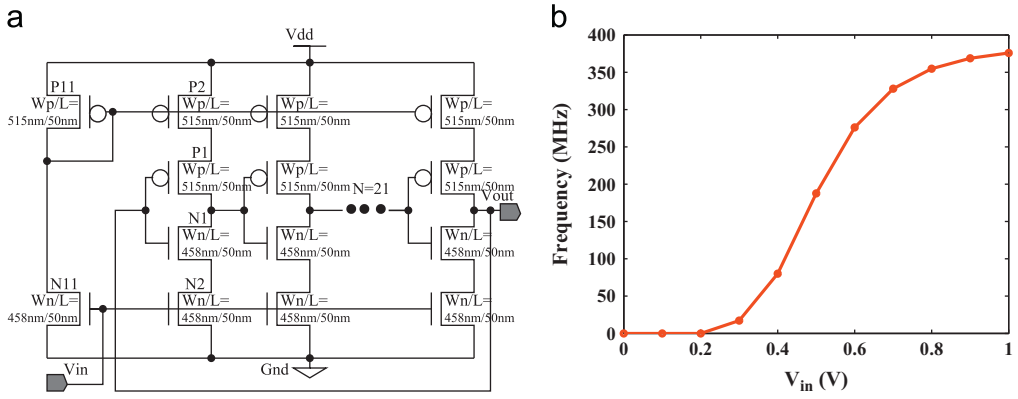

Fig. 10. Optimized VCO using normalized-value optimization algorithm: (a) optimized VCO circuit with sizes and (b) optimal VCO characteristic.

by the fittest from $[W_p, W_n]''$∪ the fittest from $[W_p, W_n]'$. The *execution time* for the normalized-value optimization is ≈0.6 s.

Algorithm 2. Normalized-value optimization using weighted GA.

1: **Input:** Fitness function *FF*, $0.1 \le [W_p, W_n] \le 1$.
2: **Output:** Optimal design solution $[W_p, W_n]$.
3: Generate initial population $[W_p, W_n]$.
4: gen = 0.
5: **while** gen < maxgen − 1 **do**
6: Select mating pool $[W_p, W_n]' \subset [W_p, W_n]$.
7: Initialize set of children $[W_p, W_n]'' = \varnothing$.
8: **for** i = 0 to population size-1 **do**
9: Select individuals $[W_{pa}, W_{nb}]'$ at random from $[W_p, W_n]'$.
10: Apply crossover to $[W_{pa}, W_{nb}]'$ to produce child $[W_{pchild}, W_{nchild}]'$.
11: Randomly mutate produced child $[W_{pchild}, W_{nchild}]'$.
12: $[W_p, W_n]'' = [W_p, W_n]'' \cup [W_{pchild}, W_{nchild}]'$.
13: **end for**
14: $[W_p, W_n] = \mathrm{FF}([W_p, W_n]', [W_p, W_n]'')$.
15: gen = gen + 1.
16: **end while**

Table 6
VCO range characterization of the normalized-value optimized design.

Parameter	Value
$freq_{VCO}$-min	17.21 MHz
$freq_{VCO}$-max	375.86 MHz
VCO range	358.65 MHz
V_{in}	0–1 V

17: Assign $[W_n, W_p]$ to transistors in VCO.
18: Re-simulate VCO to report *freq* and *pwr*.

The number of iterations (generations = 3) the algorithm goes through before converging is shown in Fig. 9(a). The final values of design variables (best individuals) are shown in Fig. 9(b). Table 5 shows the final design solution set obtained from GA. The VCO circuit with optimized sizes for normalized-value optimization is presented in Fig. 10(a). The corresponding frequency–voltage characteristic is provided in Fig. 10(b). The VCO range characterization is presented in Table 6.

Table 7
Comparison of design objectives in baseline and optimized VCO for normalized-value optimization.

Design	W_p	W_n	*pwr*	*freq*	Convergence	Execution time
Baseline	1 μm	500 nm	60 μW	111.4 MHz	–	–
Optimized	515 nm	458 nm	50 μW	105.4 MHz		
Prediction from algorithm	515 nm	458 nm	51.19 μW Error=2.38%	105.3 MHz Error=0.1%	3 iterations	0.6 s

The *maximum* error between the predicted results from algorithm and simulation results in both the design flows is ≈4.02%. Normalized-value optimization has higher accuracy than actual-value optimization because of more accurate models being used (more number of samples and higher order polynomial). It is observed that while actual-value optimization (results shown in Table 5) achieves higher power minimization than normalized-value optimization (results shown in Table 7), it takes longer than the normalized-value optimization to converge. The normalized-value optimization offers ≈5*x* speedup over the actual-value optimization. The reason is that in optimization theory, the optimality conditions for interior points are usually much simpler than the optimality conditions for boundary points [32]. Boundary points appear more prominently in constrained optimization, when one tries to optimize a function, subject to several functional constraints. For this reason, the optimality conditions for boundary points are generally discussed in actual-value optimization, whereas the optimality conditions for interior points are discussed in normalized-value optimization. Also, unlike the normalized-value optimization, the actual-value optimization's search space comprises the feasible and infeasible parts. Thereby, the proportion of these two parts, the types of constraints (inequality in our case), and the location of the optimum (on the boundaries of the feasible region or within the feasible region) concurrently add difficulties to the search of the global optimum [33]. We get a number of feasible design solutions to choose from in the case of actual-value optimization, whereas normalized-value optimization converges to one unique solution in every independent trial. Hence, for an analog circuit like current-starved VCO, the actual-value optimization would be preferred for a better exploration of the design space, whereas the normalized-value optimization would be preferred for a faster solution.

10. Conclusions and future research

We have presented two design flows for polynomial regression model assisted multiobjective optimization on a 50 nm CMOS based VCO. The center frequency and power dissipation have been considered for optimization. A model-based approach is beneficial as it is faster than optimizing the actual circuit. The proposed approach leads to 21.67% power reduction for actual-value optimization (which converges in 2.99 s) and 16.67% power savings for normalized-value optimization, which converges in 0.6 s. The error introduced by the model based approach is very minimal which is in the range of 0.1–4.0%. Thus, the proposed research is a significant advancement of the state-of-art in the analog design optimization. The proposed research will help to reduce the design cycle time and reduce the chip cost.

As part of future research, regression based models will be developed, taking into account supply sensitivity, temperature sensitivity and parasitics. One possible way to study PVT variations effect on the VCO could be to develop polynomial regression models for the VCO while subjecting it to worst case process and temperature corner. These regression models may then be optimized for the worst case process, which will work well for the nominal corner. Takagi–Sugeno neuro-fuzzy logic systems will also be explored for modeling. VCO performance parameters other than power and frequency, such as phase noise, tuning linearity will also be considered. Also, the actual-value optimization problem presented in this paper will be solved using other algorithms such as the Lagrange multiplier method and artificial bee colony. The effects of process variation will be incorporated in future statistical design flows.

Acknowledgments

A preliminary version of this research was presented in the following double-blind reviewed conference [34].

References

[1] D. Ghai, S.P. Mohanty, E. Kougianos, Design of parasitic and process-variation aware nano-CMOS RF circuits: a VCO case study, IEEE Trans. Very Large Scale Integration (VLSI) Syst. 17 (9) (2009) 1339–1342.
[2] B.D. Smedt, G. Gielen, WATSON: design space boundary exploration and model generation for analog and RF IC design, IEEE Trans. Computer-Aided Design Integrated Circuits Syst. 22 (2) (2003) 213–224.
[3] Y.-S. Park, W.-Y. Choi, On-chip compensation of ring VCO oscillation frequency changes due to supply noise and process variation, IEEE Trans. Circuits Syst. II 59 (2) (2012) 73–77.
[4] T. Soorapanth, On weight optimization in multi-objective circuit design problem via geometric programming, in: Proceedings of the International Symposium on Intelligent Signal Processing and Communication Systems, 2009, pp. 509–512.
[5] G. Oltean, S. Hintea, E. Sipos, A genetic algorithm-based multiobjective optimization for analog circuit design, in: Proceedings of the 13th International Conference on Knowledge-Based and Intelligent Information and Engineering Systems: Part II, 2009, pp. 506–514.
[6] K.M. Cao, W.C. Lee, W. Liu, X. Jin, P. Su, S.K.H. Fung, J.X. An, B. Yu, C. Hu, BSIM4 gate leakage model including source–drain partition, in: Electron Devices Meeting, 2000, IEDM Technical Digest, International, 2000, pp. 815–818.
[7] T.J. Santner, B. Williams, W. Notz, The Design and Analysis of Computer Experiments, Springer-Verlag, 2003.
[8] K. Fang, R. Li, A. Sudjianto, Design and Modeling for Computer Experiments (Computer Science & Data Analysis), Chapman & Hall, CRC, 2005.
[9] O. Garitselov, S.P. Mohanty, E. Kougianos, Fast optimization of nano-CMOS mixed-signal circuits through accurate metamodeling, in: ISQED, 2011, pp. 405–410.
[10] A. Das, R. Vemuri, A graph grammar based approach to automated multi-objective analog circuit design, in: Proceedings of the Design Automation and Test in Europe, 2009, pp. 700–705.
[11] B. Liu, F.V. Fernandez, P. Gao, G. Gielen, A fuzzy selection based constraint handling method for multi-objective optimization of analog cells, in: Proceedings of the European Conference on Circuit Theory and Design, 2009, pp. 611–614.
[12] R.A. Rutenbar, G. Gielen, J. Roychowdhury, Hierarchical modeling, optimization, and synthesis for system-level analog and RF designs, Proc. IEEE 95 (3) (2007) 640–669.
[13] V. Aggarwal, Analog Circuit Optimization using Evolutionary Algorithms and Convex Optimization, Master' Thesis, Massachusetts Institute of Technology, May 2007.
[14] W. Daems, G. Gielen, W. Sansen, Simulation-based automatic generation of signomial and posynomial performance models for analog integrated circuit sizing, in: Proceedings of the 2001 IEEE/ACM International Conference on Computer-aided Design, 2001, pp. 70–74.
[15] S.K. Tiwary, P.K. Tiwary, R.A. Rutenbar, Generation of yield-aware Pareto surfaces for hierarchical circuit design space exploration, in: Proceedings of the 43rd Annual Design Automation Conference, 2006, pp. 31–36.
[16] O. Garitselov, S.P. Mohanty, E. Kougianos, Fast-accurate non-polynomial metamodeling for nano-CMOS PLL design optimization, in: 25th International Conference on VLSI Design, 2012, pp. 316–321.
[17] G. Oltean, S. Hintea, E. Sipos, Analog circuit design based on computational intelligence techniques, J. Autom. Mobile Robot. Intell. Syst. 3 (2) (2009) 63–69.

[18] T. Kiely, G. Gielen, Performance modeling of analog integrated circuits using least-squares support vector machines, in: Proceedings of the Design Automation and Test in Europe, 2004, pp. 448–453.
[19] O. Okobiah, S.P. Mohanty, E. Kougianos, O. Garitselov, Kriging-assisted ultra-fast simulated-annealing optimization of a clamped bitline sense amplifier, in: International Conference on VLSI Design, 2012, pp. 310–315.
[20] X. Ren, T. Kazmierski, Performance modelling and optimisation of RF circuits using support vector machines, in: Proceedings of the 14th International Conference on Mixed Design of Integrated Circuits and Systems, 2007, pp. 317–321.
[21] R.A. Thakker, M.S. Baghini, M.B. Patil, Low-power low-voltage analog circuit design using Hierarchical particle swarm optimization, in: Proceedings of the 2009 22nd International Conference on VLSI Design, 2009, pp. 427–432.
[22] O. Garitselov, S.P. Mohanty, E. Kougianos, Accurate polynomial metamodeling-based ultra-fast bee colony optimization of a nano-CMOS phase-locked loop, J. Low Power Electron. 8 (3) (2012) 317–328.
[23] S.P. Mohanty, E. Kougianos, O. Garitselov, J.M. Molina, Polynomial-metamodel assisted fast power optimization of Nano-CMOS PLL components, in: Proceedings of the Forum on Specification and Design Languages, 2012, pp. 233–238.
[24] O. Okobiah, S.P. Mohanty, E. Kougianos, Ordinary kriging metamodel-assisted ant colony algorithm for fast analog design optimization, in: ISQED, 2012, pp. 458–463.
[25] G. Zheng, S.P. Mohanty, E. Kougianos, Metamodel-assisted fast and accurate optimization of an OP-AMP for biomedical applications, in: ISVLSI, 2012, pp. 273–278.
[26] S.P. Mohanty, E. Kougianos, PVT-tolerant 7-transistor SRAM optimization via polynomial regression, in: Proceedings of the 2011 International Symposium on Electronic System Design, 2011, pp. 39–44.
[27] R.J. Baker, CMOS Circuit Design, Layout, and Simulation, Wiley-IEEE Press, 2010.
[28] O. Garitselov, S.P. Mohanty, E. Kougianos, A comparative study of metamodels for fast and accurate simulation of nano-CMOS circuits, IEEE Trans. Semiconductor Manuf. 25 (1) (2012) 26–36.
[29] L. Ivanciu, G. Oltean, S. Hintea, Design illustration of a symmetric OTA using multiobjective genetic algorithms, in: Proceedings of the 15th International Conference on Knowledge-based and Intelligent Information and Engineering Systems, vol. Part III, Springer-Verlag, 2011, pp. 443–452.
[30] M.T. Al-Hajri, Assessment of genetic algorithm selection, crossover and mutation techniques in reactive power optimization, in: Proceedings of the Evolutionary Computation, 2009, IEEE Congress on CEC '09, 2009, pp. 1005–1011.
[31] A. Somani, P.P. Chakrabarti, A. Patra, An evolutionary algorithm-based approach to automated design of analog and RF circuits using adaptive normalized cost functions, IEEE Trans. Evol. Comput. 11 (3) (2007) 336–353.
[32] O. Güler, Foundations of Optimization, vol. 258, Springer, 2010.
[33] Z. Cai, Y. Wang, A multiobjective optimization-based evolutionary algorithm for constrained optimization, IEEE Trans. Evol. Comput. 10 (6) (2006) 658–675.
[34] D. Ghai, S.P. Mohanty, G. Thakral, Fast analog design optimization using regression based modeling and genetic algorithm: a nano-CMOS VCO case study, in: Proceedings of the 14th IEEE International Symposium on Quality Electronic Design (ISQED), 2013, pp. 422–427.