DIPARTIMENTO DI INGEGNERIA INFORMATICA
AUTOMATICA E GESTIONALE ANTONIO RUBERTI

SAPIENZA
UNIVERSITÀ DI ROMA

A SVM Surrogate Model Based Method for Yield Optimization in Electronic Circuit Design

Angelo Ciccazzo
Gianni Di Pillo
Vittorio Latorre

A SVM Surrogate Model Based Method for Yield Optimization in Electronic Circuit Design

A. Ciccazzo,[*] G. Di Pillo,[†] V. Latorre[†]

Abstract

Yield optimization is a challenging topic in electronic circuit design. Methods for Yield optimization based on Montecarlo analysis of a circuit whose behavior is reproduced by simulations usually require too many time expensive simulations to be effective for iterative optimization. In this work we take inspiration from both the Montecarlo analysis based methods and machine learning methods in order to realize a methodology able to perform the Yield optimization in a more efficient way. The method we propose tackles the Yield optimization problem by embedding the training of a support vector machine surrogate model and the generation of a Montecarlo analysis into the optimization procedure. We report the numerical results obtained by using the proposed method for the design of two real consumer circuits provided by ST Microelectronics, and we compare these results with the ones obtained using the industrial benchmark currently adopted at ST Microelectronics for Yield optimization. These preliminary results show that the method is promising to be very efficient and capable of reaching design solutions with high values of the Yield.

Keywords. Electronic Circuit Design, Yield Optimization, SVM Surrogate Model, Derivative-free Optimization Algorithm.

---

[*]ST Microelectronics, Stradale Primosole 50, 95121 Catania, Italy E-mail: angelo.ciccazzo@st.com
[†]Dipartimento di Ingegneria Informatica Automatica e Gestionale, Università di Roma "La Sapienza", via Ariosto 25 - 00185 Roma, Italy. E-mails: dipillo@dis.uniroma1.it, latorre@dis.uniroma1.it

1   Introduction

We propose a methodology to maximize the Yield in the electronic circuit design process. The behavior of a circuit is generally described by its *performances* $f_i, i = 1, \ldots, m$ such as the Gain, the Delay between two waveforms, the Phase Margin, the Dissipated Power, and so on. For a circuit to be in full working order, all the m performances f_i must satisfy certain *specifications* that are generally given in terms of lower and upper bounds:

$$l_i \leq f_i(x_d, x_o, x_p) \leq u_i \quad i = 1, \ldots, m, \tag{1}$$

where:

- x_d, Design Variables: these variables represent the geometrical dimensions of the components in the circuits (e.g. channel widths and lengths);

- x_o, Operating Variables: these variables model operating and environmental conditions (e.g. supply voltage and temperature);

- x_p, Process Variables: these variables are usually subject to uncertainty due to fluctuations in the manufacturing process and are generally modelled by Gaussian or Uniform Distributions (e.g. oxide thickness, threshold voltage and channel length reduction).

We denote by $\mathcal{A}$ the feasible set of the design, operating and process variables:

$$\mathcal{A} = \{(x_d, x_o, x_p) \mid l_i \leq f_i(x_d, x_o, x_p) \leq u_i, \quad i = 1, \ldots, m\}. \tag{2}$$

The Yield represents the probability of a circuit to be in full working order for a certain design choice x_d, subject to given operating variables x_o and taking account of the process and environmental variations x_p.

In recent years there is an increasing interest in Yield optimization due to both the increasing number of components in a single circuit and the decrease of its size. Handling these conflicting trends is becoming more and more difficult, because they enhance the sensitivity of the circuit performances to the statistical variations in the manufacturing process, such as variations in intra- and inter-die channel length, oxide thickness, doping concentration and so on.

Some major challenges come from the need of making the final design robust toward these variations. This implies the use of accurate computer-aided design processes with computer simulations employed in order to evaluate the circuit performances. Circuit simulations are based on the circuit topology, the mathematical models of the underlying devices, the design, operating and process parameters. Simulations consist in the numeric solution of usually non-linear circuit equations which are used for evaluating the performances of interest. Simulations are generally very time expensive and this reason prompts the interest in developing methods capable of performing reliable analysis with the use of less simulations as possible.

Various optimization methods have been applied to the problem of robust circuit design. For instance, in [7] the authors describe a robust derivative free method for circuit Yield optimization, in [8] the authors use a trust region type algorithm to solve a bilevel circuit

optimization problem, in [5, 12, 14] Geometric Programming has been used in a series of papers for several tasks in circuit design.

Focusing on Yield optimization, two classes of methods are mainly adopted, the Geometric Yield Optimization and the Statistical Yield Optimization [9].

The Geometric Yield Optimization [2, 3] is based on the definition of worst case distances of each performance from its lower/upper bound. The optimal design parameters are those which maximize the worst case distances of the performances. Therefore we are led to a *maxmin* optimization problem, where the minimization determines the worst case distances. The main drawbacks of this approach are:

- the worst case distances must be computed for every performance in every design choice. This gives raise to a complex multi-objective, derivative free optimization problem whose solution is expensive to obtain especially with large dimensions of process parameters and performance features;

- the calculation of the worst case distance generally is a non-linear optimization problem with several local solutions. Therefore the method has to be run repeatedly in order to know the best Yield value that can be achieved;

these kinds of methods are among the most popular in Yield optimization and are employed by WiCkeD [1], a suite for circuit analysis, modeling, sizing, optimization and surrogate model generation. WiCkeD is the result of the research performed on the geometric Yield analysis and is considered an industrial standard for these kinds of applications.

The Statistical Yield Optimization uses a Montecarlo (MC) analysis in order to evaluate the Yield, and has the advantages of greater generality and higher accuracy [13]. However the large number of simulations required to obtain an accurate measure of the Yield makes the use of MC analysis really expensive and almost impracticable in iterative optimization. Therefore the efforts in literature are focused on decreasing the number of simulation while preserving the accuracy of the Montecarlo analysis. These efforts are based on two principal strategies:

- Surrogate Models based methods [4]: these methods create macro-models for the Yield over the design, operating and process variables. These strategies enable the practitioners to explore the design alternatives with little computational effort, but such models suffer from a trade-off between the number of simulations employed for the model training and its accuracy;

- improved MC based methods: these methods employ alternative methods to perform the MC analysis with less simulations, but without losing information on the Yield. Some are based on the Latin Hypercube Sampling (LHS) [19] or the Quasi-Montecarlo method [18]. Other methods employ strategies to avoid useless MC simulation and use advanced optimization strategies to increase the rate of convergence [10].

The approach proposed in this paper is a Statistical Yield Optimization methodology which takes inspiration from both the Surrogate Models based methods and the improved MC based methods. As a matter of facts we combine:

- an accurate surrogate model, the Support Vector Machine (SVM) [20], to generate a reliable MC analysis and evaluate the Yield;

- an efficient derivative free optimization (DFO) method with fast and reliable convergence properties to maximize the Yield.

The MC analysis has to consider only the process variabilities when is generated, and in our methodology the SVM only models the process variabilities with the aim of being as much accurate as possible. Consequently we embed the training of the SVM in the iterations of the optimization algorithm. At every iteration the DFO algorithm selects suitable design parameters, then a SVM is trained that only handles the information on the process parameters. This SVM is then used to generate a MC analysis with a large number of points to calculate the objective function for the DFO. Therefore, our main contributions are:

- The adoption of an efficient derivative free algorithm for circuit Yield optimization that, instead of using macro models over the design, operating and process variabilities to perform the MC Yield analysis, generates surrogates models only in the design points of actual interest for the algorithm. Such surrogate models only handle the process variabilities, resulting in accurate models obtained with less simulations;

- A numerical testing obtained on two real consumer electronics circuits provided by STMicroelectronics, a well known company specialized in the design and production of circuit for consumer electronics. Such experimental results also include the comparisons with the results obtained by using WiCkeD, the industrial standard currently employed by ST-Microelectronics for circuit design.

The DFO algorithm we adopt in our numerical testing is DFL-box [11], a mixed-integer line search based optimization method. We are interested in mixed-integer methods because often discrete parameters must be considered in analog sizing. Such parameters predominantly appear when the layout properties should be considered, like in circuits where the transistor lengths and widths must lie on a manufacturing grid or if transistor multipliers should be used for scaling. The creation of methods suitable for handling discrete variables in circuit optimization is an important topic [15, 16] in circuit design because it is well known that applying continuous optimization and then rounding the results generally leads to sub-optimal solutions. Showing that the proposed methodology efficiently handles circuits with discrete variables is another contribution of this paper.

In the following we will refer to our method for Yield optimization as the SVM-DFO method.

2 Problem formulation

As we said in the introduction, the aim of this work is to determine the design variables x_d in such a way that in the production process the Yield is maximized. The Yield corresponding to a design x_d, under given operating conditions $\bar{x}_o$ is formally defined as:

$$Y(x_d) = \int_{-\infty}^{+\infty} \ldots \int_{-\infty}^{+\infty} \delta(x_p) \cdot pdf(x_p) \cdot dx_p = E\{\delta(x_p)\} \tag{3}$$

where

$$\delta(x_p) = \begin{cases} 1, & x_p \in A_p \\ 0, & otherwise \end{cases},$$

with

$$\mathcal{A}_p = \{x_p \mid l_i \le f(x_d, \bar{x}_o, x_p) \le u_i, i = 1, \ldots, m; \},$$

and $pdf(x_p)$ denotes the probability density function of the process variables x_p. The operating variables are fixed to their worst case values found by using a worst case operating parameters optimization [9]. It is assumed that if the performance specifications are satisfied at the worst cases, they are satisfied for any other feasible values of the operating variables. As it is not possible to calculate analytically the integral in (3) we measure the Yield by means of the expectation value:

$$\hat{Y}(x_d) = \hat{E}\{\delta(x_p)\} = \frac{1}{n_s} \sum_{\mu=1}^{n_s} \delta(x_p^{(\mu)}) = \frac{n_{ok}}{n_s}, \tag{4}$$

where $x_p^{(\mu)}, \quad \mu = 1, \ldots, n_s$ are normally distributed samples of x_p and n_s is the number of samples in the Montecarlo analysis. Therefore, the estimator is given by the number n_{ok} of the samples which satisfy the specifications divided by the total number of samples n_s. It is possible to use $\hat{Y}$ given by (4) as objective function in the optimization problem, but such objective function does not capture the geometry of the problem and it treats in the same way any point out of $\mathcal{A}_p$ no matter how far it is from the feasible region. Therefore, rather than maximizing $\hat{Y}$ in (4) we minimize the following function:

$$\begin{aligned} \phi(x_d) = \quad & \sum_{\mu=1}^{n_s} \sum_{j=1}^{l} \sum_{i=1}^{m} \Big\{ \log\big(\max\{0, l_i - f_i(x_d, \bar{x}_{o,j}, \bar{x}_p^{(\mu)})\} + \epsilon\big) \\ & + \quad \log\big(\max\{0, f_i(x_d, \bar{x}_{o,j}, \bar{x}_p^{(\mu)}) - u_i\} + \epsilon\big) \Big\}, \end{aligned} \tag{5}$$

where ϵ is a positive parameter close to zero and l is the number of operative cases. Function (5) penalizes how much the performances of a point of the MC analysis are outside their bounds. In the case a performance satisfies its specification no penalty is applied at all. Furthermore the logarithm is used to smooth the max function. This is a strategy generally used to handle zero norm problems like in [17]. Therefore the problem we will to solve is given by:

$$\begin{aligned} \min_{x_d} \quad & \phi(x_d) \\ s.t. \quad & x_d \in X_d \\ & x_d^i \in \mathbb{Z}, \ i \in I_z, \end{aligned} \tag{6}$$

where $X_d = \{x_d \in \mathbb{R}^n : l_{x_d}^i \le x_d^i \le u_{x_d}^i, \ i = 1, \ldots, n\}$ is the feasible set of the design variables, and I_z indicates the set of indexes of the design variables that can only assume integer values.

We point out that formulation (6) is the standard formulation for bound constrained mixed-integer optimization problems. Since the derivatives of function ϕ are not available, we need to resort to a derivative free mixed integer optimization algorithm.

3 The SVM surrogate model based optimization procedure

In this section we explain the procedure to generate the MC analysis used to evaluate the objective function ϕ in (5) and we introduce the Derivative Free Optimization (DFO) algorithm used to minimize it.

As said in the introduction the MC analysis of the circuit is performed using a surrogate model given by Support Vector Machines. The methods for the screening of the process parameters and the generation of the SVM are explained in detail in [6]. Every time the DFO algorithm needs to evaluate the objective function ϕ in a design point x_d for fixed values of the operating variables $\bar{x}_o$, the following procedure is executed, as shown in Figure 1:
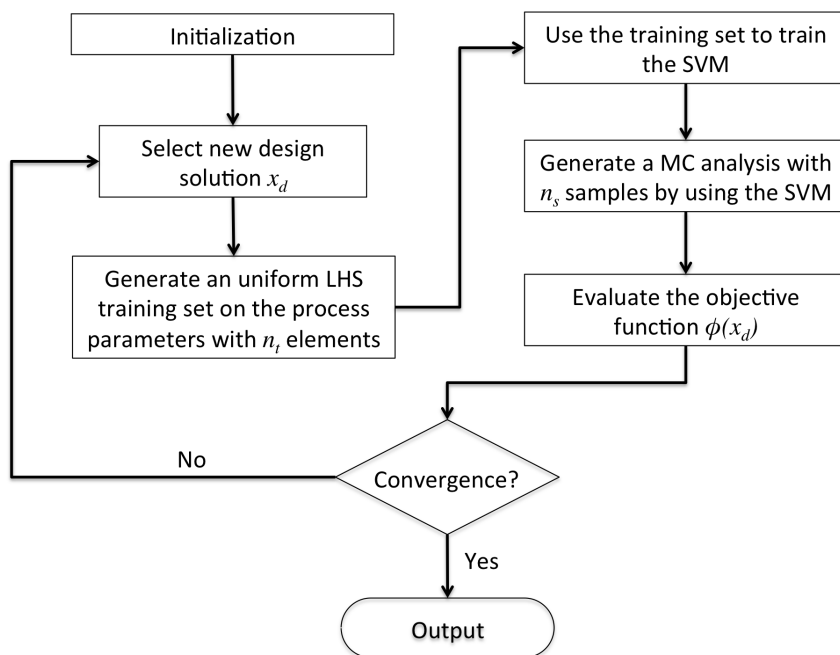

Figure 1: Yield optimization procedure.

1. The values of the design parameters $\bar{x}_d$ are given in input to the MC generation subroutine together with the desired number n_t of samples used for training the SVM;

2. The subroutine generates an uniformly distributed LHS design of experiment for n_t values of the process variables with a standard deviation equal to $\sigma = 5$. We remind the readers that the LHS is created only in respect to the process variables;

6

3. The n_t realizations of the process variables are given to the simulator that runs n_t simulations for the circuit performances, creating the training set for the SVM;

4. The n_t samples are used for training the SVM as surrogate model for the performances of the circuit, given the values of the process variables. For the validation of the model we apply a $k - fold$ cross validation with $k = 5$;

5. The SVM trained model is used instead of the simulator to evaluate a Montecarlo analysis of $n_s = 10000$ samples of the process variables;

6. The 10000 values of the circuit performances are used to calculate the value of (5) in $\bar{x}_d$.

As concerns the DFO algorithm for bound constrained mixed-integer minimization of $\phi(x_d)$, we adopt the algorithm described and analyzed in [11]. This algorithm can be thought as a distributed algorithm in the sense that all coordinates are considered cyclically and a different search procedure is adopted depending on of variable type, according to if it is continuous or discrete. Here we just give a sketch of the algorithm, while we refer to [11] for all technical details. In this sketch $\phi(x)$ is a function of a variable $x \in \Re^n$ with components both continuous and discrete, and X is the hyper-interval such that $\{l^i \leq x^i \leq u^i, \ i = 1, \ldots, n\}$.

Algorithm: a mixed integer derivative-free optimization framework

---

    Input: an initial point $x_0 \in X$.
    Output: a stationary point of $\phi(x)$ (as defined in reference [11])

    repeat
        for $i = 1, 2, \ldots, n$ **do**
            if i-th variable is continuous **then** do a continuous search along i-th direction
            else do a discrete search along i-th direction
            end if
        end for
        Try to (heuristically) improve the current point
    until convergence

The above sketch helps us to understand the main iteration loop of the algorithm which basically analyzes one coordinate at a time and performs a different procedure depending on the type of variable under examination. Two basic ingredients of the method are apparent: the two search procedures for, respectively, continuous and discrete variables.

It is well known in optimization computations that the most time consuming tasks are the line searches, because in such subroutines many evaluations of the objective function are carried out. Therefore it is of main concern to reduce the number of time consuming simulations required to perform a reliable Monte Carlo analysis of the circuit performances subject to the process variations. The use of the SVM as surrogate model of the circuit is precisely intended to overcome this main difficulty.
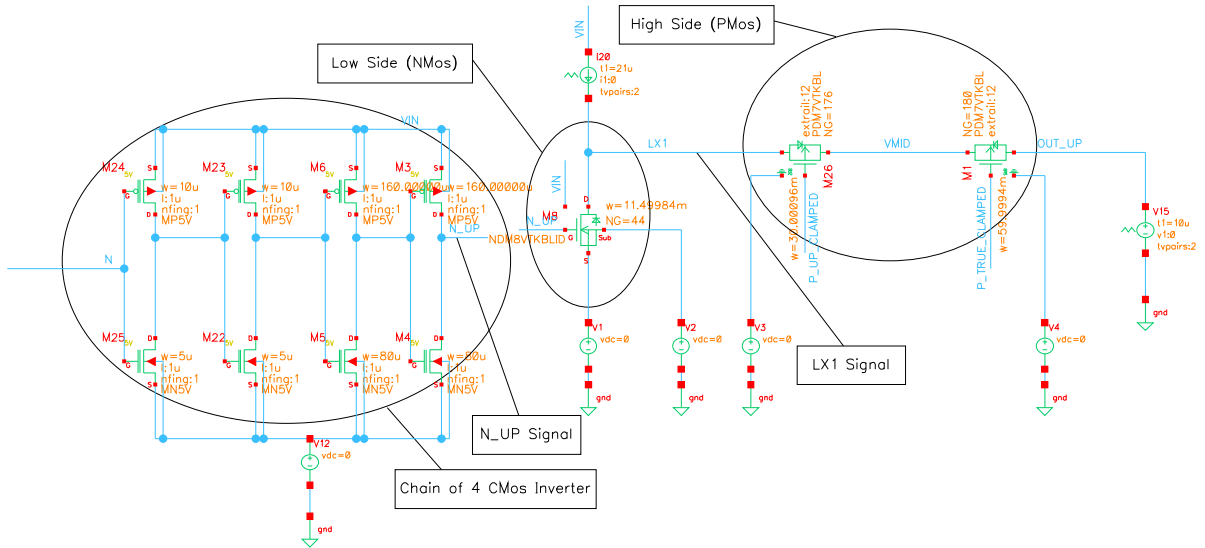
Figure 2: Scheme of the DC-DC converter.

4 Experimental results

In this section we analyze the numerical results obtained for two circuits developed by ST-Microelectronics, a DC-DC converter and a chain of 15 buffers. The numerical tests are performed at ST-Microelectronics headquarter in Catania on a computational grid with more than 800 processors. This brings a further challenge in evaluating the quality of the results. Indeed the processors used for the simulations are chosen according to the load on the grid and the different processors cannot be expected to have the same performances. Consequently the computing time required by the algorithms is not a suitable measure for the speed of the method. Rather we will use the total number of simulations needed in a run, as the simulations are considered the main computational load of the procedure.

In order to ascertain the accuracy of the method we have performed several tests lowering the number of n_t samples in the training set of the SVM. First we have performed tests with $n_t = 50$ training samples, which we considered, by looking at the number of process parameters in both circuits (9 in the DC-DC converter and 13 in the chain of buffers), a suitable number of samples in order to have an accurate SVM surrogate model. Then we have performed tests with 40, 30, 20, 15 and 10 samples.

We report the optimal design parameters for the different runs using the DFO algorithm. Then we evaluate the Yield corresponding to these optimal design parameters using a MC analysis with 10000 samples obtained by using the simulator. Finally we compare the Yield obtained by the SVM-DFO method and the optimal Yield obtained by the circuit designers at ST-Microrelectronicsat using WiCkeD. In the optimization procedures the initial point, provided by the circuit designers, is the same for SVM-DFO and WiCkeD.

4.1 DC-DC Converter

The DC-DC converter of concern increases the voltage level from a partially lowered battery to the voltage level needed for the different circuits composing AMOLED displays in portable consumer electronics devices. This circuit integrates two main components: a step up and an inverting DC-DC converter.

We are interested in the optimal design of a specific section of the converter, an integrated circuit given by:

- a chain of 4 CMOS inverter;

- the High Side (PMOS) and the Low Side (NMOS) output stages;

- the driving signals (N_UP, LX1).

In the circuit, the average value of voltage (and current) fed to the load is controlled by turning the switch between supply and load on and off at a fast rate. The switch operation takes time which has to be kept as low as possible, avoiding both powers on at the same time (i.e. we need to keep low the signal delay). Therefore, the longer this delay is, the more the power loses increase. We report the scheme of the circuit in Figure 2. In this case the chain of 4 CMOS inverters is used as a buffer that drives a large fan-out (the power stage composed by the low side NMos and the high side PMos). The increase in the load capacitance proportionally increases the propagation delay. Buffering with multiple inverter is used to maintain the speed performance of the circuit. The sizes of the components of this device must assume values that are scaled with the sizes of the other components. In order to avoid problems during the layout preparation (strange dimensions of devices) we have chosen to fix the design variables to integer values that refer all to the width of one device (W_M3, width of PMos M3).

As concerns the variables, we have

- **8 Design Variables:**

  - K1, K2 , K3, K4: the scale factor between PMOS and NMOS (discrete);
  - Mult2, Mult3, Mult4: the scale factor along the inverter chain (discrete);
  - W_M3: the width of the last PMOS inverter in the chain (continuous).

  The first seven design variables (i.e. K1, K2 , K3, K4, Mult2, Mult3, Mult4) can assume only the integer values $x_d^i \in \{1, 2, \ldots, 10\}$, $i = 1, \ldots, 7$. The last variable, W_M3, must satisfy the bound constraint $10^{-3} \leq$ W_M3 $\leq 1.6 \cdot 10^{-3}$.

  We remark that the width of a specific component in the chain can be easily obtained by the width of the last PMOS (e.g. W_M4=W_M3/K4, W_M6=W_M3/Mult4, ...). The design variables are reported in Figure 3.

- **2 Operating Variables:**

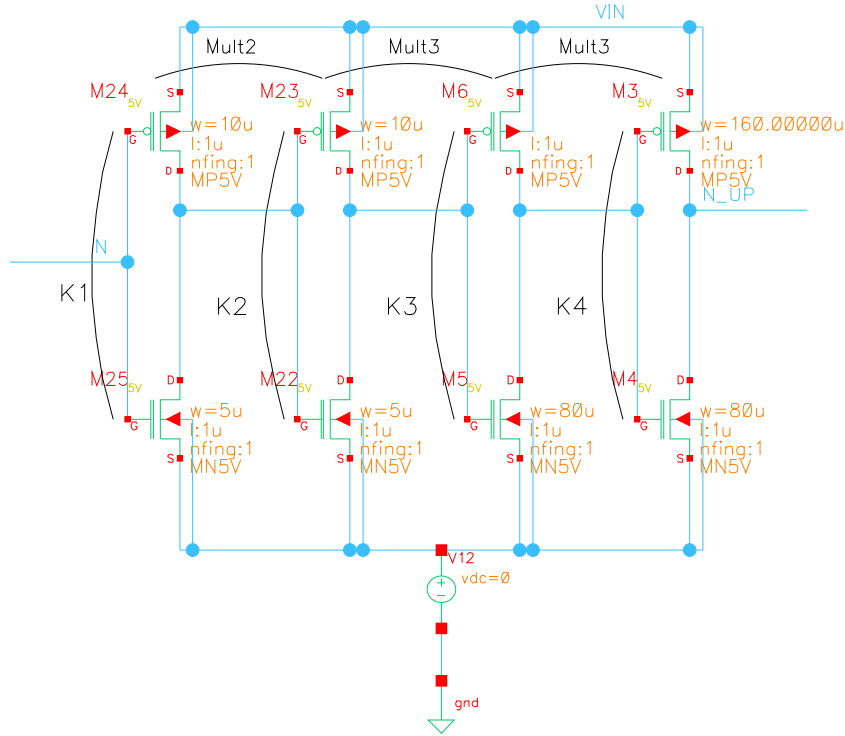 - V, T: supply voltage and temperature.

Figure 3: Circuit variables.

- **9 Process Variables**: The process variables involved in our example are related to the Nmos and Pmos devices; they are 44 but after a sensitivity analysis 35 of them were screened out. The nine remaining variables have a Gaussian distribution centered around the mean value 0 and a standard deviation equal to 1.

As concerns the performance features, they are given by the delays of the circuit (See Figures 2 and 4):

- **3 Performance Features:**

  - $Delay_1$: it represents the propagation delay between the signals V(N_UP) and V(LX1) when V(N_UP) is rising above the VTH1 threshold and V(LX1) is falling below the VTH2 threshold. For this performance the lower and upper bounds l_1, u_1 are respectively l_1= 0ns, , u_1= 21ns;

  - $Delay_2$: it represents the propagation delay between the signals V(N_UP) and V(LX1) when V(N_UP) is falling below the VTH1 threshold and V(LX1) is rising above the VTH2 threshold. Lower and upper bounds are the same of $Delay_1$;

  - $Delay_S$: The Delay Symmetry defined as $Delay_1 - Delay_2$. This performance represents the overall efficiency of the circuit, and it is the performance the designers are the most interested in. Lower and upper bounds l_3, u_3 are respectively l_3= -3.15ns, u_3= 3.15ns.
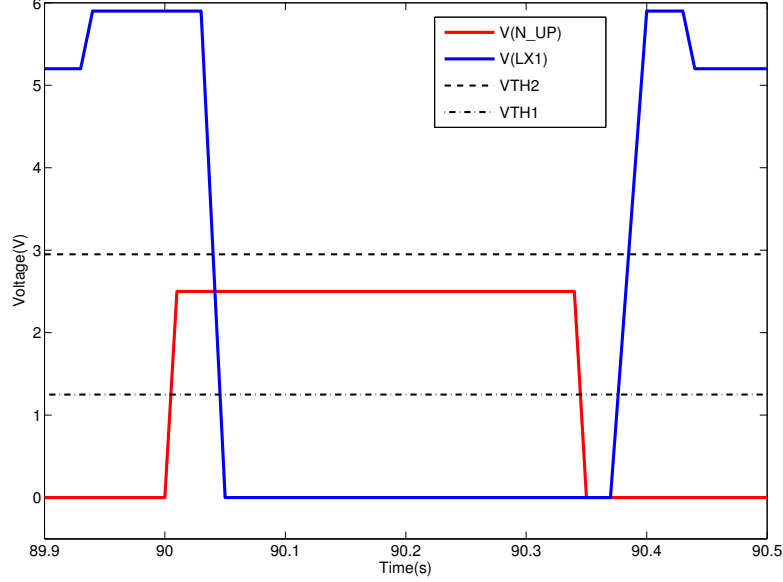
10

Figure 4: Performance features.

The values of the operating variables are fixed at 4 worst cases values:

- Worst case for $Delay_1$ and $Delay_2$ at the lower bound of the performances: V=2.3 V, T=120 $°C$;

- Worst case for $Delay_1$ and $Delay_2$ at the upper bound of the performances: V=4.8 V, T=120 $°C$;

- Worst case for $Delay_S$ at the lower bound of the performance: V=2.3 V, T=-40 $°C$;

- Worst case for $Delay_S$ at the upper bound of the performance: V=4.8 V, T=-40 $°C$.

In Table 1 and 2 we report:

- the number of samples used for training the SVM as surrogate model of the circuit;

- the number of the iterations of the DFO algorithm to reach the optimal solution;

- the number of simulations required in the optimization procedure;

- the Yield obtained by the model with a MC analysis over 10000 samples of process parameters;

- the real Yield obtained using the same 10000 samples;

- the optimal design point x_d^* used in evaluating the estimated and the real Yields.

11

In Table 2 we also report the optimal design point found by WiCkeD. From these results we notice that the DFO algorithm is able to handle the integer variables and to find a mixed integer solution in a reasonable number of iterations. Moreover we remark that the algorithm reaches the same optimal solution no matter the number of samples in the training set. This is an interesting behavior, that indicates that not only the algorithm is able to find a design point with a satisfactory Yield, but also that it is not affected by the decreasing accuracy of the surrogate model. In this particular case we notice that the the estimated Yield of the model is slightly lower that the actual Yield, indicating that the prediction of the SVM is conservative.

In Table 3 we report the results of the SVM surrogate model based method compared to those obtained using WiCkeD. In particular, we report the Yield resulting for each performance and the total Yield. It can be seen that no method is superior to the other. WiCkeD performs better for $Delay_1$ Upper and slightly better for $Delay_2$ Upper, while the SVM surrogate model based method fares better for $Delay_2$ Lower and for $Delay_S$ Upper. The difference in performance between the two methods in the $Delay_1$ Upper case makes the total Yield of WiCkeD superior. On the other side, the most important performance is the $Delay_S$ making the solution obtained by the DFO better from a design point of view because it reaches the 100% of Yield. For what concern the efficiency of DFO, it reaches the optimal solution in much less simulations than WiCkeD, therefore obtaining a superior design point with much less computational effort.

4.2   Chain of Buffers

The second circuit considered for testing the Yield optimization method is a chain of 15 buffers that are used to generate a programmable delay of the input signal. A signal can be delayed by a programmable quantity by switching on or off each buffer in the chain. The buffers are composed by two inverters and consists of four MOS, two NMOS and two PMOS devices, as showed in Figure 5.

The design parameters are the widths W and lengths L of the four MOS devices constituting a buffer in the chain and the goal of the optimization process is to minimize the low-to-high and high-to-low propagation delays and the power dissipated by the circuit when all the 15 buffers of the chain are on. Once the optimal buffer size has been found, all buffers in the

n_t	Iter.	Sim.	Model Yield	Real Yield
10	86	4300	0.8531	0.875
15	86	6450	0.8259	0.875
20	86	8600	0.8240	0.875
30	86	12900	0.8305	0.875
40	86	17200	0.8382	0.875
50	86	21500	0.8314	0.875

Table 1: Results for the DC-DC circuit with different sizes of the SVM training set.
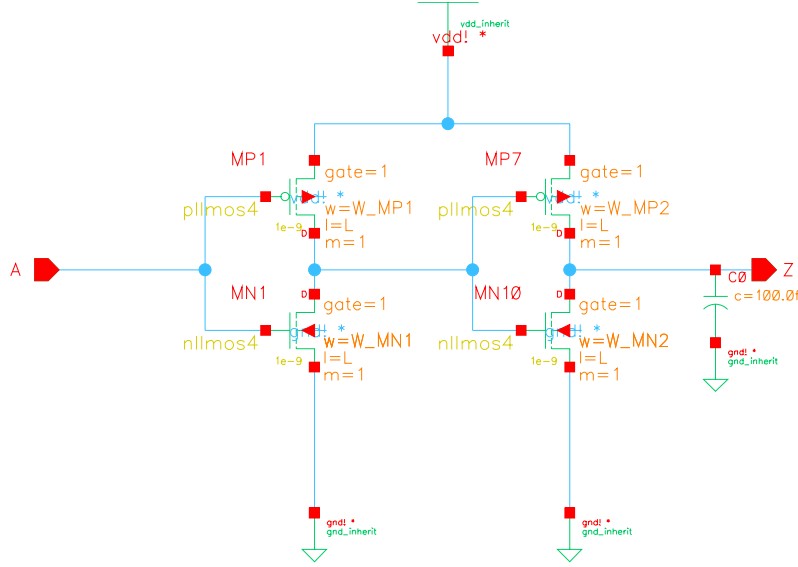
Figure 5: A cell in the Chain of Buffers.

chain will have this optimal size. One additional specification is to minimize the difference between high-to-low and low-to-high propagation delays. For this reason we have defined one additional performance which is the difference between the aforementioned propagation delays. The main objective of the optimization process is to make the behavior of the two delays as similar as possible.

The variables of the circuit are:

- **5 Design Variables** (continuos):

    - W_MN1: Width of transistor MN1, with $10\mu m \leq W\_MN1 \leq 300\mu m$;

    - W_MN2: Width of transistor MN2, with $10\mu m \leq W\_MN2 \leq 300\mu m$;

    - W_MP1: Width of transistor MP1, with $10\mu m \leq W\_MP1 \leq 300\mu m$;

    - W_MP2: Width of transistor MP2, with $10\mu m \leq W\_MP2 \leq 300\mu m$;

    - L: Length of transistors, with $0.28\mu m \leq L \leq 0.40\mu m$.

- **2 Operating Variables**:
  The Voltage V and the temperature T.

	x_d^*
SVM-DFO , $n_t = 10 - 50$	$(1, 3, 2, 2, 9, 3, 6, 1.6 * 10^-3)$
WiCkeD	$(1, 3, 1, 2, 6, 3, 7, 1.6 * 10^-3)$

Table 2: The optimal design parameters for the DC-DC circuit.

13

Performance	Temp.	Voltage	Y. WiCkeD	Y. SVM-DFO
Delay 1 Lower	120	2.3	100%	100%
Delay 1 Upper	120	4.8	92.54%	88.26%
Delay 2 Lower	120	2.3	100%	100%
Delay 2 Upper	120	4.8	95.80%	93.56%
Delay S Lower	-40	2.3	99.26%	100.00%
Delay S Upper	-40	4.8	96.54%	100.00%
Total Yield			90.12%	87.50%
Total Simulations			11000	[4300,21500]

Table 3: Comparisons of the results between SVM-DFO and WiCkeD.

- **13 Process Variables**:
  The initial number of process parameters is 236, and 223 have been screened for a total of 13 variables affecting this circuit.

We consider $m = 4$ **performance features**:

- t_{HL}: High-to-low propagation delay, with $100ps \leq t_{HL} \leq 3ns$;

- t_{LH}: Low-to-high propagation delay, with $100ps \leq t_{LH} \leq 3ns$;

- t_D: $t_{HL} - t_{LH}$, with with $0s \leq t_D \leq 20ps$;

- pw: the power dissipated by the circuit, with $4mW \leq pw \leq 6mW$.

From the preprocessing on the circuit, we can observe an interesting feature. The higher the temperature is, the slower the two delays become. Therefore we only need to analyze the circuit with the temperature at its upper bound, that is $T = 150°C$: if the delays are small enough at such temperature, then they will also be small enough at lower temperatures. The voltage is set at the value V=1.2 V.

n_t	Iter.	Sim.	Model Yield	Real Yield
10	100	1000	0.947	0.772
15	108	1620	0.983	0.964
20	108	2160	0.983	0.963
30	102	3060	0.992	0.963
40	105	4200	0.981	0.960
50	112	5600	0.977	0.953

Table 4: Results for the chain of buffers with different sizes of the SVM training set.

The results of the six runs of the method are presented in Table 4 and 5. In the last row of Table 5 we also report the optimal design point found by WiCkeD. We notice that the accuracy of the SVM surrogate models in predicting the real Yield is quite satisfactory, with

the exception of the case with 10 samples in the training set. Such behavior is expected as the number of samples in the training set is lower than the dimensionality of the predicted function, that is 13. The remaining optimal design parameters all produce a Yield over the 95%.

We notice also that the optimal design parameters found are all close to each other, with changes only at the third significant digit with the sole exception of x_d^{*2} in the run with 10 samples in the training. This shows a certain level of reliability in the proposed methodology. Even with really few samples in the training set, and a limited computational effort, it is possible to find good enough design parameters. The small differences in the results are due to the different values that the objective function reaches because of the different trained models, but such small differences do not prevent to provide a high value of the Yield.

The comparison with the results obtained using WiCkeD is reported in Table 6 and is quite encouraging. In the second column we report the results of WiCkeD for every performance and the number of simulations, while in the third column we report the results of SVM - DFO with the minimum and the maximum Yield for every performance and the minimum and maximum number of simulations for the runs with 15, 20, 30, 40, 50 samples in the training set.

It is possible to see that not only SVM - DFO uses from one tenth to an half of the simulations needed by WiCkeD to find a suitable solution, but also that the Yield of the SVM - DFO is quite superior to the one obtained by WiCkeD. In detail, only for the t_D at the lower bound the Yield of the SVM - DFO is lower by a significant percentage, while at the upper bound it is possible to observe a substantial improvement of SVM - DFO with respect to WiCkeD in all the considered runs. These results clearly show once again that SVM - DFO is able to obtain a better design solution than WiCkeD with substantially less computational effort.

5 Conclusions

In this paper we have presented a novel approach for Yield optimization in electronic circuit design that combines an accurate surrogate model with an efficient derivative-free optimization algorithm which is able to solve mixed integer nonlinear problems. The surrogate models

	x_d^{*1}	x_d^{*2}	x_d^{*3}	x_d^{*4}	x_d^{*5}
SVM-DFO					
$n_t = 10$	2.800E-05	4.069E-05	4.052E-05	3.906E-05	4.154E-07
$n_t = 15$	2.816E-05	4.112E-05	4.068E-05	3.904E-05	4.154E-07
$n_t = 20$	2.816E-05	4.112E-05	4.068E-05	3.904E-05	4.154E-07
$n_t = 30$	2.800E-05	4.106E-05	4.056E-05	3.910E-05	4.158E-07
$n_t = 40$	2.808E-05	4.118E-05	4.068E-05	3.906E-05	4.154E-07
$n_t = 50$	2.812E-05	4.118E-05	4.068E-05	3.906E-05	4.154E-07
WiCkeD	2.820E-05	4.174E-05	4.074E-05	3.918E-05	4.144E-07

Table 5: The optimal design parameters for the chain of buffers.

Performance	Y. WiCkeD	Y. SVM-DFO
t_{HL} Lower	100%	[100%,100%]
t_{HL} Upper	99.2%	[99.0%, 99.5%]
t_{LH} Lower	100%	[100%, 100%]
t_{LH} Upper	98.8%	[98.6%, 99.4%]
t_D Lower	99.6%	[98.1%, 99%]
t_D Upper	93.7%	[97.5% 98.8%]
Power Lower	100%	[100%,100%]
Power Upper	99.9%	[99.9, 100%]
Total Yield	92.1%	[95.3%, 96.3%]
Simulations	10000	[1620,5600]

Table 6: Comparisons of the results between SVM-DFO and WiCkeD.

are imbedded in the optimization procedure so that the complexity they have to handle is only related to the process variables, resulting in reliable models even when the number of circuit simulation is reasonably limited.

The method has been experimented using two real consumer electronic circuits provided by ST-Microelectronics. The optimal design variables found using the method show high values of the Yield even for such difficult test benches. The method also shows a good behavior when really few samples are used to train the SVM, and compares very well with WiCkeD, the software suite used by ST-Microelectronics for circuit design, finding good design choices with less computational effort for both the analyzed circuits. From our computational experience, it seems that it possible to obtain accurate results by using as many training samples for the surrogate model as the number of process parameters. We would suggest to double such number to be conservative.

Taking into account the fact that WiCkeD is being developed as a commercial tool since more than ten years, we can conclude that the method described in this paper in its first stage of experimentation, looks quite promising, and worth of further development effort.

6   Acknowledgement

References

[1] K. J. Antreich, J. Eckmueller, H. Graeb, M. Pronath, F. Eschenkel, R. Schwencker, and S. Zizala. Wicked: Analog circuit synthesis incorporating mismatch. In *Custom Integrated Circuits Conference, 2000. Proceedings of the IEEE 2000*, pages 511–514. IEEE, 2000.

[2] K. J. Antreich, H. Graeb, and C. U. Wieser. Practical methods for worst-case and yield analysis of analog integrated circuits. *International Journal of High Speed Electronics and Systems*, 4:261–282, 1993.

[3] K. J. Antreich, H. Graeb, and C. U. Wieser. Circuit analysis and optimization driven by worst-case distances. *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, 13:57–71, 1994.

[4] S. Basu, B. Kommineni, and R. Vemuri. Variation-aware macromodeling and synthesis of analog circuits using spline center and range method and dynamically reduced design space. In *VLSI Design, 2009 22nd International Conference on*, pages 433–438. IEEE, 2009.

[5] S. P. Boyd, S.-J. Kim, D. D. Patil, and M. A. Horowitz. Digital circuit optimization via geometric programming. *Operations Research*, 53:899–932, 2005.

[6] A. Ciccazzo, G. Di Pillo, and V. Latorre. Support vector machines for surrogate modeling of electronic circuits. *Neural Computing and Applications*, 24:69–76, 2014.

[7] A. Ciccazzo, V. Latorre, G. Liuzzi, S. Lucidi, and F. Rinaldi. Derivative-free robust optimization for circuit design. *Journal of Optimization Theory and Applications*, pages 1–20, Doi: 10.1007/s10957–013–0441–2, 2013.

[8] A.R. Conn and L.N. Vicente. Bilevel derivative-free optimization and its application to robust optimization. *Optimization Methods and Software*, 27:561–577, 2012.

[9] H. Graeb. *Analog design centering and sizing.* Springer, Dordrecht, The Netherlands, 2007.

[10] B. Liu, F. V. Fernández, and G. E. Gielen. Efficient and accurate statistical analog yield optimization and variation-aware circuit sizing based on computational intelligence techniques. *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, 30:793–805, 2011.

[11] G. Liuzzi, S. Lucidi, and F. Rinaldi. Derivative-free methods for bound constrained mixed-integer optimization. *Computational Optimization and Applications*, 53:505–526, 2012.

[12] E. Mintarno, J. Skaf, R. Zheng, J. Velamela, Y. Cao, S. Boyd, R. Dutton, and S. Mitra. Optimized self-tuning for circuit aging. *Proceedings Design Automation and Test in Europe*, 5:586–591, 2009.

[13] A. A. Mutlu, N. G. Gunther, and M. Rahman. Concurrent optimization of process dependent variations in different circuit performance measures. In *Circuits and Systems, 2003. ISCAS'03. Proceedings of the 2003 International Symposium on*, volume 4, pages IV–692. IEEE, 2003.

[14] D. Patil, S. Yun, S.-J. Kim, A. Cheung, M. Horowitz, and S. Boyd. A new method for design of robust digital circuits. *Proceedings of the International Symposium on Quality Electronic Design*, pages 676–681, 2005.

[15] M. Pehl and H. Graeb. Ragazi: a random and gradient-based approach to analog sizing for mixed discrete and continuous parameters. In *Integrated Circuits, ISIC'09. Proceedings of the 2009 12th International Symposium on*, pages 113–116. IEEE, 2009.

[16] M. Pehl and H. Graeb. Tolerance design of analog circuits using a branch-and-bound based approach. *Journal of Circuits, Systems, and Computers*, 21, 2012.

[17] F. Rinaldi, F. Schoen, and M. Sciandrone. Concave programming for minimizing the zero-norm over polyhedral sets. *Computational Optimization and Applications*, 46:467–486, 2010.

[18] A. Singhee, S. Singhal, and R. A. Rutenbar. Practical, fast monte carlo statistical static timing analysis: why and how. In *Proceedings of the 2008 IEEE/ACM International Conference on Computer-Aided Design*, pages 190–195. IEEE Press, 2008.

[19] M. Stein. Large sample properties of simulations using latin hypercube sampling. *Technometrics*, 29:143–151, 1987.

[20] V. N. Vapnik. *Statistical learning theory*, volume 2. Wiley New York, 1998.