ON ROBUSTNESS OPTIMIZATION BASED ON METAMODELS

Anamaria OROS[1)], Marina ŢOPA[1)], Marius NEAG[1)], Monica RAFAILA[2)], Georg PELZ[2)]
[1)] *Technical University of Cluj-Napoca, Str. Baritiu 26-28, Cluj-Napoca, Romania*
[2)] *Infineon Technologies, Neubiberg, Germany*
anamaria.oros@bel.utcluj.ro

Abstract: Electronic systems are usually required to maintain the value of their parameters within set limits over a wide range of conditions, for large variations of internal or external factors. Given the large number of factors that need be considered it is often un-practical to check all and each combination of factors. New approaches, such as the parameter design, are necessary for designing robust systems, with defined performance for all operational conditions, while keeping the development time and production cost to acceptably low levels. This paper describes two methods for parameter design: the classical, analytical approach, based on underlying assumptions and a new simulation based methodology, focused on real-life cases. The proposed method helps the designer to bring a system response to the target and reduce its sensitivity to varying operating conditions. It is demonstrated on a real-life case: optimized sizing of the external circuitry of a low dropout voltage regulator, with the view of reducing its sensitivity to external factors such as temperature and load current. A three-response optimization was achieved also by applying the method on each of the considered outputs and then intersecting the solutions.

Keywords: *Robustness Optimization, Metamodels, Parameter design, Voltage Regulator.*

I. INTRODUCTION

The industry of electronics is coping with great challenges when trying to improve the quality of products. As the complexity of a system increases faster, the verification becomes harder to be handled. The large number of parameters forms a multi-dimensional, continuous verification space. Robust systems are needed in order to reduce the influence of variations. Even more, when talking about systems (as automotive, aircraft), where safety related features are absolutely necessary, they have to be stable and of high performance even if unpredictable variation sources are present.

Robustness optimization is addressed to problems in which the data is uncertain and/or varies. The classical approach to achieve robustness is to redesign the system, using tighter tolerances or better and purer materials [1]. This option might lead to overdesign, is time consuming, expensive and more difficult to manufacture.

To address these issues, an important role is given to simulation-based verification. Multivariate sensitivity analyses are used to find the effect of each parameter on the system. To reduce the effort, the focus has been put on automatic simulation and on planning simulation experiments to yield maximum information with minimum number of runs [1].

There are some optimization tools like WiCkeD [2] that can be used for robust optimization of electronic systems, but optimization is done at the transistor level and does not take into consideration the perturbations induced by a block to the other components of the system.

To take into consideration most of uncertainties (related to materials, geometry, environment, loading, etc.) statistical approaches are to be used; the methods for robustness optimization, that consider variability are based on [3]:

- Monte Carlo simulation;
- Sensitivity-based variability estimation;
- Design of experiments.

Using Monte Carlo (MC) simulation techniques the system is randomly simulated with the purpose of statistically characterizing the response i.e. the focus is on the mean, variance, distribution type, etc. The disadvantage of this approach is the high price: several simulations are necessary for an acceptable estimation.

Sensitivity-based variability estimation is based on Taylor's series expansions – first or second order. The main disadvantage for the first order expansion is that it loses accuracy when responses are not close to linear. The second order expansion is better in terms of approximation error, but computationally expensive.

Design of experiments (DoE) is an approach used to obtain statistical information about how factors and interactions impact the response and identify the reasons for changes in the response [1]. DoE is used as a complement to multivariate data analysis to increase the efficiency. Using DoE in the context of robustness optimization the researcher intends to determine the operating conditions so that the response is close to the desired one and variability is minimal. Some factors can be controlled (control factors) and other not (noises); they both have impact on the response. For instance: temperature is from DoE point of view a noise and affects all the parameters of a circuit. Significant information about factor effects and interactions is extracted with a minimum number of tests; DoE is more appropriate in terms of costs as the other two methods. In DoE the behavior of the system with respect to the factors, e.g. a circuit response, is approximated by a metamodel.

Manuscript received March 14, 2014; revised June 6, 2014

The classical DoE methods used to obtain robust systems are presented in Section II. Section III proposes a simulation based method with no underlying assumptions as the classical ones. Section IV illustrates the proposed method with an example and conclusions are drawn in section V.

II. STATE OF THE ART

The first method for robustness optimization, using statistical tools, was developed by the Japanese engineer Taguchi and had three stages [1]:

1. System design: provides the basic performance parameter, general structure, describes design function and operation.
2. Parameter design: makes system less variable or more robust in the face of variation; optimizes design parameters to meet the quality requirements. This is the most important step, because planned tests are used to find out which of the parameters is more likely to have an important effect on the output and which is the best tuning of those parameters to curtail the variations.
3. Tolerance design: fine tuning of components which are proven to be critical as stated by Step 2.

Taguchi's method was often discussed in literature. Its main drawbacks are described in [4-7]. The two most important ones are: i) it provides no estimation of the effect the noise factors have on the system performance; ii) large amount of effort and simulation time is required to use Taguchi's orthogonal arrays (OA) in experiments, which brings a good coverage of the design space.

Despite the disadvantages, the method has helped to define concepts like robust parameter design (RPD). This is an experimental design exploiting the interactions between controllable and uncontrollable factors. The purpose is to find the settings of the control factors that minimize the response's variation caused by uncontrollable factors. Figure 1 presents the key idea of RPD – the case where there is interaction between a control factor *x* and a noise factor *z*. The control factor *x* has two variation levels: low and high levels. It can be noticed that it is advantageous to have control factors settled at low levels because it produces less variation in the response *y* tightened distribution.

Based on Taguchi's method, Montgomery proposed his own version of RPD [1]. The method emphasizes the idea of control by noise-controllable factors interactions. His method improves the time consumed to perform the experiments because he uses Combined Arrays instead of Orthogonal Arrays. It starts by taking into consideration a system as illustrated in Figure 2, where ***x*** represents the input factors, ***z*** the noise factors and ***y*** the output response. The input factors ***x*** can be controlled in given ranges and the noise factors ***z,*** though uncontrollable, for the sake of experimentation, are considered in a given range.

To solve the RPD problem, Montgomery proposed a three step methodology [1]:

1. The performance of the system is approximated by a model as in equation (1):

$$y(\mathbf{x},\mathbf{z}) = g(\mathbf{x}) + h(\mathbf{x},\mathbf{z}) + \varepsilon , \qquad (1)$$

where ε is the random error, ***g(x)*** describes the main effects of controllable factors ***x***; ***h(x, z)*** involves the main effects of the noise factors *z* and the interactions between the controllable and the noise factors. Both functions are described by equation (2). The advantage of using this kind of model is that both controllable factors and noise factors are placed in the same experimental design:

$$g(\mathbf{x}) = \beta_0 + \sum_{i=1}^{N} \beta_i \cdot x_i + \sum_{i=1}^{N} \sum_{i<j} \beta_{ij} \cdot x_i \cdot x_j ;$$
$$h(\mathbf{x},\mathbf{z}) = \sum_{i=1}^{N} \sum_{j=1}^{M} \chi_{ij} \cdot x_i \cdot z_j + \sum_{j=1}^{M} \delta_i \cdot z_i \qquad (2)$$

where β_i, β_{ij}, χ_{ij}, δ_j are real coefficients and *N* and *M* are the number of the control factors **x** and of the noises **z**.

2. The mean of the model is computed, taking into account Taguchi's assumptions: the noises are distributed normally with zero mean; zero covariance between different noises, the random error has zero covariance. The mean response $E[y(\mathbf{x},\mathbf{z})]$ is given by:

$$E[y(\mathbf{x},\mathbf{z})] = g(\mathbf{x}) \qquad (3)$$

In the optimization process, a constraint is set: the mean of the output response should be at a target/desired value.

3. The variance of the response $Var[y(\mathbf{x},\mathbf{z})]$ is computed with the following expression:

$$Var[y(\mathbf{x},\mathbf{z})] = \sum_{j=1}^{M} \left[\frac{\partial y(\mathbf{x},\mathbf{z})}{\partial z_j} \right]^2 \sigma_{z_j}^2 + \sigma^2 , \qquad (4)$$

where σ_{z_j} is the variance of the noise z_j and σ is the variance of the random error. The additional constraint is that the variability around the target value should be as small as possible.

Montgomery's method is an improvement of Taguchi's because it requires a rough approximation of the number of runs needed, but still it has some initial assumptions such as low order polynomial responses, normal distribution of noise with zero mean, etc. Another step forward was made by Steinberg [6]. He proposed a different expression for the variance, taking into account the covariance between the factors and the noise. Although both Montgomery's and Steinberg's approaches are better than Taguchi's, they have the disadvantage of underlying assumptions that are not realistic: they do not take into account the interaction between noises and the control factors.

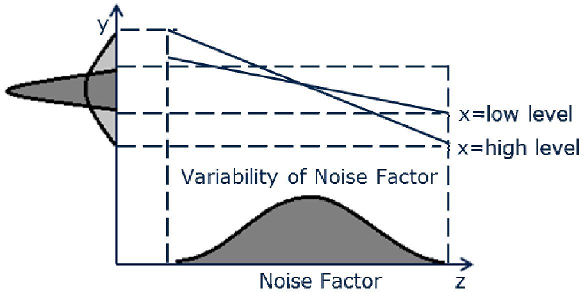

Figure 1: Key idea of Robust Parameter Design

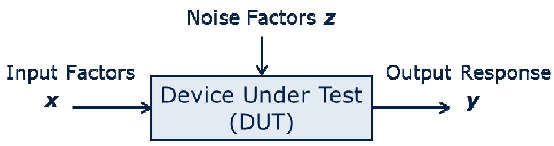

Figure 2: General model of a system

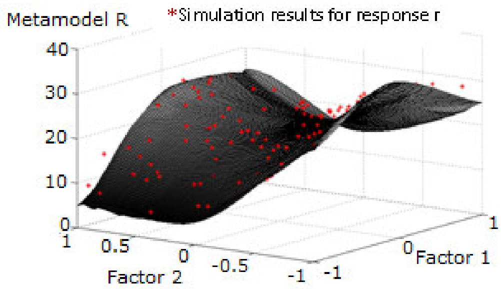

Figure 3: Response approximation.

III. GENERAL FORMULATION

The methods presented so far: Taguchi's, Montgomery's and Steinberg's RPD methods consider the followings:

1) The responses are approximated by a linear model;
2) The noise variables have zero mean;
3) The variance of noise variables is known σ_z^2;
4) The covariance of noise variables is zero;
5) The noise variables and the random errors ε have zero covariance.

In all real cases assumptions 2), 3), 4) and 5) are not true i.e. for real systems noises cannot be estimated. Let us focus on the response approximation; there are three major cases.

- *Case A*: The response follows a linear evolution, which is rarely the case. The mean response is expressed by equation (3) and the variance can be calculated with:

$$
\begin{aligned}
Var\left[y(\mathbf{x},\mathbf{z})\right] = &Var\left[g(\mathbf{x})\right] + Var\left[h(\mathbf{x},\mathbf{z})\right] + Var[\varepsilon] \\
&+2\mathrm{Cov}[g(\mathbf{x}), h(\mathbf{x},\mathbf{z})] + 2\mathrm{Cov}[g(\mathbf{x}),\varepsilon] \\
&+2\mathrm{Cov}[h(\mathbf{x},\mathbf{z}),\varepsilon]
\end{aligned} \tag{6}
$$

where

$$
\begin{aligned}
&Var\left[g(\mathbf{x})\right] = \sum_{i=1}^{N} \beta_i^2 \sigma_{x_i}^2 + 2\sum_{i=1}^{N}\sum_{k>i}^{N} \beta_i \beta_k Cov(x_i, x_k) \\
&Var(h(x,z)) = \sum_{\substack{i=1 \\ j=i+1}}^{N} \beta_{ij}^2 Var(x_i z_j) + 2\sum_{1\le i<j}^{N} \beta_i \beta_j Cov(x_i, x_j) + \\
&\quad + \sum_{i=1}^{N} \beta_i^2 Var(z_i) + 2\sum_{1\le i<j}^{N} \beta_i \beta_j Cov(z_i, z_j) + \\
&\quad + 2Cov\Big(\sum_{i=1}^{N} \beta_{ij} x_i z_j, \sum_{i=1}^{N} \beta_i z_i\Big)
\end{aligned} \tag{7}
$$

Since there is no information about the correlation of the factors (as Steinberg considers) and usually no assumptions can be made, it is rather difficult to obtain a final result [1].

- *Case B*: The response is approximated as follows: ***g(x)*** is quadratic model, ***h(x, z)*** is linear. The expressions for the mean response and for the variance are similar with the ones in Case A. The difference is that in the mean, the function *g(x)* contains the quadratic effects of the input factors [8].
- *Case C*: The response is approximated as follows: ***g(x)*** is quadratic model; ***h(x, z)*** is quadratic, cubic, or exponential. In this case no analytic expressions for the mean and variance can be provided.

While the real cases A and B were offered a solution in literature, the case C is avoided because of lack of information about mean and variance. In this case simulation based methodologies are suitable; they ensure accuracy and efficiency e.g. the method could use approximation models to evaluate points that were not simulated, assess the impact of noise and also integrate in the simulation analysis the optimization techniques [9], [10].

Proposed Approach

Our proposed methodology uses DoE and metamodels: we build an approximation of the response, called metamodel or surrogate from the evaluation responses values for a carefully selected set of design. The advantages of using these approaches are: the evaluation of a metamodel is computationally inexpensive compared to the simulation costs; a large set of samples which approximate the response can be obtained with no effort; the same sample set can be used for multiple purposes i.e. extract more response characteristics or for optimization purposes.

The proposed method has four main steps [11]:

A. **Run simulations. Build and validate the metamodel.**

First we select the response of interest and identify the complete set of factors. Next a test bench must be created and simulations are run with a minimum number of tests. The obtained results are processed: the performance of the device under test (DUT) is approximated, by extracting a metamodel as illustrated in Figure 3. Regression analysis is used to fit the metamodel to the simulation data. The metamodels complexity (polynomial: linear, quadratic, cubic, kringing, DACE, etc.) depends on the test-case and chosen factors [11], [12].If the metamodel is accomplished by polynomial approximation, the response is given by:

$$
y(f) = c_o + \sum_{o=1}^{m}\sum_{i=1}^{n} c_i^{(o)} f_i^{(o)} + \sum_{j=1}^{n-1}\sum_{k=j+1}^{n} c_{jk} f_j f_k \tag{8}
$$

where ***f*** is the *n*-dimensional vector of DUT's factors, ***c*** are the coefficients of the metamodel and represent the effects of factors on the response, *m* is the order of the polynomial. One of the benefits of this method is that each factor f_i may be considered as controllable or uncontrollable. Under normal operating conditions, uncontrollable factors causes variability, but can be controlled during the experiments.

After building up the metamodel, a residual analysis is performed to see if the metamodel is well fitted. The residuals are the differences between observed values and model predicted values. For experiment planning and metamodelling (fitting, evaluation and validation) a library of functions is used [13], [14], [15].

B. **Derive Metamodel.**

In this step, the metamodel obtained in Step A is derived to approximate the following measures: mean, dispersion and extreme points of the response. The procedure of fitting derived metamodels is similar with the one in step A, but they will depend only on the controllable factors, while the noise factors are kept at a fixed level.

The derived forms of the initial metamodel are obtained with the following procedure: first, the set of factors is divided into 2 subsets: controllable and uncontrollable/noise factors. Second, the controllable factors are kept at fixed levels (e.g. nominal values) and the initial metamodel is evaluated only with respect to the noise factors. Third, the noise factors are kept at fixed levels while the controllable factors are varied. The statistical properties of the response are estimated as follows:

$$Mean = \frac{1}{n}\sum_{k=1}^{n} y_k \qquad (9)$$
$$Dispersion = q95(y) - q05(y)$$

where *q95* and *q05* are the 95 and 5% quantiles of the response samples. The obtained metamodels are not more complex than the initial metamodel.

C. **Optimize Metamodel.** The optimization is done according to the principle of Robust Parameter Design i.e. factors' levels have to be chosen to achieve two objectives: set the output response's mean at the desired target while the variability around this target is as small as possible. The task is accomplished by searching the points where both optimization conditions are met. At the end of this step, an optimal set of values for the control factors will be provided. Estimates on the response' dispersion, extremes, with respect to the uncontrollable factors, when the control factors are at the initial, respectively optimal levels are also extracted.

D. **Validate Results.** The results obtained in Step C must be verified if they are optimal: the initial metamodel and the optimized one are compared. The initial and final response's dispersion are computed. The improvement is validated by histograms: the distributions of the metamodels are compared and the optimized metamodel should have a tighter distribution.

Practical Implementation

MATLAB was used as development environment, because it has several advantages including: available libraries of functions for optimized array and vector operations, for optimization and statistical analysis; fast debugging, fast integration with other verification environments, built-in simulators (e.g. Simulink) [15], [16].

Step A is fulfilled when a schematic of the DUT is available; the tests of interest are planned and performed. The simulation flow is the usual one: the test-bench built around the DUT verifies the design by providing several scenarios that check key requirements on the output responses; to cover the design space, during simulation, each input variable of the DUT is varied; the data resulted from simulations is processed using MATLAB functions [17]. A metamodel using regression analysis is built to approximate the behavior of the DUT. The residual analysis is used for validation. When the metamodel is a fit one, then it can be trusted and can be used further on in the analysis. Otherwise this step is repeated using another type of regression, another design of experiments or rechecking the parameters and their range.

We integrated steps B, C and D in the tool presented in Figure 4; it runs in interactive mode, is user friendly and provides reusable data as output.

The first action in the tool is to load the metamodel built in Step A. Using the option *Load Metamodel* a pop-up window appears so that the metamodel is browsed. A *Visualize* button is also available so that metamodels can be plotted for a better understanding of how factors and their interactions affect the response.

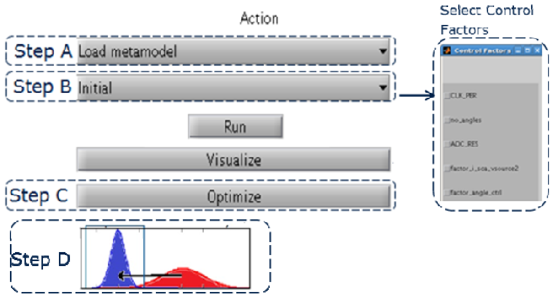

Figure 4: Tool for the proposed method

The second action consists in building derived metamodels depending on the control factors. The proposed method derives metamodels which are less complex. They have to be validated, so another residual analysis is performed in the background. The tool offers the possibility to pick up the factors that one considers to be significant and build a metamodel based only on them. The factors are listed in a pop-up window activated when the type of the derived metamodel is selected. After the user chooses the controllable factors, the *Run* button must be used.

In Step C the derived metamodels are used for robustness optimization i.e. control factors' settings are found so that the mean is put on target and the variance is minimized. The tool identifies by grid searching the setting of controllable factors leading to optimal system performance and provides it as output to be further used for design simulation verification.

In Step D, the obtained set of parameter values must be validated: the initial and final dispersion values for the response are put in contrast and the optimized metamodel is compared against the initial one. The tool provides a pop-up window with their histograms.

IV. CASE STUDY AND RESULTS

In this section the considered DUT is a low drop voltage regulator (LDO). Generally, a voltage regulator is a block designed to automatically maintain a constant output voltage level independent of its input, as long as the input voltage level is higher than the output [18]. The voltage regulator is built using a pass transistor whose gate is driven by an error amplifier. The low dropout voltage regulator is a particular case of voltage regulator. It operates with a very small input-output differential voltage which comes as a great advantage because assures a lower minimum operating voltage. Two measurements are specific for this kind of blocks: load regulation i.e. change of output voltage depending on the change of the load current and line regulation i.e. change of the output voltage depending on the input voltage. Among these, there are other important responses that could be of interest:

- *Dropout voltage*: difference between input and output voltage that allows the circuit to regulate the output voltage. Its value is dependent on the junction temperature and load current and should be as small as possible to minimize the power dissipation [18].
- *Quiescent current*: difference between input and output currents. Its value depends on the ambient temperature and must be kept low [18].
- *Output current*: the current that the voltage regulator injects in a load. Its maximum value should be limited even for low voltage levels at the output [18].

- *Reset pulse delay time*: the time needed for a reset -is measured when there is a low to high transition in the output voltage and, ideally, should be as small as possible [18].
- *Temperature coefficient*: the change of output voltage with temperature.

Figure 5 shows the LDO application setup. The LDO's main features are: operating DC supply voltage range 5 V-28 V, output current up to 150 mA, low dropout voltage less than 0.5 V, current consumption during idle (quiescent current) less than 500 μA, protection features (over-current, over-voltage), programmable reset pulse delay with external capacitor. In this paper we considered the following LDO responses: dropout voltage V_{dr}, time reset delay t_{RD}, output current I_{OUT}.

First of all we will focus on the minimum dropout voltage V_{dr}. If the target is set around 200 mV then the output current (I_{OUT}) has a value below 100 mA. The main factors that are taken into consideration in this test are: the resistance R_1 and capacitance C_1 of the output filter, the load current $I_Q$and the junction temperature T_J. Table 1 provides the minimum, maximum, nominal values and type of the factors.

The methodology presented in the previous section was applied on the considered output V_{dr}. Initially, a test-bench was built. The simulations were performed on a SPECTRE model of the chosen system. The factors described in Table 1 were modified within their set ranges. The simulation time for a test took less than 3 minutes. The change of the response was monitored and a quadratic metamodel was fitted and validated. Figure 6 shows the metamodel for the dropout voltage V_{dr} versus factors. Each curve shows the evolution of V_{dr} when only one factor is varied and the others are fixed; factors I_Q and T_J have great impact on the response.

To derive the metamodels, factors were divided in two subsets – see the column *Type* from Table 1. Control factors are considered to be elements of the output filter: R_1 and C_1. The uncontrollable factors are the temperature T_J and the load current I_Q. Using the tool, new quadratic metamodels are obtained. Figure 7 presents the obtained metamodels for the dispersion and mean response. Notice how the derived metamodels depend on R_1 and C_1.

The optimization for the dropout voltage is done considering the following constraints: the mean has to be about 160 mV and the variance near it as small as possible. The provided solution is a set of values for the control factors that fulfills the constraints.

Robustness is measured either using the variance or percentile difference; in our paper we chose the second metric. Figure 8 illustrates the results validation: the red histogram corresponds to the initial metamodel while the blue one to the optimized metamodel. One can observe that the variance of the response decreased i.e. if the initial response was distributed between 0.09 and 0.26, after optimization the response is squeezed between 0.1 and 0.24, that means an improvement of about 18%.

Our next goal is to optimize the system to meet three major specifications. As the methodology is already implemented in a user-friendly tool, the task of multiple-response optimization becomes easy to fulfill. The other two responses i.e. time reset delay t_{RD} and output current I_{OUT} depend on the same factors as the dropout voltage i.e. three controllable factors C_1, R_1, C_T and one noise factor T_J. Using the built tool a set of optimal values is returned for each response and the intersection of these sets is the optimal space for the robustness optimization of the three responses.

Table 2 presents the results. It can be noticed that the value of the reset capacitor C_T is the same no matter the response; as for the output filter elements (C_1 and R_1) it was not possible to find a single solution, because the responses are conflicting. The optimization process results in a set of solutions that, mathematically, are equally good. The optimal solution can be found by defining the exact application of the LDO and considering possible trade-offs between the optimized responses. Considering that these values neutralize the effect of the temperature which is a noise factor, we can state that our proposed methodology can be also used for multi-response optimization.

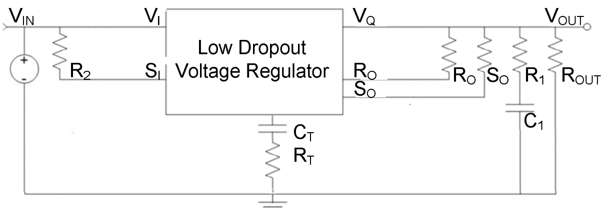

Figure 5: Application setup for the LDO.

Table 1: Factors divided in two subsets: controllable and noise factors.

Parameter [Unit]	Name	Min	Max	Nom.	Type
Output filter capacitor [μF]	C_1	1	10	4.7	Control
Output filter resistor [Ω]	R_1	0.1	1	1	Control
Load current [mA]	I_Q	10	100	50	Noise
Junction temperature [^oC]	T_J	-40	125	25	Noise

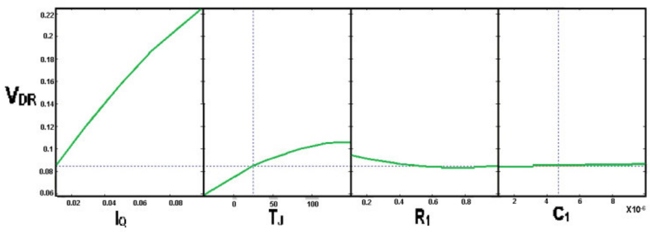

Figure 6: Metamodel for dropout voltage versus factors

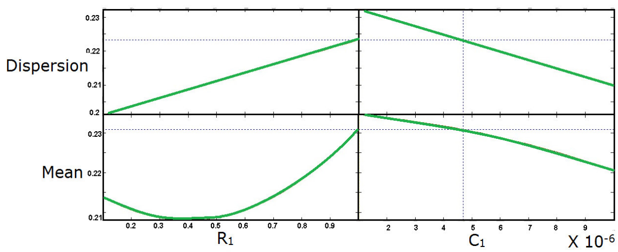

Figure 7: Derived metamodels: Dispersion (top), Mean (bottom).

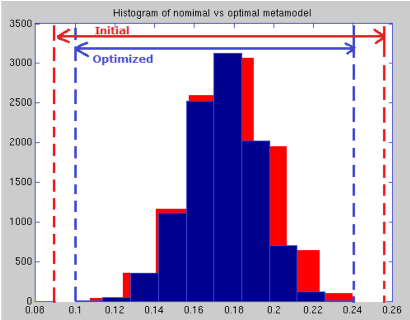

Figure 8: Initial and optimized distributions of the dropout voltage

Table 2: Values of factors that global optimize the LDO

Factor	Min	Max	Nom	Optimal range
Output filter capacitor C_1 [μF]	1	10	4.7	[8.875, 10]
Output filter resistor R_1 [Ω]	0.1	1	1	[0.1, 0.2]
Reset capacitor C_T [nF]	80	120	100	100

V. CONCLUSIONS

Robustness optimization is addressed to problems in which the data is uncertain and/or varies. The classical approach to achieve robustness is to redesign the system, using tighter tolerances or better and purer materials. This option might lead to overdesign, is time consuming, expensive and more difficult to manufacture.

The classical approach, based on DoE, exploits the interactions between controllable and uncontrollable factors and finds the settings of the control factors that minimize the response's variation caused by uncontrollable factors. If little or no information about the variance, covariance of the factors and noises is available, the classical methods cannot provide valuable solutions.

The paper proposes a simulation-based method for robust optimization, which can be applied on any system, in real-life cases, because no underlying assumptions are considered. The method builds first a metamodel of the response from the evaluation responses values for a carefully selected set of design; it considers all factors and their interactions, any type of regression. Next the method derives metamodels of the mean, dispersion and extreme points of the response, finds control factors' settings that put the mean on target and minimize the dispersion, provides them to the user for validation. The method is integrated in a MATLAB tool which might come very helpful for those who seek design optimization solutions in short time.

The method is illustrated by an example: the robustness optimization of the dropout voltage response of a voltage regulator. It was shown that for a desired target, the response's distribution was squeezed with 18%; that means the system became less sensitive to variations. Then the method was used for optimizing other two major specifications: time reset delay and output current. By intersecting all the obtained optimal sets of factors, a multiple response optimization was fulfilled.

Further work will be devoted to develop an automatic multiple response optimization based on the proposed robustness optimization method.

AKNOWLEDGMENT

This research project is supported by the German Government, Federal Ministry of Education and Research under the grant number 01M3195 in the project RESCAR 2.0.

REFERENCES

[1]. Montgomery, *Design and Analysis of Experiments*, John Wiley & Sons, 2013.
[2]. F. Schenkel, M. Pronath, H. Graeb, Design Centering and Mismatch Analysis for Analog Integrated Circuits, *D.A.T.E Designer Forum*, 2002.
[3]. P.N. Koch, R.-J. Yang and L. Gu, Design for six sigma through robust optimization, *Structural and Multidisciplinary Optimization*, 26, pp. 235–248, 2004.
[4]. G. Box, S. Bisgaard, C. Fung, An Explanation and Critique of Taguchi's Contribution to Quality Engineering, *Quality and Reliability Engineering International Journal*, 1988.
[5]. G. Box, K.B. Wilson, On the Experimental Attainment of Optimum Condition, *Journal of the Royal Statistical Society*, Series B, no. 13, pp. 1-45, 1951.
[6]. D. Steinberg. D. Bursztyn, Noise Factors, Dispersion Effects and Robust Design, *Statistica Sinica,* no. 8, pp. 67-85, Tel-Aviv University, 1998.
[7]. A. Khuri, Response Surface Models with Mixed Effects, *Journal of Quality Technology*, Vol. 28, pp. 177-186, 1996.
[8]. G. Dellino, J.P.C. Leijnen, C. Meloni, Robust Optimization in Simulation: Taguchi and Response Surface Methodology, *International Journal Production Economics*, no. 125, pp. 52-59, 2010.
[9]. G. Deng, *Simulation based Optimization*, A dissertation submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy, Mathematics and Computation in Engineering, University of Wisconsin-Madison, 2007.
[10]. N. Stander, K.J. Craig, On the Robustness of a Simple Domain Reduction Scheme for Simulation-Based Optimization, *Engineering Computation*, Vol. 19(4), pp. 431-50, 2002.
[11]. A. Oros, M. Rafaila, M. Ţopa, M. Neag, G. Pelz, Application-oriented robustness optimization based on metamodels, *Proceedings IEEE 18th International Symposium for Design and Technology in Electronic Packaging (SIITME)*, pp.37-41, 25-28 October 2012.
[12]. H. Fang, M. Rais-Rohani, Z. Liu, M.F. Horstemeyer, A comparative study of metamodelling methods for multi objective crashworthiness optimization, *Computers & Structures*, Volume 83, Issues 25–26, September 2005.
[13]. G. Dellino, *Robust Simulation-Optimization Methods using Kriging Metamodels*, Doctoral Thesis, UniversitaDegli Studi di Bari, 2009.
[14]. M. Răfăilă, *Planning Experiments for the Validation of Electronic Control Units*, Doctoral Thesis, Faculty of Electrical Engineering and Information Technology, Vienna, 2010.
[15]. MATLAB, Statistics Toolbox, http://www.mathworks.com/help/toolbox/stats/
[16]. MATLAB, Curve Fitting Toolbox, http://www.mathworks.de/products/curvefitting/
[17]. M. Rafaila, C. Decker, C. Grimm, G. Pelz, *Design of Experiments for Effective Pre-silicon Verification of Automotive Electronics*, Proc. Forum on specification and design languages, 2009.
[18]. J. Williams, "*High Efficiency Linear Regulators*", Linear Technology, Application note, March 1989.