APPLICATION FITNESS ANALYSIS BASED ON METAMODELING TECHNIQUES – A CASE STUDY: E-BIKE

Alexandra IOSUB[1,2] , Ingrid KOVACS[2,4] , Andi BUZO[2] , Jerome KIRSCHER[2] , Georg PELZ[2] , Liviu GORAS[1,3] , Arcadie CRACAN[1]
[1] *'Gheorghe Asachi' Technical University of Iaşi, Romania,* [2] *Infineon AG Neubiberg, Germany;*
[3] *Institute of Computer Science, Romanian Academy, Iaşi,* [4] *Bases of Electronics Department, Technical University of Cluj - Napoca, Romania*

Abstract: **Performance and behavior verification of electronic components and systems in order to fit applications requirements are indispensable nowadays. There is a particular interest in finding the application yield caused by the process variation of the electronic components and revealing the component's factors which cause failure in the application. In order to address these two problems, the Failure Probability (FP) estimation of an electronic system is computed, and the impact of component's factors is analyzed, by considering factors variations as well as various operational conditions. Since FP estimation can be computationally expensive, it is necessary to find more efficient methods. In this paper, we propose a method that can both estimate FP and quantify the impact of factors based on behavioral system modeling and system's performances metamodeling. FP is calculated by re-sampling the metamodel, hence, converting a small numbers problem into a large numbers problem. A comparison between this method with other existing methods such as counting and distribution fitting is made for an E-Bike case. The results show that the approach based on metamodeling can lead to comparable results with state-of-the-art for FP estimation methods with the advantage of revealing the factors which cause the application failure.**
Keywords: *Application Fitness, FP, DoE, Distribution Fitting, Metamodeling, E-Bike, SystemC-AMS.*

I. INTRODUCTION

The goal of this study is to verify whether a component, evaluated with regard to its electrical specification, will equitably serve the final application that works in an environment with uncertainties. The impact of the electrical components can be verified based on the component specification combined with the verification of the component within the application it will work in. This kind of analysis leads to valuable conclusions regarding the impact that different component characteristics may have on the application. In this paper, we propose a methodology that efficiently determines if the application requirements are satisfied when a component with known characteristics is used.
The component's factors can be a component feature like an error, gain, delay, or a subcomponent's (a component inside the considered component) feature.

We split the application fitness problem in two parts:

(1) *FP estimation*: refers to the assessment of a candidate component in the application considering that in a lot of components the performance varies due to the fabrication process;

(2) *Factor impact estimation*: refers to the revealing of a candidate component's factors which cause the application failure, and estimate their impact.

In the former case, we try to estimate the failure probability, i.e. what percentage of components will lead to the violation of the requirements at the application level. In the latter case, we try to estimate the relationship between the component's characteristics and application performances in order to detect the yield detractors, i.e. the component characteristics whose variation causes the yield loss.

Both issues are approached with the use of metamodels. The proposed methodology is evaluated with an E-Bike application where the candidate components are current sensors.

II. RELATED WORK

There are few investigations regarding the component assessment inside an application. State of the art especially offers solutions of yield component analysis and less discussion is done for the application yield problem. In [1] a multidimensional method is proposed to analyze the importance of components in a system and their effects on the reliability from four aspects: centrality, entropy, costumer's preference and service priority. The presented method offers an interactive approach to identify the significant set of components during the architecture design phase. However, the limitation of the proposed methodology is that the approach uses concepts from research of influence and importance in social and online networks, which are quite different contexts than electro-mechanical systems. Another reliability approach, but to transmission maintenance planning is described in [2]. It focuses on quantified impacts of a planned transmission outage on the overall system reliability, rather than the status of individual system components. Paper [3] presents a probabilistic extension to Interface Automata applied for quantitative model checking and analysis, addressing the compositional construction of software specifications problem. In [4] a new framework for component-based system development is presented that combines formal specification and verification of functional requirements with a method for representing stakeholders' qualitative preferences over the properties of a system.

Manuscript received December 3, 2017; revised December 13, 2017

The component assessment problem is discussed in [5], where various simulation methods are used to estimate small FP and also the importance of sampling is discussed.
In [6] it is shown that straightforward sampling of a surrogate model can lead to erroneous results, even if the surrogate model is accurate enough. A hybrid approach is proposed by sampling both the surrogate model in a ‘‘large” portion of the probability space and the original system in a ‘‘small” portion.

The main limitation of state of the art is the nature of the preserved systems.

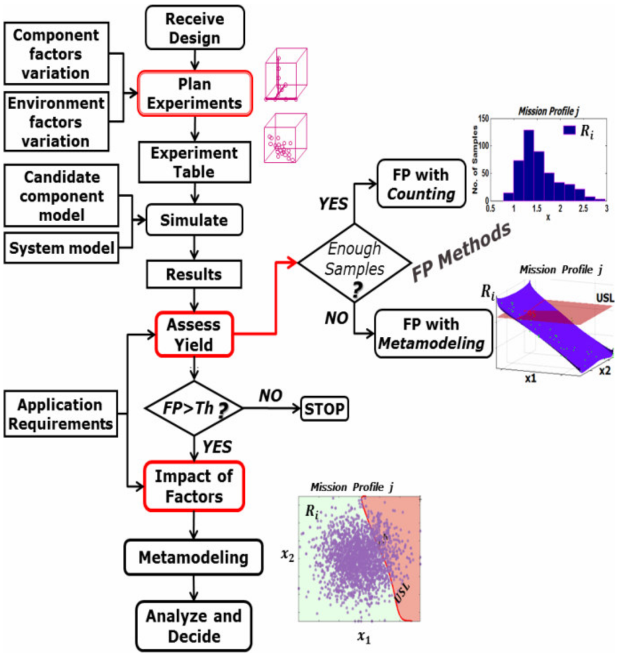

Figure 1: Application Fitness Flow

Most of them are related to software applications and software architecture reliability, or, addresses small electric sub-systems (such as standalone component from the final application) and not the behavior of an electric component in the context of the all application functionality and requirements.

III. APPROACH DESCRIPTION

The main goal of this work is to verify the suitability of a candidate component in the target application when several sources of uncertainty are considered, such as load and process variation. In order to accomplish this, we propose a methodology for the application yield assessment -*FP estimation*, extended with solutions for the *Factor impact estimation*.

The application yield (*FP estimation*) is assessed by using the statistical analysis that considers a known distribution of the parameters. Knowing the Probability Density Function (PDF) of the input vector $\bar{X}$ with k factors and the model response $R_i(\bar{X})$ of the i^{th} system performance, we are able to estimate the FP of the system. In a general application we have defined:

$$R_{i\_LSL} \leq R_i(\bar{X}) \leq R_{i\_USL} \quad (1)$$

where $R_{i\ USL}$ and $R_{i\ LSL}$ are the upper and lower specification limits for the target performance. The FP can directly be estimated by using simple counting based on Monte Carlo sampling. As FP is in general required to be ~10^{-5} it needs a large number of simulations to obtain an estimate with a narrow confidence interval. In many applications it can be a problem since the performance function $R_i(\bar{X})$ is the output of a complex model which requires a long simulation time. To overcome this issue, we propose a flow (Fig.1) which includes a solution based on metamodeling techniques [9] where a mathematical approximation of $R_i(\bar{X})$ is derived.

A. *General Flow Description*

The proposed flow from Fig.1 starts with the “Receive Design” step where the available (or initial) system design is proposed for the investigation. Then, a proper plan of experiments (various types of DoE) is chosen by using a mix of design factors (continuous, discrete). For the analysis purpose, only the factors of the candidate component (the component for which the application fitness is verified) are used in the experiment design while the rest of the system factors are kept constant at their nominal values. Having all models, we automatically perform simulations according to the experiment table. Next, all simulation results are collected and with the existing application requirements the “Assess Yield” phase can be applied. Here, we estimate the FP of the application performances in each given operating point. The FP is estimated by using different methods. In Section V we compare the estimation of FP obtained based on metamodeling techniques with classical counting and distribution fitting approach “distfit” [8]. The reason of it is that the metamodeling approach is suitable for the second problem of *Factor impact estimation.* Therefore, the same approach will be able to solve both addressed problems.

If the FP results higher than a preset threshold, then the “Impact of Factors” step takes place because we want to find out what are the yield detractors. For this, as anticipated, we propose the metamodeling approach as a solution. Here, the computed metamodels are used to assess the impact of each component factor in each performance (the metamodel output). By using interactive plots, the impact of factors is revealed regardless the potential interaction between factors. By using contour plots, a 2D Feasible Space is drawn: the metamodel is evaluated by selecting only 2 factors while the rest are set to their nominal values. The generated contour plots are used for a better illustration of the factors interaction impact.

The flow ends with the “Analyze and Decide” phase, where, based on previous results, the engineers can take the proper decisions (such as re-design, system optimization or relaxing of the system performances) in order to overcome the failure mode.

B. FP Estimation by Metamodeling Approach

Based on the proposed flow from Fig.1, after the simulation step, all results are collected and the distribution of each performance is obtained. Now, the FP can be estimated. In our approach, each performance has distinct specification limits for different operating conditions proper to the target application [9]. The basic steps for a metamodel realization are presented in Fig.2: The main idea is to use few simulations in order to build a metamodel that can predict the response for any combination of the values of factors. Then, using the model we can generate millions of response values without the need of simulations. If the response is obtained from data where factors are varied as in reality, then the distribution of the response will be similar to the one that could have been obtained by the production process. At this point, having millions of data, the FP can be estimated by simple counts.

1. Metamodel fitting and Validation

The metamodel is obtained by applying Least Squares Method (LSM) in order to obtain the regression coefficients [11] of the predicted response. In our case, we obtained metamodels of different degree of complexity: linear, quadratic, linear and quadratic with interaction effects, and also exponential expressions.

Next, the obtained metamodel is validated by the "Coefficient of determination" metric or R^2 (R squared). R^2 specifies the proportion of the variance of the dependent variable (*y*) predicted from the independent variables (*X*). It can be computed as follows:

$$R^2 = 1 - \frac{RMSE}{RMSE_0}; 0 < R^2 < 1 \quad (2)$$

where RMSE is the Root Mean Squared Error of the difference between predicted output $\hat{y}$ and actual output y and $RMSE_0$ is the Root Mean Squared Error of the difference between actual output y and the average of the actual output values.

For metamodeling we used three types of methods: GLM (General Linear Model) where the predictors are combined in a linear way to model their effect on the output; GMP (Gaussian Mixture Process) – a nonparametric kernel-based probabilistic model [7] and RTM (Regression Tree Model) which is more flexible in the case of nonlinear patterns in the responses.

2. New Data Generation

When the metamodel is initially trained it has to be constructed in such way to obtain a complete space exploration. A Grid Search design sampling method is indicated to be used, but this may lead to a high number of simulations. Therefore, we choose a Full Factorial design (meaning every combination of factors' extremes) because the number of factors is small. However, for a better space filling, we complete the design with a random uniform based on Latin Hypercube Sampling.

On the other hand, when new data should be generated for the FP estimation based on the metamodel sampling, a design with original factor's distribution must be used for each factor. For example, when the metamodel is used to predict new output (system response) values, we evaluate the metamodel by using a normal distribution for each factor. We generate in this way a number of 500000 samples for each design factor.

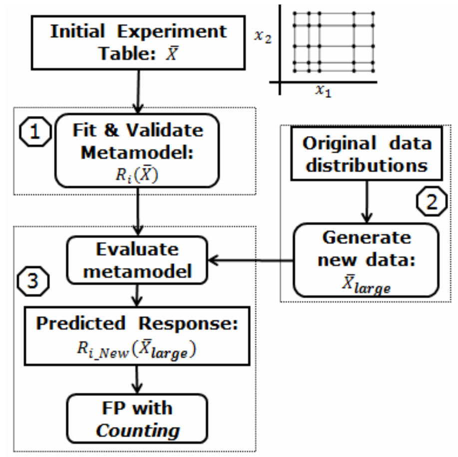

Figure 2: Flow for FP with Metamodeling

3. Computing FP through counting

The new response values are predicted based on obtained regression model, through interpolation. At this step, no additional simulation is required, only the metamodel evaluation. By generating sufficient data points (from previous step), we can now estimate the FP ratio by apply simple counting method:

$$FP_{count} = nRuns_{FAIL}/nRuns \quad (3)$$

where $nRuns_{FAIL}$ represents the number of simulations which violates the received application requirements and $nRuns$ is the total number of simulations.

Even if the component's factor impact estimation can be solved by sensitivity techniques, our solution for this problem is based on metamodeling methods. We justify the proposed solution by enumerating a few (those who count for the addressed problems) metamodeling advantages:

- the metamodel can reveal the interaction between factors (when it is the case);
- the factors impact can be quantified (we can deduce a specific factor value that produces a system failure case);
- through interpolation, the metamodel can predict new response values for un-sampled factor's values. Therefore, we are able to obtain complete response information with fewer simulations, reminding that a sensitivity analysis step would require supplementary simulations.

These characteristics of the metamodel are important when complex systems are analyzed because these systems come with problems regarding the computational budget and time constraints.

IV. E-BIKE CASE STUDY

As a case study, we investigated an E-Bike application. An E-Bike is a bicycle with an integrated electric motor used for propulsion. Such an application model involves dealing with heterogeneous simulations (digital and analog), and with a long simulation time which has to be spent in order to reach the steady-state regime, (transients are characterized by large time constants due to load inertia). The model of the E-Bike (Fig.3) was implemented by using SystemC-AMS language and it is described in [9]. The current sensor based on a shunt resistor (Fig.4) is selected as a candidate component. This sensor is placed at the source of the low-side switch from the inverter block. The circuit performs a current to voltage conversion via the shunt resistor $Rshunt$ and amplification corresponding to $GainA$. A source of noise, caused by the differential input stage of the amplifier, is also introduced in the model.

The main sensor equation is:

$$V_{out} = OffsetA + GainA(Rshunt \cdot Is) + V_{pkNoise} \qquad (4)$$

where $V_{pkNoise}$=$LevelNoise \cdot V_{rms}$ is the peak noise corresponding to the measured root mean squared noise (V_{rms}), V_{out} is the output voltage of the sensor, *OffsetA* is the input offset of the amplifier stage, and I_s is the one phase current flowing through the motor winding. *LevelNoise* is an adjustment factor introduced to assess the impact of the noise level on the application performances.

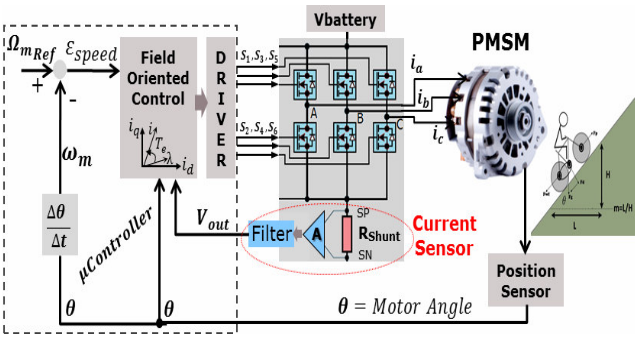

Figure 3: E-Bike application

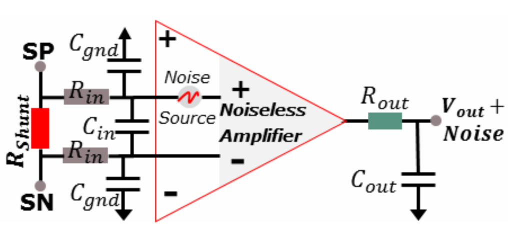

Figure 4: Candidate Component: Current Sensor based on shunt resistor

V. EXPERIMENTAL SETUP AND RESULTS

This section presents the results obtained after applying the described methodology (Section III) on an E-Bike application. The application yield is analyzed by using FP estimated for each performance. In our study, six application performances ($i = 1, \dots, 6$) are analyzed: $R_i(\bar{X})$={*Torque Ripple*; *Speed Ripple*; *Efficiency*; *Speed Error*; *Speed Overshoot*; *Acceleration Time*}. Then, the impact of factors in all six performances is quantified by using metamodeling techniques.

A. System Performances and Mission Profile

Each performance is evaluated in 3 different operating conditions (also called Mission Profiles) selected by a categorical factor with $MP1$, $MP2$, $MP3$ labels. These labels are ordered from perfect to suboptimal conditions [9]. The metrics used for each performance are shown in Table 1, while in Table 2 are indicated the USL and LSL values (upper and lower specification limits) for each target performance in each Mission Profile. If a performance has just one specification limit, then, the other one is denoted with (-) in Table 2.

Fig.4 shows the system signals (for which the performances/attributes from Table 1 are assessed) based in the selected mission profile label. For the *Electrical Torque* (blue line) developed by the motor we assess the *Torque Ripple* measured under steady state conditions. For the *Mechanical Speed* signal, four responses are investigated: *Speed Ripple* and *Speed Error* (measured in steady state); *Speed Overshoot and Acceleration Time* (measured in transient). The last system response, *Efficiency*, is the ratio between output and input power (measured in steady state).

Table 1: System Performances

Performance	Metric	Metric Type
Torque Ripple [Nm]	$3\sigma_{Te}$	Steady State
Speed Ripple [rad/s]	$3\sigma_{Wm}$	Steady State
Efficiency [%]	$\frac{P_{out}}{P_{in}} = \frac{avg(\Omega_m T_e)}{V_{Batt} I_{consumed}}$	Steady State
Speed Error [rad/s]	$\Omega_{m_{Ref}}$-$\Omega_{m_{Avg}}$	Steady State
Speed Overshoot [%]	$100\,\frac{\Omega_{m_{Pk}}-\Omega_{m_{Avg}}}{\Omega_{m_{Avg}}}$	Dynamic
Acceleration Time [s]	$t_2\vert_{\frac{d\omega_m}{dt}=0}$-$t_1\vert_{\frac{d\omega_m}{dt}\gg 0}$	Dynamic

Table 2: Application Performance Specification limits

Application Performances	*MP1*		*MP2*		*MP3*	
	-	1.6	-	2	-	2.4
Torque Ripple[Nm]	-	0.01	-	8e-3	-	9e-3
Speed Ripple [rad/s]	-	0.07	-	2.5	-	5.25
Speed Error[rad/s]	0.81	-	0.86	-	0.77	-
Efficiency [%]	-	1.65	-	1.65	-	1.65
Speed Overshoot[%]	4.26	4.84	3.7	4.2	2.1	2.36
Acceleration Time[s]	-	1.6	-	2	-	2.4

The Mission Profile model [9] gives not only the levels of applied reference speed (the desired value that has to be matched by the mechanical speed response) but also the level of applied *Resistive Torque* (or load Torque) as a contribution between: environmental (resistive) torque and human (active) torque. The *Resistive Torque* matches, in Steady State, the *Electrical Torque* developed by the motor, as it is shown in Fig.5.

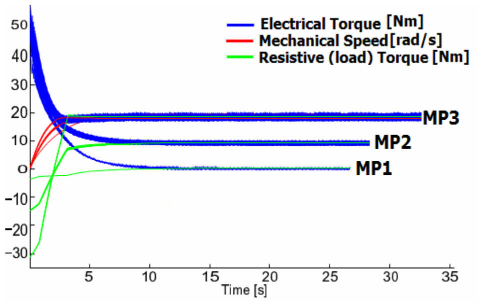

Figure 5: E-Bike main signals for 3 operating points: MP1, MP2, MP3

The metrics used for responses performances are given in Table 1, where: σ_{Te} and σ_{Ω_m} are the standard deviation of electrical Torque (T_e) and mechanical speed (Ω_m) respectively. $\Omega_{m_{Pk}}, \Omega_{m_{Ref}}, \Omega_{m_{Avg}}$ are the peak, the reference (commanded) and average mechanical speed respectively. The term P_{out} represents the output power consumed by the load and P_{in} is the input power delivered by the battery, the product of battery supply (V_{Batt}) and the current consumed by the motor branches ($I_{consumed}$). The term $\frac{d\omega_m}{dt}$ is the first derivative of the motor speed.

B. Component Parameter Variation

For counting and distribution fitting approaches, a Monte Carlo sampling with normal distribution is used for each factor. A Latin Hypercube Sampling (LHS) combined with Full Factorial design is used when the metamodeling solution is applied (Fig.6). Note that each component factor is normalized between ± 1 in order to have a common scale for all factors.

In the analysis phase we use three different datasets.

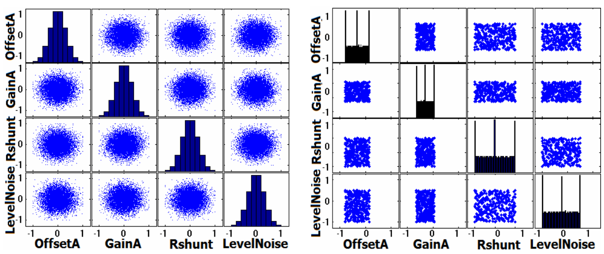

Figure 6: DoE for Distribution fitness (left) and for Metamodeling (right)

Then, based on the selected approach, we built a *training* dataset for both analyses - application yield for *FP estimation* and *Factor impact estimation*. For *FP estimation* problem, the *training* dataset contains only 500 observations per each MP and it is obtained by a random selection of data samples from the *validation* set. This *training* dataset is used to compute the FP only through counting and distribution fitting approaches. The *training* dataset used for the metamodeling approach contains only 350 observations and it is directly obtained from the proposed DoE from Fig.6 (right). In general the number of it is a function of DoE type.

A *validation* dataset contains 2650 observations per each MP and it serves for FP estimation through classical counting. This will be our ground truth for methods comparisons and it is denoted as $\boldsymbol{Count_{val}}$.

C. FP results

The comparison between the results obtained for the FP estimation by applying three different methods: simple counting, distribution fitting and metamodeling, are presented in Fig.7. The *y* axis from Fig.7 represents the FP values and the *x* axis the names of all 6 performances. In this figure only the non-zero values of FP are shown. When the *fitDist* solution is used, two types of parametric distributions are fit for the original data: Normal and Gamma. The number of fitted parametric distributions is 9 from the total number of 18 distributions (6 distributions x3MP), while the rest of 9 are nonparametric distributions.

The metamodel fitness is validated by a 10-fold cross validation with samples from the *validation* dataset.

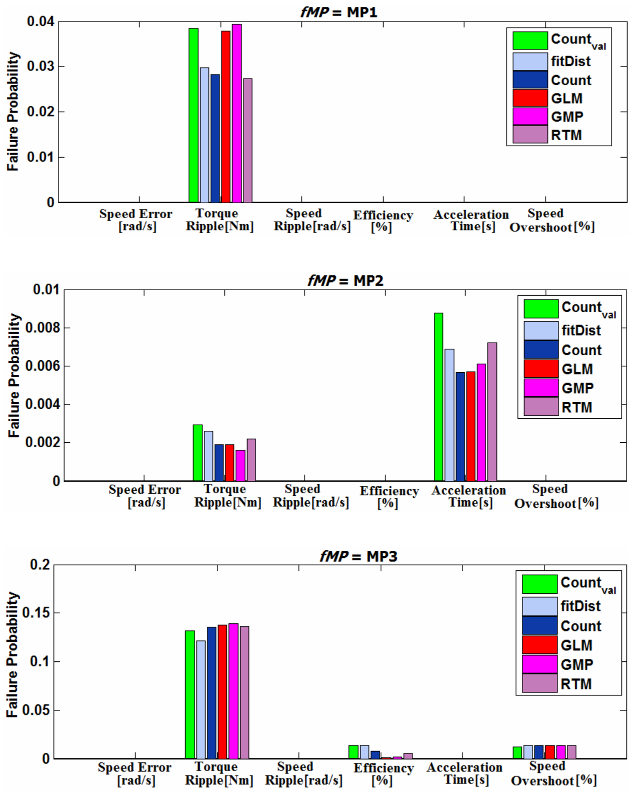

Figure 7: FP values for each performance, in each MP. Comparison between: Distribution fitting (fitDist), classical counting (Count) and metamodeling, compared with the ground truth value ($\boldsymbol{Count_{val}}$)

Table 3: *Goodness of Fit* in case of each metamodel

fMP	Performance	Metamodeling fitness		
		GLM	GMP	RTM
MP1	**Torque Ripple**	***Fit***	***Fit***	***NotFit***
MP2	Torque Ripple	***NotFit***	***NotFit***	***NotFit***
	Acceleration Time	***Fit***	***Fit***	***NotFit***
MP3	Torque Ripple	***NotFit***	***NotFit***	***NotFit***
	Efficiency	***NotFit***	***NotFit***	***NotFit***
	Speed Overshoot	***Fit***	***Fit***	***NotFit***

A 0.98 threshold of R_{sq} is considered for a fit metamodel. If the obtained metamodel has a R_{sq}> 0.98 then the metamodel is declared to be *Fit*, otherwise *NotFit*. However, acceptable results (good estimation of the FP compared with the ground truth values) are obtained also for a R_{sq} lower than 0.98 but still higher than 0.7. For FP estimation, the metamodel approach is evaluated for a number of $5x10^5$ samples. A number of 18 metamodels (*6* responses x3MP*)* are obtained using different types of metamodeling: GLM, GMP and RTM. From the all 18 metamodels, most of them have a $R_{sq} >$ 0.94 for GLM and GMP and $R_{sq} >$ 0.92 for RTM, and just four metamodels have an $R_{sq} < 0.5$.

We observe that, in case of RTM (Regression Tree Modeling) for response fitting, no fit is obtained with the *validation* dataset. Therefore, we cannot trust the further results obtained with this method.

From the FP results (Fig.7, 6) we conclude that the proposed approach based on metamodeling has an acceptable accuracy. It is comparable with the FP obtained through Count_{val} when the metamodels are *Fit*, while the FP is fairly different in case of a *NonFit* (such as *Efficiency* response in MP3). In case of *Acceleration Time* in MP2, the FP value obtained with GLM or GMP is relatively close to close to the truth value.

Regarding the suitability of the candidate current sensor in the E-bike application, this component does not fit in the application in *MP3* since the *Torque Ripple* has a large FP in this profile.

D. Factors impact

The second problem of revealing the component's factors which cause the application failure and estimation of their impact is solved by the metamodels. In order to overcome the risk of missing higher order interactions, interactive plots are used (Fig.7). In each window the dependency of the response on only one factor is shown. This dependency is conditioned by the values of the other factors, which is shown by the vertical dashed blue lines in the other window. One click on one window (new setting for the respective factor) will update the dependencies on the other factors, hence the graphics in the other windows accordingly. The green line of the plots represents the metamodel prediction, while the red dashed lines (sideways from the green line) represent the confidence interval. Fig.7 shows the impact of the significant factors in the system performances. In this figure only the *Fit* metamodels obtained with GLM are shown.

The *Torque Ripple* in *MP1* linearly increases with *LevelNoise* – the main factor of metamodel (I) from Fig.7. *Acceleration Time* in *MP2* increases when *GainA* and *Rshunt* - the main factors of metamodel (II) from Fig.7 are increasing, while *Speed Overshoot* in *MP3* linearly decreases when *GainA* and *Rshunt* - the main factors of metamodel (III) from Fig.7 are increasing.

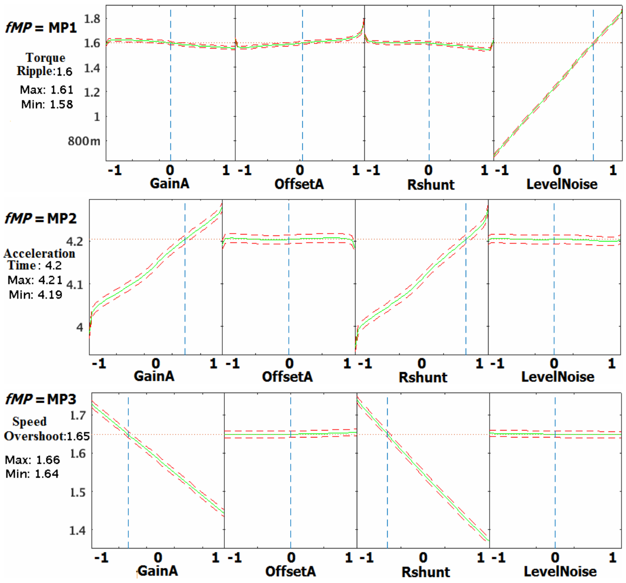

Figure 8: Metamodels: (I) Torque Ripple in MP1, (II) Acceleration Time in MP2 and (III) Speed Overshoot in MP3

Next, based on the fitted metamodels, three contour plots are drawn (Fig.8). The contour plot represents in 2-dimensional the transversal cut of the 3-dimensional surface at the height of the specification limit by plotting the contour line of the response as a function of two variables (the curve along which the response function has a constant value). The (x, y) coordinates correspond to the selected factors (in our case, the factors that have the greatest impact) and their ranges give the squared plane. Then, given the specification limit for the response, that lies on z coordinate, a red line is drawn and it connects the (x, y) coordinates where that response value occurs.

This value is written with black on the contour line: 1.6 represents the USL for *Torque Ripple in MP1,* 4.2 represents the USL for *Acceleration Time in MP2* and 1.65 represents the USL for *Speed Overshoot in MP3.* The red subarea indicates that there is a set of factor values that causes the failure (out of specification limits) for the response while the green subarea indicates the factors values that ensure a response inside of the specification limits. The purple dots represent the values of the response predicted through the metamodel. Practically, only the purple dots that exceed the red contour are counted as fails from the total number of dots.

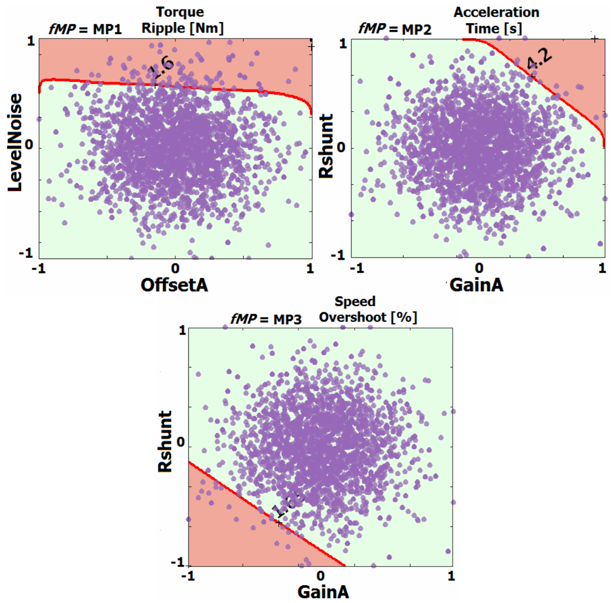

Figure 9: Feasible Region in the main parameter space: (I) Torque Ripple in MP1, (II) Acceleration Time in MP2 and (III) Speed Overshoot in MP3

From Fig.8 also the impact of the important factors on system responses can be deduced: for *Torque Ripple in MP1* we can see that the $LevelNoise$ factor has the main contribution on the response. The exceeding of *Torque Ripple* specification is caused by the large values of this factor. For *Acceleration Time in MP2* we can see an interaction effect between $GainA$ and $Rshunt$. When both factors have large values, the specification limit of the response is exceeded. The same factors impact the *Speed Overshoot in MP3*. Here, when both factors have small values, the specification limit of the response is exceeded.

For the E-Bike application the volume of the feasible region, in the parameter space, is given by the intersection of the green subarea of all six performances (from Fig.8). In this intersection region the system is not only functional but also all performances are inside of their specification limits.

The behavior of metamodel (I) from Fig.7 can be checked only by simulations of the non-linear model (the noise floor causes noisy currents which in the end impacts the torque response). From a Fast Fourier Transformation (FFT) of the Electrical Torque signal we can see, by plotting for ex. a single-side amplitude spectrum, that the magnitude of this signal's spectrum is higher when the Level of Noise is higher.

VII. DISCUSSIONS

The metamodeling approach of FP estimation is complex and requires several assumptions such as:

1. It is required to obtain a *Fit* metamodel that is validated with a *validation* dataset (different from the *training* dataset). This assumption is necessary but not sufficient.

2. If assumption 1 is satisfied, then, the accuracy of FP estimation based on metamodel depends on a few aspects:
- if the metamodel was fitted by using non-linear approximations (as in case of metamodel II - *Acceleration Time* in MP2) we observed that the accuracy of FP can decrease (its value is further to the validation FP value).
- if the metamodel was fitted with a R_{sq} closer to 1, then, we observed that the accuracy of FP is increased (becomes equal to the validation FP value).

However, as a solution to increase the FP accuracy, we recommend to take additional samples where the factors are set at those specific values that give the pass/fail contour (the red line from Fig.8 plots).

VIII. CONCLUSIONS

In this paper we have assessed the fitness of a component in an E-bike application by analyzing the application yield through the FP estimation with metamodels and by revealing the factors which lead to the application failure. Both addressed problems are analyzed for each system performance, in 3 different operating conditions (denoted here as mission profiles). Moreover, we presented a comparison between the results obtained for the FP by applying three different methods: counting, distribution fitting and metamodeling. Based on the obtained results and on the constraints of each method (complexity, resources and validation assumptions), we conclude that:

- the metamodeling is a reasonable method (acceptable accuracy for a small number of runs) for problem and a good method for revealing the component's factors impact on the failure – problem *Factor impact estimation*, only if a fit metamodel is obtained.
- the counting method is a good approach only for the problem (1): *FP estimation* , when the specification limits are far away from the obtained performance's spread.

REFERENCES

[1] X. Zhang, "A Multi-dimensional Component Importance Analysis Mechanism for Component Based Systems," 2014 IEEE 38th International Computer Software and Applications Conference Workshops, Vasteras, 2014, pp. 468-473. doi: 10.1109/COMPSACW.2014.79"

[2] Wenyuan Li, J. Korczynski; "A Reliability-Based Approach to Transmission Maintenance Planning and Its Application in BC Hydro System"

[3] E. Pavese, et.al; "Probabilistic Interface Automata";

[4] Z. J. Oster; "Reasoning with Qualitative Preferences to Develop Optimal Component-Based Systems"

[5] Au, Siu Kui Beck, James L.; "Estimation of small failure probabilities in high dimensions by subset simulation ";Journal: Probabilistic Engineering Mechanics, Field: Mechanics.

[6] Jing Li, Dongbin Xiu; "Evaluation of failure probability via surrogate models"; Journal of Computational Physics 229 (2010) 8966–8980

[7] http://www.mathworks.com/help/stats/gaussian-process-regression-models.html

[8] I. Kovacs, M. Topa, A. Buzo, G. Pelz, "Integrated Circuits' Characterization for Non-normal Data in Semiconductor Quality

Analysis", 2017 IEEE European Test Symposium (ETS), Limassol, Cyprus, pp. 1-2, 2017.

[9] A. Iosub, L.Goras, et al;"Distribution fitting Approach to Application Fitness Assessment"; [Submitted paper].

[10] Myers, R.H., D.C. Montgomery, et al; "Generalized linear models: with applications in engineering and the sciences", Wiley, 2012.

[11] D. Montgomery; "Design and analysis of experiments"'; John Wiley and Sons, 2005.