Proceedings Paper:

Jones, O.P.H., Oakley, J.E. orcid.org/0000-0002-9860-4093 and Purshouse, R.C. orcid.org/0000-0001-5880-1925 (2019) Toward a unified framework for model calibration and optimisation in virtual engineering workflows. In: 2019 IEEE International Conference on Systems, Man and Cybernetics (SMC). 2019 IEEE International Conference on Systems, Man and Cybernetics (SMC), 06-09 Oct 2019, Bari, Italy. IEEE , pp. 2148-2153. ISBN 9781728145709

https://doi.org/10.1109/smc.2019.8914600

Toward a unified framework for model calibration and optimisation in virtual engineering workflows

Oliver P. H. Jones[1], Jeremy E. Oakley[2] and Robin C. Purshouse[3]

***Abstract*— When designing a new product it is often advantageous to use virtual engineering as either a replacement or assistant to more traditional prototyping. Virtual engineering consists of two main stages: (i) development of the simulation model; (ii) use of the model in design optimisation. There is a vast literature on both of these stages in isolation but virtually no studies have considered them in combination. The model calibration and design optimisation processes both however, crucially, draw on the same resource budget for simulation evaluations. When evaluations are expensive, there may be advantages in treating the two stages as combined. This study lays out a joint framework by which such problems can be expressed through a unified mathematical notation. A previously published case study is reviewed within the context of this framework, and directions for further development are discussed.**

I. Introduction

Virtual engineering (VE) is the use of models to simulate the behaviour of complex engineering problems. When implemented as a part of a product life cycle the process is called the virtual engineering workflow (VEW) [1]. An overview of the VEW can be seen in Figure 1.

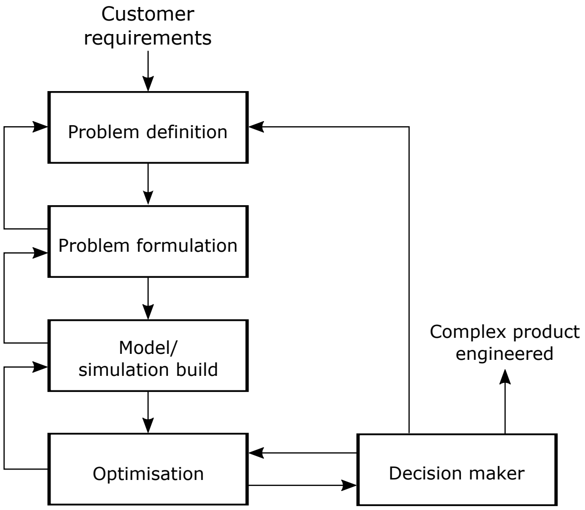

Fig. 1. Virtual engineering workflow

This work focuses on two of the main components within the VEW: model calibration and optimisation. Calibration and optimisation are traditionally separate activities. They both require multiple function evaluations from a simulation model which, for many real world problems, can be expensive either monetarily or in terms of computation time. It is sometimes the case that calibration is not completed before optimisation begins. This can lead to a restart of the optimizer thereby wasting part of the available budget of available function evaluations. We believe there will be benefits in considering these two processes together.

The optimization community has been mindful that the cost of evaluating a simulation model can be high, and have developed algorithms that aim to perform optimization on a limited budget. For a review of approaches see [2]. Typically, the total number of evaluations is limited to around 500 [3] – although both tighter and more relaxed constraints on the number of evaluations have been reported in the literature [2]. Note that these approaches typically assume that the model parameters are fully determined and do not consider any evaluations that may have been required to calibrate the parameters.

We first present a mathematical formulation of the joint problem of calibration and optimisation. A review of the real world problem on injection moulding [4] is then performed using the new joint problem formulation.

II. Mathematical formulation

To start the formulation the major elements that will be present within a problem are presented.

A. Problem framework and variables

We represent the physical system by

$$z_s = s(x, \psi), \tag{1}$$

with

- z_s: the outputs of the system;
- x: control inputs to be optimised, potentially subject to constraints;
- ψ: aleatory inputs, representing uncontrollable quantities that may differ randomly each time the the physical system is run.

Both the inputs and outputs of the system can be multi-dimensional. We also have a computer model of the system,

This work was supported by Jaguar Land Rover and the EPSRC grant EP/L025760/1 as part of the jointly funded *Programme for Simulation Innovation*. The first author acknowledges EPSRC studentship support.

[1] Oliver P. H. Jones is with the *Department of Automatic Control and Systems Engineering*, University of Sheffield, United Kingdom ophjones1@sheffield.ac.uk

[2] Jeremy E. Oakley is with the *School of Mathematics & Statistics*, University of Sheffield, United Kingdom j.oakley@sheffield.ac.uk

[2] Robin C. Purshouse with the *Department of Automatic Control and Systems Engineering*, University of Sheffield, United Kingdom r.purshouse@sheffield.ac.uk

represented by

$$z_m = f(x, \theta, \psi), \tag{2}$$

which may include additional calibration inputs θ required to 'tune' the model to the physical system. Uncertainty about θ would be epistemic.

The relationship between the model and the physical system is given by

$$s(x, \psi) = f(x, \theta, \psi) + \delta(x, \psi), \tag{3}$$

where $\delta(x, \psi)$ represents residual error in the model predictions, once the model has been tuned by selecting θ. The function $\delta()$ is often referred to as the model *discrepancy* [5].

1) Toy Formula - elements of the problem: A simple toy formulation is now presented in which a ball is thrown with the aim of hitting a target. In this scenario the objective is to minimize the distance between where the ball lands and the target. It is assumed that noisy physical measurements can be obtained for the distance the ball lands from the target, and that there a model of how far the ball lands from the target is also available. For illustration, we suppose that the true distance to the target (50m), the height at which the ball is thrown (2m), and the acceleration due to gravity (9.8m/s^2) are all unknown. The different components within the problem are as follows.

Model inputs:

- Control inputs: horizontal velocity (v_h) and vertical velocity (v_v)
- Calibrations inputs: distance to target (D_{ta}), starting height (H_i) and acceleration due to gravity (g)
- Aleatory input: constant horizontal acceleration exerted by wind (a)

We suppose that the physical system that described the distance from the target is

$$s(v_h, v_v, a) = |50 - D_{th}| \tag{4}$$

with

$$\begin{aligned} D_{th} &= (v_h t) + (0.5 a t^2) \\ t &= -\frac{v_v}{-9.8} + \sqrt{\left(\frac{v_v}{-9.8}\right)^2 - \left(\frac{4}{-9.8}\right)} \end{aligned} \tag{5}$$

The system output (to be minimised) is the distance from the target, and a constraint on the control input would be a maximum throwing speed.

We suppose that there is also an (imperfect) computer model of the system,

$$f(v_h, v_v, D_{ta}, H_i, g) = |D_{ta} - v_h t|, \tag{6}$$

with

$$t = -\frac{v_v}{g} + \sqrt{\left(\frac{v_v}{g}\right)^2 - \left(\frac{2H_i}{g}\right)} \tag{7}$$

so that there will be model discrepancy resulting from the model failing to account for the wind acceleration.

B. Model calibration

Before it is possible to consider the steps that need to be carried out to implement calibration it is first necessary to consider what information is available.

1) Available Information: The information available for the analyst will not necessarily be the same in all problems. There are two main sources from which new data could be obtained, either expert opinion or physical experiments. In this work two cases are considered.

- Case 1: Noisy observations. We obtain physical experimental data with observation error. For $i = 1, \ldots, n$, experiment i is carried out with control input setting x_i and measured aleatory inputs ψ_i, giving an observation

$$y_i = s(x_i, \psi_i) + \epsilon_i, \tag{8}$$

  where ϵ_i is an observation error.
- Case 2: Plausibility/acceptability check. An expert believes the system output would lie in some range, given the control and aleatory inputs. For $i = 1, \ldots, n$, the expert judges

$$s(x_i, \psi_i) \in S_i, \tag{9}$$

  for some set S_i.

In both cases it is assumed that the value of ψ_i is known. In the first case it is possible to work with an unknown ψ_i although it would increase the complexity of the problem.

2) Calibration: In model calibration, the aim is to reduce the error between the predicted and observed behaviour [5]. This is achieved though altering the internal parameters of the model so that it can as closely predict the real 'true' outputs as possible. The term true being used here refers to the values that would be present/obtained by running a physical experiment or having perfect knowledge of the system.

3) Determining parameters: Depending on what form of information is being supplied by the expert, the method needed for model calibration will vary.

- Case 1: Noisy observations
  The observations from the experiments are

$$Y = [y_1, \ldots, y_n]. \tag{10}$$

  If we assume a distribution for the errors $\epsilon_1, \ldots, \epsilon_n$ and for the unknown discrepancy $\delta()$, then from (3) we can construct a likelihood function $p(Y|\theta)$. We may then choose,

$$\underset{\theta}{\text{argmax}}\ p(Y|\theta), \tag{11}$$

  to obtain a single best point, or, having specified a prior distribution $p(\theta)$ derive the posterior distribution,

$$p(\theta|Y) \propto p(\theta) p(Y|\theta). \tag{12}$$

  Note that such calibration data are not just informative for θ: they would also be informative for the model discrepancy δ, so that, as in [5], we can derive a joint posterior distribution $p(\theta, \delta|Y)$. This has important implications for optimisation: via (3), we can attempt to optimise the physical system s, rather than the computer model approximation of it f.

- Case 2: Plausibility/acceptability check. We define D to be the set of judgements provided by the expert:

$$D = \{s(x_i, \psi_i) \in S_i,\, i = 1 \ldots, n\} \tag{13}$$

In this case there are two situations that need to be considered, depending on whether we assume modelling error (i.e. the model discrepancy δ).

  - No modelling error present
    In this case, the likelihood function is then

$$p(D|\theta) = \begin{cases} 1, & \text{if } f(x_i, \theta, \psi_i) \in S_i \quad \text{for } i = 1, ..., n \\ 0, & \text{otherwise.} \end{cases} \tag{14}$$

  - Modeling error present
    When using the relationship between the model and system as shown in (3) the uncertainty due to the model error becomes an issue. One method of overcoming this is to add a tolerance to each S_i based on a judgment about $\delta(x, \psi)$. We define S_i^* with $S_i \subset S_i^*$ and write the likelihood as

$$p(D|\theta) = \begin{cases} 1, & \text{if } f(x_i, \theta, \psi_i) \in S_i^* \quad \text{for } i = 1, ..., n \\ 0, & \text{otherwise.} \end{cases} \tag{15}$$

    As with the noisy observation case, we might then specify a prior for θ and derive the posterior, using the likelihood function for the data D. A maximum likelihood approach may be less suitable here, as multiple θ may produce a likelihood of 1.

C. Optimisation

We suppose that the aim is to optimise the physical system, and that model discrepancy is acknowledged, so that we must consider optimising $f(x, \theta, \psi) + \delta(x, \psi)$. Were we to ignore model discrepancy, the subsequent notation could be simplified by omitting the term $\delta(x, \psi)$ throughout.

Assuming vector output quantities, we have a standard (multi-objective) optimisation problem [6]. Writing

$$\begin{aligned} f(x, \theta, \psi) + \delta(x, \psi) =& (f_1(x, \theta, \psi) + \delta_1(x, \psi), \\ & \ldots, f_k(x, \theta, \psi) + \delta_k(x, \psi)), \end{aligned} \tag{16}$$

the optimisation problem can be written as

$$\underset{x \in X}{\text{minimize}} \; f(x, \theta, \psi) + \delta(x, \psi), \tag{17}$$

subject to any constraints on x, where X is the set of possible choices of control inputs x, and minimisation is elementwise.

1) Pareto optimality: Working with multi-objective problems a trade-off surface between the objectives, known as the Pareto front, can be obtained by getting the set of non-dominated points from within the current population of points. A point $x \in X$ is said to be non-dominated when there does not exist a point $x' \in X$ such that

$$f_i(x', \theta, \psi) + \delta_i(x', \psi) < f_i(x, \theta, \psi) + \delta_i(x, \psi), \tag{18}$$

for $i = 1, \ldots, k$.

2) Robust optimisation: Robust optimisation considers optimisation in the presence of uncertainty [7], and can provide a link between the calibration and optimisation stages described here. Starting with the optimisation problem in (17), if we now consider θ, ψ and δ as uncertain, the objective function is now uncertain for any x. To recover a deterministic optimisation function with a known objective function, we re-express the optimisation problem as

$$\underset{x \in X}{\text{minimise}} \; I(f(x, \theta, \psi) + \delta(x, \psi)), \tag{19}$$

For some appropriate functional I. Writing $\Omega = (\theta, \psi, \delta)$, with a corresponding sample space Ω_s, common choices for I are as follows.

- Worst-case scenario - determine the worst case that can be produced from within a bounded domain [8].

$$I_{wc}(x, \Omega) = \max_{\Omega \in \Omega_s} f(x, \theta, \psi) + \delta(x, \psi) \tag{20}$$

- Aggregated Value - a combination of possible values gained from the uncertain values determined by an integral measure of robustness [9]. This method uses the expectation, variance or a combination of the two as the indicator.

$$\begin{aligned} I_{exp}(x, \Omega) = E[f(x, \theta, \psi) + \delta(x, \psi)] \\ I_{var}(x, \Omega) = var[f(x, \theta, \psi) + \delta(x, \psi)] \end{aligned} \tag{21}$$

  where the bi-objective problem is,

$$\min_{x \in X} \, [I_{exp} \;,\; I_{var}] \tag{22}$$

- Threshold probability - determine how probable it is for the objective function to be better than a reference threshold [10]. The indicator determines the confidence level where q is the threshold.

$$\begin{aligned} I_{con}(f(x, \theta, \psi) + \delta(x, \psi)), q) =& p(f(x, \theta, \psi) \\ & + \delta(x, \psi) \leq q) \end{aligned} \tag{23}$$

Once one of these indicator methods has been chosen, the results of the indicator replace the random objective function which would have been used within the optimisation. Depending on the representation of the parameter, there are two methods that can be implemented [11],

- Probabilistic - a method which works with distributions.
- Possibilistic - works based on possible realizations of the parameter, often expressed as “scenarios”, either
  - a set of scenarios is used within the indicator (e.g. worst case across all scenarios), or
  - performance against the objectives under different scenarios are represented by additional objectives and/or constraints.

In a probabilistic approach, the calibration phase could supply the probability distribution for both θ and δ.

The use of robust methods can present additional issues for problems with expensive evaluations, since typically multiple evaluations for each choice of control inputs are needed in order to understand the variability in the outputs (e.g. via Monte Carlo methods). However methods are becoming available that attempt to estimate the variability without the necessity for repeated evaluations [12].

D. Toy Formula - The problem

Working with the same problem as before, if the ball is thrown at a particular velocity, a noisy measurement may be obtained as

$$y = s(v_h, v_v, a) + \epsilon. \tag{24}$$

In this case the observation error ϵ would be the difference between the measured distance the ball is from the target and the actual distance between the ball and target. The model output is expressed as,

$$z_m = f(v_h, v_v, D_{ta}, H_i, g). \tag{25}$$

If, for example, we wanted a point estimate of the calibration inputs $\theta = (D_{ta}, H_i, g)$, we would need to obtain

$$\hat{\theta} = \underset{\theta}{\operatorname{argmax}}\ p(y|\theta), \tag{26}$$

whilst also inferring the model discrepancy $\delta(v_h, v_v, a)$.

We could then use $\hat{\theta} = (\hat{D}_{ta}, \hat{H}_i, \hat{g})$ in an optimisation,

$$\underset{x \in X}{\text{minimize}}\ f(x, \hat{\theta}) + \delta(x, a), \tag{27}$$

for a given a, with $x = (v_h, v_v)$.

III. Possible solutions

There are three logical routes that solutions to this problem could take. The first would be to use a robust optimisation approach, without performing any calibration. This could save budget in terms of physical experiments, and potentially computer model runs, but may result in an overly-conservative result as a consequence of greater parameter uncertainty. A second option is to perform calibration before beginning the optimisation process. This may reduce the risk of a conservative solution, but at a greater cost in terms of the computation budget. A third option is to alternate between performing optimisation and model calibration based upon a chosen criteria. This may be beneficial if model discrepancy is a concern: as the non-dominated set of control inputs is reduced, further calibration experiments can be performed, which may improve estimates of model discrepancy precisely in the regions of interest within the control input space.

Figure 2 provides a sketch of how such a third option could be realised. The processes of calibration and optimization are alternated, with the transition between the two determined by switching conditions. These conditions could be based on number of evaluations (or proportion of the budget), or use convergence criteria that are relevant to either calibration or optimization (e.g. respectively, robustness of optimal output to parameter uncertainty or negligible improvement in a hypervolume indicator [13]).

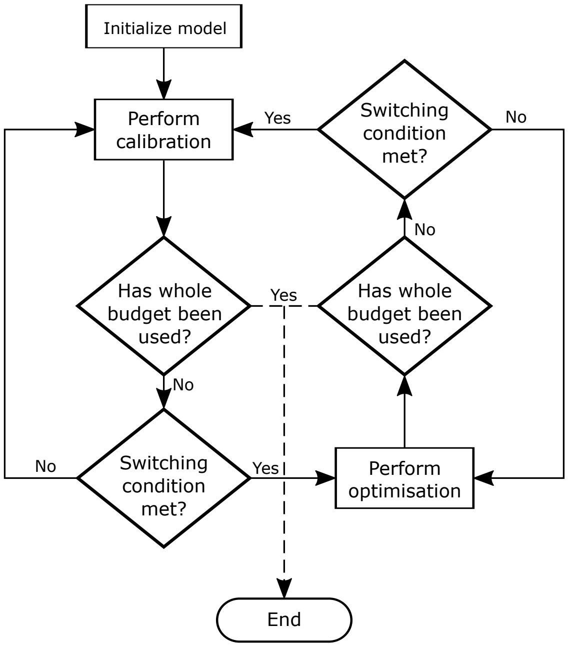

Fig. 2. Flow diagram for the proposed joint problem of model calibration and optimisation

A further area to consider is the use of low-complexity *surrogate models* (also known as *emulators* or *meta-models*) to replace expensive evaluations within either the calibration process, optimization process, or both processes [2], [5]. Any of the above approaches can incorporate surrogates. Surrogates have the further benefit that they can use information from both the calibration and optimization runs, allowing for greater information sharing between the two activities.

IV. Real world example, set within the new combined problem framework

This section sets out an example problem within the combined problem framework for optimisation and model calibration. The real world example presented here is work done by Villarreal-Marroquín and colleagues on injection moulding [4], which, to the best of our knowledge, is notable in being the only known example that has considered the two problems together.

A. Injection moulding

The physical system s has the following control inputs x: melting temperature (°C), packing time (seconds), packing pressure (MPa), Cooling time (seconds). The computer model f has the same control inputs, and three calibration inputs θ (referred to as heat transfer coefficients).

The outputs of both the physical system and the computer model are: relative shrinkage of the length, relative shrinkage of the thickness and relative shrinkage of the width. The objective is to choose control inputs that will minimise all three outputs. There are no aleatory inputs in this example.

Their methodology proceeds in a series of stages. We give a condensed list here, skipping those stages that are not directly relevant to our framework.

1) Stage 1 - design of experiment: The methodology starts by acquiring two sets of data. The first set consists of physical observed values at 19 different settings of the control inputs, each replicated four times. The second set comes from running the computer model at 35 settings of the control and calibration inputs.

2) Stage 2 - fit calibrated predictor: The computer model is computationally expensive, and so the authors do not use it 'directly' for either calibration or optimisation. They also recognise the presence of model discrepancy, which they account for in their optimisation. This is done through the construction of a 'calibrated predictor'. Using the two datasets (physical experiments and computer model runs), they

- construct a meta-model of the computer model, based on Gaussian process regression;
- construct an estimate $\hat{\delta}(x)$ of the model discrepancy;
- derive a posterior distribution $p(\theta|Y)$ of the calibration inputs θ given the two datasets.

The calibrated predictor takes the form

$$\hat{s}(x) = \mathbb{E}_{\theta}[\hat{f}(x,\theta)] + \hat{\delta}(x), \tag{28}$$

where $\hat{f}$ is an estimate of the computer model obtained from the meta-model, and the expectation is taken with respect to the posterior distribution of the calibration inputs θ.

3) Stage 3 - first attempt to optimise the physical system: The authors perform optimisation by constructing a grid of equally spaced points spanning each of the control inputs. Once the grid is constructed, the next step is to evaluate (28) at each grid point (producing three outputs per grid point). In their case study, the authors used a grid of 360 points.

4) Stage 4 - obtain Pareto front: A Pareto front is obtained by inspection of the grid. Given the averaging with respect to θ, this corresponds to the robust optimisation procedure with the functional I chosen to be the expectation operator.

5) Stage 5 - refine Pareto front: Having identified (approximately) the Pareto front, a second grid of control inputs is chosen in the location of the front, and (28) is again evaluated at each grid point. A refined Pareto front is then identified by inspection. The authors used 560 grid points in their case study.

6) Stage 6 - validate final front: The estimated Pareto front has been obtained from a 'calibrated predictor', evaluated at 920 settings of the control inputs. The calibrated predictor is only an estimate of the physical system, based on 19 (replicated) physical experiments, and 35 computer model runs. Consequently, there is a need to validate the estimated Pareto front. This is done by selecting five control input settings from the front, and then performing physical experiments at these settings. The system outputs are then compared with those from the initial 19 physical experiments.

B. Discussion

The case-study has illustrated the combined process of calibration and optimisation, but there are clearly further issues to consider, suggesting several directions for further research, specifically with regard to how available budgets for both physical experiments and computer model runs should be used.

Physical experiments can be used to calibrate the computer model, and learn about the model discrepancy. How should such experiments be designed? Ideally, we would like to run experiments at or close to optimal settings of the control inputs, so that the model discrepancy can be best understood in the most relevant part of the control input space. However, to identify this region of input space, we first need to optimise the computer model, suggesting the need for an iterated/sequential procedure. A further consideration is that the budget for physical experiments may be severely limited.

Computer model runs are needed both for calibration and optimisation, and the strategy used in [4] was to construct a meta-model, to be used in both, based on a single set of model runs. We may not need the meta-model to have the same level of precision over the entire control input space, if particular regions of input space are dominated. Sequential approaches for model runs have been suggested for optimisation [3], but we are not aware of the extra requirement of model calibration being simultaneously addressed.

Currently, laying out a problem within the joint problem framework has the possibility of providing benefits such as helping with identifying the overall structure of the workflow. A unified framework will also make it easier to track and asses the impact of both epistemic uncertainty as well as control inputs which may have otherwise been lost between stages. When choosing how to solving a problem, knowledge of what components need to be considered at each stage can aid in the selection of appropriate methods.

V. Conclusion

This paper has laid out a joint problem framework for the separate steps of optimisation and model calibration. A real world cases study has been presented showing the feasibility and advantages of setting such problems within the new framework, and directions for further research have been suggested.

The utility of the framework for further real world VEW problems should be assessed. The framework then provides a coherent platform for the development of combined calibration-optimisation activities.

References

[1] H.G. Lemu. "Virtual engineering in design and manufacturing." Advances in Manufacturing, vol. 2, no. 4, pp. 289-294, December, 2014.

[2] T. Chugh, K. Sindhya, J. Hakanen and K. Miettinen, "A survey on handling computationally expensive multiobjective optimization problems with evolutionary algorithms." Soft Computing, vol. 23, no. 9, pp. 3137-3166, July, 2019.

[3] J. Knowles, "ParEGO: a hybrid algorithm with on-line landscape approximation for expensive multiobjective optimization problems." IEEE Transactions on Evolutionary Computation, vol. 10, no. 1, pp. 50-66, February, 2006.

[4] M.G. Villarreal-Marroquín, P.H. Chen, R. Mulyana, T.J. Santner, A.M. Dean and J.M. Castro, ”Multiobjective optimization of injection molding using a calibrated predictor based on physical and simulated data.” Polymer Engineering & Science, vol. 57, no. 3, pp. 248-257, March, 2017.

[5] M.C. Kennedy and A. O'Hagan, ”Bayesian calibration of computer models.” Journal of the Royal Statistical Society: Series B (Statistical Methodology), vol. 63, no. 3, pp. 425-464, 2001.

[6] D. Kalyanmoy, ”Multi-Objective Optimization Using Evolutionary Algorithms”. John Wiley & Sons, Inc., New York, NY, USA. 2001.

[7] H.G. Beyer and B. Sendhoff, ”Robust optimization–a comprehensive survey.” Computer methods in applied mechanics and engineering, vol. 196, no. 33-34, pp. 3190-3218, July, 2007.

[8] M. Ehrgott, J. Ide and A. Schöbel, ”Minmax robustness for multi-objective optimization problems.” European Journal of Operational Research, vol. 239, no. 1, pp. 17-31, November, 2014.

[9] Z.P. Mourelatos and J. Liang, ”A methodology for trading-off performance and robustness under uncertainty.” Journal of mechanical design, vol. 128, no. 4, pp. 856-863, July, 2006.

[10] F. Rambeaux, F. Hamelin and D. Sauter, ”Optimal thresholding for robust fault detection of uncertain systems.” International Journal of Robust and Nonlinear Control: IFAC-Affiliated Journal, vol. 10, no. 14, pp. 1155-1173, December, 2000.

[11] R. S. Langley, ”Unified approach to probabilistic and possibilistic analysis of uncertain systems.” Journal of engineering mechanics, vol. 126, no. 11, pp. 1163-1172, November, 2000.

[12] J.A. Duro, R.C. Purshouse, S. Salomon, D.C. Oara, V. Kadirkamanathan and P.J. Fleming, ”sParEGO–A Hybrid Optimization Algorithm for Expensive Uncertain Multi-objective Optimization Problems.” In International Conference on Evolutionary Multi-Criterion Optimization. Springer, Cham, pp. 424-438, March, 2019.

[13] E. Zitzler, D. Brockhoff and L. Thiele, ”The hypervolume indicator revisited: On the design of Pareto-compliant indicators via weighted integration.” In International Conference on Evolutionary Multi-Criterion Optimization. Springer, Berlin, Heidelberg, pp. 862-876, March, 2007.