

Min-max hyperparameter tuning, with application to fault detection

Julien Marzat ^{*,**} H el ene Piet-Lahanier ^{*} Eric Walter ^{**}

^{*} ONERA DCPS, Chemin de la Huni ere, Palaiseau, France,
julien.marzat@onera.fr, helene.piet-lahanier@onera.fr

^{**} Laboratoire des Signaux et Syst emes (L2S),
CNRS-SUPELEC-Univ Paris-Sud, France, eric.walter@lss.supelec.fr

Abstract: In order to reach satisfactory performance, fault diagnosis methods require the tuning of internal parameters, usually called *hyperparameters*. This is generally achieved by optimizing a performance criterion, typically a trade-off between false-alarm and non-detection rates. Perturbations should also be taken into account, for instance by considering the worst possible case. A new method to achieve such a tuning is described, which is especially interesting when the simulations required are so costly that their number is severely limited. It achieves min-max optimization of the tuning parameters via a relaxation procedure and Kriging-based optimization. This approach is applied to the worst-case optimal tuning of a fault diagnosis method consisting of an observer-based residual generator followed by a statistical test. It readily extends to the tuning of hyperparameters in other contexts.

Keywords: efficient global optimization, expected improvement, fault detection and isolation, Gaussian processes, hyperparameter tuning, Kriging, min-max, worst-case optimization.

1. INTRODUCTION

So many methods have been developed to address fault detection and isolation that users may be at a loss to select the most suitable one for a given system. A fair comparison requires an adequate tuning of the internal parameters, often called *hyperparameters*, of each of the candidate methods. This step is a major issue, as it has a strong impact on performance and robustness. Tuning can be tackled by optimizing a suitable performance criterion on a representative benchmark.

Very recently, Falcoz et al. (2009) used an evolutionary algorithm to tune the covariance matrices of Kalman filters for fault detection. In (Marzat et al. (2010)), we introduced an alternative approach for the tuning of hyperparameters in fault diagnosis relying on tools developed in the context of computer experiments (Santner et al. (2003)). This approach is especially relevant when the simulations required are so costly that their number is severely limited. It was applied to the tuning of hyperparameters of several change-detection methods so as to minimize some trade-off between false-alarm and non-detection rates. Such a performance index is seen as the output of a black-box computer simulation whose inputs are the hyperparameters to be tuned. Kriging (also known as regression via Gaussian processes) is used along with recursive Bayesian optimization based on the concept of Expected Improvement (Jones (2001)) to facilitate optimal tuning by reducing its computational cost. This methodology is not limited to fault diagnosis and could be applied to many types of systems.

An important concern, left aside in our previous study, is robustness to the effect of environmental variables, i.e.,

disturbances, modeling errors or measurement uncertainty. The tuning of a fault detection method should remain valid for a reasonable set of possible values of the environmental variables. In the present paper, the previous procedure is extended to take into account such environmental disturbances. Most of the studies on computer experiments in this context use a probabilistic modeling of the environmental variables (Lehman et al. (2004)). Following a different path, we assume that bounds of the environmental space are available and look for an optimal tuning in the worst-case sense. Worst-case optimality is a concern that has been raised in many studies on fault detection. In particular, pioneering work by Chow and Willsky (1984) suggests a min-max choice of parameters for parity space methods with respect to modeling uncertainty. In the same vein, adaptive thresholds, proposed by Frank and Ding (1997) (see also Zhang and Qin (2009)), or H_∞ methods (Falcoz et al. (2010)) focus on minimizing the impact of disturbances on the residuals.

This paper is organized as follows. First the tuning approach of Marzat et al. (2010) is briefly recalled in Section 2, and a new method combining a min-max optimization procedure by relaxation (Shimizu and Aiyoshi (1980)) and Kriging-based optimization is introduced in Section 3 to deal with environmental variables. This method is then applied in Section 4 to the tuning of a fault detection strategy comprising an observer-based residual generator coupled with a statistical test. The environmental variables considered are the noise level and the size of the fault. Numerical results demonstrate the interest and practicality of the methodology. Conclusions and perspectives are in Section 5.

2. TUNING HYPERPARAMETERS WITH EGO

In this section, which summarizes the results of Marzat et al. (2010), the environmental conditions are considered as fixed.

2.1 Problem formulation

The vector of the hyperparameters of a candidate fault detection method is $\mathbf{x}_c \in \mathbb{X}_c$, where \mathbb{X}_c is a known compact set. It is assumed that a continuous scalar cost function $y(\mathbf{x}_c)$, reflecting the performance level, can be evaluated through a computer simulation of the system considered. Tuning can then be formalized as the search for

$$\hat{\mathbf{x}}_c = \arg \min_{\mathbf{x}_c \in \mathbb{X}_c} y(\mathbf{x}_c) \quad (1)$$

As $y(\cdot)$ can only be evaluated at sampled points, we use a black-box global optimization method that has become very popular in the context of computer experiments. The overall procedure is recursive and presupposes that we have already computed a set of training data $\mathbf{y}_{c,n_c} = [y(\mathbf{x}_{c,1}), \dots, y(\mathbf{x}_{c,n_c})]^T$, corresponding to an initial sampling of n_c points in \mathbb{X}_c , $\mathcal{X}_{c,n_c} = [\mathbf{x}_{c,1}, \dots, \mathbf{x}_{c,n_c}]$. The main ingredients of this method, namely Kriging and Efficient Global Optimization are now recalled.

2.2 Kriging

In Kriging (Matheron (1963)), the black-box function $y(\cdot)$ is modeled as a Gaussian process, written as

$$Y(\mathbf{x}_c) = \mathbf{f}^T(\mathbf{x}_c) \mathbf{b} + Z(\mathbf{x}_c) \quad (2)$$

where $\mathbf{f}(\mathbf{x}_c)$ is some known regression function vector (usually chosen constant or polynomial in \mathbf{x}_c), \mathbf{b} is a vector of unknown regression coefficients to be estimated, and $Z(\cdot)$ is a zero-mean Gaussian process with known (or parametrized) covariance function $k(\cdot, \cdot)$. Kriging is then the search for the *best linear unbiased predictor* (BLUP) of $Y(\cdot)$ (Kleijnen (2009)).

The actual covariance $k(\cdot, \cdot)$ is usually unknown. It is expressed as

$$k(Z(\mathbf{x}_{c,i}), Z(\mathbf{x}_{c,j})) = \sigma_Z^2 R(\mathbf{x}_{c,i}, \mathbf{x}_{c,j}) \quad (3)$$

where σ_Z^2 is the process variance and $R(\cdot, \cdot)$ is a parametric correlation function. Both σ_Z^2 and the parameters of $R(\cdot, \cdot)$ must be chosen or estimated from the available data. Under a stationarity assumption, $R(\mathbf{x}_{c,i}, \mathbf{x}_{c,j})$ depends only on the displacement vector $\mathbf{x}_{c,i} - \mathbf{x}_{c,j}$, denoted by \mathbf{h} in what follows. A frequent choice of correlation function, also adopted in the present paper, is the *power exponential correlation function*

$$R(\mathbf{h}) = \exp \left(- \sum_{k=1}^d \left| \frac{h_k}{\theta_k} \right|^{p_k} \right) \quad (4)$$

where $0 < p_k \leq 2$, and h_k is the k -th component of \mathbf{h} . Note that $R(\mathbf{h})$ tends to 1 when \mathbf{h} tends to $\mathbf{0}$, and to 0 when \mathbf{h} tends to infinity. The covariance parameters θ_k may be estimated from the data by maximum likelihood, to get what is known as *empirical Kriging*, which we used for our application. A wide range of other choices for the correlation function is available (Santner et al. (2003)).

Define \mathbf{R} as the $n \times n$ matrix such that

$$\mathbf{R}(i, j) = R(\mathbf{x}_{c,i}, \mathbf{x}_{c,j}) \quad (5)$$

$\mathbf{r}(\mathbf{x}_c)$ as the n vector

$$\mathbf{r}(\mathbf{x}_c) = [R(\mathbf{x}_c, \mathbf{x}_{c,1}), \dots, R(\mathbf{x}_c, \mathbf{x}_{c,n})]^T \quad (6)$$

and \mathbf{F} as the $n \times \dim \mathbf{b}$ matrix

$$\mathbf{F} = [\mathbf{f}(\mathbf{x}_{c,1}), \dots, \mathbf{f}(\mathbf{x}_{c,n})]^T \quad (7)$$

The maximum-likelihood estimate $\hat{\mathbf{b}}$ of the vector of the regression coefficients \mathbf{b} from the available data $\{\mathcal{X}_{c,n_c}; \mathbf{y}_{c,n_c}\}$ is

$$\hat{\mathbf{b}} = (\mathbf{F}^T \mathbf{R}^{-1} \mathbf{F})^{-1} \mathbf{F}^T \mathbf{R}^{-1} \mathbf{y}_{c,n_c} \quad (8)$$

The prediction of the mean of the Gaussian process at $\mathbf{x}_c \in \mathbb{X}_c$ is then

$$\hat{Y}(\mathbf{x}_c) = \mathbf{f}^T(\mathbf{x}_c) \hat{\mathbf{b}} + \mathbf{r}(\mathbf{x}_c)^T \mathbf{R}^{-1} (\mathbf{y}_{c,n_c} - \mathbf{F} \hat{\mathbf{b}}) \quad (9)$$

This prediction is linear in \mathbf{y}_{c,n_c} and interpolates the training data, as $\hat{Y}(\mathbf{x}_{c,i}) = y(\mathbf{x}_{c,i})$. Another interesting property of Kriging, which is crucial regarding global search, is the possibility to compute the *variance of the prediction error* (Schonlau (1997)) at $\mathbf{x}_c \in \mathbb{X}_c$, when the parameters of the covariance are known, by

$$\hat{\sigma}^2(\mathbf{x}_c) = \sigma_Z^2 \left(\mathbf{1} - \mathbf{r}(\mathbf{x}_c)^T \mathbf{R}^{-1} \mathbf{r}(\mathbf{x}_c) \right) \quad (10)$$

2.3 Efficient Global Optimization (EGO)

The idea of EGO (Jones et al. (1998)) is to use the Kriging predictor \hat{Y} to find the $(n_c + 1)$ -st point at which the computer simulation will be run. This point is chosen optimally according to a criterion $J(\cdot)$ that measures the interest of an additional evaluation at \mathbf{x}_c , given the past results \mathbf{y}_{c,n_c} obtained at \mathcal{X}_{c,n_c} and the Kriging prediction of the mean $\hat{Y}(\mathbf{x}_c)$ and variance $\hat{\sigma}^2(\mathbf{x}_c)$,

$$\mathbf{x}_{c,n_c+1} = \arg \max_{\mathbf{x}_c \in \mathbb{X}_c} J \left(\mathbf{x}_c, \mathcal{X}_{c,n_c}, \mathbf{y}_{c,n_c}, \hat{Y}(\mathbf{x}_c), \hat{\sigma}^2(\mathbf{x}_c) \right) \quad (11)$$

A common choice for $J(\cdot)$ is EI, for *Expected Improvement* (Jones (2001)). The best available estimate of the minimum of $y(\cdot)$ after the first n_c evaluations is

$$y_{\min}^{n_c} = \min_{i=1 \dots n_c} \{y_i = y(\mathbf{x}_{c,i})\}$$

With

$$u = \left(y_{\min}^{n_c} - \hat{Y}(\mathbf{x}_c) \right) / \hat{\sigma}(\mathbf{x}_c) \quad (12)$$

the EI is expressed in closed-form as

$$\text{EI}(\mathbf{x}_c) = \hat{\sigma}(\mathbf{x}_c) [u \Phi(u) + \phi(u)] \quad (13)$$

where Φ is the cumulative distribution function and ϕ the probability density function of the normalized Gaussian distribution $\mathcal{N}(0, 1)$. Maximizing EI achieves a trade-off between local search (numerator of u) and the exploration of unknown areas (where $\hat{\sigma}$ is high), and is thus well suited for global optimization.

Algorithm 1 summarizes the procedure. A preliminary sampling, by Latin Hypercube Sampling (LHS) for example, is required to obtain the n_c points of the initial design \mathcal{X}_{c,n_c} . New points \mathbf{x}_c maximizing EI are recursively explored, until the value of EI becomes lower than some threshold $\varepsilon_{\text{EI}}^c$ or the maximal number of iterations allowed $n_{c,\text{max}}$ is reached. The estimate of the minimizer is then the argument of the empirical minimum on the points explored.

Algorithm 1. Efficient Global Optimization (EGO)

- 1: Choose $\mathcal{X}_{c,n_c} = \{\mathbf{x}_{c,1}, \dots, \mathbf{x}_{c,n_c}\}$ by LHS in \mathbb{X}_c
- 2: Compute $\mathbf{y}_{c,n_c} = [y(\mathbf{x}_{c,1}), \dots, y(\mathbf{x}_{c,n_c})]^T$
- 3: **while** $\max_{\mathbf{x}_c \in \mathbb{X}_c} \{\text{EI}(\mathbf{x}_c)\} > \varepsilon_{\text{EI}}^c$ **and** $n_c < n_{c,\text{max}}$ **do**
- 4: Fit the Kriging model on the known data points $\{\mathcal{X}_{c,n_c}; \mathbf{y}_{c,n_c}\}$ as described in Section 2.2
- 5: Find $y_{\min}^{n_c} = \min_{i=1 \dots n_c} \{y(\mathbf{x}_{c,i})\}$
- 6: Find $\mathbf{x}_{c,n_c+1} = \arg \max_{\mathbf{x}_c \in \mathbb{X}_c} \{\text{EI}(\mathbf{x}_c)\}$
- 7: Compute $y(\mathbf{x}_{c,n_c+1})$, append it to \mathbf{y}_{c,n_c} and append \mathbf{x}_{c,n_c+1} to \mathcal{X}_{c,n_c}
- 8: $n_c \leftarrow n_c + 1$
- 9: **end while**

3. WORST-CASE TUNING WITH ENVIRONMENTAL DISTURBANCES

In this section, we consider the optimal tuning of the vector of hyperparameters $\mathbf{x}_c \in \mathbb{X}_c$ when the vector of the environmental variables \mathbf{x}_e is not fixed, and only assumed to belong to a known compact set \mathbb{X}_e .

3.1 Formulation of the problem

The objective is now to find $\hat{\mathbf{x}}_c$ and $\hat{\mathbf{x}}_e$ such that

$$\{\hat{\mathbf{x}}_c, \hat{\mathbf{x}}_e\} = \arg \min_{\mathbf{x}_c \in \mathbb{X}_c} \max_{\mathbf{x}_e \in \mathbb{X}_e} y(\mathbf{x}_c, \mathbf{x}_e) \quad (14)$$

We are thus searching for the best hyperparameter tuning of the considered fault diagnosis method for the worst values of the environmental variables.

Global optimization of such a min-max criterion, where \mathbb{X}_c and \mathbb{X}_e are compact sets of continuous values, is not straightforward (for a survey, see Rustem and Howe (2002)). A simple idea would be to find the minimizer $\hat{\mathbf{x}}_c$ on \mathbb{X}_c for a fixed value $\mathbf{x}_e \in \mathbb{X}_e$, then to maximize on \mathbb{X}_e for this fixed value $\hat{\mathbf{x}}_c$, and to alternate these steps. However, the convergence of this algorithm, known as *Best Replay* (see Rustem (1998)), is not guaranteed and it turns out very often to cycle through useless values of candidate solutions. To overcome these drawbacks, we instead use iterative relaxation as described by Shimizu and Aiyoshi (1980).

3.2 Min-max optimization

Shimizu and Aiyoshi transform the initial minimax problem (14) by introducing an intermediate scalar τ to be minimized,

$$\begin{cases} \min_{\mathbf{x}_c \in \mathbb{X}_c, \tau} \tau \\ \text{subject to } y(\mathbf{x}_c, \mathbf{x}_e) \leq \tau, \forall \mathbf{x}_e \in \mathbb{X}_e \end{cases} \quad (15)$$

This is an equivalent minimization problem with respect to an infinite number of constraints, which is still intractable. It is then solved approximately by following an iterative relaxation procedure, where the constraints in (15) are replaced by

$$y(\mathbf{x}_c, \mathbf{x}_e) \leq \tau, \forall \mathbf{x}_e \in \mathcal{R}_e \quad (16)$$

with \mathcal{R}_e a *finite* set that contains the values of \mathbf{x}_e already explored.

Algorithm 2 presents the resulting procedure. Note that if this procedure is stopped before the ϵ threshold is reached, an approximate solution is still obtained, corresponding to a higher threshold ϵ' . This algorithm has been proven to terminate after a finite number of iterations if the following reasonable assumptions hold:

- \mathbb{X}_c and \mathbb{X}_e are nonempty and compact.
- $y(\cdot, \cdot)$ is continuous in \mathbf{x}_e , differentiable with respect to \mathbf{x}_c and with first partial derivatives continuous in \mathbf{x}_c .

Algorithm 2. Min-max procedure

- 1: Choose randomly an initial point $\mathbf{x}_e^{(1)} \in \mathbb{X}_e$ and set $\mathcal{R}_e = \{\mathbf{x}_e^{(1)}\}$ and $i = 1$.

- 2: Solve the current relaxed problem

$$\mathbf{x}_c^{(i)} = \arg \min_{\mathbf{x}_c \in \mathbb{X}_c} \left\{ \max_{\mathbf{x}_e \in \mathcal{R}_e} y(\mathbf{x}_c, \mathbf{x}_e) \right\}$$

- 3: Solve the maximization problem

$$\mathbf{x}_e^{(i+1)} = \arg \max_{\mathbf{x}_e \in \mathbb{X}_e} y(\mathbf{x}_c^{(i)}, \mathbf{x}_e)$$

- 4: If

$$y(\mathbf{x}_c^{(i)}, \mathbf{x}_e^{(i+1)}) - \max_{\mathbf{x}_e \in \mathcal{R}_e} y(\mathbf{x}_c^{(i)}, \mathbf{x}_e) < \epsilon$$

then return $\{\mathbf{x}_c^{(i)}, \mathbf{x}_e^{(i+1)}\}$ as an approximate solution to the initial min-max problem.

Else, append $\mathbf{x}_e^{(i+1)}$ to \mathcal{R}_e and go to Step 2 with $i \leftarrow i + 1$.

This algorithm is generic, and leaves the choice of optimization procedures to solve Steps 2 and 3 to the user. It can thus be easily combined with EGO (Algorithm 1) to address the min-max optimization of expensive-to-evaluate black-box functions through computer experiments.

3.3 Min-max Kriging-based optimization

Steps 2 and 3 of Algorithm 2 are performed using two independent EGO optimizations. This implies that two samplings will be required, one on \mathbb{X}_c and one on \mathbb{X}_e . An alternative would be to fit a global Kriging model on $\mathbb{X}_c \times \mathbb{X}_e$ and then to apply Algorithm 2 to find the min-max solution and continue the procedure iteratively. This strategy is also investigated but beyond the scope of this paper.

The objective of Step 2 of Algorithm 2 is to find a minimizer of the function $y_{\text{relax}}(\mathbf{x}_c, \mathcal{R}_e) = \max_{\mathbf{x}_e \in \mathcal{R}_e} \{y(\mathbf{x}_c, \mathbf{x}_e)\}$, where \mathcal{R}_e contains a finite number of values of \mathbf{x}_e obtained from previous iterations of the procedure. To find the minimizer $\mathbf{x}_c^{(i)}$, the following steps are carried out:

- Choice of a design $\mathcal{X}_{c,n_c} = \{\mathbf{x}_{c,1}, \dots, \mathbf{x}_{c,n_c}\}$ of n_c points in \mathbb{X}_c by LHS.
- Computation of $\mathbf{y}_{c,n_c} = [y_{\text{relax}}(\mathbf{x}_{c,1}), \dots, y_{\text{relax}}(\mathbf{x}_{c,n_c})]^T$. This is done by finding, for each point of the design \mathcal{X}_{c,n_c} , the maximum of $y(\cdot, \cdot)$ for each element of \mathcal{R}_e , which leads to the evaluation of this function ($n_c \times \dim \mathcal{R}_e$) times at each iteration. To reduce computational cost and evaluate it only at n_c new points, the same design \mathcal{X}_{c,n_c} is used at each iteration of the global min-max procedure.

- The *while* loop then proceeds as described in Algorithm 1 until the approximate solution is deemed satisfactory or the sampling budget is exhausted. The maximum value of y on \mathcal{R}_e for the newly sampled \mathbf{x}_c is to be computed at Step 7 of EGO.

Step 3 maximizes $y(\cdot, \cdot)$ with respect to \mathbf{x}_e , for the fixed value $\mathbf{x}_c = \mathbf{x}_c^{(i)}$ by looking for

$$\mathbf{x}_e^{(i+1)} = \arg \min_{\mathbf{x}_e \in \mathbb{X}_e} \{-y(\mathbf{x}_c^{(i)}, \mathbf{x}_e)\} \quad (17)$$

with EGO. As in the relaxation step, the same initial LHS sampling \mathcal{X}_{e,n_e} on \mathbb{X}_e can be used for all the iterations of the min-max optimization procedure. A widely used rule of thumb for the initial samplings \mathcal{X}_{c,n_c} and \mathcal{X}_{e,n_e} is to draw ten points per dimension of \mathbb{X}_c and \mathbb{X}_e (Jones et al. (1998)).

The complete min-max optimization strategy involves five parameters to be chosen, namely

- the tolerance ϵ on the stopping condition of Step 4 of Algorithm 2;
- the maximal numbers of iterations allowed $n_{c,\max}$ and $n_{e,\max}$ for each of the EGO algorithms;
- the tolerances ϵ_{EI}^c and ϵ_{EI}^e on the values of EI for each of the EGO algorithms.

4. APPLICATION TO FAULT DIAGNOSIS

As a simple illustrative example, we consider the tuning of a fault detection scheme composed of a residual generator (via an observer) and a CUSUM test (Basseville and Nikiforov (1993)). The test case is the reduced longitudinal model of a missile flying at a constant altitude of 6000 m. The state vector, consisting of the angle of attack, angular rate and Mach number, is $\mathbf{x} = [\alpha, q, M]^T$. The control input is the rudder angle, $u = \delta$, and the available measurement is the normal acceleration, $\gamma = a_z$. The linearized model around the operating point $\mathbf{x}_0 = [\bar{\alpha}, \bar{q}, \bar{M}]^T = [20 \text{ deg}, 18.4 \text{ deg/s}, 3]^T$ is given by the following state-space model, after discretization with a time step of 0.02s,

$$\begin{cases} \mathbf{x}_{k+1} = \mathbf{A}\mathbf{x}_k + \mathbf{B}u_k \\ \gamma_k = \mathbf{C}\mathbf{x}_k + \mathbf{D}u_k + w_k + f_k \end{cases} \quad (18)$$

where

$$\mathbf{A} = \begin{bmatrix} 0.9163 & 0.0194 & 0.0026 \\ -5.8014 & 0.9412 & 0.5991 \\ -0.0485 & -0.005 & 0.996 \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} -0.0279 \\ -2.5585 \\ -0.0019 \end{bmatrix}$$

$$\mathbf{C} = [-2.54 \ 0 \ -0.26], \quad \mathbf{D} = -0.204 \quad (19)$$

This model is simulated on a time horizon of 50 seconds. A sensor fault f on the measurement of a_z occurs at time 25 seconds. This incipient fault is simulated by a ramp of slope s which is added to the current measured value. The measurement noise w is uniformly distributed between bounds $[-\zeta, \zeta]$. These two parameters $\mathbf{x}_e = [\zeta, s]^T$ are the environmental variables to which the tuning should be robust. \mathbb{X}_e is such that $\zeta \in [10^{-7}, 10^{-3}]$ and $s \in [10^{-3}, 10^{-1}]$.

The fault diagnosis method considered consists of a residual generator and a statistical test. An observer is used

to estimate the state and then the output of the system, which is compared to its measured value to generate a residual sensitive to the sensor fault considered. The nominal mean and variance of this residual on the first 100 values are estimated. The signal is then normalized to zero mean and unit variance according to these estimates, in order to compensate for the differences of behavior induced by a change of values of the environmental variables. Thus, the same tuning of a statistical test is applicable to different levels of noise. A CUSUM test is used on the residual to decide whether it is significantly beyond its initial mean and to provide a Boolean decision on the presence of the fault. The observer has three poles to be placed p_1, p_2, p_3 , and the CUSUM test has two parameters: the size of the change to be detected μ and the associated threshold λ . These five parameters $\mathbf{x}_c = [p_1, p_2, p_3, \mu, \lambda]^T$ are the hyperparameters of the method to be tuned. The set \mathbb{X}_c is defined as follows: $p_1, p_2, p_3 \in [0; 0.8]$, $\mu \in [0.01; 1]$ and $\lambda \in [1; 10]$.

The cost function $y(\mathbf{x}_c, \mathbf{x}_e)$ is $y = r_{\text{fd}} + r_{\text{nd}}$ where r_{fd} and r_{nd} are respectively false-detection and non-detection rates, as defined in Bartyś et al. (2006). It achieves a trade-off between those contradictory objectives and takes bounded values between 0 and 2.

The parameters of the optimization procedure have been set to $\epsilon = 10^{-6}$, $n_{c,\max} = n_{e,\max} = 100$, $\epsilon_{\text{EI}}^c = \epsilon_{\text{EI}}^e = 10^{-4}$. The prior mean of the Gaussian process is assumed constant, while its variance and the parameters θ_k of the correlation function (4) are estimated from the available data by maximum likelihood at each iteration of EGO. Our implementation is based on the toolbox SuperEGO by Sasena (2002). The DIRECT optimization algorithm by Jones et al. (1993) is used to achieve Step 6 of Algorithm 1. One hundred runs of the entire procedure have been performed to assess convergence, repeatability and dispersion of its results. Mean, median and standard deviation for the hyperparameters and environmental variables, along with corresponding values of the cost function and number of evaluations are reported in Table 1. Mean and median values are close, which suggests a bilateral dispersion without too many outliers. Relative dispersion of the results suggest that several tuning of the fault diagnosis method may be suitable to reach acceptable performance. It is important to note that the number of evaluations is quite low with an average sampling of approximately 60 points per dimension, leading to a quick tuning. Moreover, the repetition of the procedure suggests that on this example an acceptable value for the tuning is always obtained, and that the worst-case is correctly identified.

The space of environmental variables $\mathbf{x}_e = [\zeta, s]^T$ is displayed on Figure 1, showing the results for the 100 runs of the procedure on the test case. These results indicate that the worst environmental conditions are located near the smallest value of the fault and highest value of the noise, which makes sense. Figure 2 shows the values obtained for the hyperparameters $\mathbf{x}_c = [p_1, p_2, p_3, \mu, \lambda]^T$ and the associated cost function $y(\mathbf{x}_c, \mathbf{x}_e)$.

Figure 3 shows the residual and corresponding Boolean decision obtained via the observer and the CUSUM test tuned at the mean of their estimated values, for the mean of the evaluated worst-case environmental condi-

Table 1. Results for 100 replications of the tuning procedure

	Median	Mean	Std. deviation
Hyperparameter vector \mathbf{x}_c			
Pole p_1	0.73	0.7363	$5.2 \cdot 10^{-2}$
Pole p_2	0.726	0.7058	$6.6 \cdot 10^{-2}$
Pole p_3	0.72	0.72	$5.3 \cdot 10^{-2}$
Change size μ	0.065	0.0714	$4.9 \cdot 10^{-2}$
Threshold λ	4.553	4.5379	0.2
Environmental vector \mathbf{x}_e			
Noise level ζ	$9.8 \cdot 10^{-4}$	$9.3 \cdot 10^{-4}$	$1.1 \cdot 10^{-4}$
Fault slope s	$1 \cdot 10^{-3}$	$1.1 \cdot 10^{-3}$	$2 \cdot 10^{-4}$
Cost and number of evaluations			
Cost function y	0.114	0.125	$4.7 \cdot 10^{-2}$
Evaluations	419	430.9	220

tion. For comparison, two sample points are taken in the environmental space at $\mathbf{x}_{e,1} = [10^{-4}; 0.01]^T$ and $\mathbf{x}_{e,2} = [10^{-3}; 0.02]^T$. The associated residuals and decision functions are respectively displayed in Figures 4 and 5. Those residuals react more strongly to the fault than in the worst case, and will therefore lead to easier decision. This is confirmed by the corresponding decision functions, obtained by applying the CUSUM test with optimized parameters to the previous residuals. The worst-case residual satisfactorily detects this incipient fault, as no false detection is observed – only a reasonable detection delay. Applying the same tuning of the method for different areas of the environmental space lead to excellent results, as there is still no false detection but also very small detection delays. Figure 6 shows the value of the objective function y over \mathbb{X}_e for the mean worst-case optimal tuning of the hyperparameters. It confirms that the worst-case approach does not degrade performance too much in less adverse operating conditions.

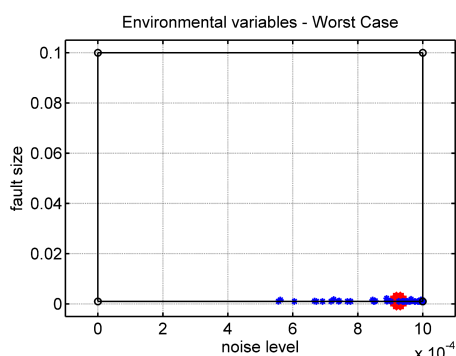


Fig. 1. Worst-case values for the 100 replications of the procedure (blue dots) with mean value (red spot) and \mathbb{X}_e boundaries (black rectangle)

5. CONCLUSIONS AND PERSPECTIVES

A new strategy to address the robust tuning of hyperparameters of fault diagnosis methods has been proposed in this paper. Such a tuning has been formalized as a min-max optimization problem for a black-box function. The hyperparameters should optimize the performance level of the fault detection procedure, for the worst-

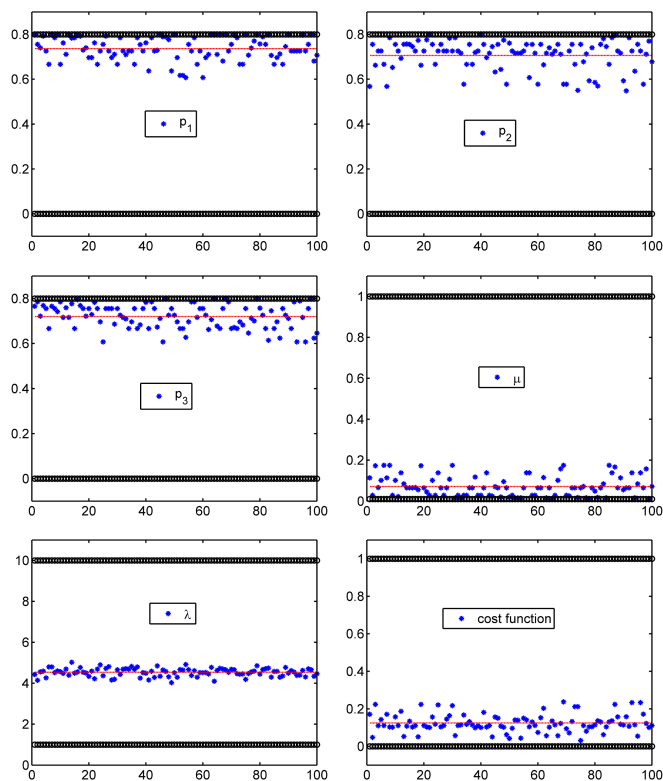


Fig. 2. Dispersion for the 5 hyperparameters $\mathbf{x}_c = [p_1, p_2, p_3, \mu, \lambda]^T$ and cost function. Red line indicates the mean value and thick black lines correspond to space boundaries

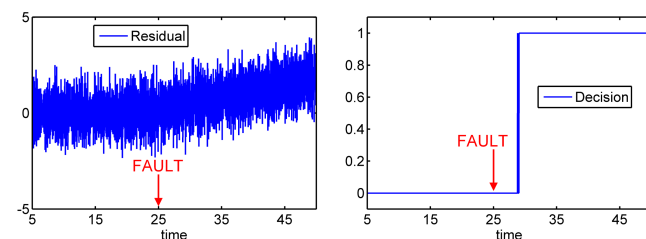


Fig. 3. Residual (left) and Boolean decision (right) for the mean of the estimated worst-case environmental variables $\mathbf{x}_e = [9.3 \cdot 10^{-4}, 1.1 \cdot 10^{-3}]^T$, with the mean of the estimates of the min-max optimal hyperparameters

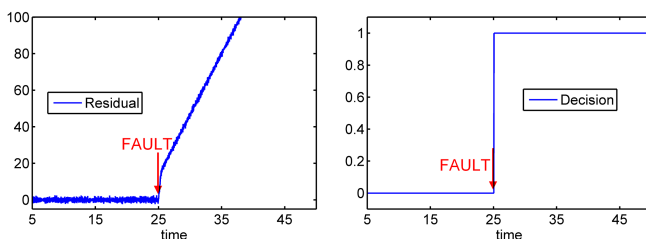


Fig. 4. Residual (left) and Boolean decision (right) for sample point $\mathbf{x}_{e,1} = [10^{-4}; 0.01]^T$, with the mean of the estimates of the min-max optimal hyperparameters

case of environmental variables (measurement noise, disturbances, model uncertainty...). Our solution combines Kriging-based global optimization with a relaxation procedure for solving min-max problems.

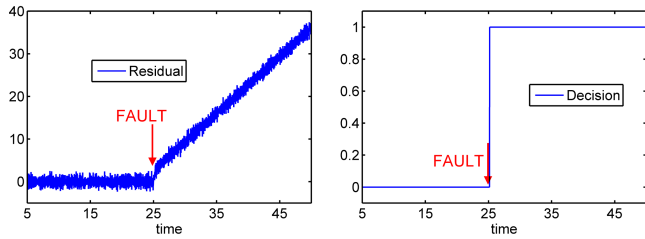


Fig. 5. Residual (left) and Boolean decision (right) for sample point $\mathbf{x}_{e,2} = [10^{-3}; 0.02]^T$, with the mean of the estimates of the min-max optimal hyperparameters

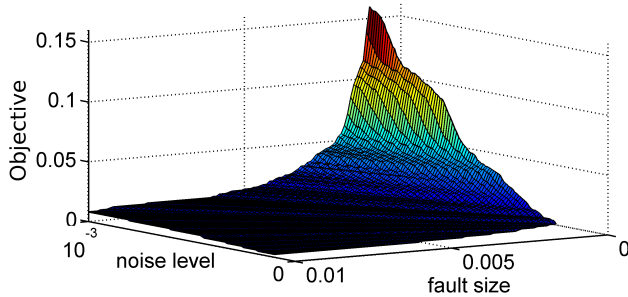


Fig. 6. Value of the objective function over \mathbb{X}_e for the mean of the estimates of the minimax-optimal hyperparameters

The methodology has been illustrated through the tuning of a fault detection strategy comprising a residual generator (observer) and a statistical test (CUSUM) to detect a sensor fault on a dynamical system. The results support the choice of a min-max strategy for the tuning of fault detection methods, as the worst-case tuning may provide a good performance level on all of the environmental space.

Much more complex problems than the simple illustrative example described in this paper can be considered. Other types of hyperparameter tunings can also benefit from this methodology.

REFERENCES

- Bartyś, M., Patton, R.J., Syfert, M., de las Heras, S., and Quevedo, J. (2006). Introduction to the DAMADICS actuator FDI benchmark study. *Control Engineering Practice*, 14(6), 577–596.
- Basseville, M. and Nikiforov, I.V. (1993). *Detection of Abrupt Changes: Theory and Application*. Prentice Hall Englewood Cliffs, NJ.
- Chow, E.Y. and Willsky, A. (1984). Analytical redundancy and the design of robust failure detection systems. *IEEE Transactions on Automatic Control*, 29, 603–614.
- Falcoz, A., Henry, D., and Zolghadri, A. (2009). A nonlinear fault identification scheme for reusable launch vehicles control surfaces. In *Proceedings of the 7th IFAC Symposium on Fault Detection, Supervision and Safety of Technical Processes, SAFEPROCESS 2009, Barcelona, Spain*.
- Falcoz, A., Henry, D., and Zolghadri, A. (2010). Robust fault diagnosis for atmospheric reentry vehicles: a case study. *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans*, 40(5), 886–899.
- Frank, P.M. and Ding, X. (1997). Survey of robust residual generation and evaluation methods in observer-based fault detection systems. *Journal of Process Control*, 7(6), 403–424.
- Jones, D.R., Perttunen, C.D., and Stuckman, B.E. (1993). Lipschitzian optimization without the Lipschitz constant. *Journal of Optimization Theory and Applications*, 79(1), 157–181.
- Jones, D.R., Schonlau, M.J., and Welch, W.J. (1998). Efficient global optimization of expensive black-box functions. *Journal of Global Optimization*, 13(4), 455–492.
- Jones, D. (2001). A taxonomy of global optimization methods based on response surfaces. *Journal of Global Optimization*, 21(4), 345–383.
- Kleijnen, J.P.C. (2009). Kriging metamodeling in simulation: A review. *European Journal of Operational Research*, 192(3), 707–716.
- Lehman, J.S., Santner, T.J., and Notz, W.I. (2004). Designing computer experiments to determine robust control variables. *Statistica Sinica*, 14(2), 571–590.
- Marzat, J., Walter, E., Piet-Lahanier, H., and Damongeot, F. (2010). Automatic tuning via Kriging-based optimization of methods for fault detection and isolation. In *Proceedings of the IEEE Conference on Control and Fault-Tolerant Systems, SYSTOL 2010, Nice, France*.
- Matheron, G. (1963). Principles of geostatistics. *Economic Geology*, 58(8), 1246.
- Rustem, B. (1998). *Algorithms for Nonlinear Programming and Multiple Objective Decisions*. John Wiley & Sons, Ltd.
- Rustem, B. and Howe, M. (2002). *Algorithms for Worst-Case Design and Applications to Risk Management*. Princeton University Press.
- Santner, T.J., Williams, B.J., and Notz, W. (2003). *The Design and Analysis of Computer Experiments*. Springer-Verlag, Berlin-Heidelberg.
- Sasena, M. (2002). *Flexibility and Efficiency Enhancements for Constrained Global Design Optimization with Kriging Approximations*. PhD thesis, University of Michigan, USA.
- Schonlau, M. (1997). *Computer Experiments and Global Optimization*. PhD thesis, University of Waterloo, Canada.
- Shimizu, K. and Aiyoshi, E. (1980). Necessary conditions for min-max problems and algorithms by a relaxation procedure. *IEEE Transactions on Automatic Control*, 25(1), 62–66.
- Zhang, Y. and Qin, S.J. (2009). Adaptive actuator fault compensation for linear systems with matching and unmatched uncertainties. *Journal of Process Control*, 19(6), 985–990.