

# Tuning and comparing fault diagnosis methods for aeronautical systems via Kriging-based optimization

Julien Marzat <sup>\*\*\*</sup>, Hélène Piet-Lahanier <sup>\*</sup>, Frédéric Damongeot <sup>\*</sup>, Eric Walter <sup>\*\*</sup>

<sup>\*</sup>ONERA – The French Aerospace Lab, F-91761 Palaiseau, France

<FirstName>.<LastName>@onera.fr

<sup>\*\*</sup>L2S, CNRS-SUPELEC-Univ Paris-Sud, Gif-sur-Yvette, France

<FirstName>.<LastName>@lss.supelec.fr

## Abstract

Many approaches address fault detection and isolation (FDI) based on analytical redundancy. To rank them, it is necessary to define performance indices and realistic sets of test cases on which these performance indices will be evaluated. For the ranking to be fair, each of the methods under consideration should have its internal parameters (often called *hyperparameters*) tuned optimally. However, no mathematical model linking hyperparameters and performance is available *a priori*. In this paper, we propose to use a combination of tools developed in the context of computer experiments to build such a model from a limited number of numerical evaluations of the performance indices at carefully chosen values of the hyperparameters. The optimal tuning of fault diagnosis methods may prove to be strongly sensitive to specifics of the test cases. This is why the methodology is extended so as to provide a tuning that is robust to variability in the conditions of use. The performance criteria are then replaced by their worst-case values when the sources of variability are assumed to belong to some predefined sets. This methodology is applied to tune fault diagnosis approaches on an aeronautical case study.

## 1. Introduction

Many approaches address fault detection and isolation (FDI) based on analytical redundancy. A review of the main methods can be, e.g., found in [1–3]. Ranking these methods makes it necessary to define performance indices and realistic sets of test cases on which these performance indices will be evaluated. For the ranking to be fair, each of the methods under consideration should have its internal parameters (often called *hyperparameters*) tuned optimally. This is a key issue, as the potential of each method derives directly from this tuning.

However, except in trivial cases, no mathematical model linking hyperparameters and performance is available *a priori*. In this paper, we propose to use a combination of tools developed in the context of computer experiments [4] to build such a model from a limited number of numerical evaluations of the performance indices at carefully chosen values of the hyperparameters. The methodology to be presented is particularly well-suited to computationally-intensive performance indices, which are the rule if the test cases are realistic enough. The approach is based on Kriging [5, 6], which is used to build approximate models linking hyperparameters to performance indices, based on a small number of simulation results. These models are then used to optimize the hyperparameters using a procedure known as *Efficient Global Optimization (EGO)* [7, 8].

Optimal tuning of fault diagnosis methods should not be too sensitive to specifics of the test cases. This is why the methodology is extended so as to provide a tuning that is robust to variability in the conditions of use. The performance criteria are then replaced by their worst-case values when the sources of variability are assumed to belong to some predefined set.

The procedure, in its basic and robust versions, is illustrated by the tuning and comparison of two classical fault diagnosis architectures applied on an aeronautical test case.

## 2. Problem formulation

This paper focuses on fault detection in aeronautics, for instance applied to an air-to-air missile (see Section 3.3). Several methods can be used for detecting faults occurring on this system such as an observer-based residual generator or other types of filters. The quality of the results obtained via each method depends strongly on the adequacy of its tuning to the problem to be treated. This tuning is parameterized by a vector of hyperparameters, defined as  $\mathbf{x}_c \in \mathbb{X}_c$ , where  $\mathbb{X}_c$  is assumed to be a known compact set. The performance measure of fault detection is assumed to be quantified by the real scalar  $y(\mathbf{x}_c)$  to be minimized. The numerical value of  $y(\mathbf{x}_c)$  for any feasible numerical value of  $\mathbf{x}_c$  is obtained by simulating fault scenarios for the aeronautical vehicle and evaluating the quality of detection using the method under consideration for this value of the hyperparameter vector. The complex simulation of such a test case is illustrated in Figure 1, including the dynamics of the aircraft, its control loop and a fault detection scheme composed of a residual generator and an evaluation test.

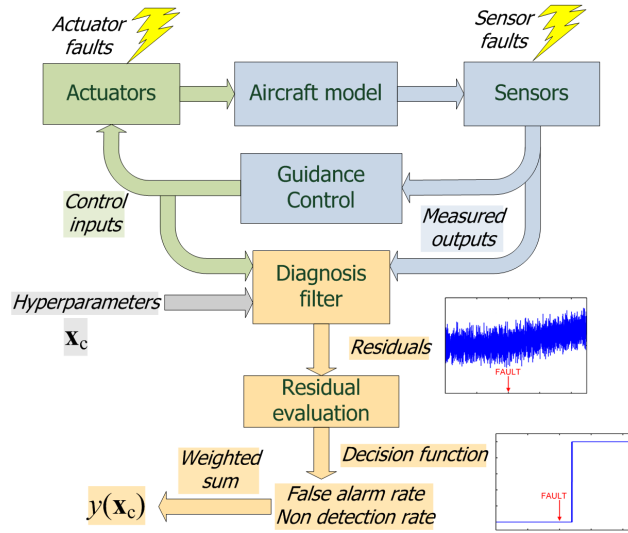


Figure 1: Simulation of the test case

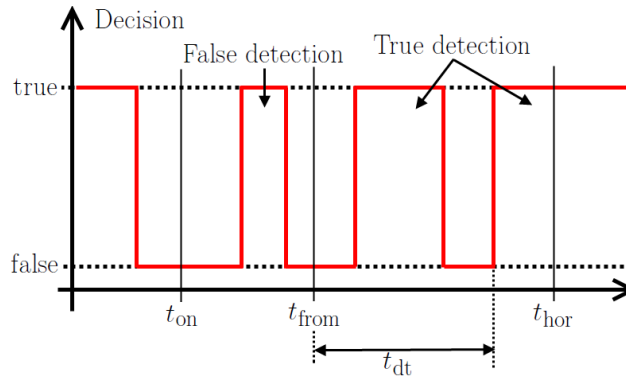


Figure 2: Boolean decision function

Assuming that the output of the detection procedure is a Boolean signature (see Figure 2) on whether a specific fault has appeared, the performance measure  $y(\cdot)$  can be defined as some trade-off between false-alarm and non-detection rates. The test is supposed to run from  $t_{on}$  to  $t_{hor}$ , and the fault starts at  $t_{from} \in ]t_{on}, t_{hor}[$ . Then, the false-alarm rate is

$$r_{fd} = \frac{\sum_i t_{fd}^i}{t_{from} - t_{on}}, \quad (1)$$

where  $t_{fd}^i$  is the  $i$ -th period of time where the decision is *true* in  $[t_{on}; t_{from}]$  and the non-detection rate

$$r_{nd} = \frac{\sum_i t_{nd}^i}{t_{hor} - t_{from}}, \quad (2)$$

where  $t_{nd}^i$  is the  $i$ -th period of time where the decision is *false* in  $[t_{from}; t_{hor}]$ .

The cost function to be used in this paper to measure performance of fault diagnosis methods is

$$y(\mathbf{x}_c) = r_{fd} + r_{nd}. \quad (3)$$

This is a simple trade-off between the conflicting objectives of minimizing false alarms and non detections. Tuning can then be formalized as the search for

$$\widehat{\mathbf{x}}_c = \arg \min_{\mathbf{x}_c \in \mathbb{X}_c} y(\mathbf{x}_c). \quad (4)$$

### 3. Tuning via Kriging

As  $y(\cdot)$  can only be evaluated via possibly very costly numerical simulation at sampled points, a black-box global optimization method is used, which has become very popular in the context of computer experiments. It has been applied in [8] to tune several fault detection methods on the simple example of a change in the mean of a signal subject to Gaussian or uniformly distributed noise. The overall procedure is recursive and presupposes that a set of values of the performance index has already been computed,  $\mathbf{y}_{c,n_c} = [y(\mathbf{x}_{c,1}), \dots, y(\mathbf{x}_{c,n_c})]^T$ , corresponding to an initial sampling of  $n_c$  points in  $\mathbb{X}_c$ ,  $\mathcal{X}_{c,n_c} = [\mathbf{x}_{c,1}, \dots, \mathbf{x}_{c,n_c}]$ . First, a Kriging model (Section 3.1) is fitted to approximate the mapping from the hyperparameter vector  $\mathbf{x}_c$  to the scalar performance measure  $y(\cdot)$ . Then, an iterative optimization procedure (Section 3.2), taking advantage of the Kriging model, is employed to look for a feasible global minimizer  $\widehat{\mathbf{x}}_c$  of the performance index.

#### 3.1 Kriging

The unknown performance function  $y(\cdot)$  is approximated by

$$Y(\mathbf{x}_c) = \mathbf{f}^T(\mathbf{x}_c) \mathbf{b} + Z(\mathbf{x}_c), \quad (5)$$

where  $\mathbf{f}^T(\mathbf{x}_c) \mathbf{b}$  is a classical regression model with the entries of  $\mathbf{f}(\mathbf{x}_c)$  polynomial in the entries of  $\mathbf{x}_c$  and the regression coefficients  $\mathbf{b}$  to be estimated.  $Z(\cdot)$  is a zero-mean Gaussian process with known (or parametrized) covariance function  $k(\cdot, \cdot)$ . Based on this modelling, Kriging is the search for the *best linear unbiased predictor* (BLUP) of  $Y(\cdot)$  [5, 9].

The covariance function  $k(\cdot, \cdot)$  is usually considered as unknown, and expressed as

$$k(Z(\mathbf{x}_{c,i}), Z(\mathbf{x}_{c,j})) = \sigma_Z^2 R(\mathbf{x}_{c,i}, \mathbf{x}_{c,j}), \quad (6)$$

where  $\sigma_Z^2$  is the process variance and  $R(\cdot, \cdot)$  some parametric correlation function. The parameters of  $R(\cdot, \cdot)$  and  $\sigma_Z^2$  must be fixed or estimated from available data.  $R(\mathbf{x}_{c,i}, \mathbf{x}_{c,j})$  is often assumed to depend only on the displacement vector  $\mathbf{h} = \mathbf{x}_{c,i} - \mathbf{x}_{c,j}$ . In this paper, we adopt the classical power exponential correlation function [4],

$$R(\mathbf{h}) = \exp\left(-\sum_{k=1}^d \left|\frac{h_k}{\theta_k}\right|^2\right), \quad (7)$$

where  $h_k$  is the  $k$ -th component of  $\mathbf{h}$ . For our application, *empirical Kriging* was used, which means that the covariance parameters are estimated in the maximum-likelihood sense. To simplify presentation, they will be considered as known in what follows and will not appear in the formulas. Define  $\mathbf{R}$  as the  $n \times n$  matrix such that

$$\mathbf{R}(i, j) = R(\mathbf{x}_{c,i}, \mathbf{x}_{c,j}), \quad (8)$$

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

$$\mathbf{r}(\mathbf{x}_c) = [R(\mathbf{x}_c, \mathbf{x}_{c,1}), \dots, R(\mathbf{x}_c, \mathbf{x}_{c,n})]^T \quad (9)$$

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

$$\mathbf{F} = [\mathbf{f}(\mathbf{x}_{c,1}), \dots, \mathbf{f}(\mathbf{x}_{c,n})]^T. \quad (10)$$

The maximum-likelihood estimate  $\widehat{\mathbf{b}}$  of the vector of the regression coefficients  $\mathbf{b}$  is

$$\widehat{\mathbf{b}} = (\mathbf{F}^T \mathbf{R}^{-1} \mathbf{F})^{-1} \mathbf{F}^T \mathbf{R}^{-1} \mathbf{y}_{c,n_c}. \quad (11)$$

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

$$\widehat{Y}(\mathbf{x}_c) = \mathbf{f}^T(\mathbf{x}_c) \widehat{\mathbf{b}} + \mathbf{r}(\mathbf{x}_c)^T \mathbf{R}^{-1} (\mathbf{y}_{c,n_c} - \mathbf{F} \widehat{\mathbf{b}}). \quad (12)$$

Another crucial property of Kriging that will be exploited in the search for a global optimization of the performance index is the ability to compute the *variance of the prediction error* [10] as

$$\widehat{\sigma}^2(\mathbf{x}_c) = \sigma_Z^2 (1 - \mathbf{r}(\mathbf{x}_c)^T \mathbf{R}^{-1} \mathbf{r}(\mathbf{x}_c)). \quad (13)$$

This variance can be used to assess the confidence in the prediction (12) at any point  $\mathbf{x}_c \in \mathbb{X}_c$ . In particular, it is equal to zero for all the sampled points in  $\mathcal{X}_{c,n_c}$  if simulation errors are deemed negligible as here.

### 3.2 Efficient Global Optimization (EGO)

The idea of EGO [11] is to use the Kriging predictor (12) and the additional confidence information provided by (13) to find an  $(n_c + 1)$ -st vector in  $\mathbb{X}_c$  at which the complex simulation should be run. This point is chosen by maximizing a criterion that measures the interest of an additional evaluation at  $\mathbf{x}_c$ , given the past results  $\mathbf{y}_{c,n_c}$  obtained at  $\mathcal{X}_{c,n_c}$ . A possible choice for this criterion is the *Expected Improvement* (EI) [7],

$$\text{EI}(\mathbf{x}_c, y_{\min}^{n_c}) = \widehat{\sigma}(\mathbf{x}_c) [u \Phi(u) + \phi(u)], \quad (14)$$

where  $\Phi$  is the cumulative distribution function and  $\phi$  the probability density function of the normalized Gaussian distribution  $\mathcal{N}(0, 1)$ , with also

$$u = \frac{y_{\min}^{n_c} - \widehat{Y}(\mathbf{x}_c)}{\widehat{\sigma}(\mathbf{x}_c)}, \quad (15)$$

$$y_{\min}^{n_c} = \min_{i=1 \dots n_c} \{y(\mathbf{x}_{c,i})\}. \quad (16)$$

The maximization of EI achieves a trade-off between local search (numerator of (15)) and the exploration of unknown areas (where  $\widehat{\sigma}$  is high), and is thus well suited for global optimization. The EGO algorithm proceeds as follows.

1. Choose an initial sampling  $\mathcal{X}_{c,n_c}$  and compute  $\mathbf{y}_{c,n_c}$ .
2. Fit a Kriging predictor on those data points.
3. Compute (16) and find a new point  $\mathbf{x}_{n_{c+1}}$  where to evaluate  $y(\cdot)$  by maximizing (14)
4. Append  $\mathbf{x}_{n_{c+1}}$  to  $\mathcal{X}_{c,n_c}$ , append  $y(\mathbf{x}_{n_{c+1}})$  to  $\mathbf{y}_{c,n_c}$  and increment  $n_c$  by one.
5. Go to step 2 until  $\{\max \text{EI}(\cdot)\}$  hits a lower threshold  $\varepsilon$  or until  $n_c$  exceeds a predefined sampling budget  $n_{\max}$ .

The estimate of the global minimum is finally given by a last call to (16), and the corresponding argument in  $\mathcal{X}_{c,n_c}$  is an estimate of a global minimizer. To obtain the results presented in this paper, we used the toolbox SuperEGO [12]. Its parameters have been set to  $\varepsilon = 10^{-4}$  and  $n_{\max} = 100$ .

### 3.3 Illustrative example

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,

$$\begin{cases} \mathbf{x}_{k+1} = \mathbf{A}\mathbf{x}_k + \mathbf{B}u_k \\ \gamma_k = \mathbf{C}\mathbf{x}_k + \mathbf{D}u_k + w_k + f_k \end{cases} \quad (17)$$

where

$$\mathbf{A} = \begin{bmatrix} 0.9163 & 0.0194 & 0.0026 \\ -5.8014 & 0.9412 & 0.5991 \\ -0.0485 & -0.005 & 0.996 \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} -0.0279 \\ -2.5585 \\ -0.0019 \end{bmatrix}, \quad (18)$$

$$\mathbf{C} = \begin{bmatrix} -2.54 & 0 & -0.26 \end{bmatrix}, \quad \mathbf{D} = -0.204.$$

This model is simulated on a time horizon of 50 seconds. Measurement  $\gamma = a_z$  suffers at time = 25 s from a progressive fault  $f$  steadily increasing with a slope equal to 0.1. This measurement is also corrupted by a Gaussian noise  $w$ , with zero mean and standard deviation  $10^{-3}$ .

### 3.3.1 Fault diagnosis methods

Two methods are considered to detect this sensor fault, in line with the architecture described in Figure 1. The first one uses a Luenberger observer and the other is based on a Kalman filter [13]. A residual is produced by each filter as the difference between the available measurement and the predicted output value. In both schemes, the residuals are analyzed using a CUSUM test [14].

The performance of a Luenberger observer depends on the settings of its three poles  $[p_1, p_2, p_3]$ , while the tuning of a Kalman filter relies on the choice of an initial covariance matrix  $\mathbf{W}$  for the state vector perturbation (dimension 3) and of a covariance matrix  $\mathbf{V}$  for measurement noise (dimension 1), both matrices are assumed to be diagonal. This corresponds to 4 hyperparameters  $[w_1, w_2, w_3, v_1]$ , such that

$$\mathbf{W} = \begin{bmatrix} w_1 & 0 & 0 \\ 0 & w_2 & 0 \\ 0 & 0 & w_3 \end{bmatrix}, \quad \mathbf{V} = v_1 \quad (19)$$

The initial value of the covariance matrix  $\mathbf{P}$  of the state estimation error is arbitrarily set to identity (dimension 3). Initial estimated state vector is identical for both filters and taken as  $[17, 21, 2.5]^T$ . The CUSUM test has two hyperparameters, namely the minimum size of change to be detected  $\mu$  and a threshold  $\lambda$ . The tuning hyperparameters of each scheme are those of the residual generator and of the statistical test. Search spaces for these values for each of the competing methods are as indicated in Table 1.

Table 1: Space search of hyperparameters for the two diagnostic methods

Observer and CUSUM	$p_1 \in [0; 1]$	$p_2 \in [0; 1]$	$p_3 \in [0; 1]$	$\mu \in [0.01; 5]$	$\lambda \in [0.1; 20]$
Kalman and CUSUM	$w_1 \in [0; 2]$	$w_2 \in [0; 2]$	$w_3 \in [0; 2]$	$v_1 \in [0; 10]$	$\mu \in [0.01; 5]$ $\lambda \in [0.1; 20]$

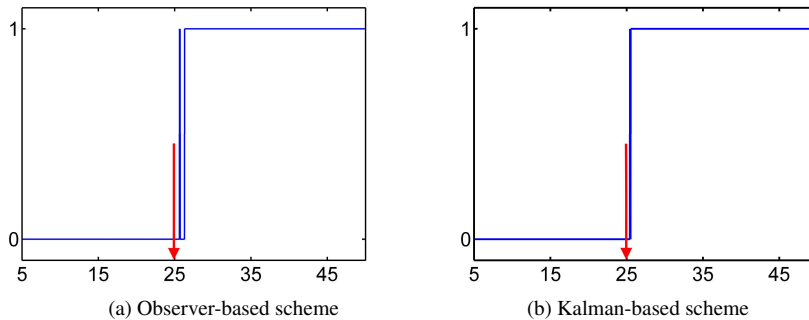


Figure 3: Boolean decision functions for both schemes with best hyperparameter tuning

### 3.4 Results

One hundred initialisations with the tuning methodology have been effected. The number of simulations required for obtaining a suitable tuning remains low (Table 2), and so is the variation on the results. The use of a Kalman filter proves to be more efficient in this setup than that of a Luenberger observer. The resulting decisions are illustrated on Figure 3 .

Table 2: Tuning of complete decision approaches

	Observer and CUSUM	Kalman and CUSUM
<i>Ranking</i>	2	1
<b>Median of performance</b>	<b>0.0455</b>	<b>0.0184</b>
standard deviation of performance	0.009	0.0024
<b>Average number of simulations</b>	<b>102</b>	<b>136</b>
standard deviation of number of simulations	20	21
False alarm rate	0	0
Non detection rate	0.0455	0.0184
Hyperparameter values for median of performance	$p_1 = 0.7675$ $p_2 = 0.8498$ $p_3 = 0.8951$ $\mu = 0.4537$ $\lambda = 2.91$	$w_1 = 0.5428$ $w_2 = 0.2593$ $w_3 = 1.1015$ $v_1 = 2.2291$ $\mu = 0.134$ $\lambda = 4$

## 4. Extension to robustness to environmental conditions

The previous approach does not take into account the variability of environmental conditions (model and measurement uncertainty, variation in atmospheric conditions or magnitude of faults, etc). However, the simulation response can vary greatly with these conditions at the risk of making the optimal tuning inadequate. Environmental conditions are assumed to be described by a vector of environmental variables  $\mathbf{x}_e$  belonging to a known compact set  $\mathbb{X}_e$ .

The tuning of a fault detection method should remain valid for a reasonable set of possible values of the environmental variables. Most of the studies on computer experiments in this context use a probabilistic modeling of the environmental variables [15]. We assume instead that bounds of the values taken by these variables 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, since [13, 16].

Since no algorithm seemed to have been reported to deal with environmental variables for the robust optimization of black-box functions evaluated by costly computer experiments, we have proposed such a procedure in [17]. The problem to be solved can be formalized as that of finding  $\widehat{\mathbf{x}}_c$  and  $\widehat{\mathbf{x}}_e$  such that

$$\{\widehat{\mathbf{x}}_c, \widehat{\mathbf{x}}_e\} = \arg \min_{\mathbf{x}_c \in \mathbb{X}_c} \max_{\mathbf{x}_e \in \mathbb{X}_e} y(\mathbf{x}_c, \mathbf{x}_e), \quad (20)$$

which is a difficult problem since both spaces  $\mathbb{X}_c$  and  $\mathbb{X}_e$  are now continuous. This corresponds to the search for the best hyperparameters for the worst values of the environmental variables. The optimization procedure proposed to solve (20) combines the relaxation algorithm described in [18] with EGO (Section 3.2). A detailed description of the resulting robust tuning algorithm can be found in [17]. The main principles are to transform (20) into

$$\left\{ \begin{array}{l} \min_{\mathbf{x}_c \in \mathbb{X}_c, \tau} \tau \\ \text{subject to } y(\mathbf{x}_c, \mathbf{x}_e) \leq \tau, \forall \mathbf{x}_e \in \mathbb{X}_e \end{array} \right., \quad (21)$$

and then to solve this minimization problem (that has an infinite number of constraints) by an iterative relaxation procedure that computes a finite number of constraints from values in  $\mathbb{X}_e$ . The resulting procedure involves two intertwined global-optimization steps, which are addressed with EGO.

#### 4.1 Illustrative example

The test case and the methods to be compared remain the same as in Section 3.3. The sources of variability are the magnitude of the incipient sensor fault  $f$  and the noise level on  $w$ . The slope of the fault  $s$  and the standard deviation  $\zeta$  of the Gaussian white noise are then the environmental variables to which the tuning should be robust,  $\mathbf{x}_e = [\zeta, s]^T$ . The search space  $\mathbb{X}_e$  is such that  $\zeta \in [10^{-7}, 10^{-3}]$  and  $s \in [10^{-3}, 10^{-1}]$ . The hyperparameter spaces for the two schemes are as before (see Table 1).

One hundred runs of the entire procedure have again been performed to assess convergence, repeatability and dispersion of its results. Performance level, number of evaluations and values of the worst-case environmental variables and of the best hyperparameter tuning for both schemes are reported in Table 3. It is important to note that the number of evaluations is quite low with an average sampling of approximately 33 points per dimension, leading to a quick robust 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. Indeed, for both strategies, the worst environmental conditions are located near the smallest value of the fault and highest value of the noise, which is what common sense suggests.

Table 3: Results for 100 replications of the minimax tuning procedure

	Observer and CUSUM	Kalman and CUSUM
<i>Ranking</i>	2	1
<b>Median minimax performance</b>	<b>0.114</b>	<b>0.0312</b>
standard deviation of performance	$4.7 \cdot 10^{-2}$	$1.97 \cdot 10^{-2}$
<b>Average number of simulations</b>	<b>168</b>	<b>199</b>
standard deviation nb simulations	26	8
<b>Environmental vector <math>\mathbf{x}_e</math></b>		
Noise level $\zeta$	$9.8 \cdot 10^{-4}$	$9.81 \cdot 10^{-4}$
Fault slope $s$	$1 \cdot 10^{-3}$	$1 \cdot 10^{-3}$
<b>Hyperparameter vector <math>\mathbf{x}_c</math></b>		
Hyperparameter values for the median performance	$p_1 = 0.73$ $p_2 = 0.726$ $p_3 = 0.72$ $\mu = 0.065$ $\lambda = 4.553$	$w_1 = 1.57$ $w_2 = 1.11$ $w_3 = 1.06$ $v_1 = 2.04$ $\mu = 0.12$ $\lambda = 3.33$

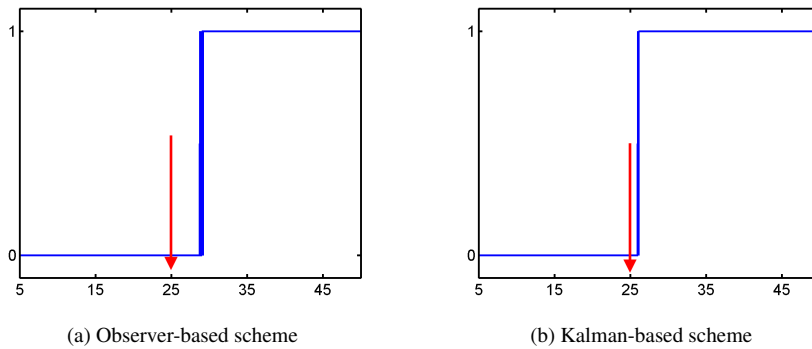


Figure 4: Boolean decision function for both schemes with robust tuning and the worst-case environmental variables

Figure 4 shows the decision functions obtained for both diagnosis methods tuned at the best hyperparameters under the conditions of the worst-case environmental conditions. It clearly appears on this example that the Kalman filter scheme is much better at detecting small incipient faults than the observer scheme, which is not conceived to take

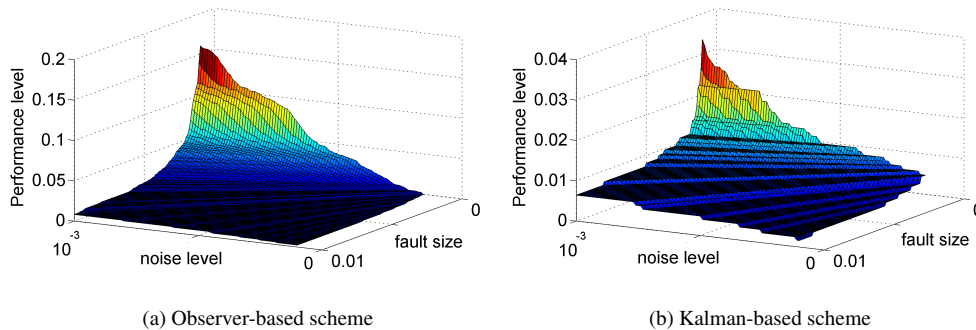


Figure 5: Value of the objective function over  $\mathbb{X}_e$  for the minimax-optimal hyperparameters

into account such noise level. Figure 5 shows the value of the objective function  $y$  over  $\mathbb{X}_e$  for the estimated worst-case optimal tuning of the hyperparameters of the two methods. This shows that the optimal worst-case tuning leads to good performance for other environmental conditions.

## 5. Conclusions and perspectives

This article has presented a framework for automatically tuning fault diagnosis methods. The setup only requires a simulation of a collection of test cases, where the internal parameters of candidate methods may be adjusted and a measure of performance computed. The tuning methodology relies on fitting the mapping from hyperparameters to performance level via Kriging, which is then used to find the optimal hyperparameters. Since any simulation should take into account uncertainty, a robust version of this tuning procedure has been developed in the worst-case sense. It allows one to find at the same time the optimal hyperparameters and the worst environmental conditions. These strategies have been illustrated on an academic version of an aeronautical case study, where two competing fault diagnosis schemes comprising a residual generator and a statistical test were applied. The number of evaluations of the simulation to achieve an acceptable tuning value is always very low, which makes it practicable when the associated computational cost is expensive. The proposed procedures may be readily extended to other more realistic aerospace applications.

## References

- [1] J. Marzat, H. Piet-Lahanier, F. Damongeot, and E. Walter. Autonomous fault diagnosis: State of the art and aeronautical benchmark. In *Proceedings of the 3rd European Conference for Aero-Space Sciences, Versailles*, 2009.
- [2] D. Henry, S. Simani, and R. Patton. Fault detection and diagnosis for aeronautic and aerospace missions. *Fault Tolerant Flight Control*, pages 91–128, 2010.
- [3] S. X. Ding. *Model-based Fault Diagnosis Techniques: Design Schemes, Algorithms, and Tools*. Springer Verlag, Berlin Heidelberg, 2008.
- [4] T. J. Santner, B. J. Williams, and W. Notz. *The Design and Analysis of Computer Experiments*. Springer-Verlag, Berlin-Heidelberg, 2003.
- [5] G. Matheron. Principles of geostatistics. *Economic Geology*, 58(8):1246, 1963.
- [6] C.E. Rasmussen and C.K.I. Williams. *Gaussian Processes for Machine Learning*. Springer-Verlag New York, 2006.
- [7] D.R. Jones. A taxonomy of global optimization methods based on response surfaces. *Journal of Global Optimization*, 21(4):345–383, 2001.



- [8] J. Marzat, E. Walter, H. Piet-Lahanier, and F. Damongeot. Automatic tuning via Kriging-based optimization of methods for fault detection and isolation. In *Proceedings of the 1st IEEE Conference on Control and Fault-Tolerant Systems, SysTol'10, Nice, France*, pages 505–510, 2010.
- [9] J. P. C. Kleijnen. Kriging metamodeling in simulation: A review. *European Journal of Operational Research*, 192(3):707–716, 2009.
- [10] M. Schonlau. *Computer Experiments and Global Optimization*. PhD thesis, University of Waterloo, Canada, 1997.
- [11] D. R. Jones, M. J. Schonlau, and W. J. Welch. Efficient global optimization of expensive black-box functions. *Journal of Global Optimization*, 13(4):455–492, 1998.
- [12] M. J. Sasena. *Flexibility and Efficiency Enhancements for Constrained Global Design Optimization with Kriging Approximations*. PhD thesis, University of Michigan, USA, 2002.
- [13] P. M. Frank and S. X. Ding. Survey of robust residual generation and evaluation methods in observer-based fault detection systems. *Journal of Process Control*, 7(6):403–424, 1997.
- [14] M. Basseville and I. V. Nikiforov. *Detection of Abrupt Changes: Theory and Application*. Prentice Hall Englewood Cliffs, NJ, 1993.
- [15] J. S. Lehman., T. J. Santner, and W. I. Notz. Designing computer experiments to determine robust control variables. *Statistica Sinica*, 14(2):571–590, 2004.
- [16] E. Y. Chow and A. S. Willsky. Analytical redundancy and the design of robust failure detection systems. *IEEE Transactions on Automatic Control*, 29(7):603–614, 1984.
- [17] J. Marzat, E. Walter, and H. Piet-Lahanier. Min-max hyperparameter tuning with application to fault detection. In *Proceedings of the 18th IFAC World Congress, Milan, Italy*, 2011.
- [18] K. Shimizu and E. Aiyoshi. Necessary conditions for min-max problems and algorithms by a relaxation procedure. *IEEE Transactions on Automatic Control*, 25(1):62–66, 1980.