TUNING AND COMPARING FAULT DIAGNOSIS METHODS FOR AERONAUTICAL SYSTEMS VIA KRIGING-BASED OPTIMIZATION

J. Marzat[1,2], H. Piet-Lahanier[1], F. Damongeot[1], and E. Walter[2]

[1]ONERA — The French Aerospace Lab
Palaiseau F-91123, France

[2]L2S, CNRS-SUPELEC-Univ Paris-Sud
Gif-sur-Yvette F-91192, France

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 they will be evaluated. For the ranking to be fair, each of the methods under consideration should have its internal parameters tuned optimally. The work presented uses a combination of tools developed in the context of computer experiments to achieve this tuning from a limited number of numerical evaluations. The methodology is then extended so as to provide a robust tuning in the worst-case sense.

1 INTRODUCTION

Many approaches address FDI based on analytical redundancy. A review of the main methods can be found, e.g., 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 is strongly affected by this tuning [4].

However, except for trivial cases, no mathematical model linking hyperparameters and performance is available *a priori*. In this paper, a combination of tools developed in the context of computer experiments [5] is employed 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 [6, 7], which is used to build approximate models linking hyperparameters to performance indices from a small number of simulation results. These models are then used for hyperparameter optimization with a procedure known as *Efficient Global Optimization* (EGO) [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 sets.

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. The example considered corresponds to an air-to-air missile (see subsection 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 specified by a vector of hyperparameters $\mathbf{x}_c \in \mathbb{X}_c$ where $\mathbb{X}_c$ is some known compact set. The performance of fault detection is assumed to be quantified by a scalar performance index, $y(\mathbf{x}_c)$, 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 $\mathbf{x}_c$. The complex simulation of such a test case is illustrated in Fig. 1, including the dynamics of the aircraft, its control loop, and a fault detection scheme consisting of a residual generator and an evaluation test.

Assume that the output of the detection procedure is a Boolean signature (Fig. 2) on whether a specific fault has appeared. The performance index $y(\cdot)$ can then be defined as some trade-off between false-alarm and nondetection rates. The test runs from t_{on} to t_{hor}, and the fault starts at $t_{\mathrm{from}} \in]t_{\mathrm{on}}, t_{\mathrm{hor}}[$. The false-alarm rate is then

$$r_{\mathrm{fd}} = \frac{\sum\limits_{i} t_{\mathrm{fd}}^{i}}{t_{\mathrm{from}} - t_{\mathrm{on}}}$$

where t_{fd}^{i} is the ith period of time where the decision is *true* in $[t_{\mathrm{on}}; t_{\mathrm{from}}]$ and the nondetection rate
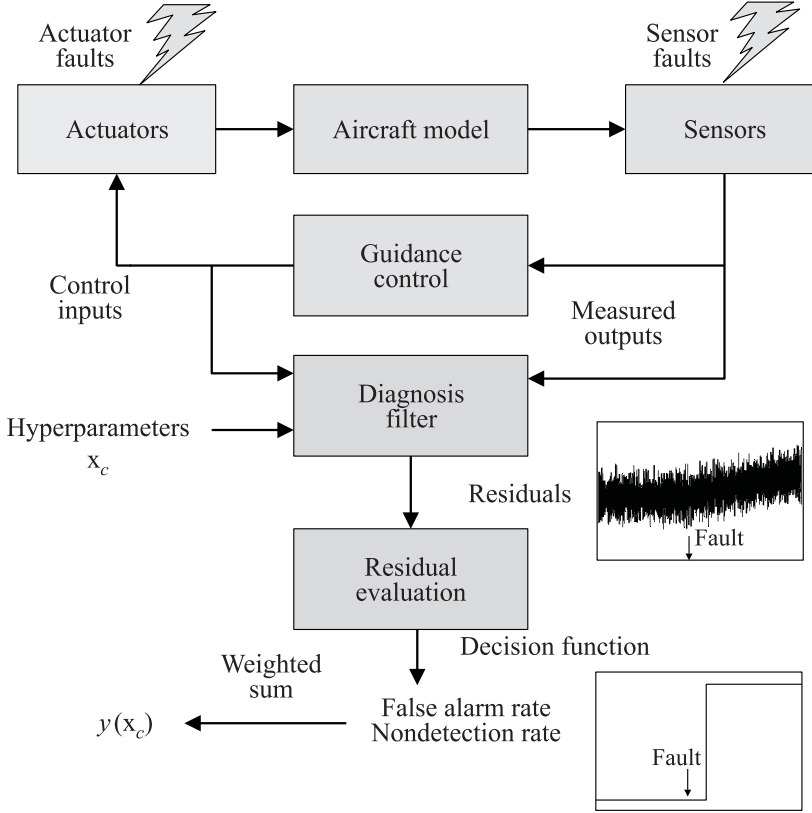
Figure 1 Test case simulation

$$r_{\mathrm{nd}} = \frac{\sum_i t^i_{\mathrm{nd}}}{t_{\mathrm{hor}} - t_{\mathrm{from}}}$$

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

The performance index to be used in this paper is

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

It is thus a simple trade-off between the conflicting objectives
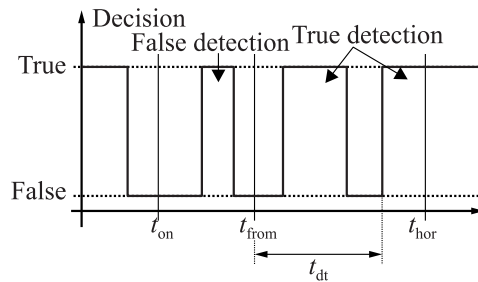

Figure 2 Boolean decision function

of minimizing false alarms and nondetections. Tuning can then be formalized as the search for

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

3 TUNING VIA KRIGING

As $y(\cdot)$ can only be evaluated via possibly very costly numerical simulations at sampled points in $\mathbb{X}_c$, a black-box global optimization method is used, which has become very popular in the context of computer experiments. It has been applied in [9] 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 starts from a set of values of the performance index already computed, $\mathbf{y}_{c,n_c} = [y(\mathbf{x}_{c,1}), \ldots, y(\mathbf{x}_{c,n_c})]^{\mathrm{T}}$, corresponding to an initial sampling of n_c points in $\mathbb{X}_c$, $\mathcal{X}_{c,n_c} = [\mathbf{x}_{c,1}, \ldots, \mathbf{x}_{c,n_c}]$. First, a Kriging model (see subsection 3.1) is fitted to approximate the mapping from the hyperparameter vector $\mathbf{x}_c$ to the scalar performance index $y(\mathbf{x}_c)$. An iterative optimization procedure (see subsection 3.2), taking advantage of the Kriging model, is then employed to look for a feasible global minimizer $\widehat{\mathbf{x}}_c$ of the performance index.

3.1 Kriging

The unknown performance index $y(\cdot)$ is approximated by the Gaussian process:

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

where $\mathbf{f}^{\mathrm{T}}(\mathbf{x}_c)\mathbf{b}$ is the classical regression model with $\mathbf{b}$ being a vector of regression coefficients to be estimated; and $Z(\cdot)$ is the zero-mean Gaussian process with known (or parametrized) covariance function $k(\cdot, \cdot)$. Based on this modeling, Kriging is the search for the *best linear unbiased predictor* (BLUP) of $Y(\cdot)$ [6].

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

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

where σ_Z^2 is the process variance and $R(\cdot, \cdot)$ is the correlation function, which may depend on unknown parameters. The parameters of $R(\cdot, \cdot)$ and σ_Z^2 may be 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, the classical power exponential correlation function is adopted [5]:

$$R(\mathbf{h}) = \exp\left(-\sum_{k=1}^{\dim \mathbf{x}_c} \left|\frac{h_k}{\theta_k}\right|^2\right)$$

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

$$\mathbf{R}(i,j) = R\left(\mathbf{x}_{c,i}, \mathbf{x}_{c,j}\right) \; ;$$

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

$$\mathbf{r}\left(\mathbf{x}_c\right) = \left[R\left(\mathbf{x}_c, \mathbf{x}_{c,1}\right), \ldots, R\left(\mathbf{x}_c, \mathbf{x}_{c,n_c}\right)\right]^{\mathrm{T}} \; ;$$

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

$$\mathbf{F} = \left[\mathbf{f}\left(\mathbf{x}_{c,1}\right), \ldots, \mathbf{f}\left(\mathbf{x}_{c,n_c}\right)\right]^{\mathrm{T}} \; .$$

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

$$\widehat{\mathbf{b}} = \left(\mathbf{F}^{\mathrm{T}}\mathbf{R}^{-1}\mathbf{F}\right)^{-1} \mathbf{F}^{\mathrm{T}}\mathbf{R}^{-1}\mathbf{y}_{c,n_c} \; .$$

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

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

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

$$\widehat{\sigma}^2\left(\mathbf{x}_c\right) = \sigma_Z^2\left(1 - \mathbf{r}\left(\mathbf{x}_c\right)^{\mathrm{T}}\mathbf{R}^{-1}\mathbf{r}\left(\mathbf{x}_c\right)\right) \; . \tag{2}$$

This variance can be used to assess confidence in the prediction (1) at any $\mathbf{x}_c \in \mathbb{X}_c$. In particular, it is equal to zero at all the sampled points in $\mathcal{X}_{c,n_c}$ if simulation errors are deemed negligible, as here.

3.2 Efficient Global Optimization

Efficient Global Optimization [11] uses the Kriging predictor (1) and the additional confidence information provided by (2) to find an $(n_c + 1)$ vector in $\mathbb{X}_c$ at which the complex simulation should be run. This point is chosen by maximizing the *Expected Improvement* (EI) [8]:

$$\mathrm{EI}(\mathbf{x}_c, y_{\min}^{n_c}) = \widehat{\sigma}\left(\mathbf{x}_c\right)\left[u\Phi\left(u\right) + \phi\left(u\right)\right] \tag{3}$$

which measures the interest of an additional evaluation at $\mathbf{x}_c$, given the past results $\mathbf{y}_{c,n_c}$ obtained at $\mathcal{X}_{c,n_c}$. In (3), $\Phi(\cdot)$ is the cumulative distribution function,

$\phi(\cdot)$ is the probability density function of the normalized Gaussian distribution $\mathcal{N}(0, 1)$:

$$u = \frac{y_{\min}^{n_c} - \widehat{Y}(\mathbf{x}_c)}{\widehat{\sigma}(\mathbf{x}_c)} \; ; \tag{4}$$

and

$$y_{\min}^{n_c} = \min_{i=1,\ldots,n_c} \{y(\mathbf{x}_{c,i})\} \; . \tag{5}$$

The maximization of EI achieves a trade-off between local search (numerator of (4)) 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 (5) and find a new point $\mathbf{x}_{c,n_c+1}$ where to evaluate $y(\cdot)$ by maximizing (3).

4. Append $\mathbf{x}_{c,n_c+1}$ to $\mathcal{X}_{c,n_c}$, append $y(\mathbf{x}_{c,n_c+1})$ to $\mathbf{y}_{c,n_c}$, and increment n_c by one.

5. Go to Step 2 until $\{\max \mathrm{EI}(\cdot)\}$ gets lower than a given threshold ε or until n_c exceeds a predefined sampling budget $n_{\max}$.

The estimate of the global minimum is finally obtained by a last call to (5), 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, the toolbox SuperEGO [10] was used with $\varepsilon = 10^{-4}$ and $n_{\max} = 100$.

3.3 Illustrative Example

The test case is based on 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 $\mathbf{x} = [\alpha, q, \mathrm{M}]^{\mathrm{T}}$. The control input is the rudder angle, $u = \delta$, and the available measurement is the normal acceleration, $\gamma = a_{\mathrm{z}}$. The linearized model around the operating point $\mathbf{x}_0 = [\bar{\alpha}, \bar{q}, \bar{\mathrm{M}}]^{\mathrm{T}} = [20°, 18.4 \text{ deg/s}, 3]^{\mathrm{T}}$ is given by the following state-space representation, after discretization with a time step of 0.02 s:

$$\mathbf{x}_{k+1} = \mathbf{A}\mathbf{x}_k + \mathbf{B}u_k \; ; \quad \gamma_k = \mathbf{C}\mathbf{x}_k + \mathbf{D}u_k + w_k + f_k$$

where

368

$$\mathbf{A} = \begin{bmatrix} 0.9163 & 0.0194 & 0.0026 \\ -5.8014 & 0.9412 & 0.5991 \\ -0.0485 & -0.0050 & 0.9960 \end{bmatrix} ; \quad \mathbf{B} = \begin{bmatrix} -0.0279 \\ -2.5585 \\ -0.0019 \end{bmatrix} ;$$

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

This model is simulated on a time horizon of 50 s. Measurement $\gamma = a_z$ suffers from time $t = 25$ s 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}.

Fault diagnosis methods

Two methods are considered to detect this sensor fault, in line with the architecture described in Fig. 1. The first one uses a Luenberger observer and the other is based on a Kalman filter [12]. A residual is produced by each filter as the difference between the available measurement and the predicted output. In both schemes, the residuals are analyzed using a CUSUM (CUmulative SUM) test [13].

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 here to be diagonal. This corresponds to 4 hyperparameters $[w_1, w_2, w_3, v_1]$, such that $\mathbf{W} = \text{diag}(w_1, w_2, w_3)$ and $\mathbf{V} = v_1$. The diagonal coefficients of the covariance matrix $\mathbf{P}$ of the state estimation error are arbitrarily initialized to 10^3. The initial estimate of the state vector is identical for both filters and taken as $\mathbf{x}_0 = [17, 21, 2.5]^T$. The CUSUM test has two hyperparameters, namely, the minimum size of change to be detected μ and a threshold λ. 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 Search space for the hyperparameters of the two diagnosis methods

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

3.4 Results

The procedure described above has been applied with one hundred randomly chosen initial $\mathcal{X}_{c,n_c}$. The number of simulations required for obtaining a suitable tuning remains low, as well as the variability on the results (Table 2). The use of

369

Table 2 Tuning of the two diagnosis methods

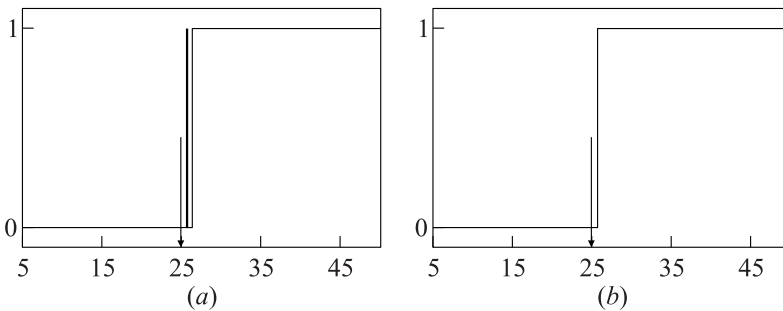| Parameter	Observer and CUSUM	Kalman and CUSUM
Ranking	*2*	*1*
Median performance index	**0.0455**	**0.0184**
Standard deviation of performance index	0.009	0.0024
Average number of simulations	**102**	**136**
Standard deviation of number of simulations	20	21
False alarm rate	0	0
Nondetection rate	0.0455	0.0184
Hyperparameter values for median performance index	$p_1 = 0.7675;$ $p_2 = 0.8498;$ $p_3 = 0.8951;$ $\mu = 0.4537;$ $\lambda = 2.91$	$w_1 = 0.5428;$ $w_2 = 0.2593;$ $w_3 = 1.1015;$ $v_1 = 2.2291;$ $\mu = 0.134;$ $\lambda = 4$

Figure 3 Boolean decision functions for both schemes with best hyperparameter tuning: (*a*) observer-based scheme; and (*b*) Kalman-based scheme

a Kalman filter proves to be more efficient in this setup than that of a Luenberger observer. The resulting decisions are illustrated in Fig. 3 .

4 ROBUSTIFICATION TO FLUCTUATIONS IN THE 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). Another interesting application of the concept of environmental conditions is that it makes it possible to introduce

370

degrees of freedom in the definition of test cases, thus making them more general. Environmental conditions are assumed in what follows to be described by a vector of environmental variables $\mathbf{x}_e$ belonging to a known compact set $\mathbb{X}_e$. Since the actual value of the performance index may vary greatly with the value of $\mathbf{x}_e$, there is a significant risk that the optimal tuning obtained for nominal environmental conditions turn out to be inadequate.

Certainly, it is desirable that the tuning of a fault detection method remains valid for a reasonable set of possible values of the environmental variables. Most studies on computer experiments in this context use a probabilistic modeling of the environmental variables [14]. Here, instead, it is assumed that bounds on the values taken by $\mathbf{x}_e$ 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 (see, e. g., [12, 15]).

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, such procedure was proposed in [16]. The problem to be solved can be formalized as that of finding $\widehat{\mathbf{x}}_c$ and $\widehat{\mathbf{x}}_e$ such that

$$\{\widehat{\mathbf{x}}_c, \widehat{\mathbf{x}}_e\} = \arg \min_{\mathbf{x}_c \in \mathbb{X}_c} \max_{\mathbf{x}_e \in \mathbb{X}_e} y(\mathbf{x}_c, \mathbf{x}_e) \,. \tag{6}$$

This is a difficult problem since both spaces $\mathbb{X}_c$ and $\mathbb{X}_e$ are continuous. It corresponds to the search for the best hyperparameters for the worst values of the environmental variables. The optimization procedure proposed to solve (6) combines the relaxation algorithm described in [17] with EGO (see subsection 3.2). A detailed description of the resulting robust tuning algorithm can be found in [16]. The main principles are to transform (6) into

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

and then to solve this minimization problem (which 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.

Illustrative Example

The test case and the methods to be compared are as in subsection 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 ζ of the Gaussian white noise are the environmental variables to which the tuning should

be robust, $\mathbf{x}_e = [\zeta, s]^{\mathrm{T}}$. 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 were performed again 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. The number of evaluations remains 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.

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 into account such a noise level. Figure 5 shows the value of the performance index $y(\cdot, \cdot)$ over $\mathbb{X}_e$ for the estimated worst-case optimal tuning of the hyperparameters of the two methods. It suggests that the optimal worst-case tuning still leads to good performance for other environmental conditions.

Table 3 Results for 100 replications of the minimax tuning procedure

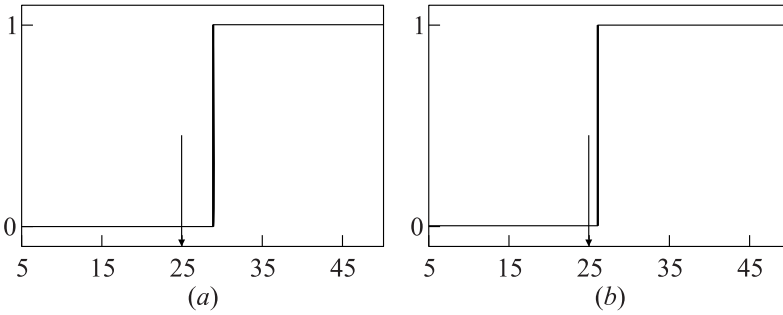| Parameter	Observer and CUSUM	Kalman and CUSUM
Ranking	*2*	*1*
Median minimax performance index	**0.114**	**0.0312**
Standard deviation of performance index	$4.7 \cdot 10^{-2}$	$1.97 \cdot 10^{-2}$
Average number of simulations	**168**	**199**
Standard deviation nb simulations	26	8
Estimated worst-case environmental vector $\mathbf{x}_e$		
Noise level ζ	$9.8 \cdot 10^{-4}$	$9.81 \cdot 10^{-4}$
Fault slope s	$1 \cdot 10^{-3}$	$1 \cdot 10^{-3}$
Estimated minimax hyperparameter vector $\mathbf{x}_c$		
Hyperparameter values for the median minimax performance index	$p_1 = 0.73$; $p_2 = 0.726$; $p_3 = 0.72$; $\mu = 0.065$; $\lambda = 4.553$	$w_1 = 1.57$; $w_2 = 1.11$; $w_3 = 1.06$; $v_1 = 2.04$; $\mu = 0.12$; $\lambda = 3.33$

Figure 4 Boolean decision function for both schemes with robust tuning and the worst-case environmental variables: (*a*) Observer-based scheme; and (*b*) Kalman-based scheme
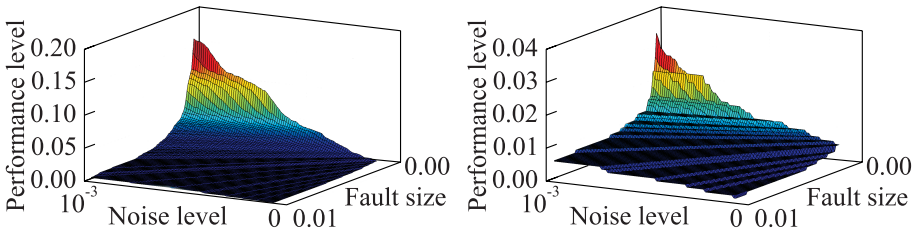

Figure 5 Value of the objective function over $\mathbb{X}_e$ for the minimax-optimal hyper-parameters: (*a*) Observer-based scheme; and (*b*) Kalman-based scheme.

5   CONCLUDING REMARKS AND PERSPECTIVES

This paper presents 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 performance index computed. The tuning methodology relies on fitting the mapping from hyperparameters to performance 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 compared. The number of possibly complex simulations required to achieve an

373

acceptable tuning value is always very low, which makes it practicable even when evaluating the performance index is computationally expensive. The proposed procedure readily extends to more realistic aerospace applications.

REFERENCES

1. Ding, S. X. 2008. *Model-based fault diagnosis techniques: Design schemes, algorithms, and tools.* Berlin–Heidelberg: Springer Verlag.

2. Henry, D., S. Simani, and R. Patton. 2010. Fault detection and diagnosis for aeronautic and aerospace missions. In: *Fault tolerant flight control.* Eds. C. Edwards, T. Lombaerts, and H. Smaili. 91–128.

3. Marzat, J., H. Piet-Lahanier, F. Damongeot, and E. Walter. 2012. Model-based fault diagnosis for aerospace systems: A survey. *IMechE Proceedings, Part G: J. Aerospace Eng.* 226(10):1329–60.

4. Efimov, D., A. Zolghadri, and P. Simon. 2010. Improving fault detection abilities of extended Kalman filters by covariance matrices adjustment. *IEEE Conference on Control and Fault-Tolerant Systems, SysTol'10, Proceedings.* Nice, France. 131–36.

5. Santner, T. J., B. J. Williams, and W. Notz. 2003. *The design and analysis of computer experiments.* Berlin–Heidelberg: Springer-Verlag.

6. Matheron, G. 1963. Principles of geostatistics. *Economic Geology* 58(8):1246–66.

7. Kleijnen, J. P. C. 2009. Kriging metamodeling in simulation: A review. *Eur. J. Operational Res.* 192(3):707–16.

8. Jones, D. R. 2001. A taxonomy of global optimization methods based on response surfaces. *J. Global Optimization* 21(4):345–83.

9. Marzat, J., E. Walter, H. Piet-Lahanier, and F. Damongeot. 2010. Automatic tuning via Kriging-based optimization of methods for fault detection and isolation. *1st IEEE Conference on Control and Fault-Tolerant Systems, SysTol'10, Proceedings.* Nice, France. 505–10.

10. Sasena, M. J. 2002. Flexibility and efficiency enhancements for constrained global design optimization with Kriging approximations. Ph.D. Thesis. University of Michigan, USA.

11. Jones, D. R., M. J. Schonlau, and W. J. Welch. 1998. Efficient global optimization of expensive black-box functions. *J. Global Optimization* 13(4):455–92.

12. Frank, P. M., and S. X. Ding. 1997. Survey of robust residual generation and evaluation methods in observer-based fault detection systems. *J. Proc. Control* 7(6):403–24.

13. Basseville, M., and I. V. Nikiforov. 1993. *Detection of abrupt changes: Theory and application.* Englewood Cliffs, NJ: Prentice Hall.

14. Lehman, J. S., T. J. Santner, and W. I. Notz. 2004. Designing computer experiments to determine robust control variables. *Statistica Sinica* 14(2):571–90.

15. Chow, E. Y., and A. S. Willsky. 1984. Analytical redundancy and the design of robust failure detection systems. *IEEE Trans. Automatic Control* 29(7):603–14.

16. Marzat, J., E. Walter, and H. Piet-Lahanier. 2011. Min-max hyperparameter tuning with application to fault detection. *18th IFAC World Congress Proceedings.* Milan, Italy. 12904–9.

17. Shimizu, K., and E. Aiyoshi. 1980. Necessary conditions for min-max problems and algorithms by a relaxation procedure. *IEEE Trans. Automatic Control* 25(1):62–66.