Preprints of the 16th IFAC Symposium on System Identification
The International Federation of Automatic Control
Brussels, Belgium. July 11-13, 2012

# Robust automatic tuning of diagnosis methods via an efficient use of costly simulations

Julien Marzat * Eric Walter ** Frédéric Damongeot *
Hélène Piet-Lahanier *

* *ONERA – The French Aerospace Lab, F–91123 Palaiseau, France,*
*julien.marzat@onera.fr, frederic.damongeot@onera.fr,*
*helene.piet-lahanier@onera.fr*
** *Laboratoire des Signaux et Systèmes (L2S),*
*CNRS-SUPELEC-Univ Paris-Sud, France, eric.walter@lss.supelec.fr*

**Abstract:** The robust tuning methodology developed in this paper aims at adjusting automatically the hyperparameters of fault-diagnosis procedures for complex case studies. The strategy should make an efficient use of computer simulations of these case studies, which will usually be computationally expensive. To this end, Kriging-based optimization is called upon. Robustness to environmental disturbances is achieved by continuous minimax optimization, and handled through an iterative relaxation procedure. This strategy is applied to the automatic tuning of a model-based fault-diagnosis scheme for a realistic aerospace application.

*Keywords:* aerospace, automatic tuning, fault detection and isolation, fault diagnosis, hyperparameter, Kriging, minimax optimization, robust tuning, satellite.

## 1. INTRODUCTION

This paper addresses the automatic tuning of fault-diagnosis procedures via the optimization of performance on simulated, realistic case studies. There are at least two reasons why such an automatic tuning is highly desirable. The first one is to make the comparison of methods fairer. A classical approach is by assessing their performance on the same set of simulated case studies. Performance, however, depends not only on the intrinsic quality of the methods, but also on their tuning. Internal parameters, (called *hyperparameters* in the following) should thus be adjusted. Unequal knowledge of the candidate methods may lead to a better tuning of the best-known ones, and thus to a biased comparison. A second reason is that the number of hyperparameters for a given method is often too large for intuition alone to allow proper tuning, especially when robustness with respect to uncertainty is taken into account, as it should. Sources of uncertainty to be faced include modeling errors, the imperfect nature of system components, measurement noise, and degrees of freedom in the case studies. The latter may be introduced voluntarily in order to avoid tuning a method too specifically with respect to the conditions of application. We assume that uncertainty can be characterized by the fact that some vector of *environmental* variables belongs to some known compact set.

Search for the best tuning for the hyperparameters is formulated as an optimization problem, where performance is evaluated through simulation. Since simulating realistic, complex test cases may be very costly, one should make an efficient use of simulation samples. A fine-grid exploration of input space or any other strategy that requires a large number of samples is thus not well suited to the task (Korniyenko et al. (2006)). We propose, instead, to rely on Kriging-based optimization. Kriging, popularized by Matheron (1963), is a method to build surrogate models that is widely used in the domain of *Computer Experiments* (Santner et al. (2003); Rasmussen and Williams (2006)). It makes it possible not only to predict performance on the basis of the simulations already carried out but also to compute a statistical measure of confidence in this prediction. These properties have been exploited to design the *Efficient Global Optimization (EGO)* procedure (Jones et al. (1998)). EGO iteratively suggests new values of the parameters to be optimized (here, the hyperparameters), where performance should be evaluated. Search is guided by a criterion that can be evaluated at low cost, based on the Kriging prediction.

The problem addressed in this paper radically differs from those previously handled with EGO, as we deal with two types of vectors of variables, namely the hyperparameter and environmental vectors. We look for a feasible hyperparameter vector that maximizes performance for the worst feasible value of the vector of environmental variables. Such a minimax optimization of costly-to-evaluate black-box functions is a difficult and important problem that does not seem to have been addressed in the literature. To fill this gap, an algorithm is described in Section 2, whose basic features have been presented in Marzat et al. (2011). It combines Kriging-based optimization with an iterative relaxation procedure to look for a minimax solution. Section 3 is dedicated to the application of this strategy to the robust automatic tuning of a scheme for the detection and isolation of faults on the actuators of an aerospace vehicle, taking into consideration several uncertainty sources.

## 2. ROBUST TUNING METHODOLOGY

The starting point to tune a fault-diagnosis procedure via the proposed methodology is to classify the input variables of a computer simulation of the complex problem under study into two vectors, namely the hyperparameters $\mathbf{x}_\mathrm{c}$ and the environmental variables $\mathbf{x}_\mathrm{e}$. Hyperparameters are the *tunable* internal parameters of the algorithms composing the fault-diagnosis scheme, while environmental variables represent the sources of uncertainty affecting the model of the system within the simulation. We assume that a scalar performance index $J(\mathbf{x}_\mathrm{c}, \mathbf{x}_\mathrm{e})$, reflecting the adequacy of the fault-diagnosis strategy, is computable for any combination of the values of $\mathbf{x}_\mathrm{c} \in \mathbb{X}_\mathrm{c}$ and $\mathbf{x}_\mathrm{e} \in \mathbb{X}_\mathrm{e}$. The only requirement on $\mathbb{X}_\mathrm{c}$ and $\mathbb{X}_\mathrm{e}$ is for them to be known compact sets.

### *2.1 Continuous minimax optimization*

The objective of tuning is to find an optimal value $\widehat{\mathbf{x}}_\mathrm{c}$ of the hyperparameter vector with respect to the performance index $J$, for the worst value $\widehat{\mathbf{x}}_\mathrm{e}$ of the environmental vector. Since $J$ can only be evaluated at sampled values of $\{\mathbf{x}_\mathrm{c}, \mathbf{x}_\mathrm{e}\}$, the problem to be solved amounts to the continuous minimax optimization of a black-box function,

$$\{\widehat{\mathbf{x}}_\mathrm{c}, \widehat{\mathbf{x}}_\mathrm{e}\} = \arg \min_{\mathbf{x}_\mathrm{c} \in \mathbb{X}_\mathrm{c}} \max_{\mathbf{x}_\mathrm{e} \in \mathbb{X}_\mathrm{e}} J\left(\mathbf{x}_\mathrm{c}, \mathbf{x}_\mathrm{e}\right). \tag{1}$$

This is a delicate issue, especially when the number of evaluations of the performance index is severely limited. Previous theoretical work on continuous minimax optimization has focused on the development of algorithms that use the analytical expressions of functions (see Rustem and Howe (2002) for an overview). This allows one to take advantage of the formal computation of gradient or sub-gradient information, which is impractical here. From another point of view, robust optimization for costly computer simulations has been addressed through the probabilistic modeling of environmental variables and the inclusion of randomness into Kriging-based optimization (Apley et al. (2006)), leaving aside the worst-case formulation considered here.

To render (1) tractable, it is first rewritten as an equivalent optimization problem with an infinite number of constraints,

$$\left\{ \begin{array}{c} \min\limits_{\mathbf{x}_\mathrm{c} \in \mathbb{X}_\mathrm{c}} \tau, \\ \text{subject to } J(\mathbf{x}_\mathrm{c}, \mathbf{x}_\mathrm{e}) \leq \tau, \ \forall \mathbf{x}_\mathrm{e} \in \mathbb{X}_\mathrm{e}. \end{array} \right. \tag{2}$$

The following procedure, proposed by Shimizu and Aiyoshi (1980), aims at finding an approximate minimax solution by relaxing the constraints iteratively.

(1) Pick $\mathbf{x}_\mathrm{e}^{(1)} \in \mathbb{X}_\mathrm{e}$, set $\mathcal{R}_\mathrm{e} = \left\{\mathbf{x}_\mathrm{e}^{(1)}\right\}$ and $i = 1$.

(2) Compute $\mathbf{x}_\mathrm{c}^{(i)} = \arg \min\limits_{\mathbf{x}_\mathrm{c} \in \mathbb{X}_\mathrm{c}} \left\{ \max\limits_{\mathbf{x}_\mathrm{e} \in \mathcal{R}_\mathrm{e}} J(\mathbf{x}_\mathrm{c}, \mathbf{x}_\mathrm{e}) \right\}$

(3) Compute $\mathbf{x}_\mathrm{e}^{(i+1)} = \arg \max\limits_{\mathbf{x}_\mathrm{e} \in \mathbb{X}_\mathrm{e}} J(\mathbf{x}_\mathrm{c}^{(i)}, \mathbf{x}_\mathrm{e})$

(4) If $J(\mathbf{x}_\mathrm{c}^{(i)}, \mathbf{x}_\mathrm{e}^{(i+1)}) - \max\limits_{\mathbf{x}_\mathrm{e} \in \mathcal{R}_\mathrm{e}} J(\mathbf{x}_\mathrm{c}^{(i)}, \mathbf{x}_\mathrm{e}) < \varepsilon$ then return $\{\mathbf{x}_\mathrm{c}^{(i)}, \mathbf{x}_\mathrm{e}^{(i+1)}\}$ as an approximate solution to the initial minimax problem (1). Else, append $\mathbf{x}_\mathrm{e}^{(i+1)}$ to $\mathcal{R}_\mathrm{e}$, increment $i$ by 1 and go to Step 2.

Constraint relaxation is achieved at Step 2, where the function to be minimized is the maximum of the performance index over the *finite* set $\mathcal{R}_\mathrm{e}$. Steps 2 and 3 leave open the choice of the optimization algorithm to compute the optima required. Since the performance index is evaluated via the costly simulation of case studies, the procedure adopted should be able to cope with a severely restricted simulation budget. To this end, we propose to employ Kriging-based optimization, already used in Marzat et al. (2010) for the automatic tuning of change-detection methods (without taking robustness into consideration).

### *2.2 Kriging-based optimization*

Consider the search for a global minimizer of a function $f(\boldsymbol{\xi})$ that is known only at sampled locations $\boldsymbol{\xi} \in \mathbb{X}$. Assume that an initial set of $n$ sampled points, $\mathcal{X}_n = \{\boldsymbol{\xi}_1, \ldots, \boldsymbol{\xi}_n\}$, is available, as well as the corresponding function values $\mathbf{f}_n = [f(\boldsymbol{\xi}_1), \ldots, f(\boldsymbol{\xi}_n)]^\mathrm{T}$. Based on this initial knowledge, Kriging aims at approximating $f$ over the continuous space $\mathbb{X}$ by a function much simpler to compute. To do so, the regression model

$$F(\boldsymbol{\xi}) = \boldsymbol{\pi}(\boldsymbol{\xi})^\mathrm{T}\mathbf{b} + Z(\boldsymbol{\xi}) \tag{3}$$

is used, where $\boldsymbol{\pi}(\boldsymbol{\xi})^\mathrm{T}\mathbf{b}$ is a mean model, scalar product of a vector of regressors $\boldsymbol{\pi}$ (e.g., polynomial in $\boldsymbol{\xi}$) and a vector of coefficients $\mathbf{b}$ to be estimated, while $Z(\boldsymbol{\xi})$ is a zero-mean Gaussian Process (GP). The covariance function of the GP is expressed as

$$\mathrm{cov}\left(Z(\boldsymbol{\xi}_i, \boldsymbol{\xi}_j)\right) = \sigma_Z^2 R(\boldsymbol{\xi}_i, \boldsymbol{\xi}_j), \tag{4}$$

where $\sigma_Z^2$ is the process variance and $R(\cdot, \cdot)$ a correlation function, possibly parameterized by a vector $\boldsymbol{\theta}$. In the application presented below, the correlation function is taken as

$$R(\boldsymbol{\xi}_i, \boldsymbol{\xi}_j) = \exp\left( - \sum_{i=1}^{\dim \mathbb{X}} \left| \frac{\xi_i(k) - \xi_j(k)}{\theta_k} \right|^2 \right), \tag{5}$$

where $\xi_i(k)$ is the $k$-th component of $\boldsymbol{\xi}_i$ and the positive coefficients $\theta_k$ are scale factors that quantify the influence of $f(\boldsymbol{\xi}_i)$ over $f(\boldsymbol{\xi}_j)$, as a function of the distance between $\boldsymbol{\xi}_i$ and $\boldsymbol{\xi}_j$. It should be kept in mind that other structures may be appropriate (Santner et al. (2003)). The Kriging predictor is then

$$\widehat{f}(\boldsymbol{\xi}) = \boldsymbol{\pi}\left(\boldsymbol{\xi}\right)^\mathrm{T} \widehat{\mathbf{b}} + \mathbf{r}\left(\boldsymbol{\xi}\right)^\mathrm{T} \mathbf{R}^{-1} \left(\mathbf{f}_n - \boldsymbol{\Pi}\widehat{\mathbf{b}}\right), \tag{6}$$

where

$$\begin{array}{l} \mathbf{R}|_{i,j} = R(\boldsymbol{\xi}_i, \boldsymbol{\xi}_j), \ \{i, j\} = 1, .., n, \\ \mathbf{r}(\boldsymbol{\xi}) = [R(\boldsymbol{\xi}_1, \boldsymbol{\xi}), ..., R(\boldsymbol{\xi}_n, \boldsymbol{\xi})]^\mathrm{T}, \\ \boldsymbol{\Pi} = [\boldsymbol{\pi}(\boldsymbol{\xi}_1), ..., \boldsymbol{\pi}(\boldsymbol{\xi}_n)]^\mathrm{T}, \\ \widehat{\mathbf{b}} = \left(\boldsymbol{\Pi}^\mathrm{T}\mathbf{R}^{-1}\boldsymbol{\Pi}\right)^{-1} \boldsymbol{\Pi}^\mathrm{T}\mathbf{R}^{-1}\mathbf{f}_n. \end{array} \tag{7}$$

The statistical properties of the Kriging model also make it possible to obtain confidence intervals for the prediction (6), by estimating the variance of the prediction error

$$\widehat{\sigma}^2\left(\boldsymbol{\xi}\right) = \sigma_Z^2 \left(1 - \mathbf{r}\left(\boldsymbol{\xi}\right)^\mathrm{T} \mathbf{R}^{-1} \mathbf{r}\left(\boldsymbol{\xi}\right)\right). \tag{8}$$

The process variance $\sigma_Z^2$ and the vector of parameters $\boldsymbol{\theta}$ of the correlation function (if any) can be estimated, for instance, by maximum likelihood.

The Efficient Global Optimization (EGO) procedure (Jones et al. (1998)), exploits this information to search iteratively for a global minimizer of $f$. The knowledge of (8) makes it possible to strike a compromise between local search (in the neighborhood of the current estimate of the minimizer) and global search (where uncertainty of the prediction is strong). EGO is initialized by sampling $n$ points in $\mathbb{X}$ (as a rule of thumb, one may use $n = 10 \times \dim \boldsymbol{\xi}$) and computing their corresponding function values. It then proceeds as follows,

(1) Fit a Kriging model on available data $\{\mathcal{X}_n, \mathbf{f}_n\}$
(2) Find $f^n_{\min} = \min_{i=1,\dots,n} \{f(\boldsymbol{\xi}_i)\}$
(3) Find $\boldsymbol{\xi}_{n+1} = \arg\max_{\boldsymbol{\xi}\in\mathbb{X}} \left\{ \mathrm{EI}\left(\boldsymbol{\xi}, f^n_{\min}, \widehat{f}, \widehat{\sigma}\right)\right\}$, see (9)
(4) Append $\boldsymbol{\xi}_{n+1}$ to $\mathcal{X}_n$ and $f(\boldsymbol{\xi}_{n+1})$ to $\mathbf{f}_n$
(5) $n \leftarrow n+1$
(6) If $n > n_{\max}$ (budget exhausted), return $f^n_{\min}$ as an estimate for the global minimum. Else go to Step 1.

The Expected Improvement (EI) criterion, used at Step 3, is evaluated as

$$\mathrm{EI}\left(\boldsymbol{\xi}, f^n_{\min}, \widehat{f}, \widehat{\sigma}\right) = \widehat{\sigma}\left(\boldsymbol{\xi}\right)\left[u\Phi\left(u\right) + \phi\left(u\right)\right], \qquad (9)$$

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

$$u = \frac{f^n_{\min} - \widehat{f}\left(\boldsymbol{\xi}\right)}{\widehat{\sigma}\left(\boldsymbol{\xi}\right)}. \qquad (10)$$

By optimizing this criterion, EGO achieves an iterative search for the global minimum of $f$ and an associated global minimizer. It boils down to solving at each step an optimization problem at a low computational cost, since EI has a closed-form expression, instead of the cumbersome initial black-box optimization problem.

Two EGO routines should be used to tackle the optimization Steps 2 and 3 of the minimax procedure described in Section 2.1. The parameters of the complete procedure are thus the budgets of evaluations, which include the evaluations involved in the initial sampling, and the threshold $\varepsilon$. Latin Hypercube Sampling is used for initialization, while the intermediate maximization of the EI criterion at Step 3 is addressed by the DIRECT optimization procedure, as recommended by Sasena (2002).

## 3. APPLICATION

A simple aeronautical example was presented in Marzat et al. (2011) to illustrate the potential of the methodology. In this section, a much more intricate problem is considered, with 6 hyperparameters and 7 sources of uncertainty.

### 3.1 Simulation

Consider a deep-space satellite with 16 thrusters (organized in 4 sets), similar to the one presented by Williamson et al. (2009) or Patton et al. (2010). Its general shape is that of a cube with side length 2 m. Figure 1 summarizes the simulation of the closed-loop control of the satellite. The thrusters may suffer faults (biases) that are to be detected by a bank of nonlinear observers coupled with statistical tests. Sensors are assumed to be non-faulty in the setup considered.

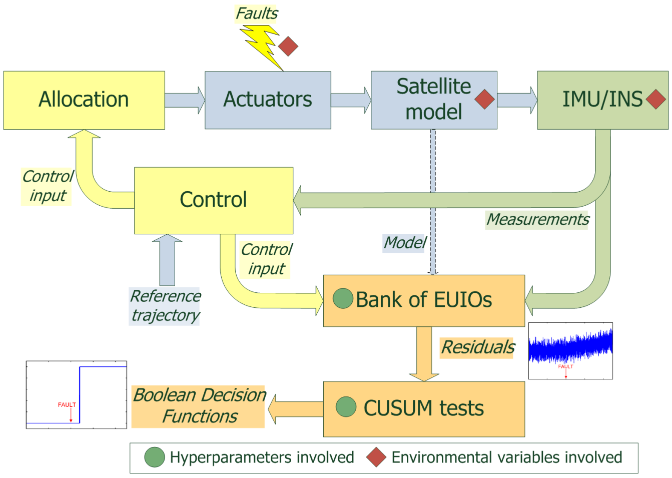


Fig. 1. Simulation scheme

*Nonlinear dynamical model* The 13-dimensional state vector comprises the position $\mathbf{p}$ and velocity $\mathbf{v}$ of the satellite (3 components each), the quaternion $\mathbf{q}$ (4 components) and the angular velocity $\boldsymbol{\omega}$ (3 components). The vector of the control variables of the 16 thrusters is denoted by $\mathbf{u}$. The deterministic part of the (nonlinear) dynamical model is then

$$\begin{cases} \dot{\mathbf{p}} = \mathbf{v} \\ \dot{\mathbf{v}} = m^{-1}\left(\mathbf{R}_{\mathrm{bi}}(\mathbf{q})\mathbf{B}_{\mathrm{t}}\mathbf{u}\right) \\ \dot{\mathbf{q}} = 0.5\left(\mathbf{R}_{\mathrm{q}}(\boldsymbol{\omega})\mathbf{q}\right) \\ \dot{\boldsymbol{\omega}} = \mathbf{I}^{-1}\left(\boldsymbol{\omega}\times\mathbf{I}\boldsymbol{\omega} + \mathbf{B}_{\mathrm{r}}\mathbf{u}\right) \end{cases}, \qquad (11)$$

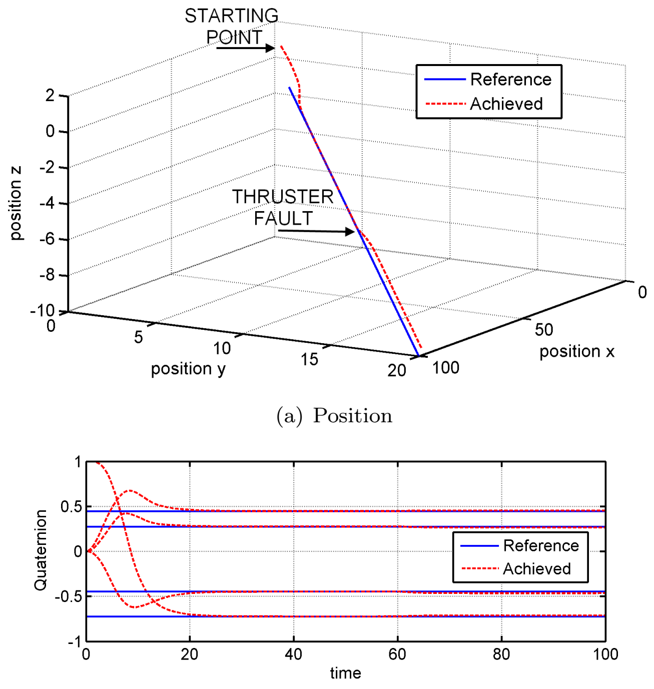
where $m$ is the satellite mass, $\mathbf{I}$ its inertia matrix (diagonal in the nominal case), $\mathbf{B}_{\mathrm{t}}$ and $\mathbf{B}_{\mathrm{r}}$ represent respectively the translational and rotational effect of the thruster inputs on the satellite motion. The matrices $\mathbf{R}_{\mathrm{q}}(\boldsymbol{\omega})$ and $\mathbf{R}_{\mathrm{bi}}(\mathbf{q})$ manage the changes of coordinates from a fixed reference frame to a frame linked to the body of the vehicle.

An inertial navigation system (INS) makes use of an inertial measurement unit (IMU) to provide an estimate of the state vector. The IMU measurements are assumed to be corrupted by an additive Gaussian white noise, with zero mean and nominal standard deviation as indicated in Table 1.

Table 1. Nominal uncertainty parameters

| Type of uncertainty | Standard deviation |
|---|---|
| Gyro noise | $0.01\,\mathrm{deg}\cdot\mathrm{s}^{-1}$ |
| Accelerometer noise | $10^{-6}\,\mathrm{m}\cdot\mathrm{s}^{-2}$ |
| Control inaccuracy | $5\,\mathrm{N}$ |

*Control law and allocation of actuators* The satellite should follow a reference trajectory with a desired orientation, while minimizing power consumption (minimal speed). Two state feedback laws are designed in a decoupled way. The first one is for position and speed control and the second one for orientation and angular velocity control. Both designs are based on proportional controllers, the outputs of which follow a change of coordinates to cope with the nonlinearities of (11) (see Wie (1998)). The nonlinear iterative pseudoinverse control allocation procedure (Jin et al. (1995)) is used for mapping the 3 forces and 3 torques provided by the control module on the 16 thrusters. Each thruster operates from 0 to 100 N and the achieved control inputs are assumed to suffer a random inaccuracy, as indicated in Table 1. Figure 2 shows the trajectory of the controlled satellite simulated on 100 seconds, on which the effect of a thruster fault to be detected could also be seen.



(a) Position

(b) Quaternion

Fig. 2. Simulation of satellite motion

*Fault-detection strategy* The fault considered is an abrupt bias occurring on the first thruster at time 60 s. A bank of Extended Unknown Input Observers (EUIO, see Witczak (2007)), each of them associated to a CUSUM statistical test (see Basseville and Nikiforov (1993)), is employed to detect and isolate this fault. Each of the 16 filters of the bank uses all the inputs but one, and is insensitive to faults on the input left out. Each residual of the bank is analyzed by the statistical test to provide a Boolean decision indicating whether the residual has reacted. Decision on the presence of a fault is confirmed if an alarm is raised by all of the Boolean decision functions but the one corresponding to the fault to be detected.

The performance index reflecting the appropriate detection and isolation of a fault on the $i$-th thruster is thus expressed as

$$J = N r^i_{\mathrm{fd}} + \sum_{j=1,\ j\neq i}^{N} \left( r^j_{\mathrm{fd}} + r^j_{\mathrm{nd}} \right), \qquad (12)$$

where $r^j_{\mathrm{fd}}$ is the false-alarm rate and $r^j_{\mathrm{nd}}$ the non-detection rate corresponding to the $j$-th decision function of the bank. This translates the need of non-reaction, during the entire simulation, for the decision function dedicated to the fault to be detected, while the other decision functions should react quickly after the occurrence of this fault. The computation of these rates is achieved as indicated in Bartyś et al. (2006). In the case considered here, $N$ is equal to 16, the number of thrusters.

### *3.2 Optimization problem setting*

The components of the simulation that involve hyperparameters or environmental variables are highlighted in Figure 1. The hyperparameters of the fault-detection scheme are the poles of the EUIOs and the parameters of the associated CUSUM tests. Since all the thrusters act symmetrically and have the same power range, we use the same observer tuning and statistical test for all the elements of the bank. Moreover, the placement of 4 multiple poles instead of 13 single ones is considered. This choice corresponds to the 4 vectors composing the state vector of model (11), whose components have similar dynamics. The 6 resulting hyperparameters and their domains of variation are indicated in Table 2.

Table 2. Hyperparameters $\mathbf{x}_{\mathrm{c}}$(dim. 6)

| Hyperparameters | Domain |
|---|---|
| Poles of observer ($p_{1-3}$, $p_{4-6}$, $p_{7-10}$ and $p_{11-13}$) | $[-10; -0.1]^4$ |
| CUSUM size of change $\mu$ | $[1; 10]$ |
| CUSUM threshold $\lambda$ | $[0.1; 1]$ |

The environmental variables are the size of fault (bias on the first thruster control input), the deviation of the noise levels on the IMU sensors with respect to their nominal values, the position error on the center of mass (CoM) along the three directions and the error on the mass of the vehicle. The size of fault and noise levels have an impact on the difficulty of fault detection. CoM and mass errors represent the discrepancy between the actual satellite model and the design model, which might disturb decoupling and state estimation. The 7 environmental variables and their domains of variation are in Table 3.

Table 3. Environmental variables $\mathbf{x}_{\mathrm{e}}$(dim. 7)

| Environmental variables | Domain |
|---|---|
| Size of fault (N) | $[10; 50]$ |
| Noise level on translation sensors (%) | $[0; 200]$ |
| Noise level on rotation sensors (%) | $[0; 200]$ |
| Center of mass error $x/y/z$ axis (m) | $[-0.2; 0.2]^3$ |
| Uncertainty on mass (%) | $[-10; 10]$ |

*Tuning methodology* The average computation time of one evaluation of the performance index is 45 seconds on a 2.2 GHz CPU. To cope with this constraint, the simulation budget (per iteration of the relaxation algorithm) has been fixed to

- 60 LHS points followed by 60 iterations for the search for $\widehat{\mathbf{x}}_{\mathrm{c}}$ (Step 2).
- 70 LHS points followed by 70 iterations for the search for $\widehat{\mathbf{x}}_{\mathrm{e}}$ (Step 3).

The threshold for the termination of the iterative relaxation loop has been fixed to $\varepsilon = 10^{-2}$.

### *3.3 Results*

Only two iterations of the minimax procedure were necessary to obtain an error smaller than the threshold for the relaxation loop. Table 4 displays the results at the end of each of these iterations, including the estimated worst case value $\widehat{\mathbf{x}}_{\mathrm{e}}$ and optimal tuning $\widehat{\mathbf{x}}_{\mathrm{c}}$, and the number of evaluations required. The approximate solution of the minimax tuning problem is given by the estimates obtained at the end of the second iteration, which needed 834 evaluations of the performance index (about 10 hours of computation). This amounts to 64 evaluations per dimension of the minimax problem, which is quite small in view of its complexity and dimension.

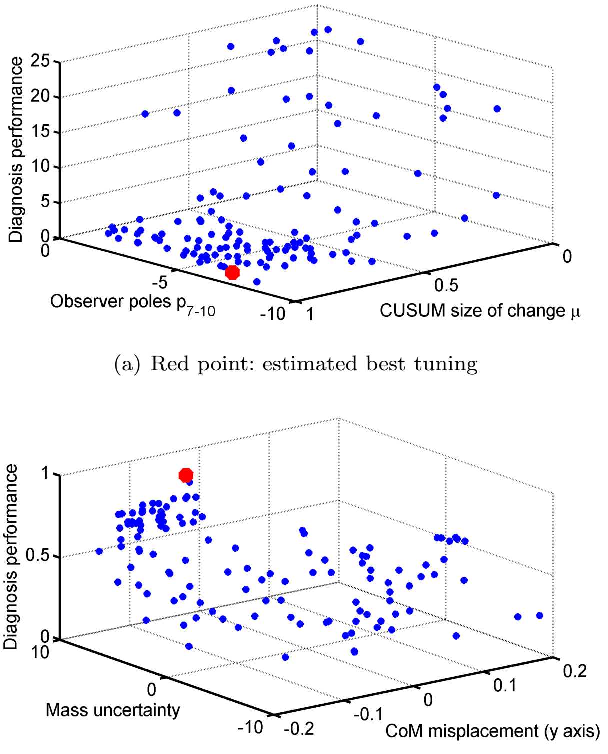
Table 4. Minimax numerical results

| Iteration number | 0 | 1 | 2 |
|---|---|---|---|
| $\widehat{\mathbf{x}}_{\mathrm{e}}$ | 11.7 | 10 | 10 |
| | 7 | 199.9 | 1.58 |
| | 168.5 | 89.9 | 44.4 |
| | −0.19 | 0.07 | −0.18 |
| | −0.17 | 0.13 | −0.02 |
| | 0 | −0.07 | −0.17 |
| | −6 | −6.3 | 10 |
| $\widehat{\mathbf{x}}_{\mathrm{c}}$ | - | −2.62 | −1.2 |
| | | −5.45 | −3.9 |
| | | −4.73 | −6.6 |
| | | −5.91 | −4.99 |
| | | 0.89 | 0.94 |
| | | 5.32 | 4.65 |
| Evaluations of $J$ | 0 | 275 | 834 |

Figure 3 displays the exploration of the hyperparameter and environmental spaces by EGO. These functions do not seem to be trivial to optimize, since there seems to exist several convenient hyperparameter tunings (Figure 3.a) as well as several local maximizers for the environmental variables (Figure 3.b). An intuitive location for the worst case would correspond to the smallest size of fault (10 N), the highest noise levels (200%) and one of the bounds of the remaining four environmental variables (errors on the three coordinates of the CoM position and mass). The performance for these 16 extrema values, the nominal case $[30, 100, 100, 0, 0, 0, 0]$ and the estimated worst case are displayed on Figure 4, via a projection on two of the environmental variables. It is thus interesting to remark that common sense was misleading, as the estimated worst case is not on the boundary of $\mathbb{X}_{\mathrm{e}}$. Figure 5 shows the 16 residuals obtained with the optimal tuning $\widehat{\mathbf{x}}_{\mathrm{c}}$ at the estimated worst case $\widehat{\mathbf{x}}_{\mathrm{e}}$. Figure 6 presents the corresponding values of the decision functions. The first residual remains globally closer to zero than the others, which react to the fault at time 60 s, even if the adverse environmental conditions diminish these reactions compared to what would have been obtained in the nominal case. The method and the associated optimal tuning thus seem sufficient to cope with the range of sources of uncertainty considered.

## 4. CONCLUSIONS AND PERSPECTIVES

This paper has described the application of a robust automatic tuning methodology for fault-diagnosis procedures to a realistic case study from the aerospace domain. The idea of the method is to consider this problem as the continuous minimax optimization of a black-box function, which is the diagnosis performance response to a choice of hyperparameters and uncertainty conditions. A dedicated relaxation algorithm is used to tackle this complex problem, taking advantage of Kriging and the associated EGO procedure for the iterative search for an optimum. For the application presented, a minimax optimal tuning is obtained at a reasonable computational cost. Beyond the results presented here, it seems important to point out



(a) Red point: estimated best tuning

(b) Red point: estimated worst case

Fig. 3. Exploration of hyperparameter (a) and environmental (b) spaces (projections)

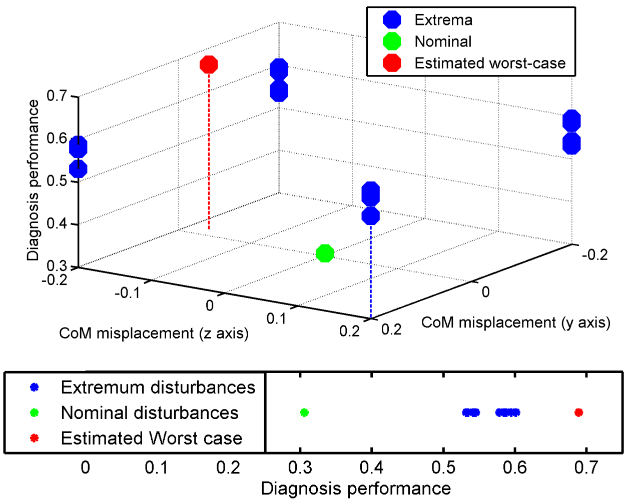


Fig. 4. Performance for extrema, nominal and worst-case (as estimated) disturbances with optimal tuning

that a method for the robust minimax tuning of hyperparameters based on a few costly simulations is now available and should find many applications in various fields. Future work includes testing other realistic applications and theoretical developments on the introduction of constraints other than bounds on the input domains.

## ACKNOWLEDGMENTS

The authors would like to thank Florian Avrillon (former ONERA trainee) and Antoine Abauzit (ONERA), who contributed to the definition of the satellite test case.

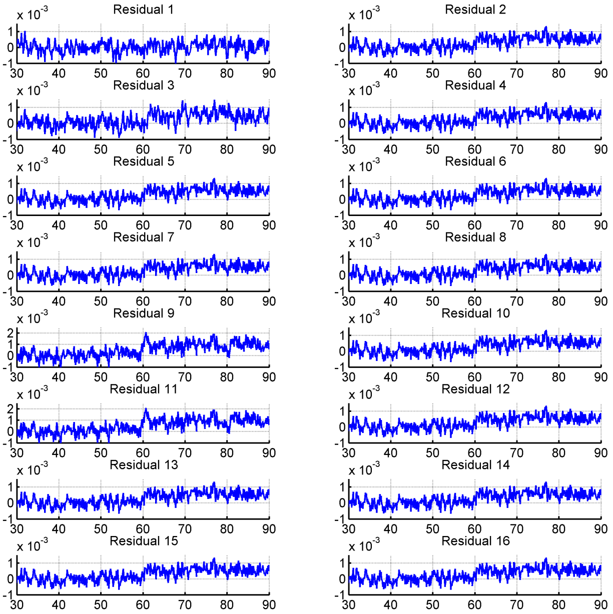


Fig. 5. Residuals for optimal tuning of hyperparameters with respect to worst case of environmental variables

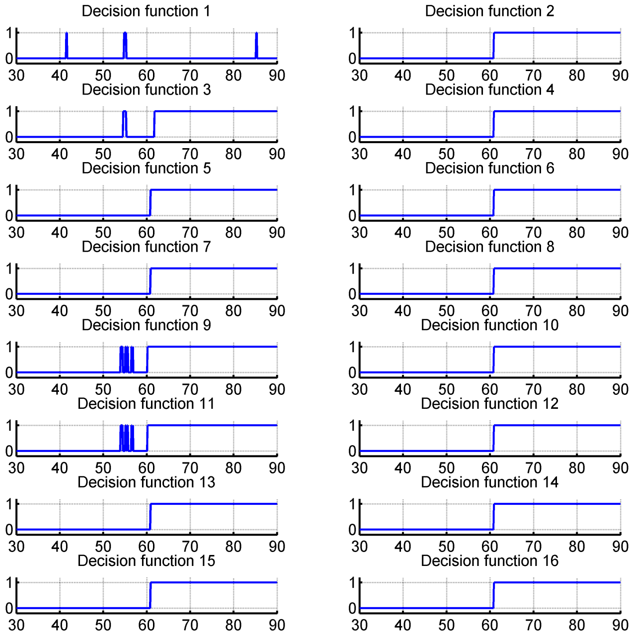


Fig. 6. Decision functions for optimal minimax tuning

## REFERENCES

Apley, D., Liu, J., and Chen, W. (2006). Understanding the effects of model uncertainty in robust design with computer experiments. *Journal of Mechanical Design*, 128, 945–958.

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.

Jin, H.P., Wiktor, P., and DeBra, D.B. (1995). An optimal thruster configuration design and evaluation for Quick STEP. *Control Engineering Practice*, 3(8), 1113–1118.

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.

Korniyenko, O.V., Sharawi, M.S., and Aloi, D.N. (2006). Neural network based approach for tuning Kalman filter. In *Proceedings of the IEEE International Conference on Electro/Information Technology, Lincoln, USA.*

Marzat, J., Walter, E., and Piet-Lahanier, H. (2011). Min-max hyperparameter tuning with application to fault detection. In *Proceedings of the 18th IFAC World Congress, Milan, Italy*, 12904–12909.

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'10, Nice, France*, 505–510.

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

Patton, R.J., Uppal, F.J., Simani, S., and Polle, B. (2010). Robust FDI applied to thruster faults of a satellite system. *Control Engineering Practice*, 18(9), 1093–1109.

Rasmussen, C.E. and Williams, C.K.I. (2006). *Gaussian Processes for Machine Learning.* Springer-Verlag, New York.

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.J. (2002). *Flexibility and Efficiency Enhancements for Constrained Global Design Optimization with Kriging Approximations.* PhD Thesis, University of Michigan, USA.

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.

Wie, B. (1998). *Space Vehicle Dynamics and Control.* AIAA Educational Series, Reston, Virginia.

Williamson, W.R., Speyer, J.L., Dang, V.T., and Sharp, J. (2009). Fault detection for deep space satellites. *Journal of Guidance, Control and Dynamics*, 32(5), 1570–1584.

Witczak, M. (2007). *Modelling and Estimation Strategies for Fault Diagnosis of Non-Linear Systems: From Analytical to Soft Computing Approaches.* Springer-Verlag, Berlin Heidelberg.