



## Fault Detection using Interval Kalman Filtering enhanced by Constraint Propagation

Jun Xiong, Carine Jauberthie, Louise Travé-Massuyès, Françoise Le Gall

### ► To cite this version:

Jun Xiong, Carine Jauberthie, Louise Travé-Massuyès, Françoise Le Gall. Fault Detection using Interval Kalman Filtering enhanced by Constraint Propagation. Conference on Decision and Control, Dec 2013, Florence, Italy. 10.1109/CDC.2013.6759929 . hal-01966325

**HAL Id: hal-01966325**

**<https://laas.hal.science/hal-01966325>**

Submitted on 28 Dec 2018

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Fault Detection using Interval Kalman Filtering enhanced by Constraint Propagation

Jun Xiong<sup>1,2</sup>, Carine Jaubertie<sup>1,3</sup>, Louise Travé-Massuyès<sup>1,4</sup> and Françoise Le Gall<sup>1,4</sup>

**Abstract**—In this paper, we consider an extension of conventional Kalman filtering to discrete time linear models with bounded uncertainties on parameters and gaussian measurement noise. To solve the interval matrix inversion problem involved in the equations of the Kalman filter and the over-bounding problem due to interval calculus, we propose an original approach combining the set inversion algorithm SIVIA and constraint propagation. The improved interval Kalman filter is applied in a fault detection schema illustrated by a simple case study.

## I. INTRODUCTION

Set-membership (SM) methods have been the focus of a growing interest and they have been applied to many tasks ([1], [2], [3]). The literature on this topic shows interesting progress in the last years. SM estimation can be based on interval analysis that was introduced by [4] and several algorithms have been proposed (for more details, see [1], [2], [5]). Other approaches dedicated to linear models include ellipsoid shaped methods ([6], [7]), parallelotope and zonotope based methods [8].

In contrast to stochastic estimation approaches, SM estimation advantageously provides a guaranteed solution. However, it does not give any precision about the belief degree and it is often criticized for the overestimation of its results. Actually, the two approaches have specific advantages and they may interact synergically. They complement more than they compete. In an estimation framework, the experimental conditions about noise and disturbances are usually properly modeled through appropriate probability distributions. However, other sources of uncertainty are not well-suited to stochastic modeling and are better represented with bounded uncertainties. This is the case of parameter uncertainties that generally arise from design tolerances and from aging. Hence, combining stochastic and bounded uncertainties may be an appropriate solution.

Motivated by the above observations, we consider the filtering problem for discrete time linear models with bounded uncertainties on parameters and gaussian measurement noise. In [9], the classical Kalman filter [10] has been extended to interval linear models. We build on this work and propose several operations that improve the filtering. In particular, the approach proposed in [9] does not provide guaranteed

results because it avoids interval matrix inversion. Our main contribution consist in proposing a method to solve the interval matrix inversion problem without loss of solutions while controlling the inherent pessimism of interval calculus. In particular the gain of the filter is obtained by a calculus based on the set inversion algorithm SIVIA (Set Inversion via Interval analysis) [11] which is combined with constraint propagation techniques.

The paper is organized as follows. Section II describes the problem formulation and the considered system. In Section III, some important concepts of interval analysis are introduced. An overview of the revised interval Kalman filtering algorithm is presented in Section IV, followed by the set of operations developed to control overestimation. The case study is then presented in section V with the results obtained for fault detection. Conclusions and future works are overviewed in Section VI.

## II. PROBLEM FORMULATION

We consider linear dynamic systems described by a set of state differences and observation equations (Kalman model [10]):

$$\begin{cases} x_{k+1} = Ax_k + Bu_k + w_k, \\ y_k = Cx_k + Du_k + v_k, k \in \mathbb{N} \end{cases} \quad (1)$$

where  $x_k \in \mathbb{R}^n$ ,  $y_k \in \mathbb{R}^m$  and  $u_k \in \mathbb{R}^p$  denote state, observation and input vectors, respectively. The matrices  $A, B, C$  and  $D$  are constant matrices such that  $A \in \mathbb{R}^{n \times n}$ ,  $B \in \mathbb{R}^{n \times p}$ ,  $C \in \mathbb{R}^{m \times n}$  and  $D \in \mathbb{R}^{m \times p}$ .  $\delta_{kl}$  being the Kronecker symbol,  $\{w_k\}$  and  $\{v_k\}$  are independent centered gaussian white noise sequences, with covariance matrices  $Q$  and  $R$  definite positive by definition, respectively:

$$\begin{aligned} E\{w_k, w_l\} &= Q\delta_{kl}, E\{v_k, v_l\} = R\delta_{kl}, \\ E\{w_k, v_l\} &= E\{w_k, x_0\} = E\{v_k, x_0\} = 0, \\ \forall (k, l) &\in \mathbb{N}^2 \end{aligned}$$

Based on the motivations reported in the introduction, we propose to combine two modeling paradigms : measurement and system noises are modeled, as usual, in a stochastic framework but parameter uncertainties are assumed to be bounded. This is achieved by considering that the matrices  $A, B, C$  and  $D$  of (1) become *interval matrices*, as defined in the following section.

## III. INTERVAL ANALYSIS

### A. Preamble

The key idea of interval analysis is to reason about intervals instead of real numbers [12] and [1]. The first motivation was to obtain guaranteed results from floating point algorithms and it was then extended to validated

<sup>1</sup> J. Xiong, C. Jaubertie, Travé-Massuyès and Françoise Le Gall are with CNRS, LAAS, 7 avenue du Colonel Roche, F-31400 Toulouse, France [jxiong,cjaubert,louise,legall@laas.fr](mailto:jxiong,cjaubert,louise,legall@laas.fr)

<sup>2</sup> J. Xiong is also with the Université de Toulouse, INP, LAAS, F-31400 Toulouse, France

<sup>3</sup> C. Jaubertie is also with the Université de Toulouse, UPS, LAAS, F-31400 Toulouse, France

<sup>4</sup> L. Travé-Massuyès and F. Le Gall are also with the Université de Toulouse, LAAS, F-31400 Toulouse, France

numerics [4]. A *guaranteed result* first means that the result set encloses the exact solution. Second, it also means that the algorithm is able to conclude about the existence or not of a solution in limited time or number of iterations.

### B. Main concepts

1) *Interval*: A real interval  $[p] = [\underline{p}, \bar{p}]$  is a closed and connected subset of  $\mathbb{R}$  where  $\underline{p}$  and  $\bar{p}$  represent the lower and upper bound of  $[p]$ , respectively. The *width* of an interval  $[p]$  is defined by  $w(p) = \bar{p} - \underline{p}$ , and its *midpoint* by  $m(p) = (\bar{p} + \underline{p})/2$ . If  $w(p) = 0$ , then  $[p]$  is degenerated and reduced to a real number.  $[p]$  is defined as positive (resp. negative), i.e.  $[p] \geq 0$  (resp.  $[p] \leq 0$ ), if  $\underline{p} \geq 0$  (resp.  $\bar{p} \leq 0$ ).

The set of all real intervals of  $\mathbb{R}$  is denoted  $\mathbb{IR}$ . Two intervals  $[p_1]$  and  $[p_2]$  are equal if and only if  $\underline{p}_1 = \underline{p}_2$  and  $\bar{p}_1 = \bar{p}_2$ . Real arithmetic operations have been extended to intervals [4]:

$$\circ \in \{+, -, *, /\}, [p_1] \circ [p_2] = \{x \circ y \mid x \in [p_1], y \in [p_2]\}.$$

An *interval vector* (or *box*)  $[\alpha]$  is a vector with interval components. It may equivalently be seen as a cartesian product of scalar intervals:

$$[\alpha] = [\alpha_1] \times [\alpha_2] \times \dots \times [\alpha_n].$$

An *interval matrix* is a matrix with interval components. The set of  $n$ -dimensional real interval vectors is denoted by  $\mathbb{IR}^n$  and the set of  $n \times m$  real interval matrices is denoted by  $\mathbb{IR}^{n \times m}$ . The width  $w(\cdot)$  of an interval vector (or of an interval matrix) is the maximum of the widths of its interval components. The midpoint  $m(\cdot)$  of an interval vector (resp. an interval matrix) is a vector (resp. a matrix) composed of the midpoints of its interval components.

Classical operations for interval vectors (resp. interval matrices) are direct extensions of the same operations for real vectors (resp. real matrices) [4].

2) *Inclusion function*: Given  $[X]$  a box of  $\mathbb{IR}^n$  and a function  $f$  from  $\mathbb{IR}^n$  to  $\mathbb{IR}^m$ , an *inclusion function* of  $f$  aims at getting a box containing the image of  $[X]$  by  $f$ . The *range* of  $f$  over  $[X]$  is given by:

$$f([X]) = \{f(x) \mid x \in [X]\}.$$

Then, the interval function  $[f]$  from  $\mathbb{IR}^n$  to  $\mathbb{IR}^m$  is an *inclusion function* for  $f$  if:

$$\forall [X] \in \mathbb{IR}^n, f([X]) \subset [f]([X]).$$

An inclusion function of  $f$  can be obtained by replacing each occurrence of a real variable by its corresponding interval and by replacing each standard function by its interval evaluation. Such a function is called the *natural inclusion function*. A function  $f$  generally has several inclusion functions, which depend on the syntax of  $f$ .

3) *Inclusion tests*: Given a subset  $S$  of  $\mathbb{R}^n$ , two tests are of particular interest: the test used to prove that all points in a given box  $[\alpha]$  satisfy a given property, i.e.  $[\alpha] \subset S$ , and the test that proves that none of them does, i.e.  $[\alpha] \cap S = \emptyset$ .

4) *Contraction*: Contraction is used to reduce a set  $[\alpha]$  to its intersection with respect to another set  $S$ . The contraction of  $[\alpha]$  with respect to  $S$  is a smaller set  $[\gamma]$  such that  $[\alpha] \cap S = [\gamma] \cap S$ . If  $S$  is the feasibility set of a problem and  $[\gamma]$  turns out to be empty, then the set  $[\alpha]$  does not contain the solution [11].

### C. SIVIA: Set Inversion Via Interval Analysis

Consider the problem of determining a solution set  $S$  for the unknown quantities  $p$  defined by:

$$\begin{aligned} S &= \{p \in P \mid \Phi(p) \in [\beta]\}, \\ &= \Phi^{-1}([\beta]) \cap P, \end{aligned} \quad (2)$$

where  $[\beta]$  is known a priori,  $P$  is an a priori search set for  $p$  and  $\Phi$  a nonlinear function not necessarily invertible in the classical sense. (2) involves computing the reciprocal image of  $\Phi$ . This can be solved using the recursive algorithm *SIVIA*, which explores all the search space without losing any solution. *SIVIA* provides a guaranteed enclosure of the solution set  $S$  as follows:

$$\underline{S} \subseteq S \subseteq \bar{S}. \quad (3)$$

The inner enclosure  $\underline{S}$  is composed of the boxes that have been proved feasible, i.e. such that  $\Phi([p]) \subseteq [\beta]$ . Conversely, if it can be proved that  $\Phi([p]) \cap [\beta] = \emptyset$ , then the box  $[p]$  is infeasible. Otherwise, no conclusion can be reached and the box  $[p]$  is said undetermined. The latter is then bisected in two sub-boxes that are tested until their size reaches a user-specified precision threshold  $\varepsilon > 0$ . Such a termination criterion ensures that *SIVIA* terminates after a finite number of iterations.

The number of bisections to be performed is generally prohibitive. Hence, recent algorithms take advantage of constraint propagation techniques to reduce the width of the boxes to be tested by *SIVIA* [13], [14]. In this context, the inclusion relations and the equations can be interpreted as the constraints of a Constraint Satisfaction Problem (*CSP*)  $H = (\mathcal{X}, \mathcal{D}, \mathcal{C})$  defined by:

- a set of variables  $\mathcal{X} = \{x_1, \dots, x_n\}$ ,
- a set of nonempty domains  $\mathcal{D} = \{D_1, \dots, D_n\}$  where  $D_i$  is the domain associated to the variable  $x_i$ ,
- a set of constraints  $\mathcal{C} = \{C_1, \dots, C_m\}$ , so that the variables involved in each constraint are defined in  $\mathcal{X}$ .

For solving a *CSP*, different types of so-called *contractors* can be used [1], [15]. Among the most well-known is the forward-backward contractor [16], which contracts the solution of the *CSP* by taking into account any one of the  $\{C_1, \dots, C_m\}$  constraint in isolation.

## IV. INTERVAL KALMAN FILTERING

Given the system (1), the conventional Kalman filter provides the minimum variance estimate  $\hat{x}_k$  of  $x_k$  and the associated covariance matrix  $P_k$  and we can write  $(\hat{x}_k, P_k) = \mathcal{K}(A, B, C, D, x_0, P_0, u_l, y_l)_{l < k}$ . When matrices  $A$ ,  $B$ ,  $C$ , and  $D$  are only known to belong to interval matrices  $[A]$ ,  $[B]$ ,  $[C]$  and  $[D]$ , respectively, both  $\hat{x}_k$  and  $P_k$  are tainted by bounded uncertainty. The interval Kalman filter aims at computing (an enclosure of) the set of all possible  $(\hat{x}_k, P_k)$ , i.e.:

$$\begin{aligned} \mathcal{X} &= ([\hat{x}_k], [P_k]) = \{(\hat{x}_k, P_k) \mid \exists A \in [A], B \in [B], C \in [C], \\ &D \in [D], (\hat{x}_k, P_k) = \mathcal{K}(A, B, C, D, x_0, P_0, u_l, y_l)_{l < k}\}. \end{aligned}$$

The algorithm proposed by [9] is based on interval conditional expectation for interval linear systems and has the same structure as the conventional Kalman filter algorithm.

Its drawback is that it does not guaranty to provide an enclosure of  $\mathcal{X}$ . In other words, some solutions are lost and the results are not guaranteed. This occurs because singularity problems in interval matrix inversion are avoided by taking the upper bound of the interval matrix to be inverted. We note this algorithm sIKF (sub-optimal interval Kalman filtering). In this paper, we propose the improved recursive estimator iIKF that includes recent advances in interval analysis and constraint propagation techniques.

#### A. Conventional Kalman filtering

There are several ways to deduce Kalman equations [10]. One can use mathematical curve-fitting function of data points from a least-squares approximation [17] or also use probabilist methods such as the Likelihood function to maximize the conditional probability of the state estimate from measurement incomes [18]. We consider the following:

- 1)  $\hat{x}_{k+1|k} \in \mathbb{R}^n$  the *a priori* state estimate vector at time  $k+1$  given state estimate at time  $k$ ,
- 2)  $\hat{x}_{k|k} \in \mathbb{R}^n$  the *a posteriori* state estimate vector at time  $k$  given observations at time  $k$ ,
- 3)  $P_{k+1|k} \in \mathbb{R}^{n \times n}$  the *a priori* error covariance matrix,
- 4)  $P_{k|k} \in \mathbb{R}^{n \times n}$  the *a posteriori* error covariance matrix.

$P_{l|k}$  is a key indicator that defines the accuracy of the state estimate :

$$P_{l|k} = E((x_l - \hat{x}_{l|k})(x_l - \hat{x}_{l|k})^T), l = k/k+1. \quad (4)$$

It is known that the Kalman filtering algorithm contains two steps for each iteration: a prediction step and a correction step [10].

#### B. Interval Kalman filtering: the iIKF algorithm

In addition to considering parameter bounded uncertainties through the interval matrices  $[A]$ ,  $[B]$ ,  $[C]$  and  $[D]$ , notice that  $x_{0|0}$ ,  $P_{0|0}$ ,  $u_k$ ,  $y_k$  could be boxes due to deterministic measurement errors and instrument precision. In the following, we evaluate the changes impacted by these assumptions on the different steps of the Kalman filtering algorithm.

1) *Estimation error covariance*: in the interval context, the estimation error covariance matrix is an interval matrix which can be rewritten as:

$$[P_{l|k}] \triangleq E((x_l - \hat{x}_{l|k})(x_l - \hat{x}_{l|k})^T), \quad (5)$$

where  $l = k$  or  $k+1$ .  $[P_{k|k}]$  is the estimation error covariance and  $[P_{k+1|k}]$  is the prediction error covariance. All elements on the diagonal of  $[P_{l|k}]$  are positive as they represent the variance of each state, thus the trace of  $[P_{l|k}]$  is positive. In the case of an interval matrix  $[P_{l|k}]$ , this constraint must also hold. If interval calculus pessimistically generates intervals containing non positive values, these are spurious and can be removed. Thus a first constraint is introduced :

$$[P_{l|k}]_{(i,i)} \geq 0, i = 1, 2, \dots, n. \quad (6)$$

2) *Prediction step*: The calculus of the *a priori* state estimate vector is directly inherited from the determinate model, while real variables are replaced by boxes :

$$[\hat{x}_{k+1|k}] = [A][\hat{x}_{k|k}] + [B][u_k]. \quad (7)$$

At the previous time  $k$ , the estimation error is characterized by  $[P_{k|k}]$ . The prediction model does not include noise so the estimation error should also be updated:

$$[\hat{P}_{k+1|k}] = [A][\hat{P}_{k|k}][A]^T + Q. \quad (8)$$

This equation can be interpreted as providing all possible *a priori* estimation error covariances between real state and *a priori* state estimate at time  $k+1$ . Accounting for (6), this leads to the following CSP :

$$\begin{cases} [\hat{P}_{k+1|k}] = [A][\hat{P}_{k|k}][A]^T + Q, \\ \mathcal{C} : [\hat{P}_{k+1|k}]_{(i,i)} \geq 0, i = 1, 2, \dots, n. \end{cases} \quad (9)$$

3) *Correction step*: from [9], the correction equation holds in the interval context:

$$[\hat{x}_{k+1|k+1}] = [\hat{x}_{k+1|k}] + [K_{k+1}]([y_{k+1}] - [\hat{y}_{k+1|k}]). \quad (10)$$

Intuitively,  $[K_{k+1}]$  aims to bring back the estimate enclosure around the real state while still retaining all the possible values corresponding to uncertainty. Equations (5) and (10) give the estimation error covariance expression. This manipulation is only valid when  $E\{v_k\} = 0$ :

$$\begin{aligned} [\hat{P}_{k+1|k+1}] &= [\hat{P}_{k+1|k}] - [K_{k+1}][C][\hat{P}_{k+1|k}] \\ &\quad - [\hat{P}_{k+1|k}][C^T][K_{k+1}]^T \\ &\quad + [K_{k+1}]\left([C][\hat{P}_{k+1|k}][C^T] + R\right)[K_{k+1}]^T. \end{aligned} \quad (11)$$

We want to find  $[K_{k+1}]$  such that it minimizes  $trace([\hat{P}_{k+1|k+1}])$ . Indeed, state variance, given by this matrix diagonal elements, is the value that indicates the estimation error:

$$\begin{aligned} \frac{\partial trace([\hat{P}_{k+1|k+1}])}{\partial [K_{k+1}]} &= -2[\hat{P}_{k+1|k}][C]^T \\ &\quad + 2[K_{k+1}]\left([C][\hat{P}_{k+1|k}][C]^T + R\right), \\ \frac{\partial^2 trace([\hat{P}_{k+1|k+1}])}{\partial [K_{k+1}]\partial [K_{k+1}]^T} &= 2\left([C][\hat{P}_{k+1|k}][C]^T + R\right). \end{aligned}$$

The second derivative is always positive in the conventional Kalman filter, which guarantees the existence of a solution to the minimization problem. In the interval context, this condition must be forced by a constraint of the same type as (6). From the first order derivative, we have:

$$[K_{k+1}] = [\hat{P}_{k+1|k}][C]^T \left([C][\hat{P}_{k+1|k}][C]^T + R\right)^{-1}. \quad (12)$$

Thus equations (11) and (12) give the estimation error covariance expression:

$$[\hat{P}_{k+1|k+1}] = (I_n - [K_{k+1}][C])[\hat{P}_{k+1|k}]. \quad (13)$$

4) *Algorithm loop*: Equations (7), (8), (12), (13) and (10) constitute a discrete interval Kalman filter algorithm.

*Initialization*:  $k=0$   $P_{0|0} = \text{Cov}\{x_0\}$ ,  $m_0 = E(x_0)$   
 $[x_0] \sim N(m_0, P_{0|0})$ ,

*Prediction*:  $[\hat{x}_{k+1|k}] = [A][\hat{x}_{k|k}] + [B][u_k]$ ,  
 $[\hat{P}_{k+1|k}] = [A][\hat{P}_{k|k}][A]^T + Q$ ,

*Correction*:

$[K_{k+1}] = [\hat{P}_{k+1|k}][C]^T \left( [C][\hat{P}_{k+1|k}][C]^T + R \right)^{-1}$ ,  
 $[\hat{P}_{k+1|k+1}] = (I_n - [K_{k+1}][C])[\hat{P}_{k+1|k}]$ ,  
 $[\hat{x}_{k+1|k+1}] = [\hat{x}_{k+1|k}] + [K_{k+1}]([y_{k+1}] - [\hat{y}_{k+1|k}])$ ,  
 $k = k + 1$ .

### C. Interval matrix inversion and overestimation control

A major issue is the pessimism introduced by interval arithmetic. Uncertainty is cumulated at each iteration and the interval matrix inversion involved in equation (12) is time consuming, sometimes divergent.

1) *Interval matrix inversion for gain value propagation*: Equation (12) involves the inversion of the interval matrix  $([C][P_{k+1|k}][C]^T + R)$ . The first problem refers to singularities, which means that the following condition should be fulfilled:

$$0 \notin \det([C][\hat{P}_{k+1|k}][C]^T + R).$$

Besides, the interval matrix inverse is obtained by approximation algorithms, like in [19], and is generally over estimated.

We propose an approach which uses the algorithm SIVIA. The idea is to solve the interval matrix inversion problem by a set of constraint propagation problems. Equation (12) is rewritten as:

$$[K_{k+1}] \left( [C][\hat{P}_{k+1|k}][C]^T + R \right) = [\hat{P}_{k+1|k}][C]^T.$$

Each component in matrix  $[K_{k+1}]$  is considered separately and the search space is the following cartesian product :

$$[K_{k+1}]_{1,1} \times [K_{k+1}]_{1,2} \times \dots \times [K_{k+1}]_{n,m}.$$

This search space is bisected and tested under SIVIA properly adapted to matrix operation. The result is a set of small boxes that satisfy Equation (14), each box providing a "small acceptable gain". The set of boxes is then injected at the correction step to update the covariance matrix and the state estimate vector. The final result is the hull of all covariance matrices and state estimate vectors corrected by each small gain.

2) *Constraint Propagation*: Constraint propagation is very useful to reduce the width of the boxes involved in a set of constraints [15]. In this work, we use the well-known forward-backward algorithm. The principle is to decompose the constraint equation  $f([x_1], \dots, [x_n]) = 0$  in a sequence of elementary operations of primitive functions like  $\{+, -, *, /\}$  and obtain a list of primitive constraints [20]. For example, consider the following equation:

$$[\hat{x}_{k+1|k+1}] = [\hat{x}_{k+1|k}] + [K_{k+1}]([y_{k+1}] - [C][\hat{x}_{k+1|k}]).$$

This equation can be decomposed, following the computation tree, into the following set of primitive constraints:

$$\begin{cases} [a_1] = [C][\hat{x}_{k+1|k}], \\ [a_2] = [y_{k+1}] - [a_1], \\ [a_3] = [K_{k+1}][a_2], \\ [\hat{x}_{k+1|k+1}] = [\hat{x}_{k+1|k}] + [a_3]. \end{cases}$$

In our problem, we want to contract  $\{[\hat{x}_{k+1|k+1}], [\hat{x}_{k+1|k}], [K_{k+1}]\}$  without changing  $\{[C], [y_{k+1}]\}$ , which are considered as inputs.

3) *Interval intersection rule*: As the associative law is no longer valid in interval arithmetic, we must redefine the product of three and four interval matrices [21]. This is the principle of the interval intersection rule.

$$\prod_{i=1}^n [M_i] \triangleq \left[ \left( \prod_{i=1}^{n-1} [M_i] \right) \cdot [M_n] \right] \cap \left[ [M_1] \cdot \left( \prod_{i=1}^{n-1} [M_{i+1}] \right) \right]. \quad (14)$$

where  $[M_1], \dots, [M_n]$  are interval matrices.

4) *Adaptative calibration*: When the interval matrix to be inverted is not regular, we must find a way to cut down the uncertainty accumulated by interval arithmetic. In this case, a calibration can be implemented to reset the iteration for limiting divergence [21]. We propose the following calibration mechanism:

$$[\hat{x}_k] \triangleq \hat{x}_k + [\zeta_k], [P_k] = [P_0]. \quad (15)$$

where  $\hat{x}_k$  is the conventional Kalman state estimate from the nominal system and  $[\zeta_k]$  is set from the state variances.

## V. FAULT DETECTION

Let us consider that the system (1) can suffer additive faults on sensors and let us adopt the single fault assumption. With the conventional Kalman filter, the principle of fault detection is to detect an abnormal change in the residual vector :

$$r_{k+1} = y_{k+1} - \hat{y}_{k+1|k}, \quad (16)$$

where  $y_{k+1}$  represents the measured output at time  $k + 1$ . When  $y_{k+1}$  is not faulty and without measurement noise, the residual is statistically reduced to zero. When a fault occurs, the residual vector is expected to become non null and at least one of its components indicates the fault.

Like for the conventional Kalman filter, we can define a confidence interval  $I_{[\hat{y}_{k+1|k}]}^i$  at 99.7% for each  $i$ th component  $[\hat{y}_{k+1|k}^i]$ ,  $i = 1, \dots, m$ , of  $[\hat{y}_{k+1|k}]$  used for fault detection thresholding:

$$\left[ \left( [\hat{y}_{k+1|k}]^i - q * [\sigma_{k+1}]^i \right), \left( [\hat{y}_{k+1|k}]^i + q * [\sigma_{k+1}]^i \right) \right], \quad (17)$$

where  $[\sigma_{k+1}]^i$  represents the standard deviation of  $[\hat{y}_{k+1|k}]^i$ . The confidence interval  $I_{[\hat{y}_{k+1|k}]}^i$  is guaranteed in the sense that it includes all the confidence intervals of the candidate values belonging to the interval output estimate. In this respect, it is quite conservative.

Fault detection is achieved, at time  $k + 1$ , by checking for consistency the confidence interval (at 99.7%)  $I_{y_{k+1}}^i$  of

$y_{k+1}^i$  against the confidence interval  $I_{[\hat{y}_{k+1|k}]}^i$  of  $[\hat{y}_{k+1|k}]$ , for  $i = 1, \dots, m$  [22]. Thus, we consider a binary variable  $\tau_k$  indexed by the time instant which infers:

$$\tau_k = \begin{cases} 1 & \text{if } \exists i \text{ s.t., } I_{y_{k+1}}^i \cap I_{[\hat{y}_{k+1|k}]}^i \neq \emptyset \\ 0 & \text{otherwise.} \end{cases} \quad (18)$$

When a fault occurs, it corrupts the output measures, which is reinjected in the iKF at the correction step. Hence, the output estimate is not reliable for representing the healthy system. Thus as soon as the fault is detected, the innovation step in the interval Kalman filter is halted until the system is restored healthy. A similar approach can be found in [23], [24], known as the *Semi-Closed Loop* (SCL) strategy.

## VI. EVALUATION

Let us consider an uncertain system described by the following equations:

$$\begin{cases} [x_{k+1}] = [A][x_k] + w_k, \\ [y_k] = [C][x_k] + v_k, k \in \mathbb{N} \end{cases} \quad (19)$$

$\{w_k\}$  and  $\{v_k\}$  are independent centered gaussian white noise sequences, whose covariance matrices, time-invariant and considered without uncertainty, are denoted  $Q$  and  $R$ . We suppose  $[A] = A + \Delta A$ ,  $[C] = C + \Delta C$  and:

$$A = \begin{bmatrix} 0.4 & 0.1 \\ -0.1 & 0.2 \end{bmatrix}, C = \begin{bmatrix} 0 & 1 \end{bmatrix}, Q = \begin{bmatrix} 10 & 0 \\ 0 & 10 \end{bmatrix}, R = 1.$$

$$\Delta A = \begin{bmatrix} [-0.1, 0.1] & [-0.15, 0.15] \\ 0 & [-0.25, 0.25] \end{bmatrix}, \Delta C = \begin{bmatrix} 0 & [-0.1, 0.1] \end{bmatrix},$$

$$E\{x_0\} = \begin{bmatrix} x_{01} \\ x_{02} \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \end{bmatrix},$$

First, we compare the results provided by three filters: the original interval Kalman filter (noted IKF) that does not include any overestimation control and makes use of Rohn's interval matrix inversion method[19], its sub-optimal version (sIKF) proposed by [9], and our improved filter (iKF). Let us define  $N$  the number of calibration times,  $O$  the number of times for which the interval state estimate does not contain the real state, and  $D$  the norm giving the distance between interval estimate bounds and the true value:

$$\begin{cases} D = \frac{\sqrt{\sum_{k=1}^K d([\hat{x}_k], x_k)^T d([\hat{x}_k], x_k)}}{\sqrt{\sum_{k=1}^K x_k^T x_k}}, \\ d([\hat{x}_k], x_k) = |[\hat{x}_k] - x_k| + |[\hat{x}_k] - x_k|, \end{cases} \quad (20)$$

where  $K$  represents the maximal iteration number.  $D$  and  $O$  indicate the efficiency of the algorithm. The precision threshold  $\epsilon$  used by SIVIA (expressed as a proportion of the original box size) is also analysed.

The simulations are run on the time stage  $[0,100]$  with the toolbox Intlab of Matlab [25]. The results of Table I are consistent with those shown in Figure 1 top and bottom. We can see that the original IKF has the largest  $D$  while the sub-optimal sIKF has the minimum  $D$  value, which is explained by the narrow bounds for the interval estimates. Nevertheless, since some solutions are lost, sIKF also obtains

Filter	$\epsilon$	N	O	D	t
IKF	-	20	14	575.38	0.83s
sIKF	-	0	56	0.85	0.75s
iKF	1	0	0	3.07	0.91s
	0.2	0	0	2.60	46s
	0.05	0	0	2.56	784s

TABLE I  
RESULTS FOR  $N$ ,  $O$  AND  $D$  USING IKF AND iKF WITH DIFFERENT  
BISECTION FACTORS  $\epsilon$  AND EXECUTION TIME  $t$ .

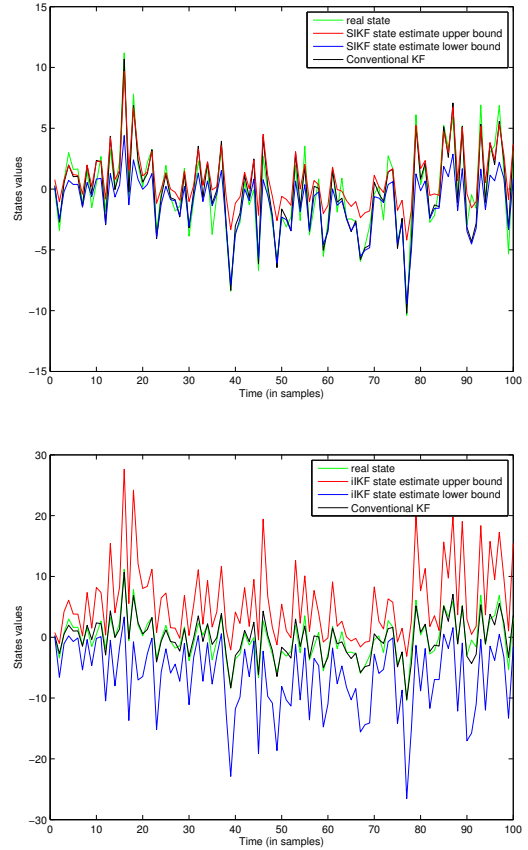


Fig. 1. Simulation results with the sub-optimal sIKF (top) and with iKF ( $\epsilon = 0.05$ ) (bottom).

the largest value of  $O$ : the real state is outside the estimated interval state half of the time.

By using iKF,  $D$  is larger than with sIKF, because it retains all solutions. We notice that the real state and the optimal estimate provided by the conventional Kalman filter are both always within the bounds of the iKF state estimate. The gain value propagated from SIVIA actually refines the interval estimation value, but the higher the precision  $\epsilon$ , the higher the computation time. Compared to the original IKF, the iKF prevents unnecessary recalibration due to divergent interval operations; compared to sIKF, iKF retains all the solutions consistent with the bounded error uncertainty. iKF hence represents a good compromise.

To test the efficiency of the proposed iKF based fault detection approach, a sensor fault affecting the system is introduced at time  $k = 50$ . This fault is persistent until time

$k = 80$ . The fault value is set to approximately 4 standard deviations. We use confidence intervals at 99,7% ( $q = 3$ ).

Figure 2 (top) provides the output prediction and the real measured output together with the fault indicator  $\tau_k$ .  $\tau_k$  rightly concludes to the occurrence of a fault at time  $k = 50$  and this fault is persistent until  $k = 80$ .

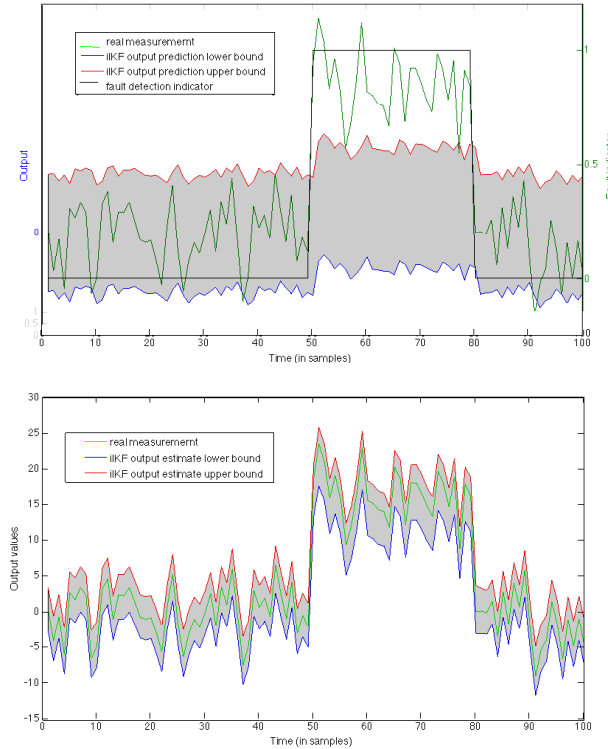


Fig. 2. Fault detection by using the iKF and the SCL strategy (top) – iKF output estimate in the faulty situation without the SCL strategy (bottom).

Figure 2 (bottom) clearly shows that the iKF output estimate produced without the SCL strategy “follows” the faulty measured output, preventing efficient fault detection. This is due to the correction in the innovation step of iKF. The *a posteriori* state estimate is indeed compensated according to the measurement, independently on whether it is faulty or not.

We should point out that in our scenario, no calibration takes place. But in more complex systems, it is likely to have singular interval matrices triggering the calibration.

## VII. CONCLUSIONS

The improved interval Kalman filter iKF proposed in this paper provides all the optimal estimates consistent with bounded errors and achieves good control of the pessimism inherent to interval analysis. Through a set of simulations, the advantages of the iKF with respect to previous versions are exhibited and the efficiency of the iKF based Semi-Closed Loop fault detection algorithm that we propose is clearly demonstrated. This work shows that the integration of statistical and bounded uncertainties in the same model

can be successfully achieved, which opens wide perspectives from a practical point of view. On the theoretical ground, this work calls for a unifying well-posed integrative theory.

## REFERENCES

- [1] L. Jaulin, M. Kieffer, O. Didrit, and E. Walter, *Applied interval analysis: with examples in parameter and state estimation, robust control and robotics*, 1st ed., ser. An emerging paradigm. Springer-Verlag, 2001.
- [2] P. Ribot, C. Jaubertie, and L. Trave-Massuyes, “State estimation by interval analysis for a nonlinear differential aerospace model,” in *Proceedings of European Control Conference*, Kos, Greece, 2007, pp. 4839–4844.
- [3] L. Jaulin, I. Braems, M. Kieffer, and E. Walter, “Nonlinear state estimation using forward-backward propagation of intervals in an algorithm,” *Scientific Computing, Validated Numerics, Interval Methods*, pp. 191–204, 2001.
- [4] R. E. Moore, *Interval analysis*. Prentice-Hall, Englewood Cliffs, Jan. 1966.
- [5] M. Kieffer, L. Jaulin, E. Walter, and D. Meizel, “Guaranteed mobile robot tracking using interval analysis,” in *Proceedings of MISC’99*, Girona, Spain, 1999, pp. 347–360.
- [6] M. Milanese and C. Novara, “Nonlinear Set Membership prediction of river flow,” *Systems & Control Letters*, vol. 53, no. 1, pp. 31–39, Sept. 2004.
- [7] S. Lescq, A. Barraud, and K. Dinh, “Numerical accurate computations for ellipsoidal state bounding,” in *Proceedings of MED’03*, Rhodes, Greece, 2003.
- [8] A. Ingimundarson, J. Manuel Bravo, V. Puig, T. Alamo, and P. Guerra, “Robust fault detection using zonotope-based set-membership consistency test,” *International Journal of Adaptive Control And Signal Processing*, vol. 23, no. 4, pp. 311–330, 2009.
- [9] G. Chen, J. Wang, and L. Shieh, “Interval kalman filtering,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. 33, no. 1, pp. 250–259, Jan. 1997.
- [10] R. Kalman, “A new approach to linear filtering and prediction problems,” *Journal of basic Engineering*, 1960.
- [11] L. Jaulin and E. Walter, “Set inversion via interval analysis for nonlinear bounded-error estimation,” *Automatica*, vol. 29, pp. 1053–1064, 1993.
- [12] R. Moore, *Automatic error analysis in digital computation*. Technical report LMSD-48421, Lockheed Missiles and Space Co, Palo Alto, CA, 1959.
- [13] D. Waltz, “Generating semantic descriptions from drawings of scenes with shadows,” *The Psychology of Computer Vision*, pp. 19–91, 1975.
- [14] E. Davis, “Constraint propagation with interval labels,” *Artificial Intelligence*, vol. 32(3), pp. 281–331, 1987.
- [15] G. Chabert and L. Jaulin, “Contractor programming,” *Artificial Intelligence*, vol. 173, pp. 1079–1100, 2009.
- [16] F. Benhamou, F. Goualard, F. Granvilliers, and J. Puget, “Revising hull and box consistency,” in *Proceedings of the International Conference on Logic Programming*, Las Cruces, New Mexico, 1999, pp. 230–244.
- [17] G. Welch and G. Bishop, “An Introduction to the Kalman Filter,” in *SIGGRAPH*, Los Angeles, California USA, 2001.
- [18] C. Masreliez and R. Martin, “Robust bayesian estimation for the linear model and robustifying the Kalman filter,” *IEEE Transactions on Automatic Control*, vol. 22, no. 3, 1977.
- [19] J. Rohn, “Inverse interval matrix,” *SIAM Journal on Numerical Analysis*, vol. 30, no. 3, pp. 864–870, June 1993.
- [20] O. Lhomme, R. La Chanterrie, A. Gotlieb, and M. Rueher, “Boosting the interval narrowing algorithm,” in *Proceedings of the 1996 Joint International Conference and Symposium on Logic Programming*. Bonn, Germany: The MIT Press, 1996, pp. 378–392.
- [21] B. Li, C. Li, and J. Si, “Interval recursive least-squares filtering with applications to video target tracking,” *Optical Engineering*, vol. 47, no. 10, p. 106401, 2008.
- [22] O. Adrot, H. Janati-Idrissi, and D. Maquin, “Fault detection based on interval analysis,” in *15th Triennial World Congress*, 2002.
- [23] E. Benazera and L. Trave-Massuyes, “A diagnosis driven self-reconfigurable filter,” *18th International Workshop on Principles of Diagnosis (DX-07)*, pp. 21–28, 2007.
- [24] L. Trave-Massuyes, T. Escobet, R. Pons, and S. Tornil, “The Ca-En diagnosis system and its automatic modelling method,” *Computacin i Sistemes, Revista Iberoamericana de Computacin*, vol. 5, no. 2, pp. 21–28, 2001.
- [25] S. Rump, “INTLAB - INTerval LABoratory,” in *Developments in Reliable Computing*, T. Csendes, Ed. Dordrecht: Kluwer Academic Publishers, 1999, pp. 77–104, <http://www.ti3.tu-harburg.de/rump/>.