2013 European Control Conference (ECC)
July 17-19, 2013, Zürich, Switzerland.

Distributed Fault Detection using Sensor Networks and Pareto Estimation

Francesca Boem, Yuzhe Xu, Carlo Fischione and Thomas Parisini

Abstract— **In this paper, a preliminary novel distributed fault detection architecture for dynamic systems using sensor networks and a distributed estimation method based on Pareto optimization is proposed. The goal is to monitor large-scale or distributed systems by using a sensor network where each node acts as a *local estimation agent* without centralized coordination. Probabilistic detection thresholds related to a given rate of false alarms are derived in several different scenarios as far as the measurement pattern and the nominal dynamics is concerned. Preliminary simulation results show the effectiveness of the proposed fault detection methodology.**

I. Introduction

The scientific interest towards large-scale and/or distributed systems has become important during last decades. One of the open challenges is the quantitative monitoring of these systems without the help of a centralized coordination agent. The conditions in which these systems operate rely on the possibility of system faults represented by critical changes in the system dynamics from the designed behavior. Therefore, fault diagnosis, which consists in the detection, isolation and identification of the fault, is a key requirement in the design of reliable modern systems (see [1] and the references cited therein). The centralized solution of this problem suffers from scalability issues and sometimes it is even not possible when dealing with large-scale systems. An interesting solution for distributed fault diagnosis can be realized by exploiting sensor networks. Some works exist addressing the problem of fault diagnosis of sensor networks, such as sensor fault detecting, packet losses and energy consumption monitoring ([2], [3], [4]), but fewer are the works using sensor networks as a tool for dynamical systems monitoring. Classical methods for quantitative fault diagnosis in the state of the art deal with the use of model-based analytical redundancy techniques. A lot of these works require the centralized collection of the information obtained from the sensor devices. Some exceptions are [5], [6], [7] and other works dealing with discrete-event systems ([8], [9]). Even if methods for distributed estimation already exist (see [10] for a survey and [11], [12] as examples), links between distributed estimation and distributed monitoring are still lacking. An exception is [13], where a distributed fault detection and isolation technique is designed, relying on decentralized Kalman state-estimation method. In fact, an important branch of research on distributed estimation is represented by distributed Kalman filters ([14]) and their combination with the diffusion mechanism ([15], [16]).

In this paper, a preliminary study is proposed to show that the distributed estimation method presented in [17] for sensor networks can be useful for monitoring and fault detection purposes. More specifically, each sensor node is a *local estimation agent* and a distributed estimation filter is designed based on Pareto optimization following the very recent approach proposed in [17]. The use of central coordination is not required. The advantages of using the distributed estimation method based on Pareto optimization are that it guarantees to minimize at the same time both bias and variance of the estimation error and, thanks to the way it is designed, it allows to compute mean and variance of the estimation error, thus allowing to obtain suitable thresholds for some suitable residual signals with known confidence levels and definable rate of false alarms. Moreover, since the distributed Pareto estimator does not require the model of the system that generates the signal to track, it can be applied to uncertain and even nonlinear systems, although, in this preliminary work, we consider mainly the linear case. We do not assume any topology of the network, unlike previous works that require an all-to-all topology (see [18] as example). The paper is organized as follows. After formulating the problem under concern in the next section, Section III presents the distributed estimation methodology, whereas in Section IV the fault detection algorithm exploiting the distributed estimation tool is presented. Some extensions are considered in Section V and simulation results are given in Section VI. Finally, some concluding remarks in Section VII.

II. Problem formulation

Let us first introduce a few useful notations that will be used throughout the paper. By $|\cdot|$ we denote the cardinality of the argument and by $\|\cdot\|$ the spectral norm of a matrix. Given a stochastic variable x, we represent as $\mathbb{E}x$ its expected value. Finally, by $\mathbf{1}$ and I we denote the vector $(1,\ldots,1)^\top$ and the identity matrix, respectively.

Let us consider the model of the plant to be monitored:

$$x(t+1) = Ax(t) + \xi(t) + \beta(t-t_0)\phi(x(t),u(t),t), \quad (1)$$

where $x \in \mathbb{R}^n$ denotes the state vector, $\xi \in \mathbb{R}^n$ is a noise with mean μ_ξ and covariance σ_ξ^2, representing the disturbances affecting the state vector, and $\beta(t-t_0)\phi(x(t),u(t),t)$ describes the dynamics of an additive fault occurring at the

F. Boem is with the Dept. of Engineering and Architecture, University of Trieste, Italy. (francesca.boem@phd.units.it)

Y. Xu and C. Fischione are with Electrical Engineering and ACCESS Linnaeus Center, KTH Royal Institute of Technology, Stockholm, Sweden. (yuzhe@kth.se, carlofi@kth.se)

T. Parisini is with the Dept. of Electrical and Electronic Engineering at the Imperial College London, UK, and also with the Dept. of Engineering and Architecture at the University of Trieste, Italy (t.parisini@gmail.com)

978-3-033-03962-9/©2013 EUCA

unknown time t_0, with $\beta(t - t_0)$ being a scalar function representing time profile of the fault (modeling both abrupt and incipient faults [7]). In healthy conditions, the model is

$$x(t+1) = Ax(t) + \xi(t).$$

A *sensor network* is available to monitor system (1) and it has to be designed to perform a distributed fault detection task. In particular, every node of the network observes the same state and exchanges measurements and state estimations with neighboring nodes in order to reach a common fault detection decision. The communication network is modeled as an undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{N}_i = \{j \in \mathcal{V} : (j,i) \in \mathcal{E}\} \cup \{i\}$ is the set of neighbors of node $i \in \mathcal{V}$ plus the node itself. It is assumed that there are no communication delays. We want to show that the distributed estimation method presented in [17], minimizing bias and variance of the estimation error simultaneously, can be extended to carry out diagnosis tasks. In the following, we propose the fault detection architecture for a linear case in which all components of the state vector are available for measurement (corrupted by noise). Later on, the extension to more general scenarios will be briefly addressed (e.g., not all components of the state are accessible, sensors do not observe the same state, the system is not linear, etc.).

III. Distributed model estimation

Let us consider a sensor network made of N nodes. For each node i, with $i = 1, \ldots, N$, we have

$$y_i(t) = x(t) + v_i(t), \qquad (2)$$

where y_i is the measurements vector obtained at node i and v_i is the zero-mean measurement noise. In this section, for the sake of simplicity, we assume that all sensors can measure all the state components. In Section V, more general measurement models are proposed.

Each node uses two steps to estimate the state variables: firstly, it communicates with neighboring nodes and computes an “a-posteriori” estimate using the Pareto optimization architecture. For each k-th component, with $k = 1, \ldots, n$,

$$\bar{x}_i^{(k)}(t) = \sum_{j \in \mathcal{N}_i^k} l_{k,i,j}(t) \bar{x}_j^{(k)}(t-1) + \sum_{j \in \mathcal{N}_i^k} h_{k,i,j}(t) y_j^{(k)}(t), \qquad (3)$$

where $l_{k,i,j}(t)$ and $h_{k,i,j}(t)$ are the time-varying weights of the filter designed in [17], $\mathcal{N}_i^k$ is the set of neighbors of node i measuring the variable $x^{(k)}$ plus the node i itself and with $\bar{x}_i^{(k)}$ we denote the k-th component of the state estimate computed by node i. The optimal values for $l_{k,i,j}$ and $h_{k,i,j}$ are obtained by Pareto optimization that minimizes both the bias and variance of the estimation errors. In Section III-A, the Pareto estimation architecture is discussed in some detail.

Subsequently, the second step consists on the “a-priori” estimate, computed on the basis of the model of the system:

$$\hat{x}_i(t+1) = A\bar{x}_i(t) + \lambda_i(\hat{x}_i(t) - \bar{x}_i(t)),$$

with $0 < \lambda_i < 1$. Thanks to the communication with neighboring nodes in the first step, each node can reduce its measurement uncertainty. Therefore, it is convenient to use filtered estimates instead of measurements, because, in this way, we can define less conservative thresholds for fault detection purposes, as explained in Section IV. In the following subsection, we go into more detail describing the distributed Pareto estimator.

A. The Pareto estimator

The distributed estimation architecture for sensor networks is introduced in [17]. For the sake of simplicity, here we consider Eq. (2) for the k-th state component. Let us consider $N_k > 1$ sensor nodes, measuring a common signal $x^{(k)}(t)$ affected by additive noise:

$$y_i^{(k)}(t) = x^{(k)}(t) + v_i^{(k)}(t), \quad i = 1, \ldots, N_k,$$

where $v_i(t)$ is a zero-mean white noise. Collecting the variables in vectors, it is possible to write:

$$y^{(k)}(t) = x^{(k)}(t)\mathbf{1} + v^{(k)}(t).$$

We assume the covariance matrix Σ_k of $v^{(k)}$ to be diagonal. For the sake of simplicity and without loss of generality, here we define $\Sigma_k = \sigma_k^2 I$, but all the results can be extended to the more general and realistic case $\Sigma_k = \mathrm{diag}([\sigma_{k1}^2, ..., \sigma_{kN_k}^2])$ in a simple way (see [17] for details). As already expressed in Eq. (3), each node i computes an estimate $\bar{x}_i^{(k)}(t)$ of the signal $x^{(k)}(t)$ as a linear combination of neighboring estimates and measurements. In vector notation, we have:

$$\bar{x}^{(k)}(t) = L_k(t)\bar{x}^{(k)}(t-1) + H_k(t)y^{(k)}(t), \qquad (4)$$

where L_k and H_k are the matrices containing time-varying weights of the filter. The algorithm is initialized with $\bar{x}_j(0) = y_i^{(k)}(0)$, $j \in \mathcal{N}_i^k$. In [17], the optimal values of these weights are derived as the solution of a multi-objective optimization problem that minimizes at the same time both the bias and the variance of the estimation error $e_i^{(k)}(t) = \bar{x}_i^{(k)}(t) - x^{(k)}(t)$ by using a Pareto optimization framework:

$$\min_{\kappa_{i,k}^\top, \eta_{i,k}^\top} \ (1 - \rho_{i,k})V_{i,k} + \rho_{i,k} B_{i,k}^2$$
$$\text{s. t. } (\kappa_{i,k}(t) + \eta_{i,k}(t))^\top \mathbf{1} = 1,$$

where $0 \le \rho_{i,k} \le 1$, $B_{i,k} = \mathbb{E}e_i^{(k)}(t)$ is the bias term of the estimation error, $V_{i,k} = \mathbb{E}(e_i^{(k)}(t) - \mathbb{E}e_i^{(k)}(t))^2$ is the variance term and we introduced $\kappa_{i,k}^\top(t)$ and $\eta_{i,k}^\top(t)$ that correspond to the non–zero elements of the i-th row of matrices $L_k(t)$ and $H_k(t)$ respectively. The constraint is a local condition needed to guarantee the convergence properties of the centralized estimation error, that are derived in [11] and that hold likewise in our case. It is equivalent to the following global assumption:

Assumption 1: We assume that $(L_k(t) + H_k(t))\mathbf{1} = \mathbf{1}$.

We derived the expressions for the bias and the estimation error variance that can be computed in a distributed way:

$$\mathbb{E}e_i^{(k)}(t) = \kappa_{i,k}^\top(t)\mathbb{E}\epsilon_{i,k}(t-1) - \kappa_{i,k}^\top(t)\delta^{(k)}(t)\mathbf{1},$$

where $\epsilon_{i,k}(t)$ collects the estimation errors available at node i for the k-th state component, ordered according to their indexes: $\epsilon_{i,k} = (e_{i_1}^{(k)}, \dots, e_{i_{M_i^k}}^{(k)})^\top$, $\quad i_1 < \dots < i_{M_i^k}$, with $M_i^k = \left|\mathcal{N}_i^k\right|$ denoting the number of neighbors of i plus i itself. Then, we compute the variance of the estimation error:

$$\begin{aligned}\mathbb{E}(e_i^{(k)}(t) - \mathbb{E}e_i^{(k)}(t))^2 = \kappa_{i,k}^\top(t)\Gamma_{i,k}(t-1)\kappa_{i,k}(t)\\ + \sigma_{i,k}^2\eta_{i,k}^\top(t)\eta_{i,k}(t),\end{aligned}$$

where $\Gamma_{i,k}(t) = \mathbb{E}(\epsilon_{i,k}(t) - \mathbb{E}\epsilon_{i,k}(t))(\epsilon_{i,k}(t) - \mathbb{E}\epsilon_{i,k}(t))^\top$ is the error covariance matrix. The obtained optimization problem can be solved in a distributed way for each node i by using Karush-Kuhn-Tucker conditions, deriving the following solution for a given $\rho_{i,k}$:

$$\kappa_{i,k}(t) = \frac{(1-\rho_{i,k})\sigma_k^2\Theta_{i,k}^{-1}\mathbf{1}}{(1-\rho_{i,k})\sigma_k^2\mathbf{1}^\top\Theta_{i,k}^{-1}\mathbf{1} + M_i^k}, \tag{5}$$

$$\eta_{i,k}(t) = \frac{\mathbf{1}}{(1-\rho_{i,k})\sigma_k^2\mathbf{1}^\top\Theta_{i,k}^{-1}\mathbf{1} + M_i^k}. \tag{6}$$

where $\Theta_{i,k}(t) = (1-\rho_{i,k})\Gamma_{i,k}(t-1) + \rho_{i,k}\Lambda_{i,k}(t)$, with $\Lambda_{i,k}(t) = (\mathbb{E}\epsilon_{i,k}(t-1) - \delta(t)\mathbf{1})(\mathbb{E}\epsilon_{i,k}(t-1) - \delta^{(k)}(t)\mathbf{1})^\top$.

In the literature, the best value of $\rho_{i,k}$ is obtained by building the Pareto trade–off curve and selecting the "knee–point" of this curve, that is, choosing $\rho_{i,k}^*$ such that $B_{i,k}$ and $V_{i,k}$, computed with the values $\kappa_{i,k}^{\top *}(\rho_{i,k}^*)$ and $\eta_{i,k}^{\top *}(\rho_{i,k}^*)$, are $V_{i,k} = B_{i,k}^2$.We chose to compute it locally using the Nelder-Mead simplex algorithm (see [19]). As an alternative, the Pareto parameter could be determined in order to define an appropriate bound on the bias. In [17] it is shown how to define a bound on the bias by appropriately setting $\rho_{i,k}$.

B. Estimation error analysis

In the previous subsections, we introduced the two-steps estimation method. Here, we analyze dynamics and features of the estimation error. By defining $e_i(t) = \bar{x}_i(t) - x(t)$ the estimation error vector of the first estimation phase, which collects the components of $e_i^{(k)}(t)$ with $k = 1, \dots, n$, we can analyze the dynamics of a residual:

$$\begin{aligned}r_i(t+1) &= \bar{x}_i(t+1) - \hat{x}_i(t+1)\\ &= \bar{x}_i(t+1) - x(t+1) + x(t+1) - \hat{x}_i(t+1)\\ &= e_i(t+1) + Ax(t) + \xi(t) - A\bar{x}_i(t) - \lambda_i(\hat{x}_i(t) - \bar{x}_i(t))\\ &= \lambda_i r_i(t) + e_i(t+1) - Ae_i(t) + \xi(t).\end{aligned}$$

Now, we derive mean and variance of the random variable $\chi_i(t) = e_i(t) - Ae_i(t-1) + \xi(t-1)$. The mean is given by

$$\begin{aligned}\mu_{\chi_i}(t) = \mathbb{E}\chi_i(t) = \mathbb{E}e_i(t) - A\mathbb{E}e_i(t-1) + \mathbb{E}\xi(t-1)\\ = B_i(t) - A\,B_i(t-1) + \mu_\xi,\end{aligned}$$

where $B_i(t)$ collects the components $B_{i,k}$ of the bias that are computed at each step by the Pareto estimator. As far as variance is concerned, it is necessary to distinguish between two cases: if the variables are independent, the variance of their sum (or difference) is the sum of their variances; otherwise, if the variables are correlated, then the variance of their sum is the sum of all their covariances: $\mathrm{Var}(X_i + X_j) = \mathrm{Var}(X_i) + \mathrm{Var}(X_j) + 2\mathrm{Cov}(X_i, X_j)$. We can assume that e_i and ξ_i are independent, since e_i is obtained based on the measurements only, while $e_i(t+1)$ is correlated with $e_i(t)$. Therefore, we can compute the variance of $\chi_i(t)$ as:

$$\sigma_{\chi_i}^2(t) = V_i(t) + A^2V_i(t-1) + 2\mathrm{Cov}(e_i(t), e_i(t-1)) + \sigma_\xi^2\,,$$

where we denote by A^2 the matrix collecting the element by element square and $B_i(t)$ collects the components $B_{i,k}$ of the variance that are computed at each step by the Pareto estimator. $\mathrm{Cov}(e_i(t), e_i(t-1))$ is again a column vector: each k-th component can be computed as $\kappa_{i,k}(t)^\top V^k$, where V^k is the column vector collecting the components $V_{i,k}$, with $i = 1, \dots, N$. Since $\kappa_i(t)$ has non null elements only in correspondence of the neighbors of i, only this components have to be communicated. This expression can be derived considering the vector notation and obtaining the global estimation error for the k-th state component $e^{(k)}(t) = \bar{x}^{(k)}(t) - x^{(k)}\mathbf{1} \in \mathbb{R}^N$:

$$\begin{aligned}e^{(k)}(t) =& L_k(t)e^{(k)}(t-1) + x^{(k)}(t)(L_k(t) + H_k(t) - I)\mathbf{1}\\ &- \delta^{(k)}(t)L_k(t)\mathbf{1} + H_k(t)v^{(k)}(t).\end{aligned} \tag{7}$$

Since

$$\begin{aligned}&\mathrm{Cov}(e^{(k)}(t), e^{(k)}(t-1))\\ &\quad = \mathbb{E}[(e^{(k)}(t) - \mathbb{E}e^{(k)}(t))(e^{(k)}(t-1) - \mathbb{E}e^{(k)}(t-1))]\end{aligned}$$

and remembering that $v^{(k)}(t)$ has zero mean and is independent of $e^{(k)}(t-1)$, we obtain

$$\begin{aligned}&\mathrm{Cov}(e^{(k)}(t), e^{(k)}(t-1))\\ &\qquad\qquad = L_k(t)\mathbb{E}[(e^{(k)}(t-1) - \mathbb{E}e^{(k)}(t-1))^2].\end{aligned}$$

It is sufficient to select the i-th row to determine the k-th element of $\mathrm{Cov}(e_i(t), e_i(t-1))$. In the next section, we will see how the computed quantities can be useful in order to derive some bounding thresholds for the estimation residual.

IV. Fault Detection by sensor networks based distributed estimation

In this section, the previous results are exploited for fault detection purposes. Now, we are going to determine how to bound the residual. If we do not know the distribution of the random variable $\chi_i(t)$, we can use the Chebyshev inequalities (without any assumption on the distribution):

$$\begin{aligned}\Pr(\mu(\chi_i)^{(k)} - \alpha\sigma(\chi_i)^{(k)} \le \chi_i^{(k)} \le \mu(\chi_i)^{(k)} + \alpha\sigma(\chi_i)^{(k)})\\ \ge 1 - \frac{1}{\alpha^2}.\end{aligned} \tag{8}$$

It is possible to find better results if we assume to know the distribution of $v_i(t)$ and $\xi_i(t)$. As example, let us assume v_i and ξ_i to be normally distributed. Then, it is possible to show that also each k-th component of $e_i(t)$ (which is the i-th component of $e^{(k)}(t)$, Eq. (7), where the only stochastic variable is $v^{(k)}(t)$) has a Gaussian distribution since it is a linear function of Gaussian stochastic variables. In the normal case, the percentages are: $68,3\%$ with $\alpha = 1$; $95,5\%$ with $\alpha = 2$; $99,0\%$ with $\alpha = 2,58$; $99,7\%$ with

$\alpha = 3$. Therefore, we can define time-varying upper and lower thresholds for the distributed estimation residuals:

$$\bar{r}_i^{(k)+}(t+1) = \lambda_i \bar{r}_i^{(k)+}(t) + \mu_{\chi_i}^{(k)}(t+1) + \alpha\sigma_{\chi_i}^{(k)}(t)\,,$$

$$\bar{r}_i^{(k)-}(t+1) = \lambda_i \bar{r}_i^{(k)-}(t) + \mu_{\chi_i}^{(k)}(t+1) - \alpha\sigma_{\chi_i}^{(k)}(t)\,.$$

In this way, it is possible to define a "α-tube" to which the residual belongs to (in a probabilistic sense) in healthy conditions, with a certain probability depending on α value:

$$\bar{r}_i^{(k)-} \leq r_i^{(k)}(t) \leq \bar{r}_i^{(k)+}(t)\,.$$

If the computed residual is outside the "α-tube", then we can conclude that a fault has occurred. The rate of false-alarms depends on the chosen value of α.

V. Extensions

In this section, we extend the simple methodology presented in the previous sections to more complex scenarios.

A. Case 1: not directly measurable state

We consider the case that all the sensors measure the same variables $C^{(k)}x(t)$, with $k = 1, \ldots, p$:

$$y_i(t) = Cx(t) + v_i(t),$$

where $y_i \in \mathbb{R}^p$ are the measurements of the i-th sensor node. We assume C is a invertible matrix. By means of the Pareto estimation method, sensors exchange measurements $y_i^{(k)}$ and estimates $\bar{y}_i^{(k)}$ of $C^{(k)}x(t)$. The first estimation step is:

$$\bar{y}_i^{(k)}(t) = \sum_{j\in\mathcal{N}_i^k} l_{k,i,j}(t)\bar{y}_j^{(k)}(t-1) + \sum_{j\in\mathcal{N}_i^k} h_{k,i,j}(t) y_j^{(k)}(t),$$

following the same procedure presented in Section III-A. From this it is possible to derive:

$$\bar{x}_i(t) = C^{-1}\bar{y}_i(t).$$

The second step is again the model-based estimate:

$$\hat{x}_i(t+1) = A\bar{x}_i(t) + \lambda_i(\hat{x}_i(t) - \bar{x}_i(t)),$$

Estimation error and threshold can be computed following the same steps as in the previous case (Section III-B - IV).

B. Case 2: measurements of different variables

In this case, sensors may measure different variables:

$$y_i(t) = C_i x(t) + v_i(t).$$

We define for each k-th measured variable a communication network $\mathcal{G}^k = (\mathcal{V}^k, \mathcal{E}^k)$, that connects all the nodes measuring variable k. Each node i communicates with all the neighboring nodes $j \in \mathcal{N}_i^k$ and derives from measurements the same state variables: $\tilde{x}_i^{(k)}(t) = C_i^{-1(k)} y_i(t)$. We assume C_i to be invertible and observability of the system. The distributed Pareto estimator is applied to $\tilde{x}_i = x(t) + C_i^{-1} v_i(t)$, obtaining the state estimate:

$$\bar{x}_i^{(k)}(t) = \sum_{j\in\mathcal{N}_i^k} l_{k,i,j}(t)\bar{x}_j^{(k)}(t-1) + \sum_{j\in\mathcal{N}_i} h_{k,i,j}(t)\tilde{x}_j^{(k)}(t).$$

The second step can be applied in the same way as before:

$$\hat{x}_i(t+1) = A\bar{x}_i(t) + \lambda_i(\hat{x}_i(t) - \bar{x}_i(t))$$

and the new thresholds can be obtained by simple algebra.

C. Case 3: general non-linear uncertain model

Since the filter presented in [17] does not need to assume the model of the system, we can use it also in the case of a general non-linear uncertain model. This comes at the cost of more conservative results and stronger assumptions. Let us consider the following non-linear system:

$$x(t+1) = f(x(t), u(t)) + \eta(x(t), u(t), t)$$

$$y_i(t) = x(t) + v_i(t),$$

where f describes the known nominal dynamics of the model and η represents model uncertainty.

Assumption 2: The uncertainty function is bounded by a known and bounded function: $|\eta(x(t), u(t), t)| \leq \bar{\eta}(x(t), u(t), t)\ \ \forall\,(x, u, t)$.

As in the simpler case, measurements are first filtered by means of the Pareto estimator and $\bar{x}_i$ is computed (Eq. (3)). Subsequently, the model-based estimator is used:

$$\hat{x}_i(t+1) = f(\bar{x}_i(t), u(t)) + \lambda_i(\hat{x}_i(t) - \bar{x}_i(t)).$$

The estimation residual can be computed:

$$r_i(t+1) = \lambda_i r_i(t) + e_i(t+1) + f(x,u) - f(\bar{x},u) + \eta(t).$$

This residual can be bounded as follows:

$$|r_i(t+1)| \leq \lambda_i\,|r_i(t)| + |e_i(t+1)| + |f(x,u) - f(\bar{x}_i,u)| + |\eta(t)|\,.$$

A threshold can then be derived for $|r_i(t)|$:

$$\bar{r}_i(t+1) = \lambda_i\bar{r}_i(t) + \bar{e}_i(t+1) + \bar{\Delta} f(t) + \bar{\eta}(t), \qquad (9)$$

where $\bar{\eta}(t)$ is known by assumption, $\bar{e}_i(t+1)$ can be bound similarly as in the simple case addressed in Section IV since we know its mean and variance,

$$\bar{\Delta} f(t) = \max_{x\in S}[f(x,u) - f(\bar{x},u)],$$

where S is the set where $x - \bar{x}_i = e_i$ is such that

$$\mu_{e_i} - \alpha\sigma_{e_i} \leq e_i \leq \mu_{e_i} + \alpha\sigma_{e_i}\,.$$

Eq. (9) is a conservative threshold and by choosing an appropriate value of α guarantees that no false alarms occur.

VI. Simulation results

In this section, some preliminary simulation results are presented. In [17], it is explained how to implement the Pareto estimator and, more specifically, how to derive the estimates of the needed quantities; moreover, a detailed computation complexity analysis of the distributed estimation algorithm is presented. As in [13], in order to show the effectiveness of the proposed approach, we consider a network of $N = 20$ sensor nodes monitoring a system representing a moving object on a plane. We decided to use the same

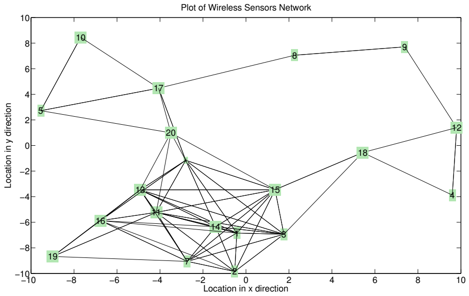

Fig. 1. The considered communication network.

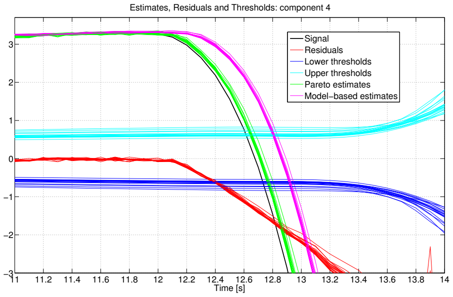

Fig. 4. Estimates, Residuals and Thresholds for state component 4.

simulation example in order to allow a qualitative comparison to the performances of the method based on distributed Kalman filters. The network is obtained by distributing the nodes randomly over a squared area of size $N/2$ and by letting two nodes communicate if their relative distance is lower than $\sqrt{N}$. The considered network can be seen in Fig. 1. The dynamics of the nominal system can be represented as: $x(t+1) = Ax(t) + \xi(t)$, where

$$A = \begin{bmatrix} 1 & \delta & 0 & 0 & 0 & 0 \\ 0 & 1-\frac{\delta\mu}{m} & \frac{\delta}{m} & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & \delta & 0 \\ 0 & 0 & 0 & 0 & 1-\frac{\delta\mu}{m} & \frac{\delta}{m} \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix},$$

where $\delta = 0.1s$ is the sampling time, $m = 0.75kg$ is the mass of the vehicle and $\mu = 0.15$ is the friction coefficient; the process noise $\xi(t)$ is a zero-mean Gaussian noise with $\sigma_\xi^2 = 10^{-4}\text{diag}(1, \ldots, 1)$. As in [13], the state vector is initialized as $x(0) = \text{col}[0, 0, 0.1, 0, 0, 0.1]$ and two types of additive faults are considered. The first consists in the sign inversion of the force given by one of the actuators along the vertical direction; the second causes the actuator governing the movement along the horizontal direction to get stuck. Here we present the case that fault 1 occurs at time $t_0 = 12s$. After t_0, $x(t+1) = Ax(t) + \xi(t) + \Phi_1 x(t)$, with

$$\Phi_1 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -\delta & 0 & 0 & 0 & 0 \\ 0 & -\frac{\delta}{m} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}.$$

It is worth noting that the fault function could be nonlinear since our approach does not require any assumption. The measurement noise is $v_i \sim WGN(0, \Sigma)$, where $\Sigma = 10^{-1}\text{diag}(1, 10, 1, 10, 1, 10)$ is the same for all sensor nodes. Fig. 2 shows the signals and the measurements collected by sensor nodes. The parameter α is set to 1. There is a trade-off between the reduction of false alarms rate and the detection time. Smaller is α, sooner the fault will be detected, but the false alarm rate increases. On the other hand, if α grows, the false alarm rate is smaller, but the fault is detected later. In Fig. 3, the fault is detected for the fourth and fifth state component. In Fig. 4, the fourth component is analyzed: all nodes can detect the fault in less than $0.6s$. The proposed methodology has been tested with different topology networks and we obtained similar results.

VII. Concluding remarks

In this preliminary paper, it was shown that the distributed estimator proposed in [17] can be exploited in the task of detecting faults by sensor networks. Several different scenarios were considered and probabilistic time-varying thresholds for fault detection have been devised. Future research efforts will be first devoted to provide a detailed theoretical analysis on fault detectability, as well as complexity. Extensive simulation results will be provided and a quantitative comparison to the methods available in the literature will be carried out. Moreover, the proposed method will be extended to the case in which correlations are present among different state components. Finally, the case of distributed interconnected systems will be addressed.

References

[1] F. Boem, R. Ferrari, T. Parisini, and M. Polycarpou, "Distributed fault detection and isolation of continuous-time nonlinear systems," *European Journal of Control*, vol. 5-6, pp. 603–620, 2011.

[2] F. Koushanfar, M. Potkonjak, and A. Sangiovanni-Vincentelli, "On-line fault detection of sensor measurements," in *Sensors, Proceedings of IEEE*, vol. 2, oct. 2003, pp. 974 – 979 Vol.2.

[3] J. Chen, S. Kher, and A. Somani, "Distributed fault detection of wireless sensor networks," in *Workshop on Dependability issues in wireless ad hoc networks and sensor networks*. ACM, 2006, pp. 65–72.

[4] Y. Liu, K. Liu, and M. Li, "Passive diagnosis for wireless sensor networks," *Networks, IEEE/ACM Transactions on*, vol. 18(4), pp. 1132–1144, 2010.

[5] N. Lechevin and C. Rabbath, "Decentralized detection of a class of non-abrupt faults with application to formations of unmanned airships," *Control Systems Technology, IEEE Transactions on*, vol. 17, no. 2, pp. 484–493, 2009.

[6] S. Stankovic, N. Ilic, Z. Djurovic, M. Stankovic, and K. Johansson, "Consensus based overlapping decentralized fault detection and isolation," *Control and Fault-Tolerant Systems Conf.*, pp. 570–575, 2010.

[7] R. Ferrari, T. Parisini, and M. Polycarpou, "Distributed fault detection and isolation of large-scale discrete-time nonlinear systems: An adaptive approximation approach," *Automatic Control, IEEE Transactions on*, vol. 57, no. 2, pp. 275 –290, 2012.

[8] P. Baroni, G. Lamperti, P. Pogliano, and M. Zanella, "Diagnosis of large active systems," *Artificial Intelligence*, vol. 110, no. 1, pp. 135–189, 1999.

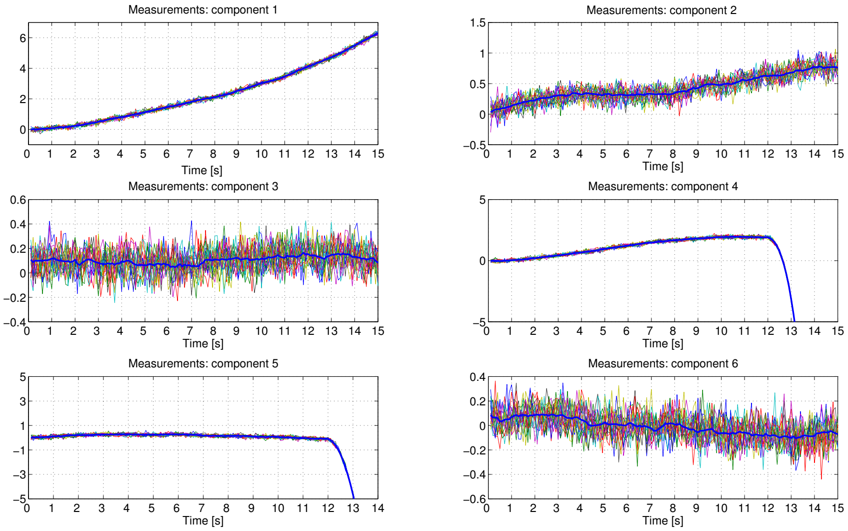

Fig. 2. The tracked signal represented by the thick blue curve and the measurements realized by the $N = 20$ nodes, with different colors for each node.

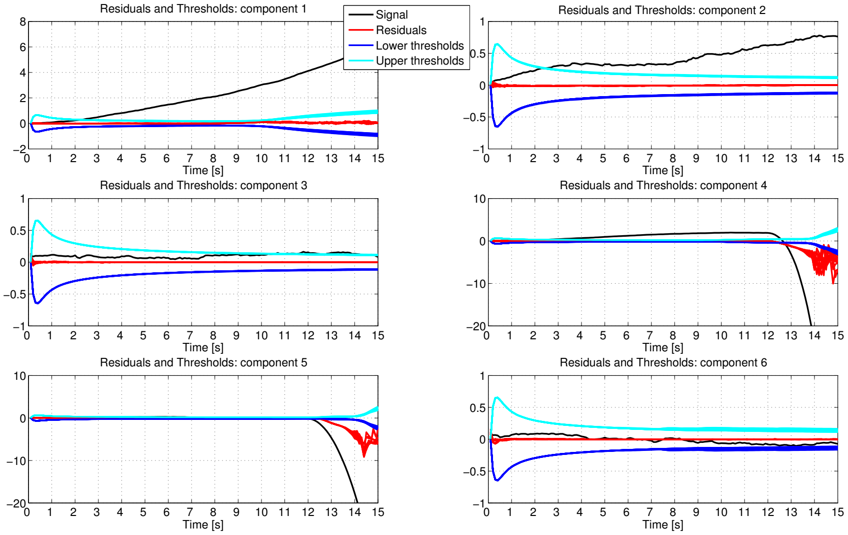

Fig. 3. Simulation results for each state component. Residuals and thresholds of all the nodes are represented in the same graph.

[9] Y. Wang, T. Yoo, and S. Lafortune, “Diagnosis of discrete event systems using decentralized architectures,” *Discrete Event Dynamic Systems*, vol. 2, pp. 233–263, 2007.

[10] F. Garin and L. Schenato, “A survey on distributed estimation and control applications using linear consensus algorithms,” in *Networked Control Systems*. Springer, 2010, vol. 406, pp. 75–107.

[11] A. Speranzon, C. Fischione, K. Johansson, and A. Sangiovanni-Vincentelli, “A distributed minimum variance estimator for sensor networks,” *IEEE Journal on Selected Areas in Communications*, vol. 26, no. 4, pp. 609–621, 2008.

[12] F. Cattivelli and A. Sayed, “Diffusion lms strategies for distributed estimation,” *Signal Processing, IEEE Transactions on*, vol. 58, no. 3, pp. 1035 –1048, 2010.

[13] E. Franco, R. Olfati-Saber, T. Parisini, and M. Polycarpou, “Distributed fault diagnosis using sensor networks and consensus-based filters,” in *Decision and Control, 45th IEEE Conf. on*, 2006, pp. 386 –391.

[14] R. Olfati-Saber, “Distributed kalman filtering for sensor networks,” in *Decision and Control, 46th IEEE Conf. on*, 2007, pp. 5492–5498.

[15] R. Carli, A. Chiuso, L. Schenato, and S. Zampieri, “Distributed kalman filtering based on consensus strategies,” *IEEE Journal on Selected Areas in Communications*, vol. 26, no. 4, pp. 622 –633, 2008.

[16] R. Olfati-Saber, “Kalman-consensus filter : Optimality, stability, and performance,” in *Decision and Control, 48th IEEE Conf. on, held jointly with 28th Chinese Control Conference*, 2009, pp. 7036–7042.

[17] F. Boem, Y. Xu, C. Fischione, and T. Parisini, “A distributed estimation method for sensor networks based on pareto optimization,” in *Decision and Control, Proc. of 51th Conf. on*, 2012, pp. 775–781.

[18] W. Chung and J. Speyer, “A general framework for decentralized estimation,” in *Proc. American Control Conference*, vol. 4, 1995, pp. 2931–2935.

[19] J. Lagarias, J. Reeds, M. Wright, and P. Wright, “Convergence properties of the nelder-mead simplex method in low dimensions,” 1998. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.120.6062