A Cognitive Fault Diagnosis System for Distributed Sensor Networks

Cesare Alippi, *IEEE Fellow,* Stavros Ntalampiras, and Manuel Roveri

***Abstract*—The paper introduces a novel cognitive Fault Diagnosis System (FDS) for distributed sensor networks which takes advantage of spatial and temporal relationships among sensors. The proposed FDS relies on a suitable functional graph representation of the network and a two-layer hierarchical architecture designed to promptly detect and isolate faults. The lower processing layer exploits a novel Change Detection Test (CDT) based on Hidden Markov Models (HMMs) configured to detect variations in the relationships between couples of sensors. HMMs work in the parameter space of linear time invariant (LTI) dynamic systems approximating, over time, the relationship between two sensors; changes in the approximating model are detected by inspecting the HMM likelihood. Information provided by the CDT layer is then passed to the cognitive one which, by exploiting the graph representation of the network, aggregates information to discriminate among faults, changes in the environment and false positives induced by the model bias of the HMMs.**

***Index Terms*—Fault diagnosis; distributed sensor network; intelligent sensors; hidden Markov model.**

I. Introduction

SENSOR networks monitoring a real environment are prone to faults or aging phenomena, whose impact affects the overall system performance. In fact, permanent or transient faults can influence the sensors, the analog electronics, the digital part of the embedded system inducing, in the best case, functional errors in the processing chain. In turn, erroneous information generates a strong side effect on the subsequent control chain leading to wrong decisions and inappropriate control actions.

A Fault Diagnosis System plays the important role of supervising the process operations in order to detect, isolate and identify a potential fault and, possibly, design accommodation actions [1]. The main components of the FDS are learned from the available data when physical descriptions for the process are unavailable. Not rarely, an appropriate model describing the underlying process is built and its validity assessed over time by the FDS through a change inspection mechanism, e.g., by observing changes in some features such as the approximating model residuals or parameters.

Unfortunately, when a change in the model is detected by the FDS, three situations might arise:

- "model bias": The model is no more representing the current data due to model approximating inefficiencies;
- "change in the environment": the environment is time-variant and the trained model is no more able to explain the current data;
- "fault": a sensor or a component of a unit is affected by a fault which induces an error in the datastream.

Distinguishing among the above perturbation classes is a major achievement which, whenever possible, allows the FDS for isolating the fault/change and proposing the appropriate accommodation action.

Published FDSs for sensor networks generally do not allow for distinguishing between faults and changes in the environment [2]. Moreover, the model bias is generally considered negligible, a hardly satisfied hypothesis in many applications. A review of the literature is given in Section II.

The paper presents a novel cognitive FDS acting on sensor datastreams that removes the strong hypotheses assumed in the literature; as such, under reasonable assumptions, it deals with the model bias case and proposes a method for discriminating between faults and changes in the environment.

Hidden a-priori information related to spatial and temporal relationships among sensor datastreams (both explicit and implicit) is exploited, leading to a functional dependency graph where nodes are the sensors and arcs are associated with the sensor-to-sensor relationship. In particular, for each sensor couple, a HMM is designed, which receives the parameters of a LTI model approximating the relationship. As such, spatial redundancy is modelled with a HMM operating in the parameter space of LTI dynamic models embedding the time dependency. When the likelihood between the HMM-based learning machine and the new incoming datastream falls below a threshold (which can be learnt as well) a change is detected (HMM-based Change-Detection Test) at the detection layer. The cognitive layer of the FDS, activated in response to a change alarm raised by the detection one, separates faults from time variant and bias cases by operating on the dependency graph of the network. At the same time, it allows us for isolating the fault for a possible subsequent accommodation phase.

The novelties of the proposed approach can be summarised as follows:

- introduction of a dependency graph representing the temporal and spatial relationships among the sensor units;
- suggestion of the joint use of LTI models and HMMs for designing a Change-Detection Test;
- design of a cognitive graph-based approach able to distinguish among changes induced by model bias, sensor faults or changes in the environment.

C. Alippi, S. Ntalampiras and M. Roveri are with the Dipartimento di Elettronica e Informazione, Politecnico di Milano, Milan, Italy, 20133 IT e-mail: {alippi,ntalampiras,roveri}@elet.polimi.it (see http://sagroup.ws.dei.polimi.it)

The paper is organized as follows. Section II surveys the fault diagnosis literature for sensor networks. Section III presents the modelling methodology for functional relationships in sensor networks, while Section IV introduces the proposed hierarchical cognitive FDS. Experimental results are given in Section V and conclusions drawn in the last Section.

II. Related Literature

The fault diagnosis system literature for sensor networks (also called sensor validation) embraces two main approaches exploiting information either coming from a single sensor or a set of sensors (a detailed review of the fault-detection literature can be found in [1], [2]). The former approach aims at detecting faults by inspecting data coming from a single sensor. Here, we have the limit checking method [3], which raises an alarm when the physical quantity under monitoring overcomes a threshold, and change-detection methods e.g., [4], [5], which aim at detecting variations in the statical behaviour of the physical phenomenon under observation. Techniques following the latter approach, and based on physical or analytical redundancy methods, detect faults by exploiting redundant/correlated information coming from multiple sensors.

Physical redundancy requires redundant sensors measuring the same physical quantity, possibly at different resolution levels. For instance, [6] suggests a fuzzy-based technique to correct data subject to sensor drifts and intermittent faults in the case of truly redundant sensors. Differently, analytical redundancy exploits functional relationships among the sensors which measure different, but correlated, physical quantities. Most design approaches for fault diagnosis rely on this concept and assume that a mathematical model of the healthy system is available. Fault diagnosis is achieved by comparing actual observations with those coming from the prediction model and, then, inspecting residual values or differences among redundant sensors [6]. [7] suggests a fuzzy rule approach for validation of highly correlated sensors (or quasi-redundant sensors). Several techniques based on artificial neural networks have been presented in the literature (e.g., [8]–[10]). For example, [8] suggested the use of autoassociative neural networks and Kohonen maps for sensor failure detection in redundant/correlated sensors. [9] describes the use of neural networks in sensor fault detection with specific attention to flight control systems. [10] applies a feedforward neural network to data coming from a space shuttle main engine. Other analytical redundancy-based approaches consider, for example, the use of Principal Component Analysis [11] and the Nadaraya-Watson statistical estimator [12]. Although these methods may achieve satisfactory performance levels they assume availability of an accurate model.

Fault detection and diagnosis issues have also been widely addressed within the Computational Intelligence community. For example, [13] suggests a model-based process supervision for fault detection and identification where a nonlinear observer based on a Radial basis function (RBF) neural network is used to approximate the unknown nonlinear dynamics (the linear part is assumed to be known). In the case of faults, a different RBF is used to identify the nonlinear characteristics of the fault profile. Similarly, [14] presents a robust fault diagnosis framework for detecting and approximating state and output faults affecting nonlinear multi input-multi output dynamical systems. Here, the nonlinear component of the process and sensor uncertainties are assumed to be unknown but bounded. [15] suggests a FDS for distributed systems based on a set of finite state machines (e.g., the model describing the message exchange within a transmission protocol). The nominal and the faulty behaviour of the system are assumed to be known; the detection phase relies on an observer comparing the ideal with the measured output datastreams. Several fault detection and diagnosis systems for application-specific scenarios can also be found in [16]–[20].

A practical way to conduct group level analysis for fault diagnosis is the K over N approach [21]: a change is detected when K out of N functionally related sensors detect it. Even though the approach may provide reasonable performance, it suffers from a severe disadvantage: it treats all sensors identically. Moreover, detection performance is heavily influenced by K. Lower values of K lead to low detection latency at the expenses of higher false positive rates. The opposite holds.

Echo state network (ESN) solutions have been recently considered for fault diagnosis, mainly at the sensor level. For instance, [22] exploits ESNs to detect faults in the temperature and moisture sensors of a mote-class device. The ESNs are trained by using data coming from the nominal state and no spatial redundancy among different units is taken into account. [23] suggests an ESN for identifying anomalies in the concentration of natural gas in underground coal mines. The ESN is trained on normal (safe condition) data coming from CH_4, CO_2, CO, O_2 and pressure sensors; the detection mechanism is based on the difference between the output of the ESN and that of the sensor (with a threshold defined according to the Neyman-Pearson test).

A different approach was suggested in [24], [25] and [26] where hidden semi-Markov models (HSMM) have been proposed to improve fault detection and identification. In particular, [24] suggested the use of HSMMs trained on temperature, speed and direction of the wind as well as the historical values of the PM2.5 concentration to predict future instances. [25] and [26] exploit HSMMs for health monitoring of hydraulic pumps. Wavelet-based features of the vibrational signals coming from the pumps have been used to train the fault detection/identification system and a fault dictionary is requested to detect/isolate the erroneous conditions. All above approaches work at the single sensor level by exploiting temporal relationships and do not consider existing spatial dependencies among the sensor units (which is one of the novelties of the proposed approach).

It should be noted that methods present in the literature neither consider a dynamic predictive model coupled with a HMM nor exploit the functional graph of the network to carry out a cognitive analysis able to separate faults from approximating model bias and environmental changes as we do here.

III. Modeling functional Relationships In Sensor Networks

Let us consider a sensor network composed of N fixed sensing units deployed within the environment $\mathcal{P}$. Each unit can host up to M sensors opening views on different physical aspects of $\mathcal{P}$ (e.g., temperature, humidity, change in slope, vibrations, rain intensity). Each j-th sensor of the i-th unit acquires a scalar datastream $X_{i,j}$.

We neither require a specific topology for the network nor a particular communication routing protocol. The FDS can be in execution at the remote control room (where all data are received) or at the base stations (e.g., within a hierarchical topology setup) if the network can be partitioned into functionally disjoint sub-networks (each of which containing a base station). Data communication among units must hence be intended between the units and the base station/control room where the FDS algorithm is executed. A synchronization algorithm (e.g., [27]) should be considered to guarantee time consistency among samples whenever poor clock generators (as those in sensor networks) are available (the clock skew can easily rise to seconds in few days [28]).

A. Modeling the network: the dependency graph

The cognitive framework for fault diagnosis relies on the ability to model functional relationships among the acquired views of $\mathcal{P}$. In more detail, each relationship captures spatial and temporal dependency from data provided by a generic couple of sensors. Fig. 1 shows an example of a sensor network with functional dependencies.

A *direct relationship* exists between a couple of sensors of the same type (e.g., temperature vs temperature): if datastreams $X_{i,j}$ and $X_{v,j}$, $i \neq v$ are correlated, an arc linking the j-th sensor of unit i with its counterpart of unit v is introduced. For instance, two clinometers insisting on the same connected structure are related; those deployed far apart probably are not.

An *indirect relationship* can be introduced between two generic sensors by means of a third entity. For instance, a clinometer and a strain gauge sensor whose readouts are affected by a parasitic thermal effect will be indirectly correlated thanks to the temperature dependency. Such a relationship can exist between sensors mounted in the same unit (i.e., $X_{i,j}$ and $X_{i,u}$, $j \neq u$) or present in different units (i.e., $X_{i,j}$ and $X_{v,u}$, $i \neq v$, $j \neq u$). Of course, indirect relations are mitigated by the presence of compensation mechanisms; in this case information useful for the analysis must be extracted before compensation takes place.

De facto, direct and indirect relationships introduce a functional constraint between couples of sensors. Denote by $f_{\{(i,j),(u,v)\}}$ the functional relationship between the generic j-th sensor of unit i and the v-th sensor of unit u. The nodes of G are the network sensors; the arcs represent the functional relationships between couples of sensors. Given a network, not all the $(N \times M)(N \times M - 1)$ relationships in G are relevant. For example, two sensors might be weakly correlated due to topological or phenomenological reasons or not correlated at all in one direction due to causality.

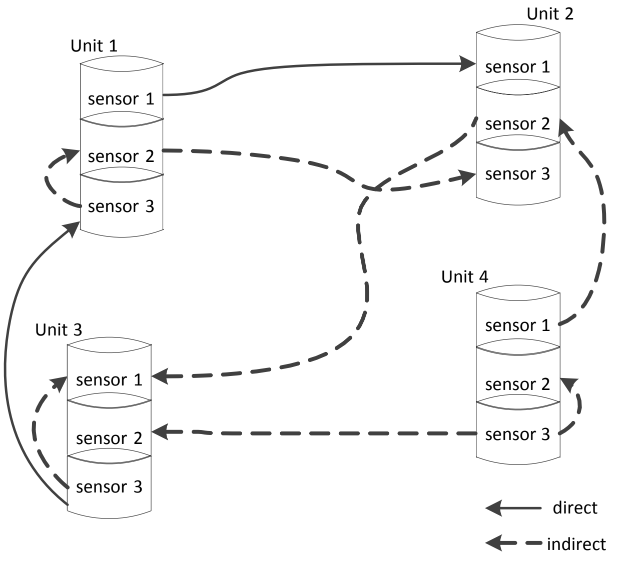

Fig. 1. Direct and indirect relationships in a distributed monitoring network.

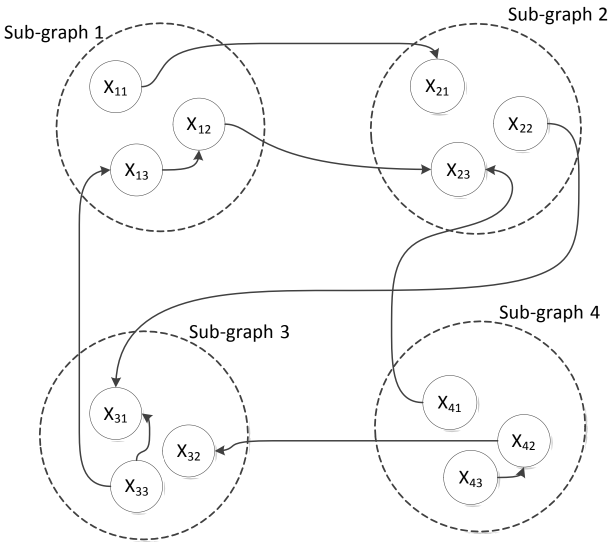

Fig. 2. The dependency graph of the sensor network og Fig. 1.

The *reduced dependency graph* is then derived from G and defined as graph

$$G_R = \{V, E\}$$

where V is the set of nodes of the graph representing the $N \times M$ sensors and E a set collecting all arcs associated with functional relationships whose correlation is above a threshold. The level of dependency associated with relationship $f_{\{(i,j),(u,v)\}}$ is here chosen to be the linear correlation index between two datastreams $X_{i,j}$ and $X_{v,u}$: when the peak of the crosscorrelation is above γ_{min}, γ_{min} being a suitably tuned threshold, the relationship is considered to be relevant and worth to be included in E; other dependency indexes can be surely derived and considered. Let $\overline{F}$ be the set of functional relationships with correlation index larger than γ_{min}.

We remove from G_R isolated nodes.

Fig. 2 shows the graph-based representation of the sensor network of Fig. 1: we have 4 units; each unit is a subgraph representing the mounted sensor.

B. Modeling the relationship between two sensors with a Hidden Markov Model

In the following we assume that the relationship between a couple of sensors $f_{\{(i,j)(u,v)\}}$ can be modelled either as a time-invariant (TI) dynamic system or as a finite sequence of TI dynamic systems satisfying the HMM hypotheses. No assumption about the linearity of the relationship is made.

Let's imagine to model now a $f_{\{(i,j)(u,v)\}}$ with the Single-Input Single-Output (SISO) Linear (LTI) model of form

$$A(z)X_{i,j}(t) = \frac{B(z)}{F(z)}X_{v,u}(t) + \frac{C(z)}{D(z)}d(t) \tag{1}$$

where z is the backward time-shift operator, $d(t)$ an independent and identically distributed (i.i.d.) random variable accounting for the noise, $A(z)$, $B(z)$, $C(z)$, $D(z)$, and $F(z)$ represent the z-transform functions of model parameters θ_A, θ_B, θ_C, θ_D and θ_F, respectively. In the following we assume $A(z)$, $B(z)$, $C(z)$, $D(z)$, and $F(z)$ to be either time-invariant or time-varying but following a Markov chain model.

A given SISO model M (e.g., ARX, ARMAX or OE) is then an element locally approximating $f^{\theta}_{\{(i,j),(u,v)\}}$, $\hat{y}_\theta = M(\theta), \theta \in D_M$ parametrized in $\theta = [\theta_A, \theta_B, \theta_C, \theta_D, \theta_F]$.

The use of linear models allows us to apply theoretical results delineated in [29], [30]. More specifically, consider a training dataset composed of N_T {*input,output*} couples $\{u(t), y(t)\}_{t=1}^{N_T}$, a loss function $V_{N_T} = \frac{1}{N_T}\sum_{N_T}(y - \hat{y}_\theta)^2$ whose minimization provides an estimate $\hat{\theta}$, of the optimal parameter $\theta^* = \arg\min_{\theta \in \mathcal{D}_\mathcal{M}} \lim_{N_T \to +\infty} \mathbb{E}\left[V_{N_T}\right]$.

Under the assumption that each $f_{\{(i,j),(u,v)\}}$ function satisfies the exponential stability for closed loop (i.e., accurate approximations of $X_{i,j}(t)$ can be generated given finite time windows of $X_{i,j}(.)$ and $X_{v,u}(.)$), from [30] we have that

$$\sqrt{N_T}P^{-\frac{1}{2}}(\hat{\theta} - \theta^*) \sim \mathcal{N}(0, I) \qquad \text{when } N_T \to \infty; \tag{2}$$

$P \in \mathbb{R}^{p \times p}$ is the covariance matrix of the p parameters of the model.

It comes out that, under the above assumption and a sufficiently large N the distribution underlying the parameter vectors $\hat{\theta}$ is a multivariate Gaussian, with mean θ^* and covariance matrix P. We emphasize that the same framework can be applied to Extreme-Learning Machines [31] or Reservoir Computing Networks [32].

By following the (2) a HMM with parameters θ ruled by a mixture of gaussians becomes a natural solution to approximate $f_{\{(i,j),(u,v)\}}$. The nodes of the HMM represent, de facto, a probabilistic ensemble of LTI models somehow minimizing the model bias $||\hat{y}_\theta - f_{\{(i,j),(u,v)\}}||$ if the training set is sufficiently informative. More in detail, the HMM is defined as

$$\mathcal{H} = \{n, \mathcal{F}, A, \pi\}, \tag{3}$$

where n is the number of states, $\mathcal{F} = \{p_1, \ldots, p_n\}$ is the set of probability density functions (pdfs) associated with each state, A is the $n \times n$ state transition probability matrix and π the $n \times 1$ initial state distribution vector. Thanks to (2) the pdf associated with each state can be safely modelled as a mixture of Gaussians (GMMs). In fact, the GMM of the i-th state is defined as

$$p_i(\hat{\theta}|\Phi_i) = \sum_{k=1}^{K_i} w_{k,i}\mathcal{N}(\hat{\theta}|\mu_{k,i}, \Sigma_{k,i}) \tag{4}$$

where K_i is the number of Gaussian mixtures for the i-th state, $w_{k,i}$ is the weight for state i and Gaussian mixture k, $\Phi_i = [\mu_{1,i}, \ldots, \mu_{k_i,i}, \Sigma_{1,i}, \ldots, \Sigma_{k_i,i}]$ with $\mu_{k,i}$ and $\Sigma_{k,i}$ the mean vector and the covariance matrix for state i and Gaussian mixture k, respectively. We considered a mixture of Gaussian functions with diagonal covariance matrices since, an L-th order full covariance GMM can be achieved using a diagonal covariance GMM of a larger order [33]. Thus, the mixture of Gaussian solution is as effective as the former at a much lower computational cost. We remark that we should consider a single Gaussian function with full covariance matrix to model a state of the process.

By modelling parameters θ with a HMM we mitigate the effect of model bias and time variance provided that the training set is sufficiently informative and explores both time variance and nonlinearity.

IV. The Cognitive Fault Diagnosis System

The proposed FDS is organized as the two-layer architecture of Figure 3. The *lower level* is composed by a set of Change Detection Tests (CDTs) each of which monitoring the stationarity of a relationship associated with a couple of sensors in G_R. Each HMM CDT works in the parameter space θ to detect variations in the relationship between the two sensors (the CDT is described in the Section IV-A). Unfortunately, a CDT is not able to distinguish among changes induced by a fault in a sensor, an environmental change in $\mathcal{P}$ or a false positive generated by a model bias since such classes are indistinguishable. To address this issue the *upper level* of the FDS has been designed to be able to discriminate between faults, changes in $\mathcal{P}$ and false positives by exploiting information associated with the network graph G_R. The upper level of the FDS relies on a cognitive algorithm aggregating decisions and log-likelihood information provided by the HMM-CDT in the lower level. The cognitive aggregation level is described in Section IV-B.

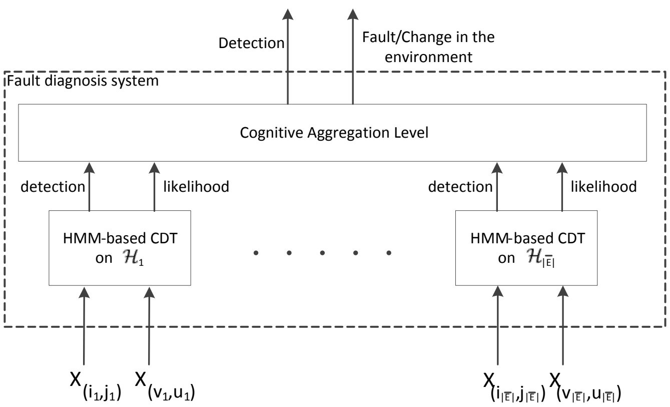
Detection
Fault/Change in the environment
Fault diagnosis system
Cognitive Aggregation Level
detection
likelihood
detection
likelihood
HMM-based CDT on $\mathcal{H}_1$
. . . . . .
HMM-based CDT on $\mathcal{H}_{|\bar{E}|}$
$X_{(i_1,j_1)}$
$X_{(v_1,u_1)}$
$X_{(i_{|\bar{E}|},j_{|\bar{E}|})}$
$X_{(v_{|\bar{E}|},u_{|\bar{E}|})}$

Fig. 3. The proposed FDS.

```
Let $\{X_{(i,j)}(t), 1 \leq t \leq T_0\}$ and $\{X_{(v,u)}(t), 1 \leq t \leq T_0\}$
  be the training set of sequences $X_{(i,j)}$ and $X_{(v,u)}$;
Create $T_0 - N_T + 1$ overlapping windows of $N_T$ data;
Estimate parameters $\hat{\theta}$ for each data window and
  generate the sequence $S_T = \{\hat{\theta}^1, \ldots, \hat{\theta}^{T_0 - N_T + 1}\}$;
Train $\mathcal{H} = \{s, \mathcal{F}, A, \pi\}$ on $S_T$;
$T_h = \min_{1 \leq s \leq T_0 - N_T + 1} l_{\{(i,j),(u,v)\}}(s)$ as in Eq. (5);
$t = T_0$, $s = T_0 - N_T + 1$;
repeat
    t = t + 1;
    Acquire $X_{(i,j)}(t)$ and $X_{(v,u)}(t)$;
    s = s + 1;
    Estimate the parameter $\hat{\theta}^s$ on
      $\{X_{(i,j)}(\bar{t}), t - N_T + 1 \leq \bar{t} \leq t\}$ and
      $\{X_{(v,u)}(\bar{t}), t - N_T + 1 \leq \bar{t} \leq t\}$;
    Compute $l_{\{(i,j),(u,v)\}}(s)$ as in Eq. (5);
    if $l_{\{(i,j),(u,v)\}}(s) < T_h$ then
        Raise an alarm: change detected;
    end
until (1);
```

Algorithm 1: The HMM-CDT for a generic functional relationship $f_{\{(i,j),(u,v)\}}$ in G_R.

```
HMM-CDT associated with $\mathcal{H}_{\{(i,j),(u,v)\}}$ detected a
  change in the $\bar{s}$-th data window;
Partition the graph into sets $E^+, E^-$ and $E^{\mathcal{P}}$ according
  to Eq. (7), (8), (9) ;
Compute $S^+$, $S^-$, $S^{\mathcal{P}}$ according to Eq. (10), (11) and
  (12);
Compute $T^+$, $T^-$, $T^{\mathcal{P}}$ according to Eq. (13), (14) and
  (15);
if $S^{\mathcal{P}} < T^{\mathcal{P}}$ then
    Change in $\mathcal{P}$ detected;
else
    if $S^+ < T^+$ and $S^- < T^-$ then
        Model Bias in $\mathcal{H}_{\{(i,j),(u,v)\}}$ detected;
    else
        if $S^+ < T^+$ then
            Fault at sensor $X_{(i,j)}$ detected;
        else
            Fault at sensor $X_{(v,u)}$ detected;
        end
    end
end
```

Algorithm 2: The algorithm which combines the HMM-based CDTs for change identification and isolation.

A. The HMM-based Change Detection Test

The proposed HMM-CDT aims at evaluating, by means of a HMM, the evolution over time of the parameters $\hat{\theta}$s approximating the relationship $f_{\{(i,j),(u,v)\}}$; $X_{(i,j)}$ is the output and $X_{(v,u)}$ the input of the LTI. $\hat{\theta}$ are estimated on overlapping windows of N_T data.

The HMM-CDT requires the training of HMM $\mathcal{H}_{\{(i,j)(v,u)\}}$, devoted to model the relationship between sensors (i,j) and (u,v) $\mathcal{H}_{\{(i,j),(u,v)\}}$, is trained by means of the Baum-Welch algorithm [34].

During the operational life, the parameter $\hat{\theta}^s$ is estimated on the s-th window of data and the log-likelihood

$$l_{\{(i,j),(u,v)\}}(s) = P(\hat{\theta}^1, \ldots, \hat{\theta}^s | \mathcal{H}_{\{(i,j),(u,v)\}}) \qquad (5)$$

computed and indicating how likely the sequence of estimated parameters $\hat{\theta}^1, \ldots, \hat{\theta}^s$ has been generated by the $\mathcal{H}_{\{(i,j),(u,v)\}}$. The log-likelihood is computed with the Viterbi algorithm [35].

When the log-likelihood decreases below a threshold T_h, a change in the relationship is detected (the sequence of inputs is no more recognized by the learning machine). Threshold T_h can be defined by the operator, who exploits a-priori available information. However, if not available, T_h can be estimated as the minimum value assumed by the log-likelihood in the training (or better validation) sequence. T_h can be scaled by a coefficient factor c_1 (with $0 \leq c_1 \leq 1$) to tradeoff the robustness of the machine w.r.t. false positives and false negatives.

The HMM-based CDT is detailed in Algorithm 1.

B. The cognitive aggregation level

The cognitive level aggregates the information coming from all sensor units to distinguish among faults, changes in $\mathcal{P}$ and false positives induced by model bias in the HMM CDT. Differently from the HMM-CDTs that are executed sequentially, the cognitive aggregation level is activated only in response to a detection alarm raised by at least a CDT-HMM. Detections and log-likelihoods of others CDTs are used to assess and, possibly, identify the change.

The motivating idea is that a change in $\mathcal{P}$ for a given type of sensors must be perceived also by a set of other CDTs, at least as a decrement in the log-likelihood values (not necessarily below the threshold). Differently, in the case of faults, only the CDTs associated with relationships that have either as input or output the faulty sensor are affected by the change. Finally, if a false positive occurs, other CDTs should not be affected.

To evaluate the reliability of the information coming from HMM-CDTs we introduce a reliability index $w_{\{(i,j),(u,v)\}}$ for the HMM $\mathcal{H}_{\{(i,j),(u,v)\}}$ defined as

$$w_{\{(i,j),(u,v)\}} = \sum_{1 \leq s \leq T_0 - N_T + 1} l_{\{(i,j),(u,v)\}}(s). \qquad (6)$$

Weights are computed on the training set; the *weighted reduced graph* is the reduced graph augmented with the weight information.

Definition: Let E^+ be the set of functional relationships such that either the source or the target node of the arc is $X_{(i,j)}$, i.e.,

$$E^+ = \{f_{(\bar{i},\bar{j})(\bar{v},\bar{u})} \in \{\overline{F} - f_{(i,j)(v,u)}\} | \\ (\bar{i} = i \text{ and } \bar{j} = j) \text{ or } (\bar{v} = i \text{ and } \bar{u} = j)\}. \qquad (7)$$

Definition: Let E^- be the set of functional relationships such that either the source or the target node of the arc is

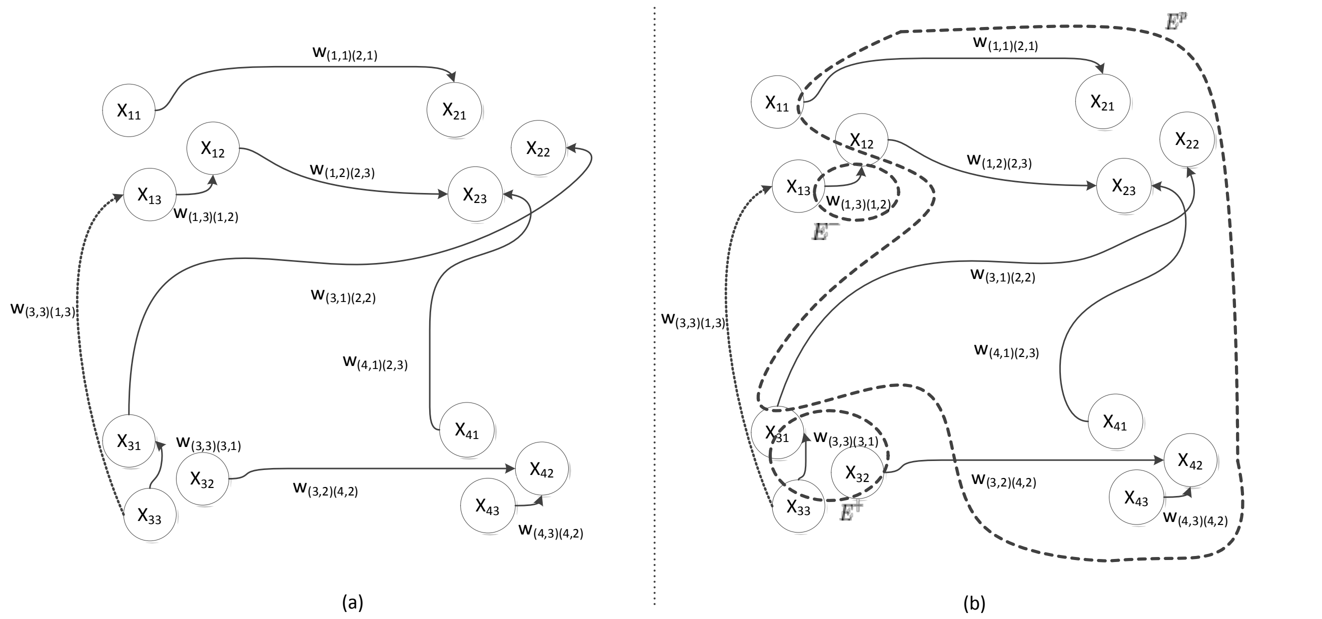

Fig. 4. The cognitive aggregation level: a) the reduced weighted dependency graph; b) an example of arcs partitioning into E^+, E^- and $E^{\mathcal{P}}$ given a change detected in the functional relationship $f_{\{(3,3),(1,3)\}}$.

$X_{(v,u)}$, i.e.,

$$E^- = \{f_{(\bar{i},\bar{j})(\bar{v},\bar{u})} \in \{\overline{F} - f_{(i,j)(v,u)}\}| \\ (\bar{i} = v \text{ and } \bar{j} = u) \text{ or } (\bar{v} = v \text{ and } \bar{u} = u)\}. \quad (8)$$

Definition: Let $E^{\mathcal{P}}$ be the set of functional relationships whose source or target node is neither $X_{(i_q,j_q)}$ nor $X_{(v_q,u_q)}$, i.e.,

$$E^{\mathcal{P}} = \overline{F} - \{E^+ \cup E^- \cup \{f_{\{(i,j)(v,u)\}}\}\} \quad (9)$$

After a change detected in $f_{\{(i,j)(v,u)\}}$ the remaining $\bar{F} - 1$ functional relationships of the weighted reduced dependency graph are partitioned into sets E^+, E^- and $E^{\mathcal{P}}$. The reason for the partitioning is as follows:

- a fault in sensor $X_{(i,j)}$ affects the functional relationships in E^+ but not those in E^- and $E^{\mathcal{P}}$;
- a fault in the sensor $X_{(v,u)}$ affects the functional relationships in E^- but not those in E^+ and $E^{\mathcal{P}}$;
- a change in $\mathcal{P}$ affects the functional relationships in E^-, E^+ and $E^{\mathcal{P}}$;
- a model bias affecting $\mathcal{H}_{\{(i,j),(u,v)\}}$ would mostly affect the functional relationship between (i,j) and (u,v) but, in principle, not the functional relationships in E^-, E^+ and $E^{\mathcal{P}}$ provided that approximating relationships are characterised by different bias contributions.

An example of partitioning is shown in Figure 4a; in Figure 4b a change is detected in relationship $f_{\{(3,3)(1,3)\}}$. We have

$$\begin{aligned} E^+ &= \{f_{\{(3,3)(3,1)\}}\}; \\ E^- &= \{f_{\{(1,3)(1,2)\}}\}; \\ E^{\mathcal{P}} &= \{f_{\{(1,1)(2,1)\}}, f_{\{(1,2)(2,3)\}}, f_{\{(3,1)(2,2)\}}, f_{\{(4,1)(2,3)\}}, \\ &\quad f_{\{(3,2)(4,2)\}}, f_{\{(4,3)(4,2)\}}\}. \end{aligned}$$

Defined $\bar{s}$ to be the index of the data window where the HMM CDT detected a change, the proposed aggregation level computes the normalized sum of the log-likelihoods, suitably weighted according to (6), of the arcs in E^+, E^- and $E^{\mathcal{P}}$:

$$S^+ = \frac{1}{\sum_{E^+} w_{\{(i,j),(u,v)\}}} \sum_{E^+} w_{\{(i,j),(u,v)\}} l_{\{(i,j),(u,v)\}}(\bar{s}); \quad (10)$$

$$S^- = \frac{1}{\sum_{E^-} w_{\{(i,j),(u,v)\}}} \sum_{E^-} w_{\{(i,j),(u,v)\}} l_{\{(i,j),(u,v)\}}(\bar{s}); \quad (11)$$

$$S^{\mathcal{P}} = \frac{1}{\sum_{E^{\mathcal{P}}} w_{\{(i,j),(u,v)\}}} \sum_{E^{\mathcal{P}}} w_{\{(i,j),(u,v)\}} l_{\{(i,j),(u,v)\}}(\bar{s}). \quad (12)$$

The core of the cognitive aggregation level is thus the ability to compute S^+, S^- and $S^{\mathcal{P}}$ by exploiting information coming from all the relationships of the weighted reduced dependency graph. S^+, S^- and $S^{\mathcal{P}}$ measure how the change detected in the relationship $f_{\{(i,j)(v,u)\}}$ is perceived in other relationships. If a fault affects sensor (i,j), S^+ should decrease, while S^- and $S^{\mathcal{P}}$ should not. Similarly, if a fault affects sensor (u,v), S^- should decrease, S^+ and $S^{\mathcal{P}}$ not. If a change in $\mathcal{P}$ occurs, $S^{\mathcal{P}}$ should decrease as well as S^+ and S^-.

To detect decreases in S^+, S^- and $S^{\mathcal{P}}$ we rely on a simple thresholding mechanism. The thresholds for S^+, S^- and $S^{\mathcal{P}}$ are computed as follows:

$$T^+ = \frac{1}{\sum_{E^+} w_{\{(i,j),(u,v)\}}} \sum_{E^+} w_{\{(i,j),(u,v)\}} T_{h,\{(i,j),(u,v)\}}; \quad (13)$$

$$T^- = \frac{1}{\sum_{E^-} w_{\{(i,j),(u,v)\}}} \sum_{E^-} w_{\{(i,j),(u,v)\}} T_{h,\{(i,j),(u,v)\}}; \quad (14)$$

$$T^{\mathcal{P}} = \frac{1}{\sum_{E^{\mathcal{P}}} w_{\{(i,j),(u,v)\}}} \sum_{E^{\mathcal{P}}} w_{\{(i,j),(u,v)\}} T_{h,\{(i,j),(u,v)\}}. \quad (15)$$

where $T_{h,\{(i,j),(u,v)\}}$ is the minimum value assumed by the log-likelihood in the training sequence for the HMM-CDT of functional relationship $f_{\{(i,j),(u,v)\}}$ as described in the previous section. Thresholds $T^{\mathcal{P}}$, T^+ and T^- can be scaled by a coefficient factor c_2 with $0 \leq c_2 \leq 1$ to increase the robustness w.r.t. false positives. We suggest to select $c_2 > c_1$ since we want to detect decreases in the likelihood which did not yet raised an alarm (and hence the likelihoods are above their respective thresholds). In fact, if we consider $c_2 \leq c_1$, we would require that the weighted average of the likelihoods computed in Eq. (10,11,12) decreases below the weighted average of the thresholds for change detection T_hs computed in Eq. (13,14,15) but this is a nonsense since relationships in E^+, E^- and $E^{\mathcal{P}}$ did not detect a change yet.

To sum up, the cognitive aggregation level acts as follows:

- If $S^{\mathcal{P}}$ decreases below threshold $T^{\mathcal{P}}$, a *change in* $\mathcal{P}$ is identified;
- If $S^{\mathcal{P}} > T^{\mathcal{P}}$ and $S^+ < T^+$ (or $S^- < T^-$), a *fault in sensor* $X_{(i,j)}$ (or in sensor $X_{(v,u)}$) is detected;
- If $S^{\mathcal{P}} > T^{\mathcal{P}}$ and $S^+ > T^+$ and $S^- > T^-$, a false positive induced by a *model bias* is detected.

If both S^+ and S^- are above their respective thresholds, we can raise the alarm *fault in either* $X_{(i,j)}$ *or* $X_{(v,u)}$ but we cannot isolate the affected sensor since not enough information is available. Further analyses could thus be performed at the sub-graph level by considering the relationships associated with the sub-graph (or the sub-graphs) of the sensors affected by the change. The cognitive aggregation level algorithm is given in Algorithm 2.

Comments on real sensor networks

The effectiveness of the proposed cognitive fault diagnosis system relies on the ability to exchange data among the units of the distributed sensor network. This can be achieved with a single, multi-hop or hybrid communication depending on the nature of the deployment and the chosen technology. In general, a centralized solution is taken, i.e., the FDS algorithm is executed at the remote control room where all data are sent and energy availability is not an issue. Differently, if the reduced dependency graph can be partitioned into not overlapping sub-networks then the FDS can be distributed at the sub-network level and executed at the base station (where local data instances are conveyed before activating the remote communication). A further particular case is that of isolated units. Here, only indirect functional relationships can be created (unless sensor redundancy is envisaged) and the FDS can be executed directly at the unit level (if enough energy and computational power is available).

Not rarely, sensor units working in a real scenario are characterized by the missing data issue, whose severity depends on two distinct causes. Data missing can either be induced by permanent/transient faults affecting the units or electromagnetic disturbances preventing the data packet to be delivered to the target (packet loss) within a best effort communication protocol framework. When the missing of data is associated with the communication aspect (e.g., noisy channel), a reliable communication protocol could be considered to compensate the problem at the expenses of an increased energy consumption and communication complexity overhead: this solution might not be acceptable in limited resource-based embedded systems.

The effect of transient faults either affecting the communication channel or the functionality of the units can be partly – and often effectively – mitigated by resorting to data reconstruction algorithms. This aspect must be taken into account whenever the datastream is expected to suffer from the missing data problem. In this case, reconstruction techniques for distributed sensor networks, e.g., see [36], [37], and [38], reconstruct the missing data by exploiting temporal and spatial redundancies among sensors. Clearly, the operation introduces an additional uncertainty on the missing data unless the reconstruction mechanism is optimal. That said, the fault diagnosis system presented here works in the parameter space with parameters learned on a window of data. It is expected that the impact on the parameter is negligible if the number of missing data is much smaller than the size of the data window; the opposite holds. However, we know exactly when the data are missing, situation that, as per se, constitutes a fault. This a priori information might suggest us to postpone the diagnosis phase until a good window of data is available (at the expenses of a delay in the detection of a potential change) or assign a confidence to the outcome of the FDS also function of the number of missing data.

V. Experiments and Results

The aim of the experimental section is to show both the detection and the recognition ability of the proposed FDS. To achieve this goal we considered two different scenarios: *detection* and *recognition*. The first refers to the case where a change affecting a datastream must be detected as soon as possible while maintaining under control false positives and false negatives. The latter refers to the case where a sensor fault or a change in $\mathcal{P}$ affects a sensor network and the proposed FDS has to promptly detect and recognize it.

To evaluate the effectiveness of the proposed FDS we considered both synthetic data and real measurements coming from various sources. Section V-A presents the detection experiments, while the recognition experiments are shown in Section V-B.

The HMMs of the proposed CDT have been configured in a fully connected topology (ergodic HMMs). The torch machine learning framework [39] was used both to construct and estimate the accuracy of the HMMs. The maximum number of *k*-means iterations for cluster initialization was set to 50; the Baum-Welch algorithm used to estimate the transition matrix was bounded to 25 iterations with a threshold of 0.001 between subsequent iterations. The number of explored states ranges from 3 to 7 while the number of Gaussian components used to build the GMM belongs to the $\{1, 2, 4, 8, 16, 32, 64, 128, 256, 512\}$ set.

The main parameters of the cognitive FDS are detailed in Table I; their values have been experimentally determined.

The SISO model (1) considered for the HMM-based CDTs is an ARX model whose autoregressive and exogenous component orders range from 1 to 6. The right order for time lags and parameters are learned during the training phase.

In the detection scenario, the proposed FDS is compared with the parity equation approach [1], [5], which inspects

Symbol	*Value*	*Description*
γ_{min}	0.5	Threshold on cross-correlation
c_1	0.1	Correction factor for T_h
c_2	0.5	Correction factor for T^+, T^- and $T^{\mathcal{P}}$

TABLE I
THE PARAMETERS OF THE PROPOSED FDS.

the discrepancy between the process behavior and the process model describing the nominal change-free behavior. The threshold is set as the highest discrepancy on unseen training data. When the discrepancy overcomes this threshold, a change is detected.

Differently, in the recognition scenario, we compared the proposed FDS with the K/N fusion method [21], commonly used to combine decisions made by different sensors. An environmental change is detected when at least K sensors (over the total number of N sensors) detect a change. As suggested in [21], K is fixed at $N/2$.

Five figures of merit have been defined to evaluate the detection and the recognition accuracy:

- *False positive index* (FP): it counts the times a test detects a change in the sequence when there it is not (percentage).
- *False negative index* (FN): it counts the times a change is not detected when there it is (percentage).
- *Detection Delay* (DD): it measures the time delay in detecting a change (number of samples).
- *Change-in-the-environment recognition rate* (CE): it measures the times a change in the environment is correctly identified (percentage).
- *Fault recognition rates* (F): it measures the times a fault is correctly identified (percentage).
- *Isolation rates*: it measures the times a fault is correctly isolated, i.e., the sensor affected by the fault is correctly identified (percentage).

A. Detection

This scenario, which examines the ability of the proposed FDS to detect statistical changes in datastreams, encompasses a synthetic dataset, a dataset coming from the Barcelona water network simulator and a dataset coming from a monitoring system working under real-world conditions for rock collapse forecasting.

1) Synthetic Dataset: This experiment refers to data generated by ARX(2,2) model

$$X_i(t) = a_1 X_i(t-1) + a_2 X_i(t-2) + \\ b_1 X_j(t-1) + b_2 X_j(t-2) + e(t)$$

where $a_1 = 0.5$, $a_2 = 0.2$, $b_1 = 0.1$, $b_2 = 0.3$, and $e(t) \sim N(0, \sigma^2)$ is a zero-mean Gaussian noise parameterized in its variance σ^2. The exogenous input X_j has been modeled as

$$X_j(k) = 5\sin(0.05k) + 3\sin(0.09k) + \epsilon(k),$$

where $\epsilon(k) \sim N(0, 0.01^2)$ is a zero-mean Gaussian noise affecting the exogenous input.

Each experiment lasts 12000 samples with the first 4000 samples used for training. After 4000 samples an artificially injected perturbation λ affects the coefficients of the ARX model. We considered two types of perturbations, i.e., abrupt changes and drifts, reflecting the occurrence of a permanent or transient fault or a smooth aging effect in the sensors, respectively. In case of abrupt changes, the parameters of the ARX model suddenly change from $\theta = \{a_1, a_2, b_1, b_2\}$ to $\theta_\lambda = \{a_1(1+\lambda), a_2(1+\lambda), b_1(1+\lambda), b_2(1+\lambda)\}$. In case of drift changes, the parameters of the ARX model slowly change from θ to θ_λ, which is now reached at the end of the experiments. The values of lambda considered in the experiments are $\lambda = \{0.03, 0.05, 0.07, 1\}$.

To evaluate the performance of the proposed method we considered different strengths for the noise $\sigma = \{0.01, 0.02, 0.04, 0.07, 0.1\}$. Simulation results are averaged over 250 runs.

The results of synthetic data for the abrupt perturbation case are shown in Table II. We see that the FP rate and the delay are increasing with the noise level. As expected, the FN rate reduces as λ increases. Similarly, for a given noise level σ, the mean delay reduces as λ increases. We observe that even in highly noisy conditions, the proposed solution guarantees high detection accuracy and low detection delays. The parity equation approach provides lower performances both in terms of detection accuracy and delay. In particular, the proposed method always guarantees lower mean delays, much lower FN rates (at the expenses of a slightly higher FP rates at low σs). The results of the drift type of change presented in Table III are in line with those of the abrupt change ones.

2) Barcelona water network simulator: This dataset consists in simulated data of the water distribution network of the city of Barcelona [40]. While the real network is quite complex (200 district metering areas and 400 control points), the network simulator provides flow meter data with respect to two related in time pumps for a time period of approximately one month. Eight types of faults are artificially induced in the second pump. The ability of the FDS to detect faults is tested by conducting an experiment for each type of fault. Each experiment lasts 1488 samples: the first 744 samples represent the normal situation, while in the remaining 744 samples the second pump is affected by a fault. Figure 5 shows the 8 datasets for the second pump.

The proposed FDS is trained with 500 data samples to model the nominal state and 100 to determine the threshold.

We report that all eight faults were correctly identified as faults by the proposed FDS with no occurrences of false positives or false negatives. The types of faults along with the respective detection delays are given in Table IV. The window of the CDT has length $N_T = 40$ samples. The proposed FDS detects each category of faults with a relatively small latency. Differently, the parity equation method provides higher detection delays and fails to detect two types of faults (negative offset abrupt additive and negative offset incipient additive).

3) Rock collapse forecasting system: This experiment refers to a real-world dataset provided by a real-time monitoring system for rock fall forecasting [41] designed by our group and deployed in the Alps.

Here we consider measurements coming from a novel

		Proposed approach			Parity equation method		
σ of the noise	λ	FP (%)	FN (%)	Delay (# of samples)	FP (%)	FN (%)	Delay (# of samples)
0.01	0.03	2.50	4.77	8	0	33.84	61.12
	0.05	2.50	0	8	0	31.12	60.67
	0.07	2.50	0	8	0	31.12	60
	0.1	2.50	0	8	0	28.09	59.04
0.02	0.03	2.78	8.59	8	0	53.46	63
	0.05	2.78	0	8	0	51.93	61.63
	0.07	2.78	0	8	0	51.27	61.28
	0.1	2.78	0	8	0	50.85	61.45
0.04	0.03	8.52	7.48	9.8	2.56	53.53	59.13
	0.05	8.52	0	9	2.56	51.91	57.63
	0.07	8.52	0	8.7	2.56	51.28	61.48
	0.1	8.52	0	8.6	2.56	50.86	61.52
0.07	0.03	9.82	8.46	12	13.43	54.32	48.83
	0.05	9.82	0	10.9	13.43	51.92	52.95
	0.07	9.82	0	9.7	13.43	51.29	52.68
	0.1	9.82	0	9	13.43	50.83	51.75
0.1	0.03	12.28	6.86	16	26.46	54.97	36.03
	0.05	12.28	0	15.7	26.46	52.07	43.93
	0.07	12.28	0	15.1	26.46	51.27	35.13
	0.1	12.28	0	15	26.46	50.84	31.45

TABLE II

Application D1 - Abrupt case - DETECTION RESULTS OF THE PARITY EQUATION APPROACH AND THE PROPOSED CHANGE DETECTION METHODS.

		Proposed approach			Parity equation method		
σ of the noise	Lambda	FP (%)	FN (%)	Delay (# of samples)	FP (%)	FN (%)	Delay (# of samples)
0.01	0.1	1.02	0	12	0.72	36.74	48.3
0.02	0.1	2.19	0.2	12	2.59	41.8	49.3
0.04	0.1	7.92	1.69	14	4.19	44.75	51
0.07	0.1	10.1	4.4	18.1	13.47	48.73	53.6
0.1	0.1	12.34	5.75	21.46	24.45	49.5	55.8

TABLE III

Application D1 - Drift case - DETECTION RESULTS OF THE PARITY EQUATION APPROACH AND THE PROPOSED CHANGE DETECTION METHODS.

Fault type	*Proposed FDS DD (Samples)*	*Parity equation DD (Samples)*
Freezing abrupt additive	12	42
Freezing incipient	12	44
Negative offset abrupt additive	34	-
Negative offset incipient additive	56	-
Positive drift abrupt additive	10	21
Positive drift incipient additive	12	20
Noise abrupt additive	11	275
Noise incipient additive	12	280

TABLE IV

THE DETECTION DELAYS FOR EACH FAULT TYPE IN THE BARCELONA WATER DISTRIBUTION NETWORK TESTBED.

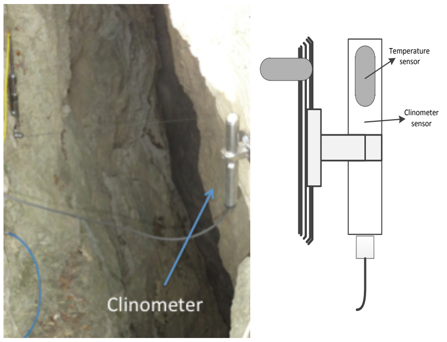

Fig. 6. An example of a intelligent clinometer sensor working in the rock collapse forecasting system deployed in the Alps.

generation of intelligent clinometer sensors (see Fig. 6) which have an internal thermal sensor to correct and compensate on-line the measurements. The goal is to exploit the indirect relationship between the clinometer and temperature sensors to detect faults or changes in $\mathcal{P}$.

In particular, we considered the temperature and clinometer measurements recorded from August 1st 2011 until October 31th 2011. The sampling rate of each sensor is one sample per hour. The dataset is composed of 2100 samples, the first 500 samples composing the training sequence, the next 100 samples used to compute the threshold and the remaining samples constitute the test set. We considered an ARX(1,2) model family as it provides the best reconstruction performance. The window of the CDT has length $N_T = 100$ samples.

Figure 7 presents data and results of both the proposed HMM-based CDT and the parity equation approach. In particular, the upper two subfigures show the temperature and the clinometer measurements while the bottom two show the detections using the HMM-based CDT and residual thresholding method. We observe that the flat part of the signal at about sample 1300 (due to a communication fault) is correctly detected. Interestingly, the detection around sample 1500 can

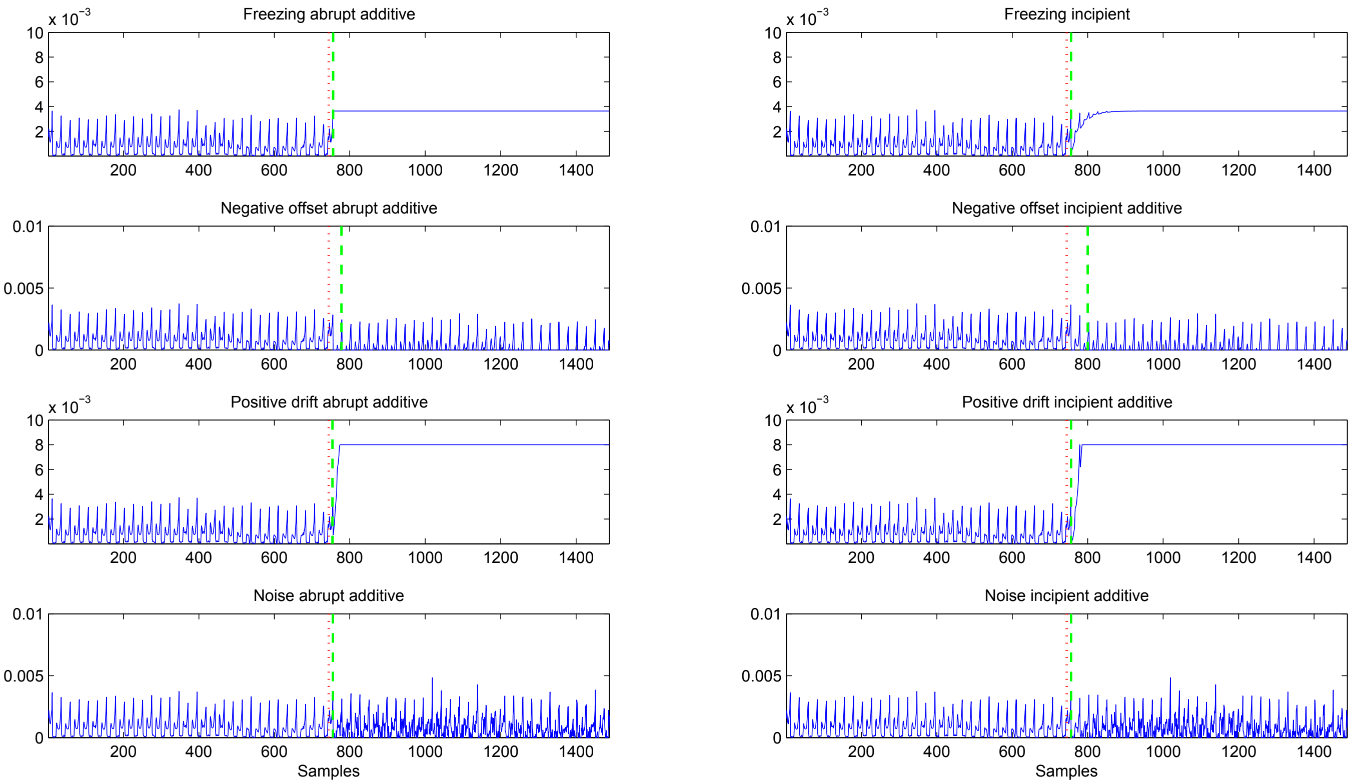

Fig. 5. The 8 datasets of the faulty pump of the Barcelona Water Network simulator. The red dotted lines represent the time instances that the fault affects the pump and the green dashed one represents the time instance a fault is detected by the proposed FDS.

be associated to a transient false detection induced by model bias, while the detections at the end of the experiment can be associated with a change in $\mathcal{P}$. These considerations will be confirmed by the results presented in the recognition scenario in which both the communication fault and the change in the environment will be perceived by the cognitive level. On the contrary the residual based method produces a false positive alarm at value 500 and it detects the environmental changes at the end of the sequence but it fails to detect the communication error.

B. Recognition

We consider a synthetic dataset, and datasets coming from a rock collapse forecasting system and the Great Barrier Reef Ocean Observing System.

1) Synthetic datasets: The synthetic experiment refers to a network composed of $N = 20$ units. Datastreams of the first ten units are modelled as

$$X_j(t) = u_1 \sin(u_2 t) + u_3 \sin(u_4 t) + \epsilon(t),$$

where $1 \leq j \leq 10$, u_1, u_2, u_3 and u_4 are randomly selected in the $[4.5, 5.5]$, $[0.04, 0.06]$, $[2.5, 3.5]$, $[0.08, 0.1]$ intervals, respectively, and $\epsilon(t) \sim N(0, 0.01^2)$ is a zero-mean Gaussian noise.

The remaining 10 units provide data according to models

$$\begin{aligned} X_{10+j}(t) = a_1 X_{10+j}(t-1) + a_2 X_{10+j}(t-2) + \\ b_1 X_j(t-1) + b_2 X_j(t-2) + e(t) \end{aligned}$$

where a_1, a_2, b_1, b_2 are randomly selected in intervals $[0.45, 0.55]$, $[0.15, 0.25]$, $[0.05, 0.15]$, $[0.25, 0.35]$, respectively and $e(t) \sim N(0, \sigma^2)$ is a zero-mean Gaussian noise of variance σ^2. Thus, to generate data we impose 10 relationships where units X_j, $1 \leq j \leq 10$ represent the inputs, while units X_j, $11 \leq j \leq 20$ represent the outputs. The data generation graph is depicted in Fig. 8, while the reduced dependency graph G_R is fully connected and is omitted for brevity.

Each experiment lasts 16000 samples with the first 4000 samples used for training the HMMs. The following 4000 samples are used by each relationship to learn its own threshold (T_h). At sample 12000 a perturbation λ affecting the coefficients of the ARX model is artificially injected. Experiments are averaged over 200 runs.

When an environmental change is considered, the perturbation λ affects all relationships of the data generation graph. We consider both abrupt and drift perturbations to model sudden or slowly drifting changes, respectively. In case of abrupt changes, the parameters of the ARX model move from $\theta = \{a_1, a_2, b_1, b_2\}$ to $\theta_\lambda = \{a_1(1+\lambda), a_2(1+\lambda), b_1(1+\lambda), b_2(1+\lambda)\}$. In case of a drift, the parameters of the ARX model change slowly from θ to θ_λ, which is reached at the end of the experiment. The values of lambda are $\lambda = \{0.01, 0.05, 1\}$.

On the contrary, when a fault is considered, only one sensor is affected by the change under the single fault case assumption. The fault is inject in an a-priori selected sensor (i.e., sensor #20) and its values were artificially increased as

			Proposed approach				K/N equation method			
	σ of the noise	λ	FP (%)	FN (%)	DD (Samples)	CE (%)	FP (%)	FN (%)	DD (Samples)	CE (%)
Abrupt	3	0.01	1.7	1.7	29	94	2.1	10.8	31.92	25
		0.05	1.7	1.6	27.9	94	2.1	11	29.2	28
		0.1	1.7	1.5	25.9	95	2.1	11.3	26.6	30
	4	0.01	2.6	0.7	29.4	95	2.7	16.3	42.9	31
		0.05	2.6	0.6	27.4	97	2.7	16.3	40.9	32
		0.1	2.6	0.5	26.4	98	2.7	16.3	38	32
	5	0.01	3.8	0.5	49.9	91	3.9	16	129	31
		0.05	3.8	0.4	44.2	94	3.9	14.3	117.4	37
		0.1	3.8	0.3	40	95	3.9	13.2	113.8	32
	6	0.01	4.2	0.8	94	92	6.1	21	215	29
		0.05	4.2	0.6	91	93	6.1	20	206	26
		0.1	4.2	0.4	92	91	6.1	19	198	28
Drift	3	0.1	2.2	1.7	29	95	1.8	8.5	31.1	29
	4	0.1	2.5	1.1	29.5	93	1.9	11.7	42.8	31
	5	0.1	3.1	1.5	49.7	91	2.1	14	128.5	27.5
	6	0.1	3.7	1.6	95.2	90	2.5	17.2	215.3	29

TABLE V
CHANGE IN THE ENVIRONMENT (ABRUPT AND DRIFT CHANGES): THE PROPOSED FDS AND THE K/N APPROACH.

			Proposed approach					K/N equation method			
	σ of the noise	λ	FP (%)	FN (%)	DD (Samples)	F (%)	Isolation (%)	FP (%)	FN (%)	DD (Samples)	F (%)
Abrupt	3	0.01	0.7	0	24.2	100	100	3.4	0.5	57	99
		0.05	0.7	0	22	100	100	3.4	0.3	53	99
		0.1	0.7	0	21	100	100	3.4	0.2	50	99
	4	0.01	1.2	0	33	100	100	5.5	1.5	77	99
		0.05	1.2	0	29	100	100	5.5	1.3	71	100
		0.1	1.2	0	24	100	100	5.5	1.2	68	99
	5	0.01	2.8	0	53	100	100	8.1	3	95	100
		0.05	2.8	0	49	100	100	8.1	2.5	87	99
		0.1	2.8	0	44	100	100	8.1	1.7	82	99
	6	0.01	3.2	0	80	100	100	13.4	6.7	126	99
		0.05	3.2	0	78	100	100	13.4	6.5	110	100
		0.1	3.2	0	75	100	100	13.4	6.4	98	100
Drift	3	0.1	1.3	0	24.2	100	100	2.5	2.4	57	100
	4	0.1	2.1	0	33	100	100	3.8	3.6	77	100
	5	0.1	3.4	0	53	100	100	4.9	4.9	95	100
	6	0.1	4.2	0	80	100	100	5.6	6.4	126	100

TABLE VI
FAULT (ABRUPT AND DRIFT CHANGES): THE PROPOSED FDS AND THE K/N APPROACH.

$X_j(t) = X_j(t)(1+\lambda)$ with $\lambda = \{0.01, 0.05, 0.1\}$. Even in this case we considered both abrupt and drift fault cases.

To evaluate the performance of the proposed method we consider the following values of $\sigma = \{3, 4, 5, 6\}$. Results are averaged over 100 runs for each λ and noise level σ.

Results presented in Table V and VI show both the promptness in detecting the changes and the ability to discriminate between faults and changes in the environment by the proposed FDS. In particular, the FDS guarantees lower FP, FN, DD and higher recognition rates w.r.t. the K/N approach.

As expected, FNs and DD decrease with the magnitude of the change λ, while FPs increase with the standard deviation of the noise.

The proposed FDS guarantees effectiveness to correctly discriminate between changes in the environment and faults. In particular, changes in the environment are correctly recognized in the 90-95% of the experiments, while faults are correctly recognized in all cases. Moreover, in case of faults affecting the network the proposed FDS is able to correctly isolate the faulty sensor in all the experiments. The K/N approach is very ineffective when the change affects the environment. The reason of this behavior is due to the fact that the K/N approach requires at least K units detecting simultaneously a change to raise "a detection in the environment" alarm.

Simulation results of the drift case are in line with those of the abrupt change one. Higher FNs and DDs are justified by the smooth effect of the drift change on the parameters of the considered ARX model.

2) Rock collapse forecasting: This experiment refers to the real-world rock collapse monitoring system described in Section V-A3. Here, we consider a system composed of three multi-sensor sensing units. Among other sensors these units are endowed with intelligent clinometer sensors. The indirect relationship between clinometer and temperature sensors are exploited.

All the 30 relationships in the network have cross-correlation larger than 0.5. For this reason, the reduced dependency graph G_R coincides with G and is fully connected.

The results of the proposed FDS, shown in Fig. 9, are particularly interesting since they show two changes in the environment. The change detected approximately between sample 1250 and 1350 is particularly interesting since there the gateway of the monitoring systems was affected by a communication problem inducing a stuck-at fault in the datastreams gathered by the sensing units. This problem has been correctly recognized by the proposed FDS as a change in

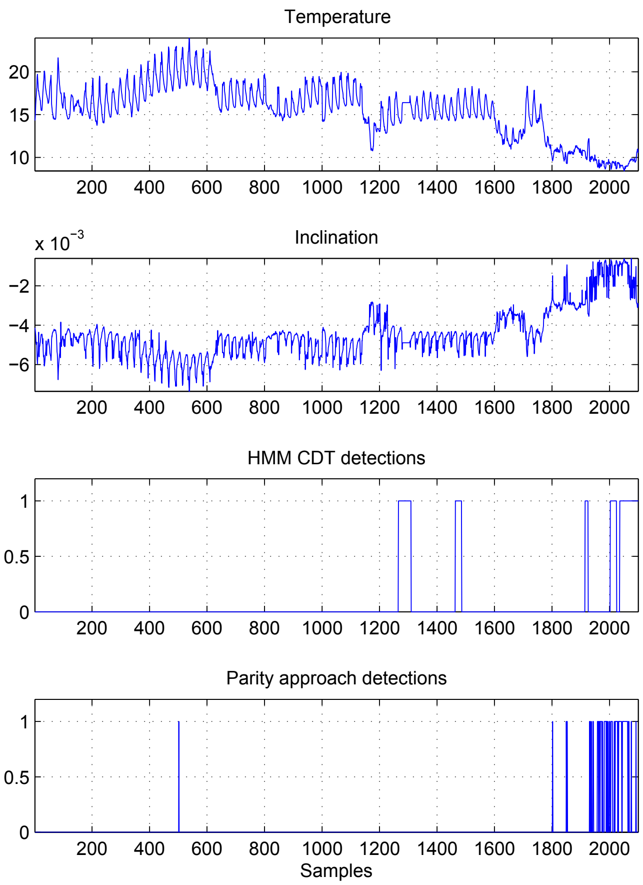

Fig. 7. The application of the proposed method on real-world data coming from the rock collapse deployment. From top to bottom: (a) internal sensor temperature measurements, (b) clinometer measurements, (c) HMM-based CDT detections and (d) Residual based detections.

the environment since all the units have been affected by the change. Interestingly, a communication problem inducing changes in all the network units can be associated with a change in the environment: no much more can be done here. The specific situation can be parallelized with the missing data challenge, where the system provides constant-valued data as long as the malfunction lasts. The second detection is labelled as a temporary environmental change and is in line with the detection experiment described in Section V-A3.

The K/N approach does not detect changes in the environment. This result emphasizes the limits of the K/N approach which suffers from the fact that it requires at least K relationships (i.e., in this case 15 relationships) to detect a change.

3) Great Barrier Reef Ocean Observing System (GBROOS) dataset: The GBROOS dataset refers to the temperature measurements of six units belonging to the Great Barrier Reef Ocean Observation System [42]. The considered units are deployed at the Heron Island, Queensland, Australia and the acquisition campaign ranges from February 21 to March 22, 2009. Interestingly, on March 9, 2009 a Category-4 cyclone affected the deployment area. The dataset lasts approximately 8400 samples and the cyclone occurs at about sample 4900. The sampling period is 5 mins.

Here the training sequence is composed of 3000 samples and the next 500 samples are used to compute the threshold.

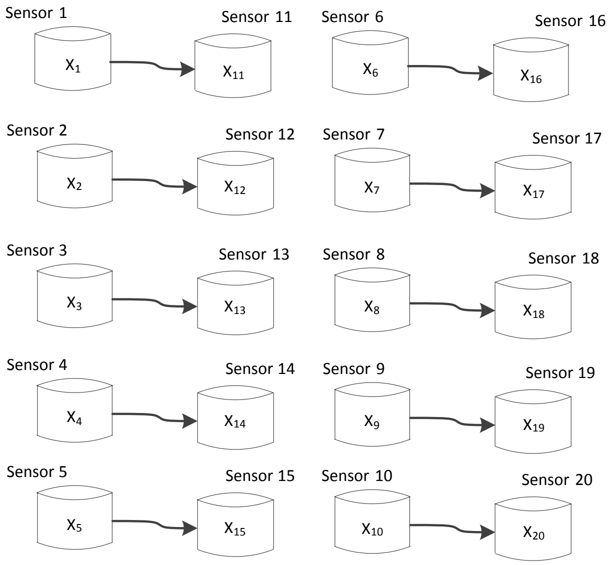

Fig. 8. The data generation graph of the recognition experiment on synthetic data.

The reduced dependency graph is a fully-connected graph since all 30 relationships produced cross-correlation higher than γ_{min}. The results of the application of the proposed FDS on the GBROSS dataset are shown in Figure 10. We observe that

1) the proposed FDS is correctly able to detect the occurrence of a change in the environment approximately at sample 5000. Interestingly, the Australian Institute of Marine Science (AIMS) asserts that the cyclone had not an immediate impact on the air temperature [42] and this might be the reason of the delay in the detection of the change in the environment. Obviously, the aim of the proposed system is not to detect the presence of a cyclone (which is quite evident by itself) but to detect variations in the relationships within a set of sensors. The GBROSS dataset well suits our needs since it provides both real measurements and a ground truth of an occurred change.
2) the change in the environment is only transient, as shown in Figure 10(a). This coincides with the evaluation of AIMS which indicates that the effect of the cyclone on the marine environment is only temporary and that the atmospheric conditions return to normal values within 16 hours [42] after the transit of the cyclone.

Even in this case the K/N approach does not provide satisfactory results since it never happened that at least K (15) relationships raised an alarm at the same time.

VI. Conclusions

The paper introduces a cognitive FDS with novelties in 1) the design of a graph-based representation of the spatial/temporal relationships among measurements in a sensor network and, 2) a two-level hierarchical FDS exploiting such temporal and spatial relationships. The lower level of the FDS relies on a HMM-CDT to promptly detect changes in a

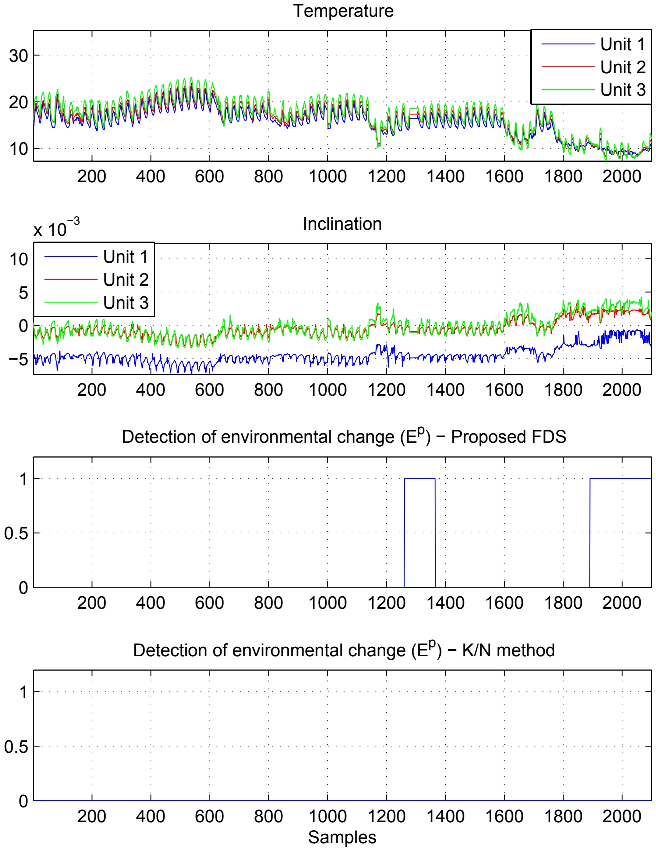

Fig. 9. Results on the data recorded by the rock collapse forecasting system deployed on the Alps. The two upper plots show the temperature and the clinometer measurements, while the lower plots presents the detection of environmental changes (1: detection, 0: not detection) based on the proposed FDS and the K/N method.

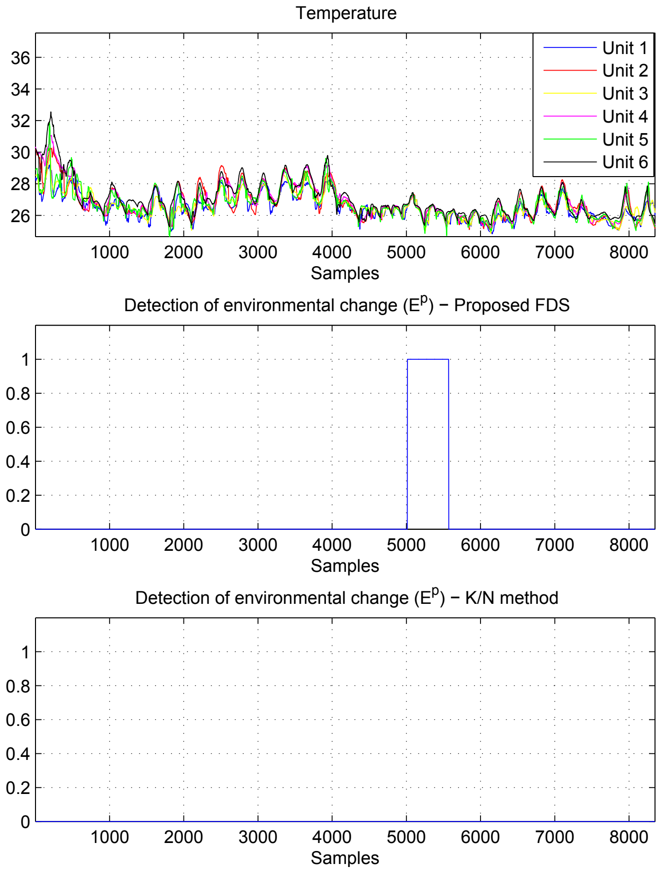

Fig. 10. Results on the GBROOS dataset. The upper plot presents the temperature measurements of the six units. The following plots present detection of environmental changes (1: detection, 0: not detection) based on the proposed FDS and the K/N method.

relationship between two measurement datastreams, while the upper level exploits the graph-based functional representation of the network to discriminate among faults, changes in the environment and false positives induced by model bias. The effectiveness of the proposed solution has been evaluated on both synthetic and real datasets.

The FDS, which is here presented with the detection and isolation phases, can be extended to consider also the identification and accommodation ones.

In fact, a set of HMMs can be trained to identify the occurrence of a fault belonging to a-priori available fault dictionary. Once detected and verified by the upper level, the FDS could select the fault from the fault dictionary associated with the HMM showing the largest likelihood value.

The accommodation phase aims at reducing the effects of a change and adapting the system to the new working conditions. In case of faults, the affected sensor is removed from the reduced dependency graph and, possibly, substituted with its virtual representation (by exploiting the temporal and spatial redundancies with the other sensors). In case of changes in the environment, the whole network representation (i.e., the reduced dependency graph and the HMMs) becomes obsolete and must be retrained from up-to-date data, thus making the FDS adaptive over time. In case of fault detection induced by model bias, the corresponding HMM-CDT could be trained on a larger dataset.

Acknowledgment

This work was supported by the FP7 EU project i-Sense, Making Sense of Nonsense, Contract No: INSFO-ICT-270428.

References

[1] R. Isermann, *Fault-diagnosis systems: an introduction from fault detection to fault tolerance*. Springer Verlag, 2006.

[2] V. Venkatasubramanian, R. Rengaswamy, K. Yin, and S. N. Kavuri, "A review of process fault detection and diagnosis: Part i: Quantitative model-based methods," *Computers Chemical Engineering*, vol. 27, no. 3, pp. 293 – 311, 2003. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0098135402001606

[3] W. Shewhart, *Economic control of quality of manufactured product*. American Society for Quality, 1931, vol. 509.

[4] T. Lai, "Sequential analysis: some classical problems and new challenges," *Statistica Sinica*, vol. 11, no. 2, pp. 303–350, 2001.

[5] M. Basseville, I. Nikiforov *et al.*, *Detection of abrupt changes: theory and application*. Prentice Hall Englewood Cliffs, 1993, vol. 15.

[6] K. Goebel and W. Yan, "Correcting sensor drift and intermittency faults with data fusion and automated learning," *Systems Journal, IEEE*, vol. 2, no. 2, pp. 189–197, 2008.

[7] J. Frolik, M. Abdelrahman, and P. Kandasamy, "A confidence-based approach to the self-validation, fusion and reconstruction of quasi-redundant sensor data," *Instrumentation and Measurement, IEEE Transactions on*, vol. 50, no. 6, pp. 1761–1769, Dec. 2001.

[8] T. Böhme, C. Cox, N. Valentin, and T. Denoeux, "Comparison of autoassociative neural networks and kohonen maps for signal failure detection and reconstruction," *Intelligent Engineering Systems through Artificial Neural Networks*, vol. 9, pp. 637–644, 1991.

[9] M. Napolitano, D. Windon, J. Casanova, M. Innocenti, and G. Silvestri, “Kalman filters and neural-network schemes for sensor validation in flight control systems,” *Control Systems Technology, IEEE Transactions on*, vol. 6, no. 5, pp. 596–611, Sept. 2002.

[10] T. Guo and J. Nurre, “Sensor failure detection and recovery by neural networks,” in *Neural Networks, 1991, IJCNN-91-Seattle International Joint Conference on*, vol. 1, 1991, pp. 221–226.

[11] R. Dunia, S. Qin, T. Edgar, and T. McAvoy, “Identification of faulty sensors using principal component analysis,” *AIChE Journal*, vol. 42, no. 10, pp. 2797–2812, 1996.

[12] S. Wellington, J. Atkinson, and R. Sion, “Sensor validation and fusion using the nadaraya-watson statistical estimator,” in *Information Fusion, 2002. Proceedings of the Fifth International Conference on*, vol. 1, 2002, pp. 321–326.

[13] S. Huang and K. K. Tan, “Fault detection and diagnosis based on modeling and estimation methods,” *Neural Networks, IEEE Transactions on*, vol. 20, no. 5, pp. 872 –881, May 2009.

[14] A. Trunov and M. Polycarpou, “Automated fault diagnosis in nonlinear multivariable systems using a learning methodology,” *Neural Networks, IEEE Transactions on*, vol. 11, no. 1, pp. 91–101, Jan. 2000.

[15] E. Athanasopoulou and C. Hadjicostis, “Probabilistic approaches to fault detection in networked discrete event systems,” *Neural Networks, IEEE Transactions on*, vol. 16, no. 5, pp. 1042–1052, Sept. 2005.

[16] C. Rodriguez, S. Rementeria, J. Martin, A. Lafuente, J. Muguerza, and J. Perez, “A modular neural network approach to fault diagnosis,” *Neural Networks, IEEE Transactions on*, vol. 7, no. 2, pp. 326 –340, Mar. 1996.

[17] Q. Liu, T. Chai, H. Wang, and S.-Z. J. Qin, “Data-based hybrid tension estimation and fault diagnosis of cold rolling continuous annealing processes,” *Neural Networks, IEEE Transactions on*, vol. 22, no. 12, pp. 2284 –2295, Dec. 2011.

[18] R. Aggarwal, Q. Xuan, A. Johns, F. Li, and A. Bennett, “A novel approach to fault diagnosis in multicircuit transmission lines using fuzzy artmap neural networks,” *Neural Networks, IEEE Transactions on*, vol. 10, no. 5, pp. 1214 –1221, Sept. 1999.

[19] D. He, R. Li, J. Zhu, and M. Zade, “Data mining based full ceramic bearing fault diagnostic system using ae sensors,” *Neural Networks, IEEE Transactions on*, vol. 22, no. 12, pp. 2022–2031, Dec. 2011.

[20] M. Seera, C. P. Lim, D. Ishak, and H. Singh, “Fault detection and diagnosis of induction motors using motor current signature analysis and a hybrid fmmcart model,” *Neural Networks and Learning Systems, IEEE Transactions on*, vol. 23, no. 1, pp. 97–108, Jan. 2012.

[21] D. Yang and H. Qi, “An effective decentralized nonparametric quickest detection approach,” in *Pattern Recognition (ICPR), 2010 20th International Conference on*, Aug. 2010, pp. 2278–2281.

[22] M. Chang, A. Terzis, and P. Bonnet, “Mote-based online anomaly detection using echo state networks,” in *Proceedings of the 5th IEEE International Conference on Distributed Computing in Sensor Systems*, ser. DCOSS ’09. Berlin, Heidelberg: Springer-Verlag, 2009, pp. 72–86.

[23] O. Obst, X. Wang, and M. Prokopenko, “Using echo state networks for anomaly detection in underground coal mines,” in *Information Processing in Sensor Networks, 2008. IPSN ’08. International Conference on*, april 2008, pp. 219 –229.

[24] M. Dong, D. Yang, Y. Kuang, D. He, S. Erdal, and D. Kenski, “Pm2.5 concentration prediction using hidden semi-markov model-based times series data mining,” *Expert Systems with Applications*, vol. 36, no. 5, pp. 9046 – 9055, Jul. 2009.

[25] M. Dong and D. He, “Hidden semi-markov model-based methodology for multi-sensor equipment health diagnosis and prognosis,” *European Journal of Operational Research*, vol. 178, no. 3, pp. 858 – 878, May 2007.

[26] ——, “A segmental hidden semi-markov model (hsmm)-based diagnostics and prognostics framework and methodology,” *Mechanical Systems and Signal Processing*, vol. 21, no. 5, pp. 2248 – 2266, Jul. 2007.

[27] M. Maróti, B. Kusy, G. Simon, and A. Lédeczi, “The flooding time synchronization protocol,” in *Proceedings of the 2nd international conference on Embedded networked sensor systems*. ACM, 2004, pp. 39–49. [Online]. Available: http://portal.acm.org/citation.cfm?id=1031501

[28] C. Alippi, G. Boracchi, R. Camplani, A. Marullo, and M. Roveri, “An hybrid monitoring system for the real-time processing of high frequency data. part b: Software aspects,” Tech. Rep., 2011, internal Report.

[29] L. Ljung, “Convergence analysis of parametric identification methods,” *Automatic Control, IEEE Transactions on*, vol. 23, no. 5, pp. 770 – 783, Oct. 1978.

[30] L. Ljung and P. E. Caines, “Asymptotic normality of prediction error estimators for approximate system models,” in *Decision and Control including the 17th Symposium on Adaptive Processes, 1978 IEEE Conference on*, vol. 17, Jan. 1978, pp. 927–932.

[31] G. Huang, Q. Zhu, and C. Siew, “Extreme learning machine: theory and applications,” *Neurocomputing*, vol. 70, no. 1, pp. 489–501, Dec. 2006.

[32] B. Schrauwen, D. Verstraeten, and J. Van Campenhout, “An overview of reservoir computing: theory, applications and implementations,” in *Proceedings of the 15th European Symposium on Artificial Neural Networks*, 2007, pp. 471–482. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.95.1215

[33] D. A. Reynolds, T. F. Quatieri, and R. B. Dunn, “Speaker verification using adapted gaussian mixture models,” *Digital Signal Processing*, vol. 10, no. 1-3, pp. 19–41, 2000. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1051200499903615

[34] L. R. Rabiner and B. H. Juang, “An introduction to hidden markov models,” *IEEE ASSP Magazine*, pp. 4–15, Jan. 1986.

[35] R. Durbin, S. R. Eddy, A. Krogh, and G. Mitchison, *Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids*. Cambridge University Press, Jul. 1998. [Online]. Available: http://www.amazon.ca/exec/obidos/redirect?tag=citeulike09-20&amp;path=ASIN/0521629713

[36] C. Alippi, G. Boracchi, and M. Roveri, “On-line reconstruction of missing data in sensor/actuator networks by exploiting temporal and spatial redundancy,” in *Neural Networks (IJCNN), The 2012 International Joint Conference on*. IEEE, 2012, pp. 1–8.

[37] S. Mirsaidi, G. Fleury, and J. Oksman, “Lms-like ar modeling in the case of missing observations,” *Signal Processing, IEEE Transactions on*, vol. 45, no. 6, pp. 1574–1583, 1997.

[38] F. Ding and J. Ding, “Least-squares parameter estimation for systems with irregularly missing data,” *International Journal of Adaptive Control and Signal Processing*, vol. 24, no. 7, pp. 540–553, 2010.

[39] Torch machine learning library. [Online]. Available: http://www.torch.ch/

[40] J. Quevedo, V. Puig, G. Cembrano, J. Blanch, J. Aguilar, D. Saporta, G. Benito, M. Hedo, and A. Molina, “Validation and reconstruction of flow meter data in the Barcelona water distribution network,” *Control Engineering Practice*, vol. 18, pp. 640–651, Jun. 2010.

[41] C. Alippi, G. Boracchi, A. Marullo, and M. Roveri, “A step towards the prediction of a rock collapse: analysis of micro-acoustic bursts,” in *Sensors, 2011 IEEE*, Oct. 2011, pp. 1273–1276.

[42] J. Bezdek, S. Rajasegarar, M. Moshtaghi, C. Leckie, M. Palaniswami, and T. Havens, “Anomaly detection in environmental monitoring networks [application notes],” *Computational Intelligence Magazine, IEEE*, vol. 6, no. 2, pp. 52–58, May 2011.