On-line Reconstruction of Missing Data in Sensor/Actuator Networks by Exploiting Temporal and Spatial Redundancy

Cesare Alippi, Giacomo Boracchi, Manuel Roveri
Politecnico di Milano, Dipartimento di Elettronica e Informazione, Milano, Italy
{alippi,boracchi,roveri}@elet.polimi.it

***Abstract*—Data streams from remote monitoring systems such as wireless sensor networks show immediately that the "you sample you get" statement is not always true. Not rarely, the data stream is interrupted by intermittent communication or sensors faults, resulting in missing data in the received sequence. This has a negative impact in many algorithms assuming continuous data stream; as such, the missing data must be suitably reconstructed, in order to guarantee continuous data availability. We suggest a general methodology for reconstructing missing data that exploits both temporal and spatial redundancy characterizing the phenomenon being monitored and the distributed system, a situation proper of many monitoring systems constituted by sensor and actuator networks. Temporal and spatial dependencies are learned through linear and non-linear non-parametric models, also encompassing neural -possibly recurrent- networks, which become the spatial transfer functions connecting the different views of the phenomenon under investigation. Missing data are finally reconstructed by exploiting the forecasting ability provided by such transfer functions. The experimental section shows the effectiveness of the proposed methodology.**

***Index Terms*—Missing data; non-linear reconstruction; fault accommodation; distributed monitoring systems, recurrent neural networks.**

I. Introduction

In real-world distributed monitoring systems, permanent or transient faults can affect units, sensors or transmission lines so as to induce fault-affected (perturbed) or missing data. For instance, embedded electronics can be affected by faults inducing errors in the measurements processing, while sensors might suffer from ageing effects and thermal drift that slowly change their behaviors over time. Faults at the physical, network or transmission level, as well as lack of energy at the units result in communication errors, which prevent units and their base stations for sharing data and commands required for continuous communication. Finally, software faults (i.e., bugs in the software) might induce unpredictable behaviors at the sensing units in specific conditions, as well as incorrect interpretations at the remote control room.

Perturbed, incorrect and missing data can heavily affect the subsequent analysis and control phases so as to possibly induce incorrect decisions or on-the-field reactions. Distributed monitoring systems designed to work in real-life scenarios must thus be able to deal with these perturbed values or missing data to guarantee, over time, the quality-of-service of the application.

Though the ability to detect and identify perturbed values is an interesting and challenging issue (we invite the reader to refer to [1], [2] for a review), here we focus on the missing data aspect. Furthermore, what here proposed is tailored for sensor/actuator networks where the number of units is reduced and insufficient to reconstruct the overall function representing the monitored (physical) phenomenon. In the case of many units or when assumptions about the regularity of the underlying function can be made, ad-hoc function-reconstruction techniques such as those based on neural networks (e.g., [3]) can be considered to solve the problem.

Several reconstruction techniques able to fill missing data in distributed monitoring applications have been presented in the literature. The most immediate solutions carry out a sensor replication scheme at the units: a redundant number of sensors guarantees both robustness in mission-critical applications and reconstruction of missing data (thus assuming that the data are lost at the sensor level). The drawback is that the complexity and the cost of the sensing units increase, and become unacceptably high whenever the application requires non silicon-integrated (low-cost) sensors. Still, communication faults are not covered by this mechanism.

More advanced techniques, e.g., see [4]–[9], reformulate the reconstruction of missing data as a forecasting problem. In this direction, a recurrent algorithm for the reconstruction of missing data in auto-regressive (AR) models is suggested in [4]. The algorithm proposed in [5] exploits a least-square recurrent estimate of the parameters of output-error (OE) models in case of irregularly missing output data. In [6], missing data are reconstructed through a weighted average of the estimates of two AR models operating forward and backward in time so as to fill the "information hole". A validation and reconstruction framework for flowmeter data in a water distribution network is described in [7]. There, temporal redundancy is exploited to reconstruct missing data by combining the time series analysis (i.e., an autoregressive integrated moving average ARIMA model) with the short-term prediction of the water consumption. The solutions presented in [8], [9] rely on a parameter-estimation technique for auto-regressive models with exogenous inputs (ARX) subject to missing data. There, the idea is to construct a state-space

formulation of the system and apply the Kalman filter in case of on-line reconstruction and a fixed-interval smoother in case of batch (or off-line) reconstruction.

Several solutions adopt neural networks for reconstructing missing data in multivariate time-series, e.g., see [10] and [11]. In particular, [10] suggests to use recurrent neural networks to reconstruct missing values, while the use of radial basis function neural networks is proposed in [11].

A different approach is suggested in [12]–[14] where the addressed problem requires either data regularization or compensation. A reconstruction algorithm for hydrometric time series based on AR models and Kalman filters has been proposed in [12], while [13] suggests a fuzzy solution exploiting redundant sensors. Again, [14] exploits Kohonen maps and spatial correlation to reconstruct corrupted data. There, the corrupted value at a sensing unit is reconstructed as the combination of the k nearest prototypes in the Kohonen map.

In contrast to approaches where a specific model hierarchy is considered, here we propose a general methodology for *on-line reconstruction* of missing data in distributed monitoring systems where the number of acquired units is limited and no hypothesis about the phenomenon being monitored is assumed. This approach encompasses state space and input-output linear and nonlinear models (including recurrent neural networks, NARX, NARMAX, NOE, where the nonlinear component is modeled by feedforward neural networks). The core idea is to exploit temporal and spatial redundancy among the sensing units showing a correlated view of the monitored phenomenon. Temporal and spatial dependencies can be jointly exploited by generating models providing the *transfer functions* connecting different sensing units deployed in the same environment. Each estimated transfer function is then considered to explain the spatial relationship between two (or more) units in the network, and during the operational life the transfer function provides predictions for the missing values. In addition, we consider the propagation delay of the phenomenon being monitored among the sensing units, thus estimating the *dependency graph*, which allows us to neglect those units that do not bring up-to-date information for missing data reconstruction (causality).

Fig. 1 illustrates a simple example of reconstruction of missing data at Unit 1 at time t: Unit 2 and 3 provide their estimates $\widehat{X}_{2,1}(t)$ and $\widehat{X}_{3,1}(t)$ for the missing value $X_1(t)$, through the estimated transfer functions $f_{2,1}$ and $f_{3,1}$, respectively. More advanced solutions, e.g., encompassing aggregation of multiple estimates or multiple-inputs reconstruction algorithms, are described in Section III.

Although the paper focuses on the reconstruction of missing data, the same methodology can be used to detect faults or changes in the environment by inspecting the residuals between the values predicted by the transfer functions and the measurements in each unit. This aspect can be further extended to monitor, besides the relationship among different units, also the sequence of measurements in each unit in a stand-alone manner, so as to speculate between fault in a single unit or a change in the environment.

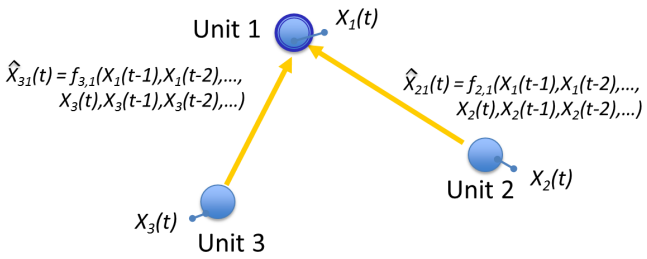

Fig. 1: Reconstruction of a missing measurement $X_1(t)$ at sensing Unit 1 and time t. Units 2 and 3 provide two estimates $\widehat{X}_{2,1}(t)$ and $\widehat{X}_{3,1}(t)$ obtained by feeding the transfer functions with past measurements coming from Unit 1 and the current and previous data coming from Units 2. and 3

The paper is organized as follows. Section II introduces the problem statement, while the proposed methodology for the on-line reconstruction is presented in Section III. Section IV defines the dependency graph and describes its use for the reconstruction of missing data. Experimental results are shown in Section V.

II. Problem Statement

Let us consider a distributed monitoring system composed of N sensing units and define $X_i(t) \in \mathbb{R}$, with $1 \leq i \leq N$, a measurement acquired by the i-th sensing unit at time t (units are assumed here to gather synchronous acquisitions). In line with the framework presented in [12] we define the variable

$$k_i(t) = \begin{cases} 0, & \text{if } X_i(t) \text{ is missing;} \\ 1, & \text{otherwise.} \end{cases}$$

to model missing data. When all data are correctly received, $k_i(t) = 1, \quad \forall t > 0$. Random missing data (e.g., due to intermittent communication errors) can be modeled by considering $k_i(t)$ as a random variable following the Bernoulli distribution, i.e., $k_i(t) \sim B(p_i)$ where p_i is the probability of missing a measurement at time instant t in the i-th sensing unit. A finite sequence of missing data (e.g., due to a transient fault affecting the units) is thus modeled by defining a temporal profile for $k_i(t)$ through two time instants t_{init} and t_{end} for which

$$k_i(t) = \begin{cases} 0, & t_{init} \leq t \leq t_{end} \\ 1, & \text{elsewhere.} \end{cases}$$

In the following, we assume that an initial training sequence $TS_j = \{X_j(1), \ldots, X_j(t_0)\}$ is available for each unit $1 \leq j \leq N$ and that the training sequences of different units have the same length. We further assume that the relationships associated with measurements coming from any pair of units are time-invariant in the training sequence. To ease the description, we further assume that at each time instant t only one unit may be affected by a fault resulting in data loss (single fault assumption). Let us assume that the i-th unit is affected by a fault at time t (i.e., $k_i(t) = 0$), and denote by $\widehat{X}_i(t)$ the best forward estimate when $X_i(t)$ is missing. In this case $\widehat{X}_i(t)$ is

TABLE I: Model hierarchies considered in the methodology.

	Linear		*Non-linear*	
	Model type	*Estimation technique*	*Model type*	*Learning technique*
Input-output	ARX	Non-iterative least-square method	NARX	Levenberg-Marquardt
	ARMAX	Iterative prediction error method	NARMAX (FFNN model)	
	OE	Algorithm minimizing prediction errors	NOE (RNN model)	Recurrent Levenberg-Marquardt
State space models (SSMs)	Linear SSMs	N4SID [15]	Nonlinear SSMs (RNN & FFNN model)	Recurrent Levenberg-Marquardt

considered to fill the gap, otherwise the unit measurement is kept; the reconstruction process can thus be summarized as:

$$X_i(t) = k_i(t)X_i(t) + (1 - k_i(t))\widehat{X}_i(t).$$

Thus, at time t, $X_i(t)$ is either the acquired measurement $X_i(t)$ or its best estimate $\widehat{X}_i(t)$. The estimate $\widehat{X}_i(t)$ is typically a forward estimate, which can be possibly improved offline, as soon as the connection is re-established, e.g., by following the backward approach presented in [6].

We emphasize that the best forward estimate $\widehat{X}_i(t)$ could be considered also when noisy measurement $X_i(t)$ are processed by regularization techniques (e.g., [13], [14]) into $X_i^r(t)$, i.e.,

$$X_i(t) = k_i(t)X_i^r(t) + (1 - k_i(t))\widehat{X}_i(t).$$

III. The Proposed Methodology

The estimate $\widehat{X}_i(t)$ can be derived by suitably fusing the predicted values $\widehat{X}_{j,i},\ j = 1, \ldots, N, j \neq i$ of $X_i(t)$ provided by the transfer functions $f_{j,i}$ connecting the generic j-th unit with the i-th faulty one. The considered time-invariant dynamic models, together with the corresponding estimation techniques [15], are reported in Table I: the transfer functions are obtained by training these models on the training sequences using the learning techniques described in Table I, and by selecting the best one for predicting $\widehat{X}_{j,i}$. Linear, non-linear, input-output and state-space models (SSMs) have been also considered to enrich the modeling expressive power of the considered model hierarchies.

The procedure for identifying the best transfer function $f_{j,i}$ for the j-th unit, provided from a training set, is rather standard and is given in Algorithm 1, where a model hierarchy is at first selected, the model family chosen and the trained model finally derived. The algorithm relies on the sample-partitioning approach where the available data set is partitioned into training S_D and validation S_E sets. The best estimate $\widehat{X}_i(t)$ can now be derived from the above models by following three different strategies which automatically fuse the spatial redundancy information. Such strategies will be presented in the subsequent subsections.

input : a training set, a set of model hierarchies, the model order
output: the model minimizing the validation error

Partition the available data set TS_j into S_D and S_E;
for *each* model hierarchy *in the set of model hierarchies* **do**
  for *each* model family *in the* model hierarchy **do**
    Estimate the parameters $\widehat{\theta}$ of the *model* minimizing the *one-step-ahead* prediction error over S_D

$$\widehat{\theta} = \underset{\theta}{\operatorname{argmin}}\,[v(\theta)] \text{ with}$$

$$v(\theta) = \frac{1}{|S_D|}\sum_{t=1}^{|S_D|}\left(X_i(t) - \widehat{X}_i(t)\right)^2;$$

    Compute the validation error of $\widehat{\theta}$ over S_E;
  end
end
Select the best model as the one minimizing the validation error over S_E ;

Algorithm 1: The algorithm to identify the best model $f_{j,i}$

A. Solution A: Best Couple

Generate, for each sensing unit i, $N-1$ single-input single-output (SISO) models $f_{j,i}, j = 1, \ldots, N\ j \neq i$, and select $f_{\bar{j},i}$ corresponding to the unit providing the highest reconstruction ability, i.e. i and $\bar{j}$ are the *best couple*. For each pair (i, j), the SISO model $f_{j,i}$ is estimated as in Algorithm 1, setting TS_i as output stream and TS_j, $1 \leq j \leq N$ and $j \neq i$ as the input one. Only the SISO model $f_{\bar{j},i}$ guaranteeing the lowest validation error on S_E is considered for describing the behavior of unit i; the SISO models obtained from other units are discarded. During the operational modality, missing data at unit i at time t are reconstructed as

$$\begin{aligned}\widehat{X}_i(t) = f_{\bar{j},i}(&X_i(t-1), X_i(t-2), \ldots, X_i(t-\tau_i),\\ &X_{\bar{j}}(t), X_{\bar{j}}(t-1), X_{\bar{j}}(t-2), \ldots, X_i(t-\tau_{\bar{j}}))\end{aligned}$$

where τ_i and $\tau_{\bar{j}}$ are the orders of the autoregressive and the exogenous components, respectively. When a sequence of data is missing (e.g., $k_i(t) = 0,\ t \in [t_{init}, t_{end}]$), the past reconstructed data are considered as a correct measurements for reconstructing the next missing measurements, i.e., $\widehat{X}_i(t_{init} + \nu)$, with $\nu \leq t_{end} - t_{init}$, is given by

$$\begin{aligned}\widehat{X}_i(t_{init} + \nu) = f_{\bar{j},i}(&\widehat{X}_i(t_{init} + \nu - 1), \ldots, \widehat{X}_i(t_{init}),\\ &X_i(t_{init} - 1) \ldots, X_i(t - \tau_i),\\ &X_{\bar{j}}(t), X_{\bar{j}}(t-1), X_{\bar{j}}(t-2), \ldots, X_i(t-\tau_{\bar{j}})).\end{aligned}$$

A graphical description of the best couple solution is presented in Fig. 2.

The best couple solution is optimal in case of linear models under the identifiability hypothesis. Other solutions could rely on the Output Error model, which generally provides a more

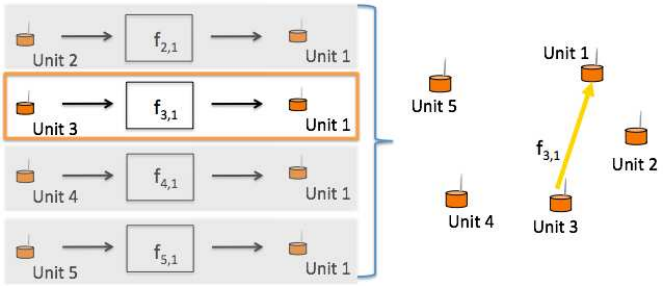

Fig. 2: Solution "A": best couple. Unit 1 is affected by missing data and the unit guaranteeing the highest reconstruction ability is Unit 3 (i.e., $\bar{j} = 3$), then only the transfer function $f_{3,1}$ is used to predict missing data at Unit 1.

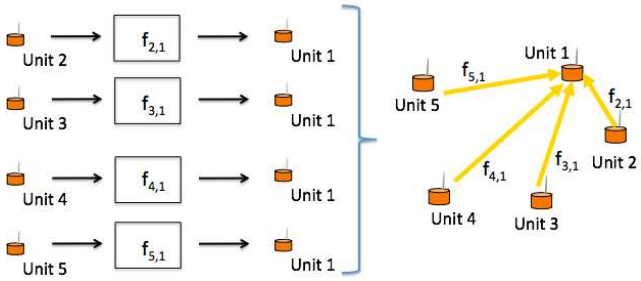

Fig. 3: Solution "B": weighted instances. The estimates of all the transfer functions $\{f_{j,1}\}_{j\neq 1}$ are averaged to estimate missing data at Unit 1.

stable estimate of the parameters (in case of nonlinear models). These considerations, which are here described for solution "A", are obviously valid also for Solution "B" and "C", presented in the sequel.

B. Solution B: Weighted Instances

A different method for reconstructing a missing value would consider the use of all the generated models suitably weighted. More in details, the missing measurement at Unit i at time t is computed as a weighted average of the estimates provided by the $N-1$ SISO models $f_{j,i}, j = 1, \ldots, N,\ j \neq i$ estimated as in Algorithm 1. The weights could be function of the validation errors (e.g., high validation errors would result in low weights) or of the correlation (e.g., high correlation among the units results in high weights). The missing value is reconstructed as

$$\widehat{X}_i(t) = \sum_{\substack{j=1 \\ j\neq i}}^{N} w_j f_{j,i}(X_i(t-1), X_i(t-2), \ldots, X_i(t-\tau_i), X_j(t), X_j(t-1), X_j(t-2), \ldots, X_i(t-\tau_j)),$$

with $\sum_{j\neq i} w_j = 1$.

An example of weighted average is presented in Fig. 3 where the missing value at Unit 1 is reconstructed by means of the weighted instances of the estimates provided by Unit 2, 3, 4 and 5.

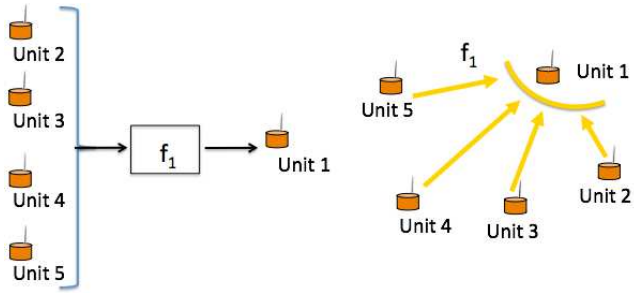

Fig. 4: Solution "C": multiple inputs. Measurement from all the units are considered into a MISO model to predict missing values at Unit 1.

C. Solution C: Multiple Inputs

Here, instead of considering $N-1$ SISO models we use a more general MISO model receiving inputs from all the units. We generate the MISO model for the i-th Unit by considering as output TS_i and as input all the other TS_j, $j = 1, \ldots, N,\ j \neq i$. During the operational modality each missing value at Unit i is estimated as

$$\widehat{X}_i(t) = f_i(X_1(t), X_1(t-1), X_1(t-2), \ldots, X_1(t-\tau_1), \ldots, X_i(t-1), X_i(t-2), \ldots, X_i(t-\tau_i), \ldots, X_N(t), X_N(t-1), X_N(t-2), \ldots X_N(t-\tau_N)).$$

Fig. 4 shows a graphical reconstruction of Unit 1 by considering a single MISO model in which Unit 2, 3, 4 and 5 represent the inputs.

IV. The Dependency Graph

In the above we considered, for data reconstruction purposes, a situation where each i-th unit was connected with $N-1$ units, thus assuming a full unit dependency over the network. However, it is convenient to consider only the most meaningful N_i units ($N_i \leq N-1$) for reconstructing data at the i-th unit: in fact, causality and correlation allow us for limiting the units constituting the dependency cluster to N_i units only.

This situation is rather common in distributed monitoring systems where the units sense the physical phenomenon (which is generally time-dependent) with intrinsic delays, depending on the dynamics of the phenomenon and the location of the units. For example, if the system aims at measuring temperature and luminosity, the influencing sequence depends both on the units position and the trajectory of the sun. Form this example it emerges clearly that the first unit radiated by the sun is able to provide information to all the other units, which, in turn, enforcing a predictive model, may forecast their future measurements (the characteristics of the deployment area and the distance among the units influence the propagation of the phenomenon). Of course, due to causality, the last unit radiated by the sun cannot provide

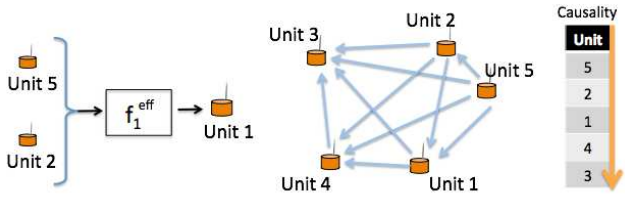

Fig. 5: Exploiting the dependency graph to reconstruct data.

useful information to other units and the first radiated unit cannot exploit measurements from other units.

It comes out that there are two important and related steps that need to be tackled: identifying the expected propagation delay of the phenomenon between two generic units, and generating the causal dependency graph.

To evaluate the temporal dependency among the units we estimate the expected delay $\tau_{i,j}$ between two generic units i and j by considering the value of the lag for which the cross correlation of their measurements is maximum. When $\tau_{i,j}$ is positive, the Unit i is in advance w.r.t. j it may provide useful information to Unit j. On the contrary, when $\tau_{i,j} < 0$, the measurements at Unit i are delayed w.r.t. Unit j and Unit i cannot deliver useful information to Unit j.

A temporal *dependency graph* can then be generated completing the causality graph with delays information. An example of dependency graph is presented in Fig. 5 where Unit 5 is the first unit perceiving the physical phenomenon and Unit 3 the last.

The N_i units to be considered for predicting missing data at the i-th Unit are straightforwardly selected from the dependency graph. Referring to Fig. 5, to reconstruct missing data at Unit 1 we consider only the measurements from Units 2 and 5, which constitute S_1, the *effective subset* for Unit 1, thus $N_1 = 2$. The effective subset S_i of Unit i contains Units j_1^i, ..., $j_{N_i}^i$, i.e., $S_i = \{j_1^i, \ldots, j_{N_i}^i\}$, that may provide useful information for reconstructing missing data at Unit i. Then, a MISO model f_i^{eff} can be generated by considering TS_i as output and the training sequences of the units belonging to S_i as inputs. It follows that, during the operational life, a missing measurement in the i-th unit at time t can be reconstructed by relying on f_i^{eff} that takes as inputs the previous measurements of Unit i and the current and the previous measurements of units in S_i, i.e.,

$$\widehat{X}_i(t) = f_i^{\text{eff}}(X_i(t-1), X_i(t-2), \ldots, X_i(t-\tau_i), X_{j_1^i}(t), X_{j_1^i}(t-1), X_{j_1^i}(t-2), \ldots, X_{j_1^i}(t-\tau_{j_1^i}), \ldots, X_{j_{N_i}^i}(t), X_{j_{N_i}^i}(t-1), X_{j_{N_i}^i}(t-2), \ldots, X_{j_{N_i}^i}(t-\tau_{j_{N_i}^i})).$$

V. Experiments

To evaluate the effectiveness of the suggested reconstruction methodology we considered both synthetically generated data (Application D1 and D2) and measurements from a rock collapse forecasting system deployed in the Alps, Italy (Application D3).

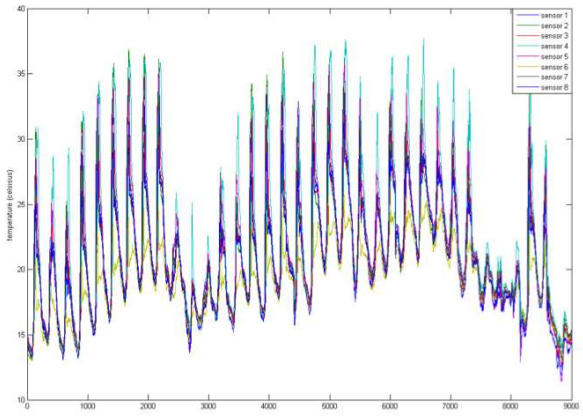

Fig. 6: Application $D3$: real measurements from 8 temperature sensors acquired from the rock collapse forecasting system deployed on the St. Martino mount, northern Italy.

Application D1 and $D2$ refer to linear and nonlinear synthetically generated data, respectively. Both applications consider $N = 4$ units sharing the same SISO model family and the same input signal $u(t)$, but they differ for the model parameters. In particular, Application $D1$ refers to the linear state space model (the model is SISO, hence a_i and b_i are scalar values)

$$\begin{aligned} x_i(t+1) &= a_i x_i(t) + b_i u(k), \\ y_i(t) &= x_i(t) + \eta_i, \end{aligned}$$

while Application $D2$ refers to the nonlinear state space model

$$\begin{aligned} x_i(t+1) &= a_i x_i(t) + b_i u(k), \\ y_i(t) &= \frac{1}{2}(x_i(t) + e^{a_i x_i(t)}) + \eta_i, \end{aligned}$$

where in both cases a_i and b_i are drawn from uniform distribution within $[0.2, 0.3]$ and $[0.5, 0.6]$, respectively, η_i is white Gaussian noise $\eta_i \sim \mathcal{N}(0, 0.02)$, and $x(0) = 0$. The input signal $u(t)$ is the set of temperature measurements in the Sensor 1 of the monitoring system described in application $D3$.

Application $D3$ refers to measurements acquired by the rock collapse forecasting monitoring system deployed on the St. Martino Mount, Lecco, Nothern Italy [16]. In particular, we consider the temperature sensors of the $N = 8$ sensing units. The dataset, depicted in Fig. 6, is composed of approximately 9000 samples covering 36 days of acquisition (from *May, 16*th, *2010* to *June, 21*th, *2010*; the acquisition rate is one sample every five minutes[1]. The first 1000 samples have been used to train the missing data reconstruction procedure, while the performance are evaluated by removing one sample at-a-time in the remaining sequence, and by computing the *1-step ahead* prediction.

[1]These dataset are available for download at http://home.dei.polimi.it/roveri/software/

We assess the effectiveness of our solutions through two figures of merit concerning the reconstruction accuracy and the whiteness of the residuals. In particular, we consider:

- the *prediction fit* defined in [15] to evaluate the reconstruction accuracy, i.e.,

$$prediction\ fit = 100 \times \left(1 - \frac{\sqrt{\sum_{t=t_0}^{T} (X_i(t) - \widehat{X}_i(t))^2}}{\sqrt{\sum_{t=t_0}^{T} \left(X_i(t) - \bar{X}_i\right)^2}} \right),$$

  where $\bar{X}_i = \sum_{t=t_0}^{T} X_i(t)$. The index ranges from $-\infty$ to 100 (perfect fit);
- the *prediction residual autocorrelation index* to evaluate the whiteness of the residual. It represents the core of the Anderson whiteness test [15], and it is defined as

$$\frac{\max_{\tau>0} |r_\epsilon(\tau)|}{p_{1-\alpha}},$$

  where $r_\epsilon(\tau)$ is the autocorrelation function at lag τ of the residual $\epsilon(t) = X_i(t) - \widehat{X}_i(t)$ and $p_{1-\alpha}$ is the maximum of the confidence interval defined by the Anderson whiteness test at confidence $1-\alpha$. This index ranges from 0 to $+\infty$ (values between 0 and 1 corresponds white noise, the higher the number the more biased is the residual).

Simulation results of Application D1 are presented in Tables II and III for solutions "A" (best couple) and "C" (multiple inputs), respectively. In particular, Table II(a) shows the best linear models for each couple of sensors. Table II(b) and (c), presenting the prediction fit and the whiteness index for the best linear models, show that the reconstruction accuracy is high and that the whiteness indexes lie below 1, thus indicating that the underlying model has been effectively estimated. Table III show that the MISO models estimated according to Solution "C" achieve reconstruction accuracy similar to those of the SISO models of Solution "A". Solution "B" provided the lowest performance both with uniform weights and weights proportional to the units correlation, and its experimental results are omitted for brevity. The drawbacks of this solutions are twofold: at first, the average of the SISO estimates provides a lower performance compared to the best couple estimate, and second, the multiple input solution better exploits the correlation among the measurements.

Tables IV and V show the simulation results for the nonlinear dataset of Application D2. In particular, Table IV presents the best linear models for Solution "A" and the corresponding prediction fit and whiteness index. As expected, linear models in Application D2 are not able to fully identify the system (which shows a strong nonlinearity) since the residual is not white. If we consider nonlinear models the results improve significantly, as shown in Table V where the nonlinear NARX model (which exploits a feedforward neural network for the nonlinear part), is considered. As expected, a nonlinear model is able to provide higher performance than linear models since it is able to properly identify the nonlinear relations between the data: prediction fits are higher than in Table IV and the residual is now white.

Tables VI and VII show that neither linear and non-linear

	Sensor 1	Sensor 2	Sensor 3	Sensor 4
Sensor 1	-	ARMAX	ARMAX	ARMAX
Sensor 2	ARMAX	-	ARMAX	ARMAX
Sensor 3	ARMAX	ARX	-	SSM
Sensor 4	OE	ARMAX	ARX	-

(a) Best linear models.

	Sensor 1	Sensor 2	Sensor 3	Sensor 4
Sensor 1	-	97.34	97.26	97.34
Sensor 2	97.23	-	97.18	97.03
Sensor 3	97.14	97.17	-	97.19
Sensor 4	97.22	97.04	97.15	-

(b) 1-step ahead prediction fit

	Sensor 1	Sensor 2	Sensor 3	Sensor 4
Sensor 1	-	0.7286	0.7918	0.5070
Sensor 2	0.7019	-	0.5570	0.5410
Sensor 3	0.6321	0.7814	-	0.4435
Sensor 4	0.7421	0.5873	0.5119	-

(c) 1-step ahead whiteness index

TABLE II: Application D1 - Solution "A".

	Sensor 1	Sensor 2	Sensor 3	Sensor 4
N-1	ARMAX	ARMAX	OE	OE

(a) Best linear models.

	Sensor 1	Sensor 2	Sensor 3	Sensor 4
N - 1	97.77	97.60	97.60	97.58

(b) 1-step ahead prediction fit

	Sensor 1	Sensor 2	Sensor 3	Sensor 4
N - 1	0.8140	0.9471	0.9537	0.8347

(c) 1-step ahead whiteness index

TABLE III: Application D1 - Solution "C".

	Sensor 1	Sensor 2	Sensor 3	Sensor 4
Sensor 1	-	ARMAX	ARMAX	ARMAX
Sensor 2	SSM	-	ARX	ARMAX
Sensor 3	ARMAX	ARMAX	-	ARX
Sensor 4	ARMAX	ARMAX	ARX	-

(a) Best linear models.

	Sensor 1	Sensor 2	Sensor 3	Sensor 4
Sensor 1	-	97.10	96.05	97.33
Sensor 2	96.99	-	95.82	97.12
Sensor 3	95.21	95.27	-	95.39
Sensor 4	97.22	97.14	95.97	-

(b) 1-step ahead prediction fit

	Sensor 1	Sensor 2	Sensor 3	Sensor 4
Sensor 1	-	1.096	1.572	1.073
Sensor 2	1.586	-	1.553	1.178
Sensor 3	1.899	1.130	-	1.402
Sensor 4	1.211	1.031	1.235	-

(c) 1-step ahead whiteness index

TABLE IV: Application D2 - Solution "A": linear models

	Sensor 1	Sensor 2	Sensor 3	Sensor 4
Sensor 1	-	97.54	96.86	97.43
Sensor 2	97.40	-	96.20	97.23
Sensor 3	96.03	95.65	-	95.94
Sensor 4	97.44	97.27	96.32	-

(a) 1-step ahead prediction fit

	Sensor 1	Sensor 2	Sensor 3	Sensor 4
Sensor 1	-	0.6596	1.2010	0.9829
Sensor 2	0.6050	-	0.6486	0.7906
Sensor 3	0.8908	0.9699	-	0.5366
Sensor 4	0.7893	0.9153	0.9067	-

(b) 1-step ahead whiteness index

TABLE V: Application D2 - Solution "A": NARX model

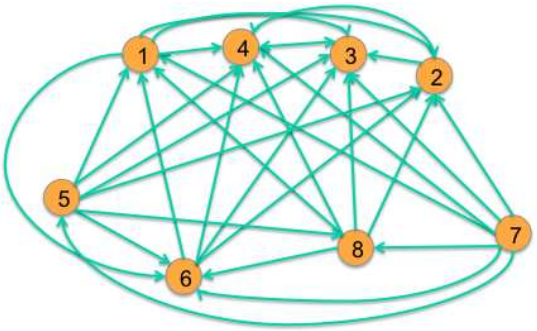

Fig. 7: The dependency graph for the real measurements of Application D3.

models are able to fully identify the system underlying the sensor measurements, as their residual is not white. We provide two possible justifications for this behavior: at first, the sensor acquisition is not strictly synchronized (e.g., we have up to ± 10s among the measurement timestamps) hence impairing the reconstruction ability among different units. Second, the underlying system might be time-variant and, hence, the time-invariant models we are considering are not able to fully capture the global dynamics.

Fig. 7 shows the dependency graph for Application D3: it is worth noting that Unit 7 is the first unit to perceive the effects of the sun during the day. By exploiting the dependencies shown by this graph we generated the subsets as input of a MISO model as described in Section IV. Results are presented in Table VIII and show that, by considering only the effective subset of units as inputs of a MISO model, it is possible to reduce the complexity of the model while maintaining the performance of multiple input solution. Results for Unit 7 are not computed since it is the root of the dependency graph (see Fig. 7) and its effective subset is empty.

VI. Conclusions

The reconstruction of missing data is a challenging and valuable research activity that is preparatory for any fault detection, isolation and identification action and for the following control action. This paper suggests a methodology for reconstructing missing data in distributed monitoring systems encompassing state-space models and input/output linear and nonlinear models (including recurrent, NARX and NOE neural-based models). The main idea behind this methodology is to exploit the temporal and spatial redundancy characterizing measurements coming from different sensing units that are monitoring the same physical phenomenon. The dependency graph, which defines the temporal dependencies among the acquisition units, allows us for reducing the complexity of the estimated model, while guaranteeing accurate reconstruction. Experiments shows the effectiveness of the proposed methodology both on synthetically generated data and measurements coming from a real-world sensor network.

Acknowledgements

This research has been funded by the European Commissions 7th Framework Program, under grant Agreement INSFO-ICT-270428 (iSense). The authors would like to acknowledge Eugenio Migliorini for his precious help in the experimental phase.

References

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

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

[3] C. Alippi and V. Piuri, "Neural modeling of dynamic systems with nonmeasurable state variables," *Instrumentation and Measurement, IEEE Transactions on*, vol. 48, no. 6, pp. 1073–1080, 1999.

[4] 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.

[5] 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.

[6] S. Bennis, F. Berrada, and N. Kang, "Improving single-variable and multivariable techniques for estimating missing hydrological data," *Journal of Hydrology*, vol. 191, no. 1-4, pp. 87–105, 1997.

[7] 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, no. 6, pp. 640–651, 2010.

[8] A. Isaksson, "System identification subject to missing data," in *American Control Conference, 1991*. IEEE, 1991, pp. 693–698.

[9] ——, "Identification of arx-models subject to missing data," *Automatic Control, IEEE Transactions on*, vol. 38, no. 5, pp. 813–819, 1993.

[10] 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, 2001.

[11] B. Hong and C. Chen, "Radial basis function neural network-based non-parametric estimation approach for missing data reconstruction of non-stationary series," in *Proc. of Neural Networks and Signal Processing*, vol. 1. IEEE, 2003, pp. 75–78.

[12] S. Bennis and N. Kang, "Multivariate technique for validating historical hydrometric data with redundant measurements," *Nordic hydrology*, vol. 31, no. 2, pp. 107–126, 2000.

[13] 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.

[14] 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.

[15] L. Ljung, *System identification*. Wiley Online Library, 1999.

[16] C. Alippi, R. Camplani, C. Galperti, A. Marullo, and M. Roveri, "An hybrid wireless-wired monitoring system for real-time rock collapse forecasting," in *Proc. of Mobile Adhoc and Sensor Systems (MASS),*. IEEE, 2010, pp. 224–231.

	Sensor 1	Sensor 2	Sensor 3	Sensor 4	Sensor 5	Sensor 6	Sensor 7	Sensor 8
Sensor 1	-	ARX	ARMAX	ARX	NARX	ARMAX	ARMAX	NARX
Sensor 2	NARX	-	ARMAX	NARX	NARX	ARX	NARX	NARX
Sensor 3	NARX	NARX	-	NARX	NARX	ARMAX	NARX	NARX
Sensor 4	ARMAX	ARMAX	ARMAX	-	NARX	ARMAX	NARX	ARMAX
Sensor 5	ARX	ARX	ARMAX	NARX	-	ARMAX	ARX	ARMAX
Sensor 6	ARMAX	ARX	ARX	ARX	NARX	-	ARMAX	ARMAX
Sensor 7	ARX	NARX	ARMAX	ARMAX	ARMAX	ARMAX	-	NARX
Sensor 8	NARX	ARMAX	ARMAX	NARX	NARX	ARMAX	NARX	-

(a) Best linear models.

	Sensor 1	Sensor 2	Sensor 3	Sensor 4	Sensor 5	Sensor 6	Sensor 7	Sensor 8
Sensor 1	-	96.48	96.46	96.43	97.03	96.42	97.89	96.99
Sensor 2	96.77	-	96.71	96.76	96.94	96.65	97.20	97.23
Sensor 3	96.92	96.71	-	96.64	96.82	96.59	96.94	96.92
Sensor 4	96.45	96.51	96.43	-	97.16	96.63	96.81	96.78
Sensor 5	96.52	96.50	96.5	96.52	-	96.59	97.19	97.02
Sensor 6	95.87	95.88	95.82	95.81	95.97	-	96.15	96.13
Sensor 7	96.51	96.55	96.51	96.52	97.11	96.78	-	97.16
Sensor 8	96.35	96.37	96.37	96.36	97.11	96.64	97.00	-

(b) 1-step ahead prediction fit

	Sensor 1	Sensor 2	Sensor 3	Sensor 4	Sensor 5	Sensor 6	Sensor 7	Sensor 8
Sensor 1	-	1.980	2.119	2.106	3.393	1.903	1.628	4.031
Sensor 2	6.163	-	2.948	3.017	1.819	2.775	3.426	2.470
Sensor 3	1.665	3.779	-	5.046	2.325	1.513	2.282	2.189
Sensor 4	2.542	2.733	2.300	-	3.160	2.005	3.236	2.048
Sensor 5	1.661	1.638	1.732	1.891	-	1.737	1.736	1.634
Sensor 6	2.126	1.808	2.223	2.277	2.075	-	2.055	2.431
Sensor 7	2.020	3.296	2.030	2.031	2.988	3.259	-	2.300
Sensor 8	2.141	2.228	2.064	2.067	2.265	2.067	1.798	-

(c) 1-step ahead whiteness index

TABLE VI: Application D3 - Solution “A”: linear and nonlinear models

	Sensor 1	Sensor 2	Sensor 3	Sensor 4	Sensor 5	Sensor 6	Sensor 7	Sensor 8
N- 1	ARX	ARX	ARX	ARX	ARX	ARMAX	ARMAX	ARMAX

(a) Best linear models.

	Sensor 1	Sensor 2	Sensor 3	Sensor 4	Sensor 5	Sensor 6	Sensor 7	Sensor 8
N- 1	90.83	97.25	97.00	97.09	97.44	96.2	97.34	97.09

(b) 1-step ahead prediction fit

	Sensor 1	Sensor 2	Sensor 3	Sensor 4	Sensor 5	Sensor 6	Sensor 7	Sensor 8
N- 1	1.238	3.935	2.225	3.224	1.453	1.918	3.406	2.735

(c) 1-step ahead whiteness index

TABLE VII: Application D3 - Solution “C”.

	Sensor 1	Sensor 2	Sensor 3	Sensor 4	Sensor 5	Sensor 6	Sensor 7	Sensor 8
Eff. subset	90.81	97.23	97.19	97.11	97.18	96.15	-	96.94

(a) 1-step ahead prediction fit

	Sensor 1	Sensor 2	Sensor 3	Sensor 4	Sensor 5	Sensor 6	Sensor 7	Sensor 8
Eff. subset	2.156	2.615	2.617	3.425	3.706	1.995	-	1.956

(b) 1-step ahead whiteness index

TABLE VIII: Application D3 - MISO solution with effective subsets.