water

MDPI

Article

Different Approaches to SCADA Data Completion in Water Networks

Pere Marti-Puig [1,*,†], Arnau Martí-Sarri [1,2,†] and Moisès Serra-Serra [3]

1 Data and Signal Processing Group, U Science Tech, University of Vic—Central University of Catalonia, c/de la Laura 13, 08500 Vic, Catalonia, Spain; arnau.marti@uvic.cat
2 Aigues de Vic S.A., c/Santiago Ramon y Cajal 60, 08500 Vic, Catalonia, Spain
3 MECAMAT Group, U Science Tech, University of Vic—Central University of Catalonia, c/de la Laura 13, 08500 Vic, Catalonia, Spain; moises.serra@uvic.cat
* Correspondence: pere.marti@uvic.cat; Tel.: +34-93-881-55-19
† These authors contributed equally to this work.

Received: 19 March 2019; Accepted: 9 May 2019; Published: 16 May 2019

Abstract: This work contributes to the techniques used for SCADA (Supervisory Control and Data Acquisition) system data completion in databases containing historical water sensor signals from a water supplier company. Our approach addresses the data restoration problem in two stages. In the first stage, we treat one-dimensional signals by estimating missing data through the combination of two linear predictor filters, one working forwards and one backwards. In the second stage, the data are tensorized to take advantage of the underlying structures at five minute, one day, and one week intervals. Subsequently, a low-range approximation of the tensor is constructed to correct the first stage of the data restoration. This technique requires an offset compensation to guarantee the continuity of the signal at the two ends of the burst. To check the effectiveness of the proposed method, we performed statistical tests by deleting bursts of known sizes in a complete tensor and contrasting different strategies in terms of their performance. For the type of data used, the results show that the proposed data completion approach outperforms other methods, the difference becoming more evident as the size of the bursts of missing data grows.

Keywords: water networks; SCADA data; tensor completion; tensor decomposition

1. Introduction

In a real distribution network of drinking water, the SCADA (Supervisory Control and Data Acquisition) system must periodically acquire, store, and validate the data collected by sensors to achieve accurate monitoring and control of the system. However, before these data can be used to improve the management of the system and the early detection of anomalies, their integrity must be verified. Otherwise, the models appear distorted and errors propagate from the very beginning. Therefore, the raw data coming from each sensor measurement, usually represented by a one-dimensional time series, must be validated before further use, as it is the only way to ensure the reliability of the results obtained afterwards. One of the most significant problems in data management is the losses that occur when either sensors or communication links fail. In both cases, this results in the loss of a burst of samples. The problem of managing lost data is ubiquitous in many situations and is especially challenging when it manifests itself in long bursts. Dealing with incomplete data is very common in real-life cases, and it is usual to find such cases in water [1] and hydrological data management [2–4]. The current work used data from the Drinking Water Treatment Station (DWTS) of Aigües de Vic S.A. For this purpose, data from several DWTS sensors that are stored by the SCADA system in a database were analyzed. The main problem with the usage of

these data for model generation is the amount of lost data that appears in the form of bursts. All of these bursts of missing data have a maximum length of <24 h that depends on the company's fault repair protocols. In Aigues de Vic S.A., there are five technicians available during working hours (4:00–00:00), with at least one of them present in the DWTS at any moment. In the water network, there are six operators working from 8:00 to 18:00 with a 2 h break for lunch. To cover the supervision of the DWTS and the water network for 24 h every day, a technician and an operator take turns on duty, even during traditional non-working hours, including the weekends. As indicated by certain research [5], it is possible to verify the integrity of the data and validate the samples obtained by using predictive techniques. Although traditional data completion approaches [5–8] work well when the proportion of missing data is low, their performance is affected when the ratio increases, mainly because these methods rarely take advantage of the hidden structure of the data [9,10]. The capture of the underlying structure of the data can instead be more easily reached by using tensor factorization techniques when data sets have more than two axes of variation [11–14]. When the problem of missing data is considered under the framework of tensors, two main common strategies are contemplated. One is known as *maximization of expectations* and involves the assignment of estimated values to fill in the missing data. The other, marginalization, ignores the missing values (marginalized) during the optimization process [11]. One of the most well known marginalization methods is the CP-Wopt (CP Weighted OPTimization), which was first introduced in [9]. This algorithm was derived from the CANDECOMP/PARAFAC (CP) tensor factorization for a case with missing data, in which it was reformulated as a weighted least-squares problem that only takes the known entries into account. In comparison with the existing tensor completion methods, our approach is based on an ad-hoc strategy that has proven to be effective when errors occur in bursts. It is based on two main steps: In the first one, a low dimensional method, such as a linear interpolation or a Wiener predictor, is employed to fill in the missing values, while, in the second step, the data are *tensorized* according to their naturally observed periodicity so that the structure of the data can be exploited beyond a single dimension, and therefore, their common features can be captured in five minute, daily, and weekly intervals. In terms of validation of the results, when vectorizing the final output, it has been observed that reconstructions of tensors from low-range models collect the daily oscillations observed in the data better than low-dimensional techniques; this phenomenon becomes increasingly evident as the burst gets longer. However, when assigning the data burst retrieved to the lost positions in order to fill the gap, discontinuities within the original data are observed at the extremes. The ad-hoc observation is such that an offset correction in the recovered data highly improves the reconstruction. This correction is based on the way that the forwards and backwards Wiener predictors are combined in the first stage of the method.

Experiments where bursts of a given size are randomly erased have shown a performance improvement with respect to the traditional methods, including the CP-Wopt [9]. The current study proposes a new algorithm that uses a fixed low-order tensor decomposition which is computationally efficient.

The algorithm encompasses three strategies in the first part of the algorithm. The first two, which are briefly explained in Section 2, are very basic, but were considered as a result of their use by Aigües de Vic S.A. The third consists uses prediction techniques, such as FIR (Finite Impulse Response) filters from Wiener [15]. The design of the predictors has two key elements: the order of the filter and the method used to estimate the auto-correlation of the signal, which is necessary for the calculation of the filter coefficients [15].

On the other hand, tensor techniques provide an effective representation of the structural properties of multidimensional data. Some of the most powerful tools of the tensor algebra are the decompositions that allow the discovery of the interrelations between dimensions. There are various decompositions, but the CANDECOMP/PARAFAC (CP) [16,17] and the Tucker [18–20] are the most well known and widely used and therefore, were considered in the present study. There is an abundance of literature on tensors. Most of the tutorials available introduce the topic from

the perspective of their applications in the fields of machine learning and/or signal processing. Some quality references that are very useful for a rigorous first contact with the subject are [12–14,21]. One of the many uses of tensor algebra is in the field of missing data recovery or tensor completion, in which it has been successfully applied. For instance, in [22], it was used to recover missing values in the visual data; in [23,24], the proposed method was mainly applied to the recovery of traffic data values; and in [25], a low-n-rank tensor recovery method was introduced. Furthermore, in [26], a completion tensor method, which automatically determines the rank of the decomposition, was put forward. However, the tensor completion strategy proposed in [9] is one of the most developed methods, and its corresponding algorithm can be used from open access libraries [27,28].

In a nutshell, when data loss is spread uniformly or even in short bursts (less than 30–40 samples, for our type of signals), all of the methods perform similarly in practice. Above that sample length, the performance of the commonly used methods begins to decrease. However, in practice, data are generally lost in bursts, which can also be very long. So, our proposal was motivated by the need to improve the performance of currently used data completion methods. In this work, we provide evidence of the improvement achieved in comparison with more straightforward and faster methods at the expense of increasing the complexity of the algorithms slightly.

After the introduction, this work continues with Section 2, where the details for reproducing the results are explained. Within this, aspects related to the data and their pre-processing are presented. The section also covers the three "non-tensor" methods used to impute values and a tensor introduction explaining how data are "tensorized", as well as some of the main tensor properties and the two most well known low-rank decompositions. To compare algorithms, a statistical strategy is put forward which consists of erasing known data (in bursts) inside the tensor in order to take an objective measure after comparing the eliminated real values with the recovered ones. The CP-Wopt tensor completion method is tested in the context of our problem and, based on the obtained results, a solution is proposed involving a final step called the "offset continuity correction". Section 3 presents a set of experiments that were designed to evaluate a range of factors, such as the configuration of the FIR, the optimal size of tensor decompositions for both Tucker and CP models, the optimal size of the tensor, and the evaluation of the algorithms according to the length of the lost burst. The main points are summarized in Section 4, and examples are included to illustrate how our approach works in terms of the representation of the recovered data.

2. Materials and Methods

2.1. Data Acquisition

The data used for this research were provided by Aigües de Vic S.A., who have a SCADA system that manages the information collected by the sensors of the DWTS and the water distribution network. The network of pipes is 428.185 km in length, and there are 2800 mechanical water meters, which need to be read by an operator every three months, as well as 21,000 remote water meters, which are connected to a database and send a daily reading. The SCADA receives 1309 different signals from the DTWS and the network, such as instant flow or accumulated water volume from a flowmeter; the pressure of a manometer; the fullness percentage of a water deposit from a level sensor; the value assigned to a pump frequency converter; pH meter data; and on/off signals of valves, pumps, and water deposit buoys.

Their measurements are stored every five minutes in a Structured Query Language (SQL) database of the SCADA server, which has been in operation since 2014, with the majority of signals saved from October 2015 onwards. The level sensor signal studied in this paper has been collected since 1 October 2015 at 07:50. However, an important loss of data occurred from 8 September 2017 at 11:15 until 27 August 2018 at 22:30. The maximum number of weeks configured in the tensor tests is 31. Because of this, approximately 8 months of useful data were required for this experiment. The tested week was chosen for two reasons. Firstly, it had only a small amount of lost original data, which allowed

verification to occur when we restored the deliberately deleted data. There were three weeks, between 23 January 2017 and 12 February 2017, with only three empty positions in the original tensor, which we restored with linear methods. Secondly, good data were obtained for the 14 weeks prior to a following the chosen week, thus ensuring the reliability of the FIR with the highest order and largest tensor calculations. In the selected weeks, the biggest range of used data was from 17 October 2016 to 1 May 2017.

The server that manages all of the information is centralized in a computer. Occasionally, this computer fails or its communication with the sensors is interrupted, resulting in the loss of data. Aigües de Vic S.A. allows access to the data history of any sensor, and we obtained this information using Matlab®from MathWorks (Natick, MA, USA). We started with the Matlab 2016a version; however, during the study we updated it to the Matlab 2018b version. This program is able to execute SQL scripts, making it easy to communicate with the database directly and thus obtain all the required data. This study focused on the sensor that measures the water deposit level of *Castell d'en Planes*, which supplies water to the city of Vic, the capital of Osona region located in the north-east of Catalonia. The population of Vic is 48,287 people (in 2016) and it has an area of 30.6 Km^2. It is very important that this deposit is never fully emptied, because it would leave the city of Vic without any water.

2.2. Data Pre-Processing

The first step that had to be taken before the sensor measurements of the water level could be obtained was to test the integrity of these measurements and to discard the incoherent data [7,8] that do not follow the laws of physics or the particular rules of the sensor environment (for instance, we know that a level sensor measurement of 0 is incorrect because this deposit never has been empty.) The NaNs (Not a Number) classification was given to positions where no data are available, either because they were not saved or because they had been discarded. The physical and temporal characteristics of every collected sample had to be verified. In the temporal domain, two checks were made. The first one was related to synchronization. The read data had to satisfy the periodicity condition set by the SCADA server, which stores the samples every 5 min. This means that the registered minutes digit of every sample must always be a multiple of 5, and the second digit must be a 0. A decision was made to eliminate samples with an inconsistent time, as hown in Figure 1, because, in most cases, the value of this sample was also inconsistent (in the case of the targeted level sensor, 288 of 212,549 samples read as 0 had an incorrect time stamp). The other type of temporal verification made was to ensure that there were no duplicated data. In this case, if the two duplicated values were the same, one was deleted, and if the values were different, the incoherent one (often being 0) was deleted. In the physical domain, the general characteristics of the sensor needed to be considered along with the particular characteristics of the sensor environment. The sensors have minimum and maximum possible values and, in general, certain physical restrictions between the value of a sample and the next one. In the case of the level sensor, the possible values range from 0 to 100 as they are percentages. However, in the case of the *Castell d'en Planes* deposit, which is the water reserve of the city, we can state that the minimum value is 30%. The sensor can only measure values below 30% when the deposit is being cleaned; however, when this happens, the sensor is turned off. Another distinctive detail of this particular deposit is that because of the pumping configuration of water filling and the city water demand, it cannot be emptied or filled at a higher rate than 1% in 5 min. To sum up our case, the range of valid values for the measurements is from 30% to 100%, and the difference between a sample and the next one must be within ±1%. All the samples that do not accomplish these conditions are discounted, as is shown in Figure 2.

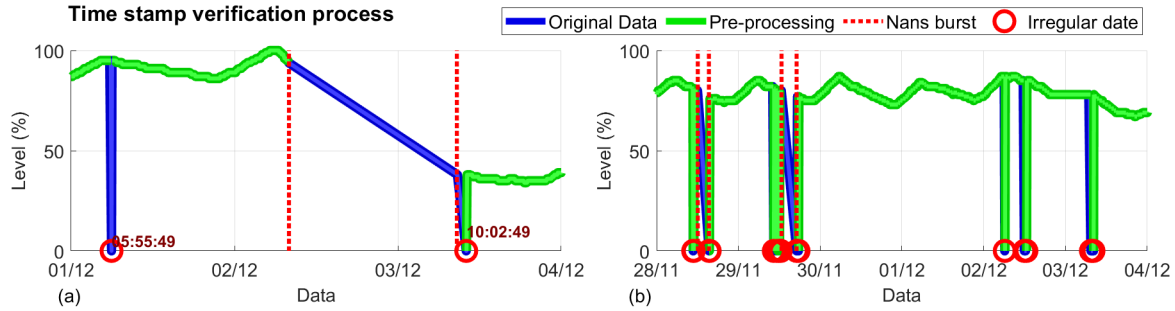

Figure 1. Particular cases of time stamp pre-processing: (**a**) shows a pair of read samples with the irregular date and a burst; (**b**) Shows a pair of bursts and many samples with a value of 0 and irregular date. In both examples, some samples of the pre-processed signal have a value of 0. This is because, at this stage, only the date is pre-processed and there are some samples, read as 0, with a correct time stamp.

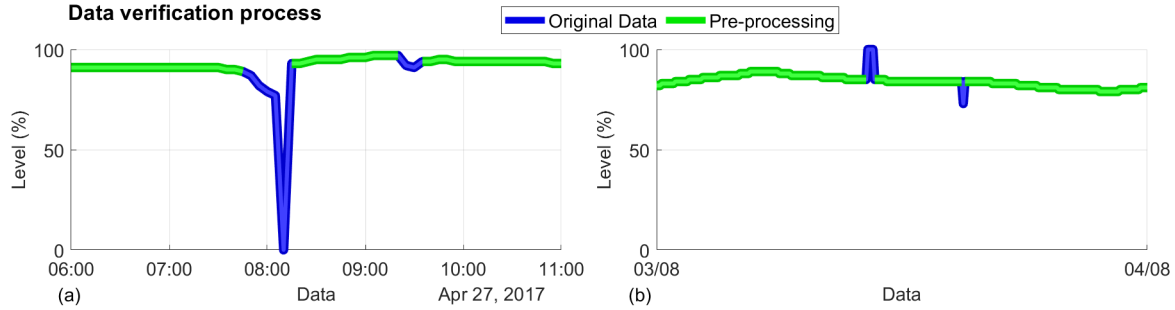

Figure 2. Particular cases of the signal pre-processing step: (**a**) Shows an impossible fall in the level. An operator remembers this case. It was caused by him touching the sensor accidentally. Minutes later he re-positioned the sensor, and approximately 1 h later, a technician re-calibrated the sensor. (**b**) Shows an unusual behavior of the sensor, which spontaneously read incorrect measurements of the level.

2.3. Linear Methods Used for Data Completion

After the pre-processing stage, we were left with a record of samples from the signal level with some gaps due to lost samples (the reasons for this ranged from a SCADA system failure, a communications problem, or a sensor malfunction) and the discarded samples. The most frequent and difficult errors to correct are those caused by bursts, which are often associated with sensor or communication failures. Once data loss is detected, the technical services of the company usually repair the problem and very rarely do the bursts of lost data go on for longer than a day. However, it is common for them to last several hours. Operators have some simple methodologies that they can use to fill the gaps, depending on the data that has been lost and the context in which they were lost. The methods that the operators can use are called the *last sample* method and the *ramp* method. In order to improve the process of filling data, we propose a complementary method, based on the Wiener predictor, which we call the *FIR* method. This third method uses linear prediction techniques, and it is the result of combining forward and backward estimations. When the fault has been repaired, the SCADA system supplies the data again, and because of this, these new data can be used to produce a backwards estimation. We have found that at a certain lost burst size, the latter method usually outperforms the two more simple ones. Figure 3 shows how the above three methods work for two examples. Following this is a brief description of the three methods.

There are different linear techniques for prediction [29] and data imputation [30,31]. Among the most commonly used linear filters, in addition to those of Viener, are techniques using the Kalman filters [32,33], which have applications in air quality [34,35] and seismic [36] data set restoration.

The three methods used are shown below, with emphasis placed the technique based on FIR filtering, which was developed for this particular application.

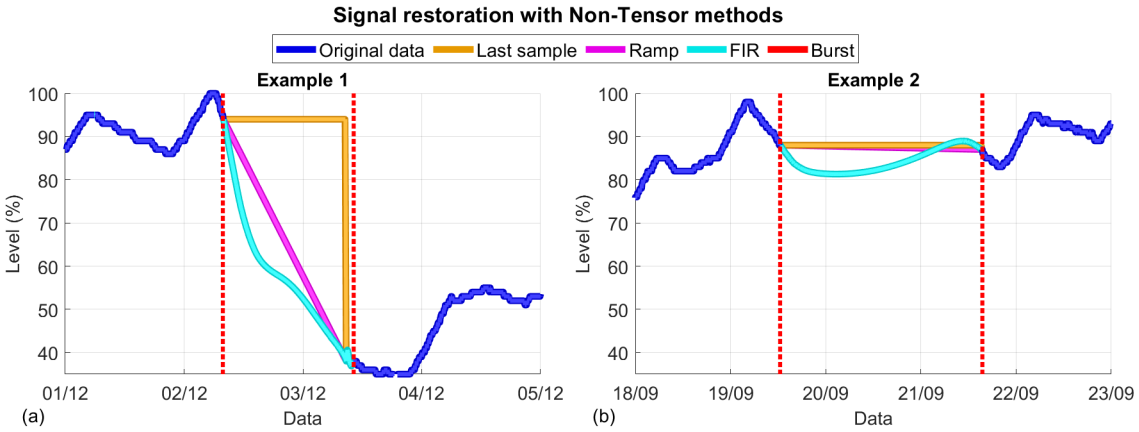

Figure 3. Particular cases of signal restoration using non-tensor methods: (**a**) Shows a middle-length burst where the *last sample* method does not work effectively but the *ramp* and *FIR* (Finite Impulse Response) methods do; (**b**) shows a large-length burst where only the *FIR* method gives a coherent result.

2.3.1. *Last Sample* Method

This is the method used by operators when only the last reliable sample before the burst of lost data is available. Since no extra information is available at the time of restoration, the last reliable value recorded is simply maintained throughout the burst. In other words, all gaps are filled with the same value. Despite its extreme simplicity, this method is adequate if the burst of lost samples is short.

2.3.2. *Ramp* Method

Operators use this other method when they have a reliable pre-burst and post-burst measurement. In this case, a very simple linear approximation is made. The increment/decrement is computed which, when regularly applied to the last known sample, allows the gaps to be filled by linking the last known sample to the next validated sample with a straight line.

2.3.3. *FIR* Method. The Wiener Predictor

A Wiener filter is used to make approximations. It is designed following a statistical criterion in order to obtain the filter coefficients by minimizing the cost function $E[|e_n|^2]$, where e_n is the estimation error, $E[\cdot]$ denotes the expectation operator, and $|\cdot|$ is the Euclidean norm. This error is defined as the difference between sample x_n and its estimation $\hat{x}_n$ which is done by using the FIR filter. According to this, we have $e_n = x_n - \hat{x}_n = x_n - \boldsymbol{a}^T\boldsymbol{x}$ where vector $\boldsymbol{a}^T = [a_0 \dots a_{L-1}]$ contains the filter coefficients and vector $\boldsymbol{x}^T = [x_{n-1} \dots x_{n-L}]$ contains the samples used to make the approximation. We can write the cost function as

$$E[|e_n|^2] = E[e_n e_n^T] = E[(x_n - \boldsymbol{a}^T\boldsymbol{x})(x_n - \boldsymbol{a}^T\boldsymbol{x})^T] = E[x_n x_n^T - x_n\boldsymbol{x}^T\boldsymbol{a} - \boldsymbol{a}^T\boldsymbol{x}^T x_n + \boldsymbol{a}^T\boldsymbol{x}\boldsymbol{x}^T\boldsymbol{a}] \quad (1)$$

which is

$$E[|e_n|^2] = E[x_n x_n] - 2E[x_n\boldsymbol{x}^T]\boldsymbol{a} + \boldsymbol{a}^T E[\boldsymbol{x}\boldsymbol{x}^T]\boldsymbol{a}. \quad (2)$$

The Wiener filter was designed to minimize the mean square error (MSE) criteria. Since Equation (2) is a quadratic function, the solution corresponds with its minimum, which is found after solving

$$\frac{\partial E[|e_n|^2]}{\partial \boldsymbol{a}} = 0, \quad (3)$$

which, when solved, provides the minimum value of the function. As

$$\frac{\partial E[|e_n|^2]}{\partial \boldsymbol{a}} = -E[x_n \boldsymbol{x}^T] + E[\boldsymbol{x}\boldsymbol{x}^T]\boldsymbol{a}, \tag{4}$$

the FIR coefficients will be

$$\boldsymbol{a} = E[\boldsymbol{x}\boldsymbol{x}^T]^{-1} E[x_n \boldsymbol{x}^T] = \boldsymbol{R_{xx}}^{-1} \boldsymbol{r_{xx}} \tag{5}$$

where $\boldsymbol{R_{xx}}$ for real valued magnitudes is a symmetric Toeplitz matrix that takes the form

$$\boldsymbol{R_{xx}} = E[\boldsymbol{x}\boldsymbol{x}^T] = \begin{bmatrix} r_0 & \cdots & r_{L-1} \\ \vdots & \ddots & \vdots \\ r_{L-1} & \cdots & r_0 \end{bmatrix} \tag{6}$$

and $\boldsymbol{r_{xx}}$ written in terms of the auto-correlation coefficients r_i is $\boldsymbol{r}_{xx}^T = [r_1 \dots r_L]$. Concerning the predictor design, there are two key points: the size of the filter, i.e., the value of coefficient L [15], and the method of estimating the auto-correlation coefficients r_i. A sliding time window of size M is used to achieve this aim according to the expression [37]

$$r_k = \frac{\sum_{i=1}^{M}(x_i - \bar{x})(x_{i-k} - \bar{x})}{\sum_{i=1}^{M}(x_i - \bar{x})^2}, \tag{7}$$

where

$$\bar{x} = \frac{1}{M}\sum_{i=1}^{M} x_i. \tag{8}$$

The window and the filter sizes, M and L, are selected after a set of experiments that are described in the next section. To begin to fill the gap, the first estimation is made. The estimated sample $\boldsymbol{a}^T\boldsymbol{x}$ is used to fill the corresponding position in the burst, and it is also used to update the vector $\boldsymbol{x}$ in order to estimate the next one. Then, the process is repeated until the whole burst has been filled. Although the values of the signal that is to be estimated are highly correlated, as we get further from the last verified sample the estimation error grows. Because of this, the last estimated sample does not correspond with the first verified sample following the end of the burst, causing a loss in the continuity of the signal. We call this estimation *forward* as it goes forwards in time. Similarly, estimation *backward* consists of repeating the same technique, but in this case, using the samples obtained after the burst to make the estimation. The estimation begins at the end of the gap and moves towards the beginning, progressing backwards in time. This estimation behaves in an opposite fashion to the previous one: it works very well with estimations at the end of the gap; however, its performance worsens as its approaches the beginning of the gap. Hence, both estimations are not able to maintain signal continuity throughout the whole gap, but both still correctly capture signal oscillations.

2.3.4. *FIR* Method. Forward and Backward Wiener Predictor Combinations

In order to take advantage of both forward $\hat{f}$ and backward $\hat{b}$ estimations, they are combined in the following way:

$$\text{for} \quad i = 1 \dots N; \qquad \hat{x}_i = \begin{cases} \dfrac{\hat{f}_i + \hat{b}_i}{2} & \text{if } N = 1 \\ \dfrac{(N-i)\hat{f}_i + (i-1)\hat{b}_i}{N-1} & \text{if } N > 1 \end{cases} \tag{9}$$

where i indexes the N lost samples from start to end, following a chronological order. Using this expression, greater weight is placed on the best estimations in each of the extremes, and the continuity

of the recovered signal with the valid data is maintained. Figure 4 shows two examples of data reconstruction using this technique. It can be observed that the forward and backward reconstructions follow the signal trend. Although both maintain the continuity of the signal at one end of the lost data burst, two methods data at the other end. By combining these estimates as proposed, continuity is maintained and a more consistent estimation result is obtained.

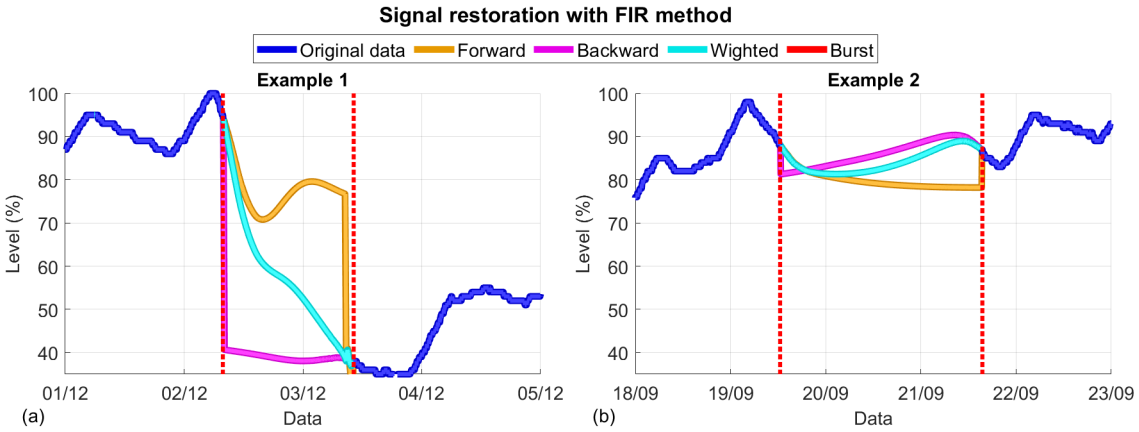

Figure 4. Particular cases of signal restoration with the *FIR* method: (**a**) shows a middle-length burst and (**b**) shows a large-length burst. In both cases, the *forward* and the *backward FIR* methods do not maintain the continuity of the signal but the *Weighted FIR* method does maintain such continuity.

2.4. Tensor-Based Methods for Data Completion

The problem of missing data completion was recently addressed under the tensor framework, and some new strategies appeared. Some of these are derived from well-known decomposition techniques, such as [38,39] for the tensor-Single Value Decomposition (t-SVD), [40] for the Tucker decomposition, and [26,41] for the CP/PARAFAC, while some others strategies originate from less-known methods such as the Riemannian optimization method [42]. As mentioned, tensor methods have been successfully applied in data mining [11] and signal processing [13]. In the presented approach, a second stage/process is added to improve the completion/restoration of the lost samples applied to each of the linear methods, based on the use of tensors, thus forming a new data restoration method that consists of two stages. In the first stage, a linear method is used realize a first estimation of the lost data using some previous and subsequent data around the burst. In the second stage, the data are introduced in a tensor. This tensor is subjected to a simplifying process (decomposition) in a way that allows us to recover the information while avoiding certain inconsistencies generated by the estimation error of the linear methods. This mathematical tool requires that a complete tensor with no missing data is introduced. Because of this, before using the tensors, a previous restoration (imputation) method must be used to fill the gaps caused by lost data.

2.4.1. Definition of a Tensor and Some of Its Basic Properties

A tensor is a container that can arrange data in N-ways or dimensions. The number of dimensions is the order of the tensor. An N-way tensor of real elements is denoted as $\chi \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$, and its elements are $x_{i_1,i_2,\ldots,i_N}$. According to this, an $N \times 1$ vector $\mathbf{x}$ is considered to be a first-order tensor, and an $N \times M$ matrix $\mathbf{X}$ is a second-order tensor. A subtensor is a part of the original tensor which is formed by fixing a subset of indexes. It works in the following way:

- By fixing every index but one, the subtensor is a vector, and it is referred to as a *fiber*. For instance, the fibers $\chi_{:,1,1}$, $\chi_{1,:,1}$, and $\chi_{1,1,:}$ are the first column, row, and *tube* of the three-way tensor χ, respectively.

- A matrix, also called a *slice*, is created by fixing all but two tensor indexes. Following the same example, the matrices $\chi_{:,:,k}$, $\chi_{:,j,:}$, and $\chi_{i,:,:}$ of the three-way tensor χ are frontal, vertical, and horizontal slices, respectively.

The process of reshaping tensors into matrices is known as matricization or matrix unfolding and plays an important role in defining the algebraic operations between tensors and matrices. The following points must be taken into account:

- Commonly, the notation $\boldsymbol{X}_{(n)}$ is used to represent the mode-n matricization of $\chi \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$, an operation which reshapes χ in a matrix $\boldsymbol{X}_{(n)} \in \mathbb{R}^{I_n \times I_1 I_2 \cdots I_{n-1} I_{n+1} \cdots I_N}$.
- Similarly, the reverse operation of mapping a matrix into a tensor is called unmatricization.

The process of reshaping tensors to vectors is named vectorization of a tensor:

- It is denoted $vec(\chi)$ and reshapes $\chi \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ into a vector $\boldsymbol{x} \in \mathbb{R}^{I_1 I_2 \cdots I_N}$.

Tensor algebra has many similarities but also many surprising differences with matrix algebra. It is particularly important to define the product between matrices and tensors in order to define the type of tensor decomposition to be used. The following two notations are used:

- The mode-n product of a tensor $\chi \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_n \times \cdots \times I_N}$ by a matrix $\mathbf{A} \in \mathbb{R}^{J_n \times I_n}$ is denoted as $\zeta = \chi \times_n \mathbf{A}$, with $\zeta \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times J_n \times \cdots \times I_N}$ being the resulting tensor.
- The mode-n product $\zeta = \chi \times_n \mathbf{A}$ also has the following matrix representation $\mathbf{Z}_{(n)} = \mathbf{A}\mathbf{X}_{(n)}$, where $\mathbf{Z}_{(n)}$ and $\mathbf{X}_{(n)}$ are the mode-n matrization of tensors ζ and χ, respectively.

All of these operations and many others are explained in more detail and with graphical examples in [11–13,21]. It is important to note that the reduction of dimensionality and the tensor decomposition techniques allow high-dimensional data to be mapped in a low-dimensional space, conserving the maximum amount of information and making it possible to determine the interactions between dimensions, overcoming the second-order restriction of order two imposed by matrix algebra.

2.4.2. Data Tensorization

The procedure of creating a data tensor from lower-dimensional original data is referred to as *tensorization*. The organization of the data in a container, the tensor, which has more dimensions than the original container allows us to find the relations between dimensions, which are difficult to perceive in more simple structures. In the present case, a three-way tensor of measures is created with the following mode indexes:

- Five minute day intervals: indicates 5 min intervals throughout the day. One day is divided into 288 such intervals.
- Day of the week: indicates the day of the week, from Monday to Sunday, which corresponds to a number from 1 to 7.
- Weeks: indicates n_w, the number of weeks included in the tensor.

The *tensorization* process is shown in Figure 5. Figure 5a shows the evolution of the water levels of a deposit over three weeks. Measures are taken every 5 min and are given as a percentage of the tank capacity as it has regular sections. The staggered effect of the graph shows that the sensors have a limited resolution. The water levels follow specific loading and unloading patterns during the day. There also appears to be some similarity during the same days of the week. For instance, the behavior at the weekends appears to differ from that observed during working days, and because of this, a weekly pattern may well exist. In order to explore and take advantage of possible regularities in the data structure, the data are packed in a tensor χ of size $288 \times 7 \times n_w$. The first dimension corresponds to the number of measures taken during the daytime cycle, the second dimension corresponds to the

days of the week, and the third, n_w, refers to the number of weeks considered. Therefore, Figure 5a shows a representation of the packing of three weeks of data according to the explained criterion to finish building a tensor $\chi^{288\times7\times3}$ of size $288 \times 7 \times 3$. Within that, Figure 5b represents the data corresponding to the first week $\chi(:,:,1)$, while Figure 5c,d portray the data corresponding to the second and third weeks, respectively. According to this notation, $\chi(:,:,1)$, $\chi(:,:,2)$, and $\chi(:,:,3)$ are matrices, and $\chi(:,1,1)$ is a column vector that contains measurements made on Monday of week 1. From the $\chi^{288\times7\times3}$ tensor, the original signal can be retrieved from the operation $vec(\boldsymbol{X}_{(1)})$, $\boldsymbol{X}_{(1)}$, which is the mode-1 matricization of χ.

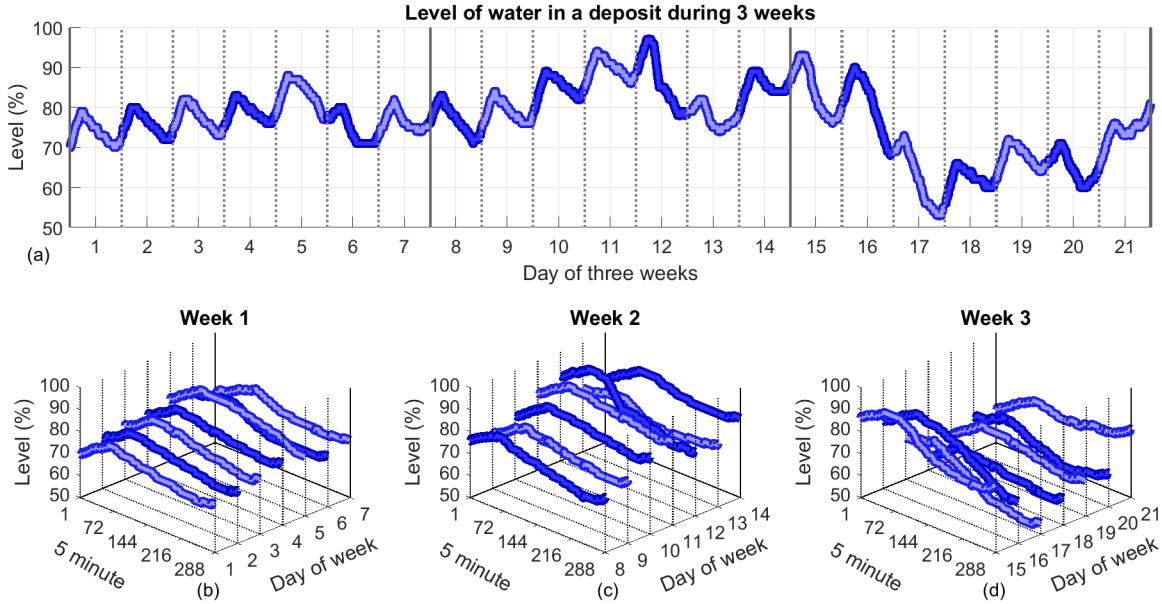

Figure 5. Representation of vector folding **x**, corresponding to the measurements of three weeks in a tensor $\chi^{288\times7\times3}$: (**a**) represents the signal **x** where the fine vertical lines denote the daily separation and the bold vertical lines show the weekly separation; (**b**) represents the information given in $\chi(:,:,1)$ of week 1, in which each vector $\chi(:,i,1)$, $i \in [1,7]$ provides the daily measures of day i; (**c**) shows the same representation for $\chi(:,:,2)$ of week 2 and (**d**) for $\chi(:,:,3)$ of week 3.

2.4.3. The Tucker and CANDECOMP/PARAFAC Models

When classic matrix factorization approaches are used, part of the multidimensional structure of the data is lost due to the collapse of some of the tensor's modes for framing matrices. However, tensor decomposition techniques allow the explicit exploitation of the multidimensional data structure. Among many existing tensor decomposition techniques, there are two main tensor approaches, which are the ones considered in this work: the Tucker decomposition and the canonical decomposition (CP) with parallel factors, also known as (CANDECOMP/PARAFAC) [11–14,21]. In fact, the CP is a particular case of Tucker decomposition. For simplicity reasons, and following the particular case considered in this work, these decompositions are shown for an order of 3. The Tucker model proposed in [18] decomposes a third-order tensor $\chi^{I\times J\times K}$ as a multilinear transformation of a typically smaller core tensor $G^{L\times M\times N}$ by the factor matrices $\boldsymbol{A}^{I\times L}$, $\boldsymbol{B}^{J\times M}$ and $\boldsymbol{C}^{K\times N}$, accounting for the linear interactions between each of the mode's components. It is common to use the notation Tucker(L,M,N) to indicate the number of vectors that belong to each mode. Using the tensor–matrix product, this decomposition can be written as

$$\chi^{I\times J\times K} \approx G^{L\times M\times N} \times_1 \boldsymbol{A}^{I\times L} \times_2 \boldsymbol{B}^{J\times M} \times_3 \boldsymbol{C}^{K\times N}. \tag{10}$$

The CP decomposition is a particular case of the Tucker decomposition where the number of vectors of each core dimension is the same, that is $L = M = N$ (=D). The interactions in CP are only between columns of the same indices, implying that the core tensor $D^{D\times D\times D}$ is diagonal and, therefore, the only non-zero elements are in the main diagonal (i.e., $d_{l,m,n} \neq 0$, if and only if $l = m = n$). The CP decomposition was independently proposed in [16,19,20]. The CP model in terms of the tensor–matrix product can be written as

$$\chi^{I\times J\times K} \approx D^{D\times D\times D} \times_1 \boldsymbol{A}^{I\times D} \times_2 \boldsymbol{B}^{J\times D} \times_3 \boldsymbol{C}^{K\times D}. \tag{11}$$

The same CP model can be given in terms of the outer product of the three vectors $\mathbf{a}_i$, $\mathbf{b}_i$, and $\mathbf{c}_i$ as

$$\chi^{I\times J\times K} \approx \sum_{i=1}^{D} \lambda_i \mathbf{a}_i \circ \mathbf{b}_i \circ \mathbf{c}_i, \tag{12}$$

where the column vectors $\mathbf{a}_i$, $\mathbf{b}_i$ and $\mathbf{c}_i$ are related to the matrices of Equation (11) according to $\boldsymbol{A}^{I\times D} = [\mathbf{a}_1 \cdots \mathbf{a}_D]$, $\boldsymbol{B}^{J\times D} = [\mathbf{b}_1 \cdots \mathbf{b}_D]$, and $\boldsymbol{C}^{K\times D} = [\mathbf{c}_1 \cdots \mathbf{c}_D]$. Figure 6a graphically shows the Tucker(L,M,N) decomposition in terms of the tensors involved and the factor matrices for the three-dimensional case, while Figure 6b shows the CP(D) decomposition, first in terms of the tensor–matrix product and then, using the outer product of the vectors.

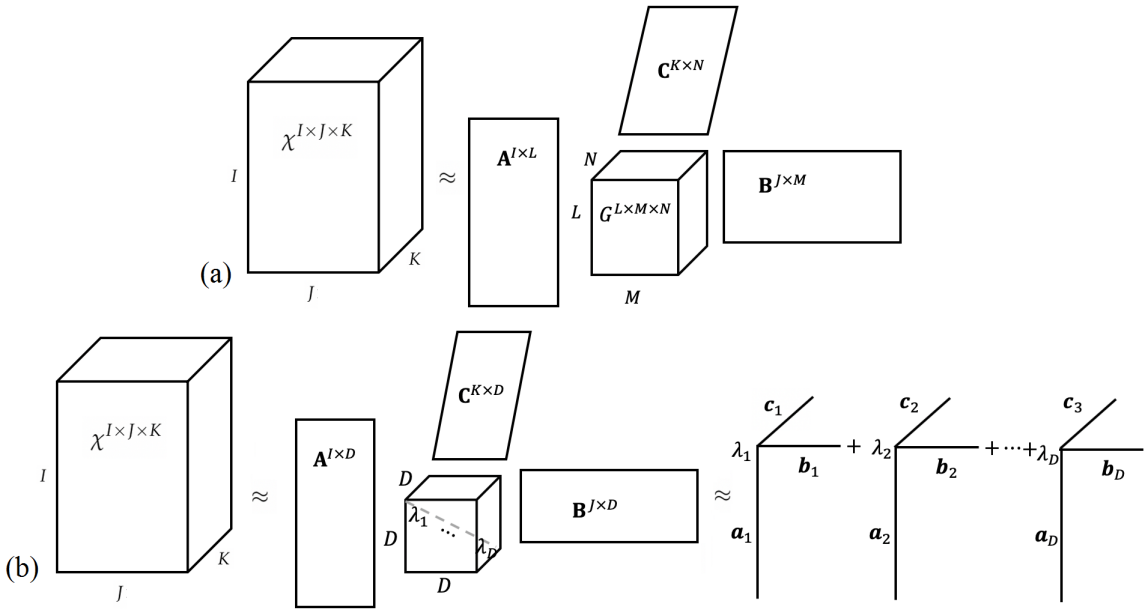

Figure 6. (**a**) Tucker model Tc(L,M,N); (**b**) CANDECOMP/PARAFAC model CP(D) .

2.4.4. CP Weighted Optimization (CP-Wopt)

In [9], a modification of the CP decomposition was used in the presence of missing data by modeling only the known entries and ignoring (marginalizing) the missing ones. The algorithm uses a first-order optimization approach to solve the weighted least-squares problem, and it is known as CP-Wopt (CP Weighted OPTimization). CP-Wopt is one of the most widely used data completion algorithms, and it was proven to be extremely useful in recovering missing data when compared with two-dimensional methods [9]. The CP-Wopt algorithm is included in the Poblano library [27]. Our first option was to use it for our particular data set for these reasons. To test the performance of the method and to be able to quantify it, it was used with known values that had been intentionally eliminated. From the results obtained, we extracted two main insights:

- The algorithm returns a tensor with the missing data filled in but with slightly different values at the continuity points at the ends of the lost burst. When we used the recovered data to fill in the original tensor, this discontinuity usually made the recovery results given by the *ramp* or *FIR* methods worse.
- It was observed that the recovered data followed the variations of the original signal in a very reliable way, although with an offset, since the use of tensors can exploit the inter-relationships between dimensions. This behaviour persisted as the size of the lost bursts increased, surpassing the performance of the FIR predictors.

An example of the way the CP-Wopt algorithm works is shown in Figures 7 and 8 for tensor $\chi^{288\times7\times3}$ with randomly eliminated bursts (shown in red). It can be appreciated that the estimation (in cyan) occasionally differs from the original data (in blue), although it reproduces original fluctuations. Figure 7 shows good results regarding the completion of a small size tensor $\chi^{288\times7\times3}$ with a burst of 100 missing samples using the CP-Wopt algorithm. In Figure 8, it can be observed that sometimes, if the corresponding cyan color section (output of CP-Wopt) is used directly to fill the empty positions, it not only makes a considerable error, but the continuity at the burst extremes is also lost. However, the fluctuation of the signal still remains despite this *offset*.

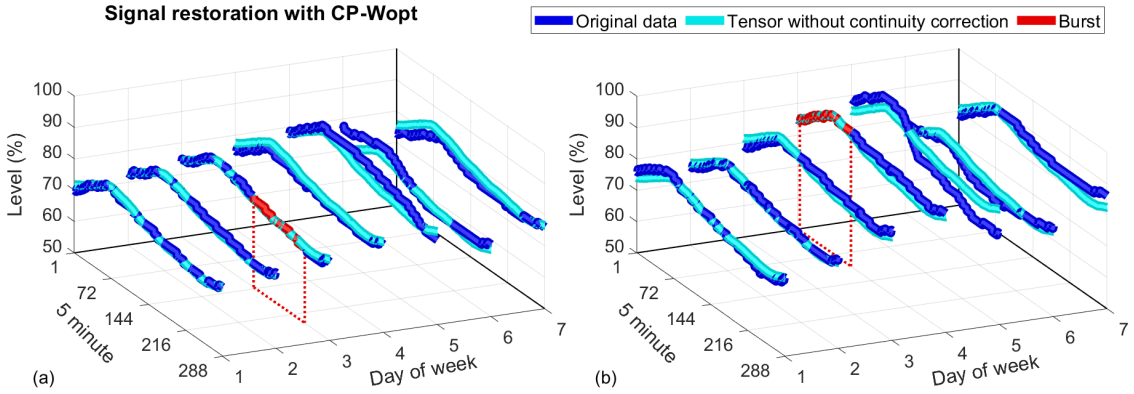

Figure 7. A particular case to show the performance of the CP Weighted OPTimization (CP-Wopt) algorithm when recovering two missing bursts in a $\chi^{288\times7\times3}$ tensor. The original signal is shown in blue, the section corresponding to the erased burst is in red, and the result of the algorithm is displayed in cyan. (**a**) Shows the data in $\chi(:,:,1)$ corresponding to week 1, and (**b**) shows the data in $\chi(:,:,2)$ corresponding to week 2. It can be observed that the output of the CP-Wopt algorithm follows the variations (the first derivative) of the original data, although with some points differing.

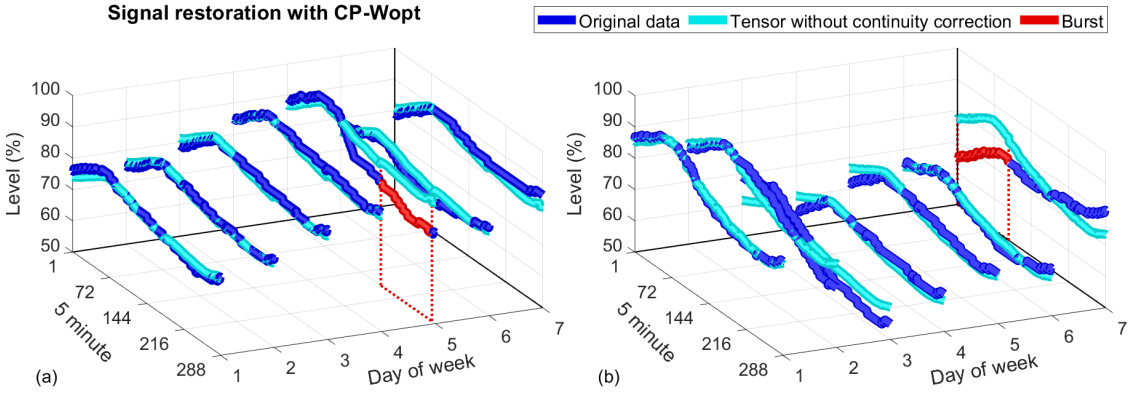

Figure 8. Representation of the offset error that the CP-Wopt algorithm sometimes performs when recovering two missing bursts in a $\chi^{288\times7\times3}$ tensor. The original signal is shown in blue, the section corresponding to the erased burst of 100 samples is shown in red, and the result of the algorithm is shown in cyan: (**a**) shows the data in $\chi(:,:,2)$ corresponding to week 2; (**b**) shows the data in $\chi(:,:,3)$ corresponding to week 3.

2.4.5. The Proposed Tensor Method with an Offset Correction

The algorithm proposed to fill the lost data takes advantage of the inter-relationships between dimensions that capture the algorithms based on tensors. Thus, the signal is tensorized, as explained in Section 2.4.2, in such a way as to take advantage of the patterns between the days of the week and between the weeks. To do this, the following steps are established:

- One of the linear methods for data completion described in Section 2.3 is used to fill the empty burst. The positions of the lost data are stored. After this step, there are no empty elements in the tensor.
- A low-range tensor approximation is performed using tensor decomposition. In our case, the popular Tucker and CP algorithms were explored, as explained in Section 2.4.3. This simple approach captures the most important variations of the original tensor, reproducing it with a high level of accuracy. The difference between the low-rank approximation and the original tensor consists of high-frequency components with low amplitudes.
- Samples $\hat{x}_i$ found in the original lost data positions are retrieved from the low-range tensor and corrected as explained below to maintain continuity when the burst ends. The correction procedure carried out is similar to the one applied to the *FIR* method. The resulting compensated data are the ones that are subsequently assigned. This last step is referred to as an offset correction.

To explain the data correction process used with the aim of preserving the continuity between the extremes, a burst of lost data of N samples is first considered. We let x_a be the last reliable received sample before the loss occurs and x_b be the first sample correctly received once the burst is over. Let $\hat{x}_i$, $i = 1, \cdots, N$ be the estimates made, regardless of the method used, corresponding to the positions where the data were missed. Let $\hat{x}_0$ and $\hat{x}_{N+1}$ be the estimates of the positions corresponding to x_a and x_b. The method that produces the offset is $O_a = x_a - \hat{x}_0$, at the beginning, and $O_b = x_b - \hat{x}_{N+1}$, at the end. Thus, to maintain continuity, the correction of the offset is performed according to the expression

$$\text{for} \quad i = 1 \ldots N; \qquad \tilde{x}_i = \begin{cases} \hat{x}_i - \dfrac{O_a + O_b}{2} & \text{if } N = 1 \\ \hat{x}_i - \dfrac{(N-i)O_a + (i-1)O_b}{N-1} & \text{if } N > 1 \end{cases} \tag{13}$$

where $\tilde{x}_i$ are the estimations with the corrected offset.

2.5. Algorithm Performance Evaluation

To be able to evaluate the algorithms and compare them, a loss-less stretch of data was selected. This data set was used as a reference, and bursts of L values were intentionally removed randomly. This way, when an algorithm fills the burst, it was possible to compare the results with the original values and thus establish an objective measurement. For each burst, the mean squared error (MSE) per sample was chosen according to the following expression

$$MSE = \frac{1}{L} \sum_{i=1}^{L} \sqrt{(x_i - \hat{x}_i)^2}. \tag{14}$$

One thousand different positions corresponding to the starts of the burst were randomly calculated to test the algorithms under the same conditions. These positions and L were saved and then used to generate the bursts in all tests. Therefore, each algorithm reconstructed the same 1000 bursts to compute the final score (the MSE per sample).

3. Results

3.1. Tuning the FIR Filters

Among the three methods used that do not involve tensors, the only one that is configurable is the method based on the FIR filters. For the first experiment, a test was performed to determine the right combinations of L and M. The experiment involved randomly erasing data bursts and recovering them with the *FIR* method while a sweep of both parameters was performed. The results were evaluated by calculating the mean squared error (MSE) made per sample. The bursts removed had a length of 100 samples. The results, which can be seen in Figure 9, are the average of 1000 experiments. The combination of parameters selected as the best option was ($L = 65$, $M = 2000$), which achieved the use of minimum values for M and L, and the MSE was, at most, 1% bigger than the minimum error. Three more combinations from both parameters were tested using similar criteria. The rule of choosing the minimum M and L values was followed. In addition, a restriction was imposed that stated that the MSE was, at most, 5%, 10%, or 15% bigger than the minimum error, respectively.

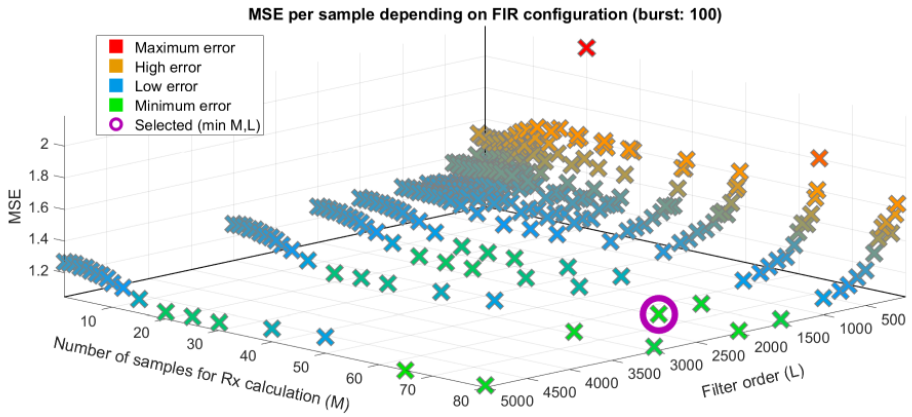

Figure 9. Mean square error (MSE) per sample of the *FIR* method depending on the FIR order (L) and the number of samples used in the auto-correlation calculation (M).

3.2. Tensor Methods—Best Factorization Size

This aim of this experiment was to determine the effects of the decomposition size on the proposed tensor-based methods. Thus, a set of data was *tensorized* in a tensor $\chi^{288\times7\times3}$ of size $288 \times 7 \times 3$, and the performance of the different algorithms were evaluated by randomly generating a burst of 1000 lost data points according to the system described in Section 2.5. Since two popular decompositions were explored, the experiments are repeated for each type of decomposition.

3.2.1. Algorithms employing the Tucker factorization

The algorithms used to perform the initial imputation of values were evaluated by means of the *last value*, the *ramp*, and certain different configurations of the *FIR* methods, followed by tensorization and corresponding Tucker decomposition of different sizes. For each case, the effect of applying or not applying the offset correction was considered. For reference purposes, the results were compared with (1) the value imputation method used directly without performing the tensorization step and (2) with the original CP-Wopt method and its modified version with the offset correction. The results are summarized using bar graphs in Figure 10. In this figure, the bar diagrams are segregated into subfigures according to the method of imputation used. In the axis of abscissas, the decomposition used is shown in parenthesis. When the imputation method used was the *FIR*, the method is referenced in the graphs as FIR (L,M), with the first number being the length of the filter and the second being the number of values used to calculate the auto-correlation coefficients. For each case, the bar of the decomposition with the minimum MSE per sample is marked in red (as well as certain other bars with

similarly low values of MSE). In addition, a red horizontal line marks the specific value that is taken as a reference for comparison with the others. In order to achieve statistically representative results, the MSE values are the result of averaging 1000 experiments. Note that all tested methods performed their best with the Tucker(3,3,1) decomposition.

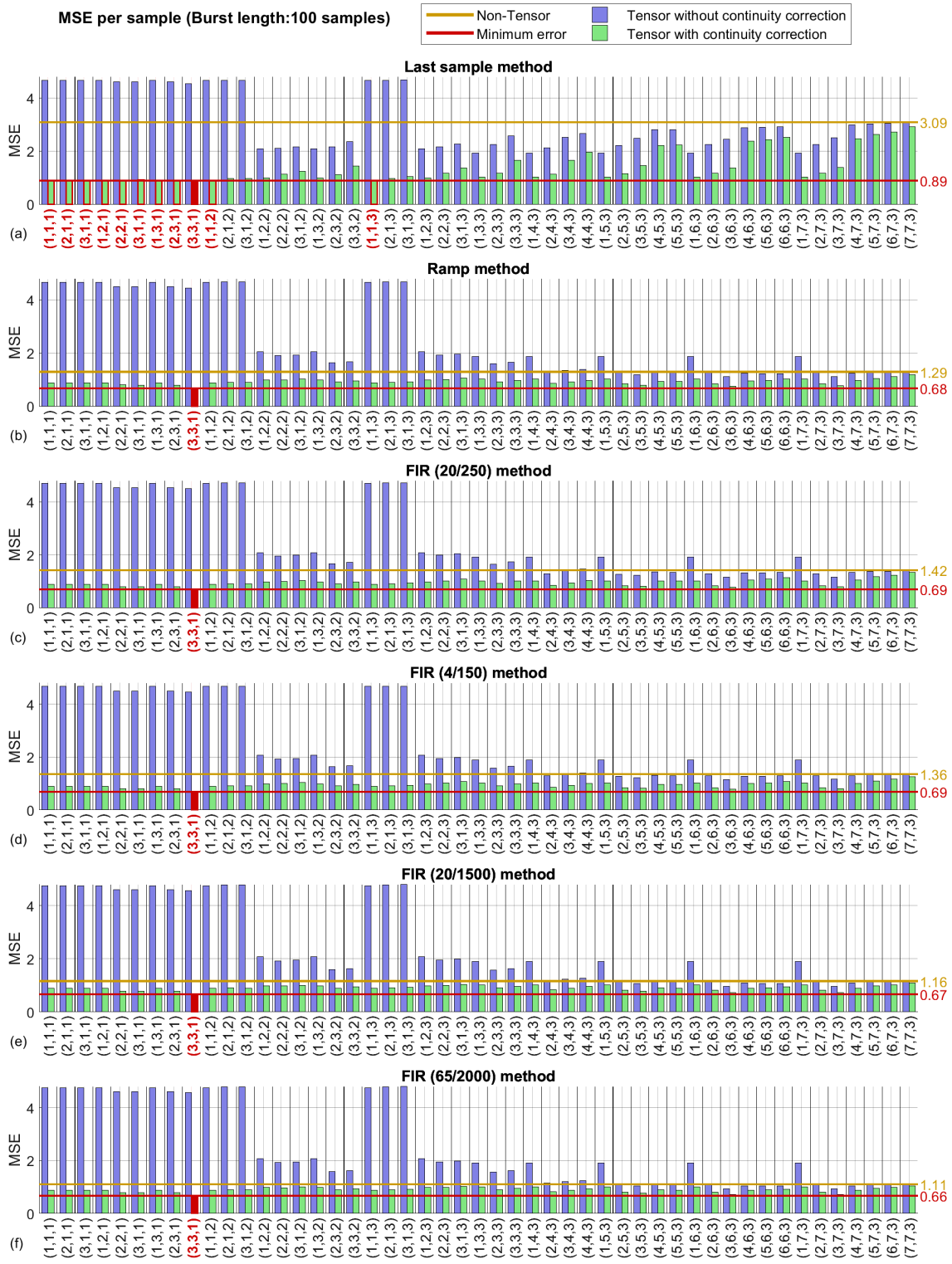

Figure 10. MSE per sample of the Tucker decomposition depending on the core tensor. The MSE of the restoration is shown with the continuity correction and without it. The bars marked in red have an error no bigger than 5% of the minimum.

3.2.2. Algorithms Employing CANDECOMP/PARAFAC or CP Factorization

A similar test was performed for the CP decomposition. In this case, only D was explored. The results are also summarized using bar graphs in Figure 11. The order of the decompositions ranged from 2 to 15. In Figure 11, for each experiment, the decomposition that provided the lowest MSE per sample again has the corresponding bar marked in red. Additionally, a horizontal line is drawn with the level of this bar to allow for comparison with the performance of the data imputation method (i.e., non-tensor method) before decomposition, which is marked with an orange horizontal line. The difference between these two lines illustrates the improvement of the method in terms of the MSE per sample. It can also be observed that the size of the decomposition that provides the best results is around $D = 6$, except for the *last value* method, which gives the best results when $D = 2$ and gets worse as the size of the decomposition increases.

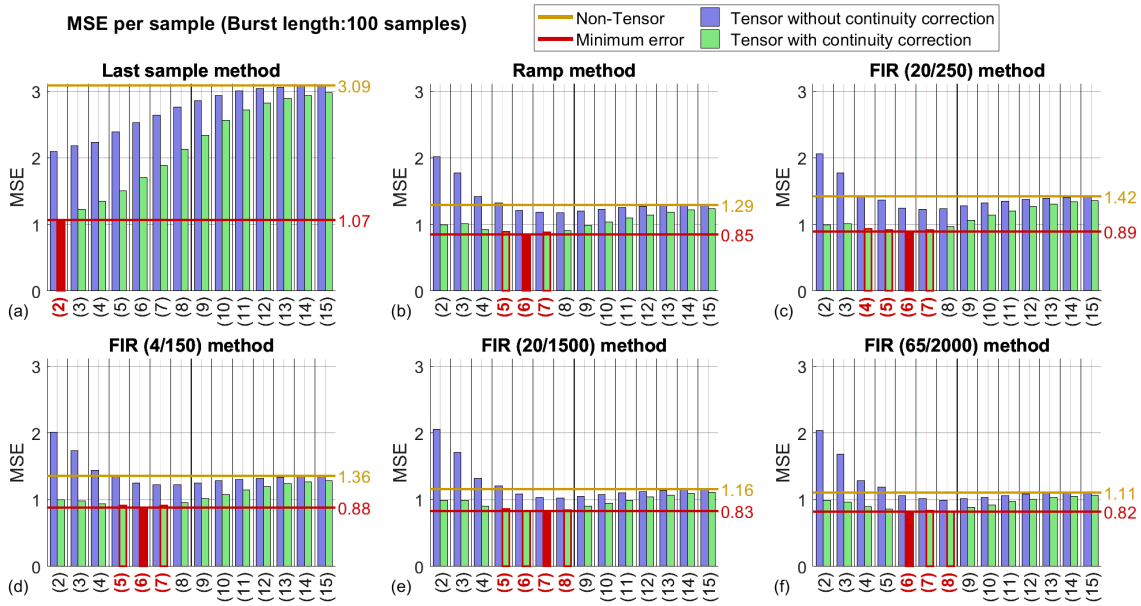

Figure 11. MSE per sample of the CP decomposition depending on the core tensor. The MSE of the restoration is shown with the continuity correction and without it. The bars marked in red have an error of no bigger than 5% of the minimum.

Figure 12 displays the results of different imputation methods using a selection of CP and Tucker decompositions that function more effectively, in particular, Tucker(2,2,1), Tucker(3,3,1), CP(2), and CP(6). In addition to these, the Tucker(1,1,1) decomposition, which is equivalent to the CP(1) (as can be intuitively observed in Figure 6) has also been included. Finally, the CP-Wopt is also shown, which works directly with the missing values and does not depend on the first stage of the restoration method.

Thus, we selected Tucker(1,1,1) = CP(1) (which are the same), CP(2), CP(6), Tucker(2,2,1), and Tucker(3,3,1) as references to evaluate the different algorithms. Figure 12 was constructed using the same experimental conditions, tenso,r and lost burst sizes as those used in the experiments summarized in the two previous figures.

Note that each algorithm is presented with and without an offset correction. As explained previously, for each method, the sample that produces the lowest MSE is marked in red, and a horizontal line is plotted at this level to facilitate comparisons. It can be seen that once again, for all the methods shown, the decomposition Tucker(3,3,1) obtains the best results (see Figures 10 and 12), except for the *last sample* value imputation method. Nevertheless, the remaining decompositions also achieve very similar results in a systematic way.

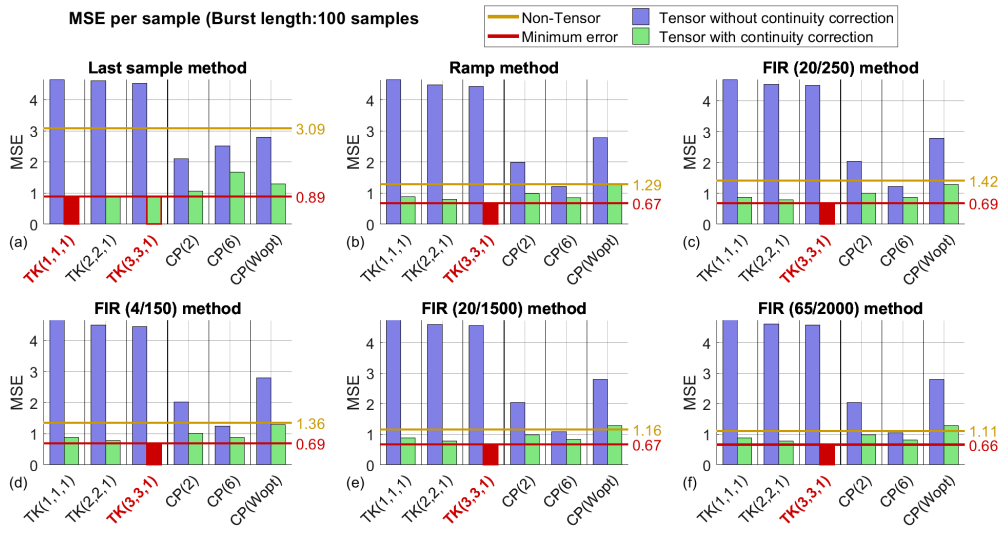

Figure 12. MSE of different value imputation methods. The results are the mean of 1000 simulations. The number of samples in the burst is 100. The position of the the burst is randomly selected. The tensor contains three weeks worth of data (288,7,3).

3.3. The Algorithm's Performance According to the Tensor Size

Once the structure of the tensor has been defined by arranging the data in five-minute, daily and weekly intervals, it is possible to change the tensor size by taking into account either more or fewer weeks. Hence, the effect of the tensor size on the recovery of the lost burst should be investigated. The experiment carried out was as follows. A burst of $N = 100$ was randomly erased from a known tensor that packs data from a different number of weeks. The same starting positions for the bursts obtained in previous experiments for the $\chi^{288\times7\times3}$ tensor were used and the performance test was run. Then, a week of known data was progressively added at the beginning and end of the tensor, and the experiment was reproduced. The configuration $\chi^{288\times7\times2i+1}$ where $i = 1$ was used as the original configuration, and $\chi^{288\times7\times31}$ where $i = 15$ was evaluated.

The results for the selected algorithm configurations are shown in Figure 13.

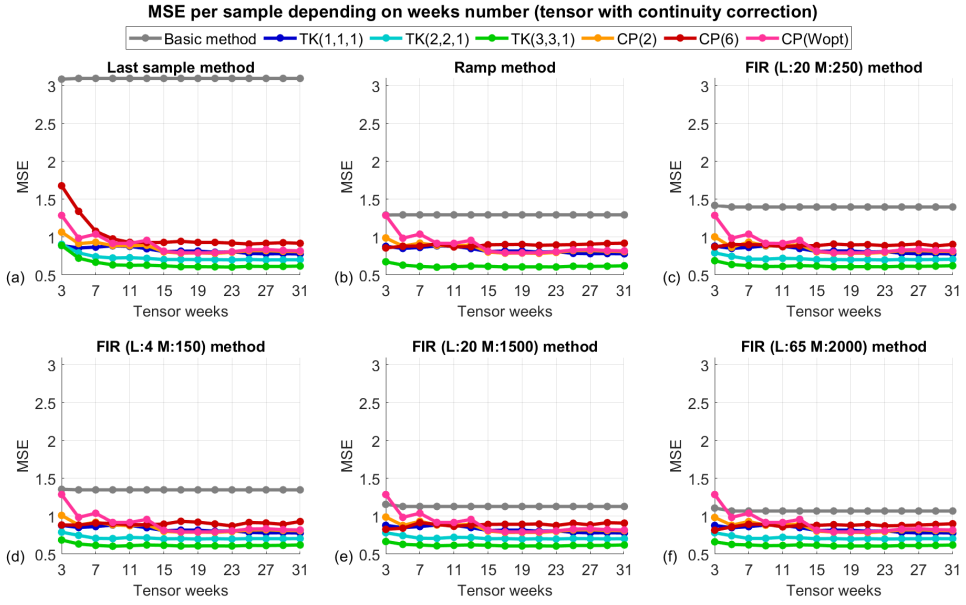

Figure 13. Trend of the MSE with an increaasing number of weeks configured in the tensor. The results are the mean of 1000 simulations. The tensor contains three weeks worth of data (288,7,3). The number of samples in the burst is 100, and the position of the burst was randomly selected.

3.4. The Algorithm's Performance According to the Length of the Missing Data Burst

Figure 14 shows the evolution of the selected algorithms according to the size of the burst, as well as the behavior of the restoration algorithm with the continuity correction using different configurations of factorization and for different "non-tensor" initial methods. Thus, the different imputation methods that alter the burst size (L = 25, 50, 100, 150, 200 and 250) were evaluated. Values of $L > 250$ imply a duration of nearly a day or more, and in this case, are challenging to find because the maintenance department has had time to repair the sensor. For the experiment, the same method of selecting the starting points of the burst as that described in the previous experiments was followed and the data were erased from an $\chi^{288\times7\times3}$-sized tensor. The behavior of the CP-Wopt method with subsequent offset correction is also shown. To facilitate a comparison, the axes of all the sub-graphics are equal.

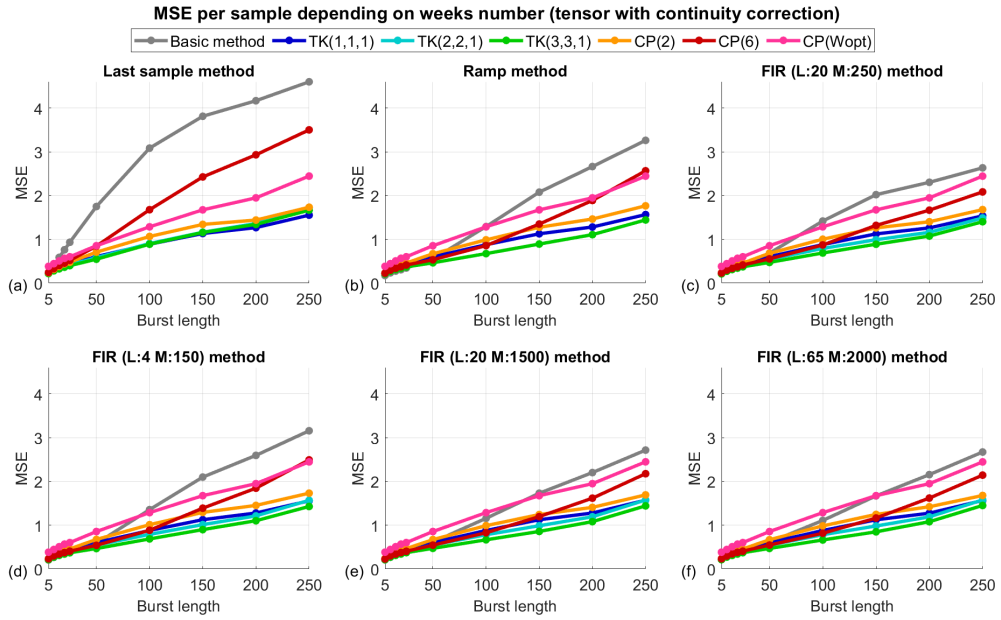

Figure 14. Trend of the MSE with a progressively increasing number of samples. The results are the mean of 1000 simulations. The tensor contains three weeks worth of data (288,7,3), and the position of the first missing sample of the burst is randomly selected.

4. Discussion and Conclusions

Nowadays, SCADA systems store a huge amount of data from a wide variety of industrial processes. The vast amount of stored data can be used to improve plant models for prognosis and diagnosis purposes. However, the first step is to ensure that the available data are reliable, as it is a major challenge to guarantee the consistency of the rest of the steps and procedures. This paper put forward a specifically derived method to recover lost data that occur in bursts, due to failure of either a sensor or the link used to transmit the measurements. In the problem addressed, the signal of interest has soft variations due to the inertia of the process behind it and the simplicity of the sensors which causes the signal to have a staggered appearance, as shown in Figure 5. The data are recorded in five minute intervals, and although the methods based on interpolation and prediction work quite well, tensorization of the signal takes advantage of the specific patterns on daily and weekly scales. This advantage becomes more significant as the size of the missing burst increases, as shown in Figure 14. Among the different strategies for recovering lost data is the allocation of values using more or less elaborate methods, such as those based on interpolation and prediction. Value imputation techniques involve the assignment of values to missing data without using tensor formulation. We compared three such techniques, two of which were very simple but that have been used historically by the water company to solve this problem, and a third, more elaborate one, based on the combination

of the forward and backward linear estimations. FIR filters are used as predictors. Their optimal lengths and the method they employ to estimate the auto-correlation coefficients to calculate the best configuration according to the Wiener formulation were studied and detailed. The use of tensor algebra makes it possible to exploit the relationships between data dimensions. Among tensor strategies, marginalization techniques avoid assumptions being made about the missing data. For certain types of signals and for certain types of missing data distributions, marginalization techniques work remarkably well [9]. For instance, the CP-Wopt is able to recover the correct underlying components from noisy data with up to 99% of missing data for third-order tensors; in contrast, two-way methods become unstable with missing data levels of only 25–40% [9]. However, for some applications, the recovery of missing data can be improved by using a specialized solution, such as in the presented case or the one in [43]. In the present work, in which the missing data were lost in bursts, combining an imputation method followed by a low-range tensor approximation with an ad-hoc offset correction to ensure continuity in the extremes of the missing burst produced a better performance over all the other tested completion methods. We developed a procedure to compare the performance of the algorithms. For the proposed approach, different configurations for the predictor were explored (Figure 9), the optimal size of tensor decomposition was determined (Figures 10 and 11), and the dimensions of the tensor that offer the best performance were obtained (Figure 13). We also compared algorithms by varying the lengths of the missing bursts (Figure 14).

Various conclusions can be drawn from the results. In the general case of very short bursts, when $N \leq 20$, all methods make similar errors. Using a rough, less precise imputation method in the first stage usually works best with very low-range decompositions. This means that for the *last sample* method, the best performance is obtained using the most elementary tensor decompositions TK(1,1,1) and CP(1). However, the *last sample* method degenerates very quickly if the burst is larger than 25 samples. If the imputation method is more sophisticated, a bigger-rank decomposition improves the performance of the algorithm. We found the CP(6) (CANDECOM/PARAFAC) and Tucker(3,3,1) to be good options in all cases. The two decompositions work very similarly, although the latter may be slightly superior. Concerning the optimal size of the tensor, we explored the third dimension, which corresponds to weekly measurement, so that for the same removed burst, the tensor with known data was increased. We first ran the algorithms with three weeks of data and then proceeded to progressively add weeks to evaluate if the recovery improved with additional known data. The results showed an improvement in the performance until approximately seven weeks, when the performance results stabilized. When comparing the different algorithms with lost bursts of different lengths, it is apparent that for burst sizes of up to $N = 100$ samples, the results obtained with the *ramp* method are as good as those obtained using predictors, and it would only be justified to use a more complex method when N is (approximately) greater than 100. Furthermore , in all the experiments where algorithms were compared, we introduced the original CP-Wopt completion method and a modification of it in which the offset is corrected in the same way as that performed for the rest of the proposed algorithms. Finally, we note that in all cases, the simple offset correction step plays a fundamental role in improving the performance. We illustrated that the offset correction works by looking at examples of a couple of particular cases. The different steps of the proposed method are represented in Figures 15 and 16. i.e., to allow a comparison to be made, the results of the algorithm of value imputation, the reconstruction of the signal through a low-range tensor, and the value of the reconstruction after correcting the offset, are visualized together with the data of the original burst that were erased at the beginning of the experiment.

An additional advantage of the method is that the execution time is not excessive. The execution time of the different parts of the algorithm was measured performed on an Intel(R) Core(TM) i5-6200U, 2.3 GHz platform with 8 GB of RAM with Windows 7 Professional and Matlab 2018b. The basic methods of imputation (last sample and ramp) only took 0, 2, and 4 ms. The FIR method required greater computation time, from 2.5 to 10 s, mainly due to auto-correlations and pseudo-inverses. Tensor decomposition took between 0.1 and 0.5 s.

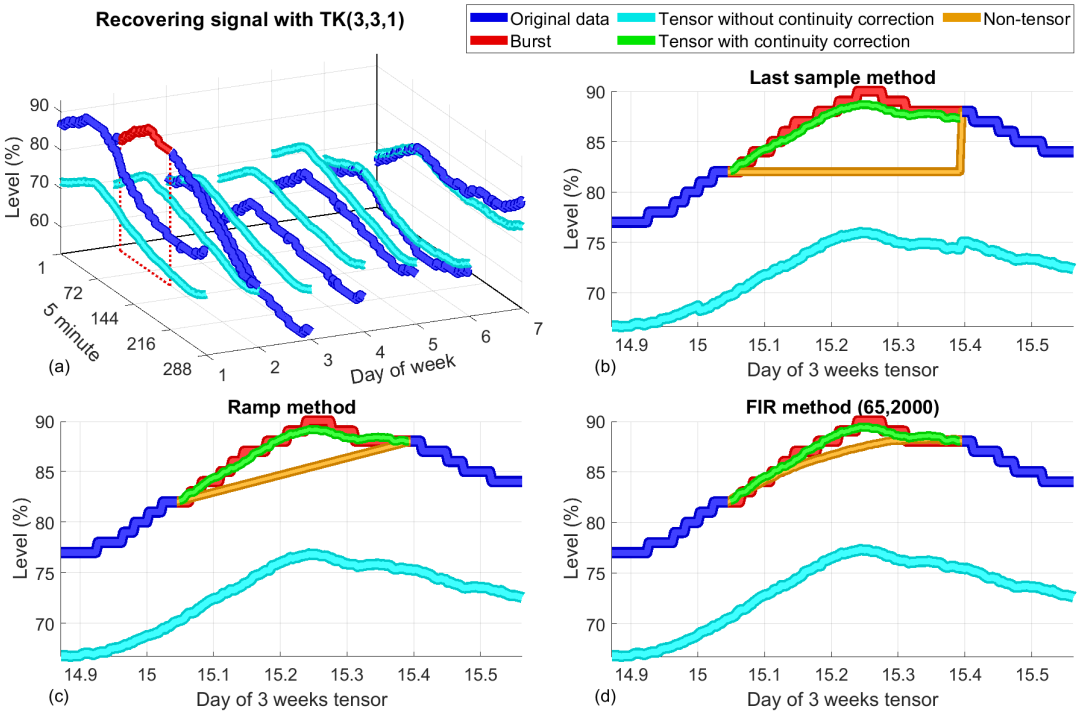

Figure 15. An example of the restoration stages. The tensor contains three weeks worth of data (288,7,3). The burst is 100 samples long, and its position is randomly selected: (**a**) shows the week of the tensor where the burst is located, $\chi(:,:,3)$, corresponding to week 3. Parts (**b**–**d**) show the three steps of the restoration process for the *last sample*, the *ramp*, and the *FIR* methods, respectively.

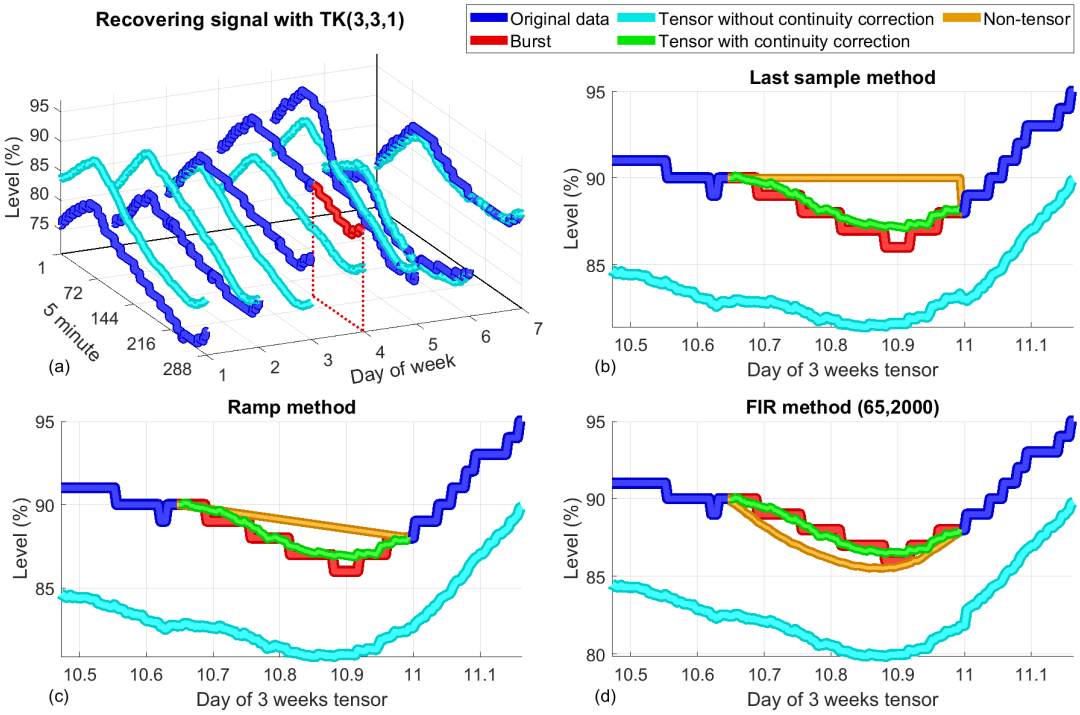

Figure 16. An example of the restoration stages. The tensor contains three weeks worth of data (288,7,3). The burst is 100 samples long, and its position is randomly selected: (**a**) shows the week of the tensor where the burst is located, $\chi(:,:,1)$, corresponding to week 2. Parts (**b**–**d**) show the three steps of the restoration process for the *last sample*, the *ramp*, and the *FIR* methods, respectively.

Finally, it is worth noting that SCADA systems operating in any field in which data losses occur in bursts could also use the proposed method to complete stored data.

Author Contributions: Conceptualization, P.M.-P., M.S.-S and A.M.-S.; methodology, P.M.-P. and M.S.-S; software, A.M.-S. and P.M.-P. ; validation, P.M.-P., M.S.-S and A.M.-S.; formal analysis, P.M.-P. and M.S.-S; investigation, P.M.-P. and A.M.-S.; resources, M.S.-S; data curation, A.M.-S.; writing–original draft preparation, P.M.-P. and A.M.-S.; writing–review and editing, P.M.-P., M.S.-S and A.M.-S.; supervision, P.M.-P. and M.S.-S; funding acquisition, P.M.-P. and M.S.-S.

Funding: Financial support by the Agency for Management of University and Research Grants (AGAUR) of the Catalan Government to Arnau Martí-Sarri is gratefully acknowledged.

Acknowledgments: We thank the company Aigües de Vic S.A. for giving us access to their databases to perform this research.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Langhammer, J.; Česák, J. Applicability of a Nu-Support Vector Regression Model for the Completion of Missing Data in Hydrological Time Series. *Water* **2016**, *8*, 560. [CrossRef]
2. Ahlheim, M.; Frör, O.; Luo, J.; Pelz, S.; Jiang, T. Towards a Comprehensive Valuation of Water Management Projects When Data Availability Is Incomplete—The Use of Benefit Transfer Techniques. *Water* **2015**, *7*, 2472–2493. [CrossRef]
3. Zhao, Q.; Zhu, Y.; Wan, D.; Yu, Y.; Cheng, X. Research on the Data-Driven Quality Control Method of Hydrological Time Series Data. *Water* **2018**, *10*, 1712. [CrossRef]
4. Ekeu-wei, I.T.; Blackburn, G.A.; Pedruco, P. Infilling Missing Data in Hydrology: Solutions Using Satellite Radar Altimetry and Multiple Imputation for Data-Sparse Regions. *Water* **2018**, *10*, 1483. [CrossRef]
5. Lamrini, B.; Lakhal, E.K.; Le Lann, M.V.; Wehenkel, L. Data validation and missing data reconstruction using self-organizing map for water treatment. *Neural Comput. Appl.* **2011**, *20*, 575–588. [CrossRef]
6. Blanch, J.; Puig, V.; Saludes, J.; Quevedo, J. Arima models for data consistency of flowmeters in water distribution networks. *IFAC Proc. Vol.* **2009**, *42*, 480–485. [CrossRef]
7. Puig, V.; Ocampo-Martinez, C.; Pérez, R.; Cembrano, G.; Quevedo, J.; Escobet, T. *Real-Time Monitoring and Operational Control of Drinking-Water Systems*; Springer: Basel, Switzerland, 2017.
8. Cugueró-Escofet, M.À.; García, D.; Quevedo, J.; Puig, V.; Espin, S.; Roquet, J. A methodology and a software tool for sensor data validation/reconstruction: Application to the Catalonia regional water network. *Control Eng. Pract.* **2016**, *49*, 159–172. [CrossRef]
9. Acar, E.; Dunlavy, D.M.; Kolda, T.G.; Mørup, M. Scalable tensor factorizations for incomplete data. *Chemom. Intell. Lab. Syst.* **2011**, *106*, 41–56. [CrossRef]
10. Signoretto, M.; Van de Plas, R.; De Moor, B.; Suykens, J.A. Tensor versus matrix completion: A comparison with application to spectral data. *IEEE Signal Process. Lett.* **2011**, *18*, 403. [CrossRef]
11. Mørup, M. Applications of tensor (multiway array) factorizations and decompositions in data mining. *Wiley Interdiscip. Rev. Data Min. Knowl. Discov.* **2011**, *1*, 24–40. [CrossRef]
12. Kolda, T.G.; Bader, B.W. Tensor decompositions and applications. *SIAM Rev.* **2009**, *51*, 455–500. [CrossRef]
13. Cichocki, A.; Mandic, D.; De Lathauwer, L.; Zhou, G.; Zhao, Q.; Caiafa, C.; Phan, H.A. Tensor decompositions for signal processing applications: From two-way to multiway component analysis. *IEEE Signal Process. Mag.* **2015**, *32*, 145–163. [CrossRef]
14. Comon, P. Tensors: A brief introduction. *IEEE Signal Process. Mag.* **2014**, *31*, 44–53. [CrossRef]
15. Vaseghi, S.V. *Advanced Digital Signal Processing and Noise Reduction*; John Wiley & Sons: Chichester, UK, 2008.
16. Harshman, R.A. Foundations of the PARAFAC procedure: Models and conditions for an "explanatory" multimodal factor analysis. In *Working Papers in Phonetics*; UCLA: Los Angeles, CA, USA, 1970; Volume 16, pp. 1–84.
17. Sørensen, M.; Lathauwer, L.D.; Comon, P.; Icart, S.; Deneire, L. Canonical polyadic decomposition with a columnwise orthonormal factor matrix. *SIAM J. Matrix Anal. Appl.* **2012**, *33*, 1190–1213. [CrossRef]
18. Tucker, L.R. Some mathematical notes on three-mode factor analysis. *Psychometrika* **1966**, *31*, 279–311. [CrossRef]
19. Carroll, J.D.; Chang, J.J. Analysis of individual differences in multidimensional scaling via an N-way generalization of "Eckart-Young" decomposition. *Psychometrika* **1970**, *35*, 283–319. [CrossRef]
20. De Lathauwer, L.; De Moor, B.; Vandewalle, J. A multilinear singular value decomposition. *SIAM J. Matrix Anal. Appl.* **2000**, *21*, 1253–1278. [CrossRef]

21. Sidiropoulos, N.D.; De Lathauwer, L.; Fu, X.; Huang, K.; Papalexakis, E.E.; Faloutsos, C. Tensor decomposition for signal processing and machine learning. *IEEE Trans. Signal Process.* **2017**, *65*, 3551–3582. [CrossRef]
22. Liu, J.; Musialski, P.; Wonka, P.; Ye, J. Tensor completion for estimating missing values in visual data. *IEEE Trans. Pattern Anal. Mach. Intell.* **2013**, *35*, 208–220. [CrossRef] [PubMed]
23. Roughan, M.; Zhang, Y.; Willinger, W.; Qiu, L. Spatio-temporal compressive sensing and internet traffic matrices. *IEEE/ACM Trans. Netw.* **2012**, *20*, 662–676. [CrossRef]
24. Wang, L.; Xie, K.; Semong, T.; Zhou, H. Missing Data Recovery Based on Tensor-CUR Decomposition. *IEEE Access* **2018**, *6*, 532–544. [CrossRef]
25. Gandy, S.; Recht, B.; Yamada, I. Tensor completion and low-n-rank tensor recovery via convex optimization. *Inverse Probl.* **2011**, *27*, 025010. [CrossRef]
26. Zhao, Q.; Zhang, L.; Cichocki, A. Bayesian CP factorization of incomplete tensors with automatic rank determination. *IEEE Trans. Pattern Anal. Mach. Intell.* **2015**, *37*, 1751–1763. [CrossRef] [PubMed]
27. Dunlavy, D.M.; Kolda, T.G.; Acar, E. *Poblano v1.0: A Matlab Toolbox for Gradient-Based Optimization*; Technical Report SAND2010-1422; Sandia National Laboratories: Albuquerque, NM, USA; Livermore, CA, USA, 2010.
28. Bader, B.W.; Kolda, T.G.; others. MATLAB Tensor Toolbox Version 2.6. February 2015. Available online: http://www.sandia.gov/~tgkolda/TensorToolbox/index-2.6.html (accessed on 11 November 2018).
29. Vaidyanathan, P. The theory of linear prediction. In *Synthesis Lectures on Signal Processing*; Morgan & Claypool Publishers: San Rafael, CA, USA, 2007; Volume 2, pp. 1–184.
30. Kailath, T.; Sayed, A.H.; Hassibi, B. Linear Estimation. In *Number Book*; Prentice Hall: Upper Saddle River, NJ, USA, 2000.
31. Mitter, S. Linear Estimation-T. Kailath, AH Sayed, and B. Hassibi. *IEEE Trans. Autom. Control* **2003**, *48*, 177–182.
32. Wang, Z.; Yang, F.; Ho, D.W.; Liu, X. Robust finite-horizon filtering for stochastic systems with missing measurements. *IEEE Signal Process. Lett.* **2005**, *12*, 437–440. [CrossRef]
33. Humpherys, J.; Redd, P.; West, J. A fresh look at the Kalman filter. *SIAM Rev.* **2012**, *54*, 801–823. [CrossRef]
34. Junninen, H.; Niska, H.; Tuppurainen, K.; Ruuskanen, J.; Kolehmainen, M. Methods for imputation of missing values in air quality data sets. *Atmos. Environ.* **2004**, *38*, 2895–2907. [CrossRef]
35. Quinteros, M.E.; Lu, S.; Blazquez, C.; Cárdenas-R, J.P.; Ossa, X.; Delgado-Saborit, J.M.; Harrison, R.M.; Ruiz-Rudolph, P. Use of data imputation tools to reconstruct incomplete air quality datasets: A case-study in Temuco, Chile. *Atmos. Environ.* **2019**, *200*, 40–49. [CrossRef]
36. Yang, Y.; Ma, J.; Osher, S. Seismic data reconstruction via matrix completion. *Inverse Probl. Imaging* **2013**, *7*, 1379–1392. [CrossRef]
37. Box, G.E.; Jenkins, G.M. *Time Series Analysis: Forecasting and Control*; Revised ed.; Holden-Day: San Francisco, CA, USA, 1976.
38. Zhang, Z.; Ely, G.; Aeron, S.; Hao, N.; Kilmer, M. Novel methods for multilinear data completion and de-noising based on tensor-SVD. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2014; pp. 3842–3849.
39. Zhang, Z.; Aeron, S. Exact Tensor Completion Using t-SVD. *IEEE Trans. Signal Process.* **2017**, *65*, 1511–1526. [CrossRef]
40. Filipović, M.; Jukić, A. Tucker factorizatio with missig data with applicatio to low-rak tesor completio. *Multidimens. Syst. Signal Process.* **2015**, *26*, 677–692. [CrossRef]
41. Yokota, T.; Zhao, Q.; Cichocki, A. Smooth PARAFAC decomposition for tensor completion. *IEEE Trans. Signal Process.* **2016**, *64*, 5423–5436. [CrossRef]
42. Kressner, D.; Steinlechner, M.; Vandereycken, B. Low-rank tensor completion by Riemannian optimization. *BIT Numer. Math.* **2014**, *54*, 447–468. [CrossRef]
43. Sole-Casals, J.; Caiafa, C.F.; Zhao, Q.; Cichocki, A. Brain-Computer Interface with Corrupted EEG Data: A Tensor Completion Approach. *Cognit. Comput.* **2018**, *10*, 1062. [CrossRef]