An Efficient and Configurable Preprocessing Algorithm to Improve Stability Analysis

An Efficient and Configurable Preprocessing Algorithm to Improve Stability Analysis

I. Sesia[1], *Member, IEEE,* E. Cantoni[1], A. Cernigliaro[2], G. Signorile[1], G. Fantino[1], and P. Tavella[1], *Senior Member, IEEE*

[1]Istituto Nazionale di Ricerca Metrologica, Strada delle Cacce 91, 10135,Torino, Italy
[2]AizoOn, Via Po 14, 10123, Torino, Italy

E-mail: i.sesia@inrim.it

Abstract

The Allan variance (AVAR) is widely used to measure the stability of experimental time series. Specifically, the AVAR is commonly used in space applications, such as for monitoring the clocks of the Global Navigation Satellite Systems (GNSSs).

In these applications the experimental data present some peculiar aspects which are not generally encountered when the measurements are carried out in a laboratory. Space clocks data can in fact present outliers, jumps and missing values which corrupt the clock characterization. Therefore, an efficient preprocessing is fundamental to ensure a proper data analysis and to improve the stability estimation performed with the AVAR or other similar variances.

In this work we propose a preprocessing algorithm and its implementation in a robust software code (in MATLAB® language) able to deal with time series of experimental data affected by nonstationarities and missing data; our method is properly detecting and removing anomalous behaviors, hence making the subsequent stability analysis more reliable.

I. INTRODUCTION

The statistical analysis of data series is sometimes a difficult task due to anomalous behaviors in the series which are not caused by the physical quantity under test, but rather by spurious and artificial external causes, such as the measurement or the data transmission system. Such anomalies have to be detected and properly removed before the data processing and analysis steps. This phase is called preprocessing and it is a crucial stage in data processing, sometimes more demanding than the analysis process itself. Preprocessing is fundamental when the clock stability is characterized through the Allan variance or similar variances, like the dynamic Allan variance [1]. Removing outliers, deterministic behaviors and clock anomalies reduces in fact the bias of the computed Allan variance.

Based on the experience gained working on ground clock data as well as on GNSS space clock data [2], we have developed a method to preprocess time series of experimental data affected by missing data, outliers, and jumps. In this work we mainly focus on the outliers filtering stage, which is characterized by three key concepts. First, we detect the outliers with a sliding window approach, which is well suited in case of clock nonstationarities. Then, to reduce the rejection of false outliers, we validate outliers before removing them. Finally, we apply the sliding window and validation approaches combining two different types of filters on the same data series, in cascade: we remove the outliers by applying first a sliding minimum sigma (SMS) filter, which we developed as an extension of the classical σ filter, and then we use a median absolute deviation (MAD) filter. This helps in properly identifying different type of outliers.

The article is organized as follows. In Sect. II we highlight the importance of clock data preprocessing before analysis. In Sect. III we describe the proposed preprocessing methodology. In Sect. IV we focus on the outlier filtering method, we review the classical σ and MAD-based filters, and we extend them to the case of missing data and nonstationarities. We present the innovation and effectiveness of our approach, and we report examples of application to experimental clock data. Finally, in Sect. V we illustrate the Graphical User Interface implementing the proposed preprocessing algorithm, which offers a simple and intuitive way to configure the preprocessing parameters.

II. THE IMPORTANCE OF DATA PREPROCESSING

When ground and space clocks are routinely analyzed, it is not unusual to find time series containing outliers, which corrupt data analysis and must be suitably filtered. Moreover, especially

in GNSS applications, the clock estimates may present a lot of missing data, as well as noise injected by the system and jumps, either due to maintenance operations or to real malfunctioning on board the satellite.

In Fig. 1 ideal clock data are compared with real clock data, which may present nonstationarities that are not part of the clock nominal stationary behavior.

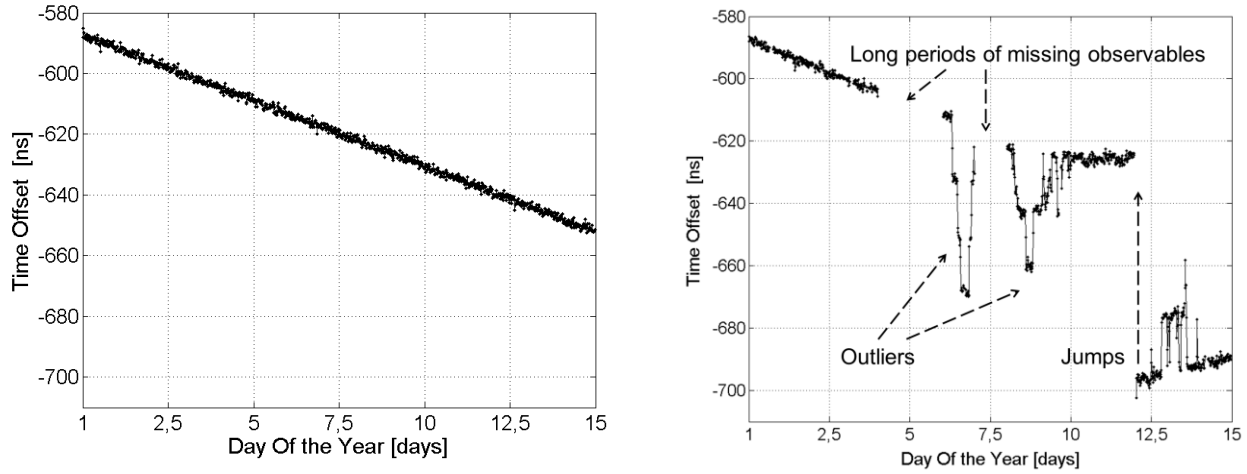

Fig. 1. On the left, ideal clock measurement data. On the right, real clock data are reported; the Italian reference time scale is compared to another remote timescale by using an INRIM GNSS receiver which was experiencing problems; the observed anomalous behaviors are not due to the compared remote timescales, which were monitored also via other techniques. The considered period is 16-30 August 2012.

Without a proper data preparation, the results of clock analysis can be degraded by the presence of anomalous values. Hence, data preprocessing is essential to ensure a robust data analysis and a proper clock characterization.

III. METHODOLOGY

We implemented the proposed preprocessing algorithm in a dedicated MATLAB® routine, which properly inserts *NaN* (*Not a Number*) values when data are not available and which then preprocess only existing data, therefore handling with care the data gaps.

The complete data preparation process we have developed includes three main functions: *Basic Analysis*, *Jump Detection* and *Outliers Filtering.* The proposed preprocessing methodology scheme is reported in Fig. 2.

First, input data are equally spaced by replacing missing values with *NaN* values.

Then, thanks to the Graphical User Interface (GUI) described in Section VI, the user can choose which of the preprocessing steps and relative sub-functions to perform and in which sequence, depending on the needs.

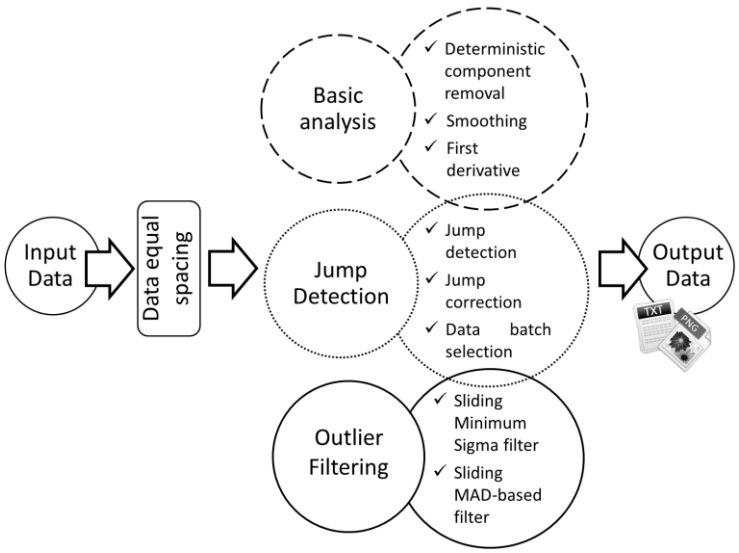

Fig. 2. Proposed preprocessing methodology scheme.

The *Basic Analysis* of the input signal can be performed in terms of:

- removal of deterministic components (periodic fluctuations, frequency drift, etc.), in case it may be of interest [3]
- noise reduction, by data smoothing
- first derivative, if needed, in case frequency measures must be obtained from phase measures

The *Jump Detection* stage allows to detect possible jumps on the analyzed signal, and to compensate them if needed or to select the batch of data for on which to perform the subsequent analysis. The detection algorithm has been developed in collaboration with Politecnico di Torino and details on its implementation can be found in [4], [5], [6], [7].

Finally, the *Outliers Filtering* step is in charge of outliers identification and removal.

In the following Sections we mainly focus on the *Outliers Filtering* step. The proposed improved outliers filtering method is presented, along with examples of application to experimental clock data.

IV. FILTERING THE OUTLIERS

When the input clock data present gaps, long periods of missing measures and nonstationarities, the outliers filtering stage asks for particular care. We here propose an extension of the classical outliers filtering techniques, resulting in an improved and robust outliers filtering method.

The proposed extended method is based on the classical σ filter and on the more robust MAD-based filter, both widely used in literature [8], [9], [10], [11].

We here review the classical definition of the σ and MAD-based filters, and we present the innovation of the proposed method.

A. Classical filtering methods

Sigma filter

Given an input signal $x(t_i)$ with N samples, its mean value μ_x is estimated as

$$\mu_x = \frac{1}{N}\sum_{i=1}^{N} x(t_i) \tag{1}$$

The dispersion of the data around the mean value is given by $\hat{\sigma}$, which is estimated by the square root of the empirical variance

$$\hat{\sigma} = \sqrt{\frac{1}{N-1}\sum_{i=1}^{N}(x(t_i) - \mu_x)^2} \tag{2}$$

The estimated $\hat{\sigma}$ is then multiplied by a factor k, where k is an integer positive value, and $x(t_i)$ is identified as outlier and discarded if

$$|x(t_i) - \mu_x| > k \cdot \hat{\sigma} \tag{3}$$

Fig. 3 shows an example of application to the experimental clock data reported in Fig. 1 (right side) of a σ filter with thresholds $k = 1$ and $k = 2$ on the entire data set. We notice that the standard σ filter is not effective when nonstationarities are present in the N input data. In this case, the use of the entire data set for the evaluation of the dispersion is not appropriate.

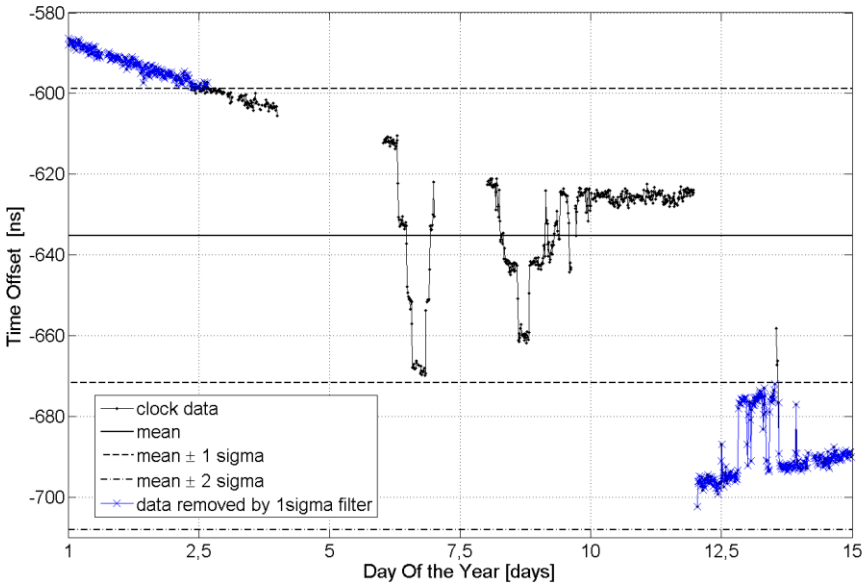

Fig. 3. Example of application of a σ filter to the experimental clock data reported in Fig. 1 (right side). The filter is applied on the whole dataset. Detected outliers are marked by a cross.

MAD-based filter

The MAD-based filter is obtained by applying the Hampel identifier [8], [10], which replaces the classical mean and standard deviation with the outlier-resistant median and median absolute deviation from the median, respectively.

As in [10], the median x^m of a generic time series is obtained by first rank-ordering it from smallest to largest, i.e.:

$$x(t_1) \leq x(t_2) \leq \cdots \leq x(t_{N-1}) \leq x(t_N) \tag{4}$$

and then taking x^m as either the middle value (if N is odd) or the average of the middle two values (if N is even).

The MAD scale estimate is then defined as:

$$S = 1.4826 \cdot median\{|x(t_i) - x^m|\} \tag{5}$$

where the factor 1.4826 in (5) is chosen so that the expected value of S is equal to the standard deviation σ for normally distributed data.

A threshold k (with k as an integer positive value) is then introduced and if

$$|x(t_i) - x^m| > k \cdot S \tag{6}$$

$x(t_i)$ is then detected as outlier.

Fig. 4 shows an example of application to the experimental clock data reported in Fig. 1 (right side) of a MAD-based filter with thresholds k =1 and k =2. We note that when the MAD-based filter is applied to the whole dataset, no outliers are removed with k =2, whereas, for k =1, the MAD-based filter does not appropriately filter the outliers, due to the presence of jumps on the data.

In case the clock behavior is nonstationary, it is then preferable to perform the outlier filtering on sliding windows rather than on the entire data set, as will be detailed in the next sub-Section *B*.

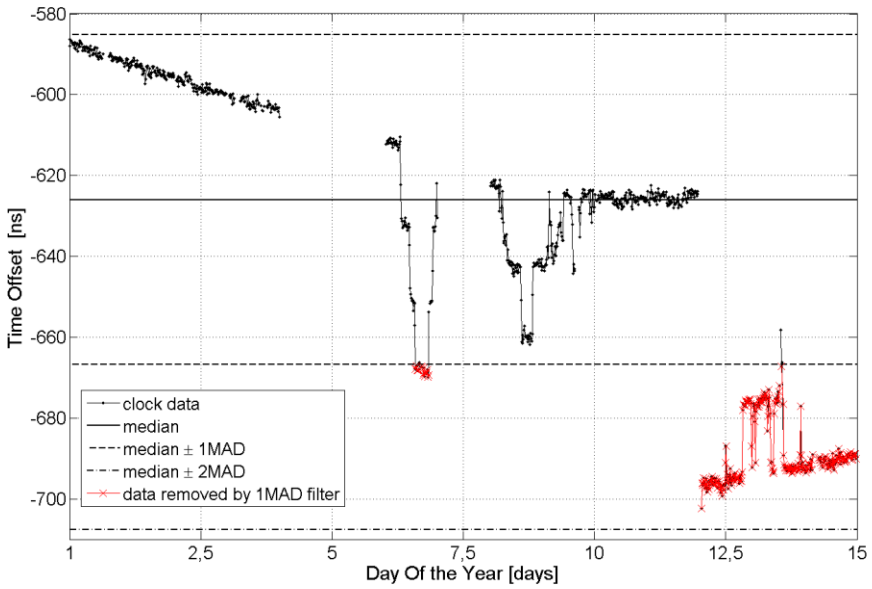

Fig. 4. Example of application of a MAD-based filter to the experimental clock data reported in Fig. 1 (right side). The filter is applied on the whole dataset. Detected outliers are marked by a cross.

B. Filtering outliers in case of missing data and nonstationarities

The proposed method is based on three key concepts:

- Outliers are removed over a *sliding window*
- Outliers are *validated* before removal
- Outliers Filtering is performed on *two-steps*, by combining two different filters

With the *sliding window* approach, outliers filtering is improved in case of nonstationarities. The input time series is truncated on a window of length W centered at the epoch t_i and the outlier filter is applied. The window is then slid to the next epoch t_{i+1} and the outlier filter is applied around

that epoch. Missing data are properly handled by the data equal spacing step and then by using MATLAB® commands which are correctly taking *NaN* into account in a robust way.

With the *validation*, outliers are removed only if they are detected over a certain percentage of sliding windows. In this way, the false outliers detection is significantly reduced.

In Fig. 5 we show an example of application of the MAD-based filter with $k = 2$, with a sliding window of 8 hours. No outliers validation is performed. We see that many false outliers are detected.

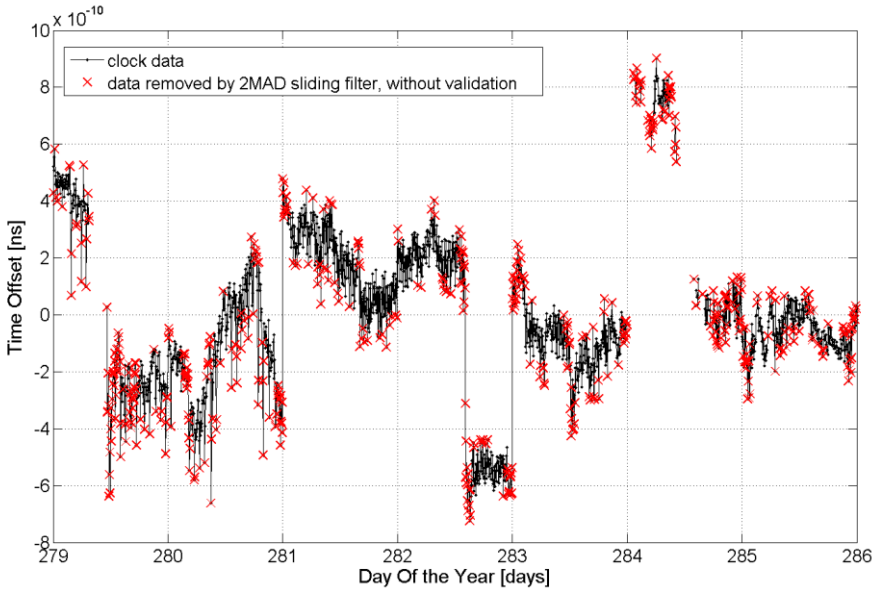

Fig. 5. Example of application of a MAD-based filter to experimental clock data. The filter is applied on a sliding window of 8 hours. Outliers are removed at first detection, without validation. Detected outliers are marked by a cross.

In Fig. 6, outliers are removed only if detected at least on 51% of the sliding windows containing that sample, that is outliers are removed only after validation.

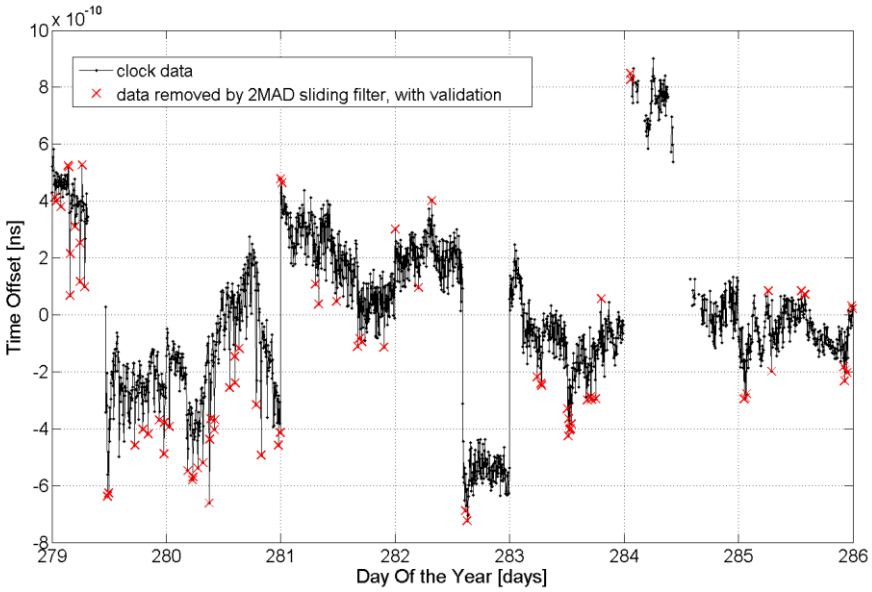

Fig. 6. Example of application of a MAD-based filter to experimental clock data. The filter is applied on a sliding window of 8 hours. Outliers are validated before removal. Detected outliers are marked by a cross.

As the comparison of Fig. 5 and Fig. 6 shows, the false outliers detection is significantly reduced when the outliers are validated before their removal. This improves the robustness of the Outlier Filtering process.

With the *two-steps* concept, we intend the subsequent application of two different filters on the same data series, as also considered in [12]. We first process the clock data with the sliding minimum sigma (SMS) filter, an improved version of the classical σ filter, and then with a MAD-based filter. Both filters are applied with the sliding window and the validation approaches described previously. We now describe the SMS filter.

Sliding minimum sigma (SMS) filter

The proposed SMS filter is a sliding version of the classical σ filter. The SMS filter repeatedly evaluates the standard deviation $\hat{\sigma}$ on a window sliding over the whole data set, then select the minimum of the estimated $\hat{\sigma}$ values, and finally multiplies it by a factor k to obtain the threshold used for outlier validation and rejection. We give a mathematical description of the SMS filter.

We consider a phase signal $x(t_i)$, whose missing values are replaced by *NaN* values (within the *Basic Analysis* phase). First, we obtain the signal $x_W(t_i)$ by truncating the signal $x(t_i)$ on a window of length W centered about the time epoch t_i. Second, we estimate the standard deviation $\hat{\sigma}(t_i)$ of $x_W(t_i)$. Then, we select the minimum of all the obtained standard deviation estimates on the entire data set,

$$\hat{\sigma}_{min} = \min(\hat{\sigma}(t_i)) \tag{7}$$

Finally, we define the σ-threshold $\hat{\sigma}_k$ as

$$\hat{\sigma}_k = k \cdot \hat{\sigma}_{min} \tag{8}$$

where k is a non-negative integer number.

Subsequently, we slide the window again over the phase signal $x(t_i)$ and we detect as outliers the samples within the window which fall outside the $\hat{\sigma}_k$ range from the mean value.

In Fig. 7 we show the estimation of $\hat{\sigma}_{min}$ (black cross) for the experimental clock data reported in Fig. 1 (right side).

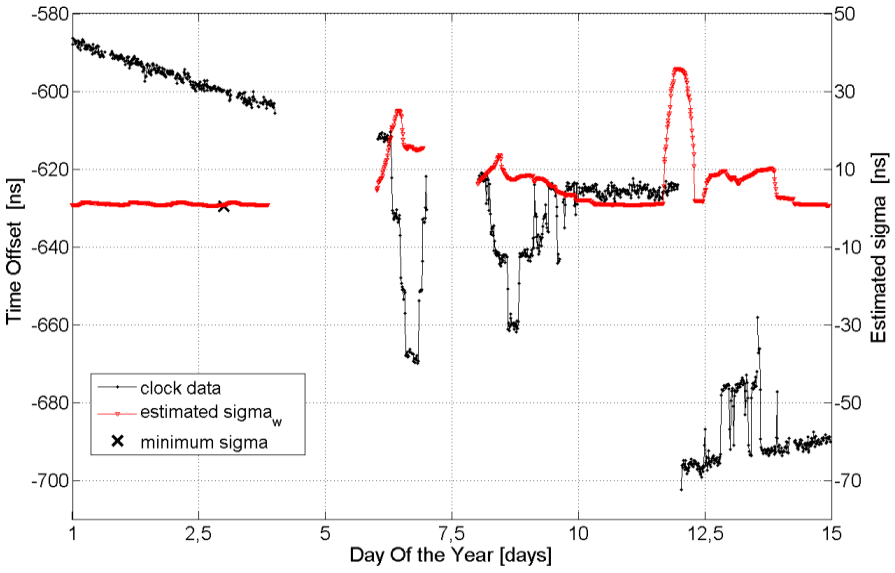

Fig. 7. Sliding minimum sigma filter: estimation of $\hat{\sigma}_{min}$ (black cross) for the experimental clock data reported in Fig. 1 (right side).

We then apply the SMS filter for k = 3 and W = 5 hours, with a validation threshold of 51%. Results are reported in Fig. 8. Most of the outliers are properly removed.

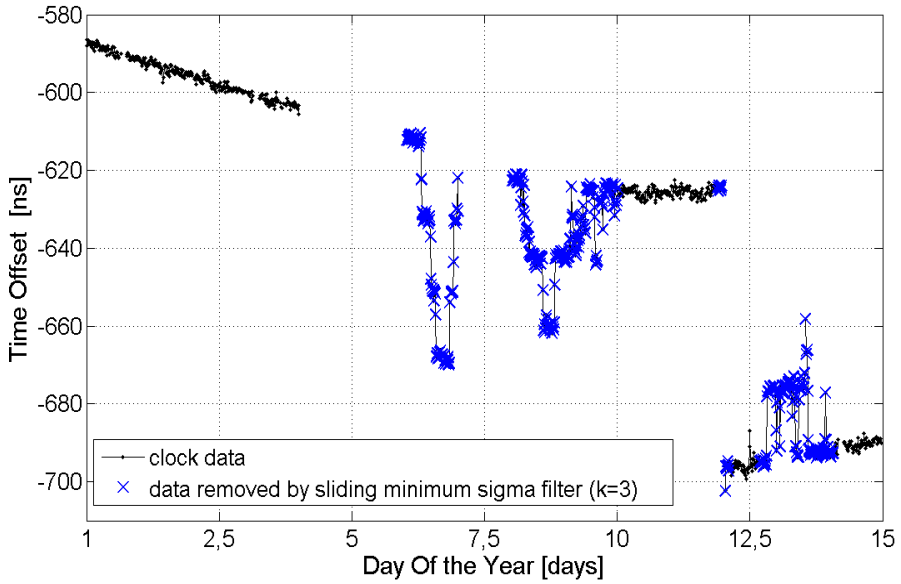

Fig. 8. Example of application of the SMS filter to the experimental clock data reported in Fig. 1 (right side). The SMS filter is applied on a sliding window of 5 hours. Outliers are validated before removal. Detected outliers are marked by a cross.

Finally, we apply a sliding MAD-based filter (for k = 2, W = 5 and validation threshold of 51%) to the clock data reported in Fig. 1 (right side), after removal of the outliers detected in Fig. 8. The result is shown in Fig. 9. We note that the proposed extended method properly filters outliers, also when the clock data present gaps and jumps.

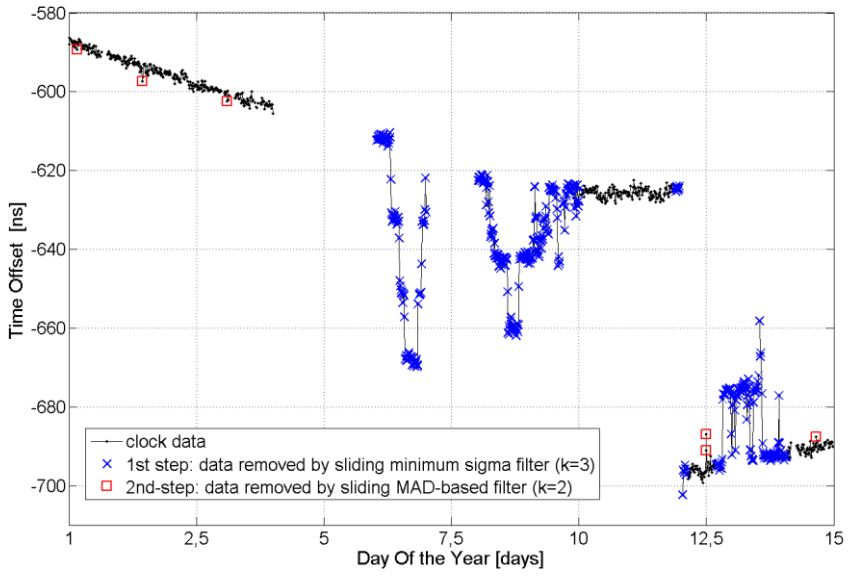

Fig. 9. Example of application of two-step filter. A sliding MAD-based filter is applied to the filtered clock data reported in Fig. 8. The filter is applied on a sliding window of 5 hours. Outliers are validated before removal. Outliers detected by the SMS filter are marked by a cross, whereas outliers detected by MAD-based filter are marked by a square.

The filter threshold k and the length W of the sliding window have to be properly configured and tuned on the considered input data, and the user can easily do that through the Graphical User Interface described in the next Section.

Based on the analysis of many ground and space clocks data, we identified as proper thresholds $k = 3$ for the SMS filter and $k = 2$ for the MAD-based filter, with a sliding window length of a few hours.

Another example of application to experimental clock data of the proposed filtering method is reported in Fig. 10. We note that the space clock data are affected by outliers, discontinuities, and gaps. Outliers are appropriately removed by the two-step filter.

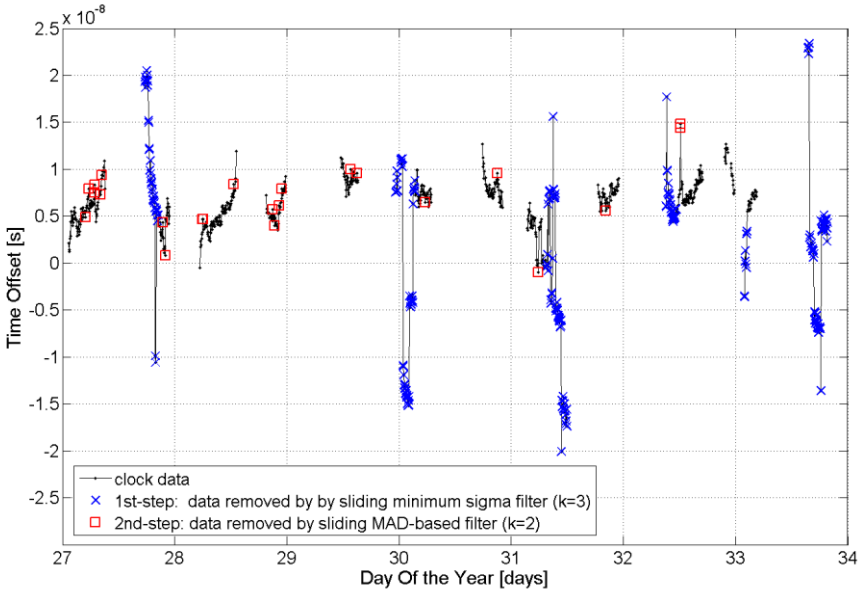

Fig. 10. Example of application of the proposed extended filtering method. We first apply the SMS filter (with $k = 3$), then the MAD-based filter ($k = 2$). The length of the sliding window is 2 hours and the validation threshold is 51% for both steps. Outliers detected by the SMS filter are marked by a cross, whereas outliers detected by MAD-based filter are marked by a square.

V. PREPROCESSING AND STABILITY ESTIMATION

The Allan variance estimation is based on the assumption of using equally spaced and stationary data (like the one reported in Fig. 1, right side). But in case data are not evenly spaced and present nonstationarities, like for example the experimental data reported in Fig. 1 (left side), a correct preprocessing and a robust AVAR estimator [13], [14] are essential to ensure a proper stability analysis.

Besides, in case of missing data or dead times we obtain different performances depending on whether time or average frequency deviation measurements are used to estimate the AVAR, and when clock phase data are available is recommended to estimate the AVAR from time deviations [13], [15]. For this reason in this work we mainly focus on the preprocessing of clock phase data, to improve a subsequent stability analysis. Nevertheless, the proposed method can also be applied to clock frequency data as well. Actually, in some cases, the identification of outliers maybe easier on frequency data rather than on clock phase data. By using the implemented GUI the user can easily perform the phase-to-frequency conversion through the dedicated *Basic Analysis - First derivative* function (see next Section VI for details).

We now present an example of application of the proposed preprocessing method with subsequent stability analysis.

We consider the clock data reported in Fig. 1 (right side), and we apply the preprocessing method by combining the Jump Detection and the Outlier Filtering functions. With the Jump Detector we identify a phase jump around Day of the Year 12, and we choose to select the batch of data just before the jump, for subsequent analysis. Then, we apply on the selected data batch the proposed filtering method, where

- the length of the sliding window is 5 hours
- the validation threshold is 51%
- the SMS filter is applied with $k = 3$
- the MAD-based filter is applied with $k = 2$

Results are reported in Fig. 11, and are consistent with what reported in Fig. 9 but limited to the selected data batch excluding the detected phase jump .

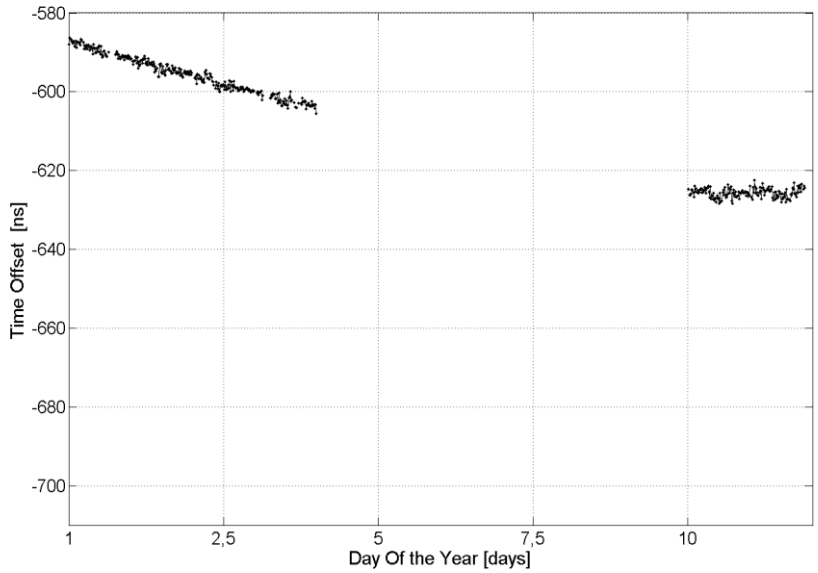

Fig. 11. Result of our preprocessing method (combining Jump Detector and Outliers Filtering functions) applied to the experimental clock data reported in Fig. 1 (right side).

Finally, we compute the Allan Deviation (ADEV) of the experimental clock data before (Fig. 1, right side) and after (Fig. 11) preprocessing. We also evaluate the ADEV of the ideal clock data reported in Fig. 1 (left side), which were obtained by simulating a noise with the same statistics and deterministic trend of the experimental clock data reported in Fig. 1 (right side). Results are reported in Fig. 12.

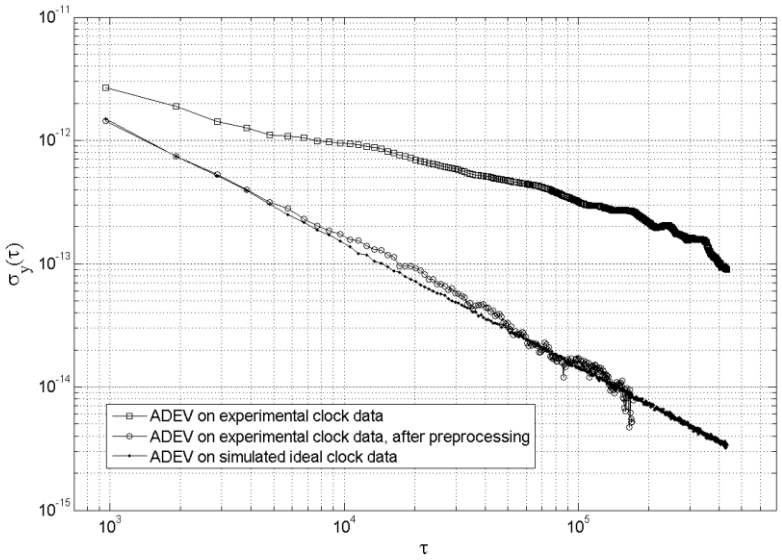

Fig. 12. Allan Deviation of the experimental clock data reported in Fig. 1 (right side) and in Fig. 11, and of the ideal clock data reported in Fig. 1 (left side). ADEV estimated on the whole experimental data set, before preprocessing, is marked by squares. ADEV estimated on the experimental data after preprocessing is marked by circles. ADEV estimated on the ideal clock data is marked by dots.

Fig. 12 shows that the Allan deviation of preprocessed data (marked by circles) is fully in line with the one estimated on the ideal clock data (marked by dots). Hence, the developed preprocessing methodology is effective on the experimental clock data, and it ensures a proper stability analysis.

VI. GRAPHICAL USER INTERFACE

Preprocessing is a crucial, although delicate, task. In particular, the outlier filtering phase request for particular care. Actually, one should remove only those outliers which are not caused by the clock itself, but rather by external effects and should try to understand if there is a physical reason for them as well as try to identify their actual cause. Hence it is important to provide the user with a Graphical User Interface (GUI), so that the automatic processes can be supported by expert user visual inspection and interaction.

As presented in Sect. III, the proposed preprocessing algorithm includes different steps. Thanks to the GUI that we are implementing in MATLAB® language, such steps can be performed by the user in any sequence depending on the needs.

In Fig. 13 we report a screenshot of the GUI.

On the left side of the GUI, the user can easily choose the analysis to be performed as well as the routine execution order. On the right side, configuration parameters are set. A preview of the input clock data is also reported. We provide the user with figures and ASCII files reporting the output of the preprocessing analysis, including a list of removed outliers and a log with indication of the executed preprocessing steps.

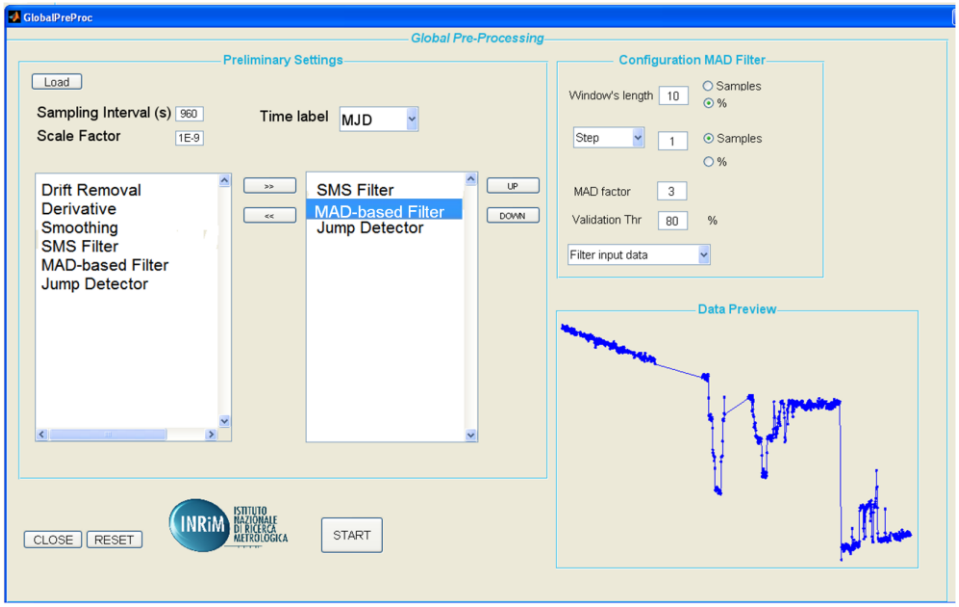

Fig. 13. Screenshot of the Graphical User Interface implementing the proposed preprocessing method.

VII. CONCLUSIONS

The Allan variance is commonly used in space applications, for example for monitoring and characterizing the clocks of GNSSs. In these applications the clock data may present gaps, long periods of missing measures, outliers, and nonstationarities. Therefore, data preprocessing is essential to ensure a reliable stability analysis. In particular, the outliers filtering phase asks for particular care. Since classical filtering techniques are not always effective in case of missing data and nonstationarities, in this paper we have presented an extended and robust outliers filtering method, to preprocess experimental data unevenly spaced in time and affected by jumps. The proposed method is effective and robust to preprocess such kind of data, as we have shown by applying it to clock experimental measures.

VIII. REFERENCES

[1] L. Galleani, P. Tavella, "Dynamic Allan variance," IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control, UFFC, vol. 56, no. 3, March 2009, pp450-464.

[2] P. Waller, F. Gonzalez, S. Binda, I. Sesia, I. Hidalgo, G. Tobias, P. Tavella, "The In-orbit Performances of GIOVE Clocks," IEEE Trans .UFFC vol 57, n 3, March 2010.

[3] A. Cernigliaro and I. Sesia, 2012, "INRIM Tool for Satellite Clock Characterization: Frequency Drift Estimation and Removal," *MAPAN Journal of Metrology Society of India*, ***27***, 41-48.

[4] L. Galleani and P. Tavella, Nov. 2010, "An algorithm for the detection of frequency jumps in space clocks," *Proc. of 42nd Annual PTTI Meeting*, Reston, VA.

[5] L. Galleani, P. Tavella, "Detection and identification of atomic clock anomalies," Metrologia 45, 6, (2008) S127-S133.

[6] L. Galleani, P. Tavella, "Detection of Atomic Clock Frequency Jumps with the Kalman Filter," IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control UFFC March 2012. vol. 59, no. 3, p. 504-509, March 2012.

[7] E. Nunzi, L. Galleani, P. Tavella, P. Carbone, "Detection of anomalies in the behavior of atomic clocks," IEEE Trans. on Instrumentation and Measurement, vol. 56, n. 2, April, 2007, pp. 523-528

[8] L. Davies, U. Gather, 1993, "The identification of multiple outliers," *J. Amer. Statist. Assoc.*, *88*, 782-801.

[9] R.K. Pearson, 2011, "Exploring process data," *J. Process Contr.*, *11*, 179-194.

[10] R.K. Pearson, 2002, "Outliers in Process Modelling and Identification," *IEEE Transactions on Control Systems Technology*, *10*, 55-63.

[11] B. Parry, 2004, "Evaluation of Outliers in Metrological data," *CAL LAB: The International Journal of Metrology*, *Jan/Feb/Mar*, 31-37.

[12] A. Harmegnies, G. Panfilo, E. F. Arias, Nov 2009, "Detection of outliers in TWSTFT data in TAI," *Proc. Of 41st Annual PTTI Meeting*, Santa Ana Pueblo, New Mexico, USA.

[13] I. Sesia and P. Tavella, 2008, "Estimating the Allan variance in the presence of long periods of missing data and outliers," *Metrologia*, *45*, 134-142.

[14] I. Sesia, L. Galleani, P. Tavella, 2011, "Application of the Dynamic Allan Variance for the Characterization of Space Clock Behavior," *IEEE Transactions on Aerospace and Electronic Systems*, *47*, 884-895.

[15] W. J. Riley, "Gaps, Outliers, Dead Time, and Uneven Spacing in Frequency Stability Data," http://www.wriley.com/Gaps.htm.