# SARI documentation - version marzo 2023

Alvaro Santamaría (alvaro.santamaria@get.omp.eu)

## Contents

---

## SARI overview

SARI allows you to visualize discrete time series data, analyze them individually, fit unidimensional models interactively and save the results using a WYSIWYG web interface.

It has been developed in the R programming language, under the interactive framework of the Shiny R package. Currently tested in R version 4.1.0.

The development of SARI started in 2017 to analyze GNSS position time series from the RESIF/RENAG network (RESIF 2017) and it is oriented towards this kind of dataset, but any other type of series can be used as long as the series format is simple, consistent and does not have uncommented alphabetic characters.

This document describes the different options and functionalities implemented in the current version of SARI. Within this document:
GUI options are given in this `red lettering`.
GUI specific input values and files are given in *italics*.
Local and external links are given in this blue.

Further details and some examples can be found in
Santamaría-Gómez, A. (2019) SARI: interactive GNSS position time series analysis software. GPS solutions, 23:52. DOI: 10.1007/s10291-019-0846-y

A SARI prebuilt Docker image is available at https://hub.docker.com/r/alvarosg/sari

The code source and installation instructions can be found at https://github.com/alvarosantamariagomez/sari

A ~40 min video tutorial is also available here. This tutorial was made with SARI version "mayo 2021".

The changelog is available here (only available from the online help).

Current SARI version: *marzo 2023* – "I know this steak doesn't exist. I know that when I put it in my mouth, the Matrix is telling my brain that it is juicy and delicious"

# SARI mechanics

The SARI interface is divided into two panels with separate vertical scrolling: the left panel contains all the processing options while the right panel is for visualization of the series and the analysis.

Click to expand/collapse any of the following five blocks on the left panel as needed.

Note that some OS and some web browsers apply a zoom to all web pages by default. It is recommended to adjust the browser zoom (usually ctrl + mouse wheel) so that all the controls, buttons, plots, etc. on both panels fit comfortably on the screen.

## I. Input data and format

This block allows uploading and setting the series format before plotting. The input series file must be a text file with standard encoding (ASCII, UTF-8) and organized in columns generally corresponding to the epochs, the observed variables and their error bars. The series does not need to be evenly sampled.

The user can upload GNSS series from local files in the standard *NEU/ENU* format (North/East, East/North, Up) or in the file formats produced by *PBO* (with extension *.pos*, version 1.1.0) and *NGL* (with extension *.tenv3*). The user only needs to set the time unit of the input series (days, weeks or years).

**NEW FEATURE:** Alternatively, the user can upload GNSS series from online files available at specific web servers. This is done via three field/value pairs (station, server and product) that must be included after the base SARI URL in any order. Similar fields are available for uploading a secondary series (see example below and the Ancillary information block).

This feature allows webmasters of any GNSS series database (or any other type of series) to include a link on their webpages to open a specific GNSS series directly with SARI on a new browser tab.

At this moment, the following servers and products are available via the URL:

| Server | Products |
|---|---|
| RENAG | UGA |
| NGL | FINAL, RAPID |
| EUREF | PBO |
| UNAVCO | CWU, NMT, PBO |
| EOSTLS | ATMIB, ATMMO, ECCO, ECCO2, ERA5IB, ERA5TUGO, ERA5HYD, ERAHYD, ERAIN, GLDAS, GLDAS2, GLORYS, GRACE, MERRA, MERRA2ATM, MERRA2HYD |

For example, adding this query string

?product=UGA&station=PIMI&server=RENAG&server2=NGL&product2=FINAL&station2=PIMI

at the end of the SARI base URL (both local on RStudio or remote on Shinyapps) will start a new SARI session, automatically upload and plot the *UGA* GNSS position series of the station *PIMI* from the *RENAG* server, and also the *NGL FINAL* series of the same station as secondary series (see example below and the Ancillary information block).

The *EOSTLS* series are those computed in the center of figure (CF) frame.

Contact the author to add more servers or products.

Once uploaded and plotted (see the Plot controls block), the three coordinates components of the GNSS series will be shown in separate tabs: 1st, 2nd and 3rd component, respectively. The components are plotted in the same order as the columns in the input file (i.e. NEU or ENU for instance). For PBO series, it is NEU-like and for NGL series, it is ENU-like. When plotting a primary and a secondary series with different formats, use the option `N@E` to match the coordinate components North and East (see the Ancillary information block).

For series in a different format, the user needs to check the *1D* option and then set the column separator (blanks, commas or semi-colons) and the column numbers containing the epochs, the variable and the error bars if available.

The input file can contain comments anywhere identified by a # at the beginning of the line. These lines will be skipped. In case uncommented text is found, the behavior depends on the requested series format:

Any non-numeric value in a NEU/ENU or 1D series will make the full line to be skipped.

For the PBO and NGL series, the headers are recognized and skipped, also the first two columns of the NGL files and the last column of the PBO files. However, any other non-numeric value in these files will stop the app.
Data records having a NA, NaN or Inf/inf entries will be treated as valid not-available or not-a-number numeric values and will be automatically skipped, but na, Na, nan, NAN or any other string will be considered as unwanted text.

For GNSS time series, the station ID is extracted from the first characters of the input file name and will be shown in the `series ID` box under the loaded series. The station ID can be modified by the user if necessary (see the Ancillary information block). For other type of series, this feature can be neglected.

If the format of the input file is not known (content of columns, separation, etc.), it is possible to print the first lines of the input series to assess the corresponding format before plotting using the `show series header` option.

If the series does not contain error bars, or the user does not want to use them, it is possible to turn the error bars off in the processing using the `use error bars` option. By default, they are always on.

It is possible to average the series and reduce the sampling, for instance from daily values to weekly or monthly, by using the `reduce sampling` option. If high-frequency data are not needed, averaging the series will save a lot of time when fitting models, finding discontinuities or when estimating the colored noise in the series (see the Additional fit block). Note that averaging an unevenly sampled series may affect the quality of the `periodic waveform` if the observed epochs become even less repetitive.
If there were points already removed from the series before reducing the sampling, these points will not contribute to the averaged values. Once the averaging is done, all the information of the original series, like the removed points, will be lost.
Also, if there is a secondary series being used with the `correct` or `average` options, the reduced sampling series will be obtained after the secondary series has been used. This means that the primary and secondary series are still being compared with their original sampling.

Finally, after the series have been plotted (see the Plot controls block), the `load SARI model` button allows the user to upload the model parameters from a previously saved SARI file containing a LS or KF fit (not necessarily from the same series).

Go to top of document

## II. Plot controls

Once the format of the series is set, this block allows plotting/resetting the time series and removing points/outliers. Outliers can be removed automatically by setting a residual threshold or a normalized residual threshold. They can also be selected and removed manually (see more details in the Interactive operation section).

The `all components` option changes between removing outliers for each coordinate component independently or from all components simultaneously, which is useful if one wants to join the results from each coordinate component into a single file afterwards (a NEU/ENU file for instance). This option works also when analyzing different columns using the 1D series format.

The `include in file` option will keep the removed outliers in the downloaded results file as commented lines. No fitting values will be provided for these points.

The `scrolling` option enables/disables the vertical scrolling of the left panel. By default the scrolling is enabled. When disabling it, the user will be able to take a screenshot of the full web page with the plots and all the input parameters that were used to make the plots, which is very convenient for sharing/archiving a specific analysis, making reports or marking assignments. The quality of the full page screenshot may depend on the browser/extension used.

Alternatively, the user can download the full SARI interface from the web browser (usually by pressing Ctrl+s) into an HTML file together with the corresponding web files in a separate directory with a similar name. The downloaded HTML file represents a frozen SARI session (i.e., not connected to the server) that can be reopened later in the web browser even offline. The advantage of the HTML page compared to the fixed screenshot is that all the numeric values are available to be selected and copied, so the same analysis can be easily replicated in a fresh session. However, before opening the HTML file in a web browser, it needs to be modified first. A Bash script called `SARIwebpage.sh` available here does the modifications automatically.
Note: Unfortunately, at this moment, the values of the UI interface, including the options selected by the user, are only saved in the HTML file when using Mozilla Firefox. Other web browsers, like Safari and the Google Chromium

family (Google Chrome, Microsoft Edge and Opera), do not keep this information when saving the web page. I hope this browser error will be fixed in future updates.

The new option `permanent` will permanently delete the next points that are removed from the series (see more details in the Interactive operation section). Permanently here means these points cannot be restored and they will not be shown again in the current session. The original series file is not modified and it has to be reloaded, after resetting the current session, to use all the points again. This option is deactivated by default, and once one or more points are removed in a single action, it will automatically deactivate itself so the next points to be removed will be available to be restored. When toggling points that were already removed, this option will delete them permanently, instead of restoring them. This option is intended, for instance, to remove extreme outliers that do not allow to plot a series correctly.

Go to top of document

## III. Ancillary information

This block allows for the uploading of files containing complementary information related to the analysis of GNSS position time series. The user can upload any of the following possibilities:

- a GNSS *sitelog* file (an example of this file can be found here - only available from the online help)
- a GAMIT-like *station.info* file (an example of this file can be found here - only available from the online help)
- an IGS-like soln discontinuity file (an example of this file can be found here - only available from the online help)
- a customized offset file to plot specific dates (for instance, earthquakes)
- a secondary series that can be `shown` next to the primary series in green color for comparison purposes and detection of common features, or used to either `correct` the primary series to analyze the difference or `average` both the primary and secondary series values. For instance, a model can be subtracted a posteriori from the series (loading, post-seismic, etc.) or even the series of a nearby station in order to remove the spatially correlated signal/noise and better detect the individual equipment changes, relative motion or common variations.
- a plate tectonic model to be `shown` on top of the series or `removed` from the series. The user can enter the station coordinates and the Euler's pole parameters manually or by uploading a file (an example of this file can be found here - only available from the online help).

Notes on the ancillary information:

1. The *sitelog*, *station.info*, *soln* and *customized* files can be uploaded to list/plot equipment changes and help identify position discontinuities. Antenna/receiver changes from the *sitelog* and *station.info* files are represented by solid/dashed blue lines, respectively. Changes from the *soln* file are represented by orange solid lines, and changes from the *customized offset* file are represented by solid green lines.

2. The dates from the *sitelog*, *station.info* and *soln* files will be transformed according to the time unit selected for the series. If the time unit is days, the dates will be transformed into Modified Julian Days. If the time unit is weeks, the dates will be transformed into GPS weeks.

3. The station ID (see the Input data and format block) will be used to extract the information from the *station.info*, *soln*, *customized offset*, and *plate model* files. The user is responsible for checking that the station(s) in these files actually correspond to the same station ID being used (many GNSS stations around the world share the same ID).

4. The *customized offset* file must contain one or two columns separated by spaces or tabulations. The first column contains the date in the same time units as the input series. The second column is optional and contains the station ID. If the second column is present, only the dates corresponding to the current station ID will be extracted. If the second column is missing, all the dates in the file will be used. A warning will be shown if the file contains more than two columns.

5. As an exception, the discontinuities file from the Nevada Geodetic Laboratory available here can be uploaded directly as a *customized offset* file without changing its date format.

6. When uploading a *secondary series*, both station IDs will be shown in the `series ID` box on the left panel, together with a "*&*" when using the option `show`, a "−" when using the option `correct`, or a "+" when using

the option `average`.

7. If the `show` option is selected, the *secondary series* will be plotted in green on the right y-axis and will not be included in the processing. This means that the *secondary series* could have a different sampling than the primary series. In case the `correct` or the `average` option is selected, only the common epochs between the primary and the *secondary series* will be shown. In the latter case, the *secondary series* must have observations at common epochs with the primary series because the series are intentionally neither filled nor interpolated at common epochs.

8. For NEU/ENU series, the *secondary series* must have the same format than the primary series. For 1D series, the user has the option to choose the column number and column separator for the *secondary series* independently of those already set for the primary series.

9. The y-axis of the *secondary series* is partially controlled by the `same scale` and `same axis` options with respect to the y-axis of the primary series.

10. If the format of the *secondary series* is different from the primary series (for instance *NEU* vs *NGL*, *ENU* vs *PBO* or *NGL* vs *PBO*), the `N@E` option will swap the North and East components of the *secondary series* to match the components of the primary series.

11. When the primary series is combined with a *secondary series* (using `correct` or `average`, but not `show`), the IDs from both series will be extracted from the *station.info*, *soln*, and *customized offset* files. Otherwise only the ID from the primary series will be used.

12. The *plate model* file must have six or seven values per line corresponding to the station ID (first value), the station coordinates (two values: latitude & longitude; or three values: X,Y,Z), and the Euler's pole parameters (three values: latitude,longitude,rotation or X,Y,Z rotations). If a station ID is present in more than one line, only the last one will be used. Once the *plate model* file is read, the extracted values will be shown on the left panel. The user can check and modify these values before showing/removing the plate model. If no values are being shown, it means no values were found for the current station ID shown in the `station ID` box of the left panel.

13. The units of the plate model parameters are degrees for latitude & longitude and degrees/Ma for rotations. The units of the Cartesian station coordinates (X,Y,Z) must be the same as the time series units (i.e., meters, millimeters, etc). If the station coordinates are given in latitude & longitude, SARI will try to guess if the values of the GNSS series are in meters or millimeters and then transform the station coordinates into Cartesian coordinates with the correct units. This guess may fail if the series are ambiguous, so station Cartesian coordinates should be preferred as input.

14. If any of the plate model parameters (station coordinates or Euler's pole parameters) is modified, the model will be deactivated until the user sets again the `show` or `remove` option with the new parameters.

Go to top of document

**IV. Fit controls**

This block allows fitting a model to the time series using

- Weighted least squares (LS)
- First-order extended Kalman filter/smoother (EKF)
- Unscented Kalman filter/smoother (UKF)

The fitted model will be represented by a red line on top of the original series and in the plot of the residuals (zero value line).

**Select model components**: the fitted functional model may include any combination of linear trend, higher-degree polynomial, offsets/discontinuities, sinusoidal signals, exponential and logarithmic decays.

Some of these components require additional parameters to be included in the model (e.g., reference epochs, periods, a priori values, etc.). Additionally, if using a KF fit, the a priori state and the process noise standard deviation can

be set for the trend and sinusoidal components only.

The fitted discontinuities will be represented by vertical red lines in the residual plot.

**Automatic discontinuity detection:** the `search discontinuities` button provides an automatic guesstimate of the location of probable discontinuities in the time series. The detector is based on the assessment of deviations from stability in different segments of the series using a classical linear regression model and a Bayesian Information Criterion (BIC). The algorithm is described by Zeileis et al. (2003). **WARNING:** this option is very time-consuming, see the notes below.

Notes on the Kalman filter:

1. The EKF/UKF fits include both the forward Kalman filter and the backward Kalman smoother using the Rauch-Tung-Striebel (RTS) algorithm. Only the smoothed solution is kept.

2. The UKF implementation is described by Julier and Uhlmann (2004) and the unscented RTS smoother is described by Särkkä (2008).

3. The KF fit does not include any dynamics (i.e., control inputs) and assumes additive, independent measurement and process noises (for the trend and the sinusoids). The measurement noise is time-invariant unless the error bars are in use. In that case, the relative change of the error bars is used to scale the measurement noise. The average of the actual measurement noise level used for the entire series will correspond to the measurement noise value indicated on the left panel. Data gaps are also accounted for by increasing the process noise between missing observations.

4. The user can set the expected (average) measurement noise level and the a priori values and uncertainties of the state for all the parameters being fitted or leave them blank. In the latter case, some values will be proposed. Note that the filter output strongly depends on all the provided a priori values. For instance, an extremely small a priori uncertainty on a parameter would "fix" it to its a priori value (which is useful when we know precisely the magnitude of a position offset, for instance).

5. The `compute measurement noise` option will obtain the most likely measurement noise level, within some bounds provided by the user. This is done through a maximum likelihood optimization of a preliminary UKF fit with respect to the provided process noise variances (on the trend and sinusoid only). The optimization uses the Brent (1971) method, and it may take up to ~20 iterations of the UKF fit. The 95% CI of the estimated measurement noise standard deviation will be shown next to the input bounds on the left panel.

6. Process noise values for the trend and sinusoidal variations are zero by defaul. Any positive value will make these parameters to change with time. When fitting a time-variable trend, a plot below the residual series will show the time series of the estimated trend, which is useful to assess whether the filter is too smooth or not, especially if other parameters were fitted together with the rate.

7. When fitting a time-variable sinusoid, there is an option to apply the process noise to the sine component (S) only or to both the sine and cosine (C) components independently. The first option will capture amplitude variations, but also phase (and maybe even frequency) variations. The second option may only capture amplitude variations, but this is not guaranteed as the estimates of the sine and cosine terms are not constrained to vary the same amount.

8. Due to the KF fit being slower than the LS fit (see details in the Known issues section), the user has to set all the parameters of the filter first and then push the `run KF` button. This way, the filter is run once and not each time one of the parameters is changed, as for the LS fit. The parameters left blank will be completed automatically (at the user's risk).

9. If the user has fitted the same model with LS before fitting it with the KF, some of the LS-estimated values will be used as the a priori state for the KF.

Notes on the sinusoidal fitting:

1. By default, the reference epoch of the sinusoidal fitting is the mean of the available measurement epochs, and phase values are estimated accordingly.

2. The sinusoidal model is linearized by separating it into sine and cosine components. The estimated sinusoidal amplitude and phase, including their formal uncertainty, are provided below the correlations of the fitted parameters.

3. The convention for the sinusoidal phase is that the sine component is positive clockwise.

Notes on the exponential/logarithmic decay fitting:

1. The expected exponential/logarithmic decay being fitted corresponds to that commonly observed in some GNSS position time series due to visco-elastic deformation from post-seismic or surface mass loading phenomena. The decay is assumed to be asymptotic towards a more recent date and lasts for several years. The internal algorithm computes its own a priori values based on this assumption, but the user is free to provide their own a priori values when fitting a different decay.

2. The success in fitting an exponential/logarithmic decay depends heavily on the quality of the a priori parameter values (asymptotic offset and decay rate). If the user does not provide these a priori values, a built-in algorithm provides a heuristic guess. The quality of this guess depends on the amount of data available shortly after the start of the decay and whether the series contains other components that may alter the decay (discontinuities, trends, periodic variations, noise). Data gaps can be particularly problematic for accurately guessing the a priori values. If a data gap occurs at the beginning of the decay, masking the epoch of start, it is recommended to set the reference epoch of the decay right before the first available observation and estimate a position offset at this epoch.

3. For each decay being fitted in the series, if any of the two a priori parameters (asymptotic offset and decay rate) is empty or deleted by the user, the built-in algorithm will kick off and update both values with its own guess.

4. If more than one model component is occurring simultaneously (e.g., a position offset with an exponential decay or an exponential decay with a logarithmic decay), the quality of the guessed a priori parameter values may be poor because they are obtained independently for each component.

5. If the a priori parameter values are not accurate enough, the fitting may provide unsatisfactory results or even fail. In this case, the a priori parameter values can be manually tuned on the left panel using an educated or an eyeball guess. For instance, the exponential/logarithmic asymptotic offset corresponds to the observed difference between the series value at the start of the decay and the asymptotic value (the asymptotic offset is positive if the series is decreasing for the exponential and increasing for the logarithmic). If the decay complies with the expected behavior in GNSS (see note 1), the decay rate should be positive. For the exponential, it is roughly the time taken to reduce the asymptotic offset to one-third of its value; for the logarithmic, the decay rate can be challenging to guess accurately.

6. If more than one decay (exponential, logarithmic or mixed) is being fitted, and the a priori values for one of them are changed (improved), it may cause the fitting to fail. This is because the a priori values left unchanged for the other decay may now be worse than before, i.e., with respect to what remains to be fitted.

Notes on the discontinuity detection:

1. The success of the automatic detector improves when any systematic variation in the series is removed beforehand, such as trends, periodics, decays, outliers. Therefore, the detector can only be applied to residual series. The user can even remove some discontinuities before running the detector or run it several times as it may detect new discontinuities when the residual series flattens.

2. The automatic detector has difficulties in detecting discontinuities near the beginning and end of the series. To minimize this limitation, the series is automatically padded with random white noise at both ends (only for the automatic detection run). However, this may sometimes provide false positives extremely close to the beginning or end of the series due to an unwanted change in the local mean value within the padded values. The user should be suspicious of these positives unless they are really evident.

3. The user can set the minimum length of the segments considered in the stability test, which is 10% of the series length by default. However, it must be large enough to include at least three observations. The length of the segments will impact the number of discontinuities found and the processing time, which increases inversely proportional to the segment length. The processing time also depends on the number of observations.

Therefore, do not hesitate to `reduce the sampling` as necessary (see the Input data and format block). Additionally, discontinuities are often easier to find in downsampled (filtered) series.

4. The epochs of the detected discontinuities (if any) are displayed on the left panel. The user can copy and paste the epochs into the model components to remove them. Note that in case the series are auto-correlated, the detected discontinuities could be caused by noise fluctuations (see the notes on the *offset verification* below).

5. Using 75 synthetic series from the RENAG "DOGEX" experiment (from Stephane Mazzotti), the automatic detector performed as good as my own eyes for low noise series (horizontal components). However, manual detection was still much better for noisy series (vertical component).

Notes on the offset verification:

1. Spurious discontinuities of unknown origin (i.e., no recorded GNSS equipment changes or earthquakes) may appear in series containing time-correlated noise. The corresponding estimated offsets may be considered statistically significant if the series is assumed uncorrelated. This happens in the LS fitting and when using the *automatic discontinuity detection*.

2. To cope with this situation, the user has the option to verify the significance of the estimated offsets against time correlation in the series provided by a power-law noise process.

3. The parameters of the power-law noise process must be provided from a priori knowledge or estimated with the *Noise analysis* option (see the Additional fit block). In case a noise analysis was already run, the estimated noise values will be included automatically in the offset verification.

4. In the offset verification run, a MLE fit will estimate the (log-)likelihood of the residual series (offsets removed) being produced by the provided noise model and compare it to the corresponding (log-)likelihood of the series with each one of the estimated offsets restored back individually. The higher the likelihood difference, the greater the chances a particular offset is likely NOT being produced by random noise.

5. From my own testing with common power-law noise found in weekly GNSS position time series, the log-likelihood difference should be higher than 6 (at 68% CI) or 9 (at 95% CI) in order to rule out a false positive. Otherwise, there are chances that the estimated offsets may be generated by random noise instead of being systematic changes. Note that these values depend on the actual noise of the series being analyzed and their sampling.

Go to top of document

## V. Additional fit

This block allows for additional time series fitting, including:

- An automated trend estimator using the Median Interannual Difference Adjusted for Skewness (MIDAS) algorithm (Blewitt et al. 2016). Depending on the linearity of the time series, this algorithm provides reasonable and robust linear trend estimates in the presence of position discontinuities in the series.

- The histogram of the residuals with its expected normal distribution (red curve) and a stationarity assessment using the Augmented Dickey-Fuller and the Kwiatkowski-Phillips-Schmidt-Shin tests.

- A non-parametric waveform for studying periodic patterns in the residual series not having a sinusoidal shape. The waveform is plotted in red as if it were part of the fitted model (see notes below).

- The amplitude and power spectrum from the weighted Lomb-Scargle periodogram of the original data, the fitted model, the model residuals, the smoothed values, or the smoother residuals. These five spectra can be plotted and compared against each other.

- A pseudo discrete wavelet transform analysis for irregularly sampled time series described in Keitt (2008) and available for the original data, the fitted model, the model residuals, the smoothed values, or the smoother residuals. **WARNING:** this option is very time-consuming, see the notes below.

- The Vondrák (1977) band-pass smoother to reduce the variability around chosen periods from uneven, unfilled, uninterpolated, and uncertain sampled observations or evenly sampled observations with gaps and varying error bars. The smoother can be applied either to the original series or to the residual series from the model fit. It is plotted in blue in both the series and residual plots.

- A noise analysis to estimate the full variance-covariance matrix that best describes the residuals by means of an MLE of the parameters of a chosen covariance model. **WARNING:** this option is very time-consuming, see the notes below.

Notes on the MIDAS trend estimates:

1. The algorithm was implemented based on the description by Blewitt et al. (2016) and using Geoff Blewitt's code available here.

2. Only data pairs strictly separated by one year (within a tolerance of half a day) are used in the computation.

3. Contrary to the original algorithm, the estimated velocity uncertainties are not scaled.

4. Outliers removed from the series will not be considered in the selected data pairs, which is useful when having periodic outliers, for instance due to snow/ice on the GNSS antenna.

5. If the user removes discontinuities from the series, two MIDAS estimates will be provided: with and without the interannual differences crossing the discontinuities. The latter is the one that should be considered. Both estimates are provided so the user can assess the level of robustness of the algorithm when applied blindly with series having known discontinuities, but not removed, or having unknown discontinuities, i.e., not recorded in any discontinuity database.

6. The histogram of the selected interannual velocities (discontinuities skipped) is also displayed to provide a qualitative assessment of the MIDAS estimate.

Notes on the waveform:

1. The non-parametric waveform is estimated by averaging the model or smoother residual series on coincident epochs within each defined period. The series must be stationary, so this feature can only be applied to residual series (assuming they are near stationarity). For unevenly sampled series, the waveform is approximated by averaging almost coincident epochs into small bins around typical epochs. This is neccesary when the time unit is years and the series sampling is in days, as commonly found in GNSS series. In this case, non-coincident observation epochs are unavoidable due to the fact that the Earth's orbital period is not a multiple of the Earth's rotation period. Note that epochs or epoch bins containing few averaged values may provide a biased value of the waveform for that epoch, for instance, with extremely uneven sampling.

2. Contrary to the sinusoidal or wavelet fit, the non-parametric waveform can take any shape. On the other hand, it will not be defined by an analytical expression.

3. By default, the estimated waveform will be added to the fitted model or smoothed series and subtracted from the residual series for visualization purposes, without actually modifying the original series when saving the results. The estimated waveform series will be saved in a separated column of the downloaded file.

4. The waveform is estimated from the residual series and, by default, it will not affect the fitted model parameters unless the `remove from series` option is activated. In this case, the estimated waveform will be extracted from the original series before the model fit, affecting the estimated model parameters. This modification of the original series can be undone by deactivating the `remove from series` option. Note that any modification of the model components or the original/residual series (i.e., outliers) will imply deactivating the `remove from series` option automatically.

Notes on the periodogram:

1. The amplitude spectrum is useful for quickly assessing the amplitude of the different sinusoidal peaks that may compose the series. The power spectrum is similar, but in squared units, and it is useful for quickly assessing the auto-correlation of the series.

2. The *oversampling period* and both the *max/min periods* to be estimated can be set by the user. The period units are the same as the series time units. The frequency resolution of the periodogram is set as the inverse of the maximum period divided by the oversampling value. By default, there is no oversampling applied. The larger the number of periods to estimate, the longer it will take to estimate the periodogram. At this moment, the periodogram is computed in the background with no indication of its progress to the user. Generally, it should be computed reasonably fast unless the user asked for a very large number of periods, for instance, from long series with high sampling.

3. With no input from the user, the *maximum period* and a reasonable *minimum period* will be provided. The minimum period is based on the assumption that observations are already integrated over a constant observing period given by the most frequent time spacing in the series. This pseudo-Nyquist period should be fine for regular sampling with gaps, as in a typical GNSS position series.

4. In case the series are made up of uneven instantaneous observations (or that could be considered instantaneous) rather than being integrated over an observing period, the actual Nyquist period exists even for extremely uneven series and can be much shorter than the typical/average sampling of the series, as proposed by Eyer and Bartholdi (1999). It is the responsibility of the user to set the minimum period that would allow recovering the maximum of frequency information from the series.

5. The user can zoom in on an area of the periodogram (see details in the Interactive operation section) and increase the *oversampling* to get a better resolution of the amplitude/variance distribution around specific periods. In this case, the periodogram will be recomputed for the selected area only, which saves processing time by reducing the number of periods to compute. The *maximum* and *minimum* periods of the new periodogram will change on the left panel. To compute the full periodogram again, the user must delete one of the *maximum* or *minimum* periods, and the full range will be used again. It is recommended to set the *oversampling* back to 1 (default) before computing the full periodogram again.

6. The implemented algorithm forces the integral of the power spectrum to equal the total variance of the observed series. That is, this is a different normalization compared to the standard Lomb-Scargle periodogram. This allows for a more intuitive and direct comparison of the power spectrum of the different series (e.g., original vs filtered).

7. In both the amplitude and power spectra, if the user clicks on the periodogram, the coordinates (period & amplitude/variance) of the clicked point will be shown under the plot. In addition, if no oversampling is applied (unity value), the scatter (standard deviation) of the series integrated up to the clicked period will also be shown under the plot. For instance, if the user clicks at the longest period of the series or beyond, and the periodogram includes all possible periods, the computed value will nearly correspond to the standard deviation of the series.

8. In case the power spectrum is selected, a pink dashed line will be plotted, representing the slope of a pure flicker noise process commonly seen in residual GNSS series. It will also show the slope of the actual power spectrum from the rightmost selected series amongst the five options (original, model, residuals, etc.) with its respective color. These lines should help the user to roughly assess the level of time correlation of the series. However, if the user runs a noise analysis, the power spectrum of the estimated noise model will be plotted instead of the aforementioned line. This will help the user to assess the quality of the estimated noise model and to assess the crossover period between the power-law noise and white noise. At this moment, the noise power spectrum that is shown is only available for WN + FN/RW/PL models with a single crossover period (i.e., not for a WN+FN+RW noise model with two crossover periods).

Notes on the wavelet transform analysis:

1. The wavelet transform is based on the inner product of the selected series and the scaled and translated complex-valued Morlet wavelet with a central frequency equal to $2\pi$ radians.

2. The user needs to set the *max/min periods*, the *period resolution*, and the *temporal resolution* (all in the same time units of the series). The max/min periods and the highest period resolution will be proposed when the user activates the wavelet option. These are based on the equivalent periodogram limits. A reasonable temporal resolution will also be initially provided, roughly corresponding to 500 epochs regularly distributed along the

series length. Note that the minimum period and the temporal resolution should be set consistently in order to get a correct decomposition of the signal at high frequency.

3. The plotted gray dashed line represents the cone of influence of the transform, i.e., for long periods outside this line, edge effects are biasing the transform. The solid black lines represent the $1\sigma$, $2\sigma$ and $3\sigma$ values of amplitude (assuming they are normally distributed). The solid black asterisk represents the location of the maximum amplitude.

4. Compared to the periodogram or the KF, the maximum period estimated by the wavelet transform corresponds to half the observed series. In addition, even if results are obtained from unevenly spaced series, data gaps will still seriously affect the wavelet transform. Also, the estimated amplitude is not entirely exact when compared to a true sinusoid; results are provided with an accuracy roughly better than 5% (estimated empirically, further testing to be done).

5. Computing the wavelet decomposition for the full spectrum at full resolution can be horribly time-consuming. A warning will be shown on the screen with a rough estimate of the time it may take. If the expected computation time is too long, it is recommended to start with a lower temporal and/or spectral resolution to get a generalized idea of the heat map. Then the user can iteratively improve the resolution as needed and at the same time reducing the max/min periods to focus on areas of interest in order to save time.

6. The time needed to compute the wavelet transform is commensurate with the amount of server memory used. Since the online server memory is limited and shared among simultaneous users (see details in the Known issues section), large wavelets may not finish, or even the server could kill the user's session, if there is not enough free memory. Unfortunately, this happens without any warning.

Notes on the Vondrák smoother:

1. The smoother acts as a band-pass filter and accepts two input periods: a *low-pass* and a *high-pass* cut-off. To smooth the variability between these two periods, input the cut-off period values as *low pass > high pass*. On the other hand, to smooth everything outside a selected band, swap the cut-off values and set *highpass > lowpass*.

2. If the *low-pass* or the *high-pass* period is left blank, the smoother will react as a high-pass or a low-pass filter, respectively.

3. The cutoff period of the smoother is only approximate and depends on the number of points in the series. An empirical scaling has been applied to the algorithm, but thre may still be a bias between the user input cut-off period and the smoother cut-off response. Fortunately, this bias is constant for each type of series. Therefore, the bias, if any, can be assessed and taken into account by the user by looking at the intersection between the periodograms of the filter series and the filter residual series.

4. Since it is based on cubic splines, the quality of the Vondrák smoother may degrade very close to the data limits at the start and end of series, but also near long data gaps.

5. If the band to be smoothed is narrow, a significant amount of signal will remain in the series. This is due to the relatively slow "spectral response" of the smoother, which is slightly worse than a Hanning window (Sylvain Loyer, personal communication).

6. The effect of filtering the series inside/outside the specified band can also be assessed by plotting the periodogram of the filter and the filter residuals, or by using the wavelet transform.

7. If the series are fitted by an LS or KF model, the residual plot will show the differences between the smoothed series and the fitted model in blue. If there is no LS or KF model, the residual plot will show the smoother residuals.

Notes on the noise analysis:

1. The covariance model is created from four proposed stochastic processes: white noise, flicker noise, random walk noise and power-law noise.

2. The white noise and the random walk covariance matrices are obtained as described in Williams (2003), i.e. identity for white noise and linearly increasing variance for random walk noise. For flicker noise, the covariance matrix is obtained using the fast Toeplitz solver as described in Bos et al. (2013). In this algorithm, flicker noise is assumed to be some sort of natural phenomenon that already exists in at least one of the processes being observed long before the observations themselves started. This assumption makes it possible to approximate the flicker noise by a stationary covariance matrix.

3. The power-law is a generalization of the other three processes, and thus it is not possible to estimate power-law with flicker or random walk, but it is compatible with white noise, though.

4. For each noise parameter to be estimated (noise standard deviation and spectral index), the user needs to provide the *max/min* parameter values for the optimization. Otherwise, a rough guess will be provided.

5. The estimated standard deviation (in the same units as the input series) for each noise process, plus the spectral index of the power-law if selected, are provided on the left panel. These parameters can be used to assess the statistical significance of the estimated offsets (see the Fit controls block).

6. The uncertainty of the estimated noise model parameters is obtained from an approximation of the Hessian of the likelihood. A semi-definite (flat) or indefinite (sloping) Hessian may occur when the model is over-parametrized, the series does not follow a Gaussian distribution (e.g., outliers), or when a local optimum is found due to the limits imposed to the model parameters by the user. If this happens, the uncertainty will be set to *NA*, and the results may be misleading as they may lie at the border of the allowed parameter space.

7. The LS model fit is not updated with the estimated covariance matrix. However, if the fitted model includes a linear trend, an approximation of its formal uncertainty is provided by taking into account the estimated power-law noise. This is done using the general formula for the uncertainty in the rate for a power-law noise from Williams (2003). The estimated rate uncertainty from the noise analysis is then included into the LS rate uncertainty. The ratio of rate uncertainty (colored vs. white estimates) will also be displayed. Note that the white noise uncertainty used in this ratio does not necessarily correspond to the LS rate uncertainty, which includes the effect of the fitted model and especially the offsets. In other words, if the fitted model has many offsets, the noise analysis will likely not contribute significantly to the rate uncertainty.

8. The *noise analysis* implemented here is not as time-efficient as in other existing tools based on compiled languages running on big-memory machines. The processing time grows considerably for long series with high sampling and complex noise models. Relatively long computations may be possible, but the interactive benefit is lost. For instance, a WN+PL fit to a six-year-long daily series will last around 9 min. Therefore, in order to save processing time, it is highly recommended to reduce the sampling of the series to the minimum necessary. For instance, the periodogram can be used to assess the period from which the power-law noise is expected and then adjust the sampling of the series to twice that period (the white noise will be mostly averaged out). For GNSS position time series, a weekly sampling is a reasonable choice. In this case, even a WN+PL fit for a 20-year-long series will take less than 2 min to complete. Additionally, the optimization may run a bit slower if very wide parameter limits were used.

---

Go to top of document

## Interactive operation

The top navigation bar contains the links to this *help page* and for changing between the different coordinate components (unless a 1D series is used). It also contains two links at the right edge to `print` this help page into a pdf file and to `save` the results of the analysis. These two links will not be visible unless they are needed. To get back to the main program interface from this help page, click on any of the coordinate component tabs.

SARI processes the time series of each coordinate component independently. However, many options and features activated while analyzing the series in one of the tab/components will still be active when changing the tab/component (if using a 3D series) or the column number (if using a 1D series). This implies that when changing the series component, the visualization of the new series will not be available until all the selected features are computed. This

allows fitting the same model for different components of the series or different series in the same input file, although it may also slow down the initial visualization of the series. It is recommended to deactivate the options that are not needed to have a faster and smoother transition between coordinate components. Also, bear in mind that a model that has been adjusted in one series may fail for another (for instance, if fitting a logarithmic/exponential decay). Since the Kalman filter is run only when clicking on the `run KF` button, the KF fit will be removed when changing the tab.

One click on any plot will provide the coordinates of the clicked point in a space right under the plots. This is useful to obtain the series values, pinpoint discontinuity epochs, or periodic lines in the spectrum.

One click and drag will select an area on any plot. Once the area is selected, its dimensions and location can be modified. One click outside the area will remove the selection. A double click inside the area of any plot will zoom in on the selection. A double click on the same plot with no selected area will zoom out (if possible) to the original view.

Points inside the red selected area on the series plot (top) or the blue selected area on the model/filter residuals plot (bottom) can be excluded using the `toggle points` button. This is useful for removing outliers or for limiting the series to a period of interest. The removed points will turn into solid red on the series plot as a remainder, but they will disappear on the model/filter residual plot. If the series has more than one dimension, the removed points for each dimension will be kept when changing tabs (see `all components` below).
Any point toggled twice on the series plot (top) will be restored as a valid point again. All *toggled* points in the tab can be restored at once with the `reset toggle` button. If the `reset toggle` button is not active, it means there are no points to be restored.

When points are removed from the series after fitting a KF, the fit results will still be available for the remaining points, but these results will not represent the actual series after the points were removed. This is because the KF is updated only when the `run KF` button is clicked (see the Fit controls block). A warning asking to run the KF again will be shown on the screen.
For the same reason, when previously removed points are included back again into the series (restored) after fitting a KF, since there is no solution estimated for these points, the full KF results will be deleted and the user will have to run the KF again.

In addition to using the selected areas, large residuals can be automatically excluded using the `auto toggle` button. Large is defined by providing the absolute *residual threshold* or the absolute *normalized residual*, i.e., the residual divided by its own error bar if available.

The `reset` button will erase all the plots and all the parameters from the server memory, except the pointer to the *station.info* and the *custom offset* files, which will remain ready to be used again (even if they are not shown to be uploaded, the `plot` and `list` options will still be available).
After finishing a series and before uploading a new one, it is mandatory to `reset` all the processing.

Some features require intensive and time-consuming processing (*noise analysis*, *automatic offset detection*, *wavelet*). In order to save server resources and make the app always accessible to anyone, after 15 min without user interaction, the server will kill the connection, go to sleep, and may even dream of electric sheep!

---

Go to top of document

## Example of use

This is how I usually estimate the linear trend in a NEU/ENU GNSS time series (the steps may change depending on the series and application):

1) Upload the series file from your computer using `browse file`, or from a web server via the URL, and set the series format and time units.

2) `Plot` the series with points (default), lines or points & lines.

3) If the series has daily sampling, reduce it to weekly sampling using the `reduce sampling` option with a period of 0.01916 years (or the equivalent 7 days or 1 week, depending on the series time unit).

4) Upload a previously unfinished fit or the fit from another series for comparison with the `load SARI model` button.

5) Upload the *sitelog* and/or *station.info* and/or *soln* and/or *custom offset* files from your computer if available. The *station.info* and *custom offset* files are already loaded if you have done this and did not refresh the page.

6) Show the known equipment changes by activating `plot changes` next to the input *sitelog*, *station.info*, *soln* and/or *custom* files.

7) Fit a low-pass `smoother` (a period of 0.1 years may be enough to fit seasonals) and remove the outliers from the filter residuals manually using the `toggle points` or automatically using a residual threshold and the `auto toggle`. Remember to activate `all components` before removing any outlier if you plan to merge the results of the three components into a single file later.

8) Remove the low-pass smoother and fit a linear trend using weighted LS.

9) Remove remaining outliers from the residual plot manually using the `toggle points` or automatically using a threshold and the `auto toggle`. Activate `all components` if you plan to merge the results of the three components into a single file later.

10) Fit position discontinuities due to known equipment changes (dates are available by activating `list` changes next to the input *sitelog*, *station.info* and/or *custom* files) or due to unknown events (dates are available under the plot by zooming in and clicking where a discontinuity is needed).

11) Alternatively, if feeling lazy today, you can also try to `search discontinuities` automatically (this is very time-consuming, so do not forget to reduce the sampling to weekly).

12) Plot the `periodogram` (amplitude or power) of the model residuals and check for significant periodic lines.

13) Remove sinusoidal variations. The exact periods can be obtained by zooming in and clicking on the periodogram. Rise the oversampling while zooming in if necessary, but return to 1 before zooming out. If the periodic variations are not sinusoidal, use the generic `periodic waveform` instead of the LS sinusoidal fit. If the periodic variations are sinusoidal, but their amplitude and/or phase are changing smoothly, use a Kalman filter fit with appropriate process and measurement noise variances. Temporal changes of amplitude and frequency can be assessed using the `wavelet` analysis (this is very time-consuming, so you may need to reduce the period bounds and/or reduce the resolution of the wavelet).

14) Add logarithmic and/or exponential terms if necessary.

15) Iterate the steps 9 to 14 while zooming in and out guided by the smoother, the known equipment changes, the other coordinate components, an independent deformation model or the series of a nearby station, which can be uploaded with the *secondary series* feature (`show` or `correct` options).

16) Check the significance of the fitted model parameters (bearing in mind that formal errors may be too optimistic) and remove unnecessary parameters.

17) Run the `noise analysis` using a white + power-law noise model to get a better formal rate uncertainty (this is very time-consuming, so do not forget to reduce the sampling to weekly).

18) Use the estimated noise parameters to check the significance of the estimated offsets using the `offset verification` option and remove offsets accordingly.

19) Plot the `histogram` if statistics of the model/filter residuals or a stationarity assessment are needed.

20) Run the MIDAS estimator for trend comparison.

21) Load the parameters of a `plate model` to check if your site is located where it should be and if it is not

moving away from its plate.

22) Save the results on your computer using the `save` icon at the top right corner. The periodogram data can be saved by clicking on the `get periodogram data` link below the periodogram.

23) Iterate through the different coordinate components and save each one of them. The same component can be saved multiple times, for instance if the model was improved.

24) `Reset` before starting a new series.

## Known issues

There is a finite number of parallel connections that can be run by the online server at Shinyapps. That number depends on the user load. For standard user loads (i.e., excluding noise analysis with spectral index, automatic discontinuity search, and large wavelet analysis), up to 24 parallel users can be sustained, possibly a few more than that. However, it is not very clear how many simultaneous connections from the same IP address the Shinyapps server allows; that limit may be set somewhere between 10 and 15. If using SARI in a class with many people and connections start being refused, I suggest changing the outbound IP address (for instance, connecting through a smartphone hotspot) or using SARI from a local server (RStudio or Docker). Contact the author for further information.

For the sake of simplicity of the model equation, SARI uses a common nonlinear LS function that most of the time converges in zero or one iteration. On very rare occasions, the fit does not converge, even with very simplistic models that usually include a sinusoid. An *unable to fit the LS model. Change the model components* message is shown on the screen. To avoid this problem, changing slightly the reference epoch of one of the model parameters should be enough to reach convergence. Contact the author if you encounter this problem.

The Kalman filter/smoother is significantly slower than the least squares fit, at least for typical GNSS position series I have tested and especially if the model has many parameters. Between EKF and UKF, UKF is theoretically more robust against nonlinearity, though this may not represent a big difference for typical GNSS series. Therefore, the UKF is the preferred Kalman flavor by default, and it is the algorithm behind the measurement noise optimization (see the Fit controls block).
However, I have noticed that, for some series and depending on the model being fitted, the UKF provides negative variances at some epochs and for some of the estimated state parameters.
In my tests, the state itself looks good, so at this point, I am not sure what is wrong with the variances, but it may be related to rounding errors or to the unscented transformation of the selected sigma points.
The uncertainty of the estimated state parameters at the affected epochs will be set to *NA* in the downloaded file. If you encounter this problem, you can use the EKF implementation.
Also, note that EKF and UKF are not necessarily providing the same fit to the series.

At this moment, I have not found a way to select the output directory on the client's side, where the user downloads the processed series, from the online server's side to avoid the repetitive, and sometimes annoying, download prompt. This is related to server/client standard secure browsing. At least, some web browsers provide extensions to automatically download to a specific directory given the file extension or to avoid the unnecessary *(1)*, *(2)*, ... added to the file name if downloaded several times.
On the other hand, if running SARI on a local server (RStudio or Docker), there is a sixth block at the bottom of the left panel, hidden for remote connections, that allows setting a directory on your machine and download the results there with just a button click (no download prompt).

## Acknowledgements

I am thankful to these people that directly or indirectly contributed to improve this software:

Valérie Ballu, Paul Rebischung, Pascal Gegout, Giorgi Khazaradze, Alexandre Michel, Emilie Klein, Jean-Michel Lemoine, Guy Wöppelmann, Sara Padilla, Sorin Nistor, Massyl Ouaddour, Kevin Gobron, Juan J. Portela Fernández, Marianne Métois, Andrea Walpersdorf, Germinal Gabalda.

---

# References

Blewitt G, Kreemer C, Hammond WC, Gazeaux J (2016) MIDAS robust trend estimator for accurate GPS station velocities without step detection. J Geophys Res Solid Earth 121(3):2054-2068. doi: https://doi.org/10.1002/2015JB012552

Bos MS, Fernandes RMS, Williams SDP, Bastos L (2013) Fast error analysis of continuous GNSS observations with missing data. J Geod 87(4):351-360. doi: https://doi.org/10.1007/s00190-012-0605-0

Brent RP (1971) An algorithm with guaranteed convergence for finding a zero of a function. Comput J 14(4):422-425. doi: https://doi.org/10.1093/comjnl/14.4.422

Eyer L, Bartholdi P (1999) Variable stars: Which Nyquist frequency? Astron Astrophys Suppl Ser 135, 1–3. doi: https://doi.org/10.1051/aas:1999102

Julier SJ, Uhlmann JK (2004) Unscented filtering and nonlinear estimation. Proc IEEE 92(3):401-422. doi: https://doi.org/10.1109/JPROC.2003.823141

Keitt TH (2008) Coherent ecological dynamics induced by large-scale disturbance. Nature 454:331-334. doi: https://doi.org/10.1038/nature06935

RESIF (2017) RESIF-RENAG French National Geodetic Network. RESIF - Réseau Sismologique et géodésique Français. doi: https://doi.org//10.15778/resif.rg

Särkkä S (2008) Unscented Rauch–Tung–Striebel Smoother. IEEE Transactions on Automatic Control, 53(3):845-849. doi: https://doi.org/10.1109/TAC.2008.919531

Scargle JD (1982) Studies in astronomical time series analysis. II. Statistical aspects of spectral analysis of unevenly spaced data. Astrophys J 263:835-853. doi: https://doi.org/10.1086/160554

Vondrák J (1977) Problem of Smoothing Observational Data II. Bull Astron Inst Czechoslov 28:84

Williams SDP (2003) The effect of coloured noise on the uncertainties of rates estimated from geodetic time series. J Geod 76(9-10):483-494. doi: https://doi.org/10.1007/s00190-002-0283-4

Zeileis A, Kleiber C, Krämer W, Hornik K (2003) Testing and Dating of Structural Changes in Practice, Computational Statistics and Data Analysis, 44, 109-123. doi: https://doi.org/10.1016/S0167-9473(03)00030-6

---

# Author

This software is developed and is maintained by

**Alvaro Santamaría**
Geosciences Environnement Toulouse
Université de Toulouse, CNRS, IRD, CNES, UPS
Observatoire Midi-Pyrénées
Toulouse

For any comments, suggestions, questions, bugs, unexpected crashes or missing features, please contact Alvaro.

# Privacy policy

This software uses a well-known web analytics package for collecting the number of visitors and their geographic location.
IP anonymization is applied, which means that it does not collect personal data from visitors.