
Iterative Correction of Sensor Degradation and a Bayesian Multi-Sensor Data Fusion Method applied to TSI Data from PMO6-V Radiometers

Luka Kolar Rok Šikonja Lenart Treven

ETH Zürich

kolarl@student.ethz.ch rsikonja@student.ethz.ch trevenl@student.ethz.ch

Abstract

Observation and measurement of Sun’s activity provide key and irreplaceable insights into the Universe. Total Solar Irradiance (TSI) is among these measurements and is in our case measured by two identical instruments of type PMO6-V that take measurements at different sampling rates and thus are exposed to more or less degradation. In this paper we propose a method to obtain a single prediction about ground-truth TSI based on two differently degraded signals. We first focus on a *a posteriori* degradation correction of the two signals and present two iterative algorithms for extracting ground-truth signal. We include theoretical analysis and prove convergence to the ground-truth signal for the noise-free measurement model. Secondly, we present a general data-driven probabilistic framework based on sparse Gaussian processes for fusion of large number of measurements from arbitrary many noisy-measuring instruments. Framework naturally handles data gaps and provides a simple and powerful method for observing the signal trends on multiple timescales (long-term and short-term signal properties). We apply the proposed algorithms to level-1 time series of PMO6-V radiometers from SOHO spacecraft and evaluate their performance on a synthetic dataset.

1 Background

The aim of VIRGO (Variability of Solar Irradiance and Gravity Oscillations) experiment on the SOHO (Solar and Heliospheric Observatory) mission is to measure the total solar irradiance (TSI), the solar “constant”. TSI is a measure of the amount of electromagnetic energy received per unit of time and area on a surface oriented perpendicular to Sun’s rays at the mean distance of the Earth from the Sun. It is an important measure for studies of global radiation balance and climate models [16].

SOHO was launched on 2 December 1995 with observations starting in mid-January 1996. The irradiance measurement part of VIRGO contains two different radiometers — PMO6-V and DIARAD. This paper is concerned only with the measurement data from PMO6-V instruments, the main instrument PMO6V-A and the back-up instrument PMO6V-B. The former samples continuously at a rate of 1 measurement per minute and the latter at a rate of 1 measurement per week, which helps to keep its degradation low. A measurement corresponds to the opening of a shutter for a period of 1 minute with the last $\bar{t} = 20$ s being used for the actual averaging of the absorbed radiation in the cavity. The measurement time is the same for both PMO6V-A and PMO6V-B [9].

The raw PMO6V-A and PMO6V-B measurements are first corrected for all *a priori* known influences, such as distance and radial velocity to the Sun, and other thermal and electrical corrections. This leads to level-1 time series shown in Figure 1 [1].

The objective of this paper is to analyze and conduct *a posteriori* degradation (sensitivity change) corrections by comparing measured values of irradiance between the main and back-up instrument,

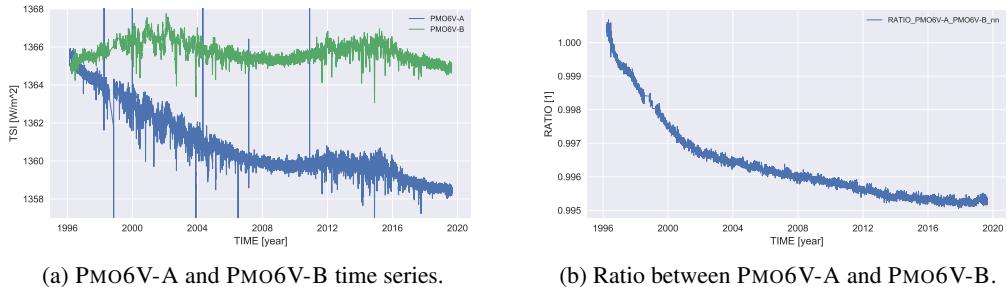


Figure 1: Level-1 PMO6V-A and PMO6V-B time series and their ratio, which illustrates the relative degradation speed of two radiometers.

and combine information from both instruments to provide a reliable TSI composite. We seek to find robust and flexible methods for degradation correction and a data fusion method which provides a probabilistic guarantee that the fused signal is close to the ground-truth signal.

Degradation of radiometers has been a subject of extensive research and several methods for its correction have been proposed [5, 8, 1]. Furthermore, a wide variety of data fusion techniques are available and well understood, and can be found in an extensive survey in [4]. More specific to the domain of TSI, authors of [6] propose a data-driven multi-scale maximum likelihood estimator approach. However, some techniques for correcting and fusing TSI time series are driven by domain knowledge and may thus have preconceived bias. We instead propose a general approach to degradation correction and explore general methods based on Gaussian processes for data fusion [15, 18].

In the initial days of the SOHO mission an increase in sensitivity is observed for both PMO6-V instruments. However, it is unfeasible to model this effect without an additional reference sensor, as proposed in [1], or introducing some model bias. We thus do not correct for this effect. Our data fusion method can naturally handle such specific anomalies by including additional measurement data from other instruments by adding probability mass near the ground-truth.

The rest of this paper is organized as follows. Section 2 describes the methods of degradation modeling and proposes two correction methods. Section 3 gives a theoretical justification for the correction methods and shows their convergence. In section 4 methods for Bayesian data fusion are presented and described. Finally, in 5 proposed methods are evaluated on synthetic and PMO6-V datasets.

1.1 Time series notation

Continuous and discrete time signals are explicitly differentiated: a parentheses notation denotes a continuous signal (e.g. $x(t)$ denotes the value of the signal x at time $t \in \mathbb{R}$) and a square bracket notation denotes a discrete time signal or time series (e.g. $x[k]$ denotes the value of the signal x at time t_k of some indexed set $T_x = \{t_1^x, \dots, t_{n_x}^x\} \subset \mathbb{R}$, where n_x denotes the length of time series x). In other words, the value of $x[k] = x(t_k)$ is available only at some time $t_k \in T_x$. Continuous time notation is used when developing the theory and discrete one for the description of algorithms.

The signal from the main instrument PMO6V-A is denoted by a and the signal from the back-up instrument PMO6V-B by b . The indexed set $T_a = \{t_1^a, \dots, t_{n_a}^a\}$ contains all sampling times of the main instrument, where n_a denotes the number of measurements, and $(a[k])_{k \in [n_a]}$ the corresponding time series of measurements. For the back-up instrument PMO6V-B notation is analogous. Let us denote the measured ground-truth signal by s . This is the actual value of total solar irradiance. Thus, the problem concerning this paper is to extract $s(t)$ from $(a[k])_{k \in [n_a]}$ and $(b[k])_{k \in [n_b]}$. Let us without the loss of generality assume that $n_a > n_b$ holds. In other words, this means that main instrument has a strictly higher sampling rate than the back-up instrument and therefore its degradation with respect to time is faster.

2 Degradation Modeling and Correction

Two assumptions are made regarding the modeling of degradation function. First assumption is that the degradation is modeled as a multiplicative effect and that it takes the value of 1 at time 0, i.e. $d(0) = 1$. This means that at the beginning we have two non-degraded instruments PMO6V-A and PMO6V-B. Secondly, degradation is assumed to be a non-increasing function. If this was not the case, then the instrument's performance would improve with time and/or exposure time, which is not in accordance with physical systems. As it turns out, first assumption is of paramount importance in the modeling of degradation, not only being intuitively true, but providing a necessity for degradation correction as we show in section 3. The second assumption improves interpretability and robustness of the proposed method.

By assumption, degradation of both instruments is modeled as a multiplicative effect on the sensitivity of both instruments. It is described by the degradation function $d(\cdot)$, which is a function of what we call the exposure or exposure “time” e . The exposure is a measure corresponding to the received amount of radiation up to a certain time during the measurement process of each instrument. In units it corresponds to the amount of energy per unit of area. We denote by $e_a(\cdot)$ and $e_b(\cdot)$ the exposure functions for the two instruments, where values $e_a(t)$ and $e_b(t)$ at time t correspond to the amount of measurements a particular instrument has made up to time t . Functions for two instruments have different values at time t , because of the different sampling rates of both instruments.

Noise-free measurement model is defined in equation (1) and used for the development of theory of degradation correction.

$$\begin{aligned} a(t) &= s(t) \cdot d(e_a(t)) \\ b(t) &= s(t) \cdot d(e_b(t)) \end{aligned} \tag{1}$$

In order to get a realistic model, which accounts for the measurement errors, we introduce normally distributed additive noise with zero mean and a constant instrument-dependent variance, σ_a^2 and σ_b^2 . We denote by ε_a and ε_b white noise signals (zero mean and independent in time) that are independent of each other. This yields noisy measurement model defined in equation (2).

$$\begin{aligned} a(t) &= s(t) \cdot d(e_a(t)) + \varepsilon_a(t), \quad \varepsilon_a(t) \sim \mathcal{N}(0, \sigma_a^2) \\ b(t) &= s(t) \cdot d(e_b(t)) + \varepsilon_b(t), \quad \varepsilon_b(t) \sim \mathcal{N}(0, \sigma_b^2) \end{aligned} \tag{2}$$

2.1 Exposure

Given a time series $(x[k])_{k \in [n_x]}$, the exposure value at t_k can be computed in two ways. Since the measurement time is the same for both instruments, NUMMEASUREMENTS method only counts the number of measurements an instrument has made up to a certain time, whereas CUMSUM is a generalization of the former that also accounts for the average amount of radiation absorbed during each measurement. The rationale for CUMSUM is that if the energy absorbed during a measurement is greater, then it should also have a greater contribution to the exposure and thus a greater effect on the degradation. The algorithm for computing values $e_x(t_k) = e_x[k]$ is described in algorithm 1. Note that by definition $e_x(0) = 0$ holds.

Algorithm 1 COMPUTEEXPOSURE($(x[k])_{k \in [n_x]}$, method)

```

1: if method = NUMMEASUREMENTS then
2:   for  $k \in [n_x]$  do
3:      $e_x[k] \leftarrow$  Count number of measurements in  $x$  up to and including time  $t_k$ 
4:   end for
5: else if method = CUMSUM then
6:   for  $k \in [n_x]$  do
7:      $e_x[k] \leftarrow \text{SUM}(x[1 : k])$ 
8:   end for
9: end if
10: return  $(e_x[k])_{k \in [n_x]}$ 

```

2.2 Degradation Modeling

The degradation function $d(\cdot)$ is the same for both instruments PMO6V-A and PMO6V-B because of their identical structure. Different rate of degradation arises from different modes of operation. Moreover, obtained $d(\cdot)$ holds only for instruments of type PMO6-V, i.e. it describes an intrinsic property of the structure and materials of a particular measuring instrument.

We follow similar approach as [9], by determining the degradation function solely from the ratio of signals $r(t) = \frac{a(t)}{b(t)}$ or more precisely from its discrete time counterpart $(r[k] = \frac{a[k]}{b[k]})_{k \in n_m}$, where $t_k \in T_m = \{t_1^m, \dots, t_{n_m}^m\} = T_a \cap T_b$. Thus, the ratio is computed only at discrete times at which both instruments took a measurement. However, our method could also be extended to the case when there are no simultaneous measurements by using interpolation methods.

Next, we propose five modeling methods (and one additional) for learning the degradation function. This can be formulated as a univariate regression problem, where we assume that there exists a function $f(\cdot, \theta)$ parametrized by θ , which describes the relation between a predictor variable x and a response variable y , taking the form of

$$y = f(x, \theta) + \varepsilon, \quad (3)$$

where ε represents random noise with zero mean.

Given a set of n observations $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^n$ with $x_i, y_i \in \mathbb{R}$, the objective is to find the value θ^* that best describes the relation between x and y , which can be in principle obtained (usually regularization term is added) by solving the following optimization problem

$$\min_{\theta} \sum_{i=1}^n (f(x_i, \theta) - y_i)^2. \quad (4)$$

2.2.1 Exponential families

Many physical phenomena can be described by the family of exponential functions, because they arise naturally as solutions to differential equations. Two such examples are capacitor charging and radioactive decay. Obtained parameters can be interpreted and provide valuable insights into the understanding of underlying physical systems.

Firstly, we propose a simple exponential function with parameters $\theta = [\theta_1, \theta_2]^T$, where $\theta_1 > 0$ is the exponential decay and θ_2 is a scaling parameter in the following form

$$f(x, \theta) = 1 - e^{\theta_1 \cdot \theta_2} + e^{-\theta_1 \cdot (x - \theta_2)}. \quad (\text{EXP})$$

Secondly, we propose an extension to EXP by introducing an additional linear dependency term, taking the following form

$$f(x, \theta) = 1 - e^{\theta_1 \cdot \theta_2} + e^{-\theta_1 \cdot (x - \theta_2)} + \theta_3 \cdot x. \quad (\text{EXPLIN})$$

The latter model was adopted from [1]. Note that both models satisfy the condition $f(0, \theta) = 1$ regardless of the value of θ . Both models are learned using the Levenberg–Marquardt iterative algorithm [13, 12].

2.2.2 Splines

Splines are an effective method for fitting each data point with the stipulation that the obtained curve is continuous and smooth [14]. What follows is summarized from [7], but we repeat the relevant parts for reader's convenience and use smoothed splines of degree 3. Given a set of knots, which is a strictly increasing sequence of real numbers, $\mathcal{K}' = \{k_4, \dots, k_{m-3}\}$, a cubic spline function f is defined on the range $[k_4, k_{m-3}]$ such that on each interval (k_j, k_{j+1}) for $j \in \{4, m-4\}$ it is given by some polynomial of degree less than or equal to 3. The function f , its first derivative f' and second derivative f'' are continuous everywhere on the interval $[k_4, k_{m-3}]$.

If we define $g(k; x) = (k - x)_+^3$, then normalized cubic B-spline s_j on $k_j, k_{j+1}, \dots, k_{j+4}$ is given as the 4-th divided difference of $g(k; x)$ multiplied by $(k_{j+4} - k_j)$ for a fixed x , this is

$$s_j(x) = (k_{j+4} - k_j) \cdot g(k_j, k_{j+1}, \dots, k_{j+4}; x). \quad (5)$$

The normalized B-spline satisfies $s_j(x) > 0$ for $k_j < x < k_{j+4}$ and $s_j(x) = 0$ elsewhere. In order to obtain a basis for the space of cubic spline functions with knots \mathcal{K}' (space is denoted by $\eta(\mathcal{K}')$), the set of knots \mathcal{K}' is extended to $\mathcal{K} = \{k_1, \dots, k_m\}$ by 6 additional knots, such that the property of strictly increasing order is preserved. Then, every spline function $f(\cdot) \in \eta(\mathcal{K}')$ has a unique representation as a linear combination of s_j ,

$$f(x) = \sum_{j=1}^{m-4} c_j \cdot s_j(x). \quad (6)$$

Given data points \mathcal{D} and a smoothing constant S , resulting spline function $f(\cdot) = f(\cdot, \boldsymbol{\theta}) = f(\cdot, (\mathcal{K}, \{c_j\}_{j=1}^{m-4}))$ is determined by the following optimization problem

$$\begin{aligned} \min_{\boldsymbol{\theta}} \sum_{j=5}^{m-4} d_j^2 &= \min_{\boldsymbol{\theta}} \sum_{j=5}^{m-4} \left(f^{(3)}(k_j + 0) - f^{(3)}(k_j - 0) \right)^2 \\ \text{s. t. } \sum_{i=1}^n (f(x_i) - y_i)^2 &\leq S \end{aligned} \quad (\text{SPLINE})$$

where d_j represents the discontinuity of third derivative of f at knot k_j . The number of knots, their positions and $\{c_j\}_{j=1}^{m-4}$ are automatically chosen by the algorithm that solves SPLINE, which is described in [7].

Additional constraint $f(0) = 1$ is added to the optimization problem SPLINE, so that the first assumption of the model is satisfied. Smoothing constant S , which determines number of knots and their positions, is chosen such that the resulting spline function has a guarantee that f is a non-increasing function and fits data best. This is done by performing a bisection on S on the interval $[0, S_{\max}]$ and is formalized in algorithm 2.

Algorithm 2 BISECTIONONS($(x[k])_{k \in [n]}, (y[k])_{k \in [n]}, [a, b]$)

```

1:  $s \leftarrow \frac{a+b}{2}$                                 ▷ Initial guess for smoothing constant  $S$ 
2:  $f \leftarrow \text{SPLINE}((x[k])_{k \in [n]}, (y[k])_{k \in [n]}, s)$     ▷ Fit spline given smoothing constant  $s$ 
3: while less than maximum number of iterations do
4:   if ISNONINCREASING( $f, (x[k])_{k \in [n]}$ ) then          ▷ Check if  $f$  is non-increasing
5:      $b \leftarrow s$                                          ▷ Update upper bound
6:   else
7:      $a \leftarrow s$                                          ▷ Update lower bound
8:   end if
9:    $s \leftarrow \frac{a+b}{2}$ 
10:   $f \leftarrow \text{SPLINE}((x[k])_{k \in [n]}, (y[k])_{k \in [n]}, s)$ 
11: end while
12: return  $f$ 

```

2.2.3 Monotonic Regression and Smoothed Monotonic Regression

One of the main modeling challenges is the robustness of proposed correction method. By assumption, degradation function has to be a monotonically decreasing function and thus monotonic regression is a natural choice for its modeling.

Monotonic or isotonic regression requires predictor values to be in strictly increasing order $x_1 < x_2 < \dots < x_n$. Its solution is a stepwise interpolating function determined by n points $\boldsymbol{\theta} = [\theta_1, \dots, \theta_n]^T$ in monotonically decreasing order, which are obtained by solving the following optimization problem with $f(x_i, \boldsymbol{\theta}) = \theta_i, \forall i \in [n]$, [17]

$$\begin{aligned} \min_{\boldsymbol{\theta}} \sum_{i=1}^n (f(x_i, \boldsymbol{\theta}) - y_i)^2 &= \min_{\boldsymbol{\theta}} \sum_{i=1}^n (\theta_i - y_i)^2 \\ \text{s. t. } \theta_i &\geq \theta_{i+1} \text{ for } i \in \{1, \dots, n-1\} \end{aligned} \quad (\text{ISOTONIC})$$

This quadratic optimization problem provides a simple and powerful framework for enforcing additional constraints, such as $f(0, \boldsymbol{\theta}) = 1$ (assume $x_1 = 0$) or convexity/concavity, which can be added to ISOTONIC as

$$\begin{aligned} \theta_1 &= 1 \quad \text{to ensure} & f(0, \boldsymbol{\theta}) &= 1 \\ \theta_{i+1} - 2 \cdot \theta_i + \theta_{i-1} &\geq 0 \text{ for } i \in \{2, \dots, n-1\} \quad \text{to ensure Convexity, } f''(x_i) \geq 0 \end{aligned} \quad (7)$$

The practical issue with monotonic regression is that it resembles a discontinuous step function while we expect the function to be continuous and smooth. In this regard, the authors in [17] propose a modification of ISOTONIC by penalizing the difference between adjacent fitted response values, θ_i and θ_{i+1} by using an L2 regularization term. This yields a smoothed monotonic regression problem formulated as

$$\begin{aligned} \min_{\boldsymbol{\theta}} \sum_{i=1}^n (\theta_i - y_i)^2 + \sum_{i=1}^{n-1} \lambda_i \cdot (\theta_{i+1} - \theta_i)^2 \\ \text{s. t. } \theta_i \geq \theta_{i+1} \text{ for } i \in \{1, \dots, n-1\}, \end{aligned} \quad (\text{SMOOTHMONOTONIC})$$

where λ_i for $i \in [n-1]$ are selected regularization parameters. Note that similarly to ISOTONIC we can introduce additional constraints, such as (7). We choose $\lambda_1 = \dots = \lambda_{n-1} = 1$. The difference in naming models, isotonic vs. monotonic, is introduced only for the purpose of clarity.

The stability of quadratic optimization problem SMOOTHMONOTONIC is guaranteed by first running ISOTONIC, then uniformly sampling obtained function $f(\cdot, \boldsymbol{\theta})$ at $\mathcal{X} = \{x_1^r, \dots, x_m^r\}$, where $n \gg m$, and lastly running SMOOTHMONOTONIC with $\mathcal{D}' = \{(x_i^r, y_i^r)\}_{i=1}^m$. Final f is obtained as a linear interpolation between each x_i^r and x_{i+1}^r .

2.2.4 Ensemble Model

Ensemble model extends previously described models by taking an arbitrary weighted linear combination of them. It consists of m models $M = \{(f_1(\cdot, \boldsymbol{\theta}_1), w_1), \dots, (f_m(\cdot, \boldsymbol{\theta}_m), w_m)\}$, where $f_i(\cdot, \boldsymbol{\theta}_i)$ is a simple model parametrized by $\boldsymbol{\theta}_i$ and a weight w_i , such that $\sum_{i=1}^m w_i = 1$ holds. The ensemble model is given by

$$f(x, \boldsymbol{\theta}) = \sum_{i=1}^m w_i \cdot f_i(\cdot, \boldsymbol{\theta}_i). \quad (\text{ENSEMBLE})$$

2.3 Degradation Correction

We now focus on eliminating degradation effect on signals a and b . This process yields their degradation-corrected versions and we denote these by a_c and b_c , respectively. Here we use the noiseless measurement model defined by equation (1), the rationale for not using (2) will be discussed later. Note that neither the ground-truth signal $s(t)$ nor the degradation function $d(\cdot)$ is known. We now present two methods with which $s(t)$ and $d(\cdot)$ can be approximately learned just from time series $(a[k])_{k \in [n_m]}$ and $(b[k])_{k \in [n_m]}$ with $t_k \in T_m$. Methods CORRECTBOTH and CORRECTONE perform iterative corrections and are formulated as follows:

$$r_n(e_a(t)) = \frac{a_n(t)}{b_n(t)} \quad a_{n+1}(t) = \frac{a_n(t)}{r_n(e_a(t))} \quad b_{n+1}(t) = \frac{b_n(t)}{r_n(e_b(t))} \quad \text{for } n = 0, 1, \dots \quad (\text{CORRECTBOTH})$$

$$r_n(e_a(t)) = \frac{a_0(t)}{b_n(t)} \quad a_{n+1}(t) = \frac{a_0(t)}{r_n(e_a(t))} \quad b_{n+1}(t) = \frac{b_0(t)}{r_n(e_b(t))} \quad \text{for } n = 0, 1, \dots \quad (\text{CORRECTONE})$$

where $a_0(t) = a(t) = s(t) \cdot d(e_a(t))$ and $b_0(t) = b(t) = s(t) \cdot d(e_b(t))$. We further present these methods as algorithms 3 and 4 to unambiguously show computational path on given time series. Algorithms return best estimates of $s(t)$ for each instrument for $t_k \in T_m$ and $d(\cdot)$. In section 3 we show that indeed, $d_c(\cdot) \rightarrow d(\cdot)$, $a_c(t) \rightarrow s(t)$ and $b_c(t) \rightarrow s(t)$ holds. After obtaining $d_c(\cdot)$ both

time series $(a[k])_{k \in [n_a]}, (b[k])_{k \in [n_b]}$ can be corrected as

$$\begin{aligned} a_c(t) &= \frac{a(t)}{d_c(e_a(t))} \text{ and } a_c[k] = \frac{a[k]}{d_c(e_a[k])} \quad \text{for } k \in [n_a], \\ b_c(t) &= \frac{b(t)}{d_c(e_b(t))} \text{ and } b_c[k] = \frac{b[k]}{d_c(e_b[k])} \quad \text{for } k \in [n_b]. \end{aligned} \tag{CORRECTION}$$

In CORRECTBOTH the ratio at every iteration is computed between the corrected signals and thus

Algorithm 3 CORRECTBOTH($(a[k])_{k \in [n_m]}, (b[k])_{k \in [n_m]}, (e_a[k])_{k \in [n_m]}, (e_b[k])_{k \in [n_m]}$)

```

1: while not converged do                                ▷ E.g.  $\|a_{i+1} - a_i\|_2 / \|a_i\|_2 + \|b_{i+1} - b_i\|_2 / \|b_i\|_2 > \varepsilon$ 
2:    $r \leftarrow \frac{a}{b}$                                      ▷ Divide signals  $a$  and  $b$  pointwise, i.e.  $r[k] = \frac{a[k]}{b[k]} \forall k \in [n_m]$ 
3:    $f(\cdot) \leftarrow \text{FITCURVETO}((e_a[k])_{k \in [n_m]}, (r[k])_{k \in [n_m]})$     ▷ Learn mapping  $f : e_a \mapsto f(e_a)$ 
4:    $a \leftarrow \frac{a}{f(e_a)}$                                ▷ Correction update of signal  $a$ 
5:    $b \leftarrow \frac{b}{f(e_b)}$                                ▷ Correction update of signal  $b$ 
6: end while
7:  $a_c \leftarrow a; b_c \leftarrow b$                          ▷ Corrected signals,  $a(t) \approx s(t)$  and  $b(t) \approx s(t)$ 
8:  $r_c \leftarrow \frac{a}{b_c}$                                      ▷ Divide signals  $a$  and  $b_c$  pointwise with  $r_c(e_a) \approx d(e_a)$ 
9:  $d_c(\cdot) \leftarrow \text{FITCURVETO}((e_a[k])_{k \in [n_m]}, (r_c[k])_{k \in [n_m]})$     ▷ Learn degradation function  $d(\cdot)$ 
10: return  $(a_c[k])_{k \in [n_m]}, (b_c[k])_{k \in [n_m]}, d_c(\cdot)$  ▷ Return corrected signals and degradation function

```

in the limit this ratio goes to 1. On the other hand, in CORRECTONE the ratio is computed between initial signal a and corrected b , therefore in the limit this ratio approximates the degradation function $d(\cdot)$. It can be argued that the latter is a variant of the former, with having signal a fixed throughout all iterations.

Algorithm 4 CORRECTONE($(a[k])_{k \in [n_m]}, (b[k])_{k \in [n_m]}, (e_a[k])_{k \in [n_m]}, (e_b[k])_{k \in [n_m]}$)

```

1:  $a_c \leftarrow a; b_c \leftarrow b$                          ▷ Initial estimate of corrected signals
2: while not converged do                                ▷ E.g.  $\|a_{i+1} - a_i\|_2 / \|a_i\|_2 + \|b_{i+1} - b_i\|_2 / \|b_i\|_2 > \varepsilon$ 
3:    $r \leftarrow \frac{a}{b_c}$                                ▷ Divide signals  $a$  and  $b_c$  pointwise, i.e.  $r[k] = \frac{a[k]}{b_c[k]} \forall k \in [n_m]$ 
4:    $f(\cdot) \leftarrow \text{FITCURVETO}((e_a[k])_{k \in [n_m]}, (r[k])_{k \in [n_m]})$     ▷ Learn mapping  $f : e_a \mapsto f(e_a)$ 
5:    $a_c \leftarrow \frac{a}{f(e_a)}$                            ▷ Correction update of signal  $a$ 
6:    $b_c \leftarrow \frac{b}{f(e_b)}$                            ▷ Correction update of signal  $b$ 
7: end while
8:  $d_c(\cdot) \leftarrow f(\cdot)$                              ▷ Final estimate of degradation function  $d(\cdot)$ 
9: return  $(a_c[k])_{k \in [n_m]}, (b_c[k])_{k \in [n_m]}, d_c(\cdot)$  ▷ Return corrected signals and degradation function

```

3 Convergence theorems

This section provides some theoretical guarantees for the convergence of our methods for the noise-free measurement model (1). We show this first for simple exposure functions $e_a(t) = t$ and $e_b(t) = \frac{t}{2}$, and then generalize it to arbitrary exposure functions.

Proposition 1 (CORRECTBOTH for $e_a(t) = t$ and $e_b(t) = \frac{t}{2}$). *Let $a_0(t) = s(t) \cdot d(t)$ and $b_0(t) = s(t) \cdot d(\frac{t}{2})$ for $t \geq 0$, where $s(t) > 0$ is the ground-truth signal and $d : \mathbb{R}_{\geq 0} \rightarrow [0, 1]$ is a continuous degradation function with $d(0) = 1$. If we run algorithm for $n = 0, 1, \dots$:*

$$r_n(t) = \frac{a_n(t)}{b_n(t)}, \quad a_{n+1}(t) = \frac{a_n(t)}{r_n(t)}, \quad b_{n+1}(t) = \frac{b_n(t)}{r_n(\frac{t}{2})},$$

then it holds $\forall t \geq 0 : \lim_{n \rightarrow \infty} a_n(t) = \lim_{n \rightarrow \infty} b_n(t) = s(t)$.

Proof. Let us fix an arbitrary $t > 0$. Then we observe

$$r_n(t) = \frac{a_n(t)}{b_n(t)} = \frac{a_{n-1}(t)}{r_{n-1}(t)} \frac{r_{n-1}(\frac{t}{2})}{b_{n-1}(t)} = r_{n-1}\left(\frac{t}{2}\right) = \cdots = r_0\left(\frac{t}{2^n}\right),$$

where the last equality follows by induction. Now, let us focus on the sequence $b_n(t)$:

$$\begin{aligned} b_n(t) &= \frac{b_{n-1}(t)}{r_{n-1}(\frac{t}{2})} = \frac{b_{n-1}(t)}{r_0(\frac{t}{2^n})} \\ &= \frac{b_{n-2}(t)}{r_0(\frac{t}{2^n}) \cdot r_{n-2}(\frac{t}{2})} = \frac{b_{n-2}(t)}{r_0(\frac{t}{2^n}) \cdot r_0(\frac{t}{2^{n-1}})} \\ &= \frac{b_0(t)}{\prod_{i=1}^n r_0(\frac{t}{2^i})}. \end{aligned} \tag{8}$$

Calculation of ratio $r_0(t)$ gives us

$$r_0(t) = \frac{a_0(t)}{b_0(t)} = \frac{s(t) \cdot d(t)}{s(t) \cdot d(\frac{t}{2})} = \frac{d(t)}{d(\frac{t}{2})}.$$

Inserting the latter observation and $b_0(t) = s(t) \cdot d(\frac{t}{2})$ into equation (8) yields

$$b_n(t) = \frac{s(t) \cdot d(\frac{t}{2})}{\prod_{i=1}^n \frac{d(\frac{t}{2^i})}{d(\frac{t}{2^{i+1}})}} = s(t) \cdot d\left(\frac{t}{2}\right) \frac{\prod_{i=1}^n d(\frac{t}{2^{i+1}})}{\prod_{i=1}^n d(\frac{t}{2^i})} = s(t) \cdot \frac{\prod_{i=1}^{n+1} d(\frac{t}{2^i})}{\prod_{i=1}^n d(\frac{t}{2^i})} = s(t) \cdot d\left(\frac{t}{2^{n+1}}\right).$$

The proof is established by sending $n \rightarrow \infty$, which gives

$$\lim_{n \rightarrow \infty} b_n(t) = \lim_{n \rightarrow \infty} s(t) \cdot d\left(\frac{t}{2^{n+1}}\right) = s(t) \cdot d\left(\frac{t}{\lim_{n \rightarrow \infty} 2^{n+1}}\right) = s(t) \cdot d(0) = s(t).$$

□

Remark 1. The fact that $\lim_{n \rightarrow \infty} a_n(t) = \lim_{n \rightarrow \infty} b_n(t)$ holds is easily obtained from

$$a_{n+1}(t) = \frac{a_n(t)}{r_n(t)} = \frac{a_n(t) \cdot b_n(t)}{a_n(t)} = b_n(t).$$

Next, let us generalize the obtained result to an arbitrary exposure function. Let us first state a straightforward corollary, the proof of which immediately follows from the proposition 1.

Corollary 1 (CORRECTBOTH for $e_a(t) = t$ and $e_b(t) = \frac{t}{k}$, $k > 1$). Let $a_0(t) = s(t) \cdot d(t)$ and $b_0(t) = s(t) \cdot d(\frac{t}{k})$ for $t \geq 0$, where $s(t) > 0$ is the ground-truth signal, $d : \mathbb{R}_{\geq 0} \rightarrow [0, 1]$ is a continuous degradation function with $d(0) = 1$ and $k > 1$ is an arbitrary sampling rate parameter. If we run algorithm for $n = 0, 1, \dots$:

$$r_n(t) = \frac{a_n(t)}{b_n(t)}, \quad a_{n+1}(t) = \frac{a_n(t)}{r_n(t)}, \quad b_{n+1}(t) = \frac{b_n(t)}{r_n(\frac{t}{k})},$$

then it holds $\forall t \geq 0 : \lim_{n \rightarrow \infty} a_n(t) = \lim_{n \rightarrow \infty} b_n(t) = s(t)$.

Proposition 2 (CORRECTBOTH for $e_a(t) = t$ and $e_b(t) = e(t)$, $e(t) < t$). Let $a_0(t) = s(t) \cdot d(t)$ and $b_0(t) = s(t) \cdot d(e(t))$ for $t \geq 0$, where $s(t) > 0$ is the ground-truth signal, $d : \mathbb{R}_{\geq 0} \rightarrow [0, 1]$ is a continuous degradation function with $d(0) = 1$ and $e(t)$ is the exposure function of signal b , for which it holds $e(0) = 0$ and $e(t) < t$ for all $t > 0$. If we run algorithm for $n = 0, 1, \dots$:

$$r_n(t) = \frac{a_n(t)}{b_n(t)}, \quad a_{n+1}(t) = \frac{a_n(t)}{r_n(t)}, \quad b_{n+1}(t) = \frac{b_n(t)}{r_n(e(t))},$$

then it holds $\forall t \geq 0 : \lim_{n \rightarrow \infty} a_n(t) = \lim_{n \rightarrow \infty} b_n(t) = s(t)$.

Proof. Let us fix an arbitrary $t > 0$ and compute $b_1(t)$, which gives

$$b_1(t) = \frac{b_0(t)}{r_0(e(t))} = s(t) \cdot d(e(t)) \cdot \frac{d(e(e(t)))}{d(e(t))} = s(t) \cdot d(e(e(t))).$$

Let $e^n(t) = \underbrace{(e \circ e \circ \cdots \circ e)}_{n \text{ times}}(t)$, then by induction we have

$$b_n(t) = s(t) \cdot d(e^{n+1}(t)).$$

By taking the limit $n \rightarrow \infty$, we get

$$\lim_{n \rightarrow \infty} b_n(t) = \lim_{n \rightarrow \infty} s(t) \cdot d(e^{n+1}(t)) = s(t) \cdot d\left(\lim_{n \rightarrow \infty} e^{n+1}(t)\right).$$

Let $\eta = \lim_{n \rightarrow \infty} e^{n+1}(t)$, then we have

$$\eta = \lim_{n \rightarrow \infty} e^{n+1}(t) = e\left(\lim_{n \rightarrow \infty} e^n(t)\right) = e(\eta)$$

but since $e(\eta) < \eta$ for $\eta > 0$ and $e(0) = 0$ we obtain $\eta = 0$. Hence we conclude this proof as

$$\lim_{n \rightarrow \infty} b_n(t) = s(t) \cdot d(0) = s(t).$$

□

Theorem 1. Let $a_0(t) = s(t) \cdot d(e_a(t))$ and $b_0(t) = s(t) \cdot d(e_b(t))$ for $t \geq 0$, where $s(t) > 0$ is the ground-truth signal, $d : \mathbb{R}_{\geq 0} \rightarrow [0, 1]$ is a continuous degradation function with $d(0) = 1$ and $e_a(t), e_b(t) : [0, \infty) \rightarrow [0, \infty)$ are the continuous exposure function of signal a and b respectively. Let us further assume $e_a(0) = e_b(0) = 0$, $e_b(t) < e_a(t)$ for all $t > 0$ and that there exist function $e_a^{-1} : [0, \infty) \rightarrow [0, \infty)$. If we run algorithm for $n = 0, 1, \dots$:

$$r_n(t) = \frac{a_n(t)}{b_n(t)}, \quad a_{n+1}(t) = \frac{a_n(t)}{r_n(t)}, \quad b_{n+1}(t) = \frac{b_n(t)}{r_n((e_a^{-1} \circ e_b)(t))},$$

then it holds $\forall t \geq 0 : \lim_{n \rightarrow \infty} a_n(t) = \lim_{n \rightarrow \infty} b_n(t) = s(t)$.

Proof. Let $h(t) = d(e_a(t))$. Then it holds $d(e_b(t)) = h(e_a^{-1} \circ e_b)(t)$. If we denote $e = e_a^{-1} \circ e_b$, then the proposed algorithm transforms to:

$$r_n(t) = \frac{a_n(t)}{b_n(t)}, \quad a_{n+1}(t) = \frac{a_n(t)}{r_n(t)}, \quad b_{n+1}(t) = \frac{b_n(t)}{r_n(e(t))},$$

with the initial setting: $a_0(t) = s(t) \cdot h(t)$ and $b_0(t) = s(t) \cdot h(e(t))$. Since $e_b(t) < e_a(t)$ holds, $e(t) < t$ holds as well, and since $e_a(0) = e_b(0) = 0$ holds, $e(0) = 0$ and $h(0) = d(e_a(0)) = d(0) = 1$ holds as well. Since the assumptions from proposition 2 are satisfied, we are done. □

Proposition 3 (CORRECTONE for $e_a(t) = t$ and $e_b(t) = \frac{t}{2}$). Let $a_0(t) = s(t) \cdot d(t)$ and $b_0(t) = s(t) \cdot d(\frac{t}{2})$ for $t \geq 0$, where $s(t) > 0$ is the ground-truth signal and $d : \mathbb{R}_{\geq 0} \rightarrow [0, 1]$ is a continuous degradation function with $d(0) = 1$. If we run algorithm for $n = 0, 1, \dots$:

$$r_n(t) = \frac{a_0(t)}{b_n(t)}, \quad a_{n+1}(t) = \frac{a_0(t)}{r_n(t)}, \quad b_{n+1}(t) = \frac{b_0(t)}{r_n(\frac{t}{2})},$$

then it holds $\forall t \geq 0 : \lim_{n \rightarrow \infty} a_n(t) = \lim_{n \rightarrow \infty} b_n(t) = s(t)$ and $\lim_{n \rightarrow \infty} r_n(t) = d(t)$.

Proof. Let us fix an arbitrary $t > 0$. Since $r_0(t) = \frac{d(t)}{d(\frac{t}{2})}$ holds, then by induction we have

$$\begin{aligned} r_n(t) &= \frac{a_0(t)}{b_n(t)} = \frac{a_0(t)}{\frac{b_0(t)}{r_{n-1}(\frac{t}{2})}} = \frac{a_0(t)}{b_0(t)} \cdot r_{n-1}\left(\frac{t}{2}\right) = \frac{a_0(t)}{b_0(t)} \cdot \frac{a_0(\frac{t}{2})}{b_0(\frac{t}{2})} \cdot r_{n-2}\left(\frac{t}{2^2}\right) \\ &= \dots = \prod_{i=0}^{n-1} \frac{a_0(\frac{t}{2^i})}{b_0(\frac{t}{2^i})} \cdot r_0\left(\frac{t}{2^n}\right) = \prod_{i=0}^n \frac{d(\frac{t}{2^i})}{d(\frac{t}{2^{i+1}})} = \frac{d(t)}{d(\frac{t}{2^{n+1}})}. \end{aligned}$$

Calculating $b_n(t)$ yields

$$b_n(t) = \frac{b_0(t)}{r_{n-1}(\frac{t}{2})} = \frac{s(t) \cdot d(\frac{t}{2})}{\frac{d(\frac{t}{2})}{d(\frac{t}{2^{n+1}})}} = s(t) \cdot d\left(\frac{t}{2^{n+1}}\right)$$

and similarly for $a_n(t)$, we have

$$a_n(t) = s(t) \cdot d\left(\frac{t}{2^n}\right).$$

By taking the limit $n \rightarrow \infty$, we get

$$\lim_{n \rightarrow \infty} b_n(t) = \lim_{n \rightarrow \infty} s(t) \cdot d\left(\frac{t}{2^{n+1}}\right) = s(t) \cdot d\left(\frac{t}{\lim_{n \rightarrow \infty} 2^{n+1}}\right) = s(t) \cdot d(0) = s(t),$$

from where it immediately follows $\lim_{n \rightarrow \infty} a_n(t) = s(t)$. Moreover, the ratio $r_n(t)$ converges to $d(t)$, which follows from continuity of d as

$$\lim_{n \rightarrow \infty} r_n(t) = \lim_{n \rightarrow \infty} \frac{d(t)}{d\left(\frac{t}{2^{n+1}}\right)} = \frac{d(t)}{\lim_{n \rightarrow \infty} d\left(\frac{t}{2^{n+1}}\right)} = \frac{d(t)}{d\left(\frac{t}{\lim_{n \rightarrow \infty} 2^{n+1}}\right)} = \frac{d(t)}{d(0)} = d(t).$$

□

As for CORRECTBOTH we have a direct corollary for an arbitrary $k > 1$.

Corollary 2 (CORRECTONE for $e_a(t) = t$ and $e_b(t) = \frac{t}{k}$, $k > 1$). *Let $a_0(t) = s(t) \cdot d(t)$ and $b_0(t) = s(t) \cdot d(\frac{t}{k})$ for $t \geq 0$, where $s(t) > 0$ is the ground-truth signal, $d : \mathbb{R}_{\geq 0} \rightarrow [0, 1]$ is a continuous degradation function with $d(0) = 1$ and $k > 1$ is an arbitrary sampling rate parameter. If we run algorithm for $n = 0, 1, \dots$:*

$$r_n(t) = \frac{a_0(t)}{b_n(t)}, \quad a_{n+1}(t) = \frac{a_0(t)}{r_n(t)}, \quad b_{n+1}(t) = \frac{b_0(t)}{r_n(\frac{t}{k})},$$

then it holds $\forall t \geq 0 : \lim_{n \rightarrow \infty} a_n(t) = \lim_{n \rightarrow \infty} b_n(t) = s(t)$ and $\lim_{n \rightarrow \infty} r_n(t) = d(t)$.

Proposition 4 (CORRECTONE for $e_a(t) = t$ and $e_b(t) = e(t)$, $e(t) < t$). *Let $a_0(t) = s(t) \cdot d(t)$ and $b_0(t) = s(t) \cdot d(e(t))$ for $t \geq 0$, where $s(t) > 0$ is the ground-truth signal, $d : \mathbb{R}_{\geq 0} \rightarrow [0, 1]$ is a continuous degradation function with $d(0) = 1$ and $e(t)$ is the exposure function of signal b , for which it holds $e(0) = 0$ and $e(t) < t$ for all $t > 0$. If we run algorithm for $n = 0, 1, \dots$:*

$$r_n(t) = \frac{a_0(t)}{b_n(t)}, \quad a_{n+1}(t) = \frac{a_0(t)}{r_n(t)}, \quad b_{n+1}(t) = \frac{b_0(t)}{r_n(e(t))},$$

then it holds $\forall t \geq 0 : \lim_{n \rightarrow \infty} a_n(t) = \lim_{n \rightarrow \infty} b_n(t) = s(t)$ and $\lim_{n \rightarrow \infty} r_n(t) = d(t)$.

Proof. Let us fix an arbitrary $t > 0$ and define $e^n(t) = \underbrace{(e \circ e \circ \dots \circ e)}_{n \text{ times}}(t)$. Since $r_0(t) = \frac{d(t)}{d(e(t))}$,

then by induction we have

$$\begin{aligned} r_n(t) &= \frac{a_0(t)}{b_n(t)} = \frac{a_0(t)}{\frac{b_0(t)}{r_{n-1}(e(t))}} = \frac{a_0(t)}{b_0(t)} \cdot r_{n-1}(e(t)) = \frac{a_0(t)}{b_0(t)} \cdot \frac{a_0(e(t))}{b_0(e(t))} \cdot r_{n-2}(e(e(t))) \\ &= \dots = \prod_{i=0}^{n-1} \frac{a_0(e^i(t))}{b_0(e^i(t))} \cdot r_0(e^n(t)) = \prod_{i=0}^n \frac{d(e^i(t))}{d(e^{i+1}(t))} = \frac{d(t)}{d(e^{n+1}(t))}. \end{aligned}$$

Calculating $b_n(t)$ yields

$$b_n(t) = \frac{b_0(t)}{r_{n-1}(e(t))} = \frac{s(t) \cdot d(e(t))}{\frac{d(e(t))}{d(e^{n+1}(t))}} = s(t) \cdot d(e^{n+1}(t))$$

and similarly for $a_n(t)$, we have

$$a_n(t) = s(t) \cdot d(e^n(t)).$$

By the same reasoning as in proposition 2, we get $\lim_{n \rightarrow \infty} d(e^{n+1}(t)) = 0$ and thus by taking the limit $n \rightarrow \infty$ on $b_n(t)$ we get

$$\lim_{n \rightarrow \infty} b_n(t) = \lim_{n \rightarrow \infty} s(t) \cdot d(e^{n+1}(t)) = s(t) \cdot d(0) = s(t),$$

from where it immediately follows $\lim_{n \rightarrow \infty} a_n(t) = s(t)$. Moreover, the ratio $r_n(t)$ converges to $d(t)$, which follows from continuity of d as

$$\lim_{n \rightarrow \infty} r_n(t) = \lim_{n \rightarrow \infty} \frac{d(t)}{d(e^{n+1}(t))} = \frac{d(t)}{\lim_{n \rightarrow \infty} d(e^{n+1}(t))} = \frac{d(t)}{d(0)} = d(t).$$

□

Theorem 2. Let $a_0(t) = s(t) \cdot d(e_a(t))$ and $b_0(t) = s(t) \cdot d(e_b(t))$ for $t \geq 0$, where $s(t) > 0$ is the ground-truth signal, $d : \mathbb{R}_{\geq 0} \rightarrow [0, 1]$ is a continuous degradation function with $d(0) = 1$ and $e_a(t), e_b(t) : [0, \infty) \rightarrow [0, \infty)$ are the continuous exposure function of signal a and b respectively. Let us further assume $e_a(0) = e_b(0) = 0$, $e_b(t) < e_a(t)$ for all $t > 0$ and that there exist function $e_a^{-1} : [0, \infty) \rightarrow [0, \infty)$. If we run algorithm for $n = 0, 1, \dots$:

$$r_n(t) = \frac{a_n(t)}{b_n(t)}, \quad a_{n+1}(t) = \frac{a_0(t)}{r_n(t)}, \quad b_{n+1}(t) = \frac{b_0(t)}{r_n((e_a^{-1} \circ e_b)(t))},$$

then it holds $\forall t \geq 0 : \lim_{n \rightarrow \infty} a_n(t) = \lim_{n \rightarrow \infty} b_n(t) = s(t)$.

Proof. Proof goes along the same lines as proof of theorem 1. □

Remark 2 (Rate of convergence). Both CORRECTONE and CORRECTBOTH, in case when $e_a(t) = t$ and $e_b(t) = \frac{t}{k}$ holds, converge in $\mathcal{O}(\log \frac{t}{\delta}) = \mathcal{O}(\log t)$ up to any fixed precision $\delta > 0$. This trivially follows from $\frac{t}{k^{n+1}} = \delta$.

We pondered how to extend our analysis from noise-free to noisy measurement model. Since the noise is assumed to be additive and Gaussian, and the expectation of a reciprocal Gaussian random variable does not exist, we were not able to show above claims for the noisy measurement model. However, empirical results suggest that it can be generalized to the noisy case. We suspect this might be due to noise elimination in FITCURVETO procedure. Furthermore, the assumption on $d(0) = 1$ was proven to be important by empirical analysis.

4 Data Fusion

From now on suppose that signals have been corrected for degradation. We are given two time series of measurements of the ground-truth signal $s(t)$, $(a_c[k])_{k \in [n_a]}$ and $(b_c[k])_{k \in [n_b]}$. Measurements are obtained through a noisy measurement process, defined by equation (9) (cf. (2)), where ε_a and ε_b are white noise signals and are independent of each other.

$$\begin{aligned} a_c(t) &= s(t) + \varepsilon_a(t), \quad \varepsilon_a(t) \sim \mathcal{N}(0, \sigma_a^2) \\ b_c(t) &= s(t) + \varepsilon_b(t), \quad \varepsilon_b(t) \sim \mathcal{N}(0, \sigma_b^2) \end{aligned} \tag{9}$$

Let us start addressing the problem of estimating the ground-truth signal s . We would like to utilize the information embedded in observations $(a_c[k])_{k \in [n_a]}$ and $(b_c[k])_{k \in [n_b]}$ to get a belief distribution about the signal s .

The underlying process of s is assumed to be Brownian motion. Kalman filter proposed by [11] provides a powerful and efficient estimation method. However, it is limited to discrete time process model. Since the process model of s is random and not known, we would also like to incorporate time differences between two consecutive measurements. Gaussian processes provide a powerful framework for this problem.

Assume, that $s(t)$ is a Gaussian process with zero mean and a covariance function $k(\cdot, \cdot)$ specified in (10).

$$s(t) \sim GP(0, k(t, t')). \tag{10}$$

The covariance or the kernel function parameters, denoted by θ , are selected by maximizing log marginal likelihood. For a fixed t , the belief of signal s given observations is computed as a posterior probability distribution.

4.1 Gaussian processes

What follows is summarized from [15], but we repeat the relevant parts for reader's convenience. We can imagine Gaussian processes as a distribution over functions. Formally we can define it as follows:

Definition 1. A Gaussian process is a collection of random variables, such that any finite subsample has a joint Gaussian distribution.

A Gaussian process f is described with mean function $\mu(\cdot)$ and covariance function $k(\cdot, \cdot)$. Assuming that our finite subsample is a vector $\mathbf{x} \in \mathbb{R}^n$, then $f(\mathbf{x}) \sim \mathcal{N}(\mu, \Sigma)$, where

$$\begin{aligned}\mu &= \mu(\mathbf{x}) \\ \Sigma &= k(\mathbf{x}, \mathbf{x})\end{aligned}$$

and $k(\mathbf{x}, \mathbf{x})$ is a matrix with entries $(k(\mathbf{x}, \mathbf{x}))_{i,j} = k(x_i, x_j)$. Let us now describe how we compute the posterior belief of function f at point x_* if we observe n points $\mathbf{y} = f(\mathbf{x})$. Since $[\mathbf{x}^T, x_*]$ is again a vector, it holds:

$$\begin{bmatrix} \mathbf{y} \\ f_* \end{bmatrix} \sim \mathcal{N} \left(\begin{bmatrix} \mu(\mathbf{x}) \\ \mu(x_*) \end{bmatrix}, \begin{bmatrix} k(\mathbf{x}, \mathbf{x}) & k(\mathbf{x}, x_*) \\ k(x_*, \mathbf{x}) & k(x_*, x_*) \end{bmatrix} \right). \quad (11)$$

Then we obtain posterior belief of f_* using formulas for conditional Gaussian distributions. In many cases there is some noise in our measurements: $\mathbf{y} = f(\mathbf{x}) + \varepsilon$, where $\varepsilon \sim \mathcal{N}(0, \sigma^2 I)$. To incorporate the noise we correct the equation (11) as follows:

$$\begin{bmatrix} \mathbf{y} \\ f_* \end{bmatrix} \sim \mathcal{N} \left(\begin{bmatrix} \mu(\mathbf{x}) \\ \mu(x_*) \end{bmatrix}, \begin{bmatrix} k(\mathbf{x}, \mathbf{x}) + \sigma^2 I & k(\mathbf{x}, x_*) \\ k(x_*, \mathbf{x}) & k(x_*, x_*) \end{bmatrix} \right). \quad (12)$$

We can generalize above formulation to include observations from arbitrary many different measurement instruments, each with its intrinsic variance of noise. Denote the set of observations of Gaussian Process s by $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^n = (\mathbf{x}, \mathbf{y})$, where for each i we know which instrument produced the observation. Here, we only show how to distinguish between two instruments, PMO6V-A and PMO6V-B. In this light, we replace term $\sigma^2 I$ in equation (12) with $\text{diag}(\boldsymbol{\sigma}^2)$, where $\boldsymbol{\sigma} = [\sigma_1, \dots, \sigma_n]^\top$ such that

$$\sigma_i^2 = \begin{cases} \sigma_a^2 & \text{if measurement } i \text{ came from instrument PMO6V-A,} \\ \sigma_b^2 & \text{if measurement } i \text{ came from instrument PMO6V-B.} \end{cases}$$

Since the analysis when differentiating between measuring instruments is similar (but its notation is more tedious) we continue as if all observations are produced by a single instrument. Usually, the mean function μ and the kernel function k are described in parametric representation. Let $\boldsymbol{\theta}$ denote all Gaussian process hyperparameters, including σ which is added due to noise.

Let us now write the explicit prediction of $y_* = f(x_*)$, given observations \mathcal{D} and x_* :

$$\begin{aligned}y_* | \mathbf{y}, \boldsymbol{\theta}, \mathbf{x}, x_* &\sim \mathcal{N}(\mu_*, \Sigma_*), \text{ where} \\ \mu_* &= \mu_{\boldsymbol{\theta}}(x_*) + k_{\boldsymbol{\theta}}(x_*, \mathbf{x}) (k_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{x}) + \sigma^2 I)^{-1} (\mathbf{y} - \mu_{\boldsymbol{\theta}}(\mathbf{x})), \quad (\text{GAUSSIANPROCESS}) \\ \Sigma_* &= k_{\boldsymbol{\theta}}(x_*, x_*) - k_{\boldsymbol{\theta}}(x_*, \mathbf{x}) (k_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{x}) + \sigma^2 I)^{-1} k_{\boldsymbol{\theta}}(\mathbf{x}, x_*).\end{aligned}$$

It is often the case that we either do not have or do not want to incorporate any prior knowledge about mean function μ , thus it is common to consequently set it to 0, i.e. $\mu(\mathbf{x}) \equiv 0$. Parameters $\boldsymbol{\theta}$ are chosen, such that the log marginal likelihood of observations \mathbf{y} at points \mathbf{x} is maximized, i.e. we maximize the quantity

$$\log p(\mathbf{y} | \mathbf{x}).$$

Using formula $p(\mathbf{y} | \mathbf{x}) = \int p(\mathbf{y} | \mathbf{f}, \mathbf{x}) p(\mathbf{f} | \mathbf{x}) d\mathbf{f}$, where $\mathbf{f} = f(\mathbf{x})$ and assumptions on Gaussian priors we obtain

$$\log p(\mathbf{y} | \mathbf{x}) = -\frac{1}{2} \mathbf{y}^T (k_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{x}) + \sigma^2 I)^{-1} \mathbf{y} - \frac{1}{2} \log |k_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{x}) + \sigma^2 I| - \frac{n}{2} \log(2\pi). \quad (13)$$

Let us discuss the main limitation of Gaussian processes. Given n observations, we need to compute the inverse of a $n \times n$ matrix. Time complexity of such operation is $\mathcal{O}(n^3)$, which turns out not to be scalable, e.g. on a personal computer one can only use a couple thousands of data points because of memory and computation speed limitations. Therefore, in the case when we have millions or even billions of data points a rather cumbersome downsampling has to be performed. However, we would still like to use the idea of Gaussian processes, so there is a need for a different approach. In the next subsection we discuss sparse Gaussian processes, which can be scaled up to millions and even billions of observations.

4.2 Sparse Gaussian Processes

Assume that we are in the same setting as above, i.e. we are given n observations $\mathcal{D} = (\mathbf{x}, \mathbf{y})$ and our goal is to find hyperparameters θ such that log marginal likelihood $\log p(\mathbf{y}|\mathbf{x})$ is maximized. As we have seen in the equation (13), evaluating $\log p(\mathbf{y}|\mathbf{x})$ for specific θ costs $\mathcal{O}(n^3)$. Since n can be large, we would like to obtain a method which would compute approximate value of $\log p(\mathbf{y}|\mathbf{x})$ for specific θ with much lower complexity. To tackle this problem we construct an expression which is a lower bound for $\log p(\mathbf{y}|\mathbf{x})$ and can be directly optimized. Define inducing points \mathbf{u} as vector of $m \leq n$ points. Authors in [2] prove the following:

$$\log p(\mathbf{y}|\mathbf{x}) \geq -\frac{1}{2}\mathbf{y}^T(Q_\theta + \sigma^2 I)^{-1}\mathbf{y} - \frac{1}{2}\log|Q_\theta + \sigma^2 I| - \frac{n}{2}\log(2\pi) - \frac{1}{2\sigma^2}\text{tr}(k_\theta(\mathbf{x}, \mathbf{x}) - Q_\theta),$$

where $Q_\theta = k_\theta(\mathbf{x}, \mathbf{u}) \cdot k_\theta(\mathbf{u}, \mathbf{u})^{-1} \cdot k_\theta(\mathbf{u}, \mathbf{x})$. Moreover, in [2] it is shown that right-hand side can be computed in $\mathcal{O}(nm^2)$, which for $m \ll n$ becomes much more tractable.

Next, the idea is to maximize the latter expression with respect to parameter θ and \mathbf{u} . Then, the optimal parameters are utilized to make statements about posterior distribution of f . Findings on variational approach introduced in [10] can be exploited here. We know that all information about the posterior obtained by maximization of lower bound is contained in a random variable with Gaussian distribution $\mathbf{z} \in \mathbb{R}^m$, $\mathbf{z} \sim \mathcal{N}(\mu_{\mathbf{u}}, \Sigma_{\mathbf{u}})$, where

$$\begin{aligned} \Sigma_{\mathbf{u}} &= k_\theta(\mathbf{u}, \mathbf{u})^{-1} - \frac{1}{\sigma^2}k_\theta(\mathbf{u}, \mathbf{u})^{-1}k_\theta(\mathbf{u}, \mathbf{x})k_\theta(\mathbf{x}, \mathbf{u})k_\theta(\mathbf{u}, \mathbf{u})^{-1}, \\ \mu_{\mathbf{u}} &= \frac{1}{\sigma^2}\Sigma_{\mathbf{u}}k_\theta(\mathbf{u}, \mathbf{u})^{-1}k_\theta(\mathbf{u}, \mathbf{x})\mathbf{y}. \end{aligned} \quad (\text{SPARSE VARIATIONAL GP})$$

Since $[\mathbf{z}^T, x_*]$ is again Gaussian distributed we can obtain f_* using formulas for conditional Gaussian distributions. This can be computed in $\mathcal{O}(nm^2)$.

As we approximated the solution, we also have to address the question about the accuracy of the approximation. It was shown in [3] that if RBF kernel is used, m needs to be of order $\log n$ in order that approximation error tends to 0. In case of Matern kernel it was shown that for $k + \frac{1}{2}$, m needs to be of order $n^{\frac{1}{2k+1}}$ that the approximation error tends towards 0. We were maximizing our objective with mini batches of 200 samples. In our example we used around 2 000 000 data points (10 000 iterations) to train Gaussian process regression model and for that we used around 1 000 inducing points.

Empirical observations showed that if the number of inducing points is small (in our case a couple of hundred) then the model is not capable of capturing fast fluctuations and is instead finding some global trends. With increasing number of inducing points the model can also capture fast fluctuations. Moreover, by increasing m , Gaussian processes do not overfit the dataset (do not show any additional small scale behavior) if fast fluctuations do not exist.

4.3 Local Gaussian Processes

An intuitive and much simpler approach can be used to overcome the computational complexity of Gaussian processes. It uses a local approximation, i.e. when predicting the distribution of y_* at x_* we only consider data points in \mathcal{D} that are “close” to x_* or more precisely we use a subset $\mathcal{D}_w \subseteq \mathcal{D}$, such that $\mathcal{D}_w = \{(x, y) \in \mathcal{D} : |x - x_*| \leq \frac{w}{2}\}$ for some window length w . Let us denote this model by LOCALGP.

Thus, for every x_* we would need to find the optimal parameters θ given points in \mathcal{D}_w . We can slightly alter this approach, such that we predict the posterior for multiple points at same time. For

a given x_* and \mathcal{D}_w , we compute the posterior for points $\mathcal{X}_{w'} = \{x \in \mathbf{x} : |x - x_*| \leq \frac{w'}{2}\}$, where $w' < w$, e.g. $w' = \frac{w}{3}$. For Matern kernel that we are using, we optimize its parameters by limiting the length-scale parameter, such that points outside the window have a negligible contribution to predicted values.

5 Results

In this section we first present an evaluation of the proposed modeling, correction and data fusion methods on a synthetically generated dataset. Next, we apply the proposed methods on the level-1 time series from instruments PMO6V-A and PMO6V-B. Output result is a probability distribution over functions given by the Gaussian process and the predicted mean function is our final level-2 time series VIRGO-PMO6-V TSI. Our implementation is available on GitHub¹ along with instructions and is further discussed in appendix A.3.

Both methods for computing exposure, NUMMEASUREMENTS and CUMSUM perform equally well on the datasets. On the VIRGO dataset this might be due to the fact that TSI signal is approximately constant, i.e. the fluctuations of a signal (approx. $\pm 2 \frac{W}{m^2}$ or 0.15 %) are significantly smaller than its absolute value (approx. $1366 \frac{W}{m^2}$). Similarly, both correction methods CORRECTONE and CORRECTBOTH perform well and converge. However, we observe that the former has a slightly faster convergence. In our experiments we use the Matern kernel for the covariance function with $\nu = \frac{1}{2}$, $k(t, t') = \sigma^2 \cdot e^{-\frac{|t-t'|}{l}}$.

5.1 Synthetic dataset

We first evaluate proposed methods on a synthetic dataset, where the ground-truth signal is known and is generated by simulating a Brownian motion of length n shifted by s_0 , such that all signals are strictly positive (avoid division by 0). Signal a contains each data point from ground-truth signal with probability p_a , whereas signal b contains each data point only with probability $p_b < p_a$, i.e. instrument a has about $\frac{p_a}{p_b}$ -times the sampling rate of instrument b . Downsampling for both signals is performed uniformly at random. Then, the exposure is computed and both signals are degraded according to EXP, where parameters θ are sampled uniformly at random. Finally, additive white noise is added to both signals. Note that the values of n , p_a , p_b and shift s_0 are arbitrary.

5.1.1 Noiseless signals

Figure 2 shows the ground-truth signal s and degraded signals a and b (left) and signal obtained after running the iterative correction algorithm. It clearly shows that corrected signals in noise free case converge to the ground-truth signal. This is empirical evidence that supports theorems proposed and proved in section 3.

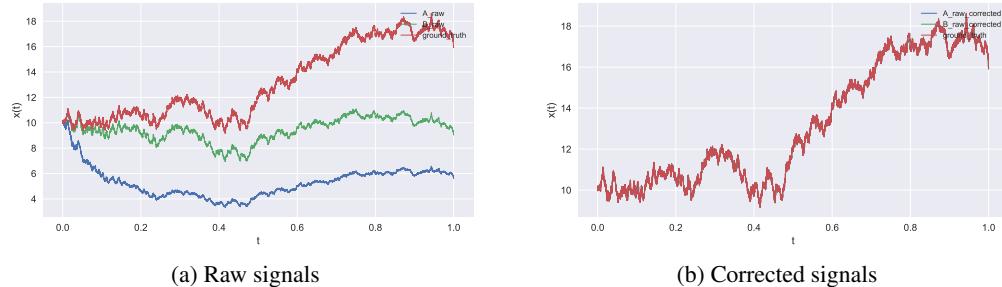


Figure 2: Raw (left) and corrected (right) noise-free synthetic signals using CORRECTONE method and SMOOTHMONOTONIC model.

¹https://github.com/roksikonja/dslab_virgo_tsi

5.1.2 Noisy signals

Now, let us consider the general case with additive white noise present. These signals are shown on the left side of figure 3, whereas on the right side the initial ratio between signals a and b is visualized. The ratio is clearly not monotonically non-increasing. However, we argue that all information for degradation correction and retrieval of s is embedded in it.

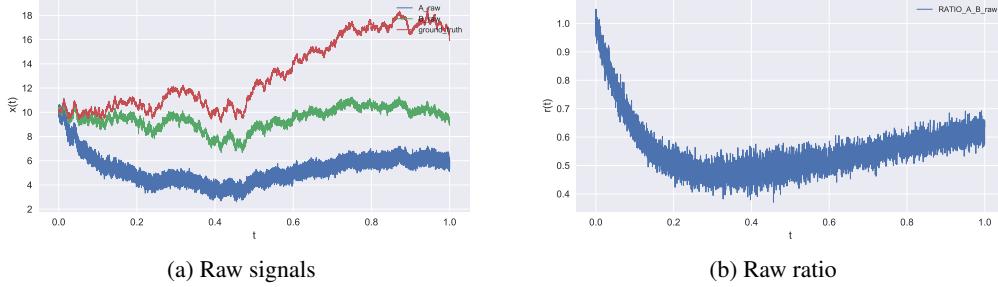


Figure 3: Raw synthetic signals and their ratio together with the ground-truth signal.

After performing correction we can see in figure 4 that both corrected signals effectively converge to the ground-truth signal, i.e. the mean of noisy signals converge to s . Moreover, the ratio of corrected signals converges to a constant unit function. The green line represents degradation of signal a , which is indeed a monotonically non-increasing function.

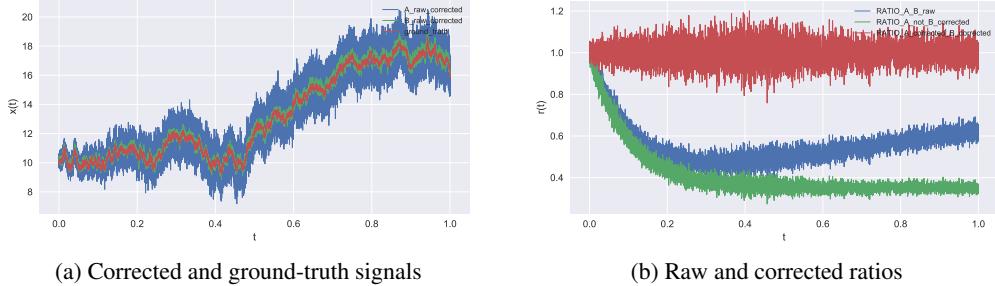


Figure 4: Corrected synthetic signals, and comparison of raw and corrected ratios using SMOOTHMONOTONIC and CORRECTONE.

Although corrected signals are centered around the ground-truth signal, the correction has amplified the noise, with the amplification being greater as t increases. This effect was expected, since correction consists of divisions by a value smaller than 1.

We apply both SPARSEVARIATIONALGP and LOCALGP data fusion methods on the corrected signals and visualize the output signals, where the bold line represents the predicted mean and the light area the 95 % confidence interval. In figure 5 we plot the output signal together with the data points obtained after correction. The output signal fits the ground-truth very well, however, it is a bit smoother because the small number of inducing points cannot capture fast fluctuations.

Sparse Gaussian processes provide a simple and powerful framework for observing the signal on multiple timescales by varying the number of inducing points m , which can be seen in figures 5 and 6 with $m = 500, 100$. We can observe how by increasing m , the level of detail increases. Moreover, right side of figure 6 shows the output of LOCALGP. It provides a simple method for merging data from multiple instruments, but because of downsampling and optimizing kernel hyperparameters in every window it can overfit. We observe that it produces confident predictions, but we do not have any theoretical justification for it to be correct.

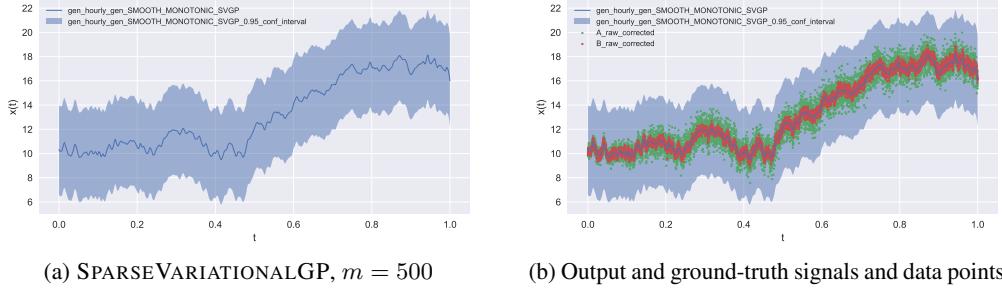


Figure 5: Ground-truth estimation with 95 % confidence interval using SPARSE VARIATIONALGP and $m = 500$.

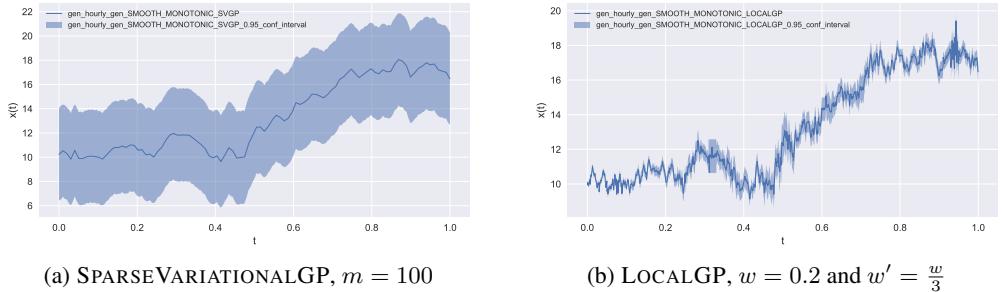


Figure 6: Illustration of output signal's multiple timescale approximation of ground-truth with 95 % confidence interval.

5.2 VIRGO PMO6-V dataset

We now apply the proposed methods on PMO6-V TSI time series from SOHO spacecraft.

Figures 7 and 8 show the result of applying proposed algorithms on whole PMO6-V TSI time series, with left part of figure 7 showing the degradation of instruments PMO6V-A and PMO6V-B and left part of figure 8 visualizing the trend of confidence interval width. It shows that over time there is a slight trend of width increase and a natural width extension, when no measurements are present for longer periods of time, e.g. during SOHO vacations.

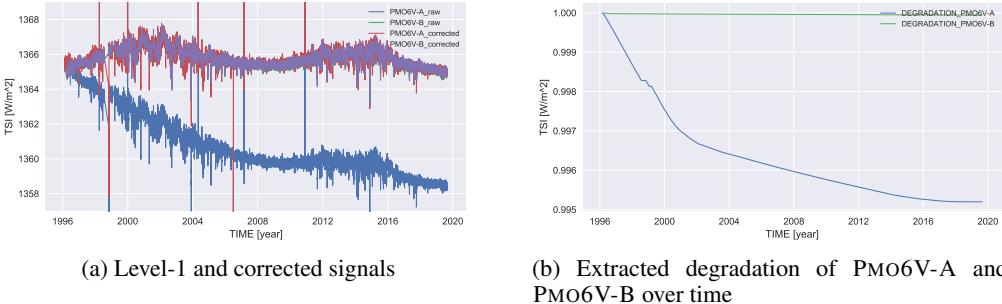
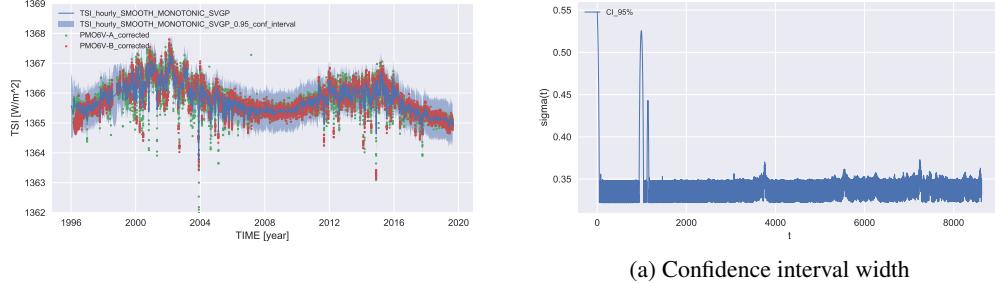


Figure 7: Comparison of raw and corrected TSI time series from PMO6V-A and PMO6V-B and their degradation during VIRGO mission.



(a) Confidence interval width

Figure 8: Output signals using SPARSE VARIATIONALGP with $m = 1000$ and confidence interval width over time.

Left and right part of figure 9 show output signal for first half of data points when using 300 and 1000 inducing points, respectively. This figure indicates that with higher number of inducing points more rapid trends can be captured by the model. We believe that after a certain number of inducing points, adding additional inducing points would not make any difference, since the density of inducing points is high enough to capture all the trends of Sun irradiance.

We observed that with around 1 000 inducing points model is also capable of capturing fast fluctuations. However, the actual number of inducing points should be decided by experts, based on the trends they would like to capture.

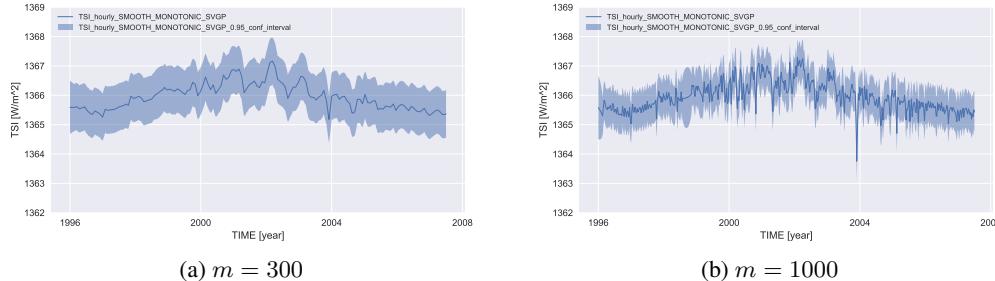


Figure 9: Output signals using SPARSE VARIATIONALGP with different number of inducing points. The smoothness and the ability of the method to capture fast fluctuations are compared.

Regarding the interpretability of the confidence interval for TSI signal, by using Gaussian processes we are required to give some prior belief or assumption on how our signal should look like, this is determined by our selection of the covariance function. The left part of figure 10 visualizes some samples from the prior distribution given the Matern kernel with $\nu = \frac{1}{2}$, whereas the right part shows posterior samples after training. Samples deviating far from the mean are less likely to be sampled.

We also compared performances of different regression models presented in section 2. In figure 11 we show the fitted degradation functions on synthetic and PMO6-V dataset. We observe that ISOTONIC, SMOOTHMONOTONIC and EXP fit the degradation function well. The first two surpass the latter in terms of robustness, because of the need of initial fitting for parameters of EXP, which are used as initialization for Levenberg–Marquardt iterative algorithm. SPLINE and EXPLIN are much less stable and robust.

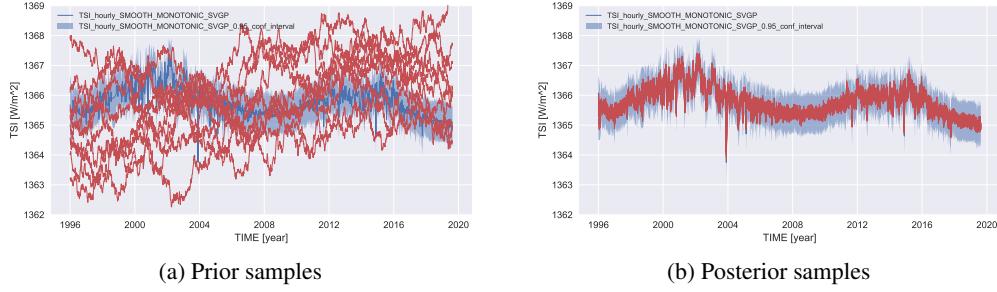


Figure 10: Sampling prior and posterior distribution given by SPARSE VARIATIONAL GP with $m = 100$.

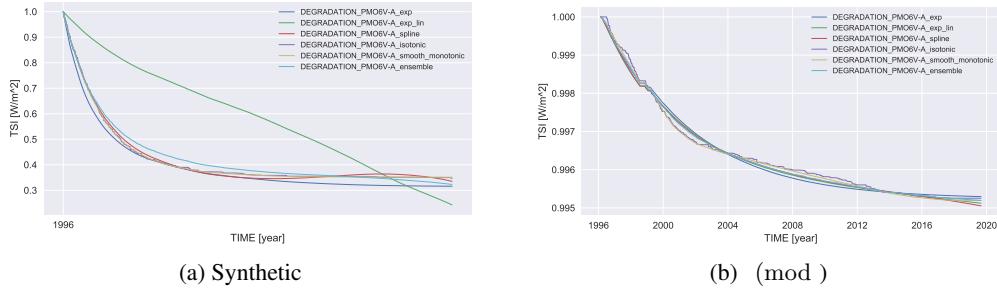


Figure 11: Degradation of synthetic signal a and PMO6V-A instrument over time.

Acknowledgments

This paper is a result of Data Science Lab at ETH Zürich taking part in Autumn 2019. We would like to thank Prof. Dr. Andreas Krause and Prof. Dr. Ce Zhang for their supervision, invaluable insights and suggestions on the modeling methods. Furthermore, we thank Dr. Wolfgang Finsterle for his domain knowledge support and data preparation.

References

- [1] M Anklin, C Fröhlich, W Finsterle, D A Crommelynck, and S Dewitte. Assessment of degradation of VIRGO radiometers on board SOHO. *Metrologia*, 35(4):685–688, aug 1998.
- [2] Matthias Bauer, Mark van der Wilk, and Carl Edward Rasmussen. Understanding Probabilistic Sparse Gaussian Process Approximations. *arXiv e-prints*, page arXiv:1606.04820, Jun 2016.
- [3] David R. Burt, Carl E. Rasmussen, and Mark van der Wilk. Rates of Convergence for Sparse Variational Gaussian Process Regression. *arXiv e-prints*, page arXiv:1903.03571, Mar 2019.
- [4] Federico Castanedo. A Review of Data Fusion Techniques. *The Scientific World Journal*, 2013:1–19, 2013.
- [5] Claus Fröhlich. Degradation of Radiometers in Space: Application to VIRGO TSI, 2014. Online; accessed 8 November 2019.
- [6] Thierry Dudok de Wit, Greg Kopp, Claus Fröhlich, and Micha Schöll. Methodology to create a new total solar irradiance record: Making a composite out of multiple data records. *Geophysical Research Letters*, 44(3):1196–1203, February 2017.
- [7] P. Dierckx. An algorithm for smoothing, differentiation and integration of experimental data using spline functions. *Journal of Computational and Applied Mathematics*, 1(3):165 – 184, 1975.

- [8] Claus Fröhlich. Long-term behaviour of space radiometers. *Metrologia*, 40(1):S60–S65, feb 2003.
- [9] Claus Fröhlich, Dominique A. Crommelynck, Christoph Wehrli, Martin Anklin, Steven Dewitte, Alain Fichot, Wolfgang Finsterle, Antonio Jiménez, André Chevalier, and Hansjörg Roth. In-Flight Performance of the VIRGO Solar Irradiance Instruments on SOHO, 1997.
- [10] James Hensman, Nicolo Fusi, and Neil D. Lawrence. Gaussian Processes for Big Data. *arXiv e-prints*, page arXiv:1309.6835, Sep 2013.
- [11] R. E. Kalman. A New Approach to Linear Filtering and Prediction Problems. *Journal of Basic Engineering*, 82(1):35–45, 03 1960.
- [12] Kenneth Levenberg. A METHOD FOR THE SOLUTION OF CERTAIN NON-LINEAR PROBLEMS IN LEAST SQUARES. *Quarterly of Applied Mathematics*, 2(2):164–168, 1944.
- [13] Donald W. Marquardt. An Algorithm for Least-Squares Estimation of Nonlinear Parameters. *Journal of the Society for Industrial and Applied Mathematics*, 11(2):431–441, 1963.
- [14] Sky Mckinley and Megan Levine. Cubic Spline Interpolation, 1998.
- [15] Carl Edward Rasmussen and Christopher K. I. Williams. *Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning)*. The MIT Press, 2005.
- [16] Greg Stickler. Educational Brief - Solar Radiation and the Earth System, 2016. Online; accessed 8 November 2019.
- [17] Oleg Sysoev and Oleg Burdakov. A smoothed monotonic regression via L2 regularization. *Knowledge and Information Systems*, 59(1):197–218, Apr 2019.
- [18] Shrihari Vasudevan. Data fusion with Gaussian processes. *Robotics and Autonomous Systems*, 60(12):1528 – 1544, 2012.

A Appendix

A.1 Iterative Correction

In figure 12 we illustrate first three steps of iterative correction algorithm, where the initial time series are visualized in figure 3. If signal b is estimated well, correction for that step is better as well. This is an intuitive idea underlying both correction methods.

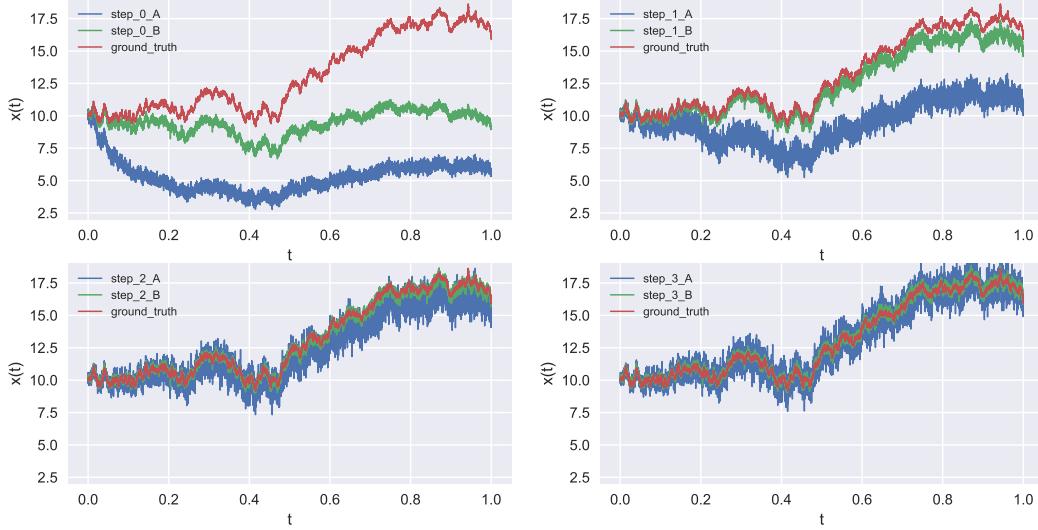


Figure 12: Convergence history.

A.2 Level-2 VIRGO TSI

The comparison of level-2 VIRGO TSI time series produced by [5] and ours is shown in 13. It has to be noted that level-2 time series by [5] were produced by utilizing information of PMO6-V and DIARAD level-1 time series. Here, we only compare the smoothed time series on the left (bold red) and the predicted mean function on the right (bold blue). Note that light red and light blue fields do no represent the same quantity: on the left the time series after degradation correction and on the right 95 % confidence interval for the ground-truth signal s .

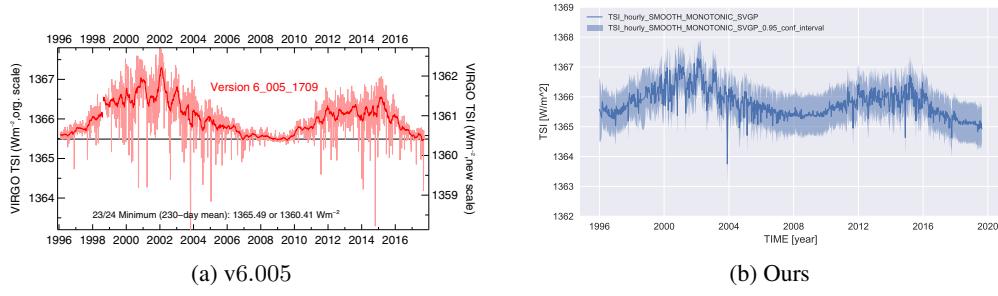


Figure 13: Comparison of level-2 data between [5] and us.

A.3 Implementation

All presented approaches are implemented in Python². Apart from bare-bones implementation, we also provide graphical user interface that can be used via browser. This way user can reproduce our results without the knowledge of any programming language. We use Flask³ to connect the interface with the implementation of our approaches. Figure 14 shows some screenshots.

²<https://www.python.org/>

³<https://www.palletsprojects.com/p/flask/>

Dashboard

Import the data and then select it from the table to begin analysis.

Loaded Data

[Import Dataset](#)

#	Dataset Name	Exposure Mode	Outlier Fraction	
1	SOHO VIRGO	Num. measurements	0.0	Select Delete

Data Analysis

Merging at: 20 %

Import Dataset

Dataset Name Filename will be taken as a default name.

Dataset File data.csv

Exposure Method

Outlier Fraction Value must be between 0 and 1.

[Import](#) [Back](#)

Analysis for SOHO VIRGO

Model

Model Parameters

Correction method

Output method

[Submit](#) [Back](#)

Figure 14: Screenshots of graphical user interface: dashboard with overview of loaded datasets while analysis is running (top), form for import of dataset (middle), and form for analysis of a loaded dataset (bottom).