

Evaluation of Kalman filter for meteorological time series imputation for Eddy Covariance applications

Simone Massaro

First Supervisor: Dr. Franziska Koebsch
Second Supervisor: Prof. Dr. Fabian Sinz

Master Thesis
Forest and Ecosystem Sciences
Ecosystem Analysis and Modelling

Faculty of Forestry and Forest Ecology
Georg-August-Universität Göttingen

March 2023

Contents

1	Introduction	3
2	Methods	7
2.1	Kalman Filter Theory	7
2.1.1	Time update	8
2.1.2	Measurement update	8
2.1.3	Smoothing	10
2.1.4	Predictions	10
2.2	Kalman Filter Implementation	10
2.2.1	Requirements	10
2.2.2	Numerical stability	11
2.2.3	Implementation in PyTorch	11
2.2.4	Time update Square Root Filter	12
2.2.5	Measurement update Square Root Filter	12
2.2.6	Predictions Square Root Filter	14
2.2.7	Smoothing Square Root Filter	14
2.3	Kalman Filter Model	14
2.3.1	Parameters	14
2.3.2	Parameters initialization	15
2.3.3	Loss Function	16
2.3.4	Metrics	17
2.3.5	Performance considerations	17
2.4	Data	17
2.4.1	Data source	17
2.4.2	Data preparation pipeline	18
2.4.3	Prediction pipeline	18
2.5	Model Training	19
2.6	Other methods	22
2.6.1	MDS	22
2.6.2	ERA	22
2.7	Code Details and Availability	22
3	Results	23
3.1	Correlation characteristics of meteorological variables	23
3.2	Comparison to other imputation methods	25
3.3	Aspect affect Kalman Filter performance	32
3.3.1	Gap Length	32
3.3.2	Control variables	32
3.3.3	Gaps in multiple variables	32
3.4	Kalman Filter training	36
3.4.1	Variable fine-tuning	36
3.4.2	Training limitations	36

4 Discussion	39
4.1 Kalman Filter performance	39
4.2 Kalman Filter application	40
4.3 Future Outlook	41
5 Conclusions	43
A Additional Results	47
A.1 Comparison to other imputation methods	47
A.2 Additional Time series	50
A.3 Additional Tables	55
A.4 Gap length distribution in FLUXNET	61
B Comparison between Standard and Square Root Kalman Filter	64

Abstract

Eddy Covariance (EC) is a state of the art technique to measure greenhouse gases exchanges. EC towers include measurement of meteorological variables, but due to instrument failures the data is no always available. Many use cases of EC data, especially Land Surface Models, require continuous meteorological time series as input. Therefore, it's necessary to impute the gaps in the meteorological time series. ONEFlux, one of the most widely used EC post-processing pipelines, imputes the missing data using either Marginal Distribution Sampling (MDS), which uses other observations from similar meteorological conditions, or ERA-Interim (ERA-I), that is a global meteorological dataset. The imputation performance of those methods is limited for short and medium gaps (up to 1 week), which represent the majority of EC meteorological gaps.

In this work, I assess an imputation method for meteorological variables based on a Kalman Filter (KF). It has the advantages of combining in the prediction information from the ERA-I dataset, inter-variable correlation and temporal autocorrelation. Moreover, the KF is a probabilistic method, so for each data point the prediction is not a single value but an entire distribution, which provides an interpretable uncertainty of the model prediction.

I evaluate the Kalman Filter by comparing the imputation performance with the state of the art approaches (MDS and ERA-I) using data from the FLUXNET site of Hainich (DE-Hai) with gaps up to one week long. The KF outperforms the state of the art approaches across all analysed variables, with the exception of precipitation. I observed an average reduction of the imputation error of 33 % compared to ERA-I and 57 % compared to MDS. I further explore aspects that influence the performance of the KF: in general the error increase with the gap length only up to 24 hours, the use ERA-I data improves the model predictions and the inter-variable correlation is effectively utilized. The main limitations of KF approach are: the best performance is achieved only when fine tuning the model parameters to the specific conditions of the gap, which increases the deployment complexity; numerical instability, that in the current implementation limits the gap length to 15 hours if all variables are missing; difficulty of learning models parameters, thus requiring careful initialization and training.

1 Introduction

Eddy Covariance Eddy Covariance (EC) is a state of the art technique for measuring greenhouse gases and energy exchange between ecosystems and the atmosphere [1]. The technique allows for non-destructive measurements at the ecosystem level with a high temporal resolution (half and hour). EC data is used for ecological and physiological research of ecosystems, for example to estimate the relation of forest age and carbon balance [3] or the effects of extreme events [27]. In addition, EC data is a key element of the validation and calibration of validation of ecosystem process model and remote sensing observations [32]. The core of EC technique a site is the 3D anemometer and a gas analyser, which allows estimating the fluxes of interests (e.g. CO₂, H₂O, CH₄). Beside the fluxes, an Eddy Covariance setups commonly comprises measurements of meteorological variables and ecosystem parameters. This additional data provides the context to use and interpret the fluxes measurements.

Meteorological gaps The acquisition of the meteorological variables can be interrupted by failures in the instruments or power outages, resulting in gaps in the time series [1]. The presence of meteorological gaps is a problem for several uses of the EC data.

An important use case of EC is the validation of Land Surface Models [2, 18, 6, 26], which are process based model that estimate fluxes and include meteorological conditions as inputs. The errors of Land Surface Models deriving from inaccuracies in the input are comparable to the errors arising from the limitation in the models formulations [44]. Secondly, meteorological observations are used as a driver to impute gaps in the fluxes measurements [1], which in turn requires complete meteorological time series. Finally, if the observations are aggregated, for example to compute the weekly average of meteorological conditions, the missing data leads to inaccurate results.

The described use cases highlight the need for high quality continuous meteorological measurement that reflect that condition at the EC station. The first approach to reduce the number of gaps is to have redundant instruments and power supply on the site. However, even a redundant system is subject to failures, so statistical models are used for imputing the remaining gaps [1].

Imputation approaches In general there are three approaches to obtain information on missing values in a multivariate time series and thus impute the missing data: 1) use other observations of the missing variable to make predictions about the gap, in particular the variable *temporal autocorrelation* can be exploited to reconstruct the missing data; 2) use *inter-variable correlation*, if not all variables are missing then the correlation between variables can be used for imputing the missing variable; 3) use *other independent measurements*, if another compatible and continuous time series is available it can be used for imputation. For instance, in the case of EC meteorological variables an independent time series can be obtained from a nearby meteorological station or a weather model reanalysis.

Imputation of missing values has been extensively researched and a wide range of methods have been developed ranging from simply replacing with the mean to more advanced approaches employing deep neural networks [30, 16, 7, 14, 43, 8]. There exists several methods specifically developed to impute meteorological time series [11, 24]. However, those methods cannot be directly employed as imputation in the EC data has some specific characteristics: the absence of a spatial component (EC site are too distant from each other), the high temporal resolution and the relatively high number of variables.

Current method imputation EC community EC post-processing pipeline impute meteorological time series. Arguably the most widely used post-processing pipeline is ONEFlux [34], which is adopted by several large networks such as FLUXNET, the global EC network, ICOS the European network as well as AmeriFlux, the American EC network. ONEFlux uses two different methods for imputing the meteorological data: Marginal Distribution Sampling (MDS) and ERA-Interim (ERA-I). The final gap-filled meteorological product uses either MDS or ERA-I, depending on the quality flag of MDS.

MDS Marginal Distribution Sampling [37] imputes the missing value by using the average of all the other data points observed in similar conditions. The similarity is both temporal, only observations from a limited time window around the gap are considered, and meteorological, the data points are restricted to times when other variables are similar. The algorithm selects all the data points where the value of the driver variables (other meteorological variables) is within a fixed threshold of the one observed at the missing data point. All the observations of the variables of interest from the selected data points are then averaged to generate the filling value. MDS starts with a time window of 7 days and if no similar condition are found in this time frame the window is progressively increased. If a driver variable is also missing it is not used in the selection of similar

conditions. In case no similar conditions are found in a time window of 14 days or all drivers are missing, the MDS fails and the imputation is done using the average value at the same time of the day. For gaps longer than 140 days the algorithm cannot impute the gap. MDS imputes each data point and each variable separately. It mainly uses inter-variable correlation and in a limited way the variable temporal autocorrelation.

The algorithm implemented in ONEFlux uses as drivers the incoming shortwave radiation (`SW_IN`), air temperature (`TA`) and Vapour pressure deficit (`VPD`). If either `TA` or `VPD` is missing, `SW_IN` is used as the only driver. MDS has a quality flag with 3 possible values (i.e. 1,2,3) that depends on the size of the time window. In ONEFlux MDS is used only if the quality flag is 1, which in general means that similar conditions are found in a time window smaller than 14 days (details are in figure A1 from [37]).

ERA-Interim ERA-Interim (ERA-I) is a global meteorological dataset provided by the European Centre for Medium-range Weather Forecast (ECMWF) [13]. Weather forecast models are used to reanalyse past observations and produce a continuous and complete dataset for all the globe. The main drawback is the low spatial resolution and temporal resolution, that are respectively 80km and 3 hours. Moreover, only a subset of the meteorological variables are available in ERA-I (see table 2) and the data is not available in real-time but with a 3 months delay.

In order to use the ERA-I in the EC context the observations have to be temporally downsampled to match the half-hourly frequency of EC data. Furthermore, the performance of ERA-I imputation can be improved by removing the systematic bias for each site. Both steps are performed in ONEFlux as described in [41]. The error correction is performed using a different linear regression for each site and each variable.

The accuracy of the ERA-I imputation is independent of the length of the gap. This is advantageous for long gaps, as ERA-I data includes long-term evolution of the weather, that is not possible to predict by only analysing the local time series. At the same time, for short gaps, the local conditions can provide a more accurate prediction. In fact, ONEFlux imputes the short gaps using MDS and long gaps with ERA-I.

Other methods Beyond ONEFlux, there are several other established EC post-processing pipeline which impute meteorological data. However, the imputation approaches in other libraries are very similar, like REddyProc [42] or OzFlux [23] are very similar. REddyProc employs only MDS for the imputation, while OzFlux uses both MDS and ERA-I. In addition, OzFlux also includes data from the Australian Weather Service (AWS) and for each gap it utilizes either ERA-I or AWS, depending on which dataset has the smallest error in a time window of 90 days around the gap.

Potential for further development I identify two possible directions to improve the accuracy of the imputation: 1) make a better use of temporal autocorrelation of the variables 2) combine different imputation approaches in one prediction.

Temporal autocorrelation MDS uses the temporal autocorrelation only in a limited way, as it takes the average of the missing variable across the whole time window and does not weight the data depending on the proximity to the gap. Similarly, the bias correction in ERA-I uses the entire dataset from a site, thus more importance is not assigned to the conditions around the gap. This is a suboptimal use of available data, as the observation close to the gap have the highest correlation with the data in the gap and the meteorological have an overall high temporal autocorrelation. This is particularly relevant for short and medium gaps (shorter 1 week), which are the majority in the EC context. In FLUXNET 2015 [34], the most extensive EC dataset with over 200 sites, almost 99 % of gaps of meteorological variables are shorter than a week (Appendix figure 17).

Combination of imputation approaches ONEFlux employs both ERA-I and MDS, but the two methods are used independently, not combined to improve the predictions. The criteria to select the method to use is only the MDS quality control flags. The information on the missing data from temporal autocorrelation, correlation with other variables and other measurements can be combined to make one more accurate prediction.

Uncertainty A limitation of the current methods is the lack of a robust assessment of the uncertainty of the imputed values. MDS has a quality flag, but it limited to only 3 possible values and it derives from constant thresholds. Moreover, in the final ONEFlux product, the quality flag indicates only which gap filling method has been used. Ideally, each predicted data point has an associated uncertainty, which varies continuously and it is interpretable, with the same physical units of the variable. In this way, the level of confidence of the model in each prediction is available to the data user. The uncertainty can be used either to discard the data above a custom threshold, which can change depending on the application, or directly included in the downstream calculations.

Kalman Filter selection For this work, I focused on methods that combine all three imputation approaches and include interpretable uncertainty. Probabilistic machine learning algorithms are particularly suited, as they directly provide an interpretability uncertainty. Gaussian Processes (GP) are one of the most important probabilistic algorithms [21]. GP can model interactions between all data points, for example they can consider both a yearly and a daily pattern in the data. This, however, leads to their main drawback: the computation cost scales cubically with the number of observations, making the use of GP computationally prohibitive. To overcome this limitation several approximations, such as sparse GP, have been developed. The Kalman Filter (KF) can be viewed as a special kind of GP, which models the time at discrete steps and where all the information about past and future observations is stored in a latent state. This drastically improves the computation efficiency, which scales linearly in the number of observations, but limits the ability to model processes with long time scales. However, in the context of EC meteorological imputation, this is an acceptable tradeoff as the majority of gaps are not long. Another advantage of the KF is the ability to include the ERA-I data in the predictions.

I further evaluated more advanced approaches like Gaussian Processes Variational AutoEncoders [17], which combine deep learning and GP or Neural Processes [19], which approximate a GP using

a neural network. However, those methods were not tested as they are significantly more complex than a Kalman Filter, which still fulfils all the requirements for this application.

The aim of this work is to develop and test an imputation method for meteorological time series in the context of EC that employs a Kalman Filter, as it promises more accurate predictions through a more efficient use of temporal autocorrelation and the inclusion of ERA-I data. Moreover, the KF provides probabilistic predictions. The imputation performance of the Kalman Filter is evaluated by comparing it with the state-of-the-art methods (ERA-I and MDS). Then the aspects that affect the performance of the KF are assessed: the impact of the length of the gap, the advantage of including ERA-I data, the importance of inter-variable correlation and different training scenarios. For this initial implementation trial, only data from one EC site, Hainich (Germany), will be used.

2 Methods

2.1 Kalman Filter Theory

Kalman Filter models over time a latent variable x , that represent the state of the system. The state cannot be directly observed, but you can observe meteorological variables y that reflect the state of the system. Kalman Filter is a probabilist machine learning algorithm, so it keeps track of the entire distribution of the latent state $p(x_t)$ [21]. The KF can update the state also when there are missing observations and then the continuous state are hence available for all time steps, which can be used to then predict the missing data points.

In order to model the state over time, assumptions on the behaviour of the system are made. The first element is to model the time as a discrete variable. Then there are three key assumptions
1) the states are connected by a Markov chain, which means that the state at time t depends only on the state at time $t - 1$ and not the states at previous times $p(x_t|x_{t-1}) = p(x_t|x_{t-1}, x_{t-2}, \dots, x_0)$
2) The value of the observed variable depends on the latent state
3) all the relationships are linear and all distributions are Gaussian. Additionally, the mean of the state at time t may depend also on an external control variable c . This control variable does not depend on the state of the models, but provide information on the change of the state mean. Equations 1 and 2 describe the assumptions on the behaviour of the system:

$$p(x_t|x_{t-1}) = \mathcal{N}(x_t; Ax_{t-1} + b + Bc, Q) \quad (1)$$

$$p(y_t|x_t) = \mathcal{N}(y_t; Hx_t + d, R) \quad (2)$$

The probability distributions of the state are computed using Bayesian inference. The computational cost of probabilistic inference is drastically reduced in this context, as can be performed using only linear algebra operations since all the relations are linear and all distributions are Gaussian.

Kalman Filter is a recursive algorithm (Figure 1), at time t the *predicted state* (x_t^-) is obtained from the previous state (x_{t-1}) and the *control variable* (c_t). Then the state is updated using the *observation* (y_t) to obtain the *filtered state*, observations can be partially or totally missing. This is repeated recursively for all time steps. At this point, at each time step, the state x_t depends only on the observations until time t . The *smoothed state* x_t^s is the final state that depends also on all the observations after time t . The smoothing phase works by starting from the last time step and

recursively updating x_t^s using x_{t+1}^s . Finally, for all the time steps where there is a gap, the *predicted observations*, \hat{y}_t^g , are calculated from the state x_t .

The model always considers the entire probability distribution for the state $p(x_t) = \mathcal{N}(x_t; m_t, P_t)$, so stores for each state at each time step the mean (m_t) and the covariance are (P_t). Similarly, the model predictions are a multivariate Gaussian distribution $p(\hat{y}_t) = \mathcal{N}(\hat{y}_t; \mu_{y_t}, \Sigma_{y_t})$.

2.1.1 Time update

The first step in a Kalman Filter is computing the probability distribution of the predicted state x_t^- , from the state at the previous time step x_{t-1} and the control variable c_t . The predicted state distribution is $p(x_{t-1}) = \mathcal{N}(m_{t-1}, P_{t-1})$. Using equation 1 and the properties of a linear map of Gaussian distributions the following equation can be derived:

$$p(x_t^-) = \mathcal{N}(x_t^-; m_t^-, P_t^-) \quad (3)$$

$$m_t^- = Am_{t-1} + Bc_t + d \quad (4)$$

$$P_t^- = AP_{t-1}A^T + Q \quad (5)$$

2.1.2 Measurement update

The predicted state probability distribution is the updated to obtain the distribution of the filtered state, using the current observation y_t . Equation 2 describes the distribution of y_t given x_t , using Bayes theorem it is possible to compute the distribution of x_t given an observation y_t :

$$p(x_t|y_t) = \mathcal{N}(x_t; m_t, P_t) \quad (6)$$

$$z_t = Hm_t^- + d \quad (7)$$

$$S_t = HP_t^-H^T + R \quad (8)$$

$$K_t = P_t^-H^TS_t^{-1} \quad (9)$$

$$m_t = m_t^- + K_t(y_t - z_t) \quad (10)$$

$$P_t = (I - K_tH)P_t^- \quad (11)$$

Missing observations The Kalman Filter is robust to missing data and can update the state even though there is missing data. If all the observations at time t are missing, the measurement update step is skipped and the filtered (x_t) is the same of the predicted state (x_t^-). If only some observations in y_t are missing, then a partial measurement step is performed. The vector containing the observations that are not missing at time t , y_t^{ng} , can be expressed as a linear transformation of y_t

$$y_t^{ng} = My_t \quad (12)$$

where M is a mask matrix that is used to select the subset of y_t that is observed. $M \in \mathbb{R}^{n^{ng} \times n}$ and is made of rows which are made of all zeros but for an entry 1 at column corresponding to the index of the non-missing observation.

For example, if $y_t = [y_{0,t}, y_{1,t}, y_{2,t}]^T$ and $y_{0,t}$ is the missing observation then

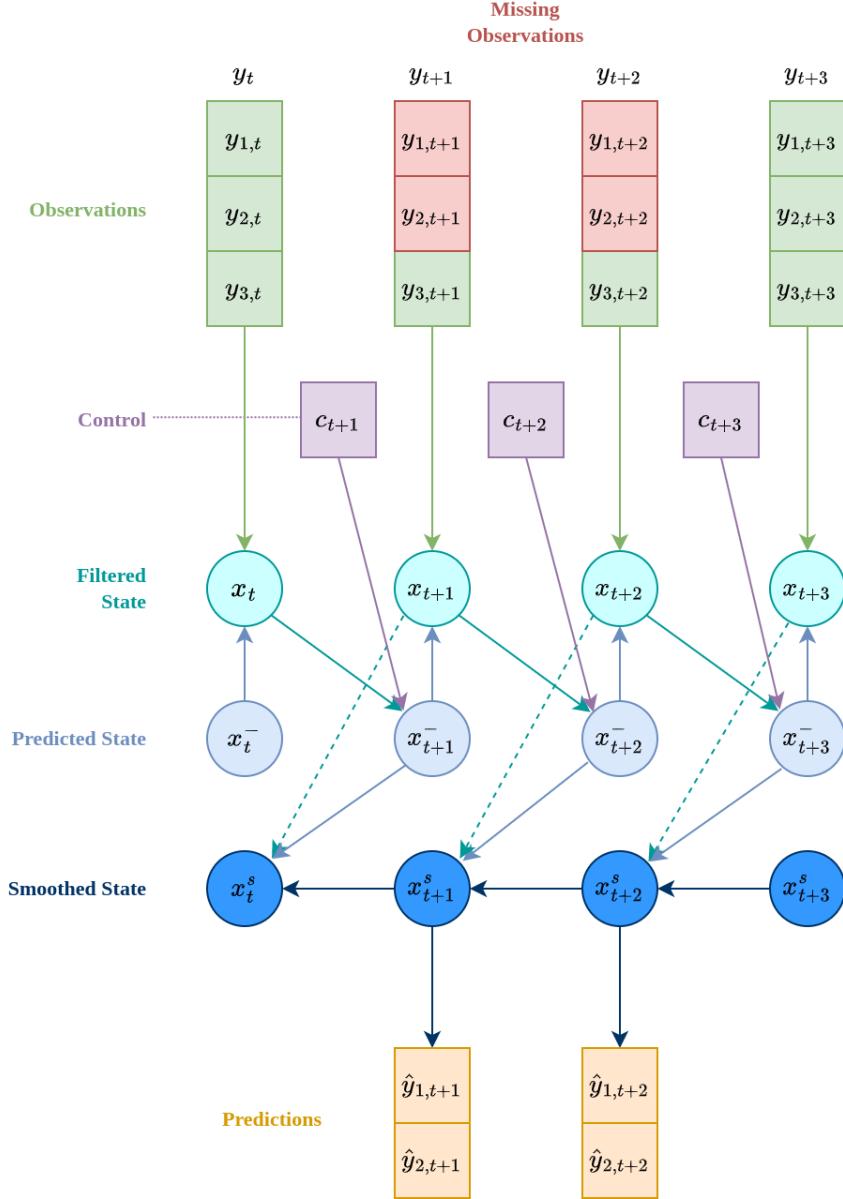


Figure 1: Schematic representation of an example Kalman Filter. The green square represent the observations of a single variable at a specific time, the observations may be missing (red square). The blue circles represent the latent state, specifically the three version of the state modelled by the KF: filtered state (cyan), predicted state (light blue) and smoother state (dark blue). The control variable are shown in purple. All the arrows show a direct dependency between elements of the figure. The figure display state and the predictions similarly to the observations, but the former are random variables with a probability distributions while the latter are single values.

$$M = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (13)$$

using the properties of linear projections of Gaussian distribution we can then derive the distribution $p(y_t^{ng} | y_t)$ and from it $p(y_t^{ng} | x_t)$

$$p(y_t^{ng} | y_t) = \mathcal{N}(y_t^{ng}; M\mu_{y_t}, M\Sigma_{y_t}M^T) \quad (14)$$

$$p(y_t^{ng} | x_t) = \mathcal{N}(y_t^{ng}; MHx_t + Mb, MRM^T) \quad (15)$$

Therefore, it is possible to perform the measurement update step when some observations are missing using a variation of equation 6, where H is replaced by MH , b by Mb and R by MRM^T .

2.1.3 Smoothing

In the smoothing step, the filtered state at time t is updated using the smoothed state $t+1$. A set of equations for the smoothing pass of a Kalman Filter has been derived by Rauch-Tung-Striebel [36]. They calculate the smoothed state x_t^s from the smoothed, filtered and predicted state at the successive time step. For the last time step, the smoothed state is set to be equal to the filtered state.

$$p(x_t^s | Y) = \mathcal{N}(x_t^s; m_t^s, P_t^s) \quad (16)$$

$$G_t = P_t A^T (P_{t+1}^-)^{-1} \quad (17)$$

$$m_t^s = m_t + G_t(m_{t+1}^s - m_{t+1}^-) \quad (18)$$

$$P_t^s = P_t + G_t(P_{t+1}^s - P_{t+1}^-)G_t^T \quad (19)$$

2.1.4 Predictions

From the state (x_t) it is possible to directly obtain the predictions of the model \hat{y}_t^g by using equation 2 and a mask, define like in equation 12

$$p(\hat{y}_t^g) = \mathcal{N}(\hat{y}_t^g; \mu_{y_t}, \Sigma_{y_t}) \quad (20)$$

$$\mu_{y_t} = MHx_t + Md \quad (21)$$

$$\Sigma_{y_t} = MRM + MHP_t^s H^T M^T \quad (22)$$

2.2 Kalman Filter Implementation

2.2.1 Requirements

Kalman Filter is a widely used algorithm and there are several python libraries that implement it (e.g. `statsmodels`, `pykalman`, `filterpy`). However, no Kalman Filter library was identified which meets all the requirements for this work. It is necessary to support gaps, partial measurements updates, control variables and be a numerically stable implementation. Therefore, a custom library for Kalman Filters was developed using the PyTorch library, which has the advantage of automatic differentiation, possibility to use GPUs and better integration with other Machine Learning methods.

2.2.2 Numerical stability

Background The direct implementation of the Kalman filter equations suffers of numerically stability issues [29, 12]. Numerical instability arises from the fact that digital computers store numbers with only a limited number of decimal digits. This results in a loss of information, so that some operations may be incorrectly performed by a computer (e.g. summing a big number and a small number).

For Kalman Filter the components that are most affect by numerical instability are the covariance matrices. To analyse the stability of the operations on these matrices it is relevant to consider the condition number for inversion [29, 25], which describes if the matrix is going to be singular on the numerical representation in the computer. The condition number $k(A)$ is the ratio between the biggest singular value and the smallest. The singular value is $\sigma^2(A) = \lambda(AA^T)$, with $\lambda(A)$ being the eigenvalue of A .

$$k(A) = \frac{\sigma_{\max}(A)}{\sigma_{\min}(A)} \quad (23)$$

The condition number it's 1 for well-conditioned matrices, and tends to infinite for ill-conditioned matrices. As a general rule, a matrix cannot be inverted when the reciprocal of the condition number for inversion is close to the machine precision $1/k(A) < \varepsilon$ [29].

Mitigation strategies

Machine precision The simplest to improve the numerical stability is to use higher accuracy in the representation of numbers [12]. Practically, this means to use 64bit floats instead of 32bit floats, which is default in PyTorch.

Matrix decomposition Another way to improve the numerical stability is to reduce the condition number of the state covariance (P). A positive definite matrix has a square root factor, $P^{1/2}$, such as that $P = P^{1/2}(P^{1/2})^T = P^{1/2}P^{T/2}$. The Cholesky decomposition is an algorithm to find a square root of a matrix, however the Cholesky decomposition calculates only one of possibly many square roots of the matrix.

Utilizing $P^{1/2}$ instead of P doubles the effective numerical resolution of the filter [25] [12] [38]. This is due to the fact that the eigenvalues of $P^{1/2}$ are the square root of the eigenvalues of P , $\lambda(P) = \lambda^2(P^{1/2})$, thus the conditioning number of P is the square of the conditioning number of $P^{1/2}$. Therefore, if in the filter implementation P is never explicitly computed, the numerical stability of the filter is significantly improved. There are several implementations of a Kalman Filter that follow this approach ([35], [9], [4]) and are generally called “square-root” filter.

2.2.3 Implementation in PyTorch

There are different approaches to square root filtering. According to [29] the best approach is the *UD* Filter ([4]), since it has the smallest computational cost. However, the filter is based on the *UD* factorization and a custom matrix factorization [29] and both of those algorithms cannot be efficiently implemented in PyTorch. The PyTorch function `torch.linalg.ldl_factor` performs an *UD* factorization, but it's an experimental function and is not differentiable. Moreover, the custom matrix factorization would need to be implemented using scalar operations, which aren't efficient with PyTorch eager execution.

For this reason, a square root filter that propagates square roots of the covariance matrices is implemented. In this way, all the required computations can be expressed in QR factorization, which is a numerically stable method and is a routine implemented in PyTorch.

2.2.4 Time update Square Root Filter

From the equations of the time update step (eq. 3) is possible to derive an algorithm to obtain $P_t^{1/2}$ given $P_{t-1}^{1/2}$, without explicitly computing P_t or P_{t-1} . The equations here described are from [29] (eq. 6.60):

Defining

$$W = \begin{bmatrix} AP_{t-1}^{1/2} & Q^{1/2} \end{bmatrix} \quad (24)$$

from equation 3 the following is true:

$$WW^T = P_t \quad (25)$$

$$WW^T = \begin{bmatrix} AP_{t-1}^{1/2} & Q^{1/2} \end{bmatrix} \begin{bmatrix} P_{t-1}^{T/2} A^T \\ Q^{T/2} \end{bmatrix} = AP_{t-1}^{1/2} P_{t-1}^{T/2} A^T + Q^{1/2} Q^{T/2} = AP_{t-1} A^T + Q = P_t \quad (26)$$

The next step is to factorize $W = LU$, where L is a lower triangular matrix and U is an orthogonal matrix, such as that $UU^T = I$. Then $WW^T = LU(LU)^T = LUU^T L^T = LL^T = P_t$. Hence, L is a square root of P_t .

This procedure never explicitly compute P_t and requires only the factorization of a matrix, which is implemented efficiently and in a numerical stable way in the PyTorch `torch.linalg.qr` function.

PyTorch implementation PyTorch doesn't support natively a LU decompositions. It implements the QR factorization: $W = QR$, where Q is an orthogonal matrix and R an upper triangular matrix. This can be easily converted into a LU factorization, as by factorizing W^T then $W^T = QR = (QR)^T = R^T Q^T$ and R^T is a lower triangular matrix.

Summary The steps of the Square Root time update are:

1. let $W = \begin{bmatrix} AP_{t-1}^{1/2} & Q^{1/2} \end{bmatrix}$
2. do a QR factorization $W^T = TR$
3. set $P_t^{1/2} = R^T$

2.2.5 Measurement update Square Root Filter

A similar procedure can be followed for the measurement update step of the filter. The equations here described are from [12].

The starting point is equation 6, for simplicity the time subscripts are omitted in the following equations.

Defining:

$$M = \begin{bmatrix} R^{1/2} & H(P^-)^{1/2} \\ 0 & (P^-)^{1/2} \end{bmatrix} \quad (27)$$

$$V = \begin{bmatrix} S^{1/2} & 0 \\ \bar{K} & P^{1/2} \end{bmatrix} \quad (28)$$

$$\bar{K} = KS^{1/2} \quad (29)$$

the following is true:

$$MM^T = VV^T \quad (30)$$

$$\begin{aligned} MM^T &= \begin{bmatrix} R^{1/2} & HP^- \\ 0 & (P^-)^{1/2} \end{bmatrix} \begin{bmatrix} R^{T/2} & 0 \\ (P^-)^{T/2}H^T & (P^-)^{T/2} \end{bmatrix} = \\ &= \begin{bmatrix} R^{1/2}R^{T/2} + H(P^-)^{1/2}(P^-)^{T/2}H^T & HP^{-1/2}P^{-T/2} \\ (P^-)^{T/2}P^{-1/2}H^T & (P^-)^{1/2}P^{-T/2} \end{bmatrix} = \\ &= \begin{bmatrix} S & HP^- \\ (P^-)^TH^T & P^- \end{bmatrix} \end{aligned} \quad (31)$$

$$\begin{aligned} VV^T &= \begin{bmatrix} S^{1/2} & 0 \\ \bar{K} & P^{1/2} \end{bmatrix} \begin{bmatrix} S^{T/2} & \bar{K}^T \\ 0 & P^{T/2} \end{bmatrix} = \begin{bmatrix} S^{1/2}S^{T/2} & S^{1/2}\bar{K}^T \\ \bar{K}S^{T/2} & \bar{K}\bar{K}^T + P^{1/2}P^{T/2} \end{bmatrix} \\ &= \begin{bmatrix} S & S^{1/2}S^{T/2}\bar{K}^T \\ KS^{1/2}S^{T/2} & KS^{1/2}S^{T/2}\bar{K}^T + P \end{bmatrix} \\ &= \begin{bmatrix} S & HP^- \\ P^-H^T & KHP + P \end{bmatrix} \end{aligned}$$

All blocks of VV^T are directly equal to MM^T , but the bottom left one, which is equal due to the measurement update for the covariance (equation 11).

Therefore, if we decompose $M = LU$ then $MM^T = LL^T = VV^T$ and the bottom left block of U of size $k \times k$ of L is a square root of P , where k is the number of dimensions of the state $x_t \in \mathbb{R}^k$.

Summary The steps of the Square Root measurement update are:

1. let $M = \begin{bmatrix} R^{1/2} & H(P^-)^{1/2} \\ 0 & (P^-)^{1/2} \end{bmatrix}$
2. do a QR factorization of $M^T = TU$
3. $P^{1/2}$ is the bottom left $k \times k$ block of U

2.2.6 Predictions Square Root Filter

The prediction equation for the square root filter are similar to the equations for the time update. defining:

$$W = \begin{bmatrix} HP_t^{1/2} & R^{1/2} \end{bmatrix} \quad (32)$$

from equation 20 the following is true:

$$WW^T = \Sigma_{y_t} \quad (33)$$

$$WW^T = \begin{bmatrix} HP_t^{1/2} & R^{1/2} \end{bmatrix} \begin{bmatrix} P_t^{T/2} H^T & R^{T/2} \end{bmatrix} = HP_t^{1/2} P_t^{T/2} H^T + R^{1/2} R^{T/2} = HP_t H^T + R = \Sigma_{y_t} \quad (34)$$

Summary The steps of the Square Root predictions are:

1. let $W = \begin{bmatrix} HP_t^{1/2} & R^{1/2} \end{bmatrix}$
2. do a QR factorization of $W^T = TU$
3. set $\Sigma_{y_t}^{1/2} = U^T$

2.2.7 Smoothing Square Root Filter

The available literature for implementing square root smoother is scarce compared to square root filter, so no solution has been identified to implement a square root smoother. Therefore, a standard smoother is employed.

Nonetheless, steps were taken to improve the numerical stability of the smoother. The computation in the smoother that is most numerically unstable is the inversion of P_{t+1}^- in equation 17 [29]. The matrix inversion is avoided by using the `torch.cholesky_solve` function. It solves for X the linear system $P_{t+1}^- X = P_t A$, which is equivalent of computing $X = (P_t A^T (P_{t+1}^-)^{-1})^T$. This use directly the square root $(P_{t+1}^-)^{1/2}$ to avoid the computation of P_{t+1}^- . A further step to improve the numerical stability is forcing the covariance matrix to be symmetric, by averaging to upper and lower part at after every time step $P_{t,sym}^s = (P_t^s + (P_t^s)^T)/2$, as suggested in [12]. This approach to numerical stability in the smoother is the same applied by the `statsmodels` library [40].

2.3 Kalman Filter Model

2.3.1 Parameters

The Kalman Filter is implemented as PyTorch module, whose parameters are described in Table 1. There is no change over time of the parameters, and the state of the filter is initialized always at the same value from the parameters m_0 and P_0 .

Table 1: Parameters of the Kalman Filter Model. n is the number of dimension of the observations, k the number of dimensions of the state, n_{ctr} the number of dimensions of the control variable.

Parameter name	Notation	Shape	Initial value
State transition matrix	A	$k \times k$	$\begin{bmatrix} I & I \\ 0 & I \end{bmatrix}$
Observation matrix	H	$n \times k$	$\begin{bmatrix} I & 0 \end{bmatrix}$
State transition covariance	Q	$k \times k$	$\text{diag}(0.1)$
Observation covariance	R	$n \times n$	$\text{diag}(0.01)$
State transition offset	d	k	0
Observation offset	b	n	0
Control matrix	B	$k \times n_{ctr}$	$\begin{bmatrix} -I & I \\ 0 & 0 \end{bmatrix}$
Initial state mean	m_0	k	0
Initial state covariance	P_0	$k \times k$	$\text{diag}(3)$

Constraint An important aspect for implementing a Kalman Filter in PyTorch is constraining the parameters that represents covariance (Q , R and P_0) to be positive definite. To achieve this goal the optimizer works on a raw parameter, which is then transformed into a positive definite matrix. The transformation into a positive definite matrix is done by transforming the raw parameter into a lower triangular matrix with a positive diagonal. The diagonal is enforced to be positive by transforming the diagonal of the raw parameter with the softplus function ($x = \log(1 + e^x)$), which is a positive function. In addition a small positive offset 1×10^{-5} is added to the diagonal in order to avoid that the diagonal is close to zero, which may result in a positive semi-definite matrix. The inverse of the positive definite transformation is implemented, so that parameters can be manually set.

This implementation of the positive definite constraint makes it is that is straightforward to obtain the Cholesky factor of the parameters, which are needed by the Square Root Filter, and at the same time the full parameters, which are needed by the smoother.

2.3.2 Parameters initialization

The model parameters could be initialized using random values, however this would increase numerical stability issues and increase the training time. Moreover, if the initial parameters are very distant from the optimal ones, it is more likely for the optimization algorithm to find only a local minimum. The simplicity of the Kalman Filter and the interpretability of its parameters allows to manually initialize the parameters with realist values.

State transition matrix A is initialized using a “local linear trend” model [15]. The idea is that half of the state $x_{l_{t-1}}$ represent the level of the current state and the second half the slope $x_{s_{t-1}}$ of a linear function that describes the rate of change of the state between time steps. The next state level is equal to the current level plus the slope and a random noise, while the slope remains constant but for another random noise. The use of a slope allows the model to retain information of several previous states.

Observation Matrix H is initialized with an identity matrix. This means that each observed variable is modelled by one variable in the state. The second part of H is 0 as in a local trend model the observations don't depend on the slope but only the level of the state.

Control matrix B is initialized to the difference between the previous observation and the current observation. The number of variables in the control may be different than the observed variables. In this initialization, the assumption is that $n_{ctr} < n$ and that there is a correspondence between the control and first n_{ctr} variables of the observations and hence of the state.

Covariances The state transition covariance Q and then observation covariance R are initialized as diagonal matrix with values of 0.1 and 0.01 respectively. This number has been chosen to represent an uncertainty in the state transition that is compatible with the standard deviation of the variables (1 as they are standardized) and a low uncertainty in the observations.

Offsets The observation and state transition offsets are initialized to zero.

Initial state The initial state is set to have as mean zero and as covariance $\text{diag}(3)$. The number 3 is an arbitrary number bigger than the state transition covariance, which should represent the high level of uncertainty for the initial state.

2.3.3 Loss Function

The loss function is used to train the model is the negative log likelihood, computed for each data point. At each time step, the model predicts a multivariate normal distribution $p(\hat{y}_t^g)$, which is used to compute the negative log likelihood given the actual observations y_t^g . The negative log likelihoods between different time steps in the same gap are summed. Then negative log likelihood is averaged between batches.

The actual loss function of the model should be the log likelihood of the joint distribution $p(Y^g)$. However, the analytical form of the joint distribution cannot to easily derived from the Kalman Filter equations. The log likelihood of marginal distributions is instead used, as it is a lower bound to the log likelihood of the joint distribution. Defining $q(x)$ the predicted joint distribution, $p(x)$ the real joint distribution and $q_i(x)$ the marginal distribution at the i th time step:

$$q_i(x) = \int q(x_1, \dots, x_k) dx_{\neg i} \quad (35)$$

Then, if the family of distribution of $q(x | \theta)$, is the same of $\prod_i q_i(x | \theta)$, where θ are the model parameters. Then

$$\max_{\theta} \langle \log q(x | \theta) \rangle_{x \sim p(x)} \geq \max_{\theta} \langle \log \prod_i q_i(x | \theta) \rangle_{x \sim p(x)} \quad (36)$$

because $\prod_i q_i(x)$ is more restricted. This means that $q(x)$ fit at least as good as $\prod_i q_i(x)$. For the Kalman Filter $q_i(x)$ is a Gaussian distribution, so $\prod_i q_i(x)$ is also a Gaussian distribution and equation 36 is true.

2.3.4 Metrics

The main metric used to assess the model performance is the *Root Mean Square Error* (RMSE).

$$\text{RMSE} = \sqrt{\frac{\sum_i^n (y_i^g - \hat{y}_i^g)^2}{n}} \quad (37)$$

The advantage of the RMSE is that it can be used also for non-probabilistic methods (e.g. MDS) and that its value has the same physical dimension as the observed variable. The main drawback is that it cannot be used for comparison between variables. For that, the *Standardized* RMSE is used, which is the RMSE computed on the standardized variables.

$$\text{RMSE}_{\text{stand}} = \frac{\text{RMSE}}{\sigma_Y} \quad (38)$$

Other metrics, like the R^2 score and the mean absolute percentage error were evaluated, however none of them are suitable for this application. The R^2 is defined as $R^2 = 1 - (\sum_i^n (y_i - \hat{y}_i)^2) / (\sum_i^n (y_i - \bar{y})^2)$, if the denominator is close to zero, then value of R^2 tends to $-\infty$. Since the gaps are often short and several variables are constant over short periods (e.g. SW_IN, SWC) the denominator of the R^2 would be close to zero and the metrics cannot be effectively used. The mean absolute percentage error is defined as $\text{MAPE} = \frac{1}{n} \sum_{i=0}^{n-1} (|y_i - \hat{y}_i|) / (|y_i|)$, which tends to ∞ when y_i tends to 0, as zero is a possible value of several variables (e.g. SW_IN, TA) this metric cannot be employed. It would be possible to use R^2 or MAPE, for subset of variables and gap lengths, but this limits the ability to perform comparison across different settings.

2.3.5 Performance considerations

The iterative nature of the filter, where the current state depends on the previous state, makes it impossible to use PyTorch vectorization across different time steps. This can significantly limit the performance of the filter, especially when executed on GPUs. In order to mitigate this issue, all functions in the Kalman Filter library support batches, so at every time step different data is processed in parallel.

2.4 Data

2.4.1 Data source

The data used to evaluate the performance of Kalman Filters is from the Hainich (Germany) site. The EC site in Hainich (DE-Hai) is on a deciduous beech forest and it managed by the bioclimatology department at the University of Göttingen. The source of the data is the FLUXNET 2015 Dataset [34], which for Hainich includes measurements with a 30 mins frequency between 2000 and 2012. In total 227952 observations are available. For simplicity, the entire dataset was used for the model training, which includes also gap-filled observations.

All the meteorological variables that are gap-filled in the FLUXNET 2015 dataset were selected for the analysis (Table 2).

ERA-Interim The FLUXNET 2015 dataset also includes the ERA-I dataset for each site. The data is bias corrected and temporally downsampled. The ERA-I data is used as the control variable for the Kalman Filter. All the variables of interest are present in ERA-I, except for TS and SWC.

Table 2: Meteorological variables used to evaluate the Kalman Filter imputation. ERA-I column indicates whether the variable is available in the ERA-Interim dataset.

Variable Name	Abbreviation	Unit	ERA-I
Air Temperature	TA	°C	✓
Incoming Shortwave Radiation	SW_IN	W/m ²	✓
Incoming Longwave Radiation	LW_IN	W/m ²	✓
Vapour Pressure Deficit	VPD	hPa	✓
Wind Speed	WS	m/s	✓
Air Pressure	PA	hPa	✓
Precipitation	P	mm	✓
Soil Temperature	TS	°C	✗
Soil Water Content	SWC	%	✗

2.4.2 Data preparation pipeline

The dataset needs to be pre-processed by dividing into data blocks, adding an artificial gap and then standardize. The data preparation pipeline takes as input a list of items and outputs the data in a format suitable for training. Each item provides all the information about a gap with the following fields a) `i` the index of the block b) `shift` the offset of the data block c) `var_sel` the variables in the gap d) `gap_len` the gap length. The pipeline perform the following steps: 1) split the index of complete data frame from Hainich into blocks of a given length and selects the i th element 2) adds the shift to move the starting point of the data block and select the data from the data frame. For the control variable it also adds the observations with a lag 1, so that at the time t the model has access to the control variable both at time t and $t - 1$ 3) creates one continuous artificial gap in the middle of the block for the variables specified in `var_sel` and with a length of `gap_len` 4) convert from Pandas data frame to a PyTorch Tensor 5) Standardize each variable, using the mean (μ_Y) and standard deviation (σ_Y) of the whole dataset.

$$y_t^z = \frac{(y_t - \mu_Y)}{\sigma_Y} \quad (39)$$

After this, the tensors are collated into a batch and potentially moved to the GPU.

2.4.3 Prediction pipeline

The model predicts the mean and the covariance for each time steps for the standardized variables. This needs to be converted back to be scale of the variable to be used for imputation. This operation needs to scale the whole distributions and not only the mean of the prediction. The standardized prediction \hat{y}_t^z is distributed $p(\hat{y}_t^z) = \mathcal{N}(\hat{y}_t^z; \mu_{\hat{y}_t^z}, \Sigma_{\hat{y}_t^z})$, the prediction in the original scale \hat{y}_t is distributed $p(\hat{y}_t) = \mathcal{N}(\hat{y}_t; \mu_{\hat{y}_t}, \Sigma_{\hat{y}_t})$ and $\Sigma_Y = \text{diag}(\sigma_Y)$.

Then from the inverse of equation 39

$$\hat{y}_t = \Sigma_Y \hat{y}_t^z + \mu_Y \quad (40)$$

Then using the properties of the linear projections of Gaussian distributions

$$p(\hat{y}_t) = \mathcal{N}(\hat{y}_t; \Sigma_Y \mu_{\hat{y}_t}^z + \mu_Y, \Sigma_Y \Sigma_{\hat{y}_t}^z \Sigma_Y^T) \quad (41)$$

2.5 Model Training

The available data is split between training and validation set, the first 80% of the data points used for training, the remaining 20% for validation. The split is not random, so the validation set doesn't contain periods of time close to the one used for training.

The filter is initialized with 9 dimensions for the observations (one for each variable). For the state 18 dimensions, the first 9 dimensions are to match the number of observed variables and the other 9 are for the slope in the local trend model. The control has 14 dimensions, with 7 for the control at time t and the other 7 for the control at time $t - 1$.

The model is trained using gradient descend, by minimizing the loss function. It employs the ADAM optimizer [kingma adam 2017].

Several versions of the Kalman Filter are trained using data with different patterns in the gaps. Figure 2 shows the different combinations. Each model version has a name that reflects the training conditions, that follows this pattern: *KF-<var missing>-<n var missing>-<range gap lengths>/-<modifier>*.

Generic model The first model to be trained is a generic model (**KF-Gen-Sin-6.336**), where each data block has a gap in one variable with the length of sampled from a uniform distribution between 6 (3 hours) and 336 (1 week). For each gap only one variable is missing, which is sampled with equal probability from the list of all variables. The shift is sampled from a normal distribution with mean 0 and standard deviation 50. For each block of data in the original data frame, 10 different artificial gaps were created, resulting in a total of 4080 data blocks used for training and 520 for validation. The length of the block of data is 446, so that at least 50 observations are available to the model before and after the gap. The batch size is 20. The model was trained for 3 epochs with a learning rate of 1×10^{-3} .

Variable fine-tuning The generic model has been fine tune for each variable (**KF-<var>-Sin-6.336**), resulting in 9 different models. The training settings are the same for the generic model, expect that the gaps is only in one variable and then number of repetitions for each is 5 (training 2040 blocks validation 260 blocks). Each variable, was fine tuned with a different combination of epochs and learning rates (lr): TA 3 epochs with lr 1×10^{-3} and 1 epoch with lr 1×10^{-5} , SW_IN 7 epochs with lr 1×10^{-3} , LW_N3 epochs with lr 1×10^{-3} , VPD 6 epochs with lr 1×10^{-3} , WS 3 epochs with lr 1×10^{-3} , PA 3 epochs with lr 1×10^{-3} and 1 epoch with lr 1×10^{-5} , P no additional training, TS 6 epochs, SWC 8 epochs with lr 1×10^{-3} and 1 epoch with lr 1×10^{-5} . The learning was manually stopped when the training loss started being constant or the validation loss started to increase.

Short gaps The numerical stability issue limits the gap length to 30 observations (15 hours) if all variables are missing. Therefore, an additional set of models has been trained with gaps in multiple variables.

Gap for all variables A version of the model was trained only with gaps in all variables (**KF-Gen-All-6-30**). The length of the gap ranges from 6 to 30 (3 to 15 hours). The training set contains, 2040 unique data block and 260 for validation. The training started from the generic model and lasted for 3 epochs with a learning rate of 3×10^{-4} .

Single gap The generic model has been fine tune short gaps (**KF-Gen-Sin-6-30**). The was training for 3 epochs with a learning rate of 3×10^{-4} .

Gap multiple variables Another version of the model has ben trained with gaps in any number of variable (**KF-Gen-Multi-6-30**). The number of variables missing has been drawn from a uniform distribution from 1 to n , and the variables missing sampled with equal probability. The total gap length ranges from 6 to 30 (3 to 15 hours). For each original data block 20 different artificial gaps where generated for a total of block 8160 in the training set and 1040 in validation. The model was trained starting from *KF-Gen-Sin-6-336* for 3 epochs with a learning rate of 5×10^{-5} and then 1 epoch with a learning rate of 1×10^{-5} .

Additional version Two more model have been trained changing the model instead of the data.

Random parameters A model has been initialized with random parameters, drawn from a uniform distribution between 0 and 1 (**KF-Gen-Multi-6-30-Rand**). The data used for training is the same of *KF-Gen-Multi-6-30*. [training is still running]

No Control The last version of the model is one where the use of the control variables was disabled (**KF-Gen-Sin-6-336-No_Contr**). The data is the same of *KF-Gen-Sin-6-336* and the training was from scratch for 3 epochs with a learning rate of 1×10^{-3} .

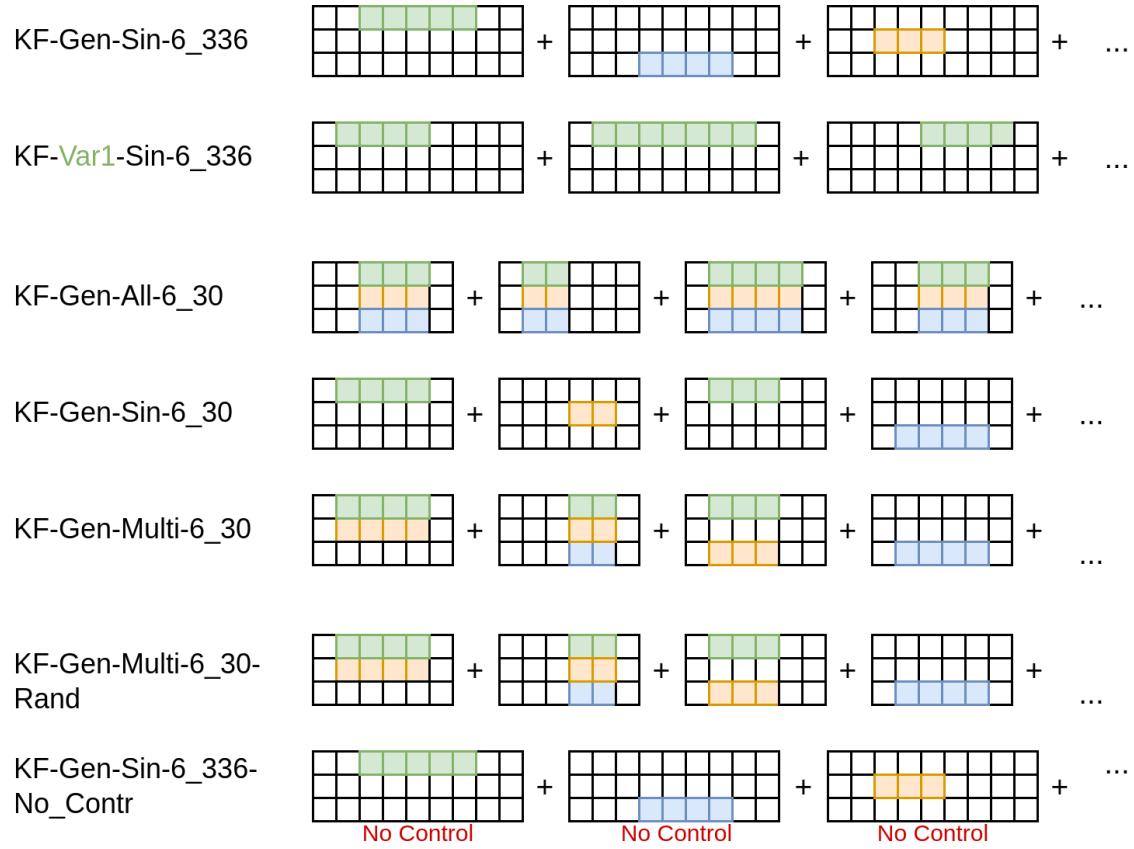


Figure 2: Schematic representation of gap pattern for training scenarios. Each rectangle is a data block used for training, where each row is a different variable and each column a different time. The highlighted areas represent artificial gaps. In the visualization only 3 variables and a small number of data points are shown.

2.6 Other methods

2.6.1 MDS

The implementation of the MDS used in the results comparison is from REddyProc ([42]). This package has been used because it provides an R interface, that can be easily integrated with Python. Conversely, ONEFlux implements only a C interface, whose integration in Python is significantly more challenging. The MDS algorithm in REddyProc and ONEFlux are fully equivalent. In detail, the function `REddyProc::sEddyProc_sMDSGapFill` was used, with the defaults setting of using SW_IN, TA an VPD are driver with a tolerance of 2.5 °C, 50 W/m^s and 5 hPa respectively as described in [reichstein's separation'2005]. The data provided to MDS has a context of at least 90 days around the gap, as required by REddyProc.

2.6.2 ERA

The imputation using ERA-Interim was performed by using the ERA variables available in the FLUXNET dataset without further correction.

2.7 Code Details and Availability

The code for this project has been developed in Python. The main libraries used are PyTorch for the model, FastAI for model training and data preparation, Altair plot plotting and Pandas and Polars for data analysis. The source code is available at https://github.com/mone27/meteo_imp and the documentation of the library at https://mone27.github.io/meteo_imp/libs. The interactive version of the results ...

3 Results

3.1 Correlation characteristics of meteorological variables

The analysis of the pattern in the variable temporal autocorrelation and inter-variable correlations supports the interpretation of the results of the imputation methods, as it highlight which mechanisms are available to the model to impute each variable

The temporal autocorrelation, for a lag up to 48 hours is shown in figure 3. Overall the meteorological variables have a high temporal autocorrelation, that decreases with the gap length. The only exception is the precipitation (P). Moreover several variables (i.e. Air Temperature TA, Incoming Shortwave Radiation SW_IN, Vapour Pressure Deficit VPD, Soil Temperature TS, Soil Water Content SWC) have a daily pattern with the temporal autocorrelation with a lag of 24 or 48 hours being higher than the one at shorter lags (eg. 12 hours). This is particularly evident in SW_IN, that has a negative correlation for a lag of 12 hours.

The variable with the highest correlation with other variable is TA, which is correlated with 5 other variables (correlation coefficient bigger 0.4). TS is highly correlated with the air temperature, thus following a similar pattern. Four variables: SW_IN LW_IN, VPD, SWC have a correlation ranging between 0.4 and 0.6 with at least two other variables, while the remaining three variables: wind speed (WS), air pressure (PA) and precipitation (P) have a low correlation with any other variable.

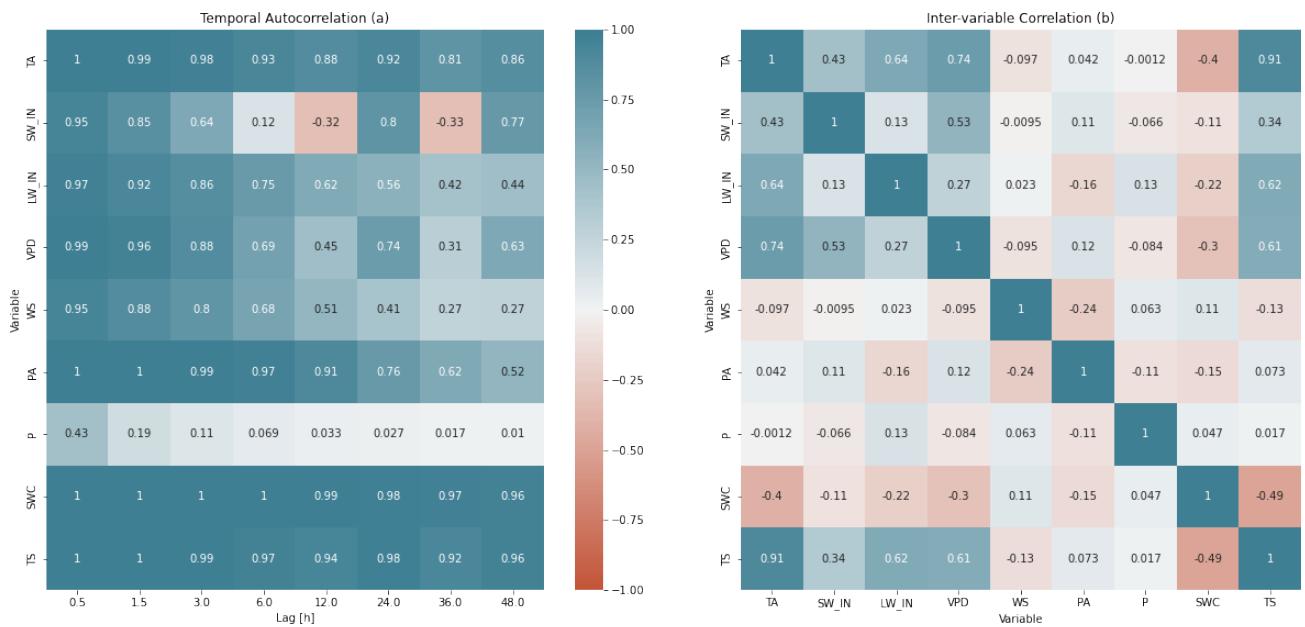


Figure 3: Temporal autocorrelation (a) and inter-variable correlation of meteorological variables. Abbreviations: Air Temperature TA, Incoming Shortwave Radiation SW_IN, Incoming Longwave Radiation LW_IN, Vapour Pressure Deficit VPD, Wind Speed WS, Air Pressure PA, Precipitation P, Soil Temperature TS, Soil Water Content SWC

3.2 Comparison to other imputation methods

The Kalman Filter (KF) has an overall better imputation performance than the other imputation methods: ERA-I and MDS. For scenarios, KF is the always best methods, while ERA-I constantly outperforms MDS (Figure 4 and Table 3). The only exception from this pattern is the precipitation, for which the three methods are roughly equivalent. I compared the models by creating artificial gaps in a single variable with four different gap lengths (i.e. 6 hours, 12 hours, 1 day, 1 week). For each combination of variable and gap length, 500 artificial gaps were generated and imputed using the 3 methods. The performance was measured using the RMSE (Figure 4 and Table 3) and the standardized RMSE (Appendix Figure 12 and Table 4). In addition, the imputation performance is compared visually using three different example time series for each variable for three gap lengths (Figures ??, ?? and Appendix figures ?? ??, ??, ??). For each variable, the fine-tuned KF model has been used (*KF-(var)-Sin-6-336* figure 2).

The average error reduction across all variable and gap length is 33% reduction of error compared to ERA-I 57% compared to MDS, if P is excluded. The improvement of the KF compared to the best model strongly depends on the variable analysed and the gap length, ranging from 54% for TA to 5% for LW_IN. Moreover, across the tested gaps the standard deviation and the maximum value of the KF RMSE is constantly smaller compared to other methods.

The variable with the highest Standardized RSME

- Overall Kalman Filter is better than state of the art
- creation artificial gaps for different lengths
- using a fine-tuned model
- information results: RMSE plot + visually time series
- numbers of average performance improvement: 33% reduction of error compared to RMSE and 57% to MDS (excluding P)
- for long gap the relative performance of Kalman Filter is lower
- Kalman Filter has a lower variability of the error. For all variables std is smaller and the max error is smaller
- the max gap length is 15 because after that the model crashed due to numerical stability
- uncertainty - doesn't vary much between variables and gaps
- Standardized RMSE
 - PA/TS/TA/SWC are the variable with the lowest error and is all comparable with standardized RMSE around .6
 - WS is the variable with worse standardized RMSE

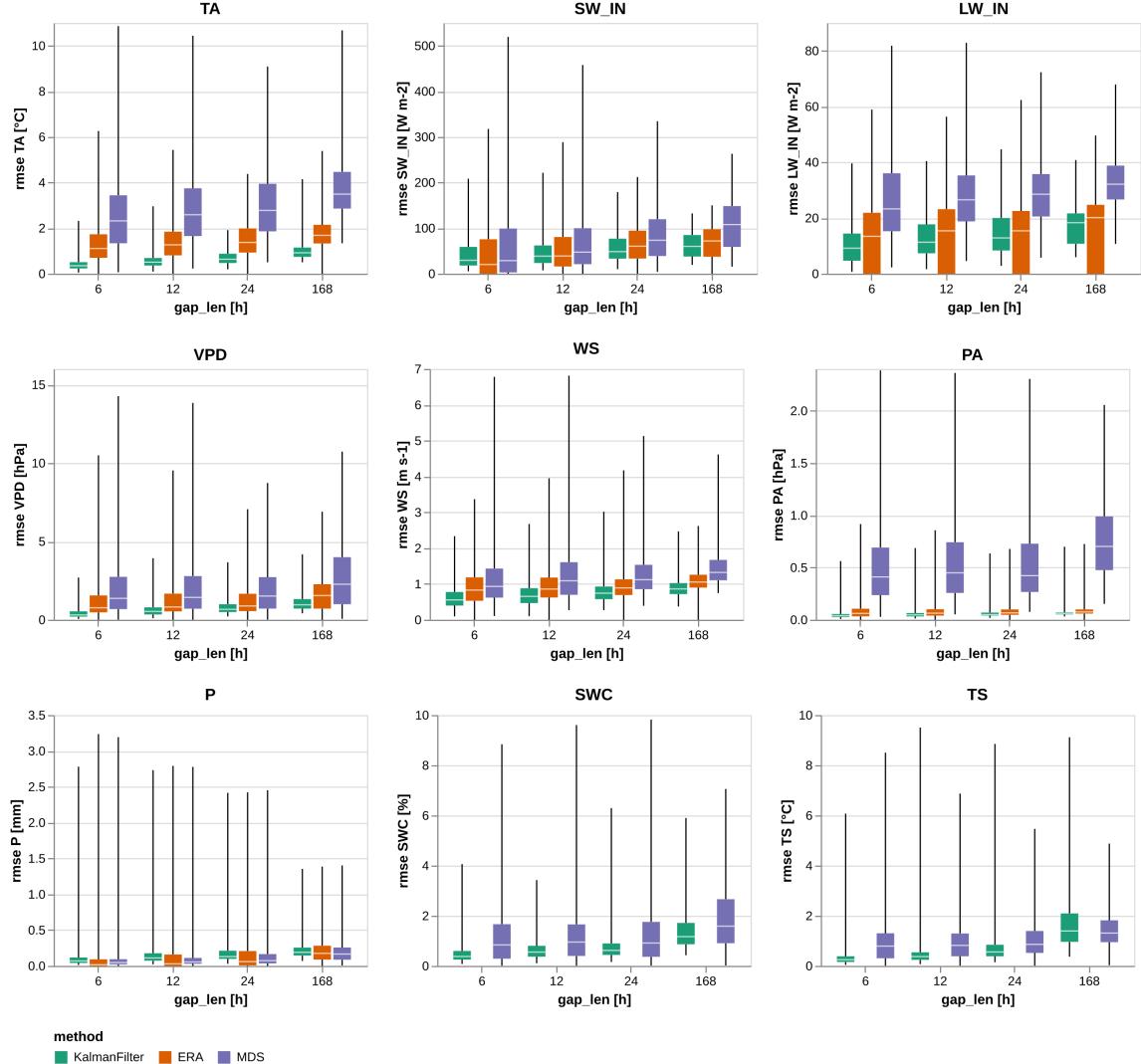


Figure 4: Imputation performance of the Kalman filter in comparison to the state of art methods: ERA-Interim (ERA-I) and Marginal Distribution Sampling (MDS) to compare. The performance was assessed calculating for each method the *Root Mean Square Error* (RMSE) for an artificial gap, with a single variable missing. For each combination of variable and gap length a sample of 500 random gaps has been used (total 18000 artificial gaps). The Kalman Filter model has been fine-tuned to each variable. ERA-I dataset doesn't contain TS and SWC so cannot be used for their imputation. The extent of the box plot vertical lines represent the maximum and minimum value.

Table 3: RMSE Comparison imputation methods. The best method for each gap length is highlighted in bold

Variable	RMSE Gap [h]	KalmanFilter		ERA		MDS	
		mean	std	mean	std	mean	std
TA [C]	6	0.405	0.258	1.347	0.998	2.713	1.897
	12	0.607	0.401	1.472	0.901	2.942	1.748
	24	0.741	0.368	1.530	0.800	3.013	1.611
	168	1.021	0.445	1.754	0.643	3.780	1.315
SW_IN [W/m ²]	6	44.637	40.465	49.333	66.242	63.537	85.402
	12	48.155	33.868	54.208	49.769	69.427	68.936
	24	56.564	30.043	65.950	40.931	86.771	59.604
	168	61.583	25.740	70.224	34.883	107.384	53.606
LW_IN [W/m ²]	6	10.902	7.736	13.805	12.988	26.680	15.022
	12	13.422	7.735	14.767	12.585	28.085	13.457
	24	14.594	7.840	14.093	12.228	29.614	12.417
	168	17.063	6.425	16.366	11.130	32.955	8.834
VPD [hPa]	6	0.428	0.363	1.297	1.547	2.084	2.149
	12	0.661	0.505	1.265	1.289	2.137	2.096
	24	0.828	0.502	1.248	1.032	1.912	1.605
	168	1.126	0.633	1.662	1.127	2.661	1.965
WS [m/s]	6	0.617	0.317	0.912	0.508	1.136	0.783
	12	0.715	0.351	0.957	0.524	1.261	0.797
	24	0.802	0.343	0.949	0.447	1.276	0.609
	168	0.950	0.363	1.089	0.349	1.495	0.615
PA [hPa]	6	0.045	0.034	0.075	0.062	0.531	0.441
	12	0.053	0.042	0.077	0.058	0.564	0.427
	24	0.059	0.039	0.079	0.051	0.557	0.404
	168	0.066	0.048	0.084	0.054	0.773	0.384
P [mm]	6	0.134	0.274	0.113	0.316	0.118	0.306
	12	0.179	0.295	0.139	0.297	0.130	0.281
	24	0.206	0.254	0.166	0.288	0.159	0.265
	168	0.240	0.174	0.223	0.202	0.215	0.197
SWC [%]	6	0.508	0.487	-	-	1.314	1.557
	12	0.665	0.472	-	-	1.278	1.323
	24	0.779	0.641	-	-	1.356	1.472
	168	1.494	0.948	-	-	1.948	1.488
TS [C]	6	0.341	0.432	-	-	0.954	0.889
	12	0.534	0.784	-	-	1.003	0.877
	24	0.787	0.852	-	-	1.078	0.857
	168	1.660	1.078	-	-	1.440	0.764

The performance of the KF for each variable is analysed in detail and compared with the other methods

Air Temperature Air temperature is the variable with the biggest improvement in performance, up to XX for 6 hours long gaps. KF outperforms ERA-I also for long gaps and the compared to MDS there is a 77% reduction in the RMSE. The visual inspection of the time series indicate an overall very good reconstruction on the missing data.

Incoming shortwave radiation The KF is the method with the smaller average RMSE, but the relative improvement is small, only 12% compared to ERA-I. The highest error is at night, where `SW_IN` is by definition always 0, but the KF often predicts sudden changes with errors in the order of 50 W/m². However, this means that during the day the KF is consistently better than other methods, which is also confirmed by visual inspections of the time series. The main factor behind the changes in `SW_IN` is the cloud cover, which ERA-I often correctly predicts and consequently the KF can access this information, while the MDS is unable to model this type of changes.

Incoming Longwave radiation The imputation performance of the KF is comparable with ERA-I. For short gaps KF is better (20 % improvement for 6 hours long gaps). This can be visualized in figure ??, where in the 12 hours long gap ERA is incorrect by an offset of roughly 50 W/m², while the KF prediction is more accurate towards the edges of the gap. KF The imputation performance of MDS is poor, especially for long gaps.

Vapour Pressure Deficit KF is the best model for all gap lengths, the relative performance is higher for short gaps (XX compared to ERA) and progressively smaller for longer gaps. The analysis of the time series suggest that the KF is overall good at reconstructing the higher frequency changes of VPD, but in some scenarios KF wrongly predicts short term variation (gap 1 week figure 13).

Wind Speed The KF is the best method imputation method for all gap lengths, with an average error reduction of 21% compared to ERA-I. The wind speed is the variable with the highest standardized RMSE (appendix figure 12), which indicates that the RMSE is high compared to the WS standard deviation. The visual inspection of the time series (figure 13 and 15), indicates that the WS has a high variability on a short time scales, which is not captured by ERA-I nor by the KF.

Air Pressure The imputation error of PA is low for KF and ERA-I. The KF outperforms ERA-I with an improvement ranging from 36% for 6 hours gaps to 20% for 1 week long gaps. The visual analysis of the time series suggest that the KF slightly overestimate the short term variability for PA. MDS is significantly worse, with the imputation error an order of magnitude bigger than ERA-I.

Precipitation The Precipitation is a variable where no methods perform well, with models in some case predicting precipitation event that do not exist and in other missing the real precipitation. The RMSE of the three methods is comparable. However, the RMSE is not a suitable metric for P, due to the very high number of data points with no precipitation. For reference, the RMSE of a null model (i.e. always predicts 0) is 0.28 mm, which is comparable with the errors of all imputation methods. The visual analysis of the time series shows that ERA-I predictions are the

one that are physically realistic, even though the precipitation amount is often incorrect, while the KF often predicts negative values for P, which is physically impossible and MDS in a tested scenario (Appendix figure 14) predicts an highly unlikely constant low amount of P.

Soil Water Content The KF is the best imputation method, for short gaps there is an error reduction of 61% compared to MDS, while for long gap (1 week) the absolute error of the KF roughly doubles and the improvement is performance is limited to 23%. SWC is a variable that is not available in ERA-I, so KF and MDS are the only available methods. Moreover, the KF does not have a control variable for SWC. The analysis of the time series shows that the mean of KF prediction is overall accurate, and notably manages also to predict sudden changes in SWC (figure 6). However, the KF constantly predict small variations in the soil water content, which are not reflected in the observations.

Soil Temperature The KF is the best imputation method for short gaps (less than 24 hours), but the MDS is better for 1 week long gaps (Figure 4). For short gaps there is a big difference in the methods error (up to 60%), but is reduced for long gaps (KF is 15% worse than MDS). In two of the long time series analysed the TS is almost constant (figure 6 and 16), but the KF incorrectly predicts important variations, while the MDS is overall constant. In another scenario (appendix figure 14), where there is a diurnal pattern in TS, the KF predictions have an overall correction shape, even though there is roughly 1 C error.

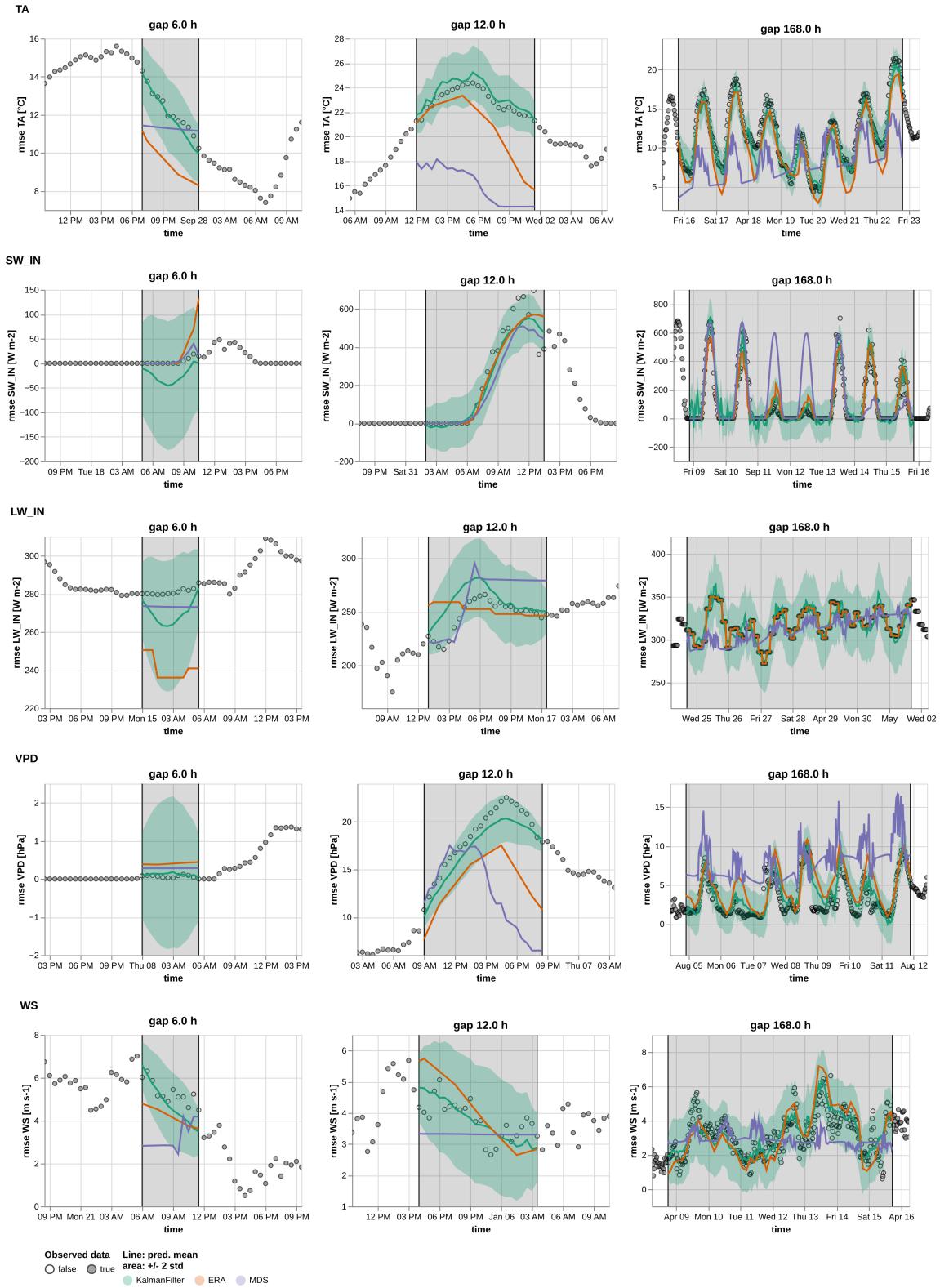


Figure 5: Illustration of the imputation TA, SW_IN, LW_IN, VPD, WS using different methods: Kalman Filter, ERA-Interim (ERA-I) and Marginal Distribution Sampling (MDS). The methods are compared by visualizing the time series of an artificial random gap for a three gap lengths. For each variable, 3 random artificial gap (length 6 hours, 12 hours, 1 week) are imputed using the three methods: Kalman Filter (green), ERA-I (orange), MDS (purple). For the Kalman Filter the shared area show the uncertainty of the prediction $\pm 2\sigma$. The grey shaded area and the vertical black lines delimit the artificial gaps, where the observations are not available to the model but are used to assess the imputation performance. The ERA-I prediction is the control variable of the Kalman Filter.

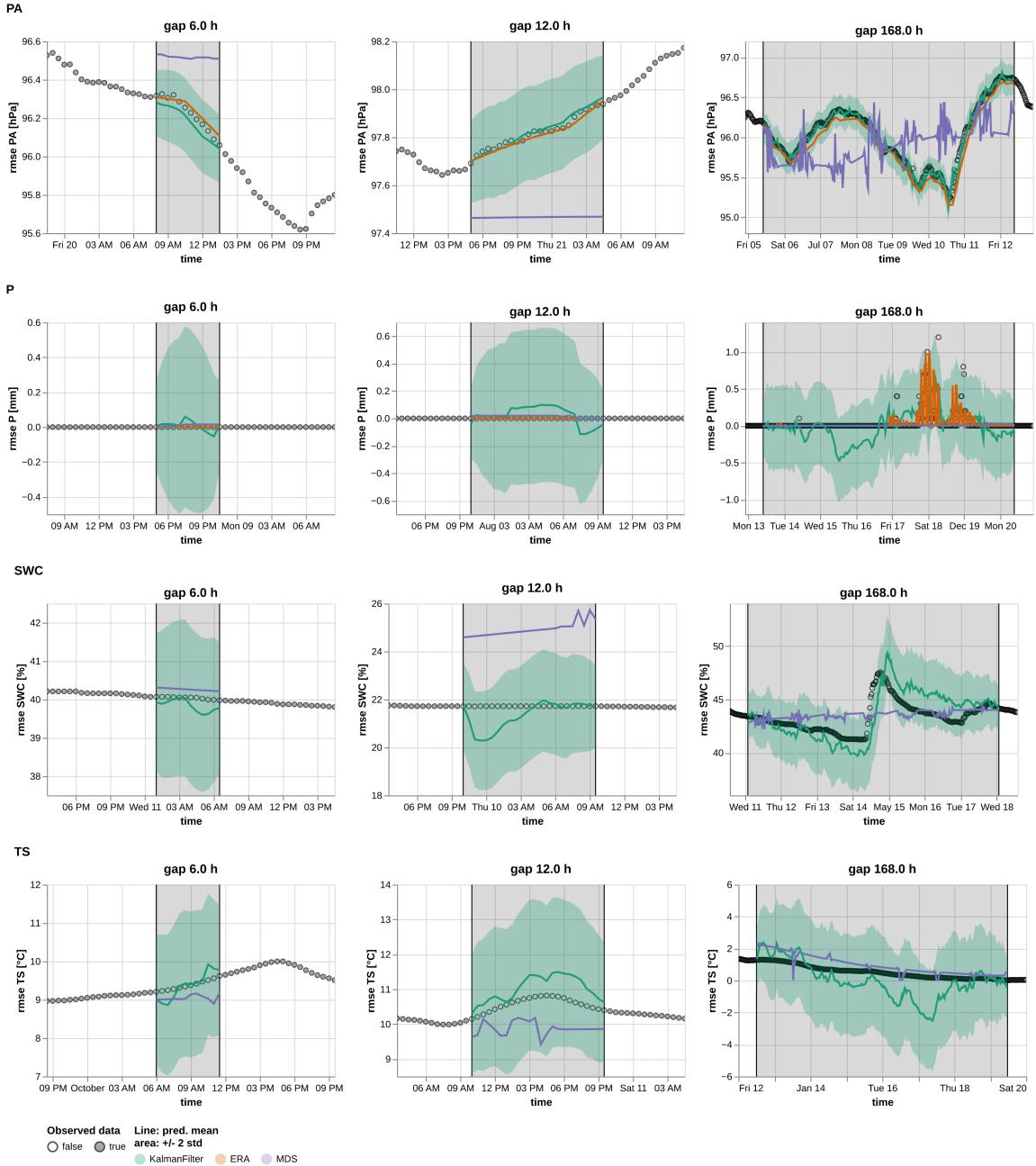


Figure 6: Illustration of the imputation PA, P, SWC, TS using different methods: Kalman Filter, ERA-Interim (ERA-I) and Marginal Distribution Sampling (MDS). The methods are compared by visualizing the time series of an artificial random gap for a three gap lengths. For each variable, 3 random artificial gap (length 6 hours, 12 hours, 1 week) are imputed using the three methods: Kalman Filter (green), ERA-I (orange), MDS (purple). For the Kalman Filter the shared area show the uncertainty of the prediction $\pm 2\sigma$. The grey shaded area and the vertical black lines delimit the artificial gaps, where the observations are not available to the model but are used to assess the imputation performance. The ERA-I prediction is the control variable of the Kalman Filter.

3.3 Aspect affect Kalman Filter performance

3.3.1 Gap Length

The effect of the gap length on the KF imputation performance is analysed by measuring the RMSE on gaps in a single variable for lengths ranging from 1 hour to 1 week (figure 7 and appendix table 5). For the majority of the variables the error increases with the gap length only up to 24 hours and then is overall constant. For three variables (P, SWC and TS) the imputation error keep increasing after 24 hours, but the rate decreases with the gap length. In all cases, the biggest change in the error happens between 1 hours and 12 hours long gaps.

For all variables, there is a significant difference in the error of very short gaps (1 hour) and long gaps (1 week) in all variables, on average the error for gaps 1 hour long is the 31% of the one for 1 week long gaps. The variability in the RMSE between gaps (i.e. std in appendix table 5) is roughly constant between the different gap lengths.

3.3.2 Control variables

The importance of the control variables is assessed by comparing the imputation error of a model that uses the control variables (*KF-Gen-Sin-6_336*) with a model that does not have access to the control variables (*KF-Gen-Sin-6_336-No_Contr*). Both models are not fine-tuned for each variable, but are generic. In general, the control variables improve the importation performance for all variables for all gap lengths. The exceptions are P and TS, where the use of the two models are equivalent, and for short gaps (6 hours) in SWC and WS. For all variables, the longer the gap, the biggest the performance improvement of the model with the control variables (Appendix table 6). Notably, the use of the control variables improve the prediction performance for SWC, even though is a variable that is not present in ERA-I. The variable where the use of the control results in the biggest improvement is PA, where for gaps 1 week long the model without the control has an error almost 6 times bigger than the one with the control.

3.3.3 Gaps in multiple variables

The importance of the inter-variable correlation for the KF predictions has been assessed by comparing the imputation for a gap with only one variable missing and then same gap with all variables missing. All the gaps were imputed using the same model, *KF-Gen-Multi-6_30* (Figure 2) and the gap length is limited to 15 hours due to numerical stability issues.

The presence of other variables in the gap is overall improving the model predictions, for some variables there is a significant error reduction (e.g. around 40 % for TA) but for others the improvement is minimal (e.g. less than 2% for WS). The pattern pattern described by the inter-variable correlation, the improvement is minimal for variable that have a low correlation with others

Across the different variables, there is an increase in the absolute values of the difference in RMSE with an increase in gap length.

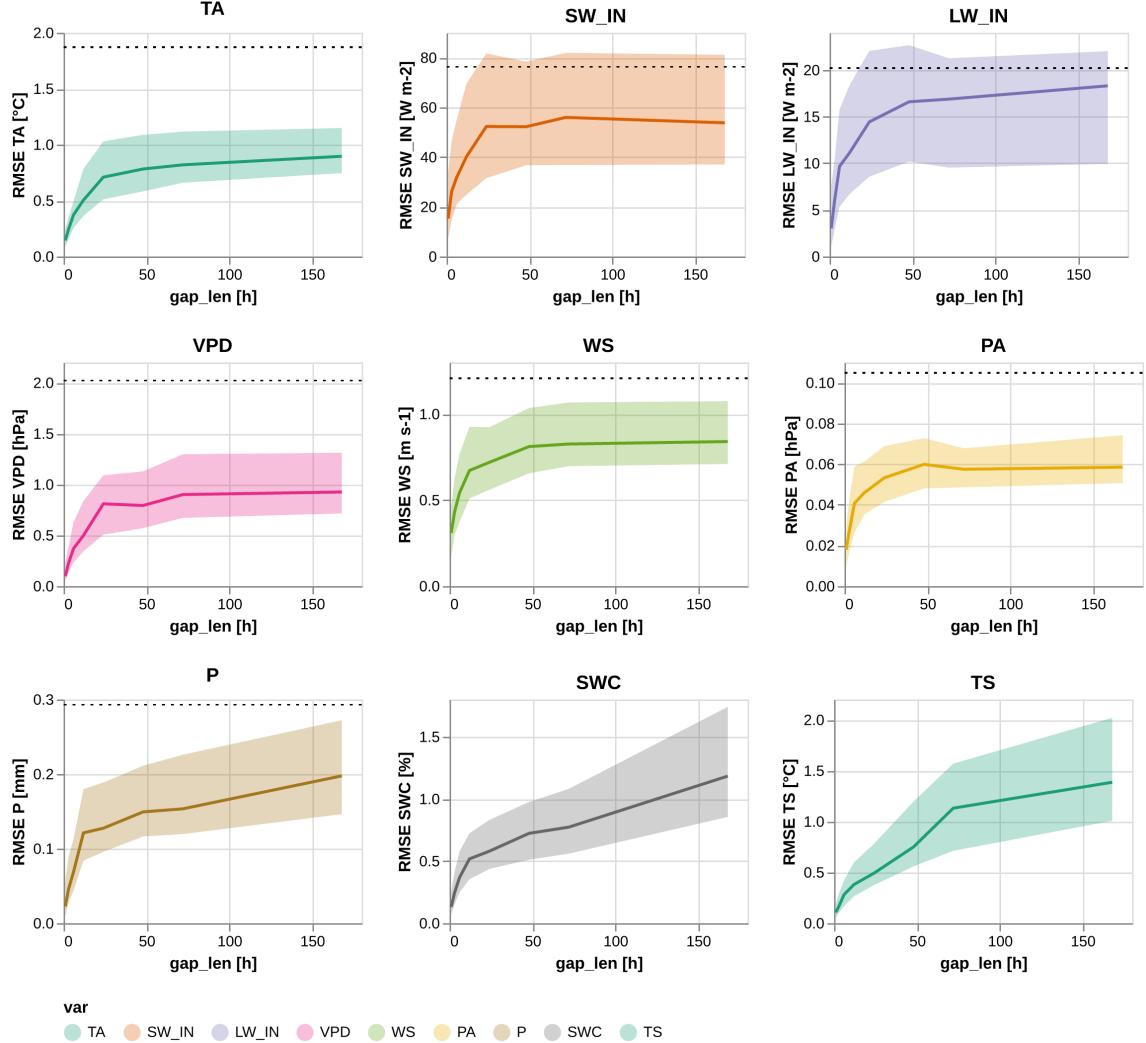


Figure 7: Effect of gap length on the KF performance. The solid line shows the median RMSE, while the shaded area is delimited by the first and third quartile. The dotted black line is the mean ERA-I error for the entire dataset (ERA-I data is not available for SWC and TS). Seven different gaps lengths were tested (1 hour, 3 hours, 6 hours, 12 hours, 1 day, 2 days, 3 days, 1 week), for each of them 500 artificial gaps were generated for each variable (total 31500 gaps). For each variable, the fine-tuned KF model has been used (*KF- $/var$ -Sin-6_336* figure 2)

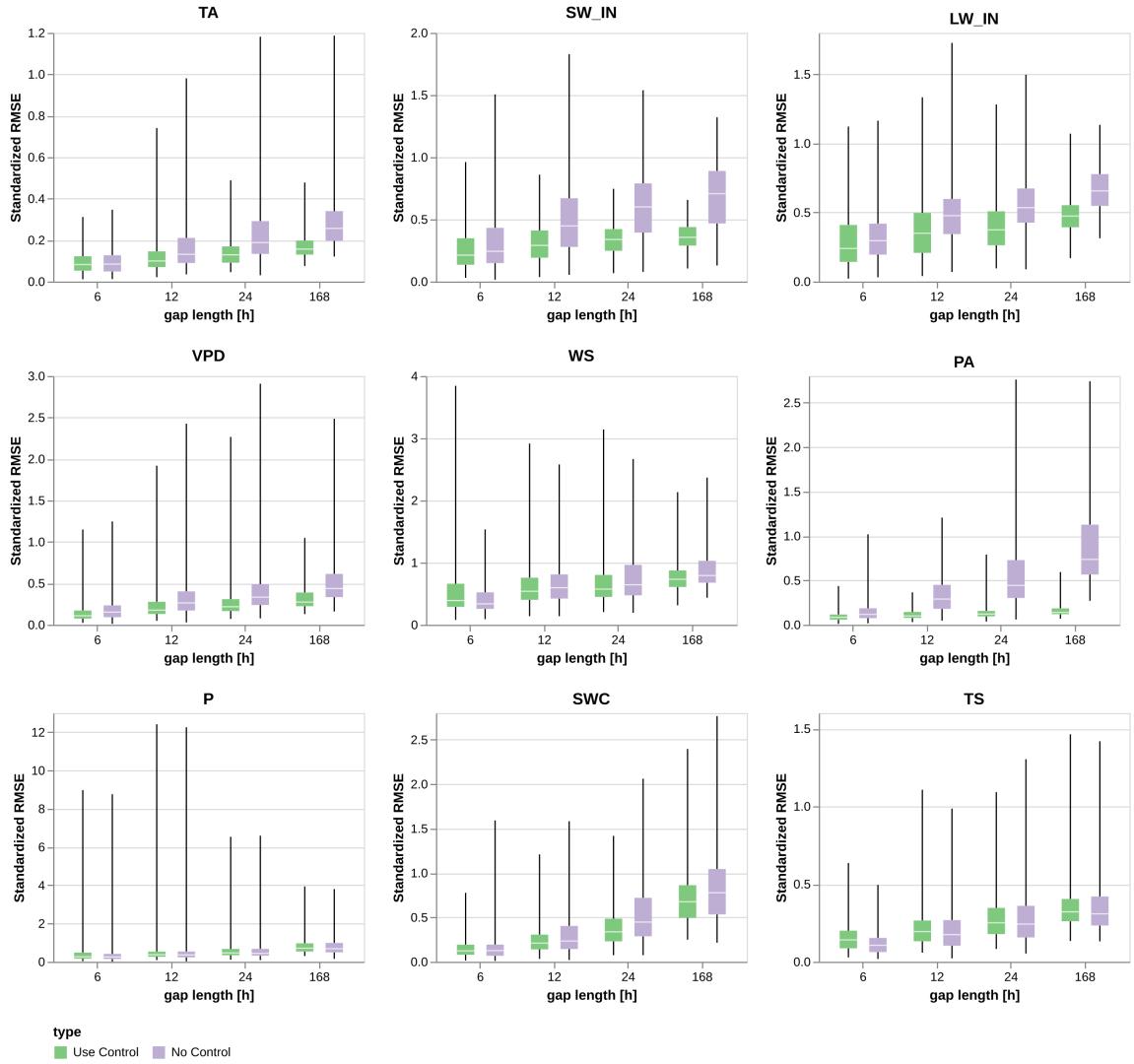
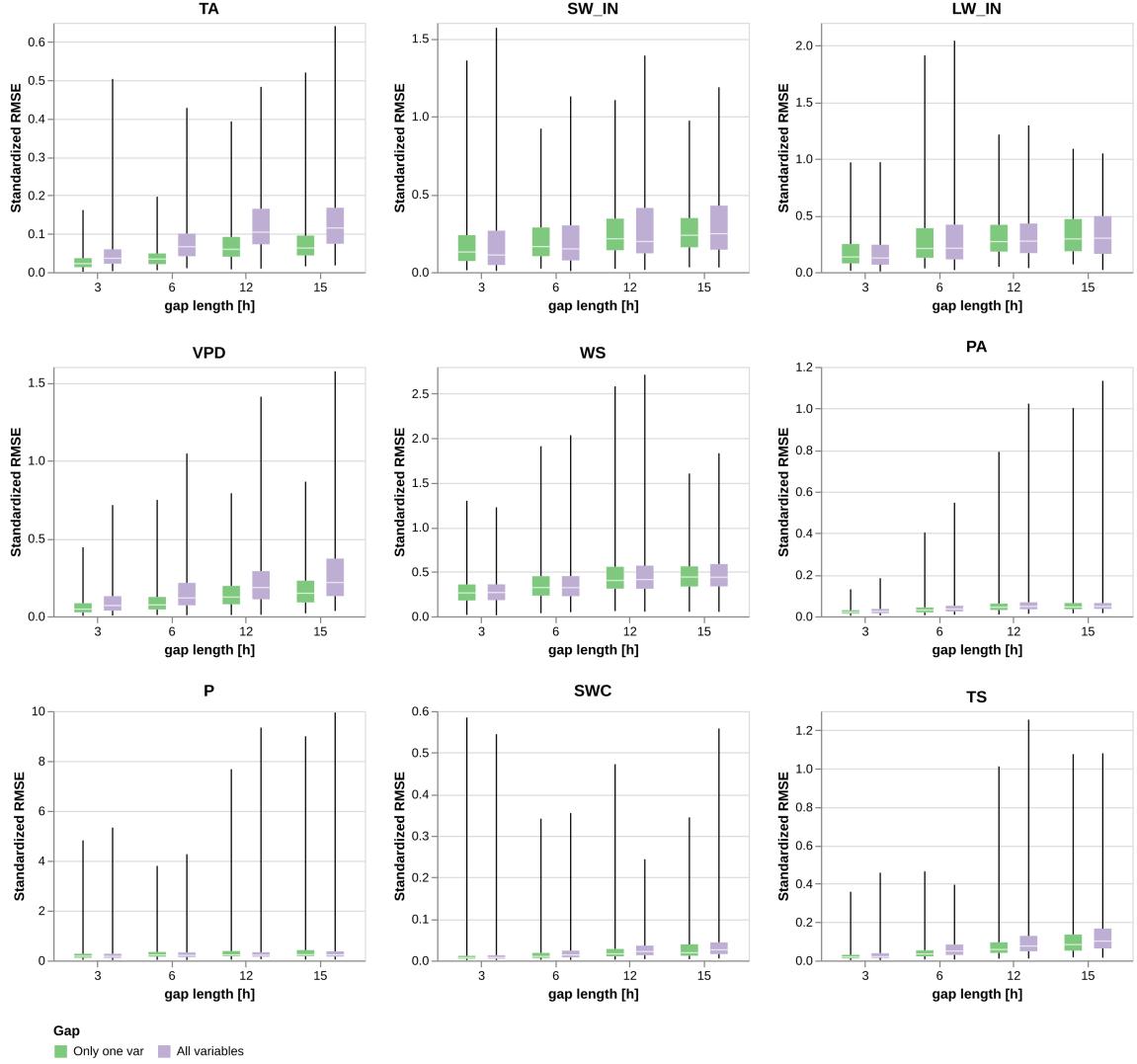


Figure 8: Comparison between KF with control variables (in green *KF-Gen-Sin-6_336*) and KF without control variable (in purple *KF-Gen-Sin-6_336-No_Contr*). For each combination of variable and gap length, 500 artificial gaps were created.



3.4 Kalman Filter training

3.4.1 Variable fine-tuning

The performance of the KF is improved if the model is fine-tuned with gaps only for one variable (e.g. only for TA). For each variable a different model has been used (*KF-(var)-Sin-6_336*) and the performance compared to a generic model that has been trained with one gap in any variable (*KF-Gen-Sin-6_336*). For each combination of gap length and variable, 500 artificial gaps were created.

The fine-tuning is reducing the error for all the variables (figure 10 and appendix table ??). The entity of the error reduction changes depending on the variable, ranging from 75% for SWC to 12% for LW_IN. The precipitation is not only variable that is not fine-tuned, as during the training no consistent improvement in performance was found. For the majority of variables the variability of the RMSE between gaps of the fine-tuned model and the maximum RMSE is also smaller for the fine-tuned model.

3.4.2 Training limitations

During the training of different KF versions I experienced limitation in the ability of the KF to learn the optimal parameters. For instance, if the KF is initialized with random parameters it will never achieve the same performance of a KF initialized with the local trend model. The training conditions are the same, so the model that starts with random parameters should be able to learn the same parameters, but this is often not the case.

The difficulty to train the KF model is represented in figure 11. There three models are compared: a model trained with gaps in multiple variables (*KF-Gen-Multi-6_336*), a mode trained with gaps of only one variable (*KF-Gen-Sin-6_336*) and one model trained with gaps in multiple variables but initialized with random parameters (*KF-Gen-Multi-6_336-Rand*). All models are trained until there is no improvement in the validation loss. The three models are used to impute a gap where only one variable is missing. The expectation is that they should have comparable performance, but this is not the case (figure 11 and appendix table ??). The training conditions between *KF-Gen-Mulit-6_336* and *KF-Gen-Multi-6_336-Rand* are exactly the same, so they should be able to achieve the same performance. The other model, *KF-Gen-Sin-6_336*, is trained on the same conditions that are used for testing, so it should not perform worse than *KF-Multi-Sin-6_336*.

The best model is always *KF-Gen-Sin-6_336*, while depending on variables the second best model is either *KF-Gen-Sin-6_336* or *KF-Gen-Multi-6_336-Rand*.

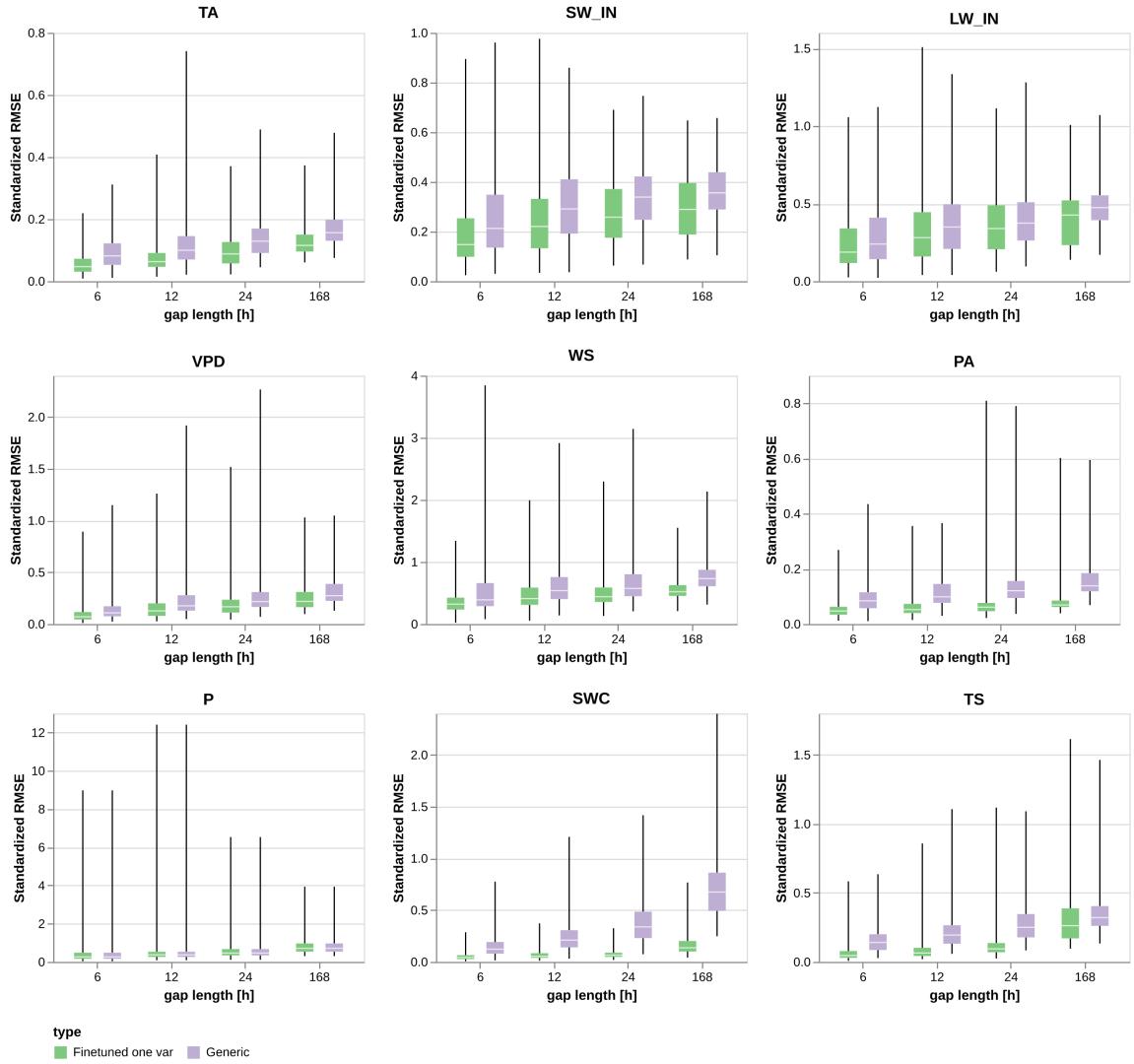


Figure 10: Comparison between KF models fine-tuned to each variable (in green $KF\text{-}\langle var \rangle\text{-}Sin-6_336$) and generic model trained for gaps in any variable (in purple $KF\text{-}Gen\text{-}Sin-6_336$). For each combination of variable and gap length, 500 artificial gaps were created.

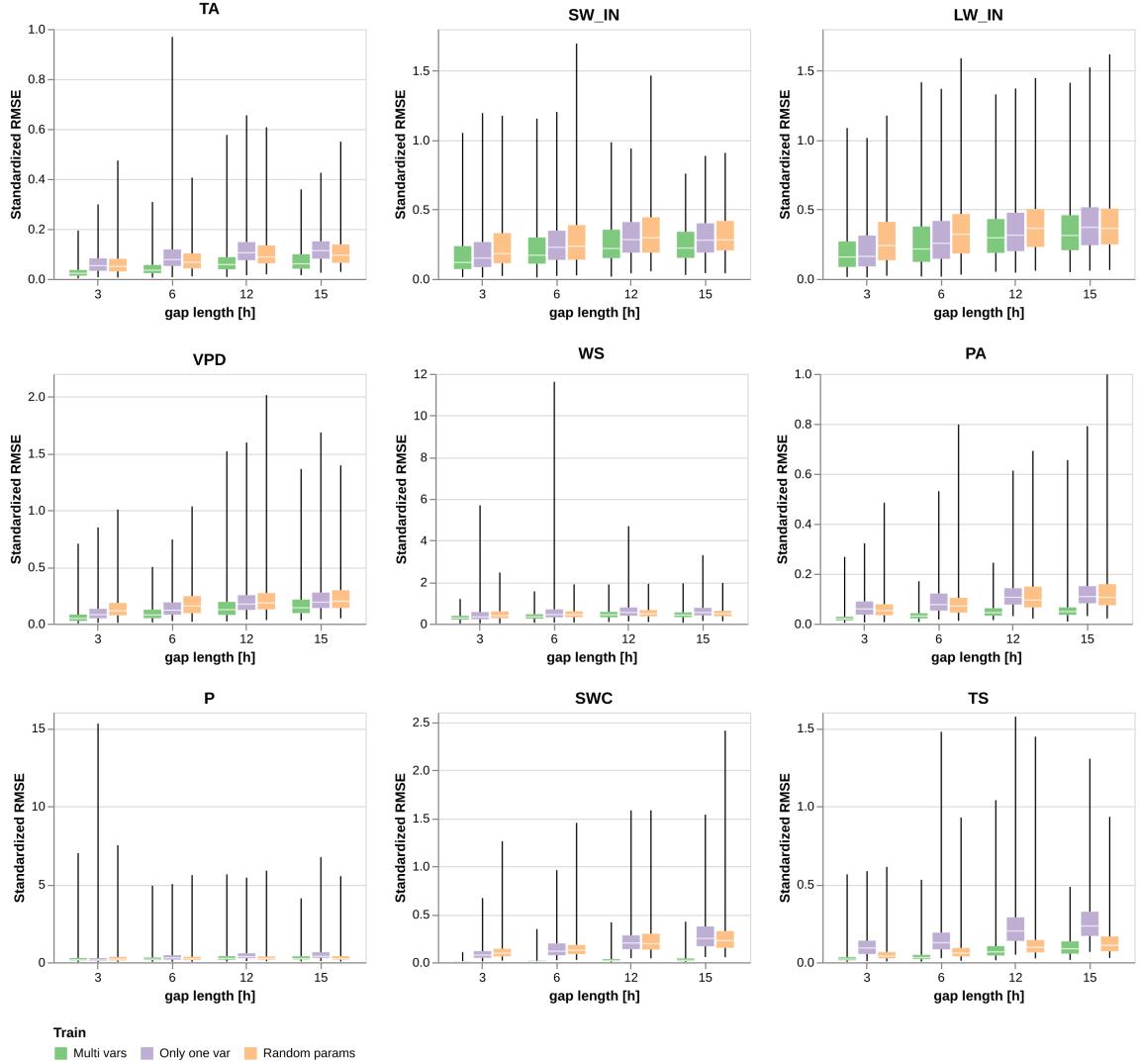


Figure 11: Visualization of KF training difficulties. For a gap in one variable, three models are compared: a model trained with gaps in multiple variables (in green *KF-Gen-Multi-6_336*), a mode trained with gaps of only one variable (in purple *KF-Gen-Sin-6_336*) and one model trained with gaps in multiple variables but initialized with random parameters (in orange *KF-Gen-Multi-6_336-Rand*). The models are expected to have the same performance. For each combination of variable and gap length, 500 artificial gaps were created.

4 Discussion

4.1 Kalman Filter performance

The results confirm the hypothesis that temporal autocorrelation is a key approach for imputing gaps in meteorological variables. One of the biggest strength of the KF is that it make better use of temporal autocorrelation, which is the likely the biggest single factor behind the improved performance. The fact that the error is increasing with the gap length only for the first 24 hours is confirming this. Moreover, the variables the relative performance of the filter is higher for variables that have a higher temporal correlation (TA, PA and VPD)

The improvement of KF performance across all variables when including the control variables confirms the hypothesis that the combination of the imputation approaches can produce better predictions. In line with the expectations, the use of the ERA-I data is crucial for the filter performance for long gaps, but results only in limited improvements for short gaps. Notably, the use of the control improve the predictions also for SWC and TS, which even if they are directly present in the ERA-I dataset, they correlated with other variables that are present in the control.

The results suggest that depending on the length of the gap the optimal imputation strategy is different, for short gaps the most important aspect is the temporal autocorrelation, while for longer gaps the control variable and inter-variable correlation plays a bigger role. This is expected, as for many variable the observations before and after the gap can provide a lot of information to reconstruct the missing data, but the effect is reduced with the gap length and other source of information are needed to correctly model the missing variable.

The numerical stability issues limited our analysis of the effect of the inter-variable correlation for short gaps, where we expect the effect to be limited. In fact, the presence of the information of other variables The results, for the of figure 9, highlight a limitation of the model training. The model trained with the gap in only one variable, should be able to learn to ignore the information of the other variable and achieve the same performance as the model trained with gaps in all variables. However, this is not the case, which is likely due to the optimizer finding a local minimum. The low importance of inter-variable correlation may also depend on the parameter initialization, where each observed variables is directly mapped to a state variable. This issue in the model training may be the source of the excessive variability in the predictions of TS and SWC. Further research would be needed to understand the behaviour of the model training and the impact of the initial parameters.

The best imputation of the KF is achieved only when the model parameters are fine-tuned to the specific conditions. We tested the fine-tuning of the model both to one variable and to short gaps, with both scenarios producing an increase in the performance. Moreover, we expect that further tuning the model to more granular conditions would improve the performance. For instance, the rate of change of the soil temperature is significantly higher in dry summer day compared to winter day with the ground covered in snow. Therefore, the optimal KF parameters would be different between those conditions.

Variables considerations

Shortwave radiation The low performance of the SW_IN at night highlights a limitation of the Kalman filter. At its core the filter considers only the evolution of the filter between consecutive time steps, and since hence it cannot model the daily pattern of SW_IN. Moreover, the rate of change

of **SW_IN** between consecutive time steps is between the day and the night is very different and since the KF uses constant parameters cannot include this variability. The control variable is the main source of information for the filter, which is confirmed by the results.

Precipitation The precipitation has a low temporal autocorrelation and low correlation with other variables, this significantly limits the ability of the KF to accurately impute missing values. The accuracy of ERA-I is limited as well, in bias correction is done using annual sums and not the precipitation at 30 mins intervals, because the timing is considered not accurate enough. Moreover, there is no precipitation in the majority of the observations (over 90% data points in Hainich), for which the RMSE is a well suited metric. In fact, [10] developed a spatial and temporal imputation method for precipitation, that employs two different models: first they predict for each data point, whether there is precipitation or not and then use another models to predict the amount of precipitation. Overall the precipitation has a high spatial and temporal variability [28] and different characteristics than other meteorological variable. The KF is not suited for the imputation precipitation, which would require a tailored modelling approach that is beyond the scope of this work.

Wind Speed The KF doesn't model accurately the high frequency variation of the wind speed, and is the variable with the highest standardized RMSE. This is consistent with the results [41]. The KF cannot extract the information about the high frequency variation from other observations of the wind sped, but KF is able to model higher frequency variations only if the information is present in either the control variable or in other variables. For the case of the wind speed this is not the case, as ERA-I has this limitation and no other variable has a high correlation with the wind speed. This limitation of ERA-I is also the likely reason behind the increased error in **WS** when using the control for short gap length, where low variability in ERA-I is negatively affecting the model prediction. This is scenario shows the limitations of the simplicity KF, while a properly designed Gaussian Process should be able to model the high frequency variation. Nonetheless, the KF has a better performance than the current imputation methods.

Training

Numerical stability becomes not positive definite, hence is impossible to compute the log likelihood appendix figures

4.2 Kalman Filter application

Our results suggest that a Kalman Filter can be employed to improve the imputation of meteorological variables for EC applications, especially for gaps shorter than one day. The highest error reduction is for short gaps (less than 1 day) and in particular for the air temperature. We expect that is relevant to reduce error in Land Surface Models, as they work on short time scales and the temperature is a key driver of core ecosystem processes. Processed like photosynthesis or respiration have a strong non-linear dependency on temperature, making inaccuracies on the temperature relevant for [5]. The relative performance of KF is reduced for medium gaps (1 week) and we assume that for longer gaps the error of the KF is going to get progressively closer to the ERA-I one. Nonetheless, the KF can be successfully applied to impute long gaps for variable that

are not available in ERA-I (e.g. TS or SWC) as it exploits the ERA-I observations for other variables and the inter-variable correlation.

In any scenario, the KF provides an interpretable estimate of uncertainty. Currently, I am not aware of any imputation methods that provide an interpretable uncertainty and thus no application have been developed for it. However, the ability to accurately assess the quality of the gap filling for each data point can be important to an effective use of the imputed time series.

The current KF implementation has two limitations that would prevent the application in a production scenario: numerical instability when all variables are missing for more than 15 hours, physically impossible predictions of SW_IN at night. However, I believe that the first two limitations are relatively straightforward to overcome, as suggested in the following section. The deployment of the KF may be impacted by the need to fine-tune the model parameters to achieve the best performance. This is inherited to the KF structure. The current models is using 9 different KF, each of them optimized for one variable. This means that while it is possible to run the KF to impute gaps all variables, the best results are only achieved when imputing each variable separately. Moreover, I tested the KF only one site, but from the results of [41] show that the accuracy of ERA-I, so the control variable parameters would likely need to fine-tuned to each variable. In addition, in the current implementation fine-tuning procedure requires a manual intervention to interrupt the training process and avoid overfitting the model. The necessity to fine-tune the KF to a specific conditions is a not critical limitation, but it is necessary to consider the increased computation cost and the deployment complexity.

4.3 Future Outlook

Model improvements This work builds the foundations to have a KF based method for imputing meteorological variables. However, we believe that there are at least two that are required before considering using the model in production.

The numerical instability of the current KF implementation limits the gaps to 15 hours when all variables are missing. The current implementation is a hybrid between a square root filter and standard Kalman Filter. The use of the full covariance matrices in the smoothing pass and the in the log-likelihood computations is the source of the numerical instability. More research is needed on developing a suitable formulation of a square root smoother, which the available literature suggest that is possible to derive [38, 33].

The KF often predicts negatives values of the shortwave radiation at night, that are physically impossible. However, imputing shortwave radiation at night is a simple problem since is always zero and the exact time of the sunrise and sunset can be computed from the day of the year and the geographic coordinates. Therefore the KF can be used to impute only the values during the day, where it is the method with the best performance.

I identify further directions...

ERA-5 Land The European Centre for Medium-Range Weather Forecasts recently released two new weather reanalysis datasets: ERA5 in 2020 [22] and ERA-5 Land in 2021 [31], which supersede the ERA-Interim dataset. The ERA5-Land dataset covers only the continent, but has a much higher spatial resolution (9 km vs 80km) ad higher temporal resolution (1 hour instead of 3 hour). The use ERA5-Land data as control variable for the KF has the potential to improve the imputation performance.

Parameter initialization and Training Our analysis suggest initial parameters have significant influence on the final parameters and thus the performance of the model. There are several approaches to initialize a Kalman Filter [durbin'time'2012-4] and a robust comparison between the different methods may reveal a better initialization strategy than the linear trend model.

Model Filter Parameters One of the advantages of the iterative nature of the KF formulation is the ability to change the parameters between time step. This opens the possibility to have a model that predicts the parameters of the filter depending on the conditions, which should overcome the necessity to fine-tune the KF parameters to each condition.

Non-linear transformation control The KF now has a linear , but the real relationship is not linear . The formulation of the Kalman Filter doesn't constraint the control variable transformation to be linear, thus a model the ERA data can be processed with a more powerful model, which as a Neural Network. The implementation of the KF using PyTorch simplify such additions.

Observations noise The uncertainty of the observations can be estimated from the instrument accuracy and used as input of the model to avoid underestimating the uncertainty of the observations. For instance, the typical accuracy of a shortwave measurement is in the order of 10 Wm^2 , which is comparable to the uncertainties of the model.

Model Evaluation This study provides a first evaluation of the imputation performance of different, but a more in-depth analysis contribute to understand the impact in the real impact of the imputation methods.

The first aspect is to test the imputation performance usgin observations from different sites. Another aspect to consider is that the meteorological data is not missing completely at random. Further research would be needed, but is reasonable to assume that there is a correlation between the gaps of different variable and that the time of the year has an impact on data availability. One reason that can contribute to patterns in the missing that is the fact that the air temperature and vapour pressure deficit are often measured from the same sensor (eg. [39]), which increase the likelihood of both variable missing at the same time. Another scenario is a station that uses solar panels, which are more likely to have power failure during winter. A robust assessment of the imputation performance should include missing data with realistic patterns.

The metric used for the evaluation is also important. The RMSE is limited to time series measure the average distance between the predictions and the observations, but doesn't compare characteristics of the time series such as the variance, the shape of the presence of a time-shift [20]. The use of additional metrics (eg. DILATE [20]) can improve the understanding of the quality of the imputation.

Finally, the impact of the KF imputation can be evaluated by the reduction in the error of Land Surface Models compared to state of the art methods. Improving the quality of the predictions of Land Surface Models is arguably the main reason to use KF imputation. This can be assess directly by running a Land Surface Model with a complete time series, and the one of application of imputed meteorological

5 Conclusions

- Kalman Filter from preliminary results have the potential to improve imputation of meteo data
- on all tested variables smaller or comparable error than other methods, but P
- lower variability
- provide interpretable uncertainty
- limitations implementation
 - numerical stability
 - SW_IN
- need to fine tune parameters is a limitation of the model
- can test different init parameters and effect of model training
- potential for further development
- robust assessment of performance in real life conditions and test different sites

References

- [1] Marc Aubinet, Timo Vesala, and Dario Papale, eds. *Eddy Covariance: A Practical Guide to Measurement and Data Analysis*. Dordrecht: Springer Netherlands, 2012. ISBN: 978-94-007-2350-4 978-94-007-2351-1. DOI: [10.1007/978-94-007-2351-1](https://doi.org/10.1007/978-94-007-2351-1). URL: <https://link.springer.com/10.1007/978-94-007-2351-1> (visited on 02/10/2023).
- [2] M. Balzarolo et al. “Evaluating the Potential of Large-Scale Simulations to Predict Carbon Fluxes of Terrestrial Ecosystems over a European Eddy Covariance Network”. In: *Biogeosciences* 11.10 (May 20, 2014), pp. 2661–2678. ISSN: 1726-4170. DOI: [10.5194/bg-11-2661-2014](https://doi.org/10.5194/bg-11-2661-2014). URL: <https://bg.copernicus.org/articles/11/2661/2014/> (visited on 02/10/2023).
- [3] Simon Besnard et al. “Quantifying the Effect of Forest Age in Annual Net Forest Carbon Balance”. In: *Environmental Research Letters* 13.12 (Dec. 2018), p. 124018. ISSN: 1748-9326. DOI: [10.1088/1748-9326/aaeaeb](https://doi.org/10.1088/1748-9326/aaeaeb). URL: <https://dx.doi.org/10.1088/1748-9326/aaeaeb> (visited on 02/24/2023).
- [4] Gerald J. Bierman and Catherine L. Thornton. “Numerical Comparison of Kalman Filter Algorithms: Orbit Determination Case Study”. In: *Automatica* 13.1 (Jan. 1, 1977), pp. 23–35. ISSN: 0005-1098. DOI: [10.1016/0005-1098\(77\)90006-1](https://doi.org/10.1016/0005-1098(77)90006-1). URL: <https://www.sciencedirect.com/science/article/pii/0005109877900061> (visited on 01/12/2023).
- [5] Gordon Bonan. *Climate Change and Terrestrial Ecosystem Modeling*. Cambridge University Press, 2019.

- [6] Gordon B. Bonan et al. “Improving Canopy Processes in the Community Land Model Version 4 (CLM4) Using Global Flux Fields Empirically Inferred from FLUXNET Data”. In: *Journal of Geophysical Research: Biogeosciences* 116.G2 (2011). ISSN: 2156-2202. DOI: [10.1029/2010JG001593](https://doi.org/10.1029/2010JG001593). URL: <https://onlinelibrary.wiley.com/doi/abs/10.1029/2010JG001593> (visited on 02/10/2023).
- [7] Stef van Buuren and Karin Groothuis-Oudshoorn. “Mice: Multivariate Imputation by Chained Equations in R”. In: *Journal of Statistical Software* 45 (Dec. 12, 2011), pp. 1–67. ISSN: 1548-7660. DOI: [10.18637/jss.v045.i03](https://doi.org/10.18637/jss.v045.i03). URL: <https://doi.org/10.18637/jss.v045.i03> (visited on 02/18/2023).
- [8] Wei Cao et al. “BRITS: Bidirectional Recurrent Imputation for Time Series”. In: (), p. 11.
- [9] NEAL A. CARLSON. “Fast Triangular Formulation of the Square Root Filter.” In: *AIAA Journal* 11.9 (1973), pp. 1259–1265. ISSN: 0001-1452. DOI: [10.2514/3.6907](https://doi.org/10.2514/3.6907). URL: <https://doi.org/10.2514/3.6907> (visited on 02/15/2023).
- [10] Benedict D. Chivers et al. “Imputation of Missing Sub-Hourly Precipitation Data in a Large Sensor Network: A Machine Learning Approach”. In: *Journal of Hydrology* 588 (Sept. 1, 2020), p. 125126. ISSN: 0022-1694. DOI: [10.1016/j.jhydrol.2020.125126](https://doi.org/10.1016/j.jhydrol.2020.125126). URL: <https://www.sciencedirect.com/science/article/pii/S0022169420305862> (visited on 02/21/2023).
- [11] Rafaela Lisboa Costa et al. “Gap Filling and Quality Control Applied to Meteorological Variables Measured in the Northeast Region of Brazil”. In: *Atmosphere* 12.10 (10 Oct. 2021), p. 1278. ISSN: 2073-4433. DOI: [10.3390/atmos12101278](https://doi.org/10.3390/atmos12101278). URL: <https://www.mdpi.com/2073-4433/12/10/1278> (visited on 05/14/2022).
- [12] Dan Simon. *Optimal State Estimation Kalman, H and Nonlinear Approaches*. 2006. ISBN: 100-47 1-70858-5.
- [13] D. P. Dee et al. “The ERA-Interim Reanalysis: Configuration and Performance of the Data Assimilation System”. In: *Quarterly Journal of the Royal Meteorological Society* 137.656 (2011), pp. 553–597. ISSN: 1477-870X. DOI: [10.1002/qj.828](https://doi.org/10.1002/qj.828). URL: <https://onlinelibrary.wiley.com/doi/abs/10.1002/qj.828> (visited on 02/24/2023).
- [14] Wenjie Du, David Cote, and Yan Liu. *SAITS: Self-Attention-based Imputation for Time Series*. May 9, 2022. arXiv: [2202.08516 \[cs\]](https://arxiv.org/abs/2202.08516). URL: [http://arxiv.org/abs/2202.08516](https://arxiv.org/abs/2202.08516) (visited on 05/14/2022).
- [15] James Durbin and Siem Jan Koopman. *Time Series Analysis by State Space Methods*. Mar. 2, 2012. URL: <https://doi.org/10.1093/acprof:oso/9780199641178.001.0001>.
- [16] Chenguang Fang and Chen Wang. “Time Series Data Imputation: A Survey on Deep Learning Approaches”. Nov. 23, 2020. arXiv: [2011.11347 \[cs\]](https://arxiv.org/abs/2011.11347). URL: [http://arxiv.org/abs/2011.11347](https://arxiv.org/abs/2011.11347) (visited on 05/12/2022).
- [17] Vincent Fortuin et al. *GP-VAE: Deep Probabilistic Time Series Imputation*. Feb. 20, 2020. DOI: [10.48550/arXiv.1907.04155](https://arxiv.org/abs/1907.04155). arXiv: [1907.04155 \[cs, stat\]](https://arxiv.org/abs/1907.04155). URL: [http://arxiv.org/abs/1907.04155](https://arxiv.org/abs/1907.04155) (visited on 07/28/2022).
- [18] Andrew D. Friend et al. “FLUXNET and Modelling the Global Carbon Cycle”. In: *Global Change Biology* 13.3 (2007), pp. 610–633. ISSN: 1365-2486. DOI: [10.1111/j.1365-2486.2006.01223.x](https://doi.org/10.1111/j.1365-2486.2006.01223.x). URL: <https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1365-2486.2006.01223.x> (visited on 02/10/2023).

- [19] Marta Garnelo et al. *Neural Processes*. July 4, 2018. doi: [10.48550/arXiv.1807.01622](https://doi.org/10.48550/arXiv.1807.01622). arXiv: [1807.01622 \[cs, stat\]](https://arxiv.org/abs/1807.01622). URL: <http://arxiv.org/abs/1807.01622> (visited on 06/23/2022).
- [20] Vincent LE Guen and Nicolas Thome. “Shape and Time Distortion Loss for Training Deep Time Series Forecasting Models”. In: (), p. 13.
- [21] Philipp Hennig. *Probabilistic Machine Learning*. lecture course. University of Tuebingen, 2020.
- [22] Hans Hersbach et al. “The ERA5 Global Reanalysis”. In: *Quarterly Journal of the Royal Meteorological Society* 146.730 (2020), pp. 1999–2049. ISSN: 1477-870X. doi: [10.1002/qj.3803](https://doi.org/10.1002/qj.3803). URL: <https://onlinelibrary.wiley.com/doi/abs/10.1002/qj.3803> (visited on 02/22/2023).
- [23] Peter Isaac et al. “OzFlux Data: Network Integration from Collection to Curation”. In: *Biogeosciences* 14.12 (June 19, 2017), pp. 2903–2928. ISSN: 1726-4170. doi: [10.5194/bg-14-2903-2017](https://doi.org/10.5194/bg-14-2903-2017). URL: <https://bg.copernicus.org/articles/14/2903/2017/> (visited on 05/12/2022).
- [24] Xin Jing et al. “A Multi-imputation Method to Deal With Hydro-Meteorological Missing Values by Integrating Chain Equations and Random Forest”. In: *Water Resources Management* 36.4 (Mar. 1, 2022), pp. 1159–1173. ISSN: 1573-1650. doi: [10.1007/s11269-021-03037-5](https://doi.org/10.1007/s11269-021-03037-5). URL: <https://doi.org/10.1007/s11269-021-03037-5> (visited on 02/18/2023).
- [25] P. Kaminski, A. Bryson, and S. Schmidt. “Discrete Square Root Filtering: A Survey of Current Techniques”. In: *IEEE Transactions on Automatic Control* 16.6 (Dec. 1971), pp. 727–736. ISSN: 1558-2523. doi: [10.1109/TAC.1971.1099816](https://doi.org/10.1109/TAC.1971.1099816).
- [26] K. Kramer et al. “Evaluation of Six Process-Based Forest Growth Models Using Eddy-Covariance Measurements of CO₂ and H₂O Fluxes at Six Forest Sites in Europe”. In: *Global Change Biology* 8.3 (2002), pp. 213–230. ISSN: 1365-2486. doi: [10.1046/j.1365-2486.2002.00471.x](https://doi.org/10.1046/j.1365-2486.2002.00471.x). URL: <https://onlinelibrary.wiley.com/doi/abs/10.1046/j.1365-2486.2002.00471.x> (visited on 01/18/2023).
- [27] Miguel D. Mahecha et al. “Detecting Impacts of Extreme Events with Ecological in Situ Monitoring Networks”. In: *Biogeosciences* 14.18 (Sept. 25, 2017), pp. 4255–4277. ISSN: 1726-4170. doi: [10.5194/bg-14-4255-2017](https://doi.org/10.5194/bg-14-4255-2017). URL: <https://bg.copernicus.org/articles/14/4255/2017/> (visited on 02/24/2023).
- [28] Utkarsh Mital et al. “Sequential Imputation of Missing Spatio-Temporal Precipitation Data Using Random Forests”. In: *Frontiers in Water* 2 (2020). ISSN: 2624-9375. URL: <https://www.frontiersin.org/articles/10.3389/frwa.2020.00020> (visited on 02/21/2023).
- [29] Mohinder S. Grewal and Angus P. Andrews. *Kalman Filtering: Theory and Practice Using MATLAB, Second Edition*. 2001. ISBN: 0-471-26638-8.
- [30] Steffen Moritz and Thomas Bartz-Beielstein. “The R Journal: imputeTS: Time Series Missing Value Imputation in R”. In: *The R Journal* 9.1 (May 10, 2017), pp. 207–218. ISSN: 2073-4859. doi: [10.32614/RJ-2017-009](https://doi.org/10.32614/RJ-2017-009). URL: <https://doi.org/10.32614/RJ-2017-009/> (visited on 02/24/2023).
- [31] Joaquín Muñoz-Sabater et al. “ERA5-Land: A State-of-the-Art Global Reanalysis Dataset for Land Applications”. In: *Earth System Science Data* 13.9 (Sept. 7, 2021), pp. 4349–4383. ISSN: 1866-3508. doi: [10.5194/essd-13-4349-2021](https://doi.org/10.5194/essd-13-4349-2021). URL: <https://essd.copernicus.org/articles/13/4349/2021/> (visited on 02/22/2023).

- [32] Dario Papale. “Ideas and Perspectives: Enhancing the Impact of the FLUXNET Network of Eddy Covariance Sites”. In: *Biogeosciences* 17.22 (Nov. 17, 2020), pp. 5587–5598. ISSN: 1726-4189. DOI: [10.5194/bg-17-5587-2020](https://doi.org/10.5194/bg-17-5587-2020). URL: <https://bg.copernicus.org/articles/17/5587/2020/> (visited on 01/18/2023).
- [33] PooGyeon Park and T. Kailath. “New Square-Root Smoothing Algorithms”. In: *IEEE Transactions on Automatic Control* 41.5 (May 1996), pp. 727–732. ISSN: 1558-2523. DOI: [10.1109/9.489212](https://doi.org/10.1109/9.489212).
- [34] Gilberto Pastorello et al. “The FLUXNET2015 Dataset and the ONEFlux Processing Pipeline for Eddy Covariance Data”. In: *Scientific Data* 7.1 (1 July 9, 2020), p. 225. ISSN: 2052-4463. DOI: [10.1038/s41597-020-0534-3](https://doi.org/10.1038/s41597-020-0534-3). URL: <https://www.nature.com/articles/s41597-020-0534-3> (visited on 05/12/2022).
- [35] JAMES POTTER and ROBERT STERN. “STATISTICAL FILTERING OF SPACE NAVIGATION MEASUREMENTS”. In: *Guidance and Control Conference*. American Institute of Aeronautics and Astronautics, 1963. DOI: [10.2514/6.1963-333](https://doi.org/10.2514/6.1963-333). URL: <https://arc.aiaa.org/doi/abs/10.2514/6.1963-333> (visited on 01/13/2023).
- [36] H. E. RAUCH, F. TUNG, and C. T. STRIEBEL. “Maximum Likelihood Estimates of Linear Dynamic Systems”. In: *AIAA Journal* 3.8 (1965), pp. 1445–1450. ISSN: 0001-1452. DOI: [10.2514/3.3166](https://doi.org/10.2514/3.3166). URL: <https://doi.org/10.2514/3.3166> (visited on 02/19/2023).
- [37] Markus Reichstein et al. “On the Separation of Net Ecosystem Exchange into Assimilation and Ecosystem Respiration: Review and Improved Algorithm”. In: *Global Change Biology* 11.9 (2005), pp. 1424–1439. ISSN: 1365-2486. DOI: [10.1111/j.1365-2486.2005.001002.x](https://doi.org/10.1111/j.1365-2486.2005.001002.x). URL: <https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1365-2486.2005.001002.x> (visited on 05/12/2022).
- [38] Mark G. Rutten. “Square-Root Unscented Filtering and Smoothing”. In: *2013 IEEE Eighth International Conference on Intelligent Sensors, Sensor Networks and Information Processing*. 2013 IEEE Eighth International Conference on Intelligent Sensors, Sensor Networks and Information Processing. Apr. 2013, pp. 294–299. DOI: [10.1109/ISSNIP.2013.6529805](https://doi.org/10.1109/ISSNIP.2013.6529805).
- [39] *Specification - Vaisala HMP3 General Purpose Humidity and Temperature Probe*. URL: <https://docs.vaisala.com/v/u/B211826EN-C/en-US> (visited on 02/18/2023).
- [40] *Statsmodels.Tsa.StateSpace.Kalman_filter.KalmanFilter — Statsmodels*. URL: https://www.statsmodels.org/dev/generated/statsmodels.tsa.statespace.kalman_filter.KalmanFilter.html#statsmodels.tsa.statespace.kalman_filter.KalmanFilter (visited on 02/19/2023).
- [41] N. Vuichard and D. Papale. “Filling the Gaps in Meteorological Continuous Data Measured at FLUXNET Sites with ERA-Interim Reanalysis”. In: *Earth System Science Data* 7.2 (July 13, 2015), pp. 157–171. ISSN: 1866-3508. DOI: [10.5194/essd-7-157-2015](https://doi.org/10.5194/essd-7-157-2015). URL: <https://essd.copernicus.org/articles/7/157/2015/> (visited on 05/11/2022).
- [42] Thomas Wutzler et al. “Basic and Extensible Post-Processing of Eddy Covariance Flux Data with REddyProc”. In: *Biogeosciences* 15.16 (Aug. 23, 2018), pp. 5015–5030. ISSN: 1726-4170. DOI: [10.5194/bg-15-5015-2018](https://doi.org/10.5194/bg-15-5015-2018). URL: <https://bg.copernicus.org/articles/15/5015/2018/> (visited on 05/15/2022).

- [43] Yifan Zhang and Peter J. Thorburn. "A Dual-Head Attention Model for Time Series Data Imputation". In: *Computers and Electronics in Agriculture* 189 (Oct. 1, 2021), p. 106377. ISSN: 0168-1699. DOI: [10.1016/j.compag.2021.106377](https://doi.org/10.1016/j.compag.2021.106377). URL: <https://www.sciencedirect.com/science/article/pii/S016816992100394X> (visited on 06/15/2022).
- [44] Y. Zhao et al. "How Errors on Meteorological Variables Impact Simulated Ecosystem Fluxes: A Case Study for Six French Sites". In: *Biogeosciences* 9.7 (July 11, 2012), pp. 2537–2564. ISSN: 1726-4170. DOI: [10.5194/bg-9-2537-2012](https://doi.org/10.5194/bg-9-2537-2012). URL: <https://bg.copernicus.org/articles/9/2537/2012/> (visited on 02/13/2023).

A Additional Results

A.1 Comparison to other imputation methods

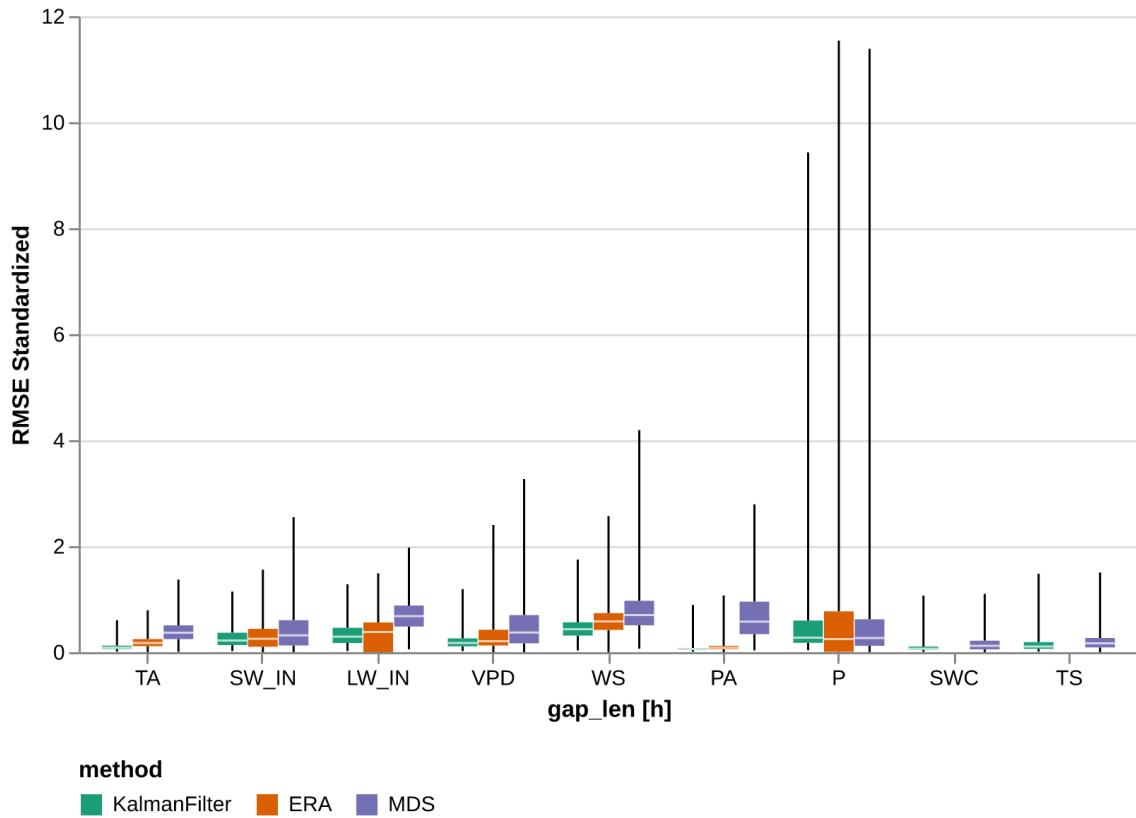


Figure 12: Box plot to compare Standardized Root Mean Square Error(RMSE) for each variable between the different methods: Kalman Filter and the state of the art methods ERA and MDS. The same data from figure 4 has been aggregated for all gap lengths

Table 4: Comparison of imputation methods using Standardized RMSE. The best method for each gap length is highlighted in bold

Variable	RMSE Gap [h]	KalmanFilter		ERA		MDS	
		mean	std	mean	std	mean	std
TA	6	0.051	0.033	0.170	0.126	0.342	0.239
	12	0.077	0.051	0.186	0.114	0.371	0.221
	24	0.094	0.046	0.193	0.101	0.380	0.203
	168	0.129	0.056	0.221	0.081	0.477	0.166
SW_IN	6	0.219	0.198	0.242	0.325	0.311	0.419
	12	0.236	0.166	0.266	0.244	0.340	0.338
	24	0.277	0.147	0.323	0.201	0.425	0.292
	168	0.302	0.126	0.344	0.171	0.526	0.263
LW_IN	6	0.260	0.184	0.329	0.310	0.636	0.358
	12	0.320	0.184	0.352	0.300	0.669	0.321
	24	0.348	0.187	0.336	0.291	0.706	0.296
	168	0.407	0.153	0.390	0.265	0.785	0.211
VPD	6	0.098	0.083	0.297	0.354	0.477	0.492
	12	0.151	0.116	0.290	0.295	0.489	0.480
	24	0.189	0.115	0.286	0.236	0.438	0.367
	168	0.258	0.145	0.380	0.258	0.609	0.450
WS	6	0.379	0.195	0.561	0.313	0.699	0.482
	12	0.440	0.216	0.588	0.323	0.776	0.490
	24	0.493	0.211	0.584	0.275	0.785	0.374
	168	0.585	0.223	0.670	0.214	0.920	0.379
PA	6	0.053	0.040	0.088	0.072	0.621	0.516
	12	0.062	0.049	0.090	0.068	0.659	0.500
	24	0.070	0.045	0.092	0.060	0.651	0.473
	168	0.078	0.056	0.098	0.063	0.904	0.449
P	6	0.478	0.978	0.404	1.126	0.420	1.090
	12	0.638	1.054	0.495	1.060	0.465	1.004
	24	0.736	0.905	0.591	1.029	0.566	0.946
	168	0.856	0.620	0.795	0.720	0.767	0.705
SWC	6	0.057	0.055	-	-	0.147	0.175
	12	0.075	0.053	-	-	0.143	0.148
	24	0.087	0.072	-	-	0.152	0.165
	168	0.168	0.106	-	-	0.219	0.167
TS	6	0.060	0.076	-	-	0.169	0.157
	12	0.094	0.139	-	-	0.177	0.155
	24	0.139	0.151	-	-	0.191	0.151
	168	0.293	0.190	-	-	0.254	0.135

A.2 Additional Time series

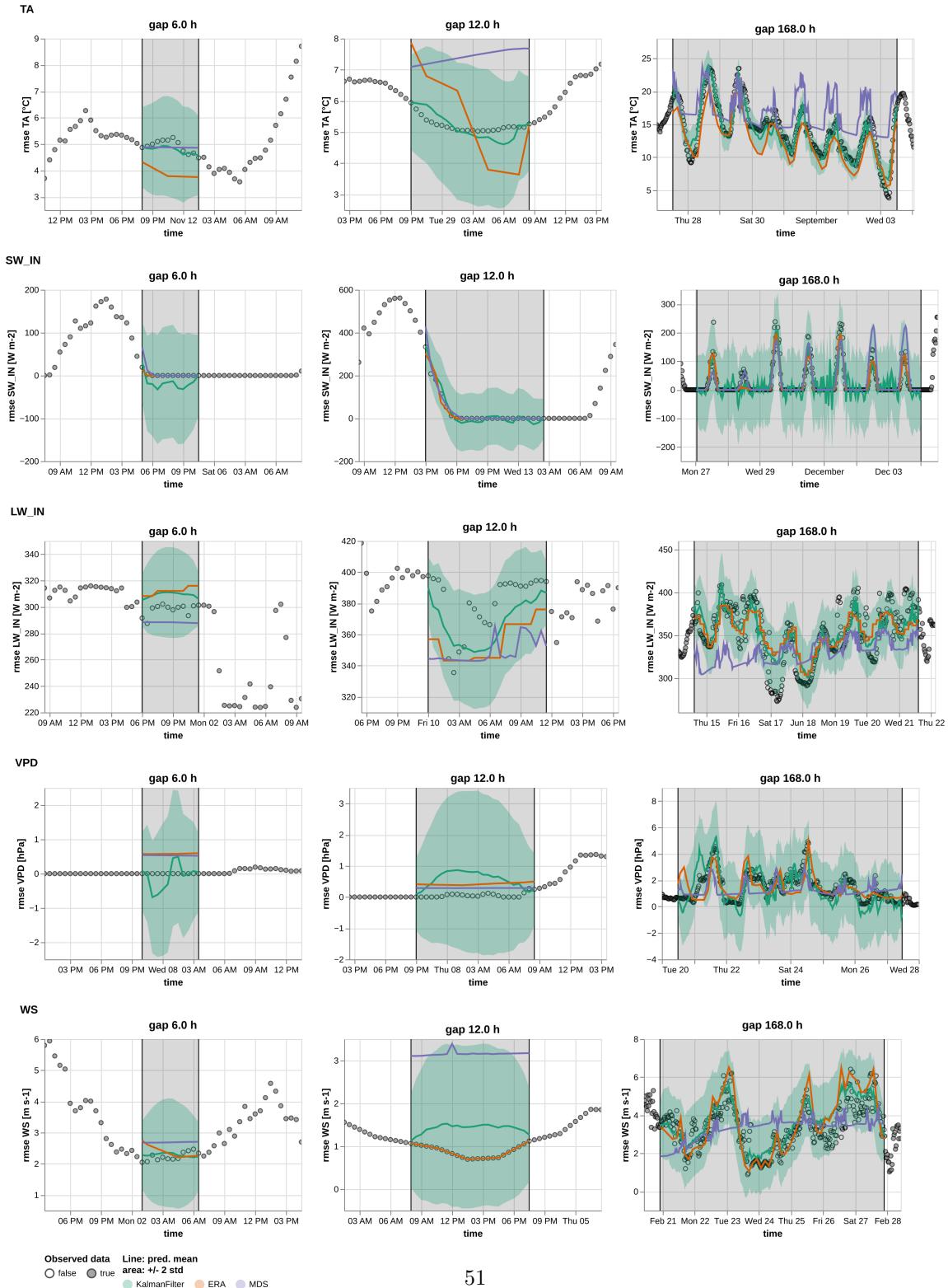


Figure 13: Example time series 2

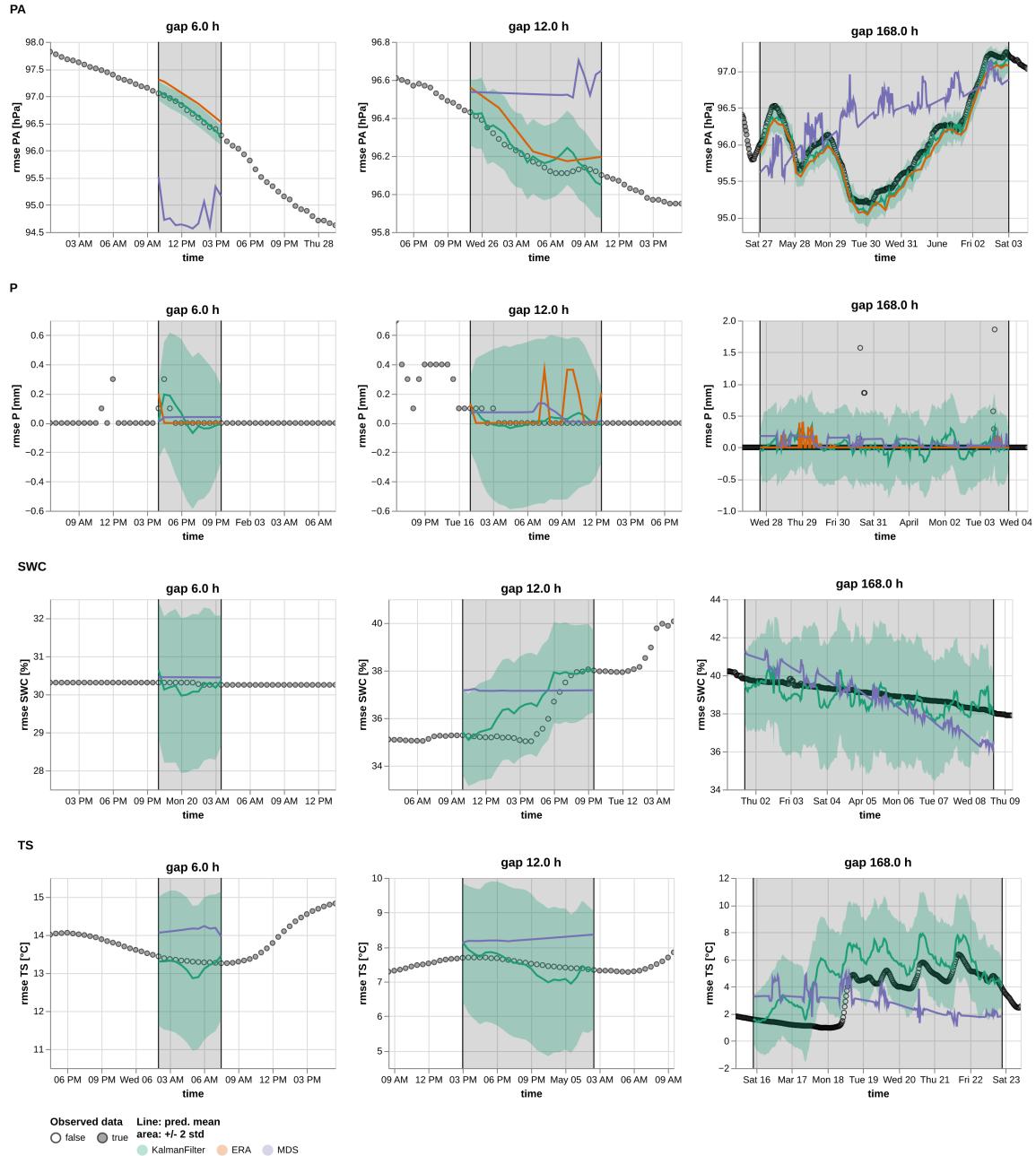
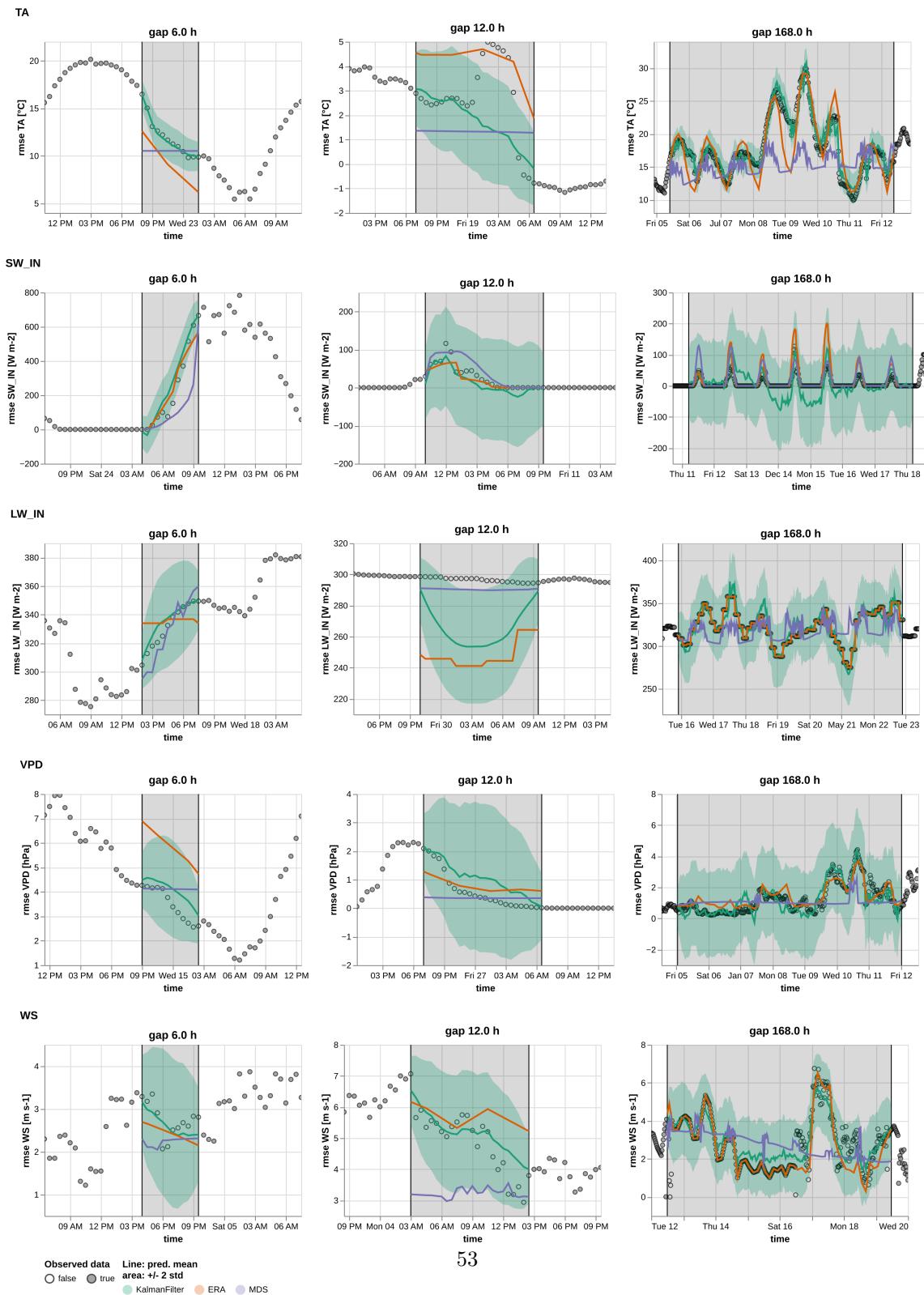


Figure 14: Example time series 2

**Figure 15:** Example time series 3

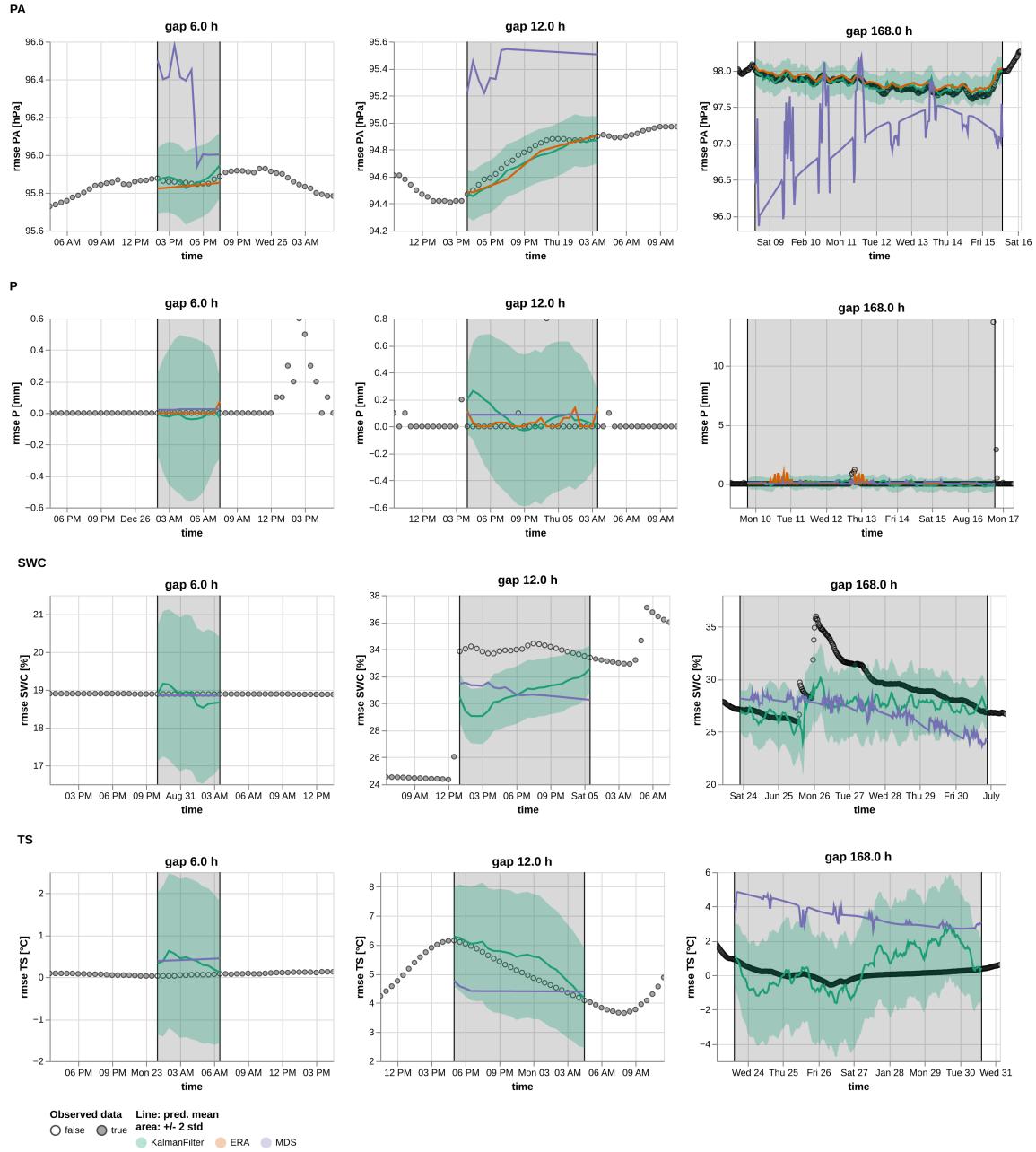


Figure 16: Example time series 3

A.3 Additional Tables

Table 5: RMSE Comparison Kalman filter for different gap lengths

Variable	RMSE	1	3	6	12	24	48	72	168
TA	mean	0.194	0.287	0.440	0.628	0.826	0.904	0.939	0.986
	std	0.170	0.192	0.319	0.394	0.441	0.449	0.408	0.348
SW_IN	mean	27.869	37.737	44.486	50.090	57.306	58.099	58.938	59.045
	std	40.675	38.361	38.011	34.548	29.070	25.845	26.383	25.259
LW_IN	mean	5.628	7.948	11.455	13.292	15.916	17.051	16.284	17.491
	std	6.613	7.565	8.073	8.415	8.775	8.078	7.010	7.286
VPD	mean	0.180	0.319	0.501	0.688	0.946	0.957	1.062	1.105
	std	0.204	0.358	0.485	0.608	0.686	0.624	0.609	0.602
WS	mean	0.361	0.495	0.616	0.770	0.830	0.906	0.917	0.914
	std	0.324	0.302	0.373	0.436	0.459	0.397	0.339	0.301
PA	mean	0.022	0.034	0.048	0.053	0.059	0.068	0.066	0.070
	std	0.018	0.023	0.048	0.030	0.040	0.063	0.051	0.050
P	mean	0.092	0.165	0.116	0.200	0.186	0.208	0.215	0.238
	std	0.273	0.539	0.158	0.344	0.204	0.191	0.186	0.151
SWC	mean	0.187	0.332	0.453	0.617	0.752	0.910	0.959	1.460
	std	0.208	0.293	0.295	0.433	0.580	0.667	0.677	0.901
TS	mean	0.152	0.232	0.382	0.591	0.677	0.995	1.358	1.663
	std	0.154	0.207	0.428	0.829	0.589	0.816	1.028	1.060

Variable	RMSE	Gap Standardized Gap [h]	Only one var		All variables		diff.
			mean	std	mean	std	
TA	3	0.029	0.024	0.049	0.048	0.020	
	6	0.040	0.028	0.078	0.052	0.039	
	12	0.073	0.050	0.129	0.083	0.056	
	15	0.079	0.057	0.138	0.092	0.059	
SW_IN	3	0.199	0.205	0.202	0.244	0.003	
	6	0.222	0.171	0.228	0.215	0.006	
	12	0.260	0.167	0.284	0.221	0.024	
	15	0.277	0.166	0.307	0.217	0.030	
LW_IN	3	0.194	0.169	0.187	0.174	-0.007	
	6	0.276	0.216	0.288	0.234	0.012	
	12	0.325	0.197	0.328	0.223	0.002	
	15	0.337	0.188	0.345	0.212	0.008	
VPD	3	0.070	0.070	0.105	0.105	0.035	
	6	0.103	0.097	0.161	0.141	0.059	
	12	0.155	0.116	0.233	0.193	0.078	
	15	0.183	0.134	0.286	0.227	0.103	
WS	3	0.297	0.179	0.301	0.181	0.004	
	6	0.366	0.202	0.371	0.211	0.005	
	12	0.460	0.242	0.468	0.248	0.009	
	15	0.481	0.230	0.494	0.249	0.013	
PA	3	0.024	0.016	0.029	0.020	0.005	
	6	0.036	0.027	0.041	0.032	0.005	
	12	0.057	0.074	0.064	0.091	0.007	
	15	0.061	0.090	0.064	0.099	0.003	
P	3	0.290	0.450	0.283	0.460	-0.007	
	6	0.386	0.524	0.396	0.592	0.010	
	12	0.390	0.550	0.393	0.640	0.003	
	15	0.442	0.675	0.449	0.767	0.008	
SWC	3	0.013	0.032	0.013	0.033	0.000	
	6	0.017	0.025	0.022	0.030	0.004	
	12	0.026	0.034	0.032	0.036	0.006	
	15	0.034	0.040	0.041	0.053	0.007	
TS	3	0.024	0.023	0.031	0.036	0.007	
	6	0.046	0.044	0.065	0.055	0.019	
	12	0.083	0.088	0.106	0.112	0.023	
	15	0.114	0.114	0.138	0.137	0.024	

Table 6: Comparison between generic model with control variable (Use Control) and generic model without control variable (No Control). 50 samples for each variable and each gap length. The best result for each for each gap length is highlighted in bold

Variable	type RMSE Standardized Gap [h]	Use Control		No Control		diff.
		mean	std	mean	std	
TA	6	0.092	0.056	0.095	0.059	0.002
	12	0.118	0.074	0.161	0.107	0.042
	24	0.147	0.077	0.233	0.159	0.086
	168	0.174	0.064	0.294	0.148	0.120
SW_IN	6	0.256	0.165	0.331	0.271	0.075
	12	0.315	0.165	0.519	0.320	0.204
	24	0.339	0.129	0.609	0.277	0.270
	168	0.358	0.103	0.693	0.248	0.335
LW_IN	6	0.299	0.201	0.330	0.185	0.031
	12	0.368	0.197	0.486	0.193	0.118
	24	0.400	0.176	0.561	0.187	0.161
	168	0.466	0.141	0.666	0.161	0.200
VPD	6	0.138	0.105	0.179	0.139	0.041
	12	0.227	0.184	0.321	0.263	0.094
	24	0.273	0.207	0.408	0.297	0.135
	168	0.325	0.154	0.499	0.252	0.174
WS	6	0.572	0.523	0.430	0.262	-0.142
	12	0.660	0.409	0.680	0.392	0.020
	24	0.676	0.374	0.767	0.430	0.091
	168	0.782	0.265	0.889	0.318	0.107
PA	6	0.099	0.066	0.145	0.105	0.045
	12	0.116	0.061	0.339	0.210	0.222
	24	0.137	0.076	0.563	0.379	0.426
	168	0.159	0.068	0.880	0.440	0.721
P	6	0.542	0.991	0.508	0.974	-0.035
	12	0.627	1.035	0.605	1.025	-0.022
	24	0.628	0.575	0.614	0.584	-0.014
	168	0.881	0.573	0.861	0.597	-0.020
SWC	6	0.152	0.107	0.151	0.141	-0.001
	12	0.245	0.154	0.305	0.234	0.061
	24	0.393	0.223	0.538	0.338	0.145
	168	0.719	0.314	0.842	0.431	0.123
TS	6	0.168	0.115	0.119	0.076	-0.049
	12	0.226	0.135	0.203	0.133	-0.023
	24	0.286	0.152	0.285	0.174	-0.001
	168	0.362	0.176	0.359	0.191	-0.003

Variable	RMSE	type Standardized Gap [h]	Generic		Finetuned one var		diff.
			mean	std	mean	std	
TA	6		0.092	0.056	0.055	0.033	-0.037
	12		0.118	0.074	0.077	0.052	-0.041
	24		0.147	0.077	0.101	0.059	-0.046
	168		0.174	0.064	0.129	0.048	-0.045
SW_IN	6		0.256	0.165	0.209	0.167	-0.047
	12		0.315	0.165	0.258	0.164	-0.057
	24		0.339	0.129	0.281	0.132	-0.059
	168		0.358	0.103	0.298	0.119	-0.060
LW_IN	6		0.299	0.201	0.257	0.200	-0.043
	12		0.368	0.197	0.326	0.209	-0.041
	24		0.400	0.176	0.359	0.186	-0.042
	168		0.466	0.141	0.406	0.165	-0.060
VPD	6		0.138	0.105	0.097	0.089	-0.041
	12		0.227	0.184	0.169	0.165	-0.058
	24		0.273	0.207	0.202	0.160	-0.070
	168		0.325	0.154	0.264	0.153	-0.061
WS	6		0.572	0.523	0.365	0.201	-0.206
	12		0.660	0.409	0.482	0.265	-0.178
	24		0.676	0.374	0.511	0.270	-0.166
	168		0.782	0.265	0.569	0.190	-0.212
PA	6		0.099	0.066	0.054	0.032	-0.045
	12		0.116	0.061	0.060	0.031	-0.056
	24		0.137	0.076	0.072	0.069	-0.065
	168		0.159	0.068	0.081	0.054	-0.078
P	6		0.542	0.991	0.542	0.991	0.000
	12		0.627	1.035	0.627	1.035	0.000
	24		0.628	0.575	0.628	0.575	0.000
	168		0.881	0.573	0.881	0.573	0.000
SWC	6		0.152	0.107	0.051	0.034	-0.102
	12		0.245	0.154	0.067	0.043	-0.177
	24		0.393	0.223	0.079	0.053	-0.314
	168		0.719	0.314	0.169	0.104	-0.550
TS	6		0.168	0.115	0.065	0.068	-0.103
	12		0.226	0.135	0.088	0.087	-0.138
	24		0.286	0.152	0.129	0.128	-0.157
	168		0.362	0.176	0.317	0.218	-0.046

Variable	Train RMSE	Standardized Gap [h]	Multi vars		Only one var		Random params	
			mean	std	mean	std	mean	std
TA	3	0.029	0.024	0.064	0.046	0.063	0.051	
	6	0.048	0.041	0.095	0.073	0.087	0.067	
	12	0.072	0.058	0.126	0.083	0.115	0.090	
	15	0.075	0.048	0.127	0.069	0.116	0.074	
SW_IN	3	0.189	0.179	0.211	0.183	0.255	0.211	
	6	0.232	0.183	0.273	0.185	0.303	0.230	
	12	0.274	0.179	0.313	0.171	0.345	0.217	
	15	0.262	0.148	0.303	0.148	0.326	0.168	
LW_IN	3	0.209	0.174	0.226	0.186	0.289	0.198	
	6	0.272	0.192	0.304	0.204	0.356	0.228	
	12	0.335	0.205	0.365	0.217	0.403	0.234	
	15	0.349	0.194	0.403	0.217	0.400	0.215	
VPD	3	0.066	0.067	0.104	0.086	0.151	0.128	
	6	0.098	0.075	0.144	0.093	0.187	0.134	
	12	0.158	0.136	0.211	0.157	0.229	0.177	
	15	0.177	0.134	0.226	0.143	0.243	0.161	
WS	3	0.305	0.160	0.499	0.557	0.460	0.302	
	6	0.375	0.197	0.623	0.841	0.492	0.270	
	12	0.475	0.246	0.682	0.523	0.543	0.285	
	15	0.478	0.251	0.653	0.403	0.547	0.293	
PA	3	0.024	0.020	0.071	0.051	0.063	0.050	
	6	0.035	0.021	0.095	0.067	0.088	0.072	
	12	0.051	0.028	0.125	0.084	0.116	0.079	
	15	0.057	0.044	0.130	0.086	0.131	0.102	
P	3	0.299	0.556	0.380	1.157	0.331	0.554	
	6	0.350	0.516	0.458	0.642	0.386	0.605	
	12	0.426	0.605	0.560	0.644	0.431	0.679	
	15	0.423	0.530	0.625	0.786	0.441	0.633	
SWC	3	0.011	0.014	0.094	0.077	0.118	0.096	
	6	0.017	0.028	0.152	0.117	0.154	0.121	
	12	0.030	0.041	0.243	0.179	0.244	0.174	
	15	0.034	0.043	0.302	0.195	0.272	0.221	
TS	3	0.029	0.039	0.112	0.084	0.054	0.059	
	6	0.043	0.046	0.160	0.148	0.080	0.089	
	12	0.089	0.098	0.236	0.158	0.127	0.124	
	15	0.106	0.071	0.269	0.155	0.135	0.099	

A.4 Gap length distribution in FLUXNET

The entire FLUXNET 2015 dataset was used to compute the distribution of gap lengths across the all the sites for each variable. A gap was definite when the QC flag of the variable is different from 0 or the data itself is missing. Figure 17 shows the complete distribution of the gaps, while figure 18 focuses only on gaps shorter than a week.

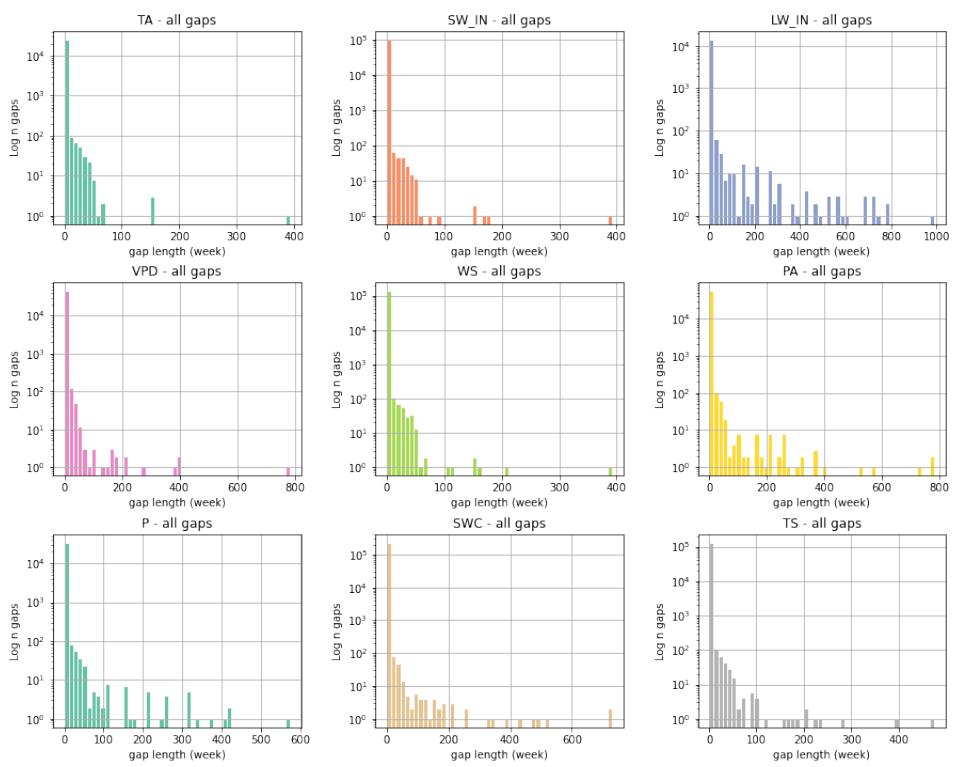


Figure 17: FLUXNET gap len

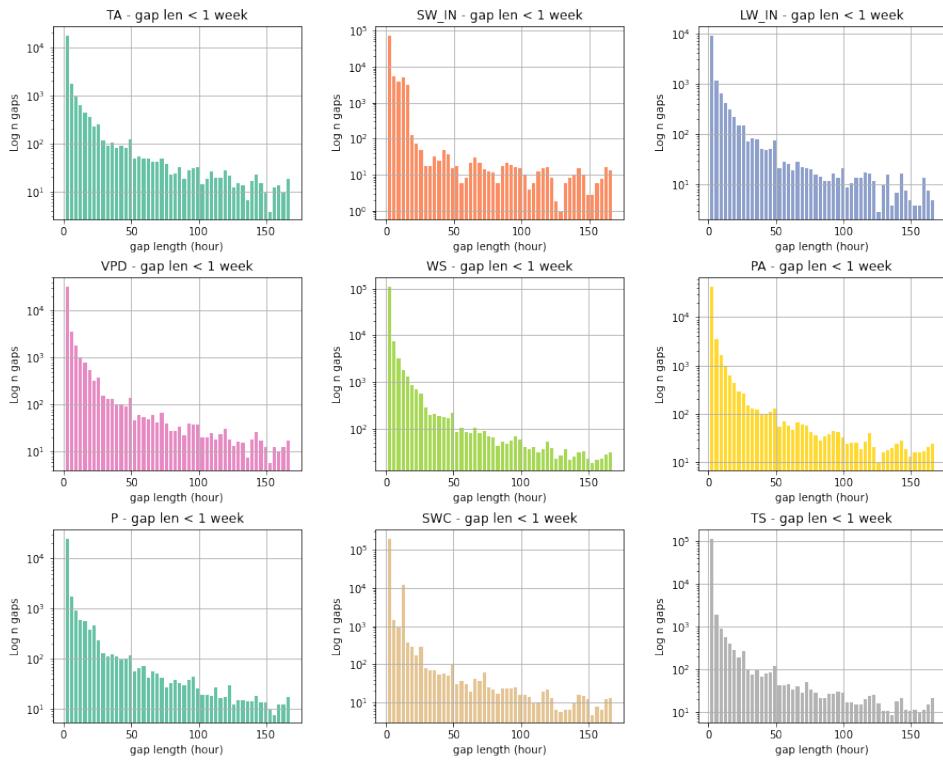


Figure 18: FLUXNET gap len less than a week

B Comparison between Standard and Square Root Kalman Filter

Standard Filter vs Square Root Filter (Mean Absolute Error of state cavariances)

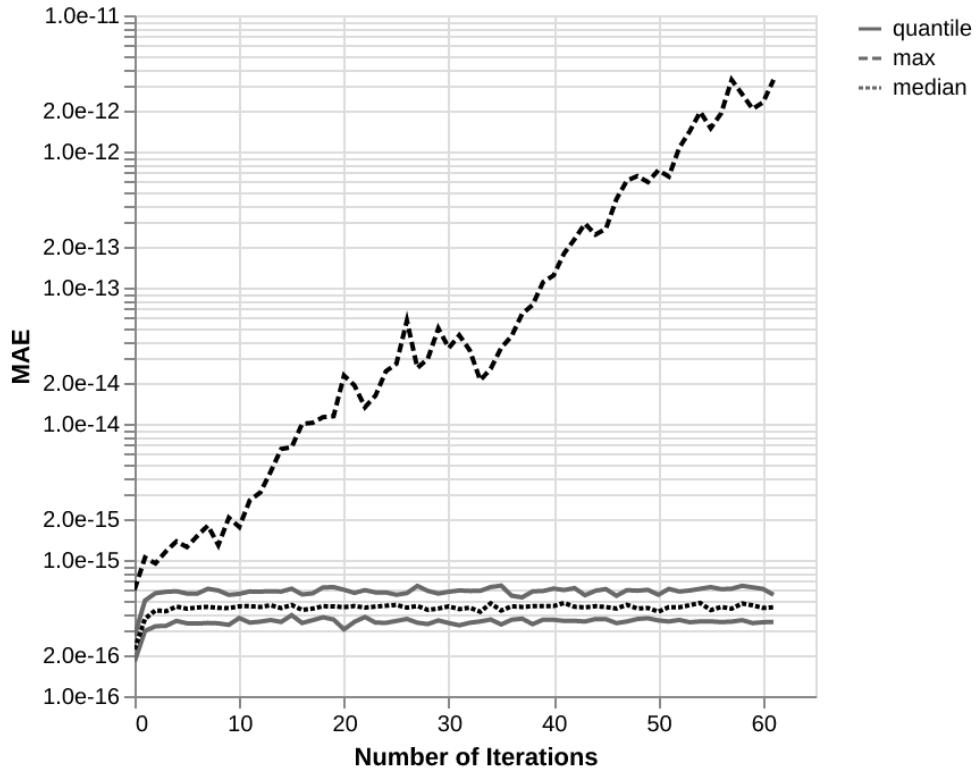


Figure 19: Numerical stability comparison between standard Kalman Filter implementation and Square Root Filter. For 100 times the filter has been initialized with random parameters (drawn from a uniform distribution range 0-1) and then filtered 100 observations. At each filter iteration the Mean Absolute Error (MAE) was calculated between the state covariance from the standard filter and the square root filter. The plot shows the median, 1 and 3 quartile and the maximum of the MAE across the 100 samples.

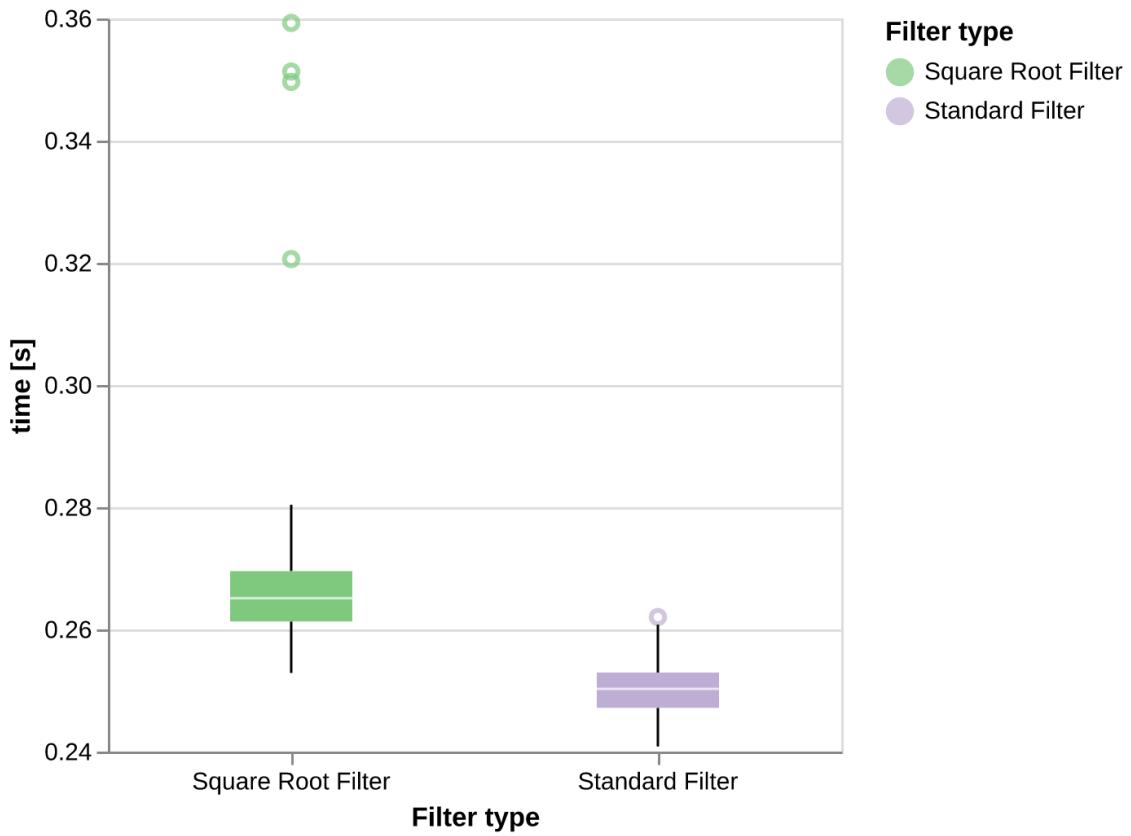


Figure 20: Performance comparison between standard Kalman Filter implementation and Square Root Filter. 100 samples with the following settings: Number of observations: 100, dimension observations 4, dimension state: 3, dimension control: 3, batch size: 5. Data and parameters are randomly generated.