02417 Times Series Analysis - Assignment 4

Authors:
- Jeppe Klitgaard <s250250@dtu.dk>
- Yunis Wirkus <s250700@dtu.dk>

Date: 2025-05-23

1 Parameter Estimation in State-Space Model

We are given a state-space model represented by the scalar system:

$$X_t = aX_{t-1} + b + e_{1,t} \tag{1}$$

Where:
- a is the scalar state transition coefficient
- b is a scalar bias term
- $e_{1,t} \sim \mathcal{N}(0, \sigma_1^2)$ is a Gaussian noise term

The associated observation model is given as:

$$Y_t = X_t + e_{2,t} \tag{2}$$

Where $e_{2,t} \sim \mathcal{N}(0, \sigma_2^2)$ is another Gaussian noise term associated with observations.

1.1 Realisations of State Vector

We are asked to perform 5 independent realisations of the state-space model with parameters $a = 0.9, b = 1, \sigma_1^2 = 1$ using the initial state $X_0 = 5$. The length of each realisation is $n = 100$.

This is done using Python and NumPy, with the relevant coding being found in `1.ipynb`.

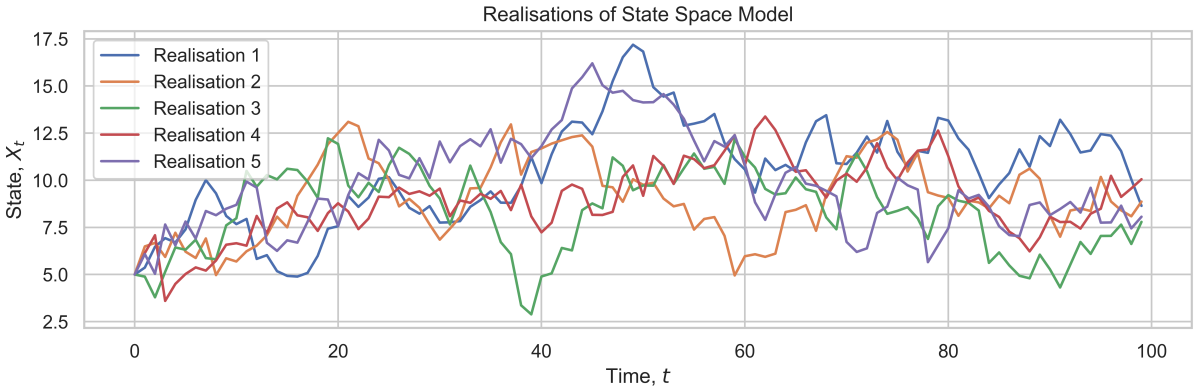

Figure 1: Realisations of the state-space model with $a = 0.9, b = 1, \sigma_1^2 = 1$ and $X_0 = 5$

The 5 independent realisations of the state as given by Eq. 1 can be seen in Figure 1. Note that we show the state vector X_t as opposed to the observation vector Y_t. While the assignment is somewhat ambiguous regarding whether the desired realisations are those of the state vector or the observation vector, we reasonably assume it to be the state vector given that the parameter σ_2 which is required to generate the observation vector is not given.

1.2 Realisation of Observation Vector

We now generate another realisation, this time also calculating the observation vector Y_t, which suffers from additional *observation noise* $e_{2,t}$. The parameters are the same as before, with the addition of $\sigma_2^2 = 1$:

$$\begin{aligned} X_t &= aX_{t-1} + b + e_{1,t} & e_{1,t} &\sim \mathcal{N}(0, \sigma_1^2) \\ Y_t &= X_t + e_{2,t} & e_{2,t} &\sim \mathcal{N}(0, \sigma_2^2 = 1), \end{aligned} \tag{3}$$

This yields Figure 2, in which we can see both the latent state vector X_t and the observation vector Y_t, which clearly is affected by the observation noise.

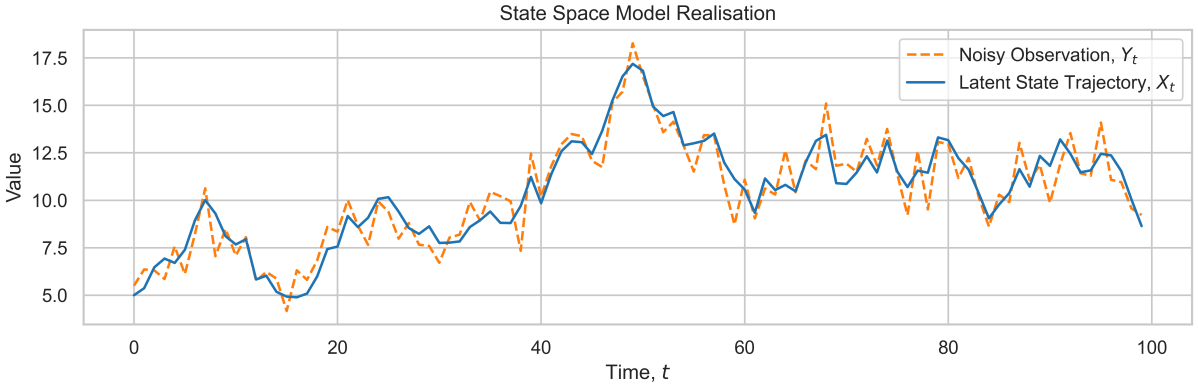

Figure 2: Realisation of the latent state vector X_t and the associated observation vector Y_t with $a = 0.9, b = 1, \sigma_1^2 = 1, \sigma_2^2 = 1$ and $X_0 = 5$

We note in Figure 2 that the observation vector appears to hover around the latent state vector with a symmetric residual, as would be expected from Eq. 3.

This corresponds to a real-world scenario where the act of *observing* something will itself introduce a noise that is independent of the underlying process.

1.3 Kalman Filter

Next we implement a simple Kalman Filter using the function template provided in `kalmanfilter.R`. We convert the function to Python and fill in the blank assignment operations as follows:

$$
\begin{aligned}
&\text{State} && \hat{X}_{1|0} = X_{\text{prior}} && (10.79)\\
&\text{State Variance} && \Sigma^{xx}_{1|0} = P_{\text{prior}} && (10.80)\\
&\text{State} && \hat{X}_{t+1|t} = a\hat{X}_{t|t} + b && (10.63)\\
&\text{State Variance} && \operatorname{Var}\big[\hat{X}_{t+1|t}\big] = \operatorname{Var}\big[\tilde{X}_{t+1|t}\big] = a^2 \operatorname{Var}\big[\hat{X}_{t|t}\big] + \sigma_1^2 && (10.54), (10.67)\\
&\text{Innovation} && \hat{Y}_{t+1|t} = C\hat{X}_{t+1|t} && (10.64)\\
&\text{Innovation Variance} && \operatorname{Var}\big[\tilde{Y}_{t+1|t}\big] = C^2 \operatorname{Var}\big[\tilde{X}_{t+1|t}\big] + R && (10.68)\\
&\text{Kalman Gain} && K_t = C\frac{\operatorname{Var}\big[\tilde{X}_{t+1|t}\big]}{\operatorname{Var}\big[\tilde{Y}_{t+1|t}\big]} && (10.75)\\
&\text{Filtered State} && \hat{X}_{t|t} = \hat{X}_{t|t-1} + K_t\big(Y_t - C\hat{Y}_{t|t-1}\big) && (10.73)\\
&\text{Filtered State Variance} && \operatorname{Var}\big[\hat{X}_{t|t}\big] = (1 - K_t)\operatorname{Var}\big[\hat{X}_{t|t-1}\big] && (10.74)
\end{aligned}
\tag{4}
$$

Where rather than regurgitating the lengthy derivations, we refer to the relevant equations in the course textbook [1, Chapt. 10].

Implementing this, we are able to use the Kalman filter on a realisation similar to that outlined in Section 1.2 and shown in Figure 2.

Helpfully, our Kalman Filter implementation already uses the latent state variance, which makes it particularly simple to compute a 95% confidence interval overlaid on the predicted state in Figure 3

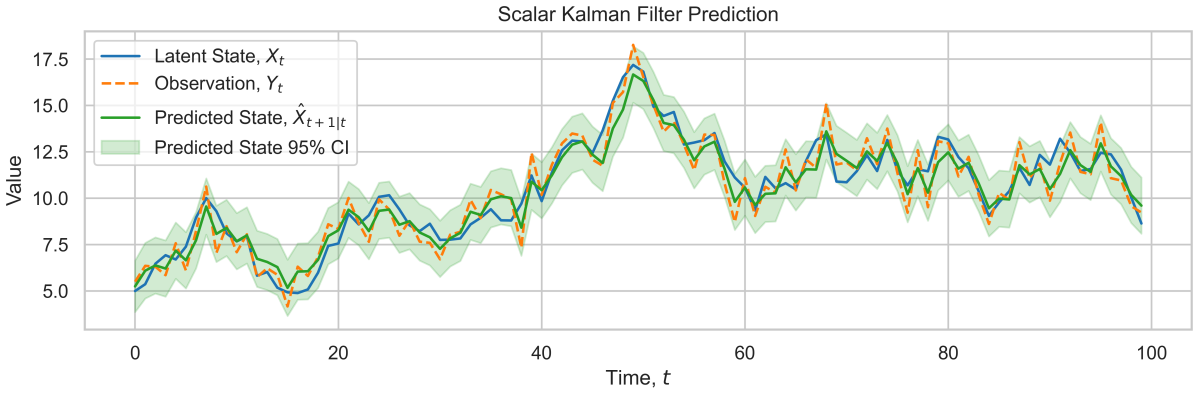

Figure 3: Scalar Kalman Filter prediction with 95% confidence intervals. We have again used parameters $a = 0.9, b = 1, \sigma_1^2 = 1, \sigma_2^2 = 1, X_0 = 5$ and use the process given in Eq. 3.

We find that the filter is able to predict the latent state using the observations well, noting that the noise variances are passed in explicitly as opposed to determined by the filter itself on the supplied data.

While further analysis may be performed, we note by inspection of Figure 3 that the confidence intervals appear to stabilise quickly after the first few observations, which is to be expected as the Kalman filter converges to the true state, which here is described by a stationary process.

The prediction intervals necessarily reach the observed size due to the inherent noise of underlying process and the observation noise. As such, the confidence intervals may not be reduced further by using a more complex model, as they are an irreducible property of the underlying process.

They may, however, be reduced somewhat by decreasing the observation noise, which may be possible in a real-world scenario by improving the experimental design, for example by using more precise expermental equipment.

1.4 Maximum Likelihood Framework

We now move into the Maximum Likelihood Framework, in which we formulate a *likelihood function* over the prediction of the Kalman Filter.

Building on the approach and functions derived in Section 1.3 and using the template function provided in `myLogLikFun.R`, we seek a Python implementation of the log-likelihood function.

By using the prior and posterior distributions of the state vector, we are able to derive the likelihood distribution as [1, Sec. 10.3.3]:

$$\begin{aligned} L = \left(\tilde{Y}_{t+1|t} \,|X_{t+1}, y_t\right)) &\sim \mathcal{N}\big(C\big(X_{t+1} - a\hat{X}_{t|t} - b\big), \sigma_2\big) \\ &\sim \mathcal{N}\big(\hat{X}_{t+1|t+1}, \sigma_2\big) \\ &= \frac{1}{\sigma_2\sqrt{2\pi}} \exp\left(-\frac{\left(Y_{t+1} - \hat{X}_{t+1|t+1}\right)^2}{\sigma^2}\right) \end{aligned} \tag{5}$$

From which we are able to construct the log-likelihood:

$$\log L = -\log(\sigma_2) - \frac{\log(2\pi)}{2} - \frac{\left(Y_{t+1} - \hat{X}_{t+1|t+1}\right)^2}{\sigma_2^2} \tag{6}$$

Using this, we are able to estimate parameters using the Kalman Filter framework through the maximum likelihood formulation based on observations.

The associated code may be found in the attached Jupyter Notebook named `1.ipynb`.

Using the implementations from previous sections, we construct $N = 100$ realisations of the observation vector with $n = 100$ samples in each and subsequently employ SciPy's optimisation suite to find parameters that maximize the log-likelihood function through minimisation of the negative log-likelihood.

Summary statistics are then calculated on the obtained parameter estimates to give insight into the estimation process.

We perform a round of simulations and estimations with parameters $a = 1, b = 0.9, \sigma_1 = 1$, which yields Figure 4.

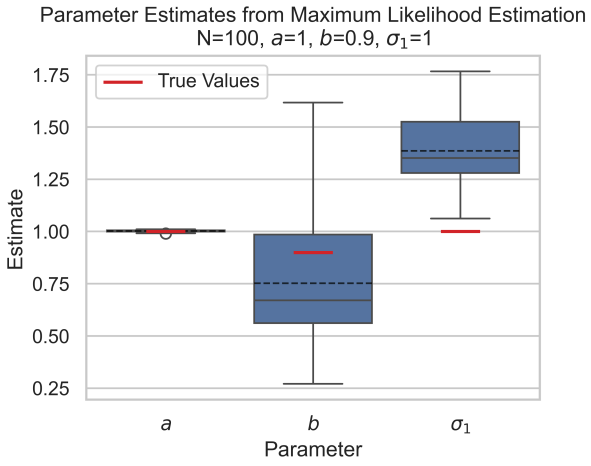

Figure 4: Parameter estimates for a, b, σ_1^2 using the Kalman Filter and maximum likelihood estimation with $a = 1, b = 0.9, \sigma_1^2 = 1$

We perform another two rounds of these simulations to produce Figure 5 and Figure 6, which show the parameter estimates for a, b, σ_1^2 using the Kalman Filter and maximum likelihood estimation with $a = 0.9, b = 0.9, \sigma_1^2 = 1$ and $a = 1, b = 0.9, \sigma_1^2 = 5$, respectively.

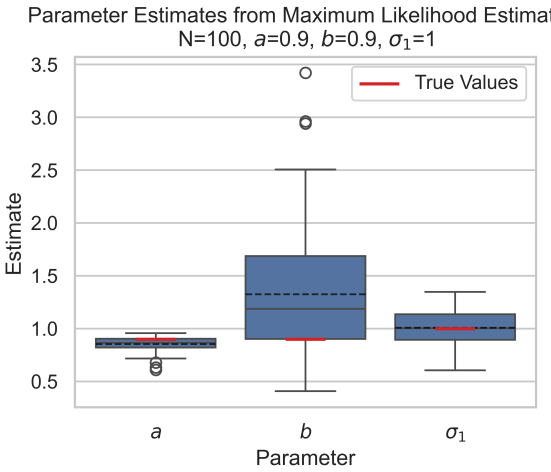

Figure 5: Parameter estimates for a, b, σ_1^2 using the Kalman Filter and maximum likelihood estimation with $a = 0.9, b = 0.9, \sigma_1^2 = 1$

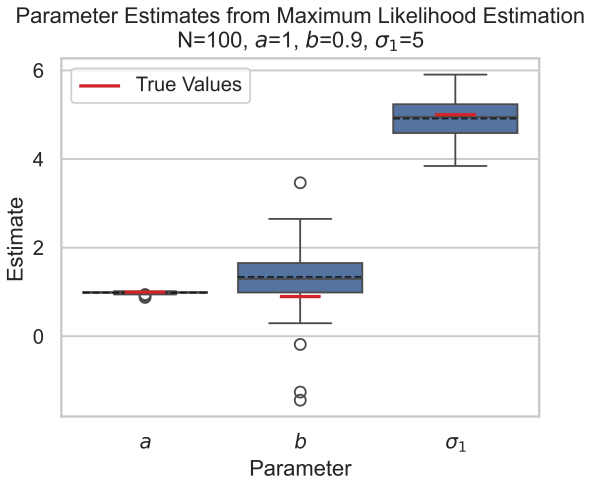

Figure 6: Parameter estimates for a, b, σ_1^2 using the Kalman Filter and maximum likelihood estimation with $a = 1, b = 0.9, \sigma_1^2 = 5$

Note that $a = 0.9$ was used for Figure 5 as opposed to the $a = 5$ given in the assignment, which was reportedly a typo, as announced on the EdStem Forum by Justinas.

Comparing across the three figures, we find that the Kalman Filter does a fairly good job of estimating the parameters, although a large variance in the estimate of the bias term b is observed. We can understand this intuitively, as the observable effect of the bias term, which simply shifts the state vector and thus the observation vector, is difficult to distinguish from a shift arising from the *random walk* arising from the integration of the noise term $e_{1,t}$.

This is particularly true when the noise term is large, as in Figure 6, where we observe a wider distribution of the bias term b.

We find that for all three rounds and across all three parameters, we are able to estimate the parameters with a reasonable degree of accuracy, though we note that the distribution of the estimates is quite large and the true value is only well-approximated when averaging across multiple realisations. We find that a is the easiest parameter to estimate, while b is the most difficult, as discussed above.

Alternatively, fewer or a single realisation may be used if the number of observations N is sufficiently large.

1.5 Non-Gaussian Noise

Our Kalman Filter was derived using the assumption of Gaussian noise, which is often reasonable in practice due to the manifestation of the Central Limit Theorem. However, there are many cases where the underlying process may not be Gaussian, which we seek to explore by employing the Kalman Filter without modification on non-Gaussian processes.

To this end, we make a number of realisations using the Student's t-distribution in place of a normal distribution. Notably, we let the observation noise $e_{2,t}$ remain normally distributed, as this will often be the case in practice per the Central Limit Theorem.

As such, our new process is given as:

$$\begin{aligned} X_t &= aX_{t-1} + b + e_{1,t} \qquad & e_{1,t} &\sim \lambda_t(\nu, \sigma_1^2) \\ Y_t &= X_t + e_{2,t} & e_{2,t} &\sim \mathcal{N}(0, \sigma_2^2 = 1), \end{aligned} \tag{7}$$

Where λ_t is a Student's t-distribution with ν degrees of freedom and scale parameter σ_1^2.

To appreciate the effect of the non-Gaussian noise, we investigate the probability density function of the Student's t-distribution with varying degrees of freedom, ν, and a normal distribution, which is shown in Figure 7.

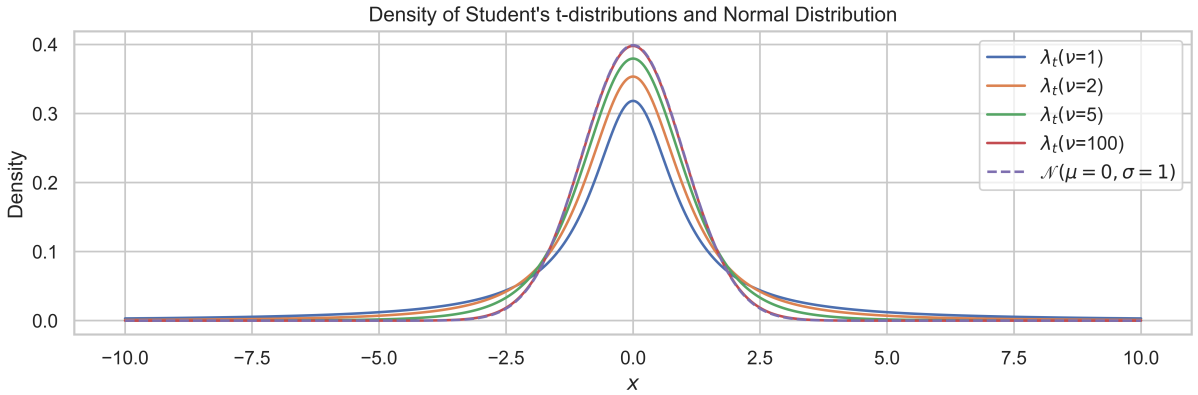

Figure 7: Probability density function of the Student's t-distribution with varying degrees of freedom $\nu \in \{1, 2, 5, 100\}$ and a standard normal distribution.

We note that as the degrees of freedom ν increases, the Student's t-distribution approaches the normal distribution as expected. For $\nu = 1$, the distribution is very heavy-tailed, which is expected to have a significant effect on the Kalman Filter. In effect, the underlying process is more likely to have noise realisations that are far from the mean, which can lead to large jumps in the latent state vector X_t.

Given that the Kalman Filter models the latent state vector as having Gaussian noise, it will be unable to account for the large number of outliers that are likely to occur in low-ν Student's t-distribution. As a result, Kalman Filter will estimate an inflated estimate of the variance of the latent state vector.

With this in mind, the Kalman Filter will still usable in many circumstances, thought with reduced performance. If the underlying process noise is known, the filter could be modified to account for the non-Gaussian noise, though this is outside the scope of this assignment.

Additionally, we note that the confidence intervals calculated in Section 1.3 are no longer valid, as they are based on the assumption of Gaussian noise.

Next, we can investigate the effect of the non-Gaussian noise on the process described by Eq. 7 by simulating realisations of the process with varying degrees of freedom ν and comparing against the Gaussian process, which is the limit $\nu = \infty$.

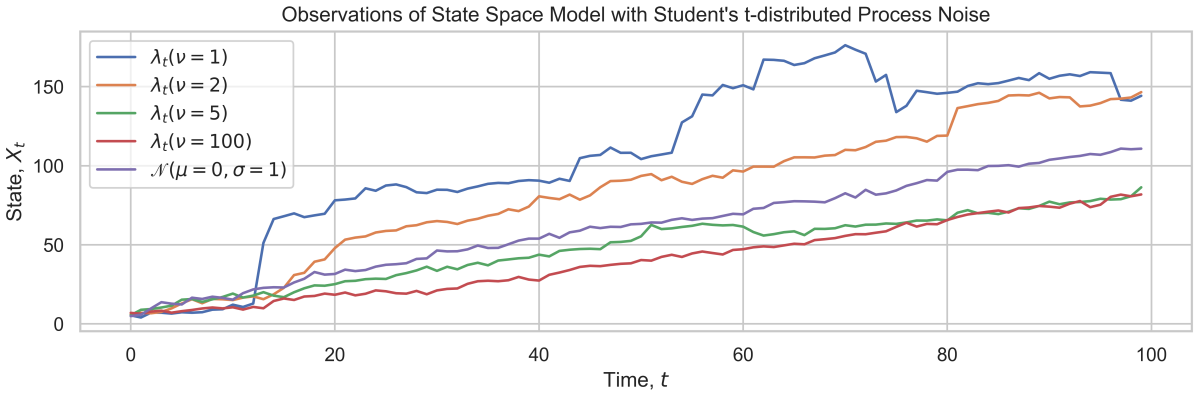

Figure 8: Observations Y_t of the processes given by Eq. 7 with t-distributed process noise. Parameters are $a = 0.9, b = 1, \sigma_1^2 = 1, \sigma_2^2 = 1$ and $X_0 = 5$

The heavy-tailed nature of the t-distribution is clearly visible in Figure 8, where we see large, sudden jumps in the observations Y_t. This is particularly pronounced for $\nu = 1$, where the process is very heavy-tailed and the observations are erratic.

1.5.a Parameter Estimation on Non-Gaussian Processes

We now perform the same parameter estimation as in Section 1.4, but using the non-Gaussian process given by Eq. 7 with $\nu = \in \{1, 2, 5, 100\}$ and remaining parameters $a = 0.9, b = 1, \sigma_1^2 = 1$. We additionally perform the estimation using the Gaussian process $\mathcal{N}(0, \sigma_1^2)$ for comparison. We simulate a total of $N = 100$ realisations of each process, with each realisation consisting of $n = 100$ observations.

The unmodified Kalman Filter is used to estimate the parameters through the maximum likelihood framework as in Section 1.4, with the results shown in Figure 9.

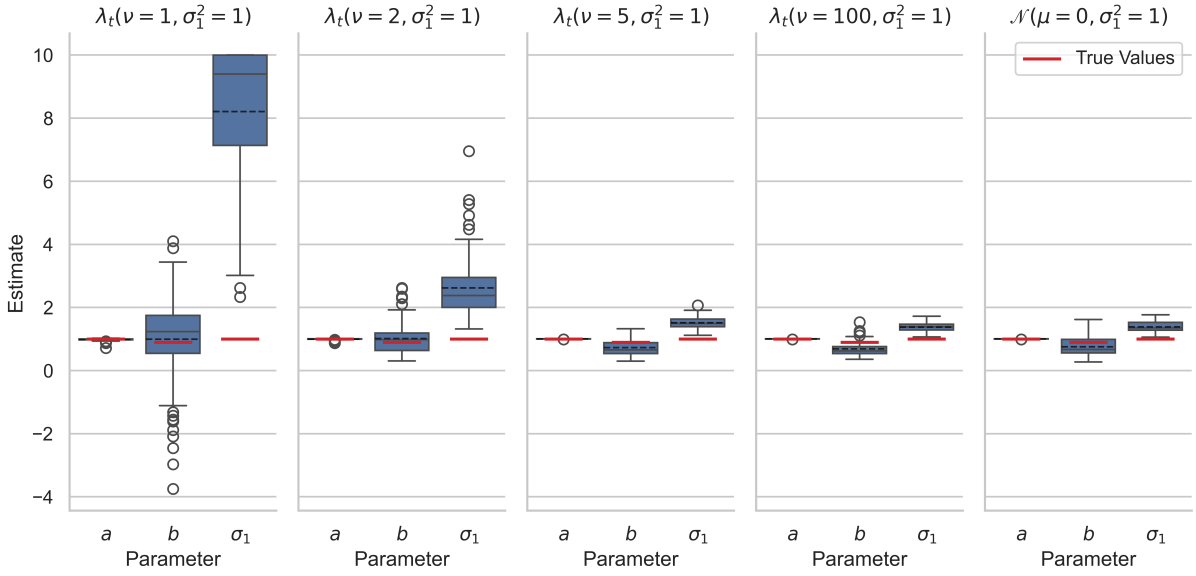

Figure 9: Parameter estimates for a, b, σ_1^2 using the Kalman Filter and maximum likelihood estimation with $a = 0.9, b = 0.9, \sigma_1^2 = 1$

As seen in Figure 9, we find reduced performance of the Kalman Filter for the very heavy-tailed cases with $\nu \in \{1, 2\}$, though the reduction in parameter estimation performance when compared against the Gaussian process is negligible for cases $\nu \geq 5$. As before, we observe good estimation of the transition coefficient a throughout all cases, but reduced accuracy on the bias term b and process noise σ_1, as expected. We observe

that the estimate of the normal distribution variance, σ_1^2 is inflated for the heavy-tailed cases, which is consistent with our earlier discussion and observations of the heavy-tailed nature of the process. Additionally, the variance of the estimate of σ_1^2 increases as the degrees of freedom ν decreases. Lastly, we find that the mean estimate of the bias term b remains accurate, but the uncertainty in this estimate increases rapidly as ν decreases.

To aide in our understanding of the predictive performance of the Kalman Filter, we can also inspect the residuals of the maximum likelihood estimation, which are shown in Figure 10.

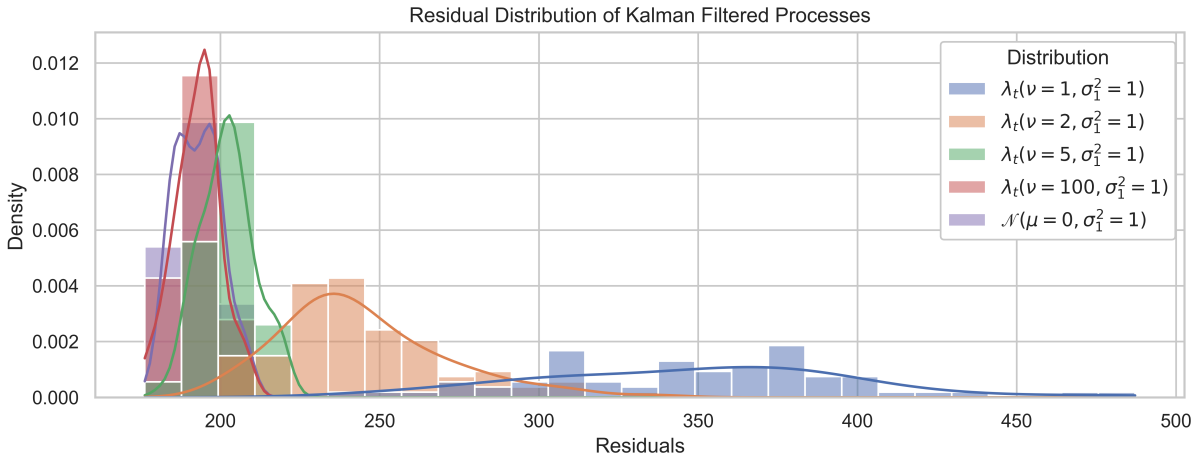

Figure 10: Distribution of the negative log-likelihood residuals from the parameter estimation of the processes given by Eq. 7 with different process noise distributions.

This confirms our earlier observations, where we see that the mean residual of the maximum likelihood estimation increases as the degrees of freedom ν decreases, corresponding to a more heavy-tailed distribution. Additionally, we observe a broadening of the distribution of the residuals, which additionally would make parameter estimation based on a small number of realisations more challenging. This is consistent with the broadening of the underlying distributions as shown in Figure 7.

In conclusion, we find that the Kalman Filter still performs well on non-Gaussian processes, though for particularly heavy-tailed processes, the performance impairment may become unacceptable. In most real-world scenarios, we would not expect the process noise to be this heavy-tailed, which explains why the Kalman Filter finds such widespread use in industry and research.

While it is often reasonable to assume that the observation noise is Gaussian due to the Central Limit Theorem, it may not be reasonable to assume the same of the process noise, $e_{\{1,t\}}$. We have demonstrated, that in most cases non-Gaussian process noise can be handled adequately by the Kalman Filter, though we note that the performance may be impaired for particularly heavy-tailed processes.

2 Modelling a Transformer Station

In this section, we will be using simple state-space models (SSMs) to gain insights into the temperature of an electrical transformer station. We are given a dataset with 168 observations as dependent variable Y_t and 3 exogenous variables $T_{a,t}, \Phi_{s,t}, \Phi_{I,t}$ which describe the outdoor temperature for the transformer in degrees Celsius, the horizontal global solar radiation at the station and the electrical load on the transformer.

2.1 Exploratory Analysis

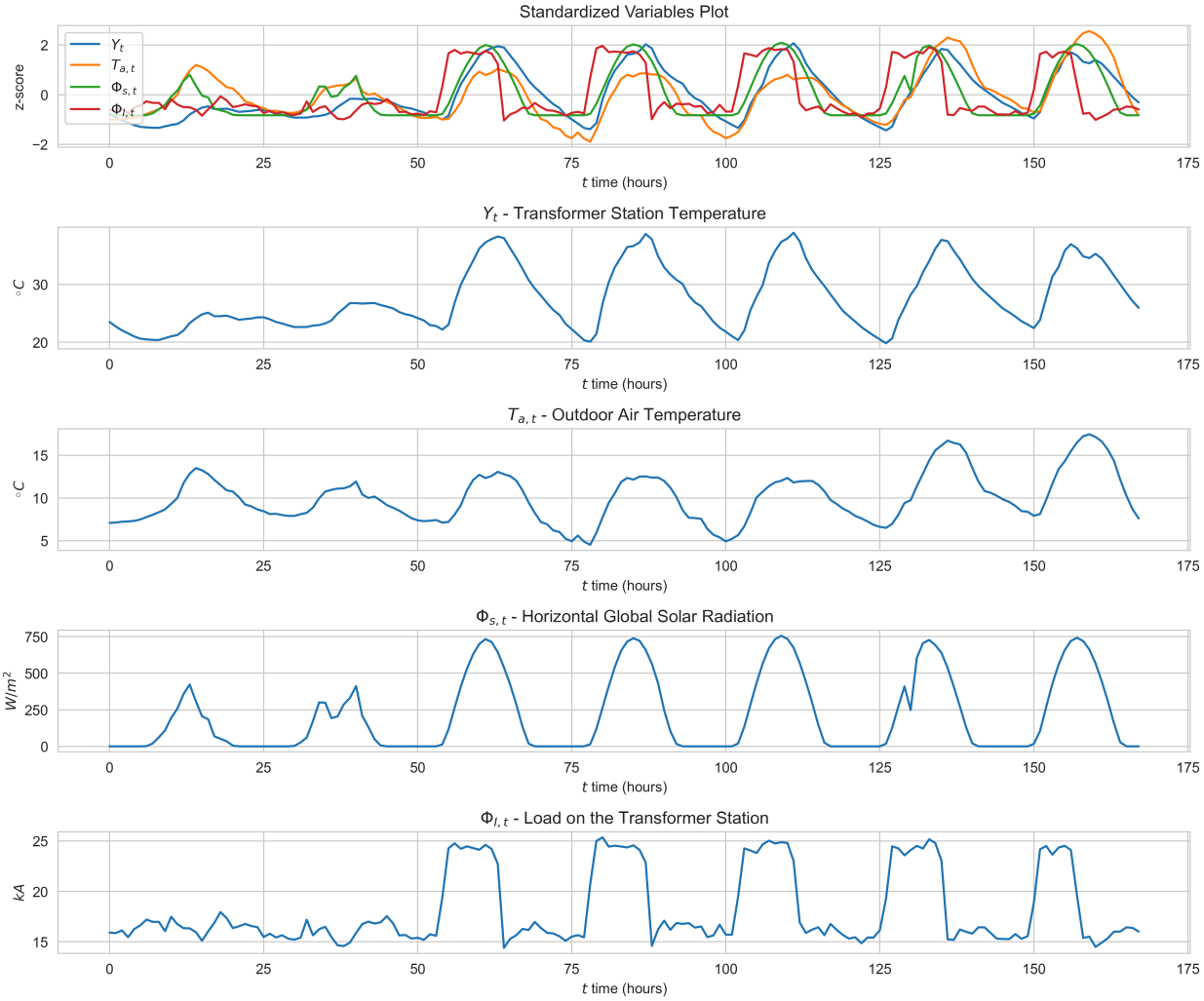

Figure 11: Given observation Y_t and exogenous variables $T_{a,t}, \Phi_{s,t}, \Phi_{I,t}$

In Figure 11, there is very clearly a seasonality for day/night-cycles in all variables. The solar radiation $\Phi_{s,t}$ drops to 0 during the night, which the load $\Phi_{I,t}$ mirrors almost exactly. It has a slightly quicker drop, once the sun is setting and during the peaks it displays a wiggle, which suggests some sort of load controller or a load maximum with excess being discharged. Lower peaks or crumples in the radiation curve could be explained by cloud cover. A curious thing to notice, is that the load on the transformer $\Phi_{I,t}$ appears to have a quicker attack-time to rise, than the solar radiation $\Phi_{s,t}$.

Intuitively, we would expect the solar radiation to lead and load to lag. The Y_t temperature follows $\Phi_{s,t}$ showing some cool-down period, once solar radiation dropped, hence a slower decay in temperature. Outdoor temperature $T_{a,t}$ not only follows the solar radiation, hence daily 24h seasonality, but also exhibits a longer period seasonality, which could be climate and wheather effects.

Physically, we can actually deduce a lot from just outdoor temperature and solar radiation cycles, especially the uninterrupted (unclouded) ones. When inspecting the graph, we can deduce about 17h of daylight, which excludes locations betwee $\approx \pm 54$ degrees N/S.

2.2 1D state-space model

The goal here is to fit (estimate the parameters) of the following model via the Kalman-Filter and MLE:

$$\begin{aligned} X_{t+1} &= aX_t + Bu_t + e_{1,t} \\ Y_t &= cX_t + e_{2,t} \end{aligned} \tag{8}$$

with

$$u_t = \left[T_{a,t}, \Phi_{s,t}, \Phi_{I,t}\right]^\top \in \mathbb{R}^{1\times 3}; \quad a \in \mathbb{R}, B \in \mathbb{R}^{1\times 3}, c \in \mathbb{R}; \quad e_{1,t} \in \mathbb{R}, e_{2,t} \in \mathbb{R}; \quad X_t \in \mathbb{R} \tag{9}$$

For the fitting, the following contraints were set: The inital value for the hidden-state value was chosen at $X_0 = 20$ and $\Sigma^{xx}_{t+1\,|\,t} = 1.0$, the parameters were initialized as $a = 0.8, B = [0.05, 0.1, 0.1]^T, c = 1, \sigma_1^2 = \log(2), \sigma_2^2 = \log(2)$. All entries of a, B, c were constrained in the interval $[-2, 2]$, while σ_1^2, σ_2^2, the variances of $e_{1,t}, e_{2,t}$ were constrained in $[1e-3, \log(10)]$.

These values were chosen somewhat arbitrarily (based on what worked well) within the boundary of the hints given in the exercise.

We fitted the model analogously to the framework introduced in Eq. 4 and Eq. 5 (for which the code can be found in the attached `2.ipynb` file). The Kalman-Filter provided predictions for the hidden-state, which where the basis for our negative log-likelihood to minimize. The resulting estimated parameters rounded to the 4th decimal digit are:

$$a = 0.7906; \quad B = [0.1313, 0.0031, 0.2524]^T; \quad c = 0.8863; \quad \sigma_1 = 1e-3; \quad \sigma_2 = 1e-3 \tag{10}$$

It was notable, that the variances σ_1^2, σ_2^2 were always pushed to the lower boundary of the given constraint, no matter the initialization. This could imply, that the model can explain the observations Y_t very well without added noise, or that the noise in the system is not of additive nature.

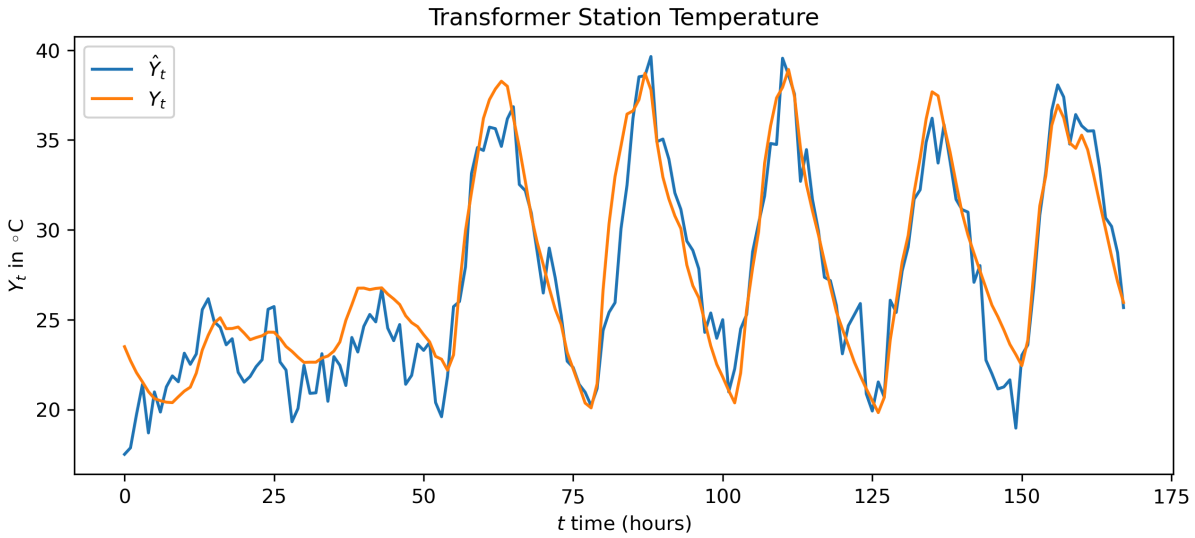

Figure 12: given observation Y_t compared to the output prediction of our Eq. 8 model, based on the parameters Eq. 10

Figure 12 shows that our simple model captures the dynamics of Y_t relatively well. It does not perform well on the first 50 hours, most likely because the data is more noisy for cloud-cover wheather conditions. Additionally, we can observe that $B_{1,3}$ is the highest coefficient in the Eq. 8 model and is the factor to exogenous variable $\Phi_{I,t}$, the load. As this variable shows the noisiest behaviour for that period, the 'culprit' is clear.

Looking at the residuals (in this case equivalent to the 'innovation') in Figure 13, we can observe that they are not fully normally distributed, which is also evident in the autocorrelation plots, where we see some statistically significant peaks at low lags, indicating some short-term correlation. Given that there are no significant correlations at higher lags, we conclude that the model does a good job of capturing the

dynamics of the system, with only very modest systematic errors associated with the larger loads during peak hours.

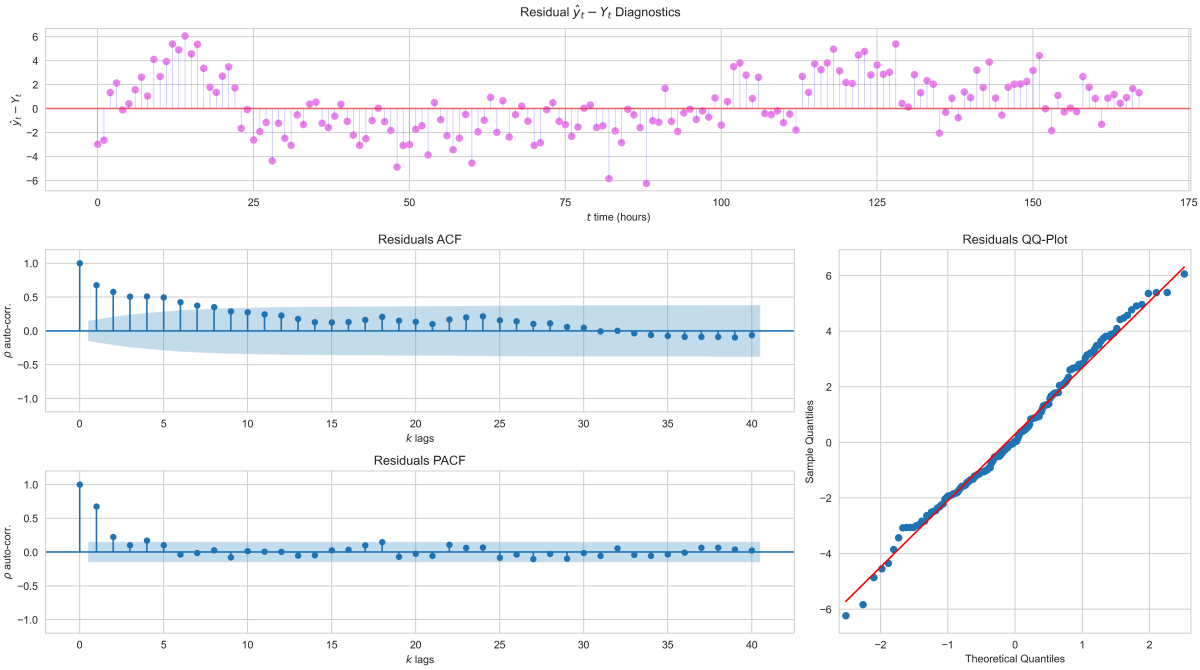

Figure 13: residual $\hat{y}_t - Y_t$ of true temperature and the output prediction of our Eq. 8 model, based on the parameters Eq. 10

Beyond, we report the AIC and BIC as model selection criteria with AIC $= 495.15$, BIC $= 517.02$. In this setting, the BIC 'advantage' of penalizing model complexity heavier than AIC has not kicked-in yet, since with $p = 7, n = 168 \Rightarrow 2p > \log(n)p$.

The physical implications and interpretations are extensively discussed in Section 2.4 for a 2D SSM. As the line of argument is analogous here for the 1D case, we will only briefly discuss the parameters.

The coefficient matrix B can be seen as the weights for a sum-composition of u_t. Consequently, we can interpret the magnitude of the coefficients as importance weight. Apparently the last value of $B_{1,3}$ is the largest (Eq. 10), which corresponds to $\Phi_{t,I}$ being weighted as the most important predictor. Via the second coefficient $B_{1,2}$, the importance of $\Phi_{t,s}$ is weighted the lowest. Most likely, because from Figure 17 we can see that those two Φ_t have a high correlation and thus share a lot of informational value for the model. Since one of these is already weighted high, there is no need to weigh the other one highly as well. $T_{t,a}$ outdoor temperature is given about half as much importance as $\Phi_{t,I}$ load.

In sum, we can interpret the physical implications as follows: The transformer load is the most significant contributor to the hidden-state X_t, but also introduces most of the noise to the system (as becomes visually clear in Figure 11). It is also leading the transformer temperature Y_t. The outdoor temperature is the second most significant contributor, as of course the environment also impacts the heating and cooling of the transformer unit. Simple the range of values of $Y_t, \Phi_{t,I}, T_{t,a}$ is somewhat similar, while $\Phi_{t,s}$ is not. Hence, it can be more directly modeled through a linear combination. Normalizing the data before fitting could have potentially mitigated this effect.

2.3 2D state-space model

The goal here is to fit (estimate the parameters) of the following extended 2D (meaning 2 hidden states) model via the Kalman-Filter and MLE:

$$\begin{aligned} X_{t+1} &= AX_t + Bu_t + e_{1,t} \\ Y_t &= CX_t + e_{2,t} \end{aligned} \tag{11}$$

with

$$u_t = \left[T_{a,t}, \Phi_{s,t}, \Phi_{I,t}\right]^\top \in \mathbb{R}^{1\times 3}; \quad A \in \mathbb{R}^{2\times 2}, B \in \mathbb{R}^{2\times 3}, C \in \mathbb{R}^{1\times 2}; \quad e_{1,t} \in \mathbb{R}^{2\times 1}, e_{2,t} \in \mathbb{R}; \quad \boldsymbol{X}_t \in \mathbb{R}^{2\times 1} \tag{12}$$

both noise terms e_t following (multivariate-)normal distributions with mean 0.

For the fitting, the following inital values were chosen:

$$\begin{gathered} X_0 = [20, 20]^T \quad \Sigma^{xx}_{t+1\,|\,t} = 10.0 \\ A = \begin{pmatrix} 0.8 & 0.1 \\ 0.0 & 0.7 \end{pmatrix} \quad B = \begin{pmatrix} -0.1 & 0.1 & 0.1 \\ 0.0 & -0.1 & 0.0 \end{pmatrix} \quad c = (1.0 \;\; 0.2) \\ \sigma_1^2 = \log(2) \quad \sigma_2^2 = \log(2) \end{gathered} \tag{13}$$

subject to the following constraints: All entries of A, B, C were constrained in the interval $[-2, 2]$, while σ_1^2, σ_2^2, the variances of $e_{1,t}, e_{2,t}$ were constrained in $[1e-3, 2]$.

These values were again chosen somewhat arbitrarily, within the ranges of the hint given in the assignment. It shall be noted, that we could achieve significantly better results, by fitting multiple times and taking the last estimated parameters as new initialisation for the next fitting. However, we decided to only have one run with intentionally 'worse' initial values, to also get an impression of the performance of the fitting procedure itself.

After fitting, we have the following estimated parameters, rounded to the 4th decimal digit:

$$\begin{gathered} A = \begin{pmatrix} -0.8303 & -0.3605 \\ 0.6865 & 0.9543 \end{pmatrix} \quad B = \begin{pmatrix} -1.7612 & 1.741 & -1.1408 \\ 0.9138 & -0.4893 & 0.8132 \end{pmatrix} \\ c = (0.264 \;\; 0.6254) \quad \sigma_1^2 = 0.001 \quad \sigma_2^2 = 0.001 \end{gathered} \tag{14}$$

Again, the variances of the noise are pushed to the lower boundary. We did initially to estimate the initial state X_0, however, even with constraints in the optimiser, the overall model yielded worse results. Yet the initial value estimates always hovered $\approx [20, 20] = X_0$, therefore we deemed it a qualified guess.

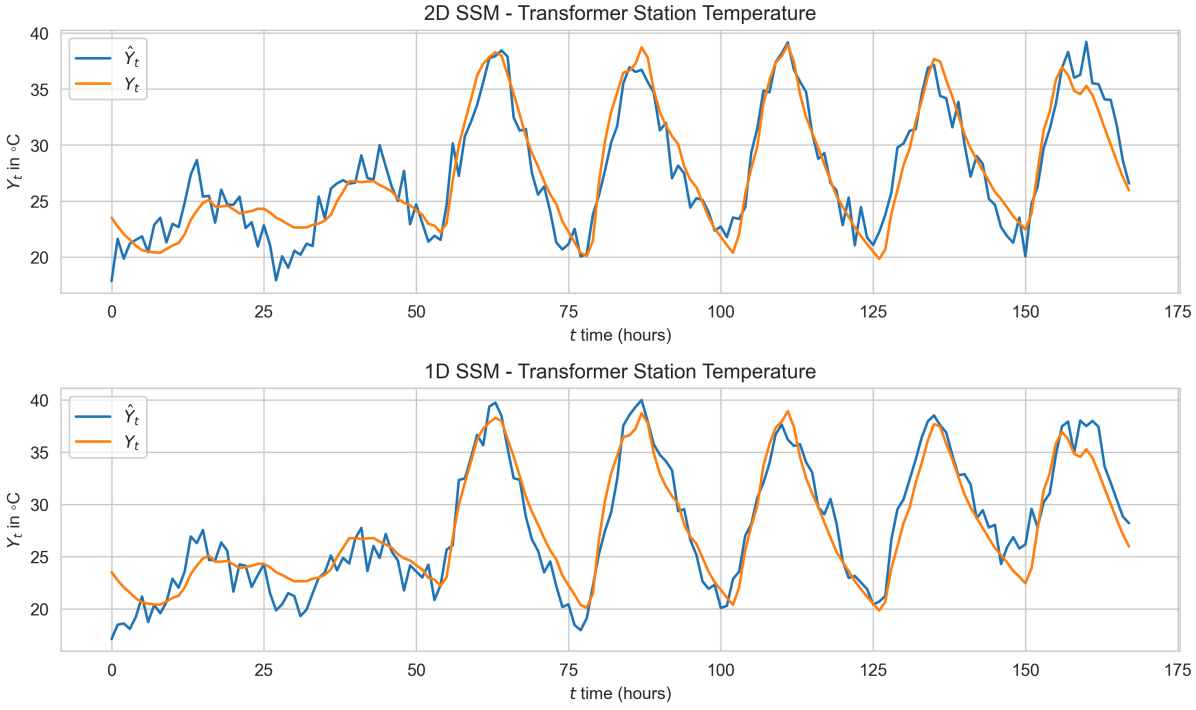

Figure 14: one-step predictions $\hat{Y}_t$ of both models Eq. 8 and Eq. 11

In Figure 14 we first look at the one-step predictions and compare the 2D to the 1D model. Visually, the performance seems approximately on par; both having significant difficulties capturing the dynamics for cloud-cover conditions and somewhat difficulties capturing peaks (both models over- and under-shooting at different peaks).

We report the information criteria statistics as AIC $= 499.23$, BIC $= 542.97$, which indicates that the 2D model does not perform significantly better. Because it is still the case, that AIC penalizes model complexity more heavily here (with $p = 14$), we can even argue that the 2D model performs slightly worse than the 1D model, compared with its complexity.

Thus, we turn to the residual analysis in Figure 15, in which we again observe some local systematic errors, in which the model systematically under- or overestimates the observations during peak hours. We find that the residuals are normally distributed at larger timescales the model performs well overall.

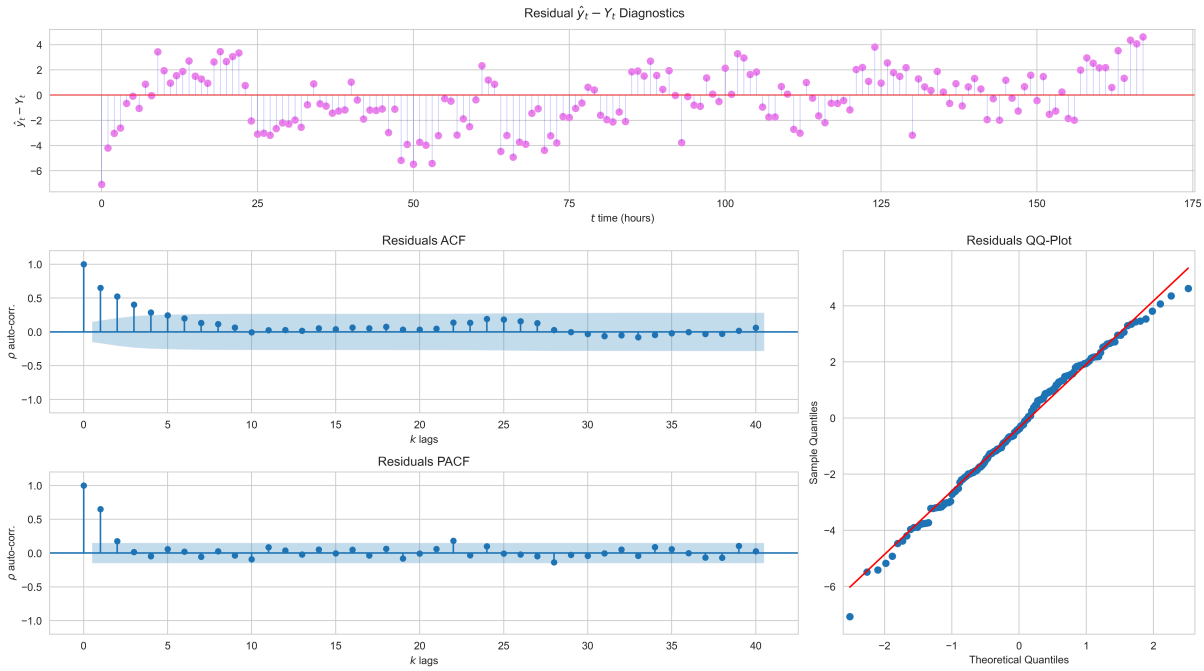

Figure 15: residual $\hat{y}_t - Y_t$ of true temperature and the output prediction of our Eq. 11 model, based on the parameters Eq. 14

The lack of performance difference between the 1D hidden-state model and the 2D hidden-state model, may suggest, that there is either no significant value addition in a 2nd hidden-state, hence, $X_{t,0}$ and $X_{t,1}$ would strongly correlate. There could also be a notion of inverse or counter-acting relationship between the two hidden-states, that nulls out or corrects any information gain of the additional second state compared to only one state. We explore this further in the next section.

2.4 2D state interpretation & discussion

The goal of this section is to inspect the estimated hidden-states of the 2D model Eq. 11 and discuss insights, that can be derived from the coefficients $A, B, C, \sigma_1, \sigma_2$ or the state time-series $\boldsymbol{X_t}$ themselves.

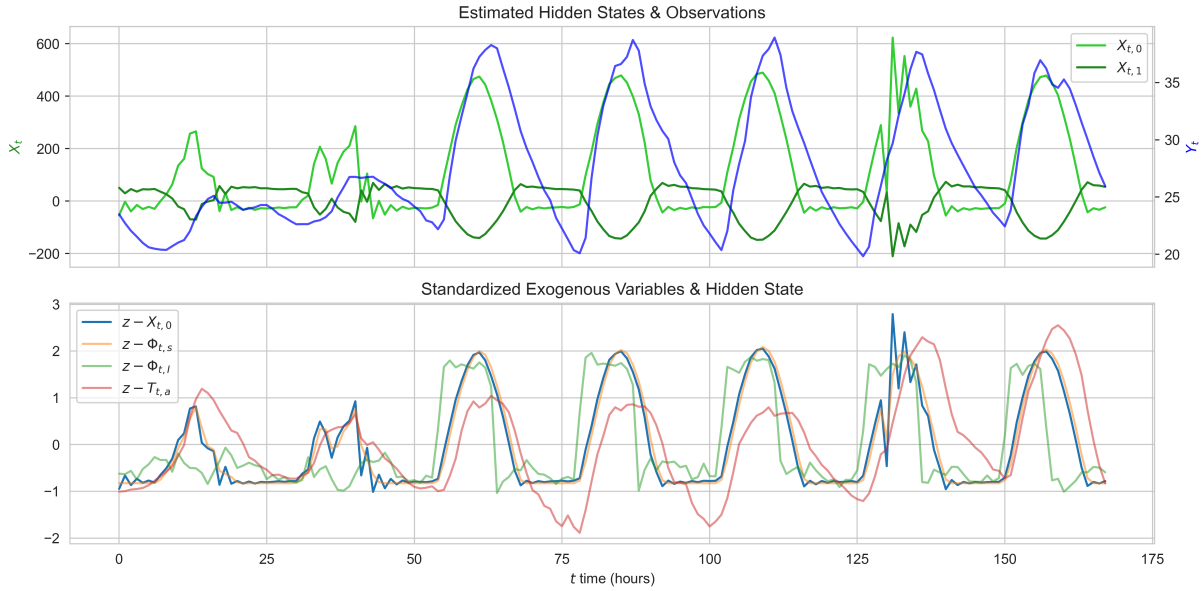

Figure 16: estimated hidden state $X_{t,0}$ (because visually closer to exog.) and exogenous variables u_t; standardized for better comparison

We start with a visual analysis of the hidden states $\boldsymbol{X_t}$ in Figure 16. We can see that both states are seasonally counter-acting, as assumed in Section 2.3. In fact, the Pearson correlation coefficient between $X_{t,0}, X_{t,1}$ is -0.9961, so almost perfect negative correlation. Thus confirms the assumption, that there is not much value added by a second hidden state. By construction, the hidden states are leading the observations in signal response (in upper graph). The value ranges for X_t are not quite insightful, as they can easily be absorbed by the magnitude of coefficients in A. Yet, the opposing nature suggests that one state is 'buffering' the other. In the bottom graph of Figure 16, we only plot one state, as already deduced that there is not much additional informational value in the second state. Overall, we know that, in principle, the hidden state is just a weighted sum (because it is a linear combination) of the exogenous variables. We can see that $X_{t,0}$ is most closely followed by $\Phi_{t,s}$ (which can be seen in Figure 17), which suggests the highest weight/coefficient assigned.

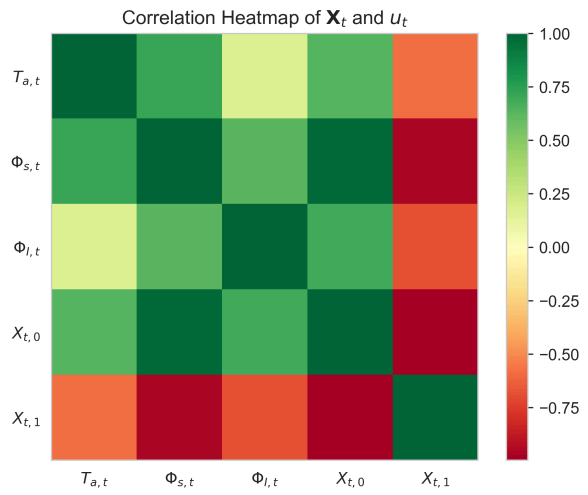

Figure 17: correlation coefficient heatmap

The heatmap Figure 17 also shows, that even the exogenous variables in u_t have high statistical correlation among themselves. This further supports the conclusion, that estimated parameters will favour one variable, as the others do not add much information to the model.

We follow with an analysis of the estimated coefficients: As mentioned above, we can interpret the coefficients in B as weights for the sum-composition of u_t for each state. The sign between the first and second row of B are flipped (c.f. Eq. 14), explaining the opposing behaviour and the 'buffering/dampening' nature. We can see in the top graph of Figure 18, that this weighted sum is somewhere in between the total sum and the average of all exogenous variables in u_t combined.

The highest weights are both in column 1 of B, the weights for $T_{t,a}$. This is likely due to the extremely different magnitude of the exogenous variables, where $T_{t,a}$ has the smalles values, so it needs to be boosted to even compete with the other exogenous variables in the weighted sum. Additionally, $T_{t,a}$ has the lowest correlation (c.f. Figure 17) with any of the other exogenous variables (or states), so it provides the most uncovered information.

A acts as the recursive state factor, also allowing for further dampening of the high magnitude of the states $\boldsymbol{X}_t$. This matrix parameter is responsible for convergence and stability of the entire SSM (c.f. Section 2.4.a). The parameter matrix C is the factor that combines both states into a prediction for the observation Y_t. It is therefore import, as it controlls the mixing of states.

Interestingly, because of the high correlations of variables in the entire SSM, in the bottom plot of Figure 18, we can observe that skipping the recursive estimation of a hidden-state entirely (by modelling $C(Bu_t)$), we still get a decent approximation of our observations Y_t. This would suggest and confirm our previous assumptions, that Y_t could possible be predicted directly by a linear model of only u_t.

To better illustrate the effects of the parameters, we visualise some time-series with the estimated states $\boldsymbol{X_t}$ and exogenous variables u_t multiplied by the parameters sequentially.

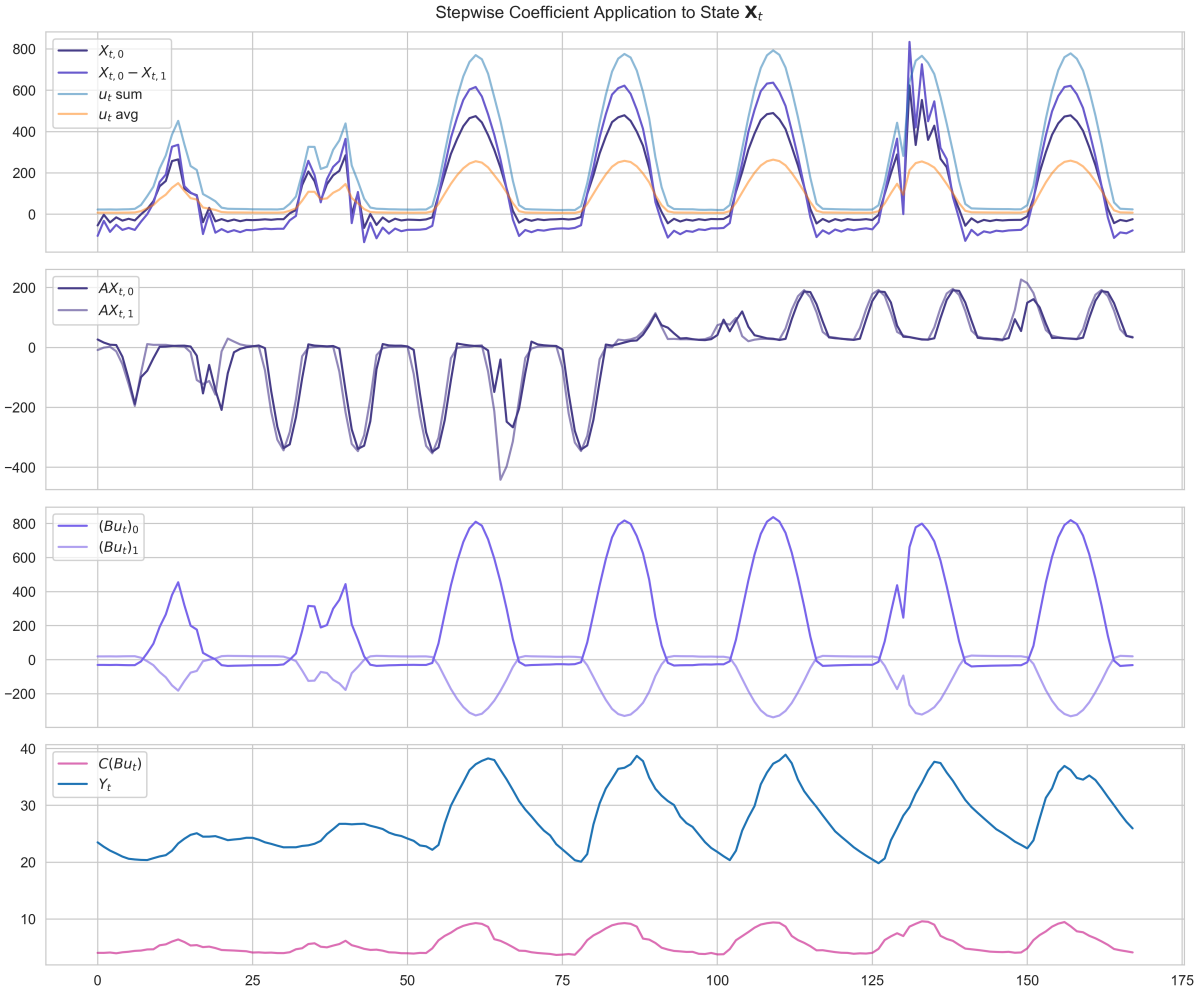

Figure 18: sequential application of estimated parameters to the states $\boldsymbol{X}_t$

The interpretation of Figure 18 is mostly conducted above. A not yet mentioned interesting insight is that the combination of $A\boldsymbol{X}_t$ switches sign in the middle of the time axis. This coincides with the slight trend notices in the residuals in Figure 13. So after all, it might be that improving the estimates of A could avoid the over- and under-shooting trend.

A physical interpretation of the states is, as mentioned, a combination of the external factors at the measurment station, u_t as predictor, mostly (c.f. Figure 17) $\Phi_{s,t}$.

One could interpret one state $X_{t,1}$ as a buffer for $X_{t,0}$ (c.f. Figure 16). Based on the above conclusions, mainly buffering/adjusting $X_{t,0}$ for the outdoor temperature $T_{t,a}$. Alas, within that line of argument, one state would represent a “cooling” and the other a “solar-radiation-load-temperature” response.

Yet, as already argued, this would not be necessary as one single state carries almost just as much information, hence this buffering effect for $T_{t,a}$ is most likely absorbed into A (not into the noise, as the noise terms both $e_{t,1}, e_{t,2} \to 0$ during the MLE).

Overall, the physical interpretation of the model makes sense. Nevertheless, it also became clear that the model is too complex for the informational value it contains/processes.

2.4.a Stability

While not asked for, this section seems important after we encountered instability in the first version of the assignment in part 1. Very briefly: For stability of an SSM we focus only on the homogenous part of the system, namely:

$$\begin{aligned} X_{t+1} &= AX_t + \dots \\ Y_t &= CX_t + \dots \end{aligned} \tag{15}$$

Assuming the exogenous part u_t is in itself is bounded and stable, it does not affect the stability of the system, it only gives a 'direction' of movement. We also assume the variance of the state equation σ_1^2 to be reasonably bounded. More specifically, within the homogenous part of the system, we look at matrix A. If the spectrum of A is contractive, in other words, if $\forall \lambda_i \in \boldsymbol{\lambda}(A) : |\lambda_i| < 1$, then the SSM is assumed to be asymptotically stable (for full details, see [2] chapter 4)

2.4.b Outlook

Having analysed two simpler SSMs, we can give an outlook on what to improve. As we have seen in this example, increasing the number of states does not significantly improve the performance of the model. Thus, we must think of other routes to improve performance.

One option is to employ Kalman-smootheners, which could help tackle the large prediction error in especially the first 50h of the expirement.

Another option would be to include an auto-regressive term in the observation equation of the model ($Y_t = \phi_1 Y_{t-1} + \dots + \phi_p Y_{t-p} + CX_t + e_{t,2}$). This could assist in better performance around the ascends and descends, as the hidden-states seem to pre-maturely drop after peeks, in some occasions. Further, as we noted in the ACF plots of Figure 13 and Figure 15, there were some significant auto-regressive terms for the residuals. Adding such components directly to the model, could help mitigate this systematic part of the error and improve normality of the residuals.

Furthermore, it could be beneficial to include another parameter-exogenous-term ($Y_t = CX_t + Du_t + e$) in the observation equation. As we saw in Figure 11, the two exogenous variables $\Phi_{s,t}, \Phi_{I,t}$ seemed to be leading in dynamics and seasonality. Re-enforcing the exogenous u_t onto the observation part, could help capture behaviour around peaks better.

Bibliography

[1] H. Madsen, *Time Series Analysis*. Chapman & Hall/CRC, 2008.

[2] J. Anderson BDO; Moore, *Optimal Filtering*. Dover Publications Inc., 1979.