Time Series Project

Axel Sjöberg, Erik Stålberg
ax3817sj-s@student.lu.se, er4047st-s@student.lu.se

January 2020

Contents

1 Introduction

2 Stationary ARMA-modelling

2.1 Data selection

Initially we wanted to select data the modelling, validation and test sets. We decided to go with a 10 week long modelling set, 2 week long validation set and two 1 week long test sets. The modelling, validation and first test sets were chosen consecutively in summer, however the second test set was chosen in winter. These data sets were chosen so as to be reasonably stationary. This can be expected to happen in summer and in winter, when the temperature is at its peak or bottom. Therefore the modelling set, validation set and first test set were chosen in the summer of 1967 and the second test set in March of 1969. Here temperatures were somewhat stationary, as can be seen in figure 1.

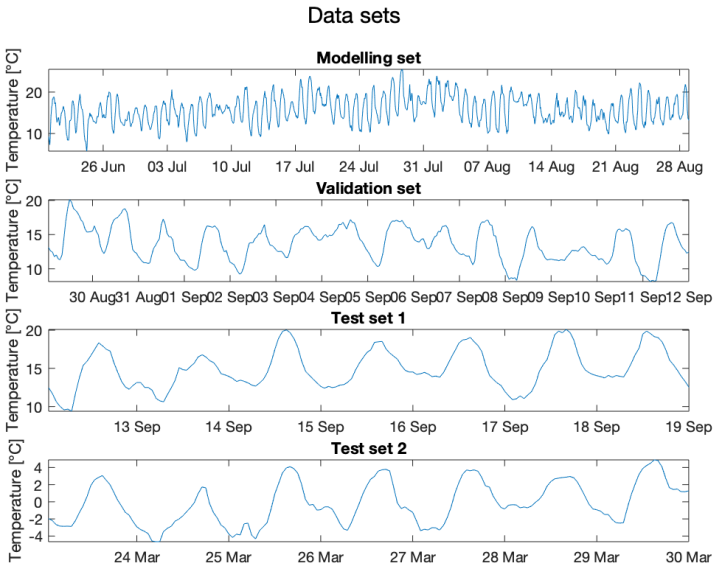

Figure 1: Temperature data sets. In degrees Celsius.

2.2 Data Analysis, Model Selection and Parameter Estimation

The data was analyzed in order to determine if it needed to be transformed or if it contained any trends. According to the Box-Cox normality plot, the log-likelihood has a maximum at $\lambda = 0.71$. This λ is closer to 0.5 than 1.0

indicating that an element wise square root transformation could improve the modelling. However, as the variance does not seem to be proportional to the mean, no transformation on the data was made. Furthermore, it is assumed that no deterministic trend is present in the data as the t-ratio is $9.56 * 10^{-26}$ and the upper $100(\alpha/2)$th percentile of the t-distribution is 3.8470 with $\alpha = 0.05$. Conclusively, no transformation is made on the data and no deterministic trends are removed.

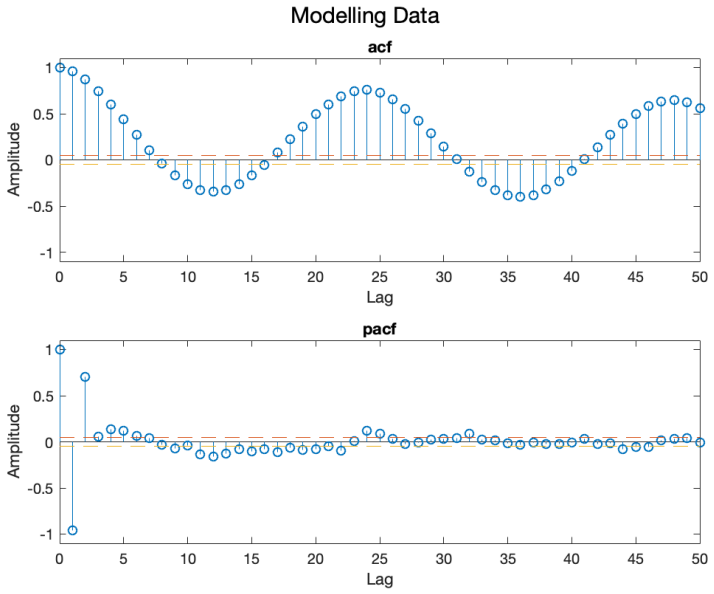

Figure 2: ACF and PACF for the modelling data.

By studying the ACF for the modelling data in figure 2 there seems to be a very strong seasonality of 24 hours. This is entirely expected, due to the day and night cycle. The data is therefore modelled as a SARMA-process with s = 24. The ACF and PACF of the differentiated data can be seen in figure 3.

In our experience it has been successful to include an MA(s) part when modelling differentiated data, so an MA(24) part is added to the SARMA model. The rest of the c_k terms are first assumed to be zero, which gives the C-polynomial $C(z) = 1 + c_{24} \cdot z^{-24}$.

To find a suitable model order ARMA(p,24)-models, where p ranged from 1 to 20, using PEM, were computed. Their respective BIC, AIC and FPE were then calculated and can be seen in figure 4.

Here, the BIC measure was clearly not a good criterion, as it too heavily punished large models. According to BIC, an AR(2,24) would be the best model, however this gave a non-white residual so this information criterion was

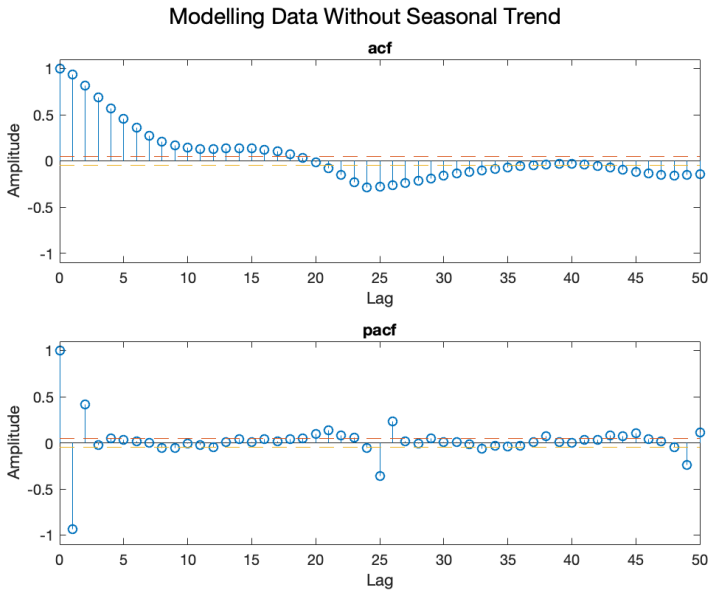

Figure 3: ACF and PACF for the modelling data without the seasonal trend.

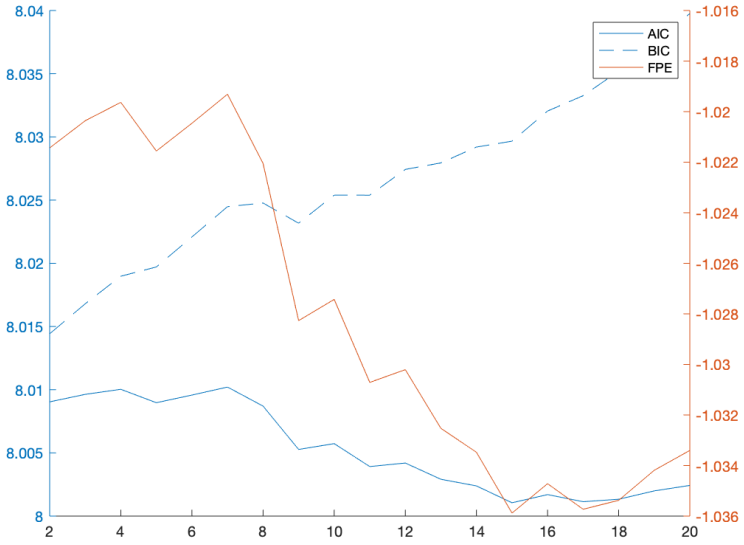

Figure 4: AIC, BIC and FPE for different AR orders and a MA(24) parameter.

ignored. The FPE and AIC measures, however, both suggests an ARMA(15,24) model, which we then proceeded with. Some of it's parameters were however insignificant and were thus removed. The final model arrived at is seen in equations 1 and 2

$$A(z)\nabla_{24}y_t = C(z)e_t \tag{1}$$

$$\begin{aligned} A(z) =& 1 - 1.35z^{-1} + 0.49z^{-2} - 0.09z^{-3} + 0.08z^{-7} \\ & - 0.05z^{-9} - 0.07z^{-13} + 0.07z^{-14} - 0.03z^{-15} \\ C(z) =& 1 - 0.8089z^{-24} \end{aligned} \tag{2}$$

2.3 Validation and testing

When choosing a model we continuously used it to predict on our validation data. The performance of the final model in this task is presented in figure 5 and table 1. The plausibly best naive predictors were chosen, given the information at hand at the time of prediction.

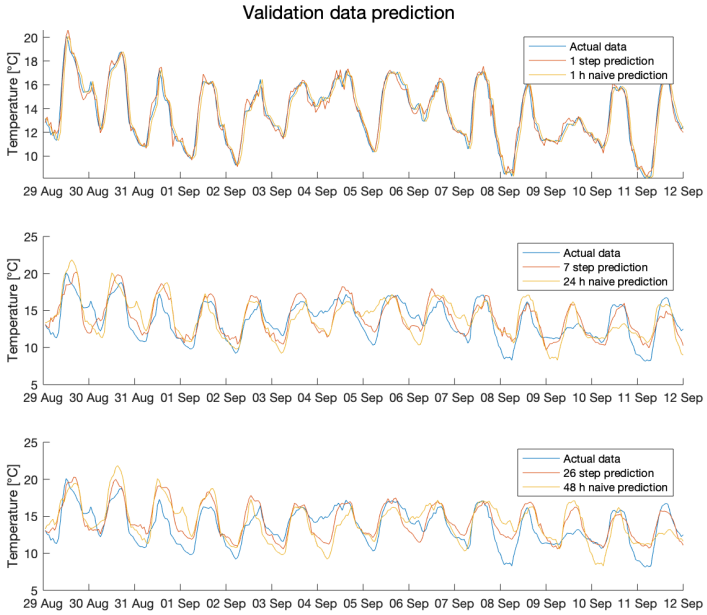

Figure 5: Naive and model predictions for the verification data

The k = 1 predictor had a residual that passed the Ljung-Box-Pierce, McLeod-Li and Monti test. The k = 7 step prediction error did not have a white residual.

Model prediction	1 step	7 steps	26 steps
Naive prediction	1 h	24 h	48 h
Prediction residual variance	0.222	2.375	2.533
Naive residual variance	0.531	3.836	4.814

Table 1: Prediction residual variances for model and naive predictions

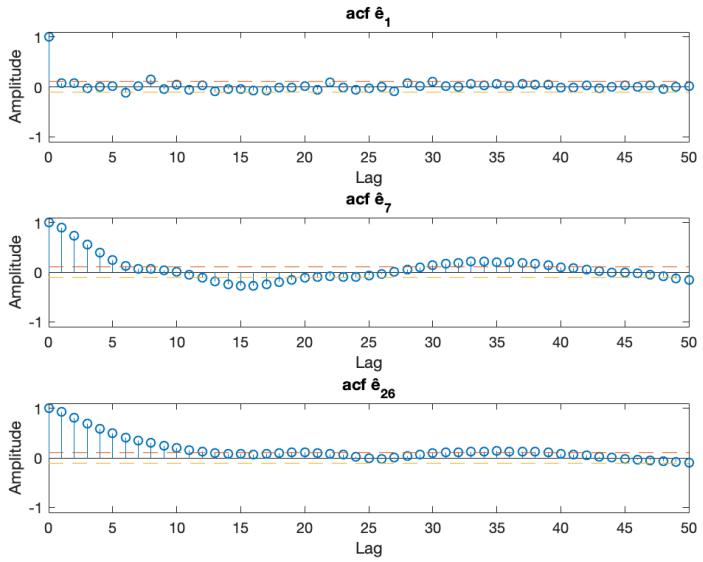

Figure 6: ACF for the k = 1,7,26 step prediction errors

The prediction error could however be modelled as a MA(6) process that in turn had a white residual, which implies that model is sufficiently accurate. Studying the ACF plots for the k = 1,7,26 step predictors all of them resembles a MA(k-1) process fairly well.

After deciding on our final model it was time to test it on our two test sets. These results are presented in figure 7 and table 2.

	Sep. 1967	Mar. 1968
24 h naive predictor	2.549	2.355
SARMA 7 h	1.447	1.736

Table 2: Prediction residual variances on the test sets

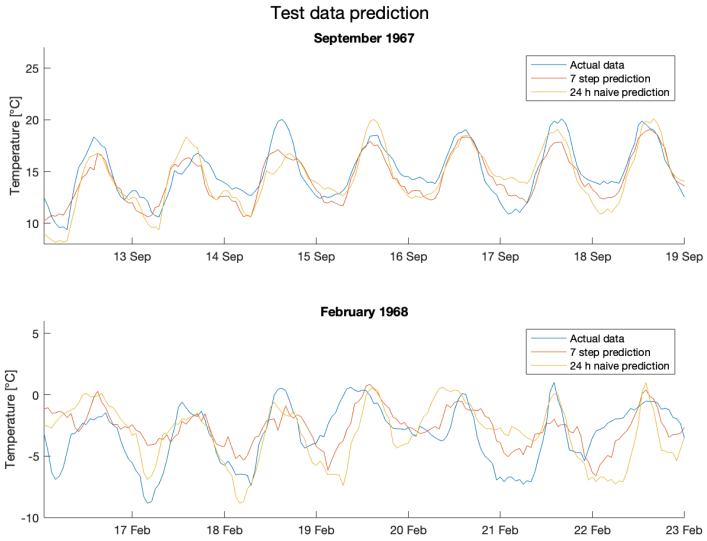

Figure 7: 7 step predictions on the test sets

3 Stationary ARMAX-modelling

3.1 Data transformation

At this stage we wanted to introduce the net radiation, $[W/m^2]$, as an input signal to our model. When doing so, we first wanted to see if it should be transformed. In the top right of figure 8 the Box-Cox normality plot for the untransformed data is presented. With a maximum value at 0.18 it suggested that either the logarithm or square root would be suitable transformations.

After some experimentation we concluded that the algorithm described in equation 3 worked best, with a maximum log-likelihood at $\lambda = 1.03$ after transformation. The transformed data and its Box-Cox normality plot can also be seen in figure 8.

$$\begin{aligned} \boldsymbol{x} &:= ln(\boldsymbol{x} - min\{\boldsymbol{x}\} + 11) \\ \boldsymbol{x} &:= \boldsymbol{x} - mean\{\boldsymbol{x}\} \end{aligned} \tag{3}$$

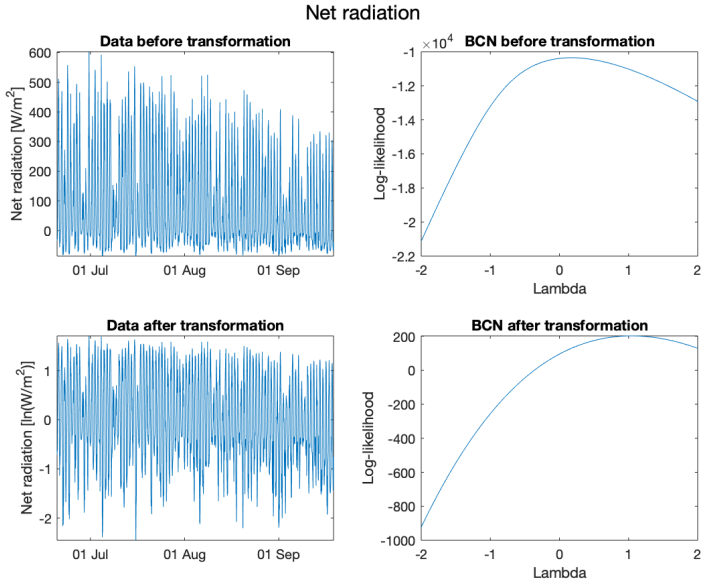

Figure 8: Data and Box-Cox normality plots, before and after transformation

3.2 Signal modelling

After transforming the signal it was time to model it. The ACF of the transformed net radiation suggested a strong seasonality of 24, as expected. The signal was therefore differentiated with a 24 hour time difference, as seen in equation 4.

$$\boldsymbol{x} := \boldsymbol{x} - 0.83\,\boldsymbol{x}\,z^{-24} \tag{4}$$

The differentiation coefficient used was not -1, but instead we fitted a differentiation model to the data, yielding a coefficient of -0.83.

Thereafter, we fitted $\boldsymbol{x}$ with an ARMA(12, 24) model. The 24:th MA-coefficient was included to compensate for the differentiation, and it was the only MA-coefficient that was non-zero and allowed to vary. One by one, the least significant of the insignificant AR-parameters were removed and the model was re-estimated. To do this we built the `removeUnsignificant.m` function seen in the appendix. The final model arrived at is described in equations 5.

$$\begin{aligned} A_3(z) =& 1 - 0.9715z^{-1} + 0.1192z^{-2} + 0.06592z^{-3} \\ & - 0.04131z^{-4} + 0.06197z^{-5} + 0.04867z^{-7} \\ & + 0.0426z^{-9} + 0.03053z^{-11} + 0.07145z^{-12} \\ C_3(z) =& 1 - 0.694z^{-24} \end{aligned} \tag{5}$$

3.3 Model Order Selection

In order to determine the order of the polynomials in the Box-Jenkins model we need to estimate the transfer function from w_t to ϵ_t, defined in equation 6.

$$\begin{aligned} w_t &= \frac{C3}{A3\nabla_{24}}\boldsymbol{x} \\ \epsilon_t &= \frac{C3}{A3\nabla_{24}}\boldsymbol{y} \end{aligned} \tag{6}$$

Their cross-correlation is seen in figure 9. These estimated impulse weights of the transfer function suggests a Box-Jenkins model of orders (d,r,s) = (0,1,0).

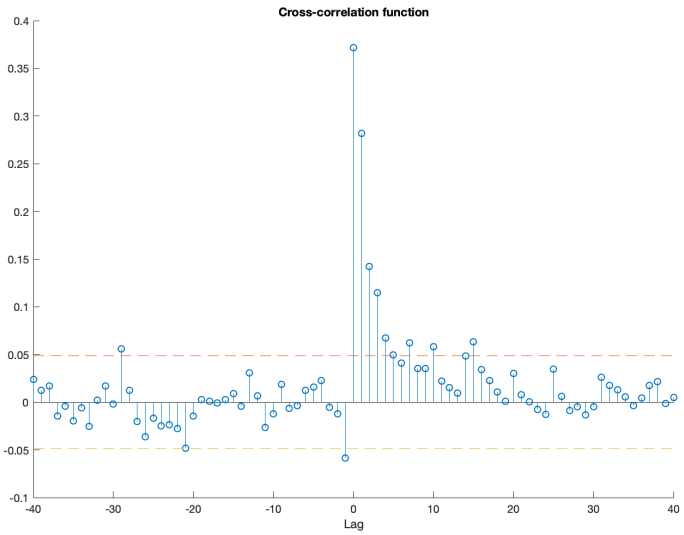

Figure 9: Cross Correlation of ϵ_t and w_t

We, however, found (d,r,s) = (0,3,0) to work better as it left our w_t less correlated with the resulting $\tilde{e}_t = y_t - Hx_t$ and $v_t = \epsilon_t - Hw_t$. These values determine the order of its B and A2 polynomials, as seen in equations 7.

$$\begin{aligned} ord\{B(z)\} &= 0 \\ ord\{A_1(z)\} &= 3 \\ Delay &= 0 \end{aligned} \tag{7}$$

Using the model orders stated above, the A2 and B polynomials were fitted using MATLAB's pem and iddata functions. Using A2 and B, $\hat{e} = Hx$ and the cross correlation of the driving noise processes were calculated. As seen in figure 11, the driving noise processes are rather uncorrelated, which is desired. The ACF and PACF for the $\hat{e}$ can be seen in figure 10.

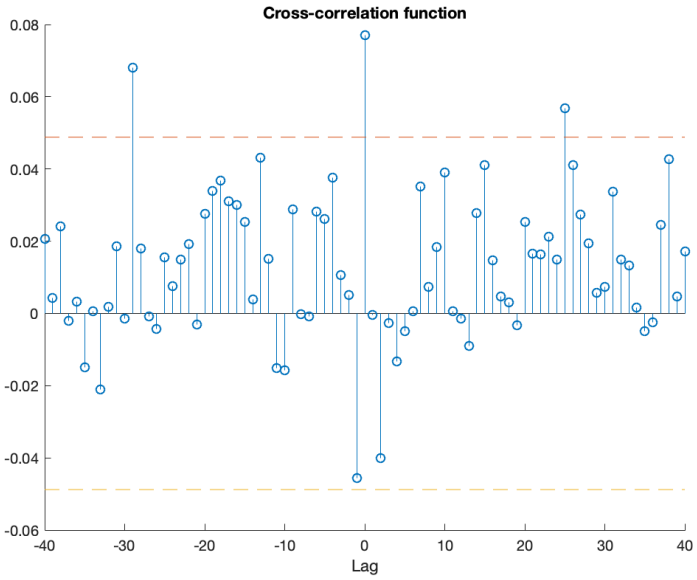

Figure 10: Cross correlation of the driving white noise processes

$\hat{e}$ was modelled using an ARMA(6,2), which in turn were the orders of the $A_1(z)$ and $C_1(z)$ polynomials in the Box-jenkins model. The A(z) and C(z) polynomial used to model $\hat{e}$ are presented in equation 8

$$\begin{aligned} A_1(z) =& 1 - 2.284z^{-1} + 3.069z^{-2} - 1.598z^{-3} \\ &+ 0.141z^{-4} - 0.044z^{-6} \\ C_1(z) =& 1 - 1.508z^{-1} + 0.5794z^{-2} \end{aligned} \tag{8}$$

Using the model orders attained for $A_2(z)$, $A_2(z)$, $C_1(z)$ and $B(z)$ the Box-Jenkins model was created. The model is presented in equation 9.

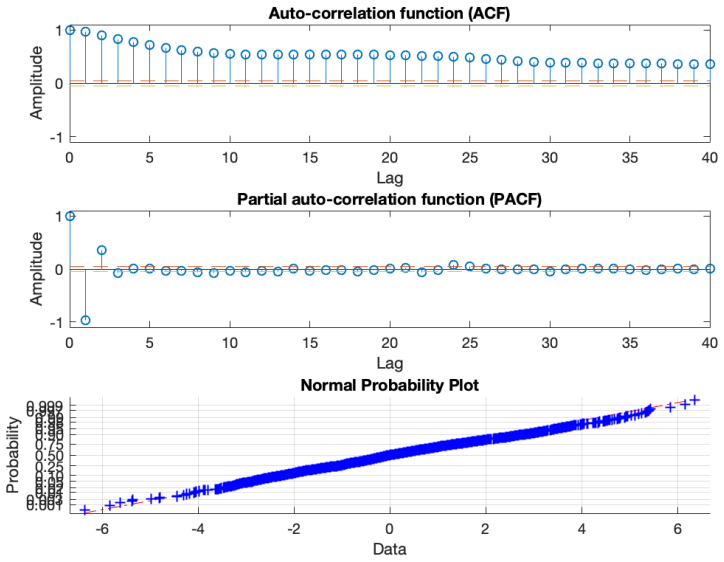

Figure 11: ACF, PACF and Normal probability plot of ê

$$
\begin{aligned}
B(z) &= 1.191 + 0.346z^{-1} - 1.083z^{-2} - 0.243z^{-3} \\
C(z) &= 1 - 1.539z^{-1} + 0.607z^{-2} \\
D(z) &= 1 - 2.875z^{-1} + 3.14z^{-2} - 1.636z^{-3} + 0.417z^{-4} - 0.037z^{-6} \\
F(z) &= 1 - 0.0.518z^{-1} - 0.986z^{-2} + 0.520z^{-3}
\end{aligned} \tag{9}
$$

3.4 Validation and Testing

The performance of the final model on this task is presented in table **??**. The best possible naive predictors were the same as in the previous tasks.

The k = 1 predictor is deemed to have a white residual. The k = 7 step predictor did not have a white residual however however it could as in the case of the previous task be modelled as an MA(6) process. After deciding on our final Box-Jenkins model the prediction was made on the two test sets which can be seen in figure 14.

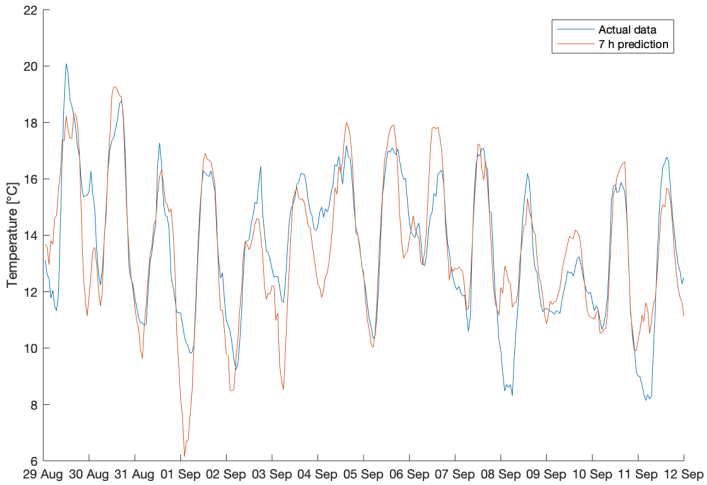

Figure 12: Predicted temperatures on the validation set with the ARMAX model

	Sep. 1967	Mar. 1968
24 h naive predictor	2.549	2.355
SARMA 7 h	1.447	1.736
ARMAX 7 h	1.147	1.679

Table 3: Prediction residual variances on the test sets

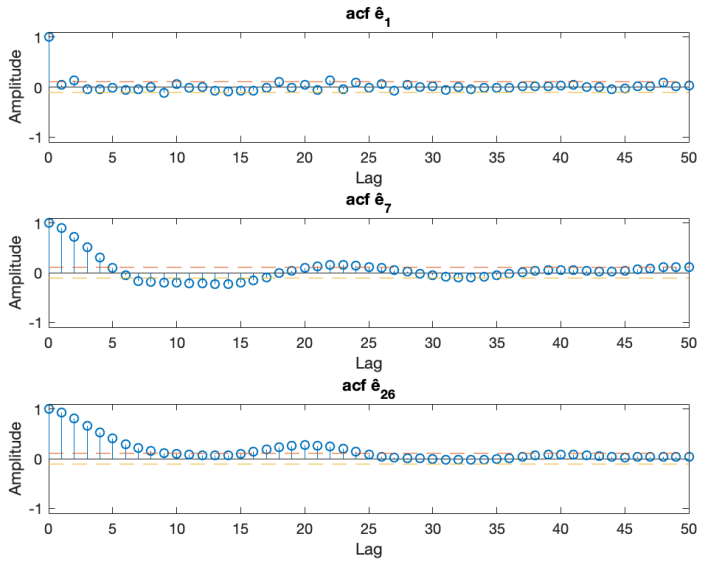

Figure 13: ACF for k = 1, 7, 26 step prediction errors on the validation data

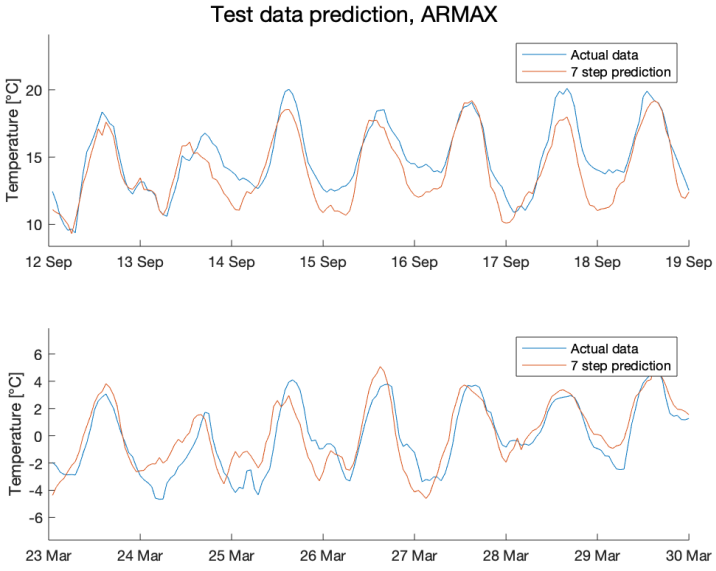

Figure 14: Predicted temperatures for the two test sets with the ARMAX model

4 Recursive Parameter Estimation

4.1 Kalman Filtering

As the parameters may vary over time the temperature was modelled as a dynamic SISO-system. This was done using a Kalman filter, see appendix for the implementation of the Kalman filter. The Box-Jenkins model presented in eq 9. can be written on the form in eq 10

$$A(z)y_t = B(z)x_t + C(z)e_t \tag{10}$$

The process in eq 10 is written on the state space form according to eq 11

$$\begin{aligned} \mathbf{x_{t+1}} &= \mathbf{A x_t} + \mathbf{R_e} \\ \mathbf{y_t} &= \mathbf{C_{t|t-1} x_{t|t-1}} + w_t \end{aligned} \tag{11}$$

where

$$\mathbf{C_{t|t-1}} = [-y_{t-1} ... - y_{t-10} u_t ... ut - 4e_{t-1} ... e_{t-6}] \tag{12}$$

and $\mathbf{A} = \mathbf{I}$, w_t is white noise and $\mathbf{R_e}$ is an additive noise. $\mathbf{R_e}$ is a diagonal matrix with where $\mathbf{R_e}(i,i) = 0$ when $i \neq i$. See eq 13

The parameters are thought to vary unequally depending on the time period. When it's cloudy in winter it's also usually warmer, while during the summer the opposite is true. The variance for the B-parameters are therefore assumed to be more differ more between winter and summer than the A and B parameters. Moreover the C-parameters are assumed to change less than the A-parameters.

The R_e matrix is presented in equation 13, where the values of the σs are presented in equations 14.

$$\mathbf{R_e} = \begin{pmatrix} \sigma_A^2 & & & & & & & & & \\ & \cdots & & & & & & & & \\ & & \sigma_A^2 & & & & & & & \\ & & & \sigma_B^2 & & & & & & \\ & & & & \cdots & & & & & \\ & & & & & \sigma_B^2 & & & & \\ & & & & & & \sigma_C^2 & & & \\ & & & & & & & \cdots & & \\ & & & & & & & & \sigma_c^2 \end{pmatrix} \tag{13}$$

$$\begin{aligned} \sigma_A^2 &= 0.3 * 10^{-1} \\ \sigma_B^2 &= 4 * 10^{-1} \\ \sigma_c^2 &= 0.7 * 10^{-1} \end{aligned} \tag{14}$$

Setting $w_t = 10^3$, the initial variance, $\mathbf{R_{x00}} = \mathbf{I} * 10^{-4}$ and the start values for the parameter estimation, $\mathbf{x0}$ = [A(2:end) B C(2:end)]. The reason for choosing a lower $\mathbf{R_{x00}}$ than $\mathbf{R_e}$ is because we are rather confident about our initial estimates. When then used our Kalman filter to predict on our validation set. The parameters achieved are presented in figure 15.

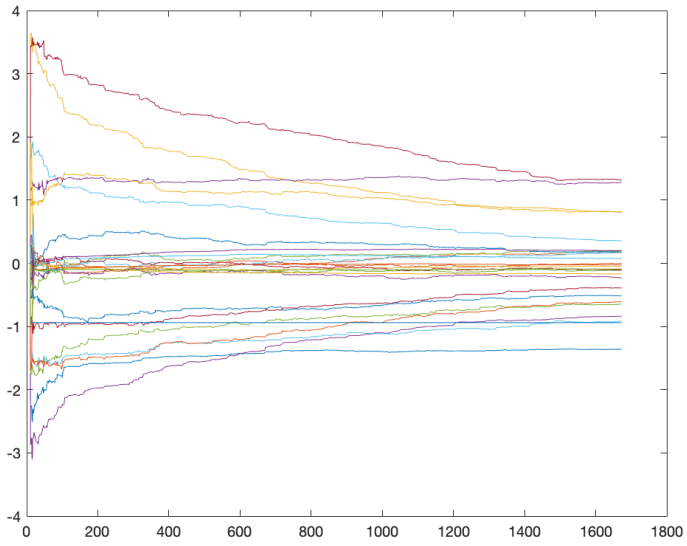

Figure 15: Estimation of parameter values in the Kalman filter

After this step the Kalman filter was iterated once again, this time with x_0 being the last state of the former iteration. The resulting parameter estimates are presented in the top of figure 16

As is clear from the plot, many of the parameter estimation behaves relatively stationary. Many of the parameter estimates are, however, very close to zero. These parameters might not be necessary to describe the structure of the time series data. If $-0.2 \leq \bar{x}_i \leq 0.2$ we deemed it to be unnecessary and removed it from the model. This resulted in the parameter estimates shown at the bottom of figure 16.

4.2 Validation and Testing

The k = 1 predictor is deemed to have a white residual. The k = 7 step predictor did not have a white residual, however it could, as previously, be modelled as an MA(6) process. The ACF of the k = 1, 7 and 26 step predictors can be studied in 20 and 21. After deciding on our final model the prediction and parameter estimation was made on the two test sets which can be seen in figures 18 and

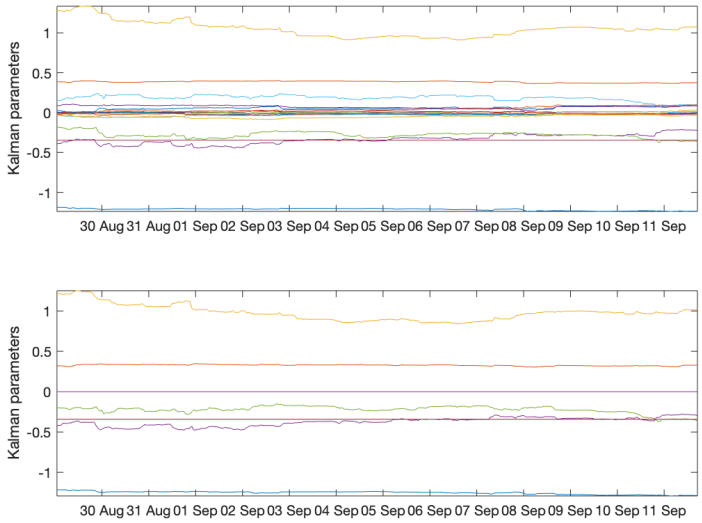

Figure 16: Second and third iteration of Kalman parameters

19. The final model is the ARMAX model presented in equation 15 and its performance in table 4, along with the models in previous sections.

	Sep. 1967	Mar. 1968
24 h naive predictor	2.549	2.355
SARMA 7 h	1.447	1.736
ARMAX 7 h	1.147	1.679
Kalman 7 h	1.141	1.382

Table 4: Prediction residual variances on the test sets

$$
\begin{aligned}
A &= 1 - 1.353Z^{-1} + 0.394Z^{-2} \\
B &= 0.948 + 0.575Z^{-1} - 0.723Z^{-2} - 0.350Z^{-3} \\
&\quad - 0.948Z^{-25} + 0.373Z^{-26} + 0.350Z^{-27} \\
C &= 1 - 0.344z^{z} - 1
\end{aligned}
\tag{15}
$$

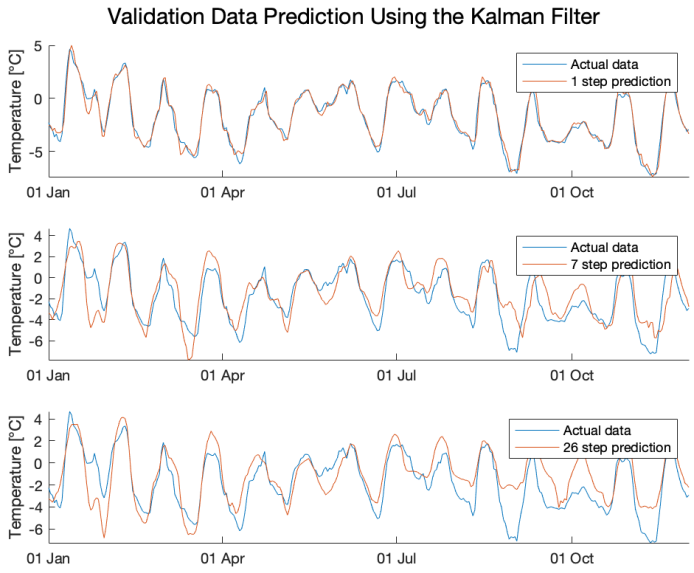

Figure 17: k = 1, 7, 26 predictions on the validation set

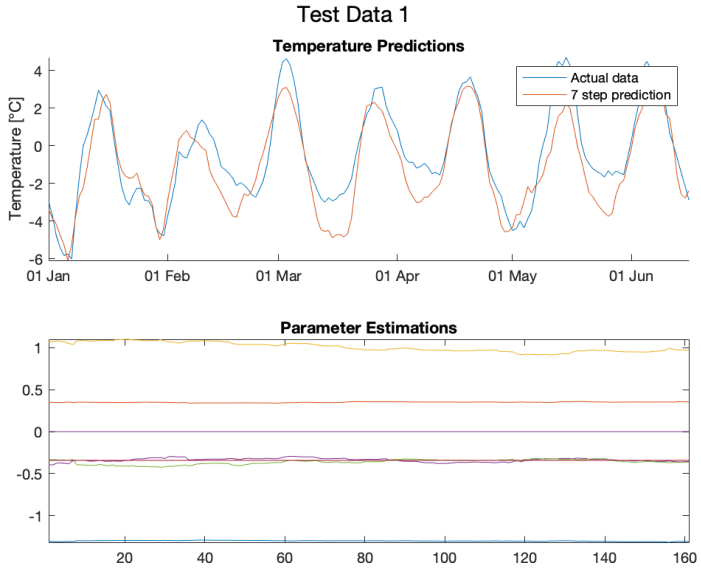

Figure 18: Predictions and parameter estimates for the first test set

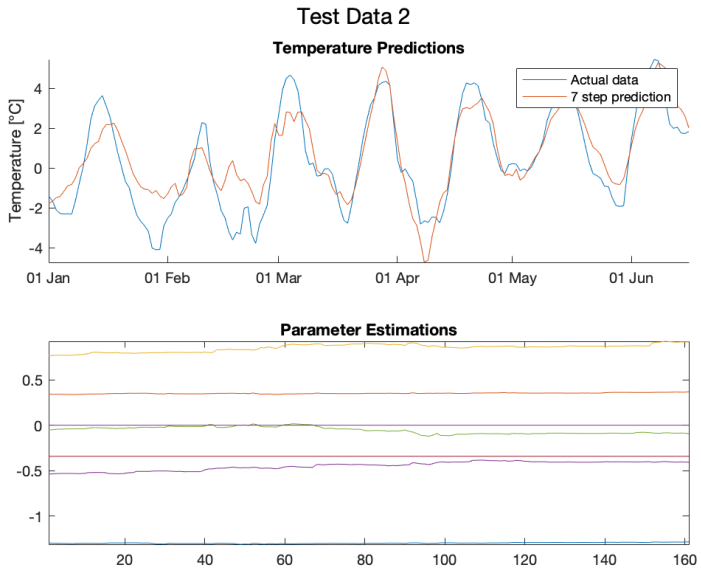

Figure 19: Predictions and parameter estimates for the second test set

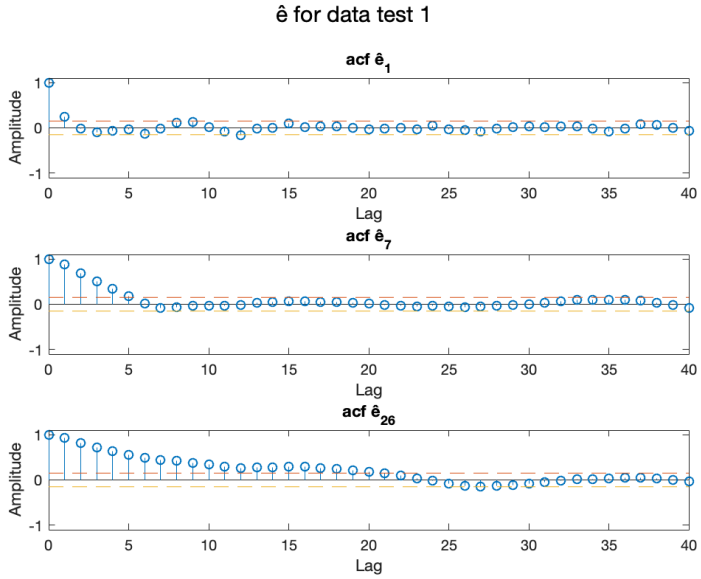

Figure 20: ACF for the k = 1,7,26 step prediction errors on the Test Data 1

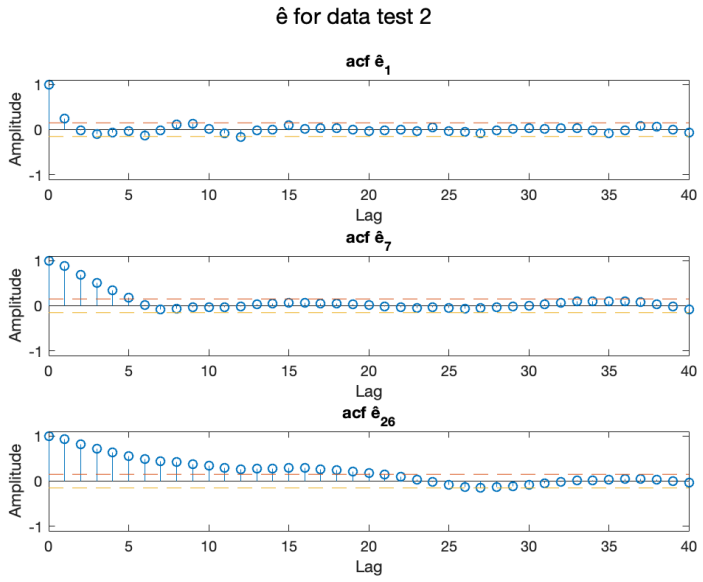

Figure 21: ACF for the k = 1,7,26 step prediction errors on the Test Data 2

5 Discussion

The ARMAX and the Kalman models behave rather similarly for 7 and 26 step predictions. As both models treat x as a know signal, this indicates that the models mainly use it for predicting. If the signal was not treated as known, but rather had to be predicted, the accuracy of the k = 7 and k = 26 step predictors would be considerably lower. To be able to model the temperature, the model orders would likely need to be quite a bit larger.

For the Test Data 1, with the Kalman filter, the estimation of parameters were very constant. This leads us to believe that we have successfully found parameters that accurately describe the process in this time period. In the Test set 2 the parameters does not experience the same stationary behavior as they are repeatedly updated. This is expected as the parameters are modelled from the modelling set which is a lot closer in time to test data 1 set than the test data 2 set, so the parameters are modelled from a process that is more similar to test data 1 than test data 2. As the parameters are allowed to change with time, with the Kalman filter, it therefore is no surprise that it performs better than the ARMAX model during the second data test period. On the first test period the Kalman filter performed extremely similar to the ARMAX model which is reasonable given the fact that the recursive estimation of the parameters over the time period are very consistent.

Appendices

removeInsignificant.m

```
function model = removeInsignificant(model, data, threshold, ...
    testWhite, plot)

if nargin < 3
    threshold = 1;
end
if nargin < 4
    testWhite = 0;
end
if nargin < 5
    plot = 0;
end
res = filter(model.A,model.C,data);
models = {model};
FPE = model.Report.Fit.FPE;
white = montiTest(res);

if white == 0
    return
end

Alen = length(model.A);

run = 1;
if length(model.A) > 1
    Afree = model.Structure.A.Free;
end
if length(model.C) > 1
    Cfree = model.Structure.C.Free;
end

i = 1;
while run
    [pvec, pvec_sd] = getpvec(model);
    signvec = abs(pvec./pvec_sd);
    [minvalue, minindex] = min(signvec);

    if minvalue < threshold
        A = model.A;
        C = model.C;

        if minindex > Alen
            C(minindex - Alen + 2)     = 0;
            Cfree(minindex - Alen + 2) = 0;
        else
            A(minindex+1)     = 0;
            Afree(minindex+1) = 0;
        end

        modelInit = idpoly(A, [], C);
        if length(model.A) > 1
```

```
51              modelInit.Structure.A.Free = Afree;
52          end
53          if length(model.C) > 1
54              modelInit.Structure.C.Free = Cfree;
55          end
56          model = pem(data, modelInit);
57
58          i = i+1;
59          res = filter(model.A,model.C,data);
60          models{i} = model;
61          FPE(i) = model.Report.Fit.FPE;
62          white(i) = int8(montiTest(res));
63      else
64          run = 0;
65      end
66  end
67
68  run = 1;
69
70  while run
71      [minimum, i] = min(FPE);
72      if white == 0
73          models{i} = [];
74      else
75          run = 0;
76      end
77  end
78
79  model = models{i};
80
81  if testWhite
82      res = filter(model.A,model.C,data);
83      whitenessTest(res);
84  end
85  if plot
86      analyzets(res);
87  end
88  end
```

superKalman2.m

```
% y    : The output vector that we're predicting
% u    : Input vector
% A    : Kalman A matrix, here we use the identity matrix
% Re   : The state noise variance matrix
% Rw   : Guess for measurment variance
% Rxx0 : Initial guess for state variance matrix
% x0   : Initial guess for state vector
% p    : Order of Box Jeniks A-polynomial
% r    : Order of Box Jeniks B-polynomial
% q    : Order of Box Jeniks C-polynomial
% k    : Number of time steps we predict ahead

function [y_hat, xsave] = superKalman2(y, u, A, Re, Rw, Rxx0, ...
    x0, p, r, q, k)

if nargin < 11
    k = 1;
end

xtt_1 = x0;   % Initial state guess
Rxx_1 = Rxx0; % Initial state variance

len   = length(A);
N     = length(y);

y_hat  = zeros(N,1);          % Prediction vector
e      = zeros(N,1);          % Stores prediction errors
y_temp = zeros(N+k,1);        % Temporarily stores predictions
y_temp(1:len) = y(1:len);     % Sets the first elements = y
xsave  = zeros(N-k,length(x0)); % Matrix for storing our states
xsave(max([p,r,q])-1,:) = x0';

for t=max([p,r,q])+1:N-k
    % C is a function of time, containing the following
    %    The previous p outputs
    %    The previous r inputs
    %    The previous q prediction errors
    C = [-y(t-(1:p)); u(t-(0:r)); e(t+1-(1:q))]';

    y_temp(t) = y(t);           % Now we know the current output
    e(t)      = y(t) - C*xtt_1; % Our best guess for the ...
        prediction error

    % Update
    Ryy = C*Rxx_1*C' + Rw;
    Kt  = Rxx_1*C'/Ryy;
    xtt = xtt_1 + Kt*e(t);
    Rxx = Rxx_1 - Kt*C*Rxx_1;

    % Predict
    Rxx_1 = A*Rxx*A' + Re;
    xtt_1 = A*xtt;

    % For loop since we're predicting k steps forward
```

```
53      for tp = t + (1:k) % Time of prediction
54          % Ck consists of:
55          %    The p previous best predictions of y
56          %    The current and r-1 previous inputs
57          %    The q previous prediction errors, or zeros if unknown
58          Ck = [-y_temp(tp-(1:p)); u(tp-(0:r)); e(tp-(1:q))]';
59          y_temp(tp) = Ck*xtt; % Current prediction k steps forward
60      end
61      y_hat(t+k) = y_temp(t+k); % Our prediction k steps ahead
62
63      xsave(t,:) = xtt; % Save states
64  end
65
66  end
```