Project - Time Series Analysis

Niels Gudde, Albert Regnell

December 2023

Contents

1 Problem description **3**

2 Reconstruction of rain data **3**
2.1 Introduction . . . 3
2.2 Evaluation of reconstructed data . . . 4
2.3 Method . . . 4
2.4 Check reconstruction with simulated rain . . . 6

3 Modeling of vegetation **6**
3.1 The Naive model . . . 6
3.2 Model without using rain as input . . . 7
3.2.1 Checking code with simulated data . . . 8
3.2.2 Performance on validation data . . . 8
3.3 Model with rain as input . . . 10
3.3.1 Modeling the rain . . . 10
3.3.2 Finding the transfer function . . . 11
3.3.3 Modeling $\tilde{e_t}$ . . . 11
3.3.4 Estimating all the parameters . . . 12
3.3.5 Box-Jenkins performance on validation data . . . 13
3.4 Time-varying model for El-Geneina . . . 16
3.4.1 Recursive model for the input data . . . 16
3.4.2 Recursive model of vegetation . . . 18
3.4.3 Modifying the state vector . . . 19
3.4.4 Model performance . . . 19
3.5 Performance on test data . . . 22

4 Modeling for Kassala **22**
4.1 SARIMA: Unchanged coefficients . . . 23
4.2 SARIMA: Re-estimating the coefficients . . . 24
4.3 The Kalman Box-Jenkins model . . . 25
4.3.1 Predicting the rain . . . 25
4.3.2 Predicting the vegetation . . . 25
4.4 Performance on Kassala . . . 28

5 Discussion **28**
5.1 El-Geneina . . . 28
5.2 Kassala . . . 29

1 Problem description

This project aims to model, predict, and reconstruct measurements for a vegetation index, NDVI, using precipitation in Sudan. The vegetation index data set contains measurements made by satellites of the reflectance of different wavelengths from the surface of the earth. From this, the Normalized Difference Vegetation Index (NDVI) can be calculated as

$$NDVI = \frac{Ch_2 - Ch_1}{Ch_2 + Ch_1} \tag{1}$$

where Ch1 and Ch2 denote the reflectance in the red spectral band (580–680 nm) and in the near infrared (725–1000 nm), respectively. The reason why this index is a good measurement of vegetation is because the chlorophyll in plants will absorb light in the red spectral band more than other wavelengths. This characteristic is special for chlorophyll compared to other objects in nature. A high value of NDVI thus suggests that there is a lot of chlorophyll in the area, meaning that the level of vegetation is high. The vegetation data set contains three measurements per month during the period from January 1982 to December 1999, giving a total of 1440 measurements, from two different measurement stations in Sudan; Kassala and El-Geneina.

The data for the precipitation was collected once a month over 40 years between 1960 and 1999, giving a total of 648 measurements. The data sets are plotted below in figure 1. Both the datasets look reasonably stationary over the entire period and thus the models are estimated on the entire data set rather than only a subset.

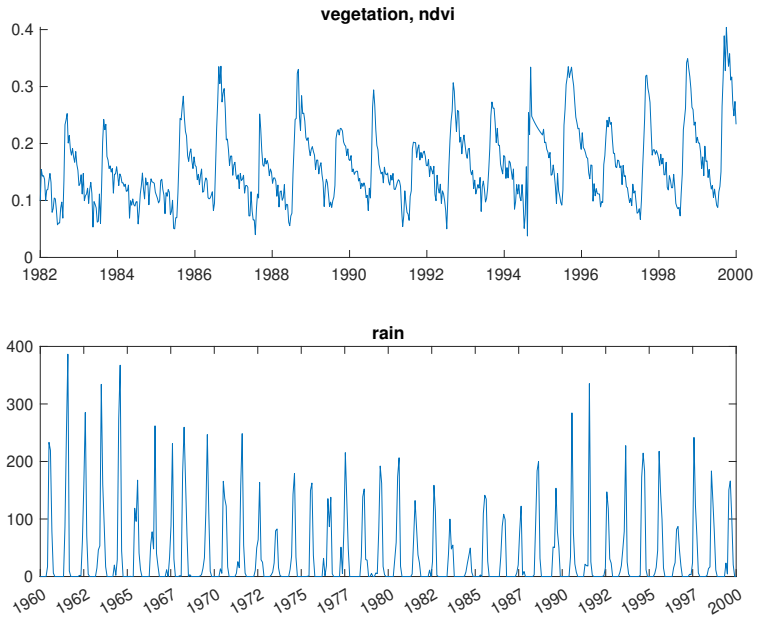


Figure 1: The rescaled vegetation data and the original rain data before preprocessing

2 Reconstruction of rain data

2.1 Introduction

The first problem that needs to be taken care of is the discrepancy between the frequency of datacollection in the two datasets. The NDVI is measured three times per month, while the rain data is only gathered once a month. This can be seen as a case of missing samples, i.e. we view the rain data as a dataset where 2 out of every three data points are missing. This then needs to be handled in an appropriate way.

2.2 Evaluation of reconstructed data

One aspect of the reconstruction that requires consideration is how to evaluate the quality of the reconstruction. This can be done in two ways, both of which using the difference between the data set given, y_t and the relevant sums of the reconstructed data. We denote this value by $\hat{e}$:

$$\hat{e} = y_t - (x_t + x_{t-1} + x_{t-2}), \quad t = 1, 4, 7, 10, ... \tag{2}$$

We obtain the first measure simply by calculating the sum of all these residuals.

$$Q_1 = \sum_{t=1}^{n} \hat{e} \tag{3}$$

Since the total amount of rain should be the same in both data sets, this sum should ideally be as close to zero as possible. The second quality measure looks at variance of the residuals:

$$Q_2 = Var(\hat{e}) \tag{4}$$

This measure also captures whether the reconstructed data has peaks and valleys in the same places as the true rain, something that would not be seen through only looking at Q_1. These residuals would preferable be white. However, we are modeling the rain with a very simple model, so this is unlikely.

2.3 Method

The reconstruction will be completed using a Kalman filter to handle the missing data points. The reconstructed rain is assumed to be well modeled as an AR(1) process, and the state space representation used in the Kalman filter is thus

$$\theta_t = A\theta_{t-1} + R_e \tag{5}$$
$$y_t = C\theta_t + R_w \tag{6}$$

where

$$A = \begin{bmatrix} a_1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}, \quad C = \begin{bmatrix} 1 & 1 & 1 \end{bmatrix}, \quad \theta_t = \begin{bmatrix} x_t \\ x_{t-1} \\ x_{t-2} \end{bmatrix} \tag{7}$$

The subzero values had to be removed, which was could be done with two different methods. Either by only putting the subzero values to zero, or by also subtracting the removed value from the next data point. The Q_2-value was notably better for the first alternative and was therefore chosen.

In the Kalman filter, there are several variables to tweak in order to get the best reconstruction possible. The first ones are choosing $R_e, R_{x,x}$ and R_w. A big value was chosen for $R_{x,x}$ since the initial states are set to zero and are thus not accurate. $R_e(1,1)$ is chosen as a relatively large value as x_t should be allowed to change. $R_e(2,2), R_e(3,3)$ are set much lower since they should not change. R_w is set to 0.1. Next is the a_1 parameter in the A matrix, corresponding to the parameter in the AR(1) representation of x_t. In order to find the best value of a_1, the Kalman filter was run for 200 different possible values ranging between -1 and 1. Afterwards, the Q_1 and Q_2, introduced above, was plotted:

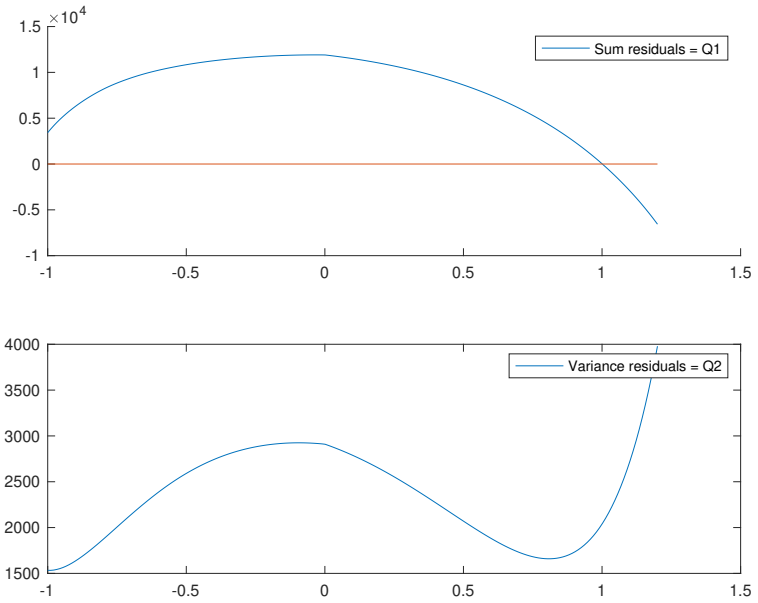


Figure 2: Comparing Q_1 and Q_2 for different a-values

The plot shows that $Q_1 = 0$ is achieved at $a = 1$ and Q_2 gives two candidates for the best value, $a = -1$ and $a = 0.87$. Firstly, $a = -1$ was tested and the resulting reconstructed rain was plotted. In the plot, every other x_t was zero, which makes sense in an AR(1) process with parameter -1. This was however deemed unreasonable and $a = -1$ was disregarded. To decide between the other two values, a trade-off between the measures was needed. Since $a = 0.87$ performs poorly in Q_1, but $a = 1$ performs quite well in both measures, $a = 1$ was chosen. Modeling the rain with $a = 1$, summing up every three x and comparing it to the original rain gives the following plot:

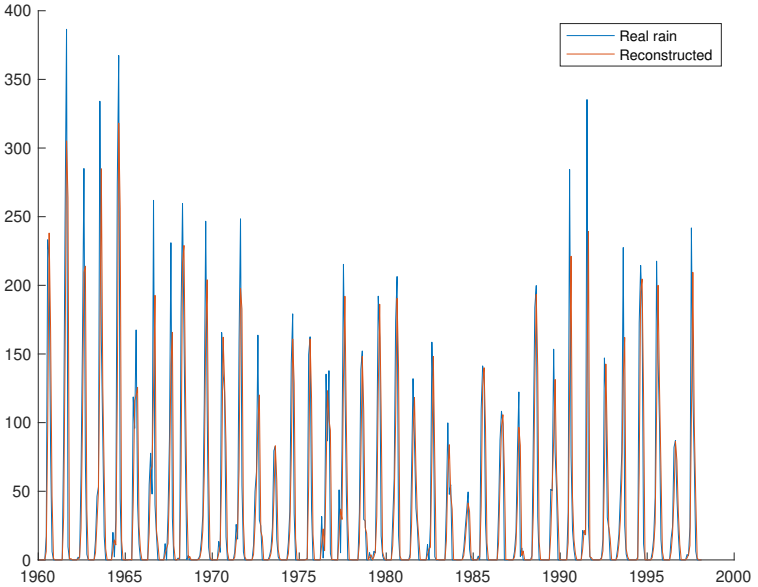
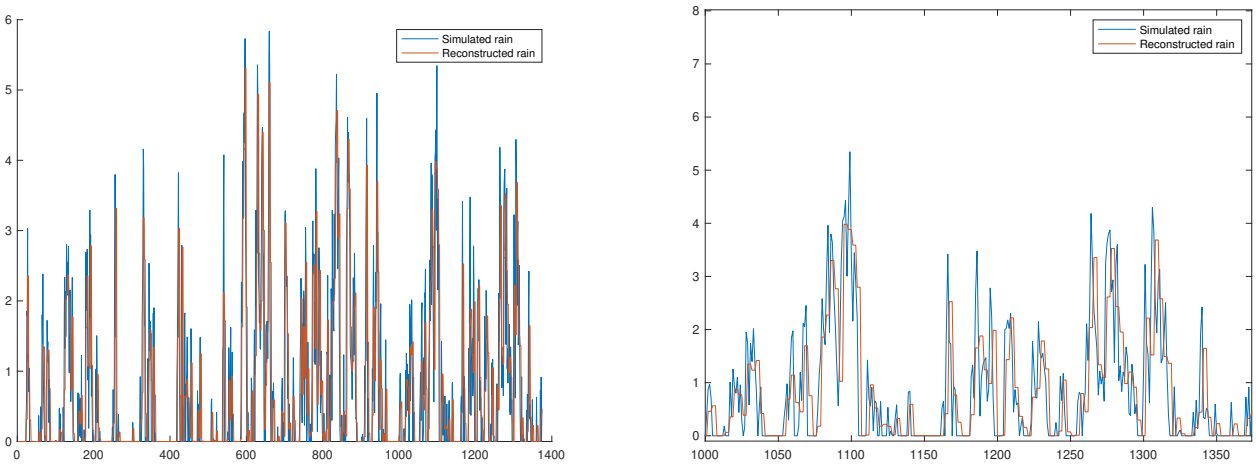


Figure 3: $x_t + x_{t-1} + x_{t-2}$ compared to original rain

The Kalman filter seems to model the rain quite well, with the exception that it does not reach the full magnitude of the peaks.

2.4 Check reconstruction with simulated rain

In order to further check if the implemented Kalman filter successfully can reconstruct rain data, a AR(1) process is simulated with 10 000 data points. All the sub-zero values were put to zero. As earlier, every three rain periods were summed up to create new vector that corresponds to the original rain data. The constructed Kalman filter was then used to reconstruct the AR(1) process. The results were:



(a) Simulated rain

(b) Zoomed simulated rain

Figure 4: Simulated rain x_t and the reconstruction of it

The plots shows that if the rain actually behaves as an AR(1) process, the Kalman filter will to be able to reconstruct it in a good way. Hence, the reconstructed rain data have good chances of giving accurate enough input to the model.

3 Modeling of vegetation

The vegetation will be modeled with two different types of models, namely with and without using rain as input. In the case with rain as input, a time-varying model where the parameters are allowed to change over time will also be tested. First of all, the vegetation data is split into modeling-, validation- and test-data sets, with 70, 20 and 10 percent of the data respectively. For the model with input, all the rain data points up to the start of the vegetation data is included in the modeling data. The remaining data is split in accordance to the vegetation data. In the original data set, the NDVI was stored as integers between 0 and 255, and since the index in reality is between -1 and 1, the data was rescaled to the true interval before any analysis was conducted.

3.1 The Naive model

During the process of constructing models it is useful to have a "naive" model that makes predictions in a simple way. The naive model can then be used as a reference point, making sure that the more complex models perform better.

From plotting the vegetation, a clear period of about $t = 36$ is visible. This would suggest that the vegetation grows in a seasonable fashion, which seems reasonable since regions near the equator experience yearly rain seasons. Therefore the naive model is constructed such that it predicts the vegetation by assuming that it will be the same as last year. The naive model give the following predictions on the validation data:

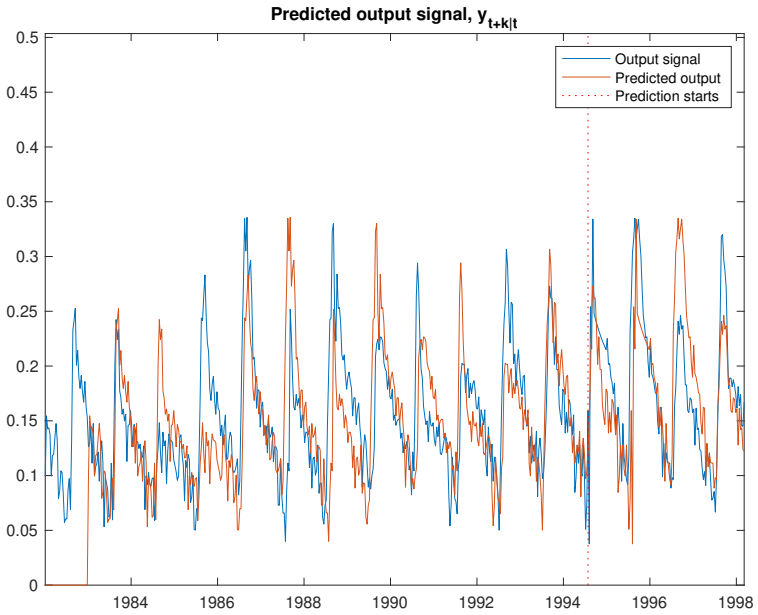


Figure 5: Naive prediction vs real data

The plots shows that the naive model actually is a relatively accurate predictor, indicating that the last years vegetation is a good guess of this years vegetation. The resulting residuals on the validation data are:

Variance of prediction residuals (10^{-4})	**Variance of normalized residuals**
31.838	0.7119

Table 1: Naive-model: Variance of residuals

The normalized variance shows that the model is explaining just a small part of the total variance of the data.

3.2 Model without using rain as input

The model construction is started off with checking if the vegetation data is Gaussian. The normplot looks relatively Gaussian, but the D'Agostino-Pearson's K2 test indicates the data is not normal distributed. The Box-Cox is plotted which has a peak at 0.22 on the x-axis, which implies that a log transformation is appropriate. Since the normalized vegetation data can have negative data, a constant has to be added to the data before log-transforming. However, this transformation does not make the data notably more Gaussian, and therefore a log-transformation is deemed unnecessary. However, the non-Gaussian data impacts the certainty of confidence intervals, since the confidence intervals used assumes Gaussian data.

The ACF of the data shows a strong seasonality with a period of 36. This seems reasonable since a period of 36 corresponds to one year. Therefore the data is differentiated with ∇_{36}. Thereafter the ACF exhibits a strong ringing and the PACF has a peak at lag $t = 1$. An AR(1) term is therefore added which leaves residuals that are white according to the Monti-test. The ACF and PACF do still exhibit a peak in $t = 36$ which indicates that an MA(36) could be useful to compensate for the ∇_{36} term. This decreases the Monti-value and all the coefficients are still significant on the 5% significant level. The model is thus a SARIMA(36, 1, 0, 36) and the coefficient values with confidence interval is the following:

$$\nabla_{36} y_t (1 - 0.7543(+/- 0.03441) z^{-1}) = (1 - 0.7448(+/- 0.03523) z^{-36}) e(t) \tag{8}$$

Where $y = NDVI$. The Monti-test gives a value of $18.19 < 36.42$, hence the residuals are deemed white.

The ACF, PACF and normplot of the residuals:

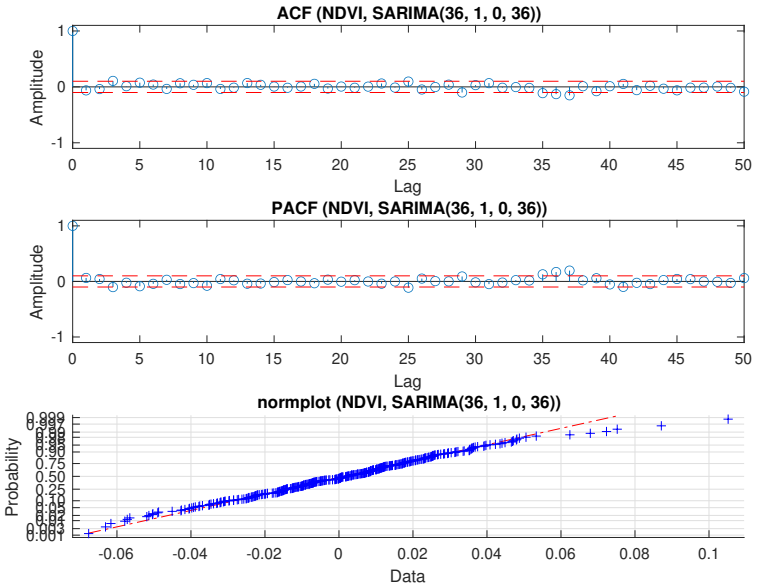


Figure 6: ACF, PACF and Normplot of residuals from Model without input.

The ACF and PACF do not exhibit any spikes over the confidence intervals, which suggest that there is no more structure to be exploited. The normplot indicates that the residuals are normal distributed. In conclusion, the figure agrees with the residuals being white.

3.2.1 Checking code with simulated data

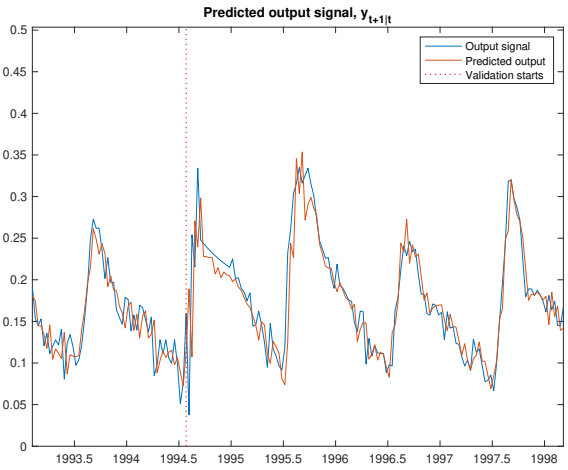
To see that the code is implemented correctly, 10 000 data points is simulated using the polynomials in the final model. Running the data points through the code the gives the resulting model:

$$\nabla_{36} y_t (1 - 0.7596(\pm 0.006533) z^{-1}) = (1 - 0.7591(\pm 0.006556) z^{-36}) e(t) \tag{9}$$

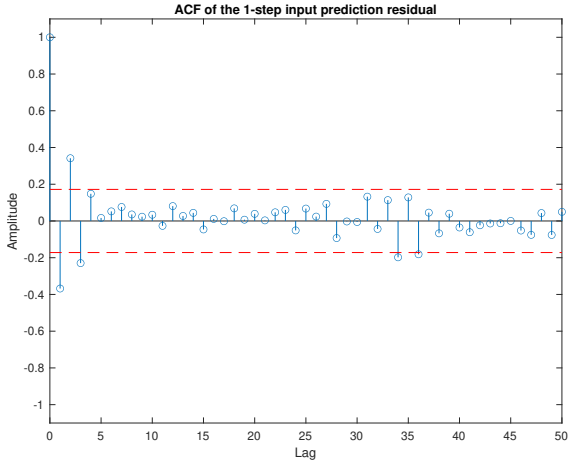
This model leaves almost perfectly white residuals and it is very similar to the final model, which implies that the code successfully can identify SARIMA(36, 1, 0, 36) models. It reinforces that a suitable model has been fitted to the vegetation data.

3.2.2 Performance on validation data

To examine the models generalization capabilities, its' predictions are tested on the validation data. This gave the following results:

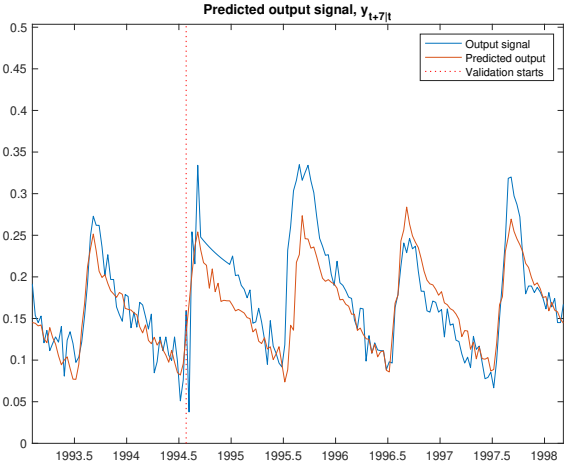


(a) 1-step prediction vs real vegetation

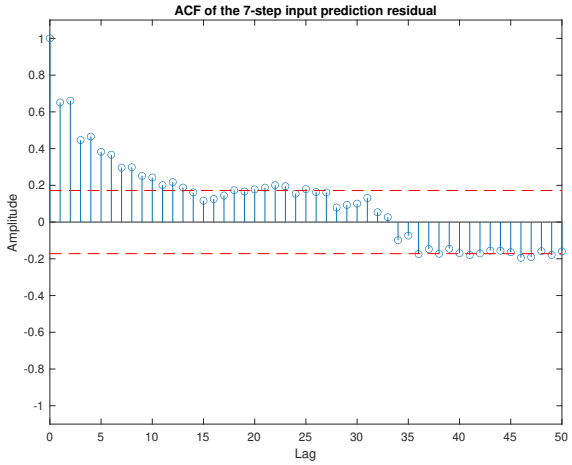


(b) 1-step prediction ACF

Figure 7: 1-step prediction and ACF of its' residuals



(a) 7-step prediction vs real vegetation



(b) 7-step prediction ACF

Figure 8: 7-step prediction and ACF of its' residuals

The 1-step prediction plot looks accurate. The 7-step prediction is a little but off and struggles especially with capturing the peaks. Ideally a k-step prediction should have residuals that behave as an MA(k-1), which none of them do. The 1-step prediction looks like an MA(3) and the 7-step as an MA(10), which is relatively close to the right order. Residuals as MA(k-1) is ideal and is not something that should be expected. Normplots of the PACF shows that it behaves rather normally. This suggest that the model is not perfect, but fairly accurately constructed.

Variance of prediction residuals	**Variance** (10^{-4}	**Normalized variance**
Naive	31.838	0.7119
SARIMA 1-step	9.4787	0.2120
SARIMA 7-step	15.746	0.3521

Table 2: SARIMA-model: Variance of residuals

The normalized variance shows that the SARIMA model is able to explain a lot more of the datas' variance than the naive model.

3.3 Model with rain as input

A better model of the vegetation might be possible to construct if the rain is taken as an input to the model. Using rain as an input makes sense since intuitively, rain should have an impact on the vegetation. Figure 1 also supports this hypothesis. To build this model, the Box-Jenkins' model is chosen, since it is more general than the ARMAX. The formula for the Box-Jenkins' model is.

$$y_t = \frac{B(z)z^{-d}}{A_2(z)}x_t + \frac{C_1(z)}{A_1(z)}e_t \tag{10}$$

Note that if $A_1 = A_2$, Box-Jenkins' is identical to the standard ARMAX-model.

3.3.1 Modeling the rain

The rain reconstructed with the Kalman filter will be used as input. Before starting any analysis, the Box-Cox plot is used to see if the data should be transformed to make it more Gaussian. The plot has a maximum in -0.5, but the log-transform is used since it generally works better. Since there are data points where the rain is 0, the data have have to be shifted before it is log-transformed. The log-transformed rain is still not normal according to D'Agostino-Pearson's K2 test, but the normplot looks a lot better. Therefore $x_t = log(rain + 1)$ will be used as input. However, since the data is still not Gaussian, the confidence intervals can not be fully trusted.

The first step in creating the model is to determine the polynomials $B(z)$ and $A_2(z)$. The transfer function from x_t to y_t is denoted as $H(z) = \frac{B(z)z^{-d}}{A_2(z)}$. Since x_t is not white, the impulse cannot be directly determined from the cross correlation from x_t to y_t. Instead the model is rewritten as:

$$\frac{A_3(z)}{C_3(z)}y_t = \frac{B(z)}{A_2(z)}w_t + \frac{A_3(z)C_1(z)}{C_3(z)A_1(z)} \tag{11}$$

where w_t and e_t is white noise. The equation is simplified as:

$$\epsilon_t = H(z)w_t + v_t \tag{12}$$

Note that the pre-whitened ϵ_t is now the output of the transfer function model, having the preferred uncorrelated signal as its input, allowing $H(z)$ to be estimated using the CCF from the white noise w_t to ϵ_t. To find w_t, x_t needs to be modeled as an ARMA process.

The ACF of x exhibits a strong seasonality with period 36. Therefore x is differentiated with period 36. However, a better model, with regard to whiteness of residuals, was found without differentiating. Instead a $a_{36}y_{t-36}$ and a $c_{36}e_{-36}$ was added. Thereafter there were still some ringing and and spikes in the ACF and PACF. After testing different solutions, a final model was found with white residuals (Monti test: $27.58 < 36.42$) and significant coefficients. The final model is an ARMA(36,36) with 6 estimated parameters. The model, along with the coefficients' and their 95% confidence interval is stated below:

$$x_t = \frac{C_3(z)}{A_3(z)}w_t \tag{13}$$

where,

$$A_3(z) = 1 - 0.4351(\pm 0.02437)z^{-3} + 0.1414(\pm 0.01813)z^{-9} - 0.518(\pm 0.02222)z^{-36} \tag{14}$$

$$C_3(z) = 1 + 1.01(\pm 0.006648)z^{-1} + 1.006(\pm 0.006669)z^{-2} + 0.01309(\pm 0.006442)z^{-36} \tag{15}$$

The residuals ACF, PACF and normplot are:

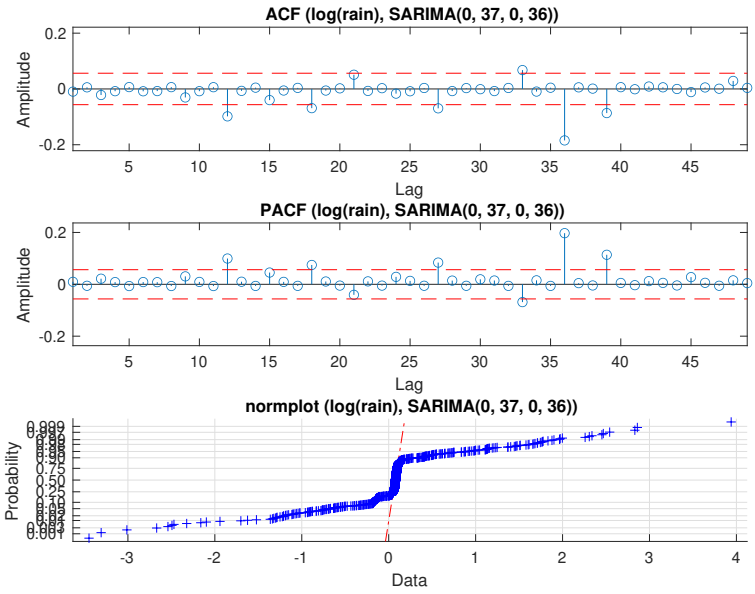


Figure 9: ACF of 1-step prediction residuals

The ACF and PACF still have peaks in $t = 36$, but since there already is an AR(36)- and MA(36)-coefficient in the model, and the residuals are deemed white, this model is considered good enough.

3.3.2 Finding the transfer function

The transfer function is found by analysing the cross correlation from w_t to ϵ_t. Unfortunately the CCF shows no significant correlation, which suggest that the correlation can't be trusted. Instead, because of lack of information, the most simple transfer function is tested. This means that $d = 0, s = 0, r = 0$, which suggest that the rain would only have a scaling effect on the vegetation, and with no lag. This seems possible since the vegetation probably needs less than 10 days to absorb and utilize the water. Fitting $y_t = H(z)x_t + \tilde{e_t}$ gives:

$$H(z) = 0.05741(\pm 0.0108) \tag{16}$$

The residuals would ideally show no correlation with x_t, but instead they exhibit a strong sinusoidal behaviour. This will however be ignored since the cross correlation has been deemed too unreliable. The simplistic transfer function is chosen primarily, and may have to be revised if the final model does not perform satisfactory.

3.3.3 Modeling $\tilde{e_t}$

Now y_t has been modeled as a function of the input x_t, but is still to be modeled as the ARMA-process in the BJ model, i.e., modeled as the ouput of white noise through the polynomials $C_1(z)$ and $A_1(z)$. Therefore, defining the ARMA-part as:

$$\tilde{e_t} = \frac{C_1(z)}{A_1(z)} e_t \tag{17}$$

By using estimated polynomials $B(z)$ and $A_2(z)$, $\tilde{e_t}$ can be estimated as

$$\tilde{e_t} = y_t - \frac{\hat{B}(z)z^{-d}}{\hat{A}_2(z)} x_t \tag{18}$$

By filtering out the input-dependent part of the process y_t, we may then estimate suitable orders for the polynomials $C_1(z)$ and $A_1(z)$ using the standard ARMA-modeling procedure. The ACF of $\tilde{e_t}$ shows a

seasonality with period 36. Therefore the data is differentiated with period 36. However, a model with white noise was not achieved with this initial step. This could be because because of the period of vegetation not being exactly one year. To capture this, AR and MA terms with lag 35,36 and 37 was added the model. This left some ringing in the ACF and a peak in $t = 1$ in the PACF. Therefore an AR(1) term was added. This resulted in white residuals, but insignificant MA(35), MA(37) and AR(35) terms, which were removed. The remaining coefficients became significant and the residuals were deemed white (Monti test: $27.89 < 36.42$). The model coefficients and their confidence intervals are:

$$A(z) = 1 - 0.7573(\pm 0.0348)z^{-1} - 0.8902(\pm 0.0486)z^{-36} + 0.6416(\pm 0.05741)z^{-37} \tag{19}$$

$$C(z) = 1 - 0.6086(\pm 0.07567)z^{-36} \tag{20}$$

The ACF, PACf and normplot of the residuals are:

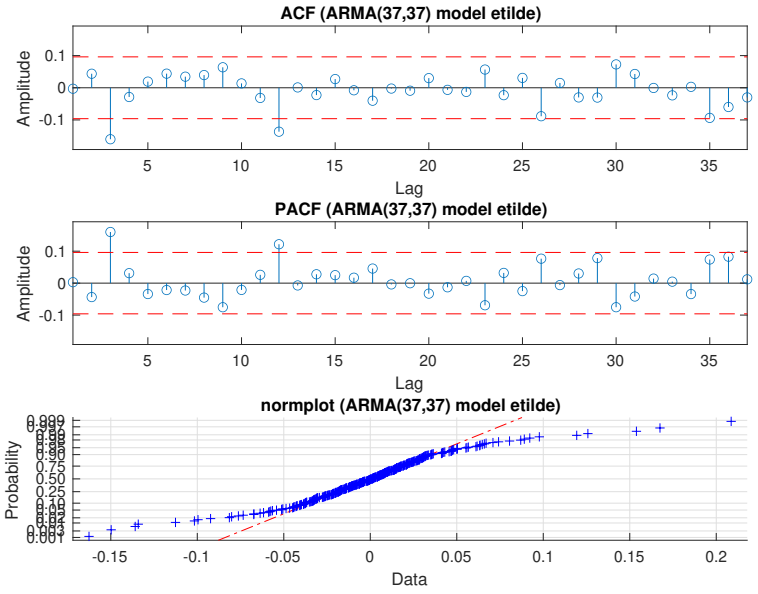


Figure 10: PACF, ACF and normplot of residuals from modeling $\tilde{(e_t)}$

Plotting the remaining cross correlation from the residuals to x_t shows that the cross correlation has been substantially removed, except for large lags. However, not much weight will be put into this since the cross correlation has been shown to be unreliable. The remaining step is to estimate all the polynomials together.

3.3.4 Estimating all the parameters

Estimating all the polynomials together gave white residuals. However, this model did not perform very well on the validation data. To give it a better chance of capturing the seasonality of $t = 36$, the MA(35), MA(37) and AR(35) were added to the model again. MA(37) and AR(37) became significant, but not MA(35), which was removed. This gave white residuals (Monti test: $21.98 < 36.42$) and this was chosen as the final model. The final estimated Box-Jenkins' model is:

$$y_t = \frac{B(z)}{A_2(z)}x_t + \frac{C_1(z)}{A_1(z)}e_t \tag{21}$$

Notice that the z^{-d} is included in the $B(z)$ polynomial. The polynomials are estimated to:

$$A_1(z) = 1 - 0.8519(\pm 0.0293)z^{-1} - 0.1468(\pm 0.0376)z^{-35} - 0.7037(\pm 0.09251)z^{-36} + 0.6942(\pm 0.07708)z^{-37} \tag{22}$$

$$A_2(z) = 1 \tag{23}$$

$$B(z) = 0.002495(\pm 0.001942) \tag{24}$$

$$C1(z) = 1 - 0.7406(\pm 0.08002)z^{-36} + 0.09535(\pm 0.03634)z^{-37} \tag{25}$$

The ACF, PACF and normplot of the residuals are:

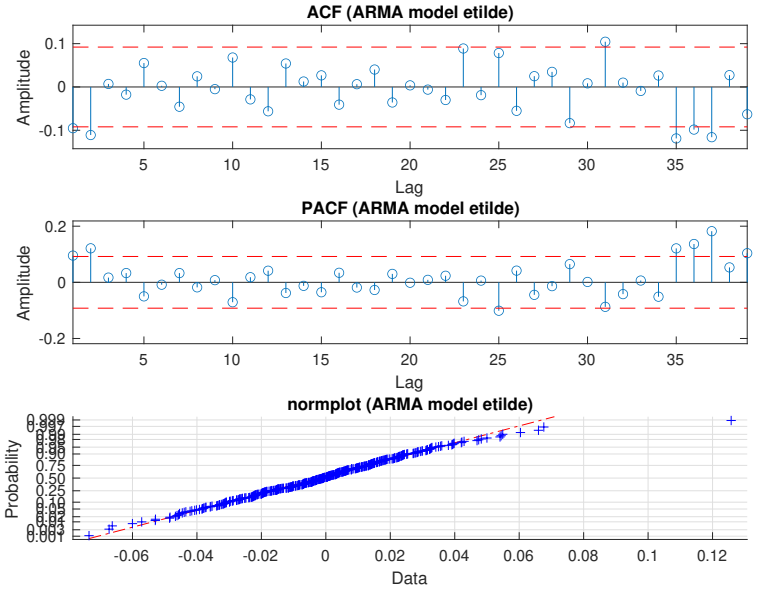


Figure 11: PACF, ACF and normplot of residuals from Box-Jenkins model

3.3.5 Box-Jenkins performance on validation data

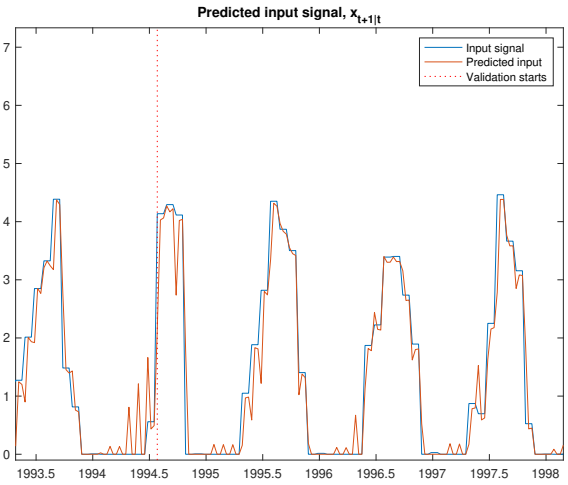
To examine how well the model generalize and to compare it to other models, it's prediction capabilities will be tested on the validation data.

Predicting the rain

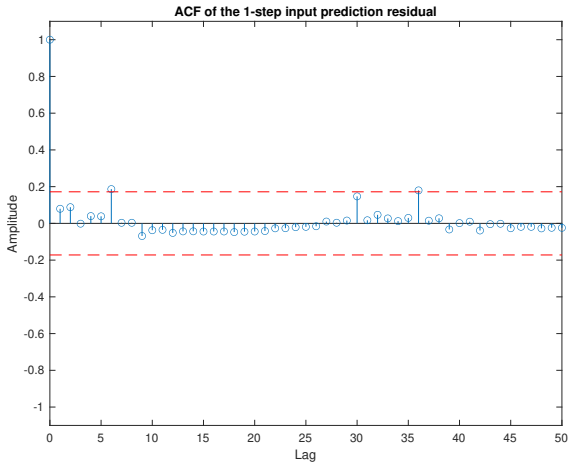
First of all the input, the $log(rain + 1)$, has to be predicted. This is done by using

$$x_t = \frac{C_3(z)}{A_3(z)} w_t \tag{26}$$

The values of $C_3(z)$ and $A_3(z)$ are stated in equation (14) and (15). Note that this can return negative values of x_t, but since $x_t = log(rain + 1)$, and the lowest value of rain is 0, they should always be positive. Therefore, to make these predictions agree with the x_t, all the values below zero will be put equal to zero. This gives the following result:

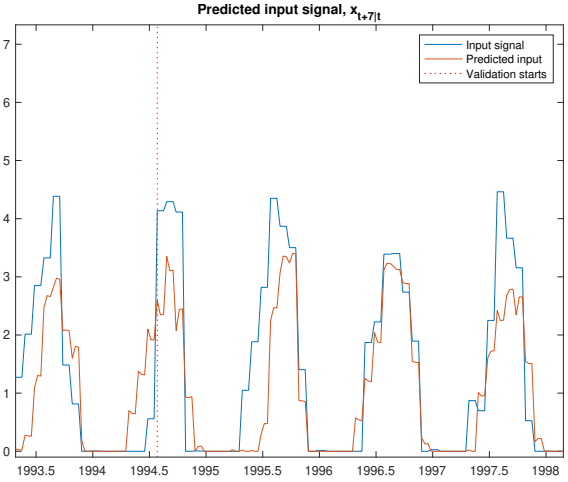


(a) 1-step prediction

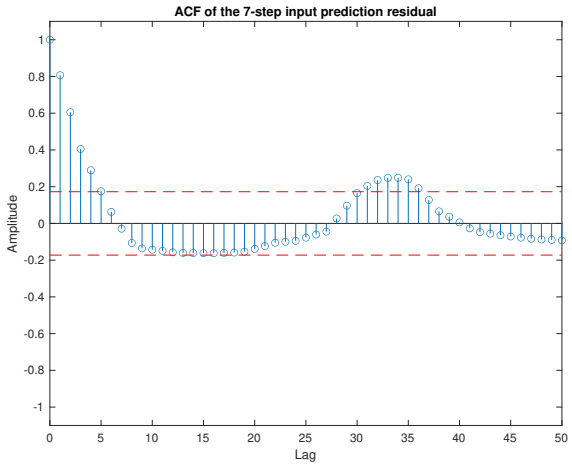


(b) ACF of residuals

Figure 12: 1-step prediction of log(rain+1) and ACF of the residuals



(a) 7-step prediction



(b) ACF of residuals

Figure 13: 7-step prediction of log(rain+1) and ACF of the residuals

Note that the reconstructed rain is treated as the true rain. This is as mentioned previously problematic due to the simplistic AR(1) modeling in the Kalman filter. However, it was earlier concluded that the Kalman filter managed to make a good enough fit. Therefore, the prediction will be deemed as good enough if the predicted rain is similar to the reconstructed rain.

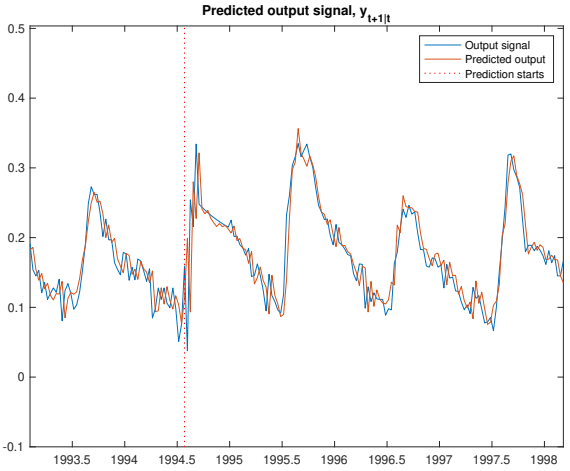
The 1-step prediction looks accurate and the ACF looks white, as it should. The 7-step prediction is systematically too small. However, the prediction seems to correctly capture the time of the peaks and the seasonality. As the rain is used as input, it is not necessarily crucial that it predicts the correct level of rain, but rather that the prediction times the peaks well. The ACF looks like an MA(5) which is close to the expected MA(6) process. Hence the prediction looks good and should be able to produce useful input to the prediction of vegetation. As an performance measure, the Q_1 and Q_2 has been calculated and can be used to compare the prediction with later models:

Rain prediction	Sum of residuals (Q_1)	Variance (Q_2)	Normalized variance
1-step prediction	-23.775	2387.2	0.7126
7-step prediction	-892.92	3430.3	0.8277

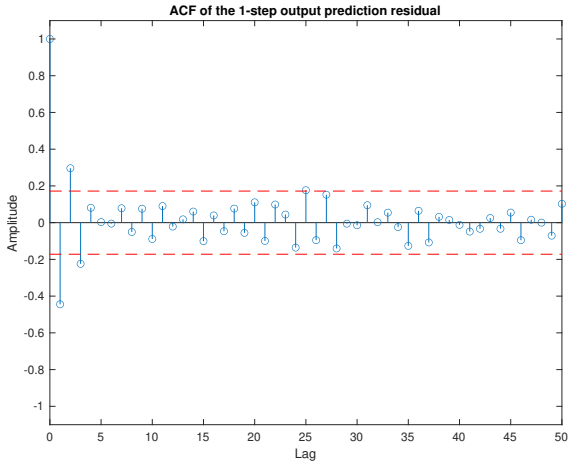
Table 3: Prediction of rain: Variance of residuals

The variance of the residuals agrees with the plots - the 1-step prediction is significantly better than the 7-step prediction. However, none of them explain a large part of the data variance. This could make it difficult for the model to utilize the rain.

Predict the vegetation The 1- and 7-step prediction of the vegetation gave the following results:

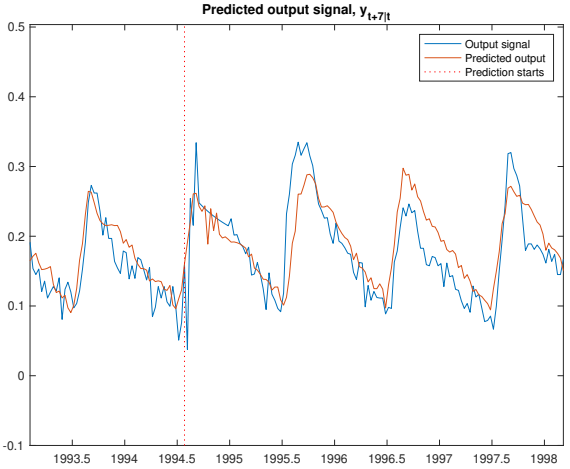


(a) 1-step prediction

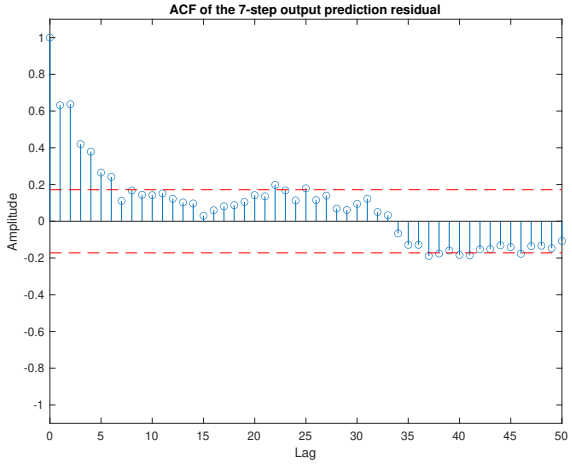


(b) ACF of residuals

Figure 14: 1-step prediction of vegetation using BJ-model



(a) 7-step prediction



(b) ACF of residuals

Figure 15: 7-step prediction of vegetation using BJ-model

The 1-step prediction looks good and seems to accurately fit the validation data. The ACF of the residuals looks like an MA(3), which is rather close to an MA(0). The 7-step prediction seems to follow the validation data rather well as well. The ACF of the residuals follows an MA(6) which is ideal for a k-step prediction. Below is table that compared the variance of the residuals with the SARIMA- and the naive model:

Variance of prediction residuals	**Variance (10^{-4})**	**Normalized**
Naive	31.838	0.7119
SARIMA 1-Step	9.4787	0.2120
BJ 1-Step	10.034	0.2244
SARIMA 7-step	15.746	0.3521
BJ 7-Step	15.886	0.3552

Table 4: Variance of prediction error residuals on validation data

The table shows that the SARIMA- and the BJ-model has similar performance on the validation set. This is to be expected, since the models are similar to each other. The most noteable difference is that the BJ-model uses the input. However, $B(z) = 0.002495(\pm 0.001942)$, which means that the $log(rain + 1)$ is scaled by 0.002495 and then added to the vegetation. This will not have a big impact on the estimated vegetation, and since $A_2 = 1$, the BJ-model is basically an ARMA-model. This suggest that using the rain as imput does not improve the predictions. At least when the rain is measured so sparsely and was reconstructed as an AR(1).

3.4 Time-varying model for El-Geneina

The next step is to investigate whether the model can be improved by allowing the parameters to change over time. This will be done through using a Kalman filter to estimate the parameters in the polynomials K_A, K_B and K_C recursively, where:

$$K_A = A_1 * A_2 \tag{27}$$
$$K_B = A_1 * B \tag{28}$$
$$K_C = A_2 * C_1 \tag{29}$$

and A_1, A_2, B and C_1 are defined as in section 3.3.4.

3.4.1 Recursive model for the input data

The first step is to model the input data, i.e. the rain, recursively. This will be done by implementing a Kalman filter where the state vector is made up by the non-zero coefficients in A_3 and C_3 defined in equation 14 and 15. In the implementation, the reconstructed rain from earlier is assumed to be accurate. That data is thus treated as the true rain with which the Kalman filter estimation is compared to to calculate the Kalman gain for every iteration. Modeling such a complex process as an AR(1) is as discussed previously a rather crude simplification. This has to be taken into account when reviewing the results.

As previously, the rain data will be transformed in order to make it more Gaussian. The transformation used is the $x_t = log(rain_t + 1)$ transform used earlier. As initial values, the non-recursive estimates were chosen as initial states.

To find good values of $R_0^{\theta,\theta}$, R_e and R_w, several combinations were tested. The best values, when looking at validation residual variance, were found to be $R_0^{\theta,\theta} = 10^{-2}$, $R_e = 10^{-6}$ and $R_w = 1$. The estimated stated can be seen in figure 16, and the final values were:

$$a_3 = -0.421(\pm 0.0310) \tag{30}$$
$$a_9 = 0.124(\pm 0.0248) \tag{31}$$
$$a_{36} = -0.583(\pm 0.0291) \tag{32}$$
$$c_1 = 0.931(\pm 0.0496) \tag{33}$$
$$c_2 = 0.883(\pm 0.0497) \tag{34}$$
$$c_{36} = -0.134(\pm 0.0522) \tag{35}$$

The 1- and 7-step predictions together with the reconstructed rain, as well as the ACFs of the resulting validation residuals can be seen in figure 17 and 18. The ACFs exhibit behaviours in line with expectations, where the 7-step residuals are very close to resembling a MA(6) and the one-step residuals are white (Monti test $30.28 < 36.42$).

To evaluate the quality of the predicted rain, the Q1 and Q2 measures introduced in section 2.2, is used. The result can be seen in table 5, and when comparing to table 3 it is clear that the Kalman prediction did not predict the rain better than the non-recursive model when looking at these measures. The ultimate goal is however not to model the rain, but the vegetation, so not much importance is placed on this.

Rain prediction	**Sum of residuals (Q_1)**	**Variance (Q_2)**	**Normalized variance**
Kalman 1-step	1522.3	4104.8	0.9046
Kalman 7-step	1551.7	4241.8	0.9348

Table 5: Prediction of rain: Variance of residuals

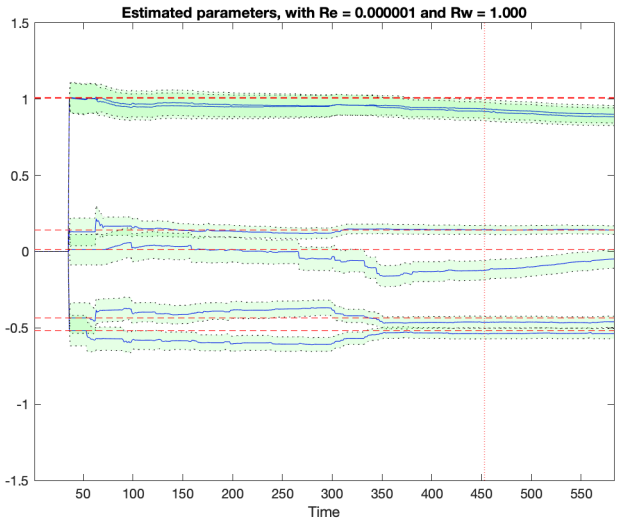
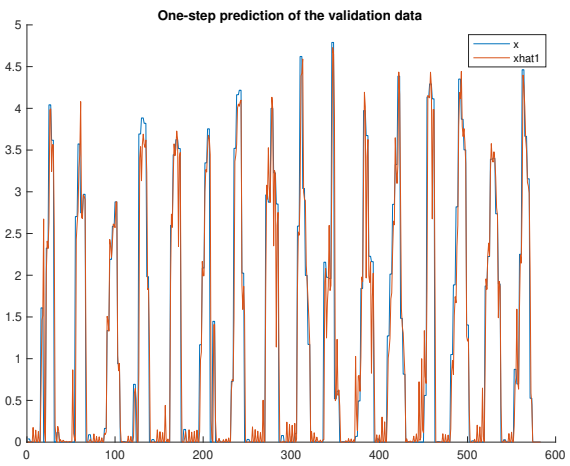
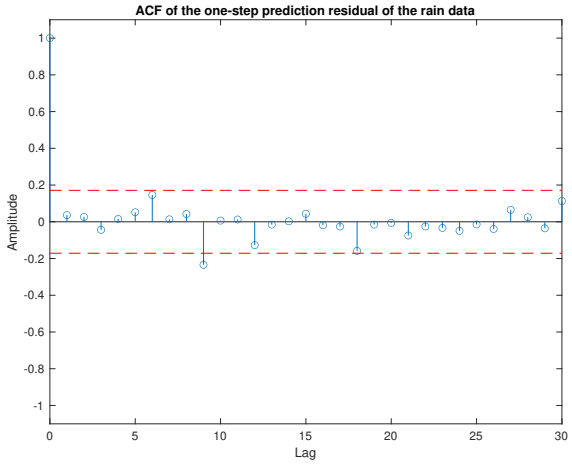


Figure 16: State estimates for the recursive modeling of rain

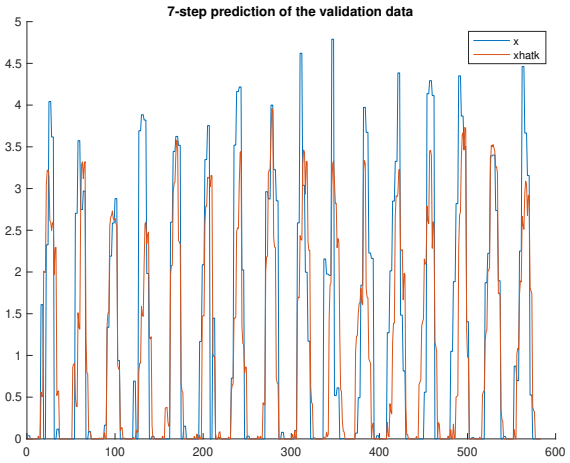


(a) 1-step prediction on validation rain data

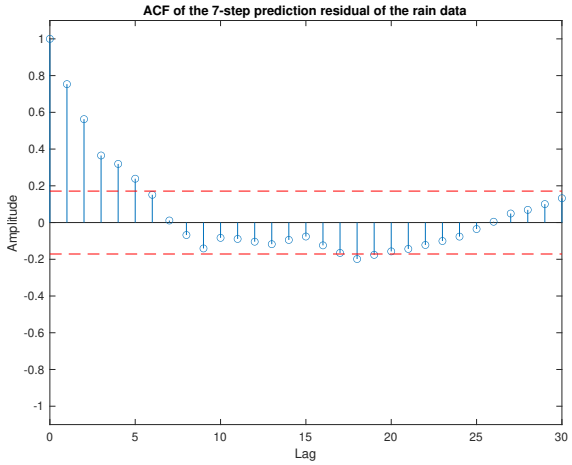


(b) ACF of one-step prediction residuals on rain data

Figure 17: One-step prediction on validation rain data using a Kalman filter



(a) 1-step prediction on validation rain data



(b) ACF of 7-step prediction residuals on rain data

Figure 18: 7-step prediction on validation rain data using a Kalman filter

3.4.2 Recursive model of vegetation

The recursively updated model is based on the results from the model with input. As can be seen in section 3.3.4, that model was found to perform best when the polynomial degrees were set as in equations 22 to 25. The Kalman filter will aim to recursively update the coefficients in those polynomials.

After choosing the state space representation on which to base the Kalman filter, there are several other parameters to tweak in order to find the optimal performance. The first one is the initial states, θ_0. These were set to the non-recursive estimates previously found by the pem function. Given that the initial states are chosen in such a way, the confidence in their accuracy is quite high, and the value of the initial covariance matrix, $R_0^{\theta,\theta}$, is thus set to a low value.

The system covariance matrix should be set according to how much the states are in general expected to change over time. To find the best value, the Kalman filter was run for different values of R_e between 10^{-8} and 10^{-2}. The performance was measured by comparing the variance of the prediction residuals on the validation data. The value of R_e that gave the lowest variance of prediction residuals on the modeling

data set was $R_e = 10^{-6}$. Finally, the value of R_w was simply set to 1 as an initial guess. By testing a few different values it was found that $R_w = 1$ was a good choice. The initial values can be seen in table 6 below.

Parameter	Initial value
θ_0	see section 3.3.4
$R_0^{\theta,\theta}$	$10^{-3} * I$
R_e	$10^{-6} * I$
R_w	1

Table 6: Initial values of hyperparameters in the Kalman filter

3.4.3 Modifying the state vector

Given that the estimated parameters do not seem to change a lot over time (the best R_e found was small), it might be possible to reduce the size of the state vector without losing performance. To check this, the Kalman filter was run for three different state space representations. One version where all 11 (non-zero) coefficients in K_A, K_B and K_C were allowed to change, one where only the K_A and K_C polynomials were estimated and the K_B polynomial was assumed constant, and finally one where only the K_A polynomial was estimated and the other two were assumed constant. The results can be seen in table 7 below.

Estimated polynomials	Variance 1-step (10^{-4})	Variance 7-step(10^{-4})
K_A, K_B, K_C	10.210	15.909
K_A, K_C	10.205	15.872
K_A	10.205	15.872

Table 7: Variance of 1- and 7-step prediction error residuals on validation data

From table 7 it can be seen that the best result was obtained when the K_B polynomial was removed from the state vector and remained constant. Whether or not the K_C polynomial was estimated or not did not impact the final results. The model where only the K_A-polynomial was recursively estimated is considered the best of these three models and is the one that will be tested on the test data as well as the Kassala data set.

3.4.4 Model performance

The state estimates can be seen in figure 20 and from it it is clear that the estimates of the coefficients in the K_A-polynomial are basically constant over time. This explains why the performances are so similar whether the parameters are estimated recursively or non-recursively. The final values of the estimated parameters are

$$a_1 = 0.852(\pm 0.0025) \tag{36}$$
$$a_{35} = 0.147(\pm 0.0025) \tag{37}$$
$$a_{36} = 0.704(\pm 0.0025) \tag{38}$$
$$a_{37} = -0.694(\pm 0.0025) \tag{39}$$

while the K_B and K_C polynomials are as mentioned previously held constant at the value found in section 3.3.4. The ACF, PACF and normplot of the model residuals are shown in figure 19. The plots look good, but there are small peaks at lag 1 in both the ACF and PACF. The normplot indicates that the confidence intervals can be trusted, and the monti test yields that the residuals are not white (Monti test $48.18 > 36.42$).

The 1- and 7-step predictions on the validation data, as well as ACFs of the resulting residuals, are plotted in figure 21 and 22 respectively.

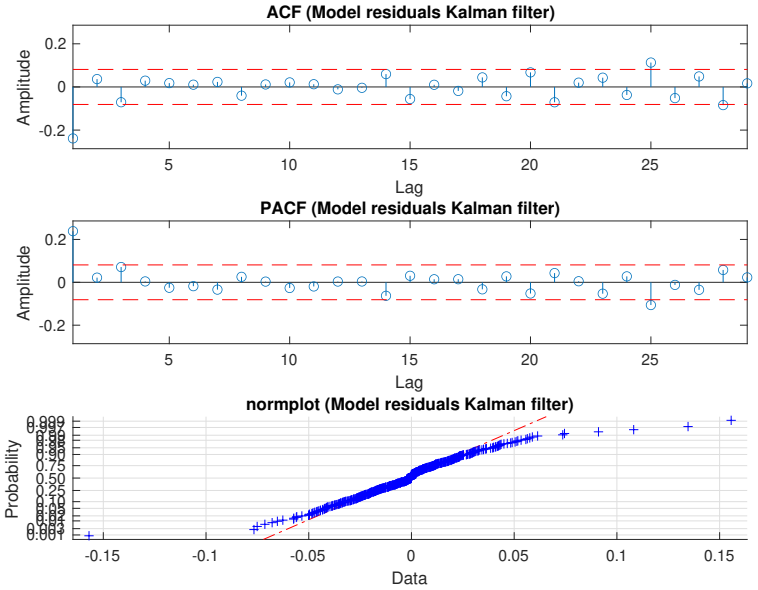


Figure 19: ACF, PACF and normplot of model residuals from Kalman filter

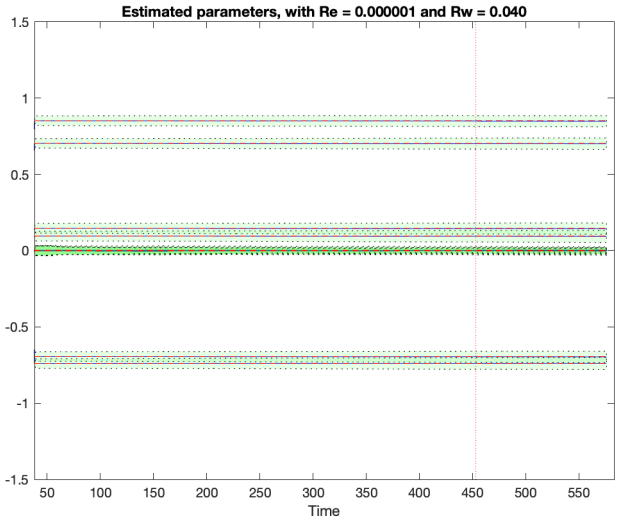
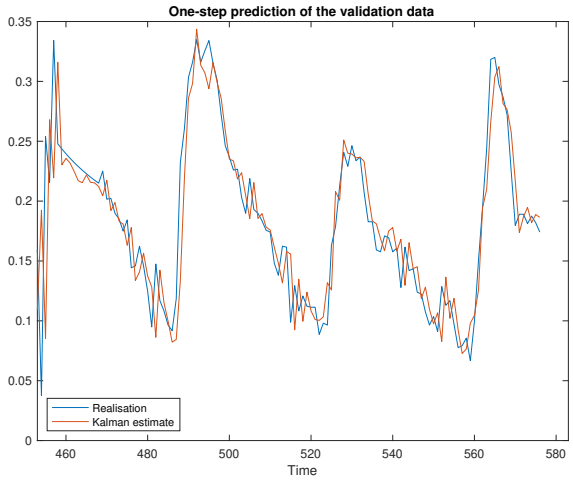
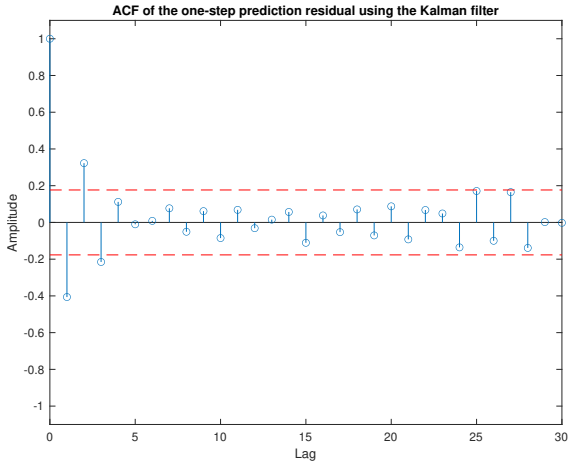


Figure 20: State estimates for the recursive modeling gof vegetation

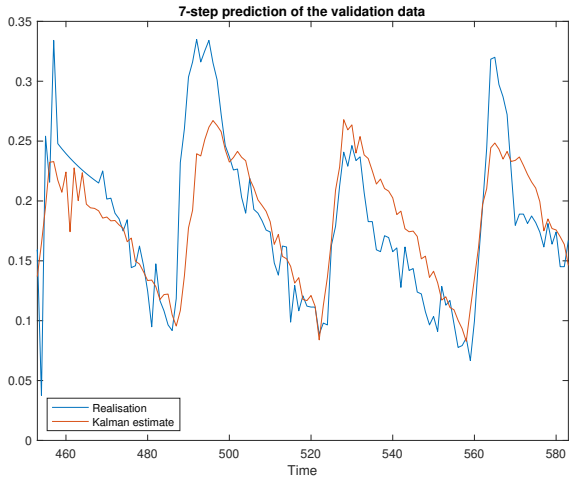


(a) 1-step prediction on validation data

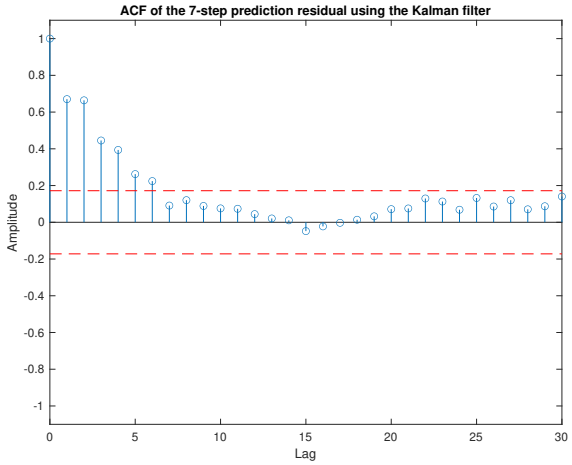


(b) ACF of one-step prediction residuals

Figure 21: One-step prediction on validation data using a Kalman filter



(a) 1-step prediction on validation data



(b) ACF of 7-step prediction residuals

Figure 22: 7-step prediction on validation data using a Kalman filter

The ACF of the 7-step prediction error residuals behaves as expected as a MA(6) process. For the one-step prediction the residuals should in theory be white, which they are not (Monti test ($40.41 > 36.42$).

The conclusion is that the recursive estimation of the parameters did not bring with it a notable improvement of the validation performance. Final comparisons between the performance of all models on the El-Geneina validation data set can be seen in table 8 and on the test data set in table 9.

Variance of prediction residuals	Variance (10^{-4})	Normalized
Naive	31.838	0.7119
SARIMA 1-Step	9.4787	0.2120
BJ 1-Step	10.034	0.2244
Kalman 1-Step	10.205	0.2282
SARIMA 7-step	15.746	0.3521
BJ 7-Step	15.886	0.3552
Kalman 7-step	15.872	0.3549

Table 8: Variance of prediction residuals on validation data

3.5 Performance on test data

Now it is time to evaluate their performances on the test data. The variance of their residuals can be seen in table 9 below:

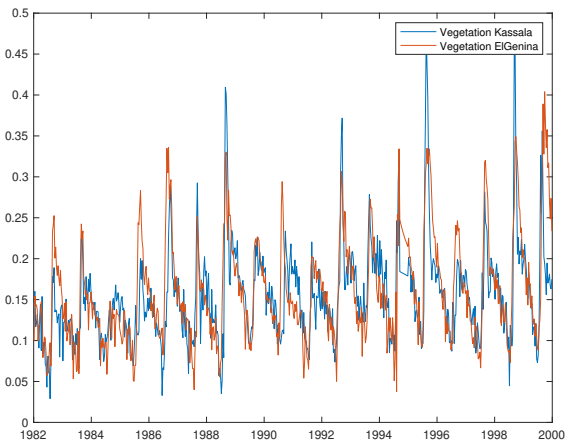
Variance of prediction residuals	Variance (10^{-4})	Normalized
Variance of data	82.955	1
Naive	13.848	0.1669
SARIMA 1-Step	7.6915	0.0927
Box-Jenkins 1-Step	8.5032	0.1025
Kalman BJ 1-Step	8.2918	0.1000
SARIMA 7-step	18.176	0.2191
Box-Jenkins 7-Step	17.809	0.2147
Kalman BJ 7-Step	17.354	0.2092

Table 9: Variance of prediction residuals on test data

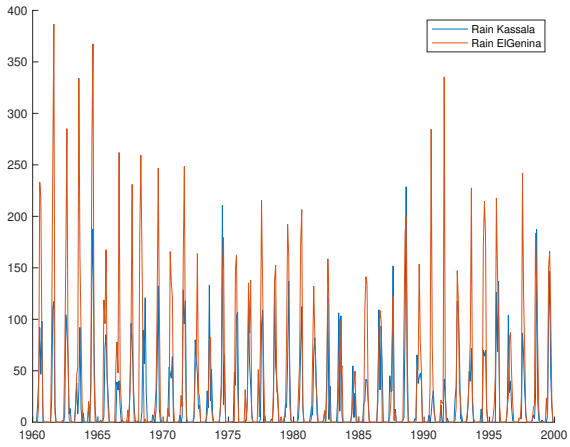
Four models (including the naive model) have been constructed and their performance on the test data have been compared. This performance measurement gives the conclusion that the SARIMA-model, the one without rain as input, is the best model on the one-step prediction. On the longer 7-step horizon, the naive model actually performs best. But the Naive models' performance is very vulnerable to change, and the performance on the test data is somewhat of a coincidence. It had relatively poor performance on both the model and validation data. Excluding the naive model makes the recursively estimated BJ-model perform best on the 7-step predictions.

4 Modeling for Kassala

The models developed will now be tested on a new dataset. The dataset Kassala contains measurements of vegetation and precipitation in the area Kassala in eastern Sudan. The collection timespan and frequency is the same as for the El-Geneina dataset. Since the precipitation data is stored in the same way in this data set as the previous one, the transformation of the data from the interval $[0, 255]$ to $[-1, 1]$ is made before any analysis is conducted. Figure 23a and 23b shows plots of the vegetation and precipitation compared to El-Geneina.



(a) Vegetation in Kassala versus Geneina



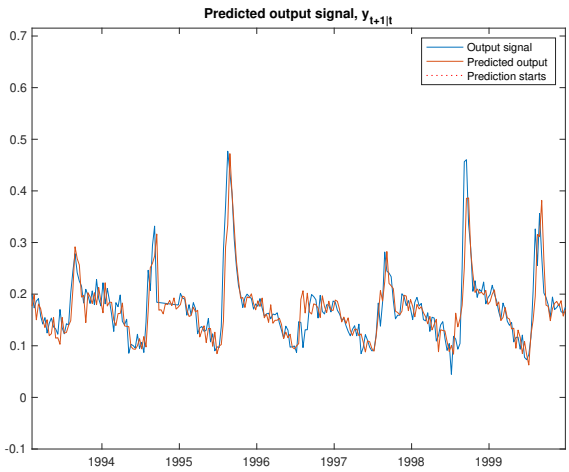
(b) Precipitation in Kassala versus Geneina

Figure 23: Comparison of vegetation and precipitation in El-Geneina and Kassala

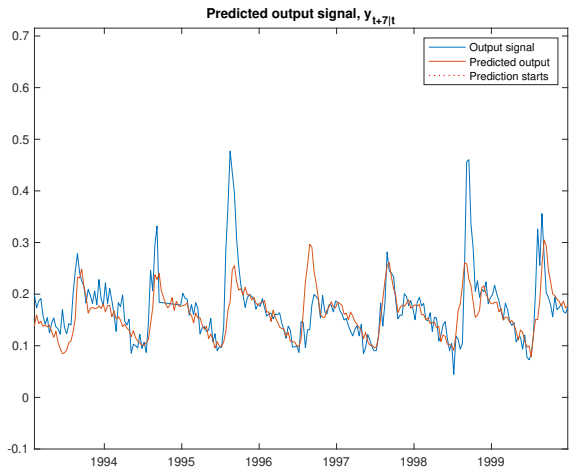
The plots shows that the vegetation seems to correlate well, but also that Kassala have somewhat higher peaks. This suggests that the models fitted to the El-Geneina data should work for the Kassala data to some degree. Although, there is a lot bigger difference in the rain data where the total amount of rain in El-Geneina is 77% higher than in Kassala.

4.1 SARIMA: Unchanged coefficients

The SARIMA-model was the best model for 1-step prediction on both the validation- and test data. As a first step, the model will be tested on the new data set without any changes. This means that the model and the polynomials remain unchanged. As the models has not been fitted for the Kassala data, it is not necessary to divide the data into different datasets. However, the residuals will be evaluated on the validation and test data as it makes it easier to compare to the re-estimated model. Also this allows the Kalman-filter to converge to appropriate parameters. The results were:



(a) 1-step prediction



(b) 7-step prediction

Figure 24: Prediction Kassala using SARIMA model

Variance of prediction residuals	Normal (10^{-4})	Normalized
Variance of data	47.726	1
Naive	52.964	1.110
SARIMA 1-Step	13.770	0.2885
SARIMA 7-step	29.319	0.6143

Table 10: Variance of residuals using SARIMA model

The ACF of the 1-step prediction looks white and the 7-step prediction looks like an MA(3).

4.2 SARIMA: Re-estimating the coefficients

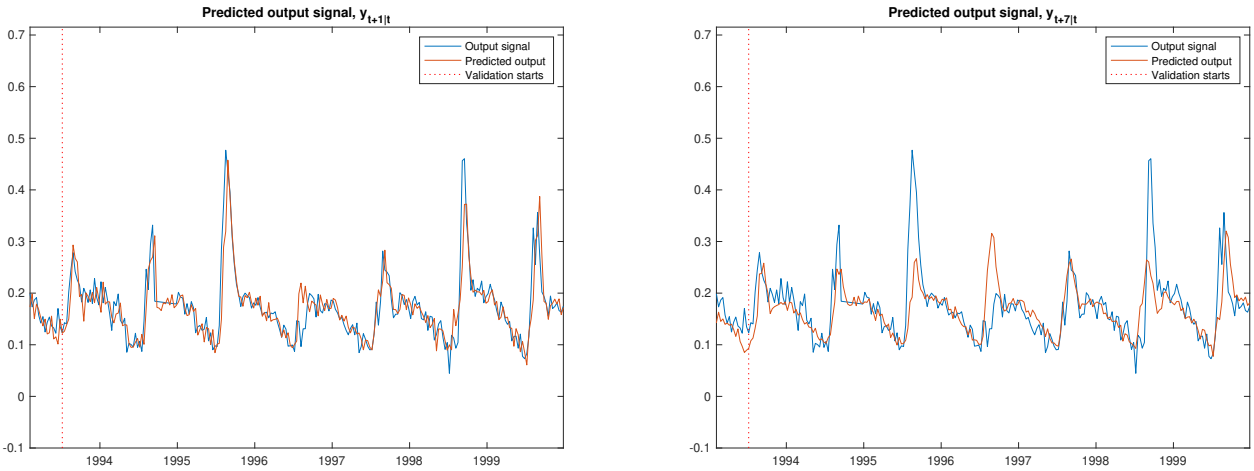
The parameters for El-Geinina were:

$$\nabla_{36} y_t (1 - 0.7543(\pm 0.03441) z^{-1}) = (1 - 0.7448(\pm 0.03523) z^{-36}) e(t) \tag{40}$$

and the parameters estimated for the Kassala data are:

$$\nabla_{36} y_t (1 - 0.6853(\pm 0.03772) z^{-1}) = (1 - 0.6832(\pm 0.03796) z^{-36}) e(t) \tag{41}$$

The coefficients are similar to the ones fitted to El-Geneina and the residuals are white. This result indicates that it is a strong correlation between the rain in Kassala and El-Genina. This is in agreement with what we saw in 4.1. The variance of the resulting plot and residuals are:



(a) 1-step prediction

(b) 7-step prediction

Figure 25: Prediction Kassala using re-estimated SARIMA model

Variance of prediction residuals	Normal	Normalized
Variance of data (10^{-4})	47.726	1
Naive	52.964	1.110
SARIMA 1-Step	14.324	0.3001
SARIMA 1-Step Re-estimated	13.770	0.2885
SARIMA 7-step	29.319	0.6143
SARIMA 7-step Re-estimated	29.938	0.6273

Table 11: Variance of residuals using re-estimated SARIMA model

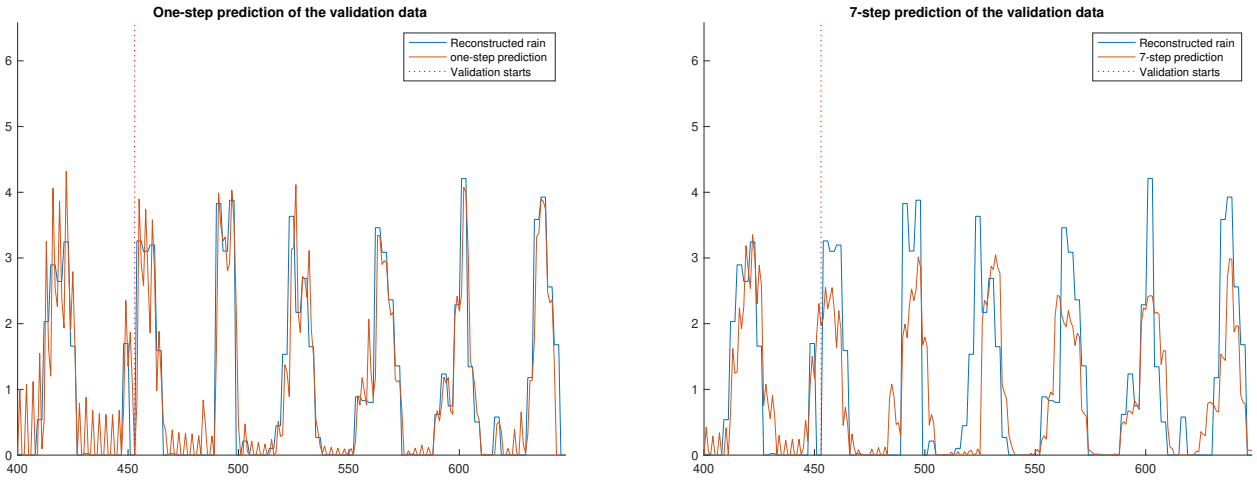
The ACF of the residuals behaves like a white process for the 1-step prediction and as an MA(3) for the 7-step prediction. Unexpectedly, the original model performed better than the re-estimated model. This shows how similar the datasets are.

4.3 The Kalman Box-Jenkins model

The Box-Jenkins model with recursively estimated parameters was found to be the best model for 7-step predictions on the test data. That model is now tested on the Kassala data set.

4.3.1 Predicting the rain

The first step is to reconstruct the rain data, and recursively estimate the coefficients in the A_3 and C_3 polynomials in order to obtain predictions for the rain. One- and 7-step predictions for the rain can be seen in figure 26. The residuals for the one-step prediction are white as expected (Monti test $17.07 < 36.42$), but the residuals for the 7-step prediction do not look like a MA(6) process. This can probably be attributed to the fact that the structure of the A_3 and C_3 polynomials are adapted to the El-Geneina data set rather than the precipitation in Kassala.



(a) 1-step prediction on Kassala rain data

(b) 1-step prediction on Kassala rain data

Figure 26: One- and 7-step predictions on Kassala rain data using a Kalman filter

4.3.2 Predicting the vegetation

The next step is to predict the vegetation by using a Kalman filter. The states are as previously the coefficients in the polynomials A_1, B and C_1 and the initial values are the ones found through the non-recursive estimate on the El-Geneina data set. The idea is that since the vegetation looks quite similar, the same values might work well as starting points. However, given that the confidence in the initial states is lower on this new data set, the value of $R_0^{\theta,\theta}$ is initially set higher than for the El-Geneina data set. As previously, the Kalman filter was run for different values of R_w and R_e and the resulting validation performances compared to find the optimal values. The state estimates can be seen in figure 27 and the validation performance of the predictions can be seen in figures 28 and 29.

Similar patterns to the ones seen for the prediction of rain can be seen in these plots. The one-step prediction residuals are white (Monti test $20.27 < 36.42$) but the 7-step residuals resemble a MA(3) rather than the desired MA(6). The plot of state estimates shows the higher degree of uncertainty required to achieve optimal performance on the Kassala data set. The final state estimates are:

$$a_1 = 0.843(\pm 0.2715) \tag{42}$$
$$a_{35} = 0.144(\pm 0.2799) \tag{43}$$
$$a_{36} = 0.702(\pm 0.2818) \tag{44}$$
$$a_{37} = -0.693(\pm 0.2779) \tag{45}$$
$$\tag{46}$$
$$b_0 = 0.001(\pm 0.0953) \tag{47}$$
$$b_1 = -0.003(\pm 0.0815) \tag{48}$$
$$b_{35} = 0.000(\pm 0.0585) \tag{49}$$
$$b_{36} = -0.001(\pm 0.0807) \tag{50}$$
$$b_{37} = 0.003(\pm 0.0626) \tag{51}$$
$$\tag{52}$$
$$c_{36} = -0.734(\pm 0.3081) \tag{53}$$
$$c_{37} = 0.086(\pm 0.3081) \tag{54}$$

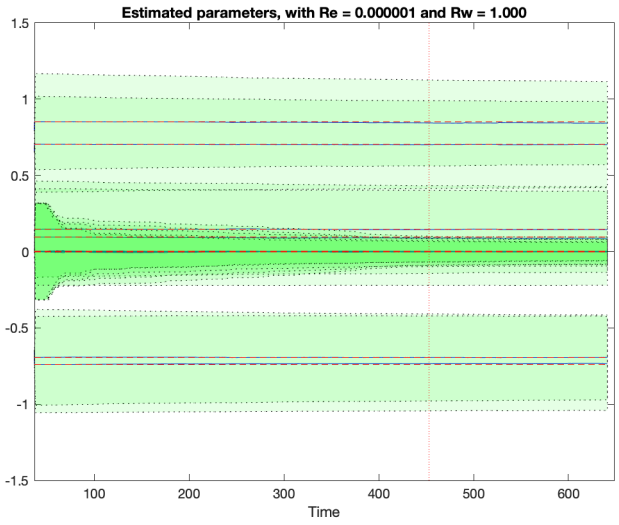
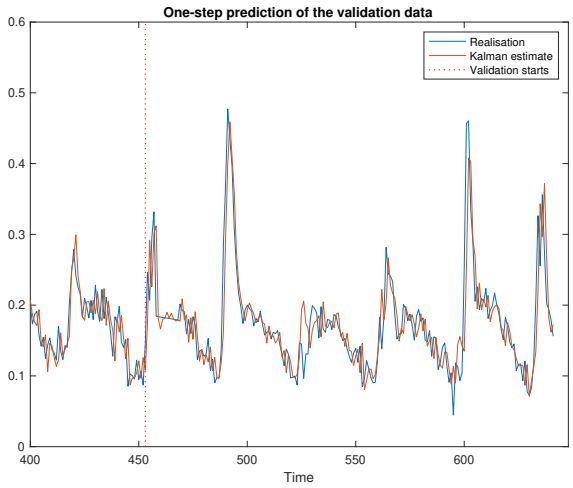
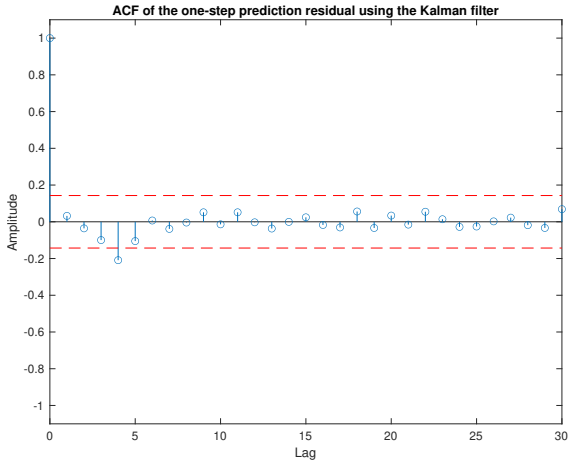


Figure 27: Kassala vegetation prediction, state estimates

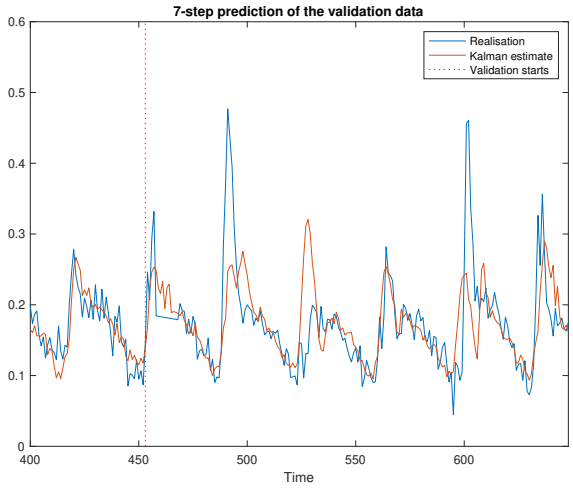


(a) 1-step prediction on Kassala rain data

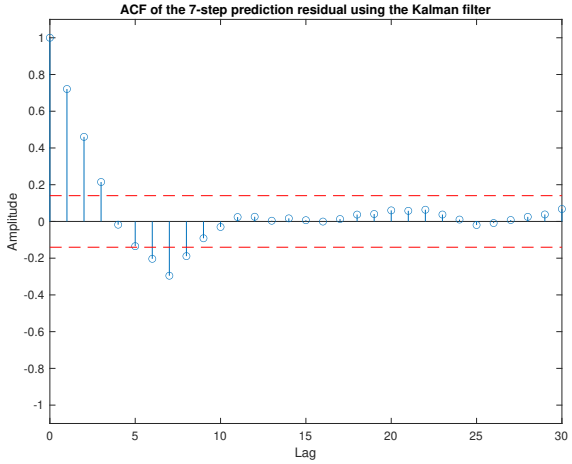


(b) ACF of one-step prediction residuals on Kassala rain data

Figure 28: One-step prediction on Kassala rain data using a Kalman filter



(a) 7-step prediction on Kassala rain data



(b) ACF of 7-step prediction residuals on Kassala rain data

Figure 29: 7-step prediction on Kassala rain data using a Kalman filter

4.4 Performance on Kassala

Variance of prediction residuals	Variance (10^{-4})	Normalized
Variance of data	47.726	1.0000
Naive	52.964	1.110
SARIMA 1-Step	13.770	0.2885
SARIMA 1-Step Re-estimated	13.770	0.2885
Kalman 1-Step	14.763	0.3093
SARIMA 7-step	29.319	0.6143
SARIMA 7-step Re-estimated	29.938	0.6273
Kalman 7-step	33.643	0.7049

Table 12: Variance of prediction error residuals on validation data

The SARIMA model performed the best on both the 1-step prediction and the 7-step prediction. Indicating that the rain was not useful us input in Kassala either.

There are notable performance differences between the normalized residuals for Kassala and the normalized residuals for the El-Geneina validation data. Note that the residuals for the Kassala data is estimated on both the validation and test data, but the normalized residuals should still be comparable. The differences could partly be explained by the fact that the structure of the models were fitted to the El-Geneina data. Another reason could be that the Kassala data is more difficult to model. Throughout the project the models have had troubles estimating the largest peaks, which the Kassala data has more of. This could cause large residuals that increase the variance of the residuals.

Even though the data in Kassala and El-Geneina has a strong correlation, there are some clear differences as well. The most distinct difference is that it rained about 77% more in El-Geneina over the total timeperiod, which is surprising since Kassala is located closer to the coast. However, one reason for this could be that El-Geneina is located on higher altitude, causing the clouds to cool down, in turn leading to precipitation. The higher precipitation probably leads to differences in vegetation, with El-Geneina having slightly higher average vegetation.

5 Discussion

5.1 El-Geneina

The SARIMA models performs marginally better on the 1-step prediction on the test data, while the recursively estimated Box-Jenkins performs marginally better on the 7-step prediction. However, the test data is a relatively small data set, and the performances should therefore not be trusted to heavily. This indicates that the rain did not necessarily improve the predictions. A model with input should not be expected to perform much better than one without. This is because of the sparse data collection. By intuition, the rain would affect the vegetation in a matter of a few days, not weeks. The causality would in that case not be picked up by a model using data every 10 days. The fact that the rain data in reality only is collected very month, and we are using a quite crude approximation of it as an AR(1) further adds to the problems. Hence, the conclusion is not necessarily that the rain data is worthless, it could also be that the reconstruction and prediction of it are too imprecise.

The recursively estimated Box-Jenkins model did not perform notably better than the non-recursive one. This supports the assumption made in the introduction that the data is stationary. This conclusion is further

supported by the fact that the performance increased slightly when parameters were removed from the state vector in the Kalman filter.

Another interesting conclusion is that the naive model performed so well on the El-Geneina test data. This shows that simply taking the vegetation level last year as the prediction for this year would work pretty well (and would save us a lot of time). However, the naive predictors weakness arises when the vegetation in two following years are very different from each other. This can be seen from its' performance on the Kassala data.

5.2 Kassala

The unchanged models performed pretty well on the Kassala data. Also the parameters changed only slightly when they were re-estimated, and the re-estimated model did not actually perform better on the test data. This further proves the strong correlation in vegetation between the two locations. The biggest difference was noted in the modeling of the rain, where data differed the most.