

# DTU - Time Series Analysis - Assignment 4: Multivariate Processes

*Emma Demarecaux (s176437)*

*07/05/2018*

Global warming is a great concern around the globe. Scientists at the Climatic Research Unit (CRU) at University of East Anglia have made several datasets with temperatures on the globe covering the past +150 years. The temperatures are expressed as anomalies from 1961-90 and in this assignment the focus will be on the changes in the average annual anomalies for the Southern and Northern hemispheres. The temperatures are estimated based on a number of measurement stations and do include measurement errors. The data includes three columns:

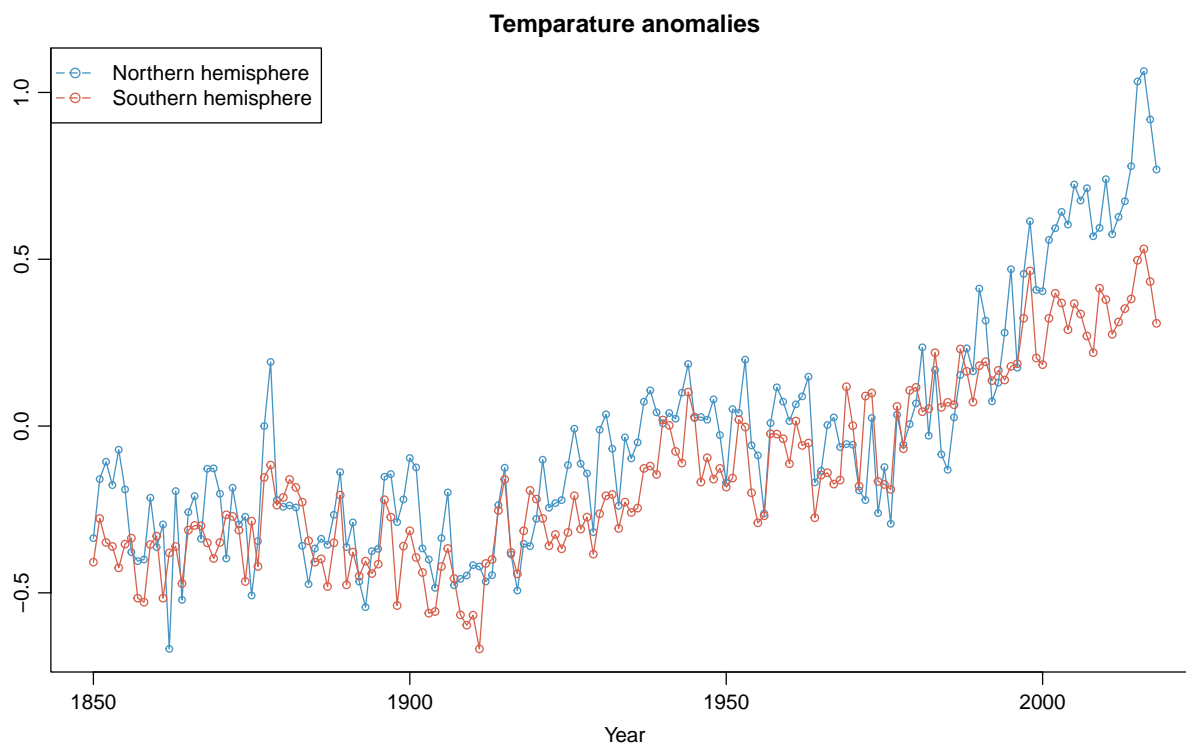
- *year*: Year for the observations
- *sh*: Temperature anomaly for the Southern hemisphere
- *nh*: Temperature anomaly for the Northern hemisphere

All observations are expected to be used in the following.

The study has been elaborated in collaboration with Adélie Marie Kookai Barre (s170075).

## Question 4.1: Presenting the data

Let us first display on the same graph the temperature anomalies for the Southern and the Northern hemispheres in order to see their relative behaviors:



It looks like the variance does not vary too much and that there are some seasonal patterns, at least oscillations, for both hemispheres.

The two time series seem to have the same behaviors over the time in terms of oscillations (their peaks are quite at the same time) and trend. They are both non stationary as there is a small increasing trend starting around 1900.

## Question 4.2: Formulating state space model

The bi-variate dynamical system is composed of two variables,  $\{x_t^n\}$  and  $\{x_t^s\}$ , respectively North and South hemispheres' temperature anomalies:

$$X_t = \begin{bmatrix} x_t^s \\ x_t^n \end{bmatrix}$$

We assume that the temperature anomalies for the two hemispheres follow independent random walks:

$$\begin{cases} x_t^s = x_{t-1}^s + e_{1,t}^s \\ x_t^n = x_{t-1}^n + e_{1,t}^n \end{cases}$$

where  $\{e_{1,t}^n\}$  and  $\{e_{1,t}^s\}$  are white noise processes with  $e_{1,t}^n \sim \mathcal{N}(0, \sigma_{e_1^n}^2)$  and  $e_{1,t}^s \sim \mathcal{N}(0, \sigma_{e_1^s}^2)$ .

Since the general form of the linear stochastic state space model is:

$$\begin{cases} \text{System equation : } X_t = AX_{t-1} + e_{1,t} \\ \text{Observation equation : } Y_t = CX_t + e_{2,t} \end{cases}$$

with:

$$A = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, C = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, e_{1,t} = \begin{bmatrix} e_{1,t}^s \\ e_{1,t}^n \end{bmatrix}, e_{2,t} = \begin{bmatrix} e_{2,t}^s \\ e_{2,t}^n \end{bmatrix}$$

where  $\{e_{1,t}\}$  and  $\{e_{2,t}\}$  are mutually uncorrelated normally distributed white noises.

We can then define the two covariance matrices as  $\Sigma_1 = \text{Var}(e_1)$  and  $\Sigma_2 = \text{Var}(e_2)$ .

In order to predict, reconstruct or interpolate this 2-dimensional state space model, we need to check if the system is observable. The requirement is:  $\text{rank}[C^T|(CA)^T] = 2$ , and we have:

$$\text{rank}[C^T|(CA)^T] = \text{rank} \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{bmatrix} = 2$$

Therefore the system is observable and we can use the Kalman filter to make reconstructions and predictions.

## Question 4.3: Kalman filtering

The Kalman filter can reconstruct and predict a multi-dimensional system  $X_t$  only with observations  $Y_t$ . Here, the aim will be to reconstruct the values of  $X_t$  from 1850 to 2018 and then predict the values of temperatures up to 2050. In order to predict these 32 values with the Kalman filter, we will consider the observations from 2019 to 2050 as missing values. The filter works as following:

## Initialization

In the first step, we need to determine reasonable initial values for the system along with the uncertainty on this initial state.

$$\begin{cases} \hat{X}_{1|0} = [\hat{x}^s & \hat{x}^n]^T \\ \Sigma_{1|0}^{XX} = V[X_1] = V_0 \\ \Sigma_{1|0}^{YY} = C\Sigma_{1|0}^{XX}C^T + \Sigma_2 \end{cases}$$

Then for  $t = 1, 2, 3, \dots$  we iterate the reconstruction and prediction step one at a time.

## Reconstruction

The aim is to reconstruct the system at time  $t$  with the observation we get at time  $t$  and what we know about the system at time  $t - 1$  such as:

$$\begin{cases} K_t = \Sigma_{t|t-1}^{XX} C^T (\Sigma_{t|t-1}^{YY})^{-1} \\ \hat{X}_{t|t} = \hat{X}_{t|t-1} + K_t (Y_t - C \hat{X}_{t|t-1}) \\ \Sigma_{t|t}^{XX} = \Sigma_{t|t-1}^{XX} - K_t \Sigma_{t|t-1}^{YY} K_t^T \end{cases}$$

$K_t$  is called the *Kalman gain*, because it determines how much the 1-step prediction error influence the update of the state estimate.

## Prediction

Then, when we have reconstructed the system at time  $t$  we can do a prediction one-step ahead. The 1-step predictions are obtained directly from the state space model:

$$\begin{cases} \hat{X}_{t+1|t} = A \hat{X}_{t|t} \\ \hat{Y}_{t+1|t} = C \hat{X}_{t+1|t} \end{cases}$$

which results in:

$$\begin{cases} \Sigma_{t+1|t}^{XX} = A \Sigma_{t|t}^{XX} A^T + \Sigma_1 \\ \Sigma_{t+1|t}^{YY} = C \Sigma_{t+1|t}^{XX} C^T + \Sigma_2 \end{cases}$$

However, what happens in the reconstruction and prediction step when the observations from 2019 to 2050 are missing ? We have  $\Sigma_{t|t-1}^{YY} \sim \infty$  and  $K_t \sim 0$ . Therefore, according to slide 16 from Lecture 12, one has:

$$\begin{cases} \hat{X}_{t|t} = \hat{X}_{t|t-1} \\ \Sigma_{t|t}^{XX} = \Sigma_{t|t-1}^{XX} \end{cases}$$

The code that was implemented for the filter is the following:

```
kalman <- function(Y,A,C,Sigma_1,Sigma_2,V0,Xhat0,n_pred)
{ ## Y      : Matrix with one observation per row. May be a vector if only one value
  ##        is observed at each time point.
  ## A, C, Sigma_1, Sigma_2 : Matrices as described in the definition of the Kalman
  ##        filter.
  ## V0      : Initial covariance of Sigma_xx_{1|0}
  ## Xhat0   : Initial estimate of the state X_{1|0}
  ## n_pred: For making forecasts beyond the data

  ## DIMENSION OF THE SYSTEM
```

```

n_obs <- dim(Y) #Number of observed parameters
n_state <- dim(A)[1] #Number of state variables

## INITIALISATION
X_hat <- Xhat0
Sigma_xx <- V0
Sigma_yy <- C%%Sigma_xx%%t(C)+Sigma_2

## PREPARATION OF THE ARRAYS USED TO SAVE THE RESULTS
X_rec <- array(dim=c(n_obs[1]+n_pred-1,n_state))
X_pred <- array(dim=c(n_obs[1]+n_pred,n_state))
K.out <- array(dim=c(dim(Sigma_xx%%t(C)%%solve(Sigma_yy)),n_obs[1]))
Sigma_xx_rec <- array(dim=c(dim(Sigma_xx),n_obs[1]+n_pred-1))
Sigma_yy_rec <- array(dim=c(dim(Sigma_yy),n_obs[1]+n_pred-1))
Sigma_xx_pred <- array(dim=c(dim(Sigma_xx),n_obs[1]+n_pred))
Sigma_yy_pred <- array(dim=c(dim(Sigma_yy),n_obs[1]+n_pred))

## RECONSTRUCTION AND PREDICTION OVER THE DATA SET
for(tt in 1:n_obs[1])
{
  K <- Sigma_xx%%t(C)%%solve(Sigma_yy)
  ## Reconstruction
  X_hat <- X_hat+K%%(Y[tt,]-C %% as.matrix(X_hat))
  X_rec[tt,] <- X_hat
  Sigma_xx <- Sigma_xx-K%%Sigma_yy%%t(K)
  ## Saving of Reconstruction error variance-covariances matrices
  Sigma_xx_rec[,tt] <- Sigma_xx
  Sigma_yy_rec[,tt] <- Sigma_yy
  ## Prediction
  X_hat <- A%%X_hat
  X_pred[tt+1,] <- X_hat
  Sigma_xx <- A%%Sigma_xx%%t(A)+Sigma_1
  Sigma_yy <- C%%Sigma_xx%%t(C)+Sigma_2
  ###Saving of Prediction error variance-covariances matrices
  K.out[,tt] <- K
  Sigma_xx_pred[,tt+1] <- Sigma_xx
  Sigma_yy_pred[,tt+1] <- Sigma_yy
}

## RECONSTRUCTION AND PREDICTION OVER THE PREDICTION SET
for(tt in (n_obs[1]+1):(n_obs[1]+n_pred-1))
{
  ## Reconstruction
  X_hat <- X_hat
  X_rec[tt,] <- X_hat
  Sigma_xx <- Sigma_xx
  ## Saving of Reconstruction error variance-covariances matrices
  Sigma_xx_rec[,tt] <- Sigma_xx
  Sigma_yy_rec[,tt] <- Sigma_yy
  ## Prediction
  X_hat <- A%%X_hat
  X_pred[tt+1,] <- X_hat
  Sigma_xx <- A%%Sigma_xx%%t(A)+Sigma_1
  Sigma_yy <- C%%Sigma_xx%%t(C)+Sigma_2

```

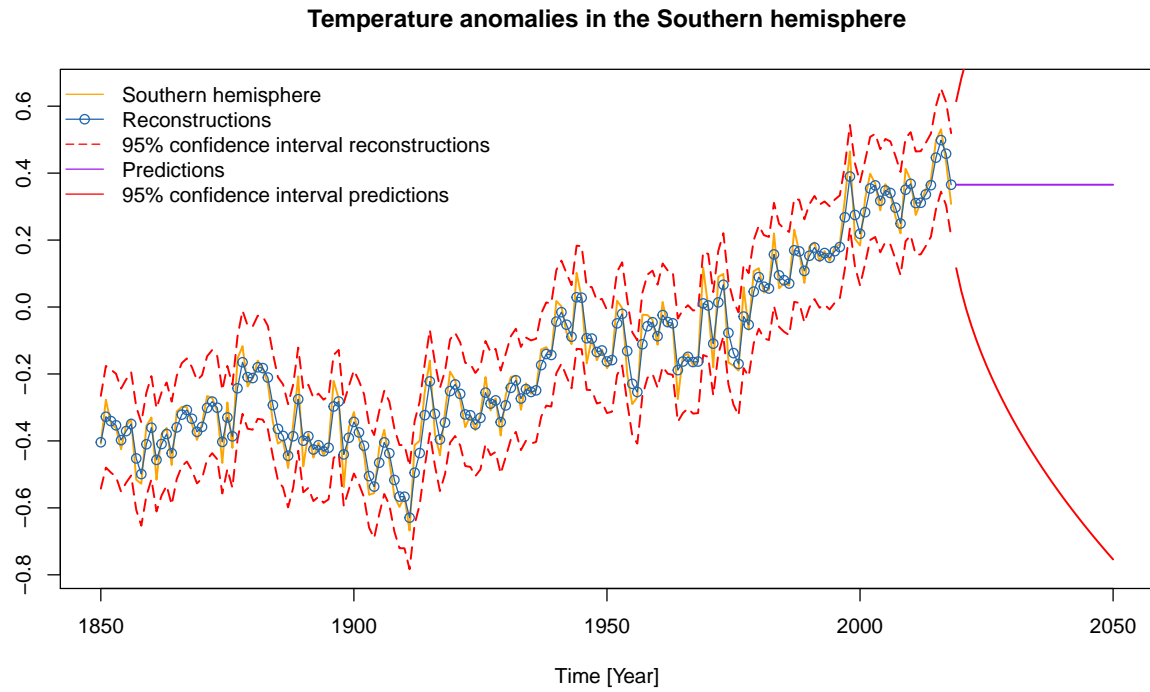
```

###Saving of Prediction error variance-covariances matrices
Sigma_xx_pred [,,tt+1] <- Sigma_xx
Sigma_yy_pred [,,tt+1] <- Sigma_yy
}
outputs <- list(rec=X_rec,pred=X_pred ,K=K.out,Sigma_xx_rec =Sigma_xx_rec ,
               Sigma_yy_rec =Sigma_yy_rec ,Sigma_xx_pred =Sigma_xx_pred ,
               Sigma_yy_pred =Sigma_yy_pred )
return(outputs)
}

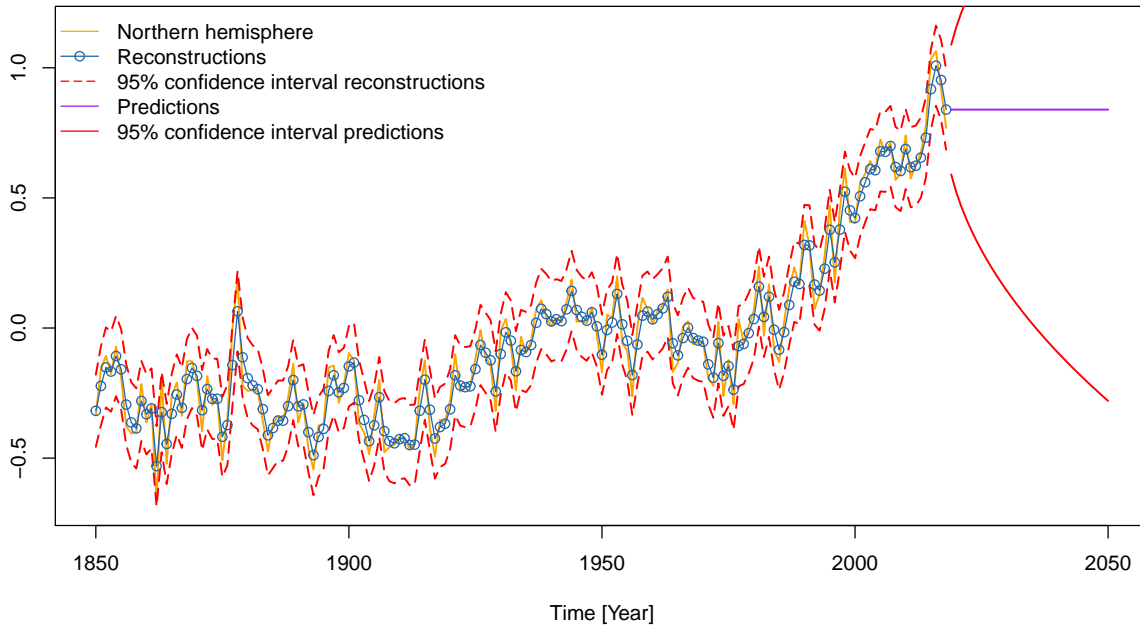
```

In this section, the initial parameters are  $\hat{X}_{1|0} = \begin{bmatrix} -0.4 \\ -0.3 \end{bmatrix}$  and  $V_0 = \Sigma_1 = \Sigma_2 = \begin{bmatrix} 0.01 & 0 \\ 0 & 0.01 \end{bmatrix}$

The reconstruction of the anomalies is presented in the following figure which include a 95% confidence interval and predictions until 2050:



### Temperature anomalies in the Northern hemisphere



The reconstruction is quite faithful to the real temperature anomalies, which was not really expected as random walk models are quite simple and as the initial values with their uncertainty were not optimized. Maybe, with more accurate parameters, the reconstruction could yield to an almost perfect fit.

The log-likelihood for these parameters is of 511.7029.

This value will be used as reference to make sure that results are better when the parameters of the system are optimized.

Here is the table with predictions for years 2020, 2030, 2040 and 2050 for the Southern hemisphere:

| ##                  | 2020       | 2030        | 2040        | 2050        |
|---------------------|------------|-------------|-------------|-------------|
| ## Predictions      | 0.36531809 | 0.36531809  | 0.36531809  | 0.36531809  |
| ## Lower boundaries | 0.04818925 | -0.33089781 | -0.56680983 | -0.75406046 |
| ## Upper boundaries | 0.68244692 | 1.06153399  | 1.29744601  | 1.48469664  |

Here is the table with predictions for years 2020, 2030, 2040 and 2050 for the Northern hemisphere:

| ##                  | 2020       | 2030       | 2040        | 2050        |
|---------------------|------------|------------|-------------|-------------|
| ## Predictions      | 0.83930561 | 0.83930561 | 0.83930561  | 0.83930561  |
| ## Lower boundaries | 0.52217678 | 0.14308971 | -0.09282231 | -0.28007294 |
| ## Upper boundaries | 1.15643444 | 1.53552151 | 1.77143353  | 1.95868416  |

Given the way random walk is defined, predictions are expected to be made using the observation at the previous time step. That is why we have constant predictions with this model. It can be concluded that the random walk does not accurately enough describe the behavior of the temperature anomalies in both hemispheres: a trend should at least be considered as the next further step as well as parameters optimisation.

## Question 4.4: Optimize parameters

In this section we estimate the initial parameters,  $\hat{X}_{1|0}$  and  $V_0$ , along with  $\Sigma_1$  and  $\Sigma_2$ , which are respectively the system noise and the observation noise.

In this case we have to estimate 5 parameters:

$$\hat{X}_{1|0} = \begin{bmatrix} \hat{x}_0^s \\ \hat{x}_0^n \end{bmatrix}, V_0 = \begin{bmatrix} \sigma^2 & 0 \\ 0 & \sigma^2 \end{bmatrix}, \Sigma_1 = \begin{bmatrix} \sigma_1^2 & 0 \\ 0 & \sigma_1^2 \end{bmatrix} \text{ and } \Sigma_2 = \begin{bmatrix} \sigma_2^2 & 0 \\ 0 & \sigma_2^2 \end{bmatrix}$$

Usually, coefficients in  $V_0, \Sigma_1$  and  $\Sigma_2$  might not be the same compared to what is written in those equations, however, in order to make the optimization easier, we assume that they are the same in this study.

Let  $\mathcal{Y}_{N*}$  contain the available observations and let  $\theta$  contain the parameters of the model. As the random variables  $\mathcal{Y}_{N*}|\mathcal{Y}_{N*-1}$  and  $\mathcal{Y}_{N*-1}$  are independent we have:

$$L(\theta; \mathcal{Y}_{N*}) = f(Y_{N*}|\mathcal{Y}_{N*-1}, \theta) f(Y_{N*-1}|\mathcal{Y}_{N*-2}, \theta) \dots f(Y_1|\theta)$$

The conditional densities can be found using the Kalman filter:

$$\begin{cases} \tilde{Y}_i = Y_i - \hat{Y}_i \\ R_i = \Sigma_{i|i-1}^{YY} \end{cases}$$

Then the likelihood function can be expressed as:

$$\log L(\theta; \mathcal{Y}_{N*}) = -\frac{1}{2} \sum_{i=1}^N (\log \det R_i + \tilde{Y}_i^T R_i^{-1} \tilde{Y}_i) + c$$

As we want to find parameters that maximize the likelihood function we search parameters which minimize the following function:

$$\log L(\theta; \mathcal{Y}_{N*}) = \frac{1}{2} \sum_{i=1}^N (\log \det R_i + \tilde{Y}_i^T R_i^{-1} \tilde{Y}_i)$$

using the function *optim* from R. This function needs initial values for the parameters to be optimized over and a function to be minimized. It returns the best set of parameters found.

This is the following function that we are trying to minimize:

```
my.obj.4 <- function(PAR){
  Kro <- kalman(Y=Y, A=A, C=C,
               Sigma_1=diag(exp(PAR[4]),2),
               Sigma_2=diag(exp(PAR[5]),2),
               Xhat0=c(PAR[1:2]),
               V0=diag(exp(PAR[3]),2),
               n_pred=n_pred)
  Y_tilde <- Y[-1,] - Kro$pred[2:n,]
  R = Kro$Sigma_yy_pred[,2:n]

  P1 = matrix(NA, nrow = 168, ncol = 1)
  P2 = matrix(NA, nrow = 168, ncol = 1)

  for (i in 1:168){
    P1[i] = log(det(R[,i]))
    P2[i] = t(Y_tilde[i,]) %*% solve(R[,i]) %*% Y_tilde[i,]
  }
  return(0.5 * sum(P1 + P2))}
```

The results are:

```
Xhat0
```

```
## [1] -0.3333821 -0.1592750
```

V0

```
##          [,1]      [,2]
## [1,] 4.585686e-09 0.000000e+00
## [2,] 0.000000e+00 4.585686e-09
```

Sigma\_1

```
##          [,1]      [,2]
## [1,] 0.002449211 0.000000000
## [2,] 0.000000000 0.002449211
```

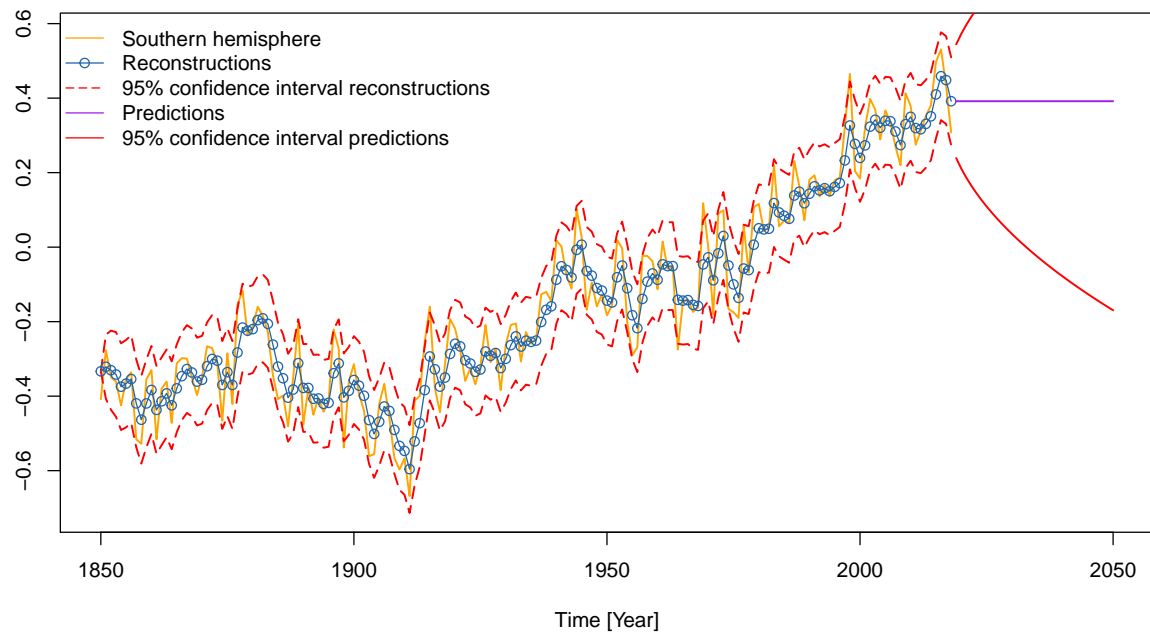
Sigma\_2

```
##          [,1]      [,2]
## [1,] 0.008957307 0.000000000
## [2,] 0.000000000 0.008957307
```

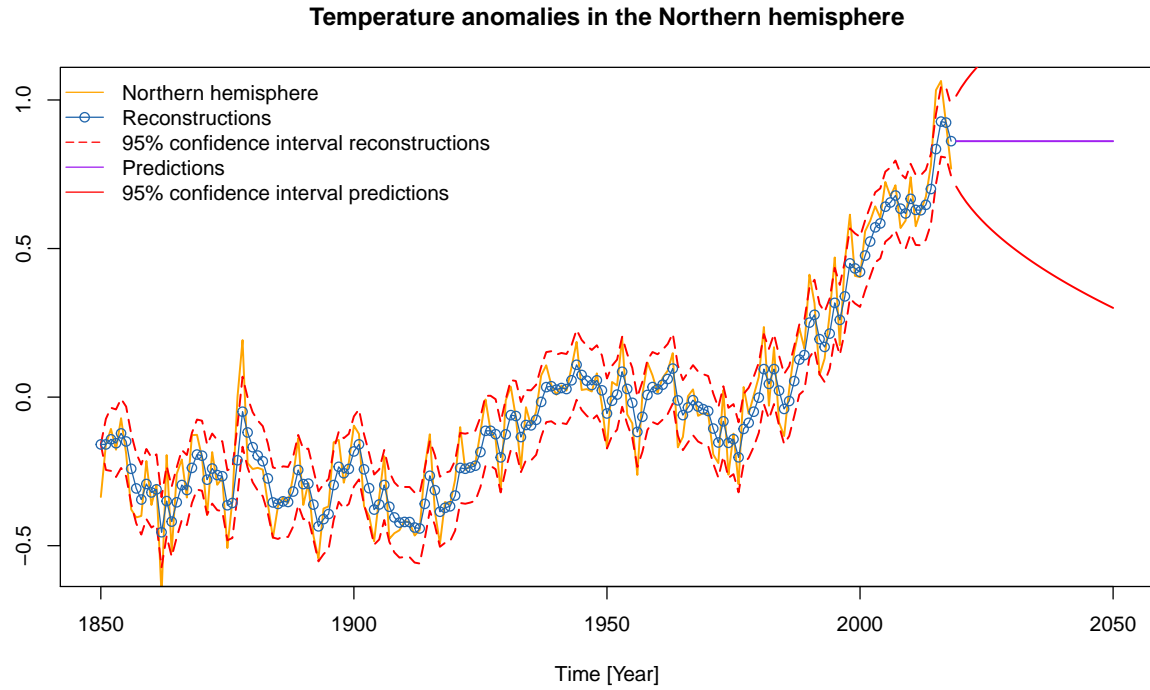
The log-likelihood for these parameters is of 537.7567.

The results of the Kalman filter with these parameters are:

### Temperature anomalies in the Southern hemisphere







The reconstruction seems on the graph to fail to capture the peaks, more than what was observed in Question 4.3 even though the likelihood is much better (538>512). However we can observe that the confidence interval is more narrow.

Here is the table with predictions for years 2020, 2030, 2040 and 2050 for the Southern hemisphere:

| ##                  | 2020       | 2030       | 2040        | 2050        |
|---------------------|------------|------------|-------------|-------------|
| ## Predictions      | 0.39172345 | 0.39172345 | 0.39172345  | 0.39172345  |
| ## Lower boundaries | 0.21086330 | 0.03563942 | -0.07825663 | -0.16949550 |
| ## Upper boundaries | 0.57258361 | 0.74780748 | 0.86170353  | 0.95294241  |

Here is the table with predictions for years 2020, 2030, 2040 and 2050 for the Northern hemisphere:

| ##                  | 2020      | 2030      | 2040      | 2050      |
|---------------------|-----------|-----------|-----------|-----------|
| ## Predictions      | 0.8613688 | 0.8613688 | 0.8613688 | 0.8613688 |
| ## Lower boundaries | 0.6805087 | 0.5052848 | 0.3913887 | 0.3001499 |
| ## Upper boundaries | 1.0422290 | 1.2174528 | 1.3313489 | 1.4225878 |

Both constant predictions decreased by 0.03. As we noticed in the first question of the report, the two time series have quite the same behaviors regarding peaks and trend. Maybe we should adjust the model to allow the system noise to be correlated as it seems to be from the data.

## Question 4.5: Optimize parameters - with correlation

In this case we have added system noise correlation. This means that non-diagonal elements in  $\Sigma_1$  might not be null. We can re-write this matrix with this new correlation. We have to estimate 6 parameters:

$$\hat{X}_{1|0} = \begin{bmatrix} \hat{x}_0^s \\ \hat{x}_0^n \end{bmatrix}, V_0 = \begin{bmatrix} \sigma^2 & 0 \\ 0 & \sigma^2 \end{bmatrix}, \Sigma_1 = \begin{bmatrix} \sigma_1^2 & \rho\sigma_1^2 \\ \rho\sigma_1^2 & \sigma_1^2 \end{bmatrix} \text{ and } \Sigma_2 = \begin{bmatrix} \sigma_2^2 & 0 \\ 0 & \sigma_2^2 \end{bmatrix}$$

We must be aware that we have to find the  $\rho$  parameter with the interval  $[-1,1]$ .

The results are:

Xhat0

```
## [1] -0.3063059 -0.1388570
```

V0

```
##           [,1]      [,2]
## [1,] 4.259312e-08 0.000000e+00
## [2,] 0.000000e+00 4.259312e-08
```

rho

```
## [1] 0.9220057
```

This value of  $\rho$  is between  $[-1,1]$  which means that  $\Sigma_1$  is well-defined.

Sigma\_1

```
##           [,1]      [,2]
## [1,] 0.007117157 0.006562059
## [2,] 0.006562059 0.007117157
```

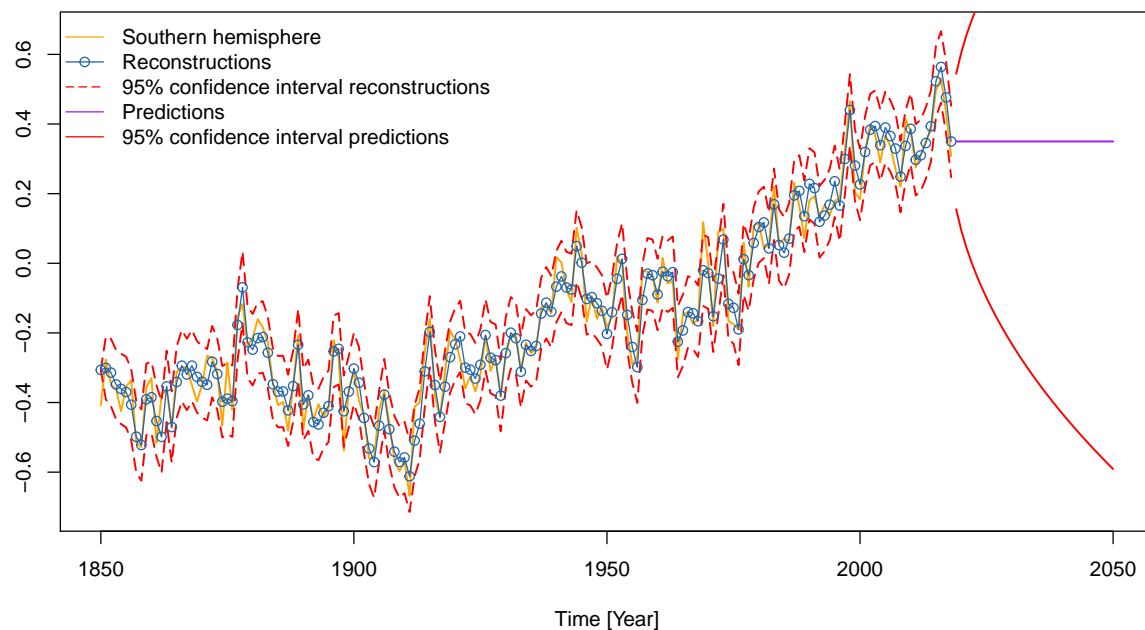
Sigma\_2

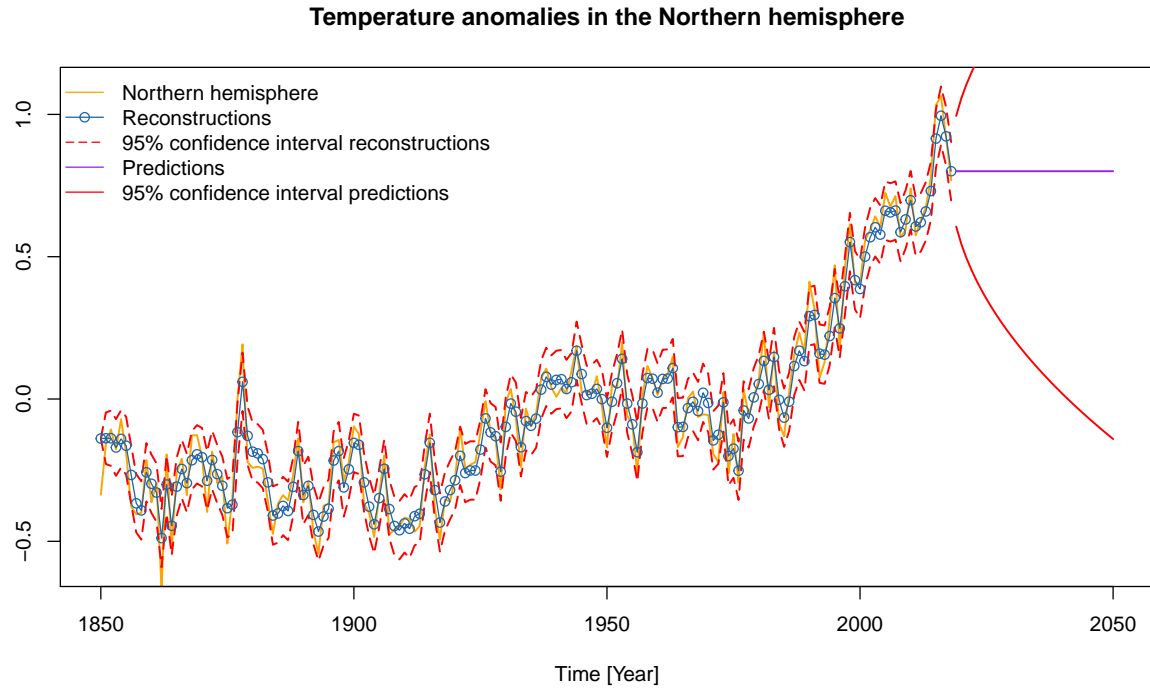
```
##           [,1]      [,2]
## [1,] 0.005214618 0.000000000
## [2,] 0.000000000 0.005214618
```

The log-likelihood for these parameters is of 563.7918 ( $>538$ ).

The results of the Kalman filter with these parameters are:

### Temperature anomalies in the Southern hemisphere





When including system noise correlation, the reconstructions fit much better the system variables, with a narrow and accurate confidence interval.

Here is the table with predictions for years 2020, 2030, 2040 and 2050 for the Southern hemisphere:

| ##                  | 2020       | 2030        | 2040        | 2050        |
|---------------------|------------|-------------|-------------|-------------|
| ## Predictions      | 0.34988181 | 0.34988181  | 0.34988181  | 0.34988181  |
| ## Lower boundaries | 0.09454709 | -0.23201034 | -0.43242321 | -0.59107717 |
| ## Upper boundaries | 0.60521653 | 0.93177396  | 1.13218682  | 1.29084078  |

Here is the table with predictions for years 2020, 2030, 2040 and 2050 for the Northern hemisphere:

| ##                  | 2020       | 2030       | 2040       | 2050        |
|---------------------|------------|------------|------------|-------------|
| ## Predictions      | 0.80040343 | 0.80040343 | 0.80040343 | 0.80040343  |
| ## Lower boundaries | 0.54506871 | 0.21851127 | 0.01809841 | -0.14055555 |
| ## Upper boundaries | 1.05573815 | 1.38229558 | 1.58270844 | 1.74136240  |

The observations have decreased again but the random walk can't accurately enough describe the behavior of the temperature anomalies in both hemispheres as we have noticed since question 1 that there is an increasing trend which should at least be considered.

## Question 4.6: Formulating state space model with common trend

The aim here is to extend the state space model by including a common trend, as it was indeed missing in the previous model. To do so, another variable will be considered,  $u_t$ , and is assumed to follow a random walk as well. The vector  $X_t$  will then be defined as

$$X_t = \begin{bmatrix} x_t^s \\ x_t^n \\ u_t \end{bmatrix}$$

Which will result in the following system :

$$\begin{cases} x_t^s = x_{t-1}^s + u_{t-1} + e_{1,t}^s \\ x_t^n = x_{t-1}^n + u_{t-1} + e_{1,t}^n \\ u_t = u_{t-1} + e_{1,t}^u \end{cases}$$

where  $\{e_{1,t}^s\}$ ,  $\{e_{1,t}^n\}$  and  $\{e_{1,t}^u\}$  are white noise processes with  $e_1^s \sim \mathcal{N}(0, \sigma_{e_1^s}^2)$ ,  $e_1^n \sim \mathcal{N}(0, \sigma_{e_1^n}^2)$  and  $e_1^u \sim \mathcal{N}(0, \sigma_{e_1^u}^2)$ . Then, the state space model for the dynamical system of three dimensions is defined as follows:

$$\begin{cases} \text{System equation : } X_t = AX_{t-1} + e_{1,t} \\ \text{Observation equation : } Y_t = CX_t + e_{2,t} \end{cases}$$

where one has :

$$A = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{bmatrix}, C = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}, e_{1,t} = \begin{bmatrix} e_{1,t}^s \\ e_{1,t}^n \\ e_{1,t}^u \end{bmatrix}, e_{2,t} = \begin{bmatrix} e_{2,t}^s \\ e_{2,t}^n \end{bmatrix}$$

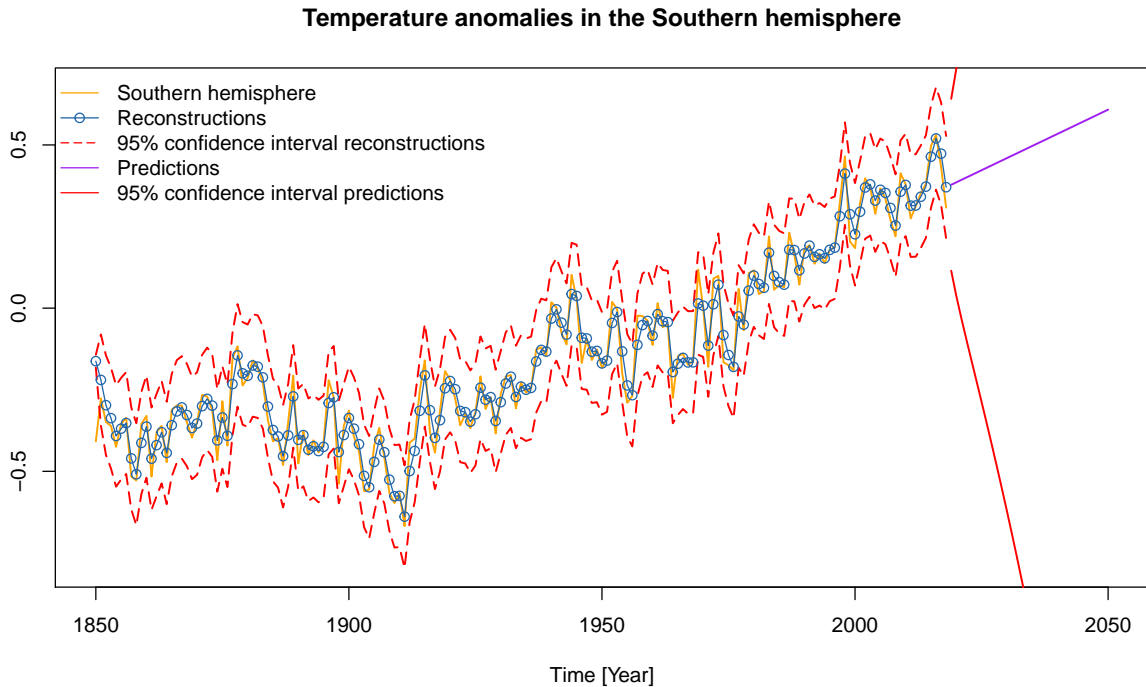
where  $\{e_{1,t}\}$  and  $\{e_{2,t}\}$  are mutually uncorrelated normally distributed white noises. The two covariance matrices are  $\Sigma_1 = \text{Var}(e_1)$  and  $\Sigma_2 = \text{Var}(e_2)$ .

In order to predict, reconstruct or interpolate this 3-dimensional state space model, we need to check if the system is observable. The requirement is:  $\text{rank}[C^T|(CA)^T|(CA^2)^T] = 3$  and we have:

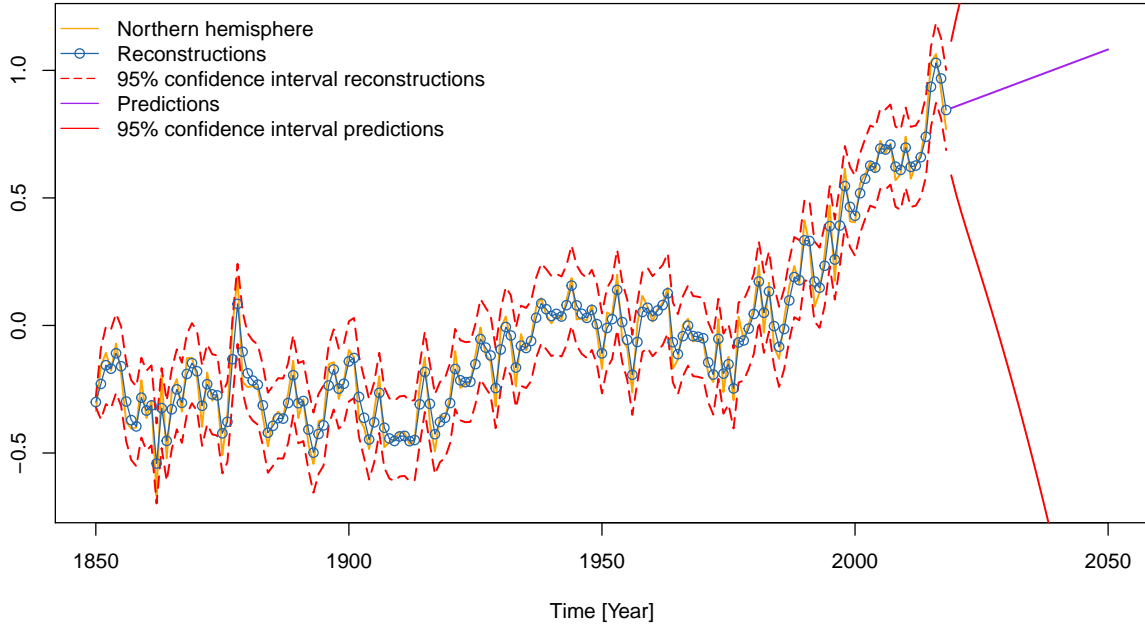
$$\text{rank}[C^T|(CA)^T|(CA^2)^T] = \text{rank} \begin{bmatrix} 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 2 & 2 \end{bmatrix} = 3$$

Therefore the system is observable and we can use the Kalman filter to make reconstructions and predictions.

The reconstruction of the anomalies is presented in the following figure which include a 95% confidence interval and the predictions until 2050:



### Temperature anomalies in the Northern hemisphere



The reconstruction is very similar to the observations. The log-likelihood for these parameters is of 503.0725. This value is much less than the one from the other models but this may result from the fact that the starting point used here is quite far from the real initial values especially for the trend for which we associated a quite low uncertainty: 0.01. Nonetheless, the reconstruction is very accurate.

Here is the table with predictions for years 2020, 2030, 2040 and 2050 for the Southern hemisphere:

| ##                  | 2020       | 2030        | 2040        | 2050        |
|---------------------|------------|-------------|-------------|-------------|
| ## Predictions      | 0.38540525 | 0.45964485  | 0.53388445  | 0.60812406  |
| ## Lower boundaries | 0.03803383 | -0.62118010 | -1.39611682 | -2.31747429 |
| ## Upper boundaries | 0.73277668 | 1.54046981  | 2.46388573  | 3.53372240  |

Here is the table with predictions for years 2020, 2030, 2040 and 2050 for the Northern hemisphere:

| ##                  | 2020      | 2030       | 2040       | 2050       |
|---------------------|-----------|------------|------------|------------|
| ## Predictions      | 0.8593928 | 0.9336324  | 1.0078720  | 1.0821116  |
| ## Lower boundaries | 0.5120214 | -0.1471926 | -0.9221293 | -1.8434868 |
| ## Upper boundaries | 1.2067642 | 2.0144573  | 2.9378733  | 4.0077099  |

Regarding the predictions, we can see on the figure above, and more accurately in the tables, that they consist of a small increasing straight line with a very large confidence interval. This extended state space model which includes a common trend (which also follows a random walk), seems to describe better the behavior of the temperature anomalies in both hemispheres. However, maybe we will get a better confidence interval when optimizing parameters in the next part and adding system noise correlation.

## Question 4.7: Optimize parameters - with correlation and common trend

For this optimization, some matrices have been extended to 3 dimensions. Now, we have to estimate 9 parameters:

$$\hat{X}_{1|0} = \begin{bmatrix} \hat{x}_0^s \\ \hat{x}_0^n \\ \hat{u}_0 \end{bmatrix}, V_0 = \begin{bmatrix} \sigma^2 & 0 & 0 \\ 0 & \sigma^2 & 0 \\ 0 & 0 & \sigma_u^2 \end{bmatrix}, \Sigma_1 = \begin{bmatrix} \sigma_1^2 & \rho\sigma_1^2 & 0 \\ \rho\sigma_1^2 & \sigma_1^2 & 0 \\ 0 & 0 & \sigma_{1.u}^2 \end{bmatrix} \text{ and } \Sigma_2 = \begin{bmatrix} \sigma_2^2 & 0 \\ 0 & \sigma_2^2 \end{bmatrix}$$

where  $\rho \in [-1, 1]$ . We assume that the new trend is not correlated with the temperature anomalies.

The results are:

Xhat0

```
## [1] -0.319044110 -0.153440123 0.004710392
```

V0

```
##           [,1]           [,2]           [,3]
## [1,] 1.217954e-06 0.000000e+00 0.000000e+00
## [2,] 0.000000e+00 1.217954e-06 0.000000e+00
## [3,] 0.000000e+00 0.000000e+00 1.007197e-08
```

rho

```
## [1] 0.9220948
```

$\rho$  is still comprised within  $[-1,1]$  so the covariance matrix  $\Sigma_1$  is well-defined.

Sigma\_1

```
##           [,1]           [,2]           [,3]
## [1,] 0.006927676 0.006387974 0.000000e+00
## [2,] 0.006387974 0.006927676 0.000000e+00
## [3,] 0.000000000 0.000000000 1.259291e-08
```

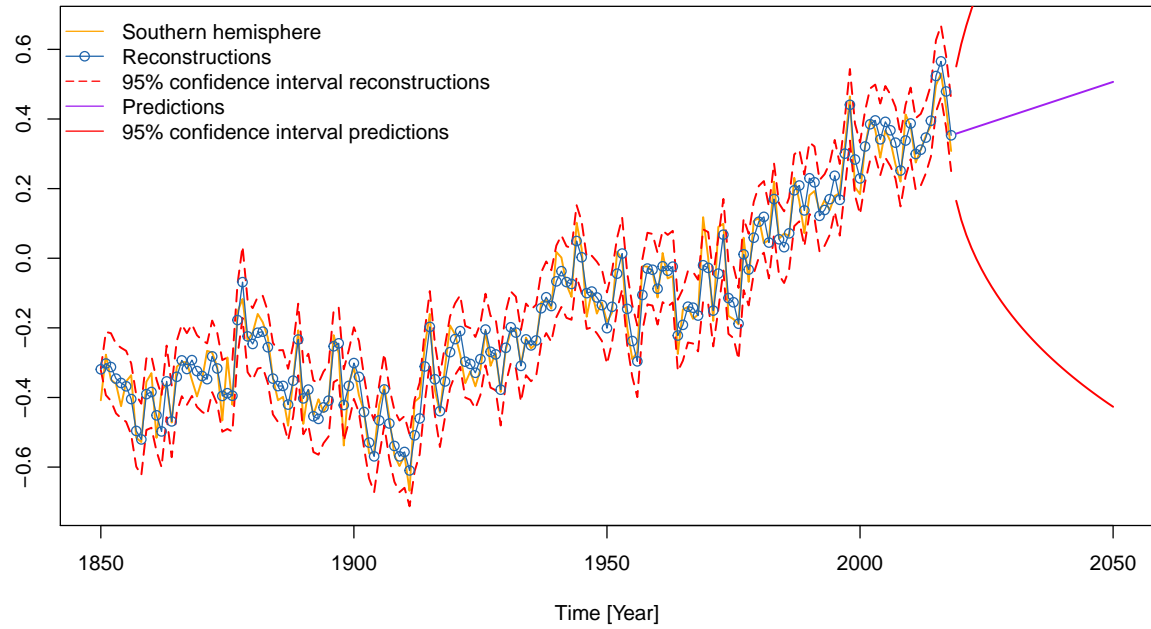
Sigma\_2

```
##           [,1]           [,2]
## [1,] 0.00530879 0.00000000
## [2,] 0.00000000 0.00530879
```

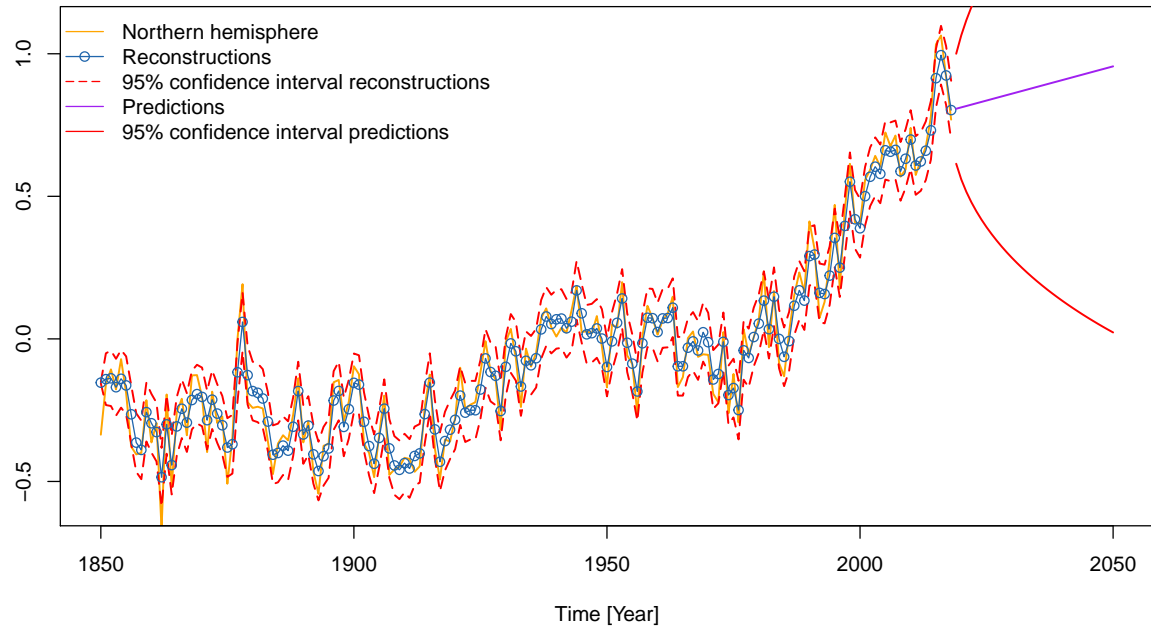
The log-likelihood for these parameters is of 564.0293 ( $> 563.7918$ ) which is a bit better than the value we had for the bi-variate system with system noise correlation and optimized parameters.

The results of the Kalman filter with these parameters are:

**Temperature anomalies in the Southern hemisphere**



**Temperature anomalies in the Northern hemisphere**



We can observe when we look at the figure above that the reconstruction is very close to the system variables even though there is dimension reduction between  $X_t$  and  $Y_t$ . The confidence interval is very narrow and this model seems to fit very well the temperature anomalies for both hemispheres.

Here is the table with predictions for years 2020, 2030, 2040 and 2050 for the Southern hemisphere:

| ##                  | 2020      | 2030       | 2040       | 2050       |
|---------------------|-----------|------------|------------|------------|
| ## Predictions      | 0.3627260 | 0.4106848  | 0.4586436  | 0.5066025  |
| ## Lower boundaries | 0.1100356 | -0.1647912 | -0.3160880 | -0.4267025 |
| ## Upper boundaries | 0.6154164 | 0.9861609  | 1.2333753  | 1.4399075  |

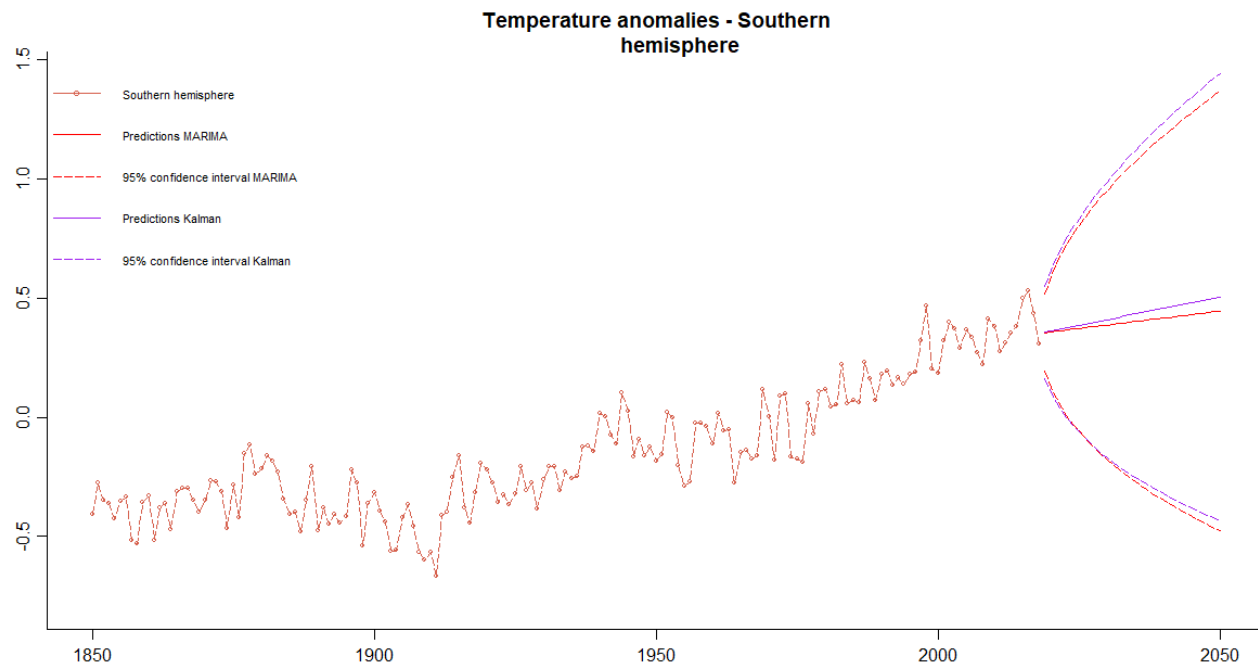
Here is the table with predictions for years 2020, 2030, 2040 and 2050 for the Northern hemisphere:

| ##                  | 2020       | 2030       | 2040       | 2050       |
|---------------------|------------|------------|------------|------------|
| ## Predictions      | 0.81197598 | 0.85993480 | 0.90789362 | 0.95585244 |
| ## Lower boundaries | 0.55928558 | 0.28445874 | 0.13316202 | 0.02254743 |
| ## Upper boundaries | 1.06466638 | 1.43541087 | 1.68262523 | 1.88915746 |

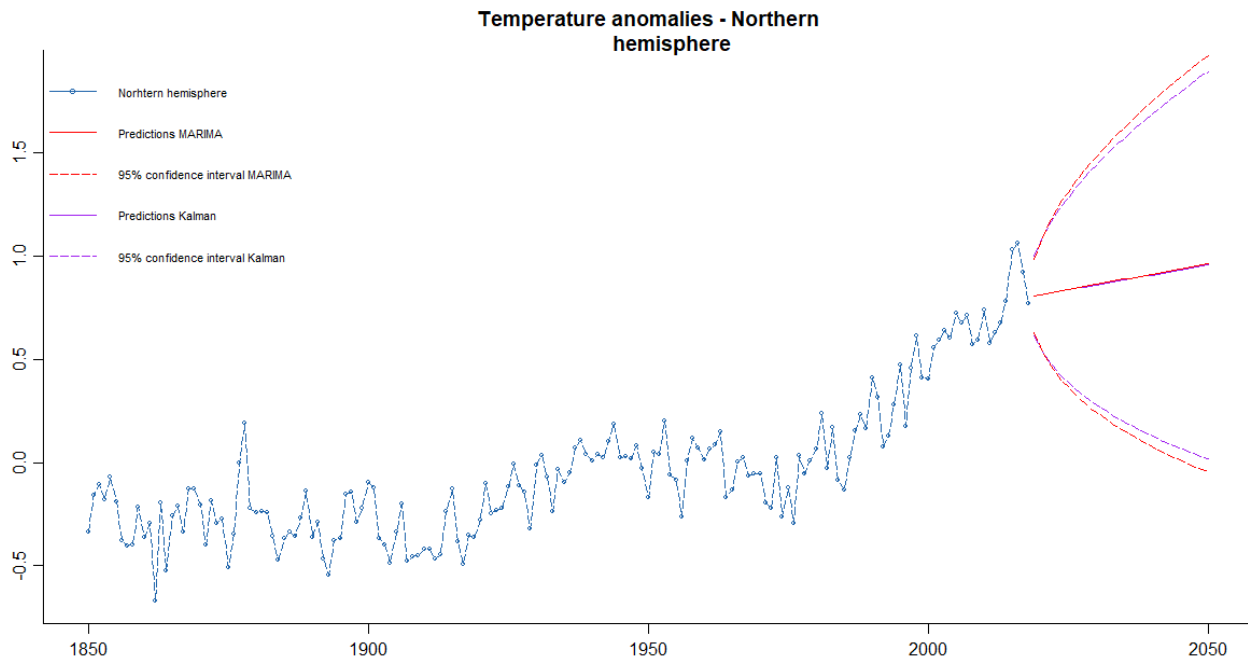
In regards to the predictions, one can see that here a trend is clearly appearing in the predictions and seems to be more realistic according to the past observations. The confidence interval is narrower than in the previous model which shows a better accuracy for these predictions.

## Question 4.8

From the previous conclusions, it seems quite obvious that the more advanced the model is, the better the predictions perform. It was therefore needed to take into account correlation between the North and South hemispheres as well as consider a common slope in order to model global warming from 1900 as well as possible. Another way possible to predict the anomalies of temperature was by implementing a MARIMA model and it was tried to do so here. In order to eliminate the slope for the variables  $s_h$  and  $n_h$  that makes the series non-stationnary, it was needed to consider differenciation in our model. Therefore, a simple MARIMA(0,1,0) was implemented on the system reconstructions  $X_t$  and then used to predict the anomalies from 2019 to 2050. The results are displayed against the ones obtained with the last Kalman filter and the corresponding optimized parameters:







The two models yield to very similar results with almost identical slopes and confidence intervals. This is due to the fact that both models are quite equivalent: while the Kalman filter captures and includes a trend for the predictions, the MARIMA model eliminates the slope and works on differentiated data to predict future observation and includes the slope back again. Though, it is probably easier, with the knowledge seen in class and the functions seen on R to implement a MARIMA model to capture the oscillations in addition to the trend we were able to find.

As I said in Question 4.4, usually, diagonal coefficients in  $V_0$ ,  $\Sigma_1$  and  $\Sigma_2$  might not be the same compared to what is written in those equations:

$$V_0 = \begin{bmatrix} \sigma^2 & 0 & 0 \\ 0 & \sigma^2 & 0 \\ 0 & 0 & \sigma_u^2 \end{bmatrix}, \Sigma_1 = \begin{bmatrix} \sigma_1^2 & \rho\sigma_1^2 & 0 \\ \rho\sigma_1^2 & \sigma_1^2 & 0 \\ 0 & 0 & \sigma_{1,u}^2 \end{bmatrix} \text{ and } \Sigma_2 = \begin{bmatrix} \sigma_2^2 & 0 \\ 0 & \sigma_2^2 \end{bmatrix}$$

Trying to optimize more parameters in these matrices might be a way for the Kalman filter to perform better.

## Appendix

```
library(RColorBrewer)
library(xtable)
library(marima)
library(matlib)

Data <- read.table(file="A4_annual.txt", sep="\t", header=TRUE)
n<-length(Data$year)
n_pred<-2050-Data$year[n]

Time <- data.frame('t'=c(1:n))
```

```

Data <- cbind(Time,Data)
Year <- 1850:2050

#----- Question 1

blues <- rev(brewer.pal(11, "RdBu"))

par(mfrow=c(2,1))
par(mgp=c(2, 0.7,0), mar=c(3,3,2,1))

plot(Data$year, Data$nh,xlab='Year', ylab='',
      main="Temperature anomalies - Northern hemisphere",
      type = 'o',pch=1,lty=5,bty='l',cex=0.5,
      col=blues[2], cex.axis = 0.7)

plot(Data$year, Data$sh,xlab='Year', ylab='',
      main="Temperature anomalies - Southern hemisphere",
      type = 'o',pch=1,lty=5,bty='l',cex=0.5,
      col=blues[9], cex.axis = 0.7)

par(mfrow=c(1,1))
par(mgp=c(2, 0.7,0), mar=c(3,3,2,1))
plot(Data$year,Data$nh,type = 'o',pch=1,lty=1,bty='l',cex=0.7,
      col=blues[3], cex.axis = 1,main='Temparature anomalies',ylab='',xlab='Year')
lines(Data$year,Data$sh,type = 'o',cex=0.8,col=blues[9])

legend('topleft',legend = c('Northern hemisphere','Southern hemisphere'),
      col = c(blues[3],blues[9]),lty = c(5,5), pch = c(1,1),cex=0.7,
      bty = n)

#----- Question 3

TSO <- -0.4
TNO <- -0.3
A=matrix(nrow = 2, ncol = 2,byrow=T,
          c(1,0,
            0,1))
C=matrix(nrow = 2, ncol = 2, byrow = T,
          c(1,0,
            0,1))
Sigma_1 <- matrix(c(0.01,0,0,0.01),nrow=2)
Sigma_2 <- matrix(c(0.01,0,0,0.01),nrow=2)
V0=Sigma_1
X <- matrix(nrow=2,ncol=n)
X[,1] <- c(TSO,TNO)
Xhat0<- X[,1]
Y<-cbind(Data$sh,Data$nh)

kal <- kalman(Y=Y, A=A, C=C, Sigma_1=Sigma_1, Sigma_2=Sigma_2, Xhat0=Xhat0, V0=V0, n_pred=n_pred)

#-----sh
Reconstructions(kal,0)
#-----nh
Reconstructions(kal,1)

```

```

Loglike(kal)
#sh
Pred<-kal$pred
Pred<-data.frame('x'=Pred[,1], 'y'=Pred[,2])
Pred_Var<-kal$Sigma_xx_pred
sd_pred<-sqrt(Pred_Var[1,1,])
t95_minus_pred<-qnorm(0.025)*sd_pred
t95_plus_pred<-qnorm(0.975)*sd_pred
Pred_plus<-Pred$x+t95_plus_pred
Pred_minus<-Pred$x+t95_minus_pred
table <- matrix(cbind(Pred$x[c(171,181,191,201)],Pred_minus[c(171,181,191,201)],
                      Pred_plus[c(171,181,191,201)]),nrow=3,byrow=TRUE)
rownames(table) <- c("Predictions","Lower boundaries","Upper boundaries")
colnames(table) <- c("2020","2030","2040","2050")
table <- as.table(table)
table
#nh
sd_pred<-sqrt(Pred_Var[2,2,])
t95_minus_pred<-qnorm(0.025)*sd_pred
t95_plus_pred<-qnorm(0.975)*sd_pred
Pred_plus<-Pred$y+t95_plus_pred
Pred_minus<-Pred$y+t95_minus_pred
table <- matrix(cbind(Pred$y[c(171,181,191,201)],Pred_minus[c(171,181,191,201)],
                      Pred_plus[c(171,181,191,201)]),nrow=3,byrow=TRUE)
rownames(table) <- c("Predictions","Lower boundaries","Upper boundaries")
colnames(table) <- c("2020","2030","2040","2050")
table <- as.table(table)
table
#----- Question 4
xpar = -0.4
ypar = -0.3
stdpar = log(0.01)
Sigma1_par = log(0.01)
Sigma2_par = log(0.01)
PAR <- c(xpar,ypar,stdpar,Sigma1_par,Sigma2_par)

Kmopt <- optim(PAR,my.obj.4,method = "L-BFGS-B")

Xhat0 = Kmopt$par[1:2]
V0=diag(exp(Kmopt$par[3]),2)
Sigma_1 = diag(exp(Kmopt$par[4]),2)
Sigma_2 = diag(exp(Kmopt$par[5]),2)

kal <- kalman(Y=Y, A=A, C=C, Sigma_1=Sigma_1, Sigma_2=Sigma_2,
             Xhat0=Xhat0, V0=V0, n_pred=n_pred)
Loglike(kal)
Reconstructions(kal,0)
Reconstructions(kal,1)

#sh
Pred<-kal$pred
Pred<-data.frame('x'=Pred[,1], 'y'=Pred[,2])
Pred_Var<-kal$Sigma_xx_pred

```

```

sd_pred<-sqrt(Pred_Var[1,1])
t95_minus_pred<-qnorm(0.025)*sd_pred
t95_plus_pred<-qnorm(0.975)*sd_pred
Pred_plus<-Pred$x+t95_plus_pred
Pred_minus<-Pred$x+t95_minus_pred
table <- matrix(cbind(Pred$x[c(171,181,191,201)],Pred_minus[c(171,181,191,201)],
                      Pred_plus[c(171,181,191,201)]),nrow=3,byrow=TRUE)
rownames(table) <- c("Predictions","Lower boundaries","Upper boundaries")
colnames(table) <- c("2020","2030","2040","2050")
table <- as.table(table)
table
#nh
sd_pred<-sqrt(Pred_Var[2,2])
t95_minus_pred<-qnorm(0.025)*sd_pred
t95_plus_pred<-qnorm(0.975)*sd_pred
Pred_plus<-Pred$y+t95_plus_pred
Pred_minus<-Pred$y+t95_minus_pred
table <- matrix(cbind(Pred$y[c(171,181,191,201)],Pred_minus[c(171,181,191,201)],
                      Pred_plus[c(171,181,191,201)]),nrow=3,byrow=TRUE)
rownames(table) <- c("Predictions","Lower boundaries","Upper boundaries")
colnames(table) <- c("2020","2030","2040","2050")
table <- as.table(table)
table
#----- Question 5
#This time we had a rho qui doit appartenir à [-1,1]
xpar = -0.3
ypar = -0.4
stdpar = log(0.01)
Sigma1_par = log(0.01)
Sigma2_par = log(0.01)
rho = 0.2
PAR <- c(xpar,ypar,stdpar,Sigma1_par,Sigma2_par,rho)

Kmopt <- optim(PAR,my.obj.5,method = "L-BFGS-B")

Xhat0 = Kmopt$par[1:2]
V0=diag(exp(Kmopt$par[3]),2)
rho=Kmopt$par[6]
Sigma_1 = matrix(c(exp(Kmopt$par[4])*exp(Kmopt$par[4]),
                  exp(Kmopt$par[4])*exp(Kmopt$par[4])*rho,
                  exp(Kmopt$par[4])*exp(Kmopt$par[4])*rho,
                  exp(Kmopt$par[4])*exp(Kmopt$par[4])),nrow=2)
Sigma_2 = diag(exp(Kmopt$par[5]),2)

kal <- kalman(Y=Y, A=A, C=C, Sigma_1=Sigma_1, Sigma_2=Sigma_2,
             Xhat0=Xhat0, V0=V0, n_pred=n_pred)
Loglike(kal)
#-----sh
Reconstructions(kal,0)
#----- nh
Reconstructions(kal,1)

#sh

```

```

Pred<-kal$pred
Pred<-data.frame('x'=Pred[,1], 'y'=Pred[,2])
Pred_Var<-kal$Sigma_xx_pred
sd_pred<-sqrt(Pred_Var[1,1,])
t95_minus_pred<-qnorm(0.025)*sd_pred
t95_plus_pred<-qnorm(0.975)*sd_pred
Pred_plus<-Pred$x+t95_plus_pred
Pred_minus<-Pred$x+t95_minus_pred
table <- matrix(cbind(Pred$x[c(171,181,191,201)],Pred_minus[c(171,181,191,201)],
                      Pred_plus[c(171,181,191,201)]),nrow=3,byrow=TRUE)
rownames(table) <- c("Predictions","Lower boundaries","Upper boundaries")
colnames(table) <- c("2020","2030","2040","2050")
table <- as.table(table)
table
#nh
sd_pred<-sqrt(Pred_Var[2,2,])
t95_minus_pred<-qnorm(0.025)*sd_pred
t95_plus_pred<-qnorm(0.975)*sd_pred
Pred_plus<-Pred$y+t95_plus_pred
Pred_minus<-Pred$y+t95_minus_pred
table <- matrix(cbind(Pred$y[c(171,181,191,201)],Pred_minus[c(171,181,191,201)],
                      Pred_plus[c(171,181,191,201)]),nrow=3,byrow=TRUE)
rownames(table) <- c("Predictions","Lower boundaries","Upper boundaries")
colnames(table) <- c("2020","2030","2040","2050")
table <- as.table(table)
table
#-----Question 6
TS0 <- -0.3
TNO <- -0.16
TU0 <- 0
A=matrix(nrow = 3, ncol = 3,byrow=T,
         c(1,0,1,
           0,1,1,
           0,0,1))
C=matrix(nrow = 2, ncol = 3, byrow = T,
         c(1,0,0,
           0,1,0))
Sigma_1 <- matrix(c(0.01,0,0,0,0.01,0,0,0,0.0001),nrow=3)
Sigma_2 <- matrix(c(0.01,0,0,0.01),nrow=2)
V0<-matrix(c(0.0001,0,0,0,0.0001,0,0,0,0.0001),nrow=3)
X <- matrix(nrow=3,ncol=n)
X[,1] <- c(TNO,TS0,TU0)
Xhat0<- X[,1]
Y<-cbind(Data$sh,Data$nh)

kal <- kalman(Y=Y, A=A, C=C, Sigma_1=Sigma_1, Sigma_2=Sigma_2, Xhat0=Xhat0, V0=V0, n_pred=n_pred)

Reconstructions(kal,0)
Reconstructions(kal,1)
Loglike2(kal)

#sh
Pred<-kal$pred

```

```

Pred<-data.frame('x'=Pred[,1], 'y'=Pred[,2])
Pred_Var<-kal$Sigma_xx_pred
sd_pred<-sqrt(Pred_Var[1,1,])
t95_minus_pred<-qnorm(0.025)*sd_pred
t95_plus_pred<-qnorm(0.975)*sd_pred
Pred_plus<-Pred$x+t95_plus_pred
Pred_minus<-Pred$x+t95_minus_pred
table <- matrix(cbind(Pred$x[c(171,181,191,201)],Pred_minus[c(171,181,191,201)],
                      Pred_plus[c(171,181,191,201)]),nrow=3,byrow=TRUE)
rownames(table) <- c("Predictions","Lower boundaries","Upper boundaries")
colnames(table) <- c("2020","2030","2040","2050")
table <- as.table(table)
table
#nh
sd_pred<-sqrt(Pred_Var[2,2,])
t95_minus_pred<-qnorm(0.025)*sd_pred
t95_plus_pred<-qnorm(0.975)*sd_pred
Pred_plus<-Pred$y+t95_plus_pred
Pred_minus<-Pred$y+t95_minus_pred
table <- matrix(cbind(Pred$y[c(171,181,191,201)],Pred_minus[c(171,181,191,201)],
                      Pred_plus[c(171,181,191,201)]),nrow=3,byrow=TRUE)
rownames(table) <- c("Predictions","Lower boundaries","Upper boundaries")
colnames(table) <- c("2020","2030","2040","2050")
table <- as.table(table)
table
#-----Question 7
xpar = -0.3
ypar = -0.16
upar = 0.2
stdpar = log(0.001)
Sigma1_par = log(0.001)
Sigma1_paru = log(0.02)
Sigma2_par = log(0.001)
V0_paru = log(0.01)
rho=0.92
A<-matrix(nrow = 3, ncol = 3,byrow=T,
          c(1,0,1,
            0,1,1,
            0,0,1))
C<-matrix(nrow = 2, ncol = 3, byrow = T,
          c(1,0,0,
            0,1,0))
PAR <- c(xpar,ypar,upar,stdpar,Sigma1_par,Sigma1_paru,Sigma2_par,V0_paru,rho)
Kmopt <- optim(PAR,my.obj.7,method = "L-BFGS-B")
Xhat0 <- Kmopt$par[1:3]
V0 <- diag(c(exp(Kmopt$par[4]),exp(Kmopt$par[4]),exp(Kmopt$par[8])))
rho <- exp(Kmopt$par[9])
Sigma_1 <- matrix(c(exp(Kmopt$par[5])*exp(Kmopt$par[5]),
                    exp(Kmopt$par[5])*exp(Kmopt$par[5])*rho,0,
                    exp(Kmopt$par[5])*exp(Kmopt$par[5])*rho,
                    exp(Kmopt$par[5])*exp(Kmopt$par[5]),0,
                    0,0,exp(Kmopt$par[6])),nrow=3)
Sigma_2 <- diag(exp(Kmopt$par[7]),2)

```

```

kal <- kalman(Y=Y, A=A, C=C, Sigma_1=Sigma_1, Sigma_2=Sigma_2,
             Xhat0=Xhat0, V0=V0, n_pred=n_pred)
Loglike2(kal)
#-----sh
Reconstructions(kal,0)
#-----nh
Reconstructions(kal,1)

#sh
Pred<-kal$pred
Pred<-data.frame('x'=Pred[,1], 'y'=Pred[,2])
Pred_Var<-kal$Sigma_xx_pred
sd_pred<-sqrt(Pred_Var[1,1])
t95_minus_pred<-qnorm(0.025)*sd_pred
t95_plus_pred<-qnorm(0.975)*sd_pred
Pred_plus<-Pred$x+t95_plus_pred
Pred_minus<-Pred$x+t95_minus_pred
table <- matrix(cbind(Pred$x[c(171,181,191,201)],Pred_minus[c(171,181,191,201)],
                      Pred_plus[c(171,181,191,201)]),nrow=3,byrow=TRUE)
rownames(table) <- c("Predictions","Lower bounderies","Upper bounderies")
colnames(table) <- c("2020","2030","2040","2050")
table <- as.table(table)
table
#nh
sd_pred<-sqrt(Pred_Var[2,2])
t95_minus_pred<-qnorm(0.025)*sd_pred
t95_plus_pred<-qnorm(0.975)*sd_pred
Pred_plus<-Pred$y+t95_plus_pred
Pred_minus<-Pred$y+t95_minus_pred
table <- matrix(cbind(Pred$y[c(171,181,191,201)],Pred_minus[c(171,181,191,201)],
                      Pred_plus[c(171,181,191,201)]),nrow=3,byrow=TRUE)
rownames(table) <- c("Predictions","Lower bounderies","Upper bounderies")
colnames(table) <- c("2020","2030","2040","2050")
table <- as.table(table)
table

Rec<-kal$rec[1:169,]
Rec<-data.frame('x'=Rec[,1], 'y'=Rec[,2])
series=Rec
ar=c(1)
ma=c(0)
kvar=2 #number of variables
struct_1<-define.model(kvar=kvar,ar=ar,ma=ma)

difference=matrix(c(1,1,2,1),ncol=2)

dif.poly<- define.dif(series=series,difference=difference)
series_diff<-t(dif.poly$dif.series)

model<-marima(series_diff, ar.pattern=NULL, ma.pattern=NULL,
              Plot="none",Check=FALSE, penalty=0)

data_to_predict<-data.frame('x'=matrix(NA, nrow = n_pred, ncol = 1),

```

```

'y'=matrix(NA, nrow = n_pred, ncol = 1))

series_full_diff<-rbind(series_diff,data_to_predict)
series_full<-rbind(series,data_to_predict)
Forecast<-arma.forecast(series = series_full , marima = model, nstart=n,
                        nstep = n_pred, dif.poly=dif.poly$dif.poly)

x_f<-Forecast$forecasts[1,(n+1):(n+n_pred)]
y_f<-Forecast$forecasts[2,(n+1):(n+n_pred)]

Predictions(Forecast,0,kal)
Predictions(Forecast,1,kal)

#-----Functions
Reconstructions <- function(kal,x){
  #l'attribut rec de kal est X_rec
  Rec<-kal$rec
  Rec<-data.frame('x'=Rec[,1], 'y'=Rec[,2])
  Rec_Var<-kal$Sigma_xx_rec
  Pred<-kal$pred
  Pred<-data.frame('x'=Pred[,1], 'y'=Pred[,2])
  Pred_Var<-kal$Sigma_xx_pred

  if (x==0){
    #-----sh
    sd<-sqrt(Rec_Var[1,1,])
    t95_minus<-qnorm(0.025)*sd
    t95_plus<-qnorm(0.975)*sd

    Rec_plus<-Rec$x+t95_plus
    Rec_minus<-Rec$x+t95_minus

    sd_pred<-sqrt(Pred_Var[1,1,])
    t95_minus_pred<-qnorm(0.025)*sd_pred
    t95_plus_pred<-qnorm(0.975)*sd_pred

    Pred_plus<-Pred$x+t95_plus_pred
    Pred_minus<-Pred$x+t95_minus_pred

    plot(Year[1:169],Data$sh,type='l', col='orange',lwd=1.5, cex=0.85
         ,xlab='Time [Year]', cex.axis = 1,xlim=c(1850,2050),ylab='',
         main='Temperature anomalies in the Southern hemisphere',
         ylim=c(min(Rec_minus[1:169]),max(Rec_plus[1:169])))
    lines(Year[1:169],Rec_plus[1:169], lty=5, col= 'red', lwd=1.5)
    lines(Year[1:169],Rec_minus[1:169], lty=5, col= 'red', lwd=1.5)
    lines(Year[1:169],Rec$x[1:169],type = 'o', pch=1, lty=1, bty='l', col=blues[2])
    lines(Year[170:201],Pred$x[170:201], lty=1, col= 'purple', pch=1, lwd=1.5)
    lines(Year[170:201],Pred_plus[170:201], lty=1, col= 'red', lwd=1.5)
    lines(Year[170:201],Pred_minus[170:201], lty=1, col= 'red', lwd=1.5)
    legend('topleft',legend = c('Reconstruction','95% confidence interval
                                reconstruction'
                                , 'Southern hemisphere','Predictions','95% confidence
                                interval predictions'),

```



```

        col = c(blues[2], 'red', 'orange', 'purple', 'red'), lty = c(1, 5, 1, 1, 1),
        pch = c(1, NA, NA, NA, NA), cex=1, bty = 'n')
if (x==1){
  #-----sh
  sd<-sqrt(Rec_Var[2,2])
  t95_minus<-qnorm(0.025)*sd
  t95_plus<-qnorm(0.975)*sd

  Rec_plus<-Rec$y+t95_plus
  Rec_minus<-Rec$y+t95_minus

  sd_pred<-sqrt(Pred_Var[2,2])
  t95_minus_pred<-qnorm(0.025)*sd_pred
  t95_plus_pred<-qnorm(0.975)*sd_pred

  Pred_plus<-Pred$y+t95_plus_pred
  Pred_minus<-Pred$y+t95_minus_pred

  plot(Year[1:169], Data$nh, type='l', col='orange', lwd=1.5, cex=0.85
        , xlab='Time [Year]', cex.axis = 1, xlim=c(1850, 2050), ylab='',
        main='Temperature anomalies in the Northern hemisphere',
        ylim=c(min(Rec_minus[1:169]), max(Rec_plus[1:169])))
  lines(Year[1:169], Rec_plus[1:169], lty=5, col= 'red', lwd=1.5)
  lines(Year[1:169], Rec_minus[1:169], lty=5, col= 'red', lwd=1.5)
  lines(Year[1:169], Rec$y[1:169], type = 'o', pch=1, lty=1, bty='l', col=blues[2])
  lines(Year[170:201], Pred$y[170:201], lty=1, col= 'purple', pch=1, lwd=1.5)
  lines(Year[170:201], Pred_plus[170:201], lty=1, col= 'red', lwd=1.5)
  lines(Year[170:201], Pred_minus[170:201], lty=1, col= 'red', lwd=1.5)
  legend('topleft', legend = c('Reconstruction', '95% confidence interval
                              reconstruction'
                              , 'Northern hemisphere', 'Predictions', '95% confidence
                              interval predictions'),
        col = c(blues[2], 'red', 'orange', 'purple', 'red'), lty = c(1, 5, 1, 1, 1),
        pch = c(1, NA, NA, NA, NA), cex=1, bty = 'n')
}

Loglike <- function(kal){
  Y_tilde <- Y[-1,] - kal$pred[2:n,]
  R = kal$Sigma_yy_pred[, 2:n]

  P1 = matrix(NA, nrow = 168, ncol = 1)
  P2 = matrix(NA, nrow = 168, ncol = 1)

  for (i in 1:168){
    P1[i] = log(det(R[, i]))
    P2[i] = t(Y_tilde[i,]) %*% solve(R[, i]) %*% Y_tilde[i,]
  }
  return(-0.5 * sum(P1 + P2))}

Loglike2 <- function(kal){
  Y_tilde <- Y[-1,] - kal$pred[2:n, 1:2]
  R = kal$Sigma_yy_pred[, 2:n]

```

```

P1 = matrix(NA, nrow = 168, ncol = 1)
P2 = matrix(NA, nrow = 168, ncol = 1)

for (i in 1:168){
  P1[i] = log(det(R[,i]))
  P2[i] = t(Y_tilde[i,]) %*% solve(R[,i]) %*% Y_tilde[i,]
}
return(-0.5 * sum(P1 + P2))}

Predictions <- function(Forecast,x,kal){
  Rec<-kal$rec
  Rec<-data.frame('x'=Rec[,1], 'y'=Rec[,2])
  Rec_Var<-kal$Sigma_xx_rec
  Pred<-kal$pred
  Pred<-data.frame('x'=Pred[,1], 'y'=Pred[,2])
  Pred_Var<-kal$Sigma_xx_pred

  if (x==0){
    sdM<-sqrt(Forecast$pred.var[1,1])
    t95_minusM<-qnorm(0.025)*sdM
    t95_plusM<-qnorm(0.975)*sdM

    Pred_plusM<-x_f+t95_plusM
    Pred_minusM<-x_f+t95_minusM

    sd_pred<-sqrt(Pred_Var[1,1])
    t95_minus_pred<-qnorm(0.025)*sd_pred
    t95_plus_pred<-qnorm(0.975)*sd_pred

    Pred_plus<-Pred$x+t95_plus_pred
    Pred_minus<-Pred$x+t95_minus_pred

    plot(Year[1:169],Data$sh,xlab='Year', ylab='', main="Temperature anomalies -
      Southern hemisphere", type = 'o',pch=1,lty=5,bty='l',cex=0.5,col=blues[9],
      cex.axis =1, xlim=c(1850,2050),ylim=c(min(Pred_minus[-c(1)]),
      max(Pred_plus[-c(1)])))
    lines(Year[170:201],Pred$x[170:201], lty=1, col= 'purple', pch=1, lwd=1.5)
    lines(Year[170:201],Pred_plus[170:201], lty=5, col= 'purple', lwd=1.5)
    lines(Year[170:201],Pred_minus[170:201], lty=5, col= 'purple', lwd=1.5)
    lines(Year[170:201],Pred_plusM, lty=5, col= 'red', lwd=1.5)
    lines(Year[170:201],Pred_minusM, lty=5, col= 'red', lwd=1.5)
    lines(Year[170:201],x_f,col='red', lwd = 1.5)
    legend('topleft',legend = c('System','Predictions MARIMA', '95% confidence
      interval MARIMA','Predictions Kalman', '95%
      confidence interval Kalman'),
      col = c(blues[9], 'red', 'red', 'purple', 'purple'),lty = c(1,1,5,1,5),
      pch = c(NA,NA,NA,NA,NA),cex=0.7, bty = 'n')})
  if (x==1){
    sdM<-sqrt(Forecast$pred.var[2,2])
    t95_minusM<-qnorm(0.025)*sdM
    t95_plusM<-qnorm(0.975)*sdM
  }
}

```

```

Pred_plusM<-y_f+t95_plusM
Pred_minusM<-y_f+t95_minusM

sd_pred<-sqrt(Pred_Var[2,2])
t95_minus_pred<-qnorm(0.025)*sd_pred
t95_plus_pred<-qnorm(0.975)*sd_pred

Pred_plus<-Pred$y+t95_plus_pred
Pred_minus<-Pred$y+t95_minus_pred

plot(Year[1:169],Data$nh,xlab='Year', ylab='', main="Temperature anomalies -
      Northern hemisphere", type = 'o',pch=1,lty=5,bty='l',
      cex=0.5,col=blues[2], cex.axis =1,
      xlim=c(1850,2050),ylim=c(min(Pred_minus[-c(1)]),max(Pred_plus[-c(1)])))
lines(Year[170:201],Pred$y[170:201], lty=1, col= 'purple', pch=1, lwd=1.5)
lines(Year[170:201],Pred_plus[170:201], lty=5, col= 'purple', lwd=1.5)
lines(Year[170:201],Pred_minus[170:201], lty=5, col= 'purple', lwd=1.5)
lines(Year[170:201],Pred_plusM, lty=5, col= 'red', lwd=1.5)
lines(Year[170:201],Pred_minusM, lty=5, col= 'red', lwd=1.5)
lines(Year[170:201],y_f,col='red', lwd = 1.5)
legend('topleft',legend = c('System','Predictions MARIMA',
      '95% confidence interval MARIMA',
      'Predictions Kalman',
      '95% confidence interval Kalman'),
      col = c(blues[2],'red','red','purple','purple'),lty = c(1,1,5,1,5),
      pch = c(NA,NA,NA,NA,NA),cex=0.7,bty = 'n'))
}

my.obj.5 <- function(PAR){
  Kro <- kalman(Y=Y, A=A, C=C,
    Sigma_1=matrix(c(exp(PAR[4])*exp(PAR[4]),
      exp(PAR[4])*exp(PAR[4])*PAR[6],
      exp(PAR[4])*exp(PAR[4])*PAR[6],
      exp(PAR[4])*exp(PAR[4])),nrow=2),
    Sigma_2=diag(exp(PAR[5]),2),Xhat0=c(PAR[1:2]),
    V0=diag(exp(PAR[3]),2),n_pred=n_pred)
  Y_tilde <-Y[-1,]-Kro$pred[2:n,]
  R = Kro$Sigma_yy_pred[,2:n]

  P1 = matrix(NA, nrow = 168, ncol = 1)
  P2 = matrix(NA, nrow = 168, ncol = 1)

  for (i in 1:168){
    P1[i] = log(det(R[,i]))
    P2[i] = t(Y_tilde[i,]) %*% solve(R[,i]) %*% Y_tilde[i,]
  }
  return(0.5 * sum(P1 + P2))}

my.obj.7 <- function(PAR){
  Kro <- kalman(Y=Y, A=A, C=C,
    Sigma_1=matrix(c(exp(PAR[5])*exp(PAR[5]),
      exp(PAR[5])*exp(PAR[5])*exp(PAR[9]),0,
      exp(PAR[5])*exp(PAR[5])*exp(PAR[9]),

```

```

                                exp(PAR[5])*exp(PAR[5]),0,
                                0,0,exp(PAR[6])),nrow=3),

                                Sigma_2=diag(exp(PAR[7]),2),Xhat0=c(PAR[1:3]),
                                V0=diag(c(exp(PAR[4]),exp(PAR[4]),exp(PAR[8]))),n_pred=n_pred)

Y_tilde <-Y[-1,]-Kro$pred[2:n,1:2]
R = Kro$Sigma_yy_pred[,2:n]

P1 = matrix(NA, nrow = 168, ncol = 1)
P2 = matrix(NA, nrow = 168, ncol = 1)

for (i in 1:168){
  P1[i] = log(det(R[,i]))
  P2[i] = t(Y_tilde[i,]) %*% solve(R[,i]) %*% Y_tilde[i,]
}
return(0.5 * sum(P1 + P2))}

```