Estimating the Trajectory of a Satellite

Assignment 5 – 02417 Time Series Analysis – Anders Hørsted (s082382)

In this report the trajectory of an orbiting satellite will be reconstructed and predicted, based on 50 measurements $\{(r_t^m, \theta_t^m)\}_{t=1}^{50}$ of the position of the satellite given in polar coordinates. Due to imprecision in the measurement process, the real position of the satellite (r_t^p, θ_t^p) is related to the measurements by

$$\begin{aligned} r_t^m &= r_t^p + \varepsilon_{rt} \\ \theta_t^m &= \theta_t^p + \varepsilon_{\theta t} \end{aligned}$$

where $\varepsilon_{rt} \sim N(0, 2000^2)$, $\varepsilon_{\theta t} \sim N(0, 0.03^2)$ and ε_{rt}, $\varepsilon_{\theta t}$ are independent.

Formulating a model

To reconstruct and predict the trajectory of the satellite, we will create a state space model. The state vector is defined as

$$\boldsymbol{X}_t = \begin{pmatrix} r_t^p \\ \theta_t^p \\ v_{\theta t}^p \end{pmatrix}$$

where r_t^p is the true distance to the satellite, θ_t^p is the angle, and $v_{\theta t}^p$ is the angle velocity. It is assumed that the trajectory is well approximated by a circle, but to include deviations from a perfect circle, we define the random deviations

$$\varepsilon_{rt}^p \sim N(0, 500^2), \quad \varepsilon_{\theta t}^p \sim N(0, 0.005^2), \quad \varepsilon_{v_\theta t}^p \sim N(0, 0.005^2)$$

and then model the dynamics of the state vector, by the equations

$$\begin{aligned} r_t^p &= r_{t-1}^p + \varepsilon_{rt}^p \\ \theta_t^p &= \theta_{t-1}^p + v_{t-1}^p + \varepsilon_{\theta t}^p \\ v_t^p &= v_{t-1}^p + \varepsilon_{v_\theta t}^p \end{aligned}$$

Comment on model formulation!!! To write the state space model on matrix form, we define the measurement vector by

$$\boldsymbol{Y}_t = \begin{pmatrix} r_t^m \\ \theta_t^m \end{pmatrix}$$

and two random vectors by

$$\boldsymbol{e}_1 = \begin{pmatrix} \varepsilon^p_{rt} \\ \varepsilon^p_{\theta t} \\ \varepsilon^p_{v_\theta t} \end{pmatrix}, \quad \boldsymbol{e}_2 = \begin{pmatrix} \varepsilon_{rt} \\ \varepsilon_{\theta t} \end{pmatrix}$$

Since $\varepsilon^p_{rt}, \varepsilon^p_{\theta t}$ and $\varepsilon^p_{v_\theta t}$ are independent, and also ε_{rt} and $\varepsilon_{\theta t}$ are independent

$$\boldsymbol{\Sigma}_1 = \mathrm{Var}[\boldsymbol{e}_1] = \begin{bmatrix} 500^2 & 0 & 0 \\ 0 & 0.005^2 & 0 \\ 0 & 0 & 0.005^2 \end{bmatrix}, \quad \boldsymbol{\Sigma}_2 = \mathrm{Var}[\boldsymbol{e}_2] = \begin{bmatrix} 2000^2 & 0 \\ 0 & 0.03^2 \end{bmatrix}$$

By setting

$$\boldsymbol{A} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{bmatrix}, \quad \boldsymbol{C} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$$

the state space model can now be expressed by the system equation

$$\boldsymbol{X}_t = \boldsymbol{A}\boldsymbol{X}_{t-1} + \boldsymbol{e}_1$$

and the observation equation

$$\boldsymbol{Y}_t = \boldsymbol{C}\boldsymbol{X}_t + \boldsymbol{e}_2$$

Using the defined model the trajectory can now be reconstructed and predicted by using the Kalman filter. To actually use the Kalman filter, it needs to be implemented.

Implementing the Kalman filter

In this section the Kalman filter is implemented in R. The implementation (see listing 1) is a regular R script that first loads the data in the variable `dat`

```
source('loaddata.R')


n = length(dat[,1])

# Setup model matrices
A.dat = c(1,0,0,
          0,1,1,
          0,0,1)
A = matrix(A.dat, nrow=3, ncol=3, byrow=TRUE)
C.dat = c(1,0,0,
          0,1,0)
C = matrix(C.dat, nrow=2, ncol=3, byrow=TRUE)

# Init error covariance matrices
S.1 = diag(c(500^2, 0.005^2, 0.005^2))
S.2 = diag(c(2000^2, 0.03^2))

# Init history variables
```

```
X.hat.now.history = matrix(0, nrow=n-1, ncol=3)
K.t.history = array(0, dim=c(3,2,n-1))

# Initialize Kalman filter
X.hat.step = matrix(c(r.m[1], theta.m[1], 0))
S.xx.step = diag(c(0,0,0))
S.yy.step = S.2

# Run Kalman filter
for (i in 2:n) {
    Y.t = matrix(c(r.m[i], theta.m[i]))

    # Update Kalman gain
    K.t = S.xx.step %*% t(C) %*% solve(S.yy.step)

    # Reconstruction
    X.hat.now = X.hat.step + K.t %*% (Y.t - C %*% X.hat.step)
    S.xx.now = S.xx.step - K.t %*% C %*% S.xx.step

    # Prediction
    X.hat.step = A %*% X.hat.now
    S.xx.step = A %*% S.xx.now %*% t(A) + S.1
    S.yy.step = C %*% S.xx.step %*% t(C) + S.2

    # Save iteration history
    X.hat.now.history[i-1,] = X.hat.now
    K.t.history[,,i-1] = K.t
}
```

Code Listing 1: Implementation of the Kalman filter

Reconstructing the trajectory

In this section the implementation of the Kalman filter is used to reconstruct the trajectory of the satellite.

Predicting the trajectory

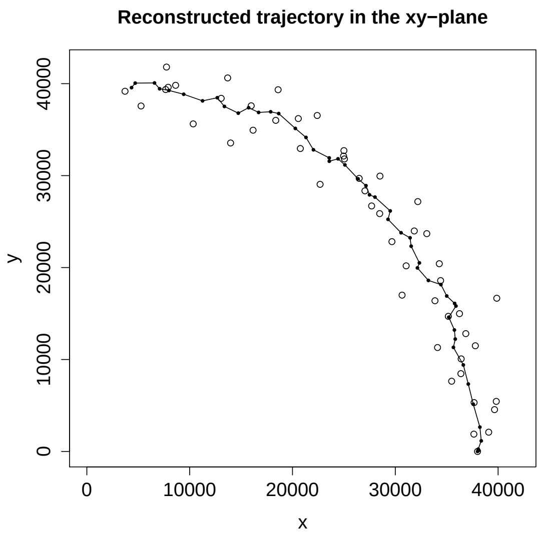

Figure 1: CAPTION XXX

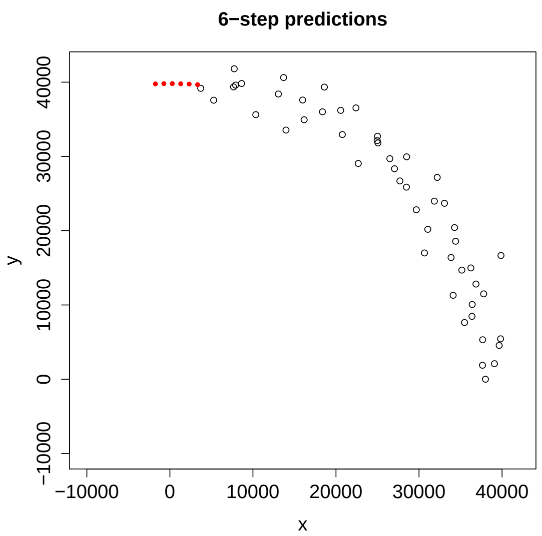

Figure 2: CAPTION XXX

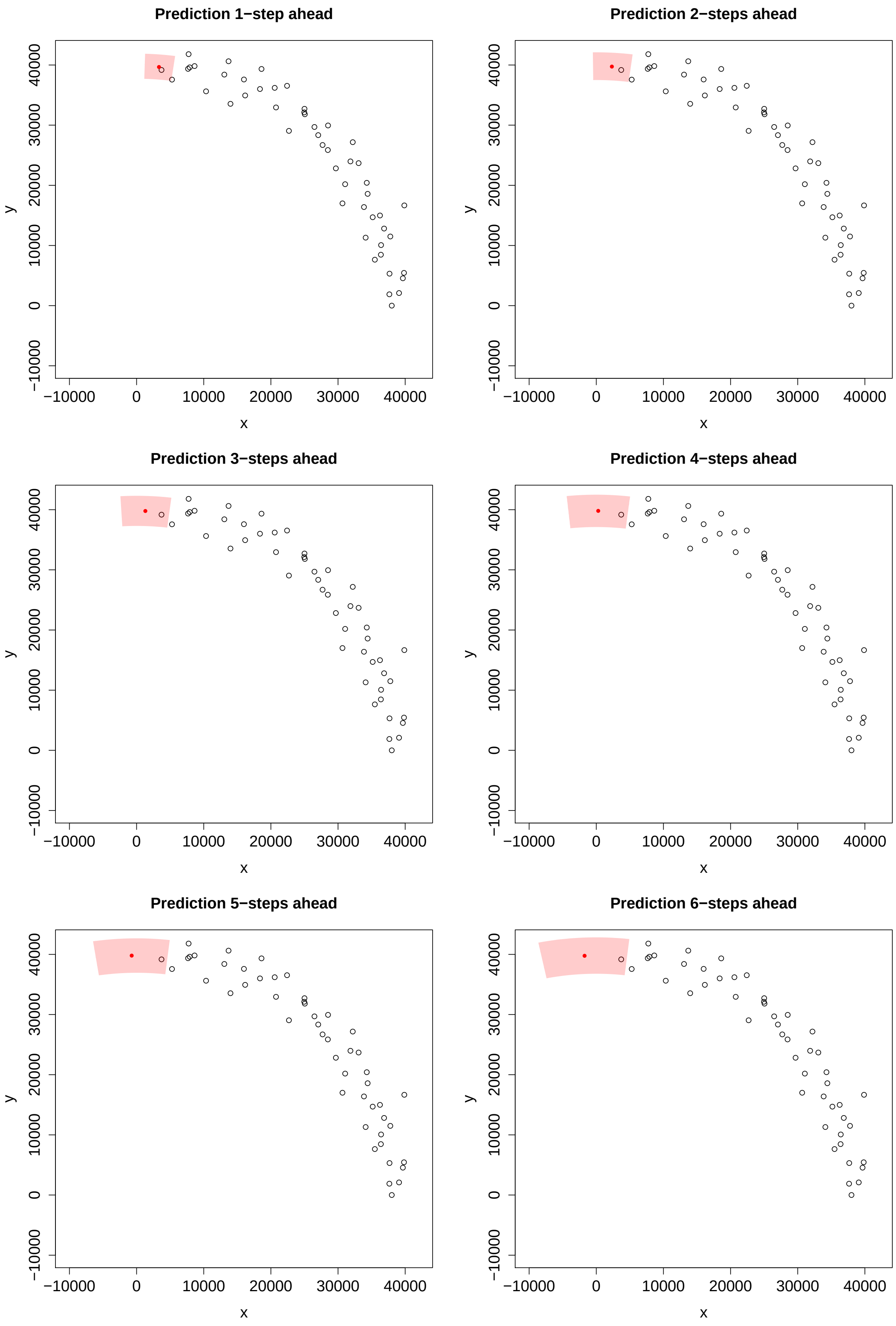

Figure 3: CAPTION

A Appendices

All R code used for this assignment is included here. All source code incl. latex code for this report can be found at `https://github.com/alphabits/dtu-fall-2011/tree/master/02417/assignment-5`

References

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