Assignment 4, part 1

Lilja Bjorg, Jón Vilberg, Daníel Bergmann

Saturday, November 01, 2014

```
sat.dat <- read.csv("Satelliteorbit.csv",header=FALSE,col.names=c("rm","theta_m","V3") )
sat.dat$rm[2] = sat.dat$theta_m[2]
sat.dat$theta_m[2] = sat.dat$V3[2]
y_rm = sat.dat$rm
y_theta= sat.dat$theta_m
y = matrix(c(y_rm,y_theta),nrow=2,ncol=50,byrow=T)
y  = cbind(y,matrix(NA,2,6))
y[2,51:56] = NA
sat.dat$V3 = NULL
```

#TASK1 - State space model

we have >>> x = t[r theta omega] we want >>> x' = A*x* +*B*u and y = Cx B*u is zero, because there is no input to the system all states are output, thus, C is an identity matrix, dimension 2x3

```
A = matrix(c(1,0,0, 0,1,1, 0,0,1),nrow=3,ncol=3,byrow=TRUE,
           dimnames = list(c("row1", "row2","row3"),
                           c("C.1", "C.2", "C.3")))
C = matrix(c(1,0,0, 0,1,0),nrow=2,ncol=3,byrow=TRUE)
```

Part 2

```
qr(cbind(t(C),t(C%*%A)))$rank
```

```
## [1] 3
```

```
sigma1 = matrix(c(500,0,0, 0, 0.005,0 , 0,0,0.005),nrow=3,ncol=3,byrow=TRUE) #var in diag
sigma2 = matrix(c(2000,0, 0,0.03),nrow=2,ncol=2,byrow=TRUE)
n = length(y[1,])
# Initialize variables for storing results
Kt.store <- matrix(NA,n,6)

Xhr <- Xhp <- matrix(NA,n,3)
Sxx.r <- Sxx.p <- matrix(NA,n,9) # 3x3 flett ut
Syy.p <- matrix(NA,nrow=n,ncol=4)            # 2x2 flett ut
```

Initiailzation

```
Xh.t.tm1 <- matrix(c(sat.dat$rm[1], sat.dat$theta_m[1],0),nrow=3,ncol=1,byrow=T)
Sxx.t.tm1 <- sigma1 #G * sigma1 * t(G)
Syy.t.tm1 <- C %*% Sxx.t.tm1 %*% t(C) +sigma2
```

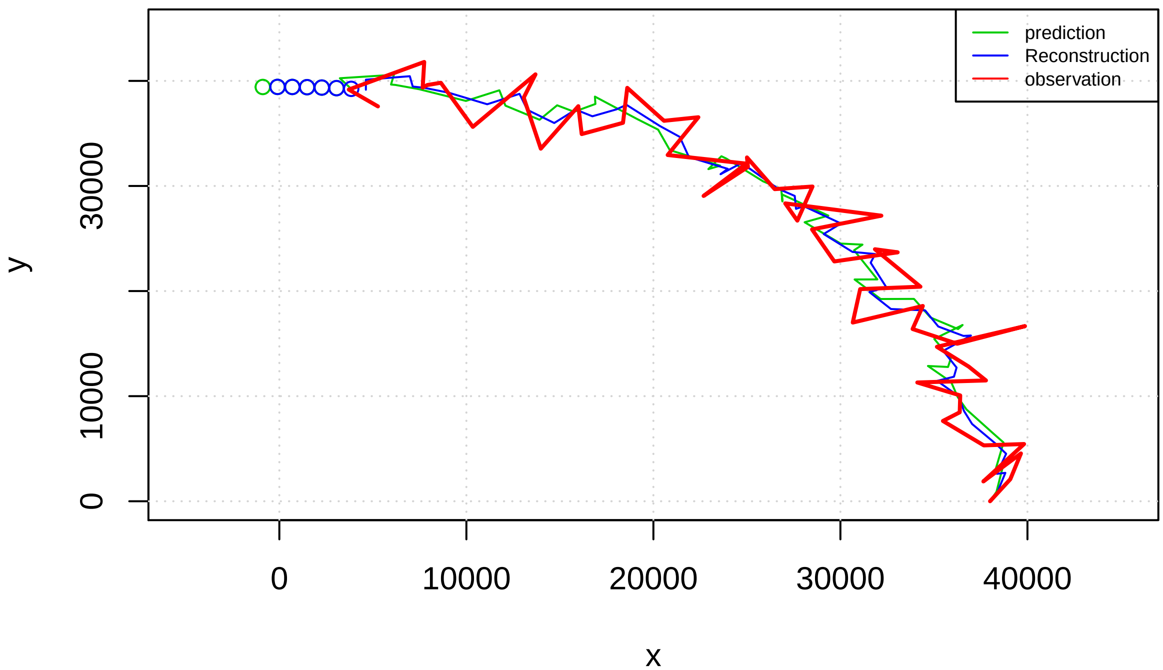
Reconstruction

```
Kt <- Sxx.t.tm1 %*% t(C) %*% solve(Syy.t.tm1)

Xh.t.t <- Xh.t.tm1 + Kt %*% (y[,1] - C%*%Xh.t.tm1)
Sxx.t.t = Sxx.t.tm1 - Kt %*% Syy.t.tm1 %*% t(Kt)
```

Prediction

```
Xh.tp1.t <- A %*% Xh.t.t
Sxx.tp1.t <- A %*% Sxx.t.t %*% t(A) + sigma1
Syy.tp1.t <- C %*% Sxx.tp1.t %*% t(C) + sigma2

# Store step result
Kt.store[1,] <- t(Kt)
Xhr[1,] <- t(Xh.t.t)
Sxx.r[1,] <- as.vector(Sxx.t.t)
Xhp[1,] <- t(Xh.tp1.t)
Sxx.p[1,] <- as.vector(Sxx.tp1.t)
Syy.p[1,] <- as.vector(Syy.tp1.t)

for(tt in 2:n){

  # Reconstruction
  if(any(is.na(y[,tt]))){  # if Y_t is missing
    #yy <- C%*%Xh.t.tm1
    Xh.t.t = Xh.tp1.t
    Sxx.t.t = Sxx.tp1.t
  }else{
    # Prepare for next iteration
    Xh.t.tm1 <- Xh.tp1.t
    Sxx.t.tm1 <- Sxx.tp1.t
    Syy.t.tm1 <- Syy.tp1.t
    yy <- y[,tt]
    Kt <- Sxx.t.tm1 %*% t(C) %*% solve(Syy.t.tm1)
    Xh.t.t <- Xh.t.tm1 + Kt %*% (yy - C%*%Xh.t.tm1)
    Sxx.t.t = Sxx.t.tm1 - Kt %*% Syy.t.tm1 %*% t(Kt)
  }

  #Prediction
  Xh.tp1.t <- A %*% Xh.t.t
  Sxx.tp1.t <- A %*% Sxx.t.t %*% t(A) + sigma1
  Syy.tp1.t <- C %*% Sxx.tp1.t %*% t(C) + sigma2

  # Store step result
  Kt.store[tt,] <- t(Kt)
  Xhr[tt,] <- t(Xh.t.t)
  Sxx.r[tt,] <- as.vector(Sxx.t.t)
  Xhp[tt,] <- t(Xh.tp1.t)
  Sxx.p[tt,] <- as.vector(Sxx.tp1.t)
```

```
  Syy.p[tt,] <- as.vector(Syy.tp1.t)

}

rm_p = Xhp[,1]
theta_p = Xhp[,2]
rm_r = Xhr[,1]
theta_r = Xhr[,2]

#plot xhat
plot(x=(rm_p[1:50]*cos(theta_p[1:50])),
     y= (rm_p[1:50] * sin(theta_p[1:50])),type='l',col=3,xlim= c(-5000,45000),ylim=c(0,45000),
     ylab = "y",
     xlab = "x")
grid()
lines(x=(rm_r[1:50]*cos(theta_r[1:50])),
     y= (rm_r[1:50] * sin(theta_r[1:50])),type='l',col=4)

lines(x=(rm_p[51:56]*cos(theta_p[51:56])),
     y= (rm_p[51:56] * sin(theta_p[51:56])),type='b',col=3)
lines(x=(rm_r[51:56]*cos(theta_r[51:56])),
      y= (rm_r[51:56] * sin(theta_r[51:56])),type='b',col=4)

lines(y = sat.dat$rm * sin(sat.dat$theta_m),
     x = sat.dat$rm * cos(sat.dat$theta_m),
     type='l',
     col=2,
     lwd=2
)
legend("topright",
       c("prediction","Reconstruction", "observation"),
       lty = 1,
       col=c(3,4, 2),
       cex=0.6)
```


prediction
Reconstruction
observation
y
30000
10000
0
0
10000
20000
30000
40000
x