Time Series Analysis

Lasse Engbo Christiansen

DTU Applied Mathematics and Computer Science
Technical University of Denmark

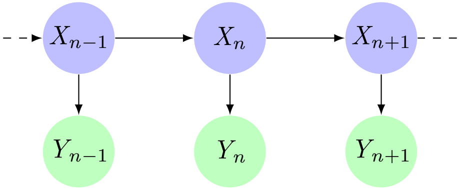
April 21, 2016

Outline of the lecture

State space models, 1st part:

- Model: Sec. 10.1
- The Kalman filter: Sec. 10.3
- An example on application of the Kalman filter.

State space models



- System model; A full description of the dynamical system (i.e. including the parameters):

$$X_t = f(X_{t-1}) + g(u_{t-1}) + e_{1,t}$$

- Observations; Noisy measurements of some parts (states) of the system:

$$Y_t = h(X_t) + e_{2,t}$$

- Goal; reconstruct and predict the state of the system

State space models; examples

- Estimate the temperature inside a solid block of material when we measure the temperature on the surface (with noise)
- Noisy measurements of the position of a ship; give a better estimate of the current position
- PK/PD-modeling: State: Amount of drug in blood, liver, muscles, ... Observations: Amount in blood (with noise), Input: Drug.

Determining the model structure

- The system model is often based on physical considerations; this often leads to dynamical models consisting of differential equations.
- An m'th order differential equation can be formulated as m 1st order differential equations.
- Sampling such a system leads to a discrete-time state space model.
- Note that the parameters may change in the discretization.
- We shall only consider linear state space models.

The linear stochastic state space model

$$\begin{aligned} \text{System equation:} \quad & \boldsymbol{X}_t = \boldsymbol{A}\boldsymbol{X}_{t-1} + \boldsymbol{B}\boldsymbol{u}_{t-1} + \boldsymbol{e}_{1,t} \\ \text{Observation equation:} \quad & \boldsymbol{Y}_t = \boldsymbol{C}\boldsymbol{X}_t + \boldsymbol{e}_{2,t} \end{aligned}$$

- $\boldsymbol{X}$: State vector
- $\boldsymbol{Y}$: Observation vector
- $\boldsymbol{u}$: Input vector
- $\boldsymbol{e}_1$: System noise
- $\boldsymbol{e}_2$: Observation noise

- $\dim(X_t) = m$ is called the order of the system
- $\{\boldsymbol{e}_{1,t}\}$ and $\{\boldsymbol{e}_{2,t}\}$ mutually independent white noise
- $V[\boldsymbol{e}_1] = \boldsymbol{\Sigma}_1$, $V[\boldsymbol{e}_2] = \boldsymbol{\Sigma}_2$
- $\boldsymbol{A}$, $\boldsymbol{B}$, $\boldsymbol{C}$, $\boldsymbol{\Sigma}_1$, and $\boldsymbol{\Sigma}_2$ are **known** matrices

- The state vector contains all information available for future evaluation; the state vector is a *Markov process*.
- It is possible to handle time-varying systems as well.

Example Air pollution in cities, NO and NO_2

$$\begin{bmatrix} X_{1,t} \\ X_{2,t} \end{bmatrix} = \begin{bmatrix} 0.9 & -0.1 \\ 0.4 & 0.8 \end{bmatrix} \begin{bmatrix} X_{1,t-1} \\ X_{2,t-1} \end{bmatrix} + \begin{bmatrix} \xi_{1,t} \\ \xi_{2,t} \end{bmatrix}, \quad \boldsymbol{\Sigma}_\xi = \begin{bmatrix} 30 & 21 \\ 21 & 23 \end{bmatrix}$$

Suppose that the NO component is missing. How could we formulate this as a state-space model?

States:

$$\boldsymbol{X}_t = \begin{pmatrix} X_t^{NO} - \mu_{NO} \\ X_t^{NO_2} - \mu_{NO_2} \end{pmatrix}$$

Parameters:

$$A = \begin{pmatrix} 0.9 & -0.1 \\ 0.4 & 0.8 \end{pmatrix}, B = 0\,, V(\boldsymbol{e}_1) = \begin{pmatrix} 30 & 21 \\ 21 & 23 \end{pmatrix}$$

$$C = \begin{pmatrix} 0 & 1 \end{pmatrix}\,, V(e_2) = 0$$

Example – a falling body I

- Height above ground: $z(t)$
- Initial conditions: Position $z(t_0)$ and velocity $z'(t_0)$
- Physical considerations: $\frac{d^2z}{dt^2} = -g$
- States: Position $x_1(t) = z(t)$ and velocity $x_2(t) = z'(t)$
- Only the position is measured $y(t) = x_1(t)$
- Continuous time description $\boldsymbol{x}(t) = [x_1(t)\ x_2(t)]^T$:

$$\boldsymbol{x}'(t) = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \boldsymbol{x}(t) + \begin{bmatrix} 0 \\ -1 \end{bmatrix} g$$

$$\boldsymbol{y}(t) = \begin{bmatrix} 1 & 0 \end{bmatrix} \boldsymbol{x}(t)$$

Example – a falling body II

- Solving the equations:

$$x_1(t) = -\frac{g}{2}(t - t_0)^2 + (t - t_0)x_2(t_0) + x_1(t_0)$$
$$x_2(t) = -g(t - t_0) + x_2(t_0)$$

- Sampling: $t = kT$, $t_0 = (k-1)T$, and $T = 1$

$$\boldsymbol{x}_k = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \boldsymbol{x}_{k-1} + \begin{bmatrix} -1/2 \\ -1 \end{bmatrix} g$$
$$\boldsymbol{y}_k = \begin{bmatrix} 1 & 0 \end{bmatrix} \boldsymbol{x}_k$$

- Adding disturbances and measurement noise:

$$\boldsymbol{x}_k = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \boldsymbol{x}_{k-1} + \begin{bmatrix} -1/2 \\ -1 \end{bmatrix} g + \boldsymbol{e}_{1,k}$$
$$\boldsymbol{y}_k = \begin{bmatrix} 1 & 0 \end{bmatrix} \boldsymbol{x}_k + e_{2,k}$$

Example – a falling body III

Given measurements of the position at time points $1, 2, \ldots, k$ we could:

- **Predict** the future position and velocity $\boldsymbol{x}_{k+n|k}$ $(n > 0)$
- **Reconstruct** the current position and velocity from noisy measurements $\boldsymbol{x}_{k|k}$
- **Interpolate or smoothen** to find the best estimate of the position and velocity at a previous time point $\boldsymbol{x}_{k+n|k}$ $(n < 0)$ (estimate the path in the state space

We will focus on reconstruction and prediction

Requirement – observability

In order to predict, reconstruct or interpolate the m-dimensional state in the system

$$\begin{aligned} \boldsymbol{X}_t &= \boldsymbol{A}\boldsymbol{X}_{t-1} + \boldsymbol{B}\boldsymbol{u}_{t-1} + \boldsymbol{e}_{1,t} \\ \boldsymbol{Y}_t &= \boldsymbol{C}\boldsymbol{X}_t + \boldsymbol{e}_{2,t} \end{aligned}$$

the system must be observable, i.e.

$$\text{rank}\left[\boldsymbol{C}^T \vdots (\boldsymbol{C}\boldsymbol{A})^T \vdots \cdots \vdots \left(\boldsymbol{C}\boldsymbol{A}^{m-1}\right)^T\right] = m.$$

For the falling body (from the discrete-time description of the system):

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

$$\left[C^T \vdots (CA)^T\right] = \left[\begin{matrix} 1 \\ 0 \end{matrix} \vdots \left(\begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}\right)^T\right] = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}$$

```
> qr( cbind(t(C), t(C %*% A)) )$rank
[1] 2
```

The Kalman filter

Initialization:

$$\widehat{\boldsymbol{X}}_{1|0} = E\left[\boldsymbol{X}_1\right] = \boldsymbol{\mu}_0$$
$$\boldsymbol{\Sigma}^{xx}_{1|0} = V\left[\boldsymbol{X}_1\right] = \boldsymbol{V}_0$$
$$\boldsymbol{\Sigma}^{yy}_{1|0} = \boldsymbol{C}\boldsymbol{\Sigma}^{xx}_{1|0}\boldsymbol{C}^T + \boldsymbol{\Sigma}_2$$

For: $t = 1, 2, 3, \ldots$

Reconstruction:

$$\boxed{\begin{aligned}
\boldsymbol{K}_t &= \boldsymbol{\Sigma}^{xx}_{t|t-1}\boldsymbol{C}^T\left(\boldsymbol{\Sigma}^{yy}_{t|t-1}\right)^{-1} \\
\widehat{\boldsymbol{X}}_{t|t} &= \widehat{\boldsymbol{X}}_{t|t-1} + \boldsymbol{K}_t\left(\boldsymbol{Y}_t - \boldsymbol{C}\widehat{\boldsymbol{X}}_{t|t-1}\right) \\
\boldsymbol{\Sigma}^{xx}_{t|t} &= \boldsymbol{\Sigma}^{xx}_{t|t-1} - \boldsymbol{K}_t\boldsymbol{\Sigma}^{yy}_{t|t-1}\boldsymbol{K}_t^T
\end{aligned}}$$

Prediction:

$$\boxed{\begin{aligned}
\widehat{\boldsymbol{X}}_{t+1|t} &= \boldsymbol{A}\widehat{\boldsymbol{X}}_{t|t} + \boldsymbol{B}\boldsymbol{u}_t \\
\boldsymbol{\Sigma}^{xx}_{t+1|t} &= \boldsymbol{A}\boldsymbol{\Sigma}^{xx}_{t|t}\boldsymbol{A}^T + \boldsymbol{\Sigma}_1 \\
\boldsymbol{\Sigma}^{yy}_{t+1|t} &= \boldsymbol{C}\boldsymbol{\Sigma}^{xx}_{t+1|t}\boldsymbol{C}^T + \boldsymbol{\Sigma}_2
\end{aligned}}$$

Multi-step predictions

- Not part of the Kalman filter as stated above
- Can be calculated recursively for a given t starting with $k = 1$ for which $\widehat{\boldsymbol{X}}_{t+k|t}$ and $\boldsymbol{\Sigma}_{t+k|t}$ are calculated in the Kalman prediction step.

$$\widehat{\boldsymbol{X}}_{t+k+1|t} = \boldsymbol{A}\widehat{\boldsymbol{X}}_{t+k|t} + \boldsymbol{B}\boldsymbol{u}_{t+k}$$
$$\boldsymbol{\Sigma}^{xx}_{t+k+1|t} = \boldsymbol{A}\boldsymbol{\Sigma}^{xx}_{t+k|t}\boldsymbol{A}^T + \boldsymbol{\Sigma}_1$$

- The future input must be known/assumed.

Naming and history

- The filter is named after Rudolf E. Kalman, though Thorvald Nicolai Thiele and Peter Swerling actually developed a similar algorithm earlier.
- It was during a visit of Kalman to the NASA Ames Research Center that he saw the applicability of his ideas to the problem of trajectory estimation for the Apollo program, leading to its incorporation in the Apollo navigation computer.

From `http://en.wikipedia.org/wiki/Kalman_filter`

The Foundation of the Kalman filter

- Theorem 2.6 (Linear projection)
- The theorem is concerned with the random vectors $\boldsymbol{X}$ and $\boldsymbol{Y}$ for which the means, variances and covariances are used
- The state is called $\boldsymbol{X}_t$ and the observation is called $\boldsymbol{Y}_t$ and we could write down the theorem for these
- We have additional information; $\mathcal{Y}_{t-1}^T = (\boldsymbol{Y}_1^T, \ldots, \boldsymbol{Y}_{t-1}^T)$
- We include this information by considering the random vectors $\boldsymbol{X}_t|\mathcal{Y}_{t-1}$ and $\boldsymbol{Y}_t|\mathcal{Y}_{t-1}$ instead

$$E[(\boldsymbol{X}_t|\mathcal{Y}_{t-1})\,|\,(\boldsymbol{Y}_t|\mathcal{Y}_{t-1})] = E[\boldsymbol{X}_t|\,\boldsymbol{Y}_t, \mathcal{Y}_{t-1}] = \\ E[\boldsymbol{X}_t|\mathcal{Y}_{t-1}] + \mathrm{Cov}[\boldsymbol{X}_t,\, \boldsymbol{Y}_t|\mathcal{Y}_{t-1}]\,V^{-1}[\boldsymbol{Y}_t|\mathcal{Y}_{t-1}](\boldsymbol{Y}_t - E[\boldsymbol{Y}_t|\mathcal{Y}_{t-1}])$$

$$V[(\boldsymbol{X}_t|\mathcal{Y}_{t-1})\,|\,(\boldsymbol{Y}_t|\mathcal{Y}_{t-1})] = V[\boldsymbol{X}_t|\,\boldsymbol{Y}_t, \mathcal{Y}_{t-1}] = \\ V[\boldsymbol{X}_t|\mathcal{Y}_{t-1}] - \mathrm{Cov}[\boldsymbol{X}_t,\, \boldsymbol{Y}_t|\mathcal{Y}_{t-1}]\,V^{-1}[\boldsymbol{Y}_t|\mathcal{Y}_{t-1}]\mathrm{Cov}^T[\boldsymbol{X}_t,\, \boldsymbol{Y}_t|\mathcal{Y}_{t-1}]$$

The Foundation of the Kalman filter II

$$
\begin{aligned}
&E[\boldsymbol{X}_t | \boldsymbol{Y}_t, \mathcal{Y}_{t-1}] = \\
&\quad E[\boldsymbol{X}_t | \mathcal{Y}_{t-1}] + \mathrm{Cov}[\boldsymbol{X}_t, \boldsymbol{Y}_t | \mathcal{Y}_{t-1}] V^{-1}[\boldsymbol{Y}_t | \mathcal{Y}_{t-1}](\boldsymbol{Y}_t - E[\boldsymbol{Y}_t | \mathcal{Y}_{t-1}]) \\
&V[\boldsymbol{X}_t | \boldsymbol{Y}_t, \mathcal{Y}_{t-1}] = \\
&\quad V[\boldsymbol{X}_t | \mathcal{Y}_{t-1}] - \mathrm{Cov}[\boldsymbol{X}_t, \boldsymbol{Y}_t | \mathcal{Y}_{t-1}] V^{-1}[\boldsymbol{Y}_t | \mathcal{Y}_{t-1}] \mathrm{Cov}^T[\boldsymbol{X}_t, \boldsymbol{Y}_t | \mathcal{Y}_{t-1}]
\end{aligned}
$$

Using definitions from previous slide the update equations are:

$$
\begin{aligned}
\widehat{\boldsymbol{X}}_{t|t} &= \widehat{\boldsymbol{X}}_{t|t-1} + \boldsymbol{\Sigma}^{xy}_{t|t-1} \left(\boldsymbol{\Sigma}^{yy}_{t|t-1}\right)^{-1} \left(\boldsymbol{Y}_t - \widehat{\boldsymbol{Y}}_{t|t-1}\right) \\
\boldsymbol{\Sigma}^{xx}_{t|t} &= \boldsymbol{\Sigma}^{xx}_{t|t-1} - \boldsymbol{\Sigma}^{xy}_{t|t-1} \left(\boldsymbol{\Sigma}^{yy}_{t|t-1}\right)^{-1} \left(\boldsymbol{\Sigma}^{xy}_{t|t-1}\right)^{T}
\end{aligned}
$$

$$
\boldsymbol{K}_t = \boldsymbol{\Sigma}^{xy}_{t|t-1} \left(\boldsymbol{\Sigma}^{yy}_{t|t-1}\right)^{-1}
$$

$\boldsymbol{K}_t$ is called the *Kalman gain*, because it determines how much the 1-step prediction error influence the update of the state estimate

The Foundation of the Kalman filter III

The 1-step predictions are obtained directly from the state space model:

$$\widehat{\boldsymbol{X}}_{t+1|t} = \boldsymbol{A}\widehat{\boldsymbol{X}}_{t|t} + \boldsymbol{B}\boldsymbol{u}_t$$
$$\widehat{\boldsymbol{Y}}_{t+1|t} = \boldsymbol{C}\widehat{\boldsymbol{X}}_{t+1|t}$$

Which results in the prediction errors:

$$\widetilde{\boldsymbol{X}}_{t+1|t} = \boldsymbol{X}_{t+1} - \widehat{\boldsymbol{X}}_{t+1|t} = \boldsymbol{A}\widetilde{\boldsymbol{X}}_{t|t} + \boldsymbol{e}_{1,t+1}$$
$$\widetilde{\boldsymbol{Y}}_{t+1|t} = \boldsymbol{Y}_{t+1} - \widehat{\boldsymbol{Y}}_{t+1|t} = \boldsymbol{C}\widetilde{\boldsymbol{X}}_{t+1|t} + \boldsymbol{e}_{2,t+1}$$

And therefore:

$$\boldsymbol{\Sigma}^{xx}_{t+1|t} = \boldsymbol{A}\boldsymbol{\Sigma}^{xx}_{t|t}\boldsymbol{A}^T + \boldsymbol{\Sigma}_1$$
$$\boldsymbol{\Sigma}^{yy}_{t+1|t} = \boldsymbol{C}\boldsymbol{\Sigma}^{xx}_{t+1|t}\boldsymbol{C}^T + \boldsymbol{\Sigma}_2$$
$$\boldsymbol{\Sigma}^{xy}_{t+1|t} = \boldsymbol{\Sigma}^{xx}_{t+1|t}\boldsymbol{C}^T$$

Example: The falling body revised

Description of the system:

$$\boldsymbol{A} = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \qquad \boldsymbol{B} = \begin{bmatrix} -1/2 \\ -1 \end{bmatrix} \qquad \boldsymbol{C} = \begin{bmatrix} 1 & 0 \end{bmatrix}$$

$$\boldsymbol{\Sigma}_1 = \begin{bmatrix} 2.0 & 0.8 \\ 0.8 & 1.0 \end{bmatrix} \qquad \boldsymbol{\Sigma}_2 = \begin{bmatrix} 10000 \end{bmatrix}$$

Initialization: Released 10000 m above ground at 0 m/s

$$\widehat{\boldsymbol{X}}_{1|0} = \begin{bmatrix} 10000 \\ 0 \end{bmatrix} \quad \boldsymbol{\Sigma}^{xx}_{1|0} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \quad \boldsymbol{\Sigma}^{yy}_{1|0} = \begin{bmatrix} 10000 \end{bmatrix}$$

Simulation of a falling body – initialization

```r
z0 <- 10000
A <- matrix(c(1,0,1,1),nrow=2)
B <- matrix(c(-.5,-1),nrow=2)
C <- matrix(c(1,0),nrow=1)
Sigma1 <- matrix(c(2,.8,.8,1),nrow=2)
Sigma2 <- matrix(10000)
g <- 9.82; N <- 300
X <- matrix(nrow=2,ncol=N) ## Allocating space
X[,1] <- c(z0,0)
Y <- numeric(N)
Y[1] <- C%*%X[,1]+sqrt(Sigma2) %*% rnorm(1)
```

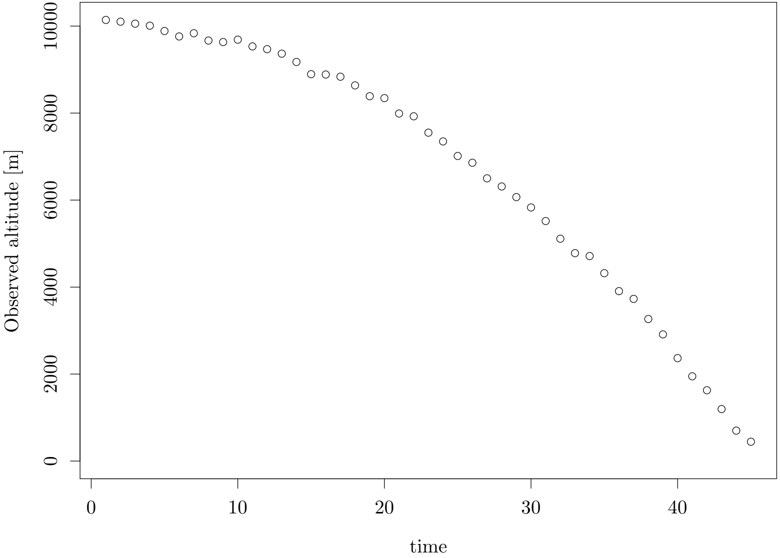
Simulation of a falling body - simulation

```
for (I in 2:N){
    X[,I] <- A %*% X[,I-1,drop=FALSE] + B%*%g +
        chol(Sigma1) %*% matrix(rnorm(2),ncol=1)
    Y[I] <- C %*% X[,I] + sqrt(Sigma2) %*% rnorm(1)
}
Nhit <- min(which(X[1,]<0))-1
X <- X[,1:Nhit]
Y <- Y[1:Nhit]
```

Why the Cholesky factorization?

- Remember that if $Z \sim N(0, I)$, then $Y = QZ \sim N(0, QQ^T)$.
- The Cholesky factorization is one way to solve $QQ^T = \Sigma$ for Q.

The falling body – observations



Kalman filter applied to a falling body II

1st observation $(t = 1)$: $y_1 = 10171$

Reconstruction: $K_1 = \left[\begin{array}{cc} 0 & 0 \end{array} \right]^T$

$$\widehat{\boldsymbol{X}}_{1|1} = \left[\begin{array}{c} 10000 \\ 0 \end{array} \right] \qquad \boldsymbol{\Sigma}^{xx}_{1|1} = \left[\begin{array}{cc} 0 & 0 \\ 0 & 0 \end{array} \right]$$

Prediction:

$$\widehat{\boldsymbol{X}}_{2|1} = \left[\begin{array}{c} 9995.09 \\ -9.82 \end{array} \right] \qquad \boldsymbol{\Sigma}^{xx}_{2|1} = \left[\begin{array}{cc} 2 & 0.8 \\ 0.8 & 1 \end{array} \right] \qquad \boldsymbol{\Sigma}^{yy}_{2|1} = \left[\begin{array}{c} 10002 \end{array} \right]$$

Kalman filter applied to a falling body III

2nd observation ($t = 2$): $y_2 = 10046$

Reconstruction: $K_2 = \left[\begin{array}{cc} 0.00020 & 0.00008 \end{array} \right]^T$

$$\widehat{\boldsymbol{X}}_{2|2} = \begin{bmatrix} 9995.1 \\ -9.81 \end{bmatrix} \qquad \boldsymbol{\Sigma}^{xx}_{2|2} = \begin{bmatrix} 2 & 0.8 \\ 0.8 & 1 \end{bmatrix}$$

Prediction:

$$\widehat{\boldsymbol{X}}_{3|2} = \begin{bmatrix} 9980.38 \\ -19.63 \end{bmatrix} \qquad \boldsymbol{\Sigma}^{xx}_{3|2} = \begin{bmatrix} 6.6 & 2.6 \\ 2.6 & 2 \end{bmatrix} \qquad \boldsymbol{\Sigma}^{yy}_{3|2} = \left[\begin{array}{c} 10006.6 \end{array} \right]$$

Kalman filter applied to a falling body IV

3rd observation ($t = 3$): $y_3 = 10082$

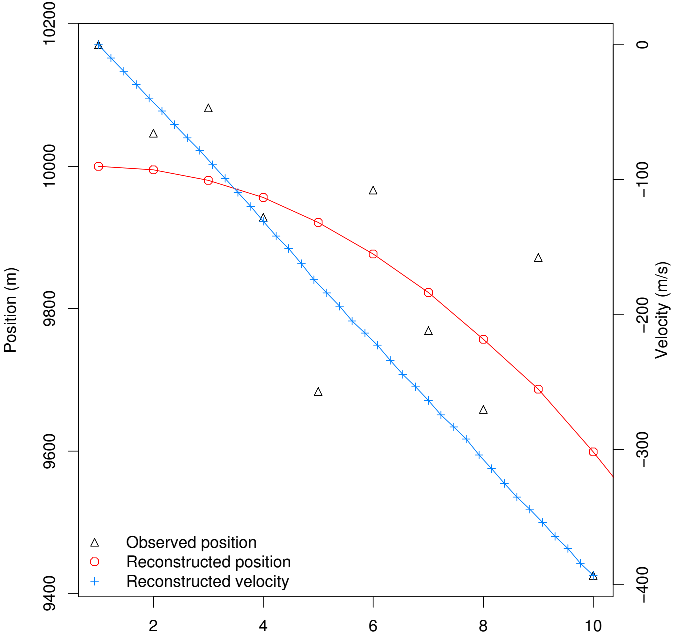
Reconstruction: $K_3 = \left[\begin{array}{cc} 0.00066 & 0.00026 \end{array}\right]^T$

$$\widehat{\boldsymbol{X}}_{3|3} = \left[\begin{array}{c} 9980.45 \\ -19.6 \end{array}\right] \qquad \boldsymbol{\Sigma}^{xx}_{3|3} = \left[\begin{array}{cc} 6.59 & 2.6 \\ 2.6 & 2 \end{array}\right]$$

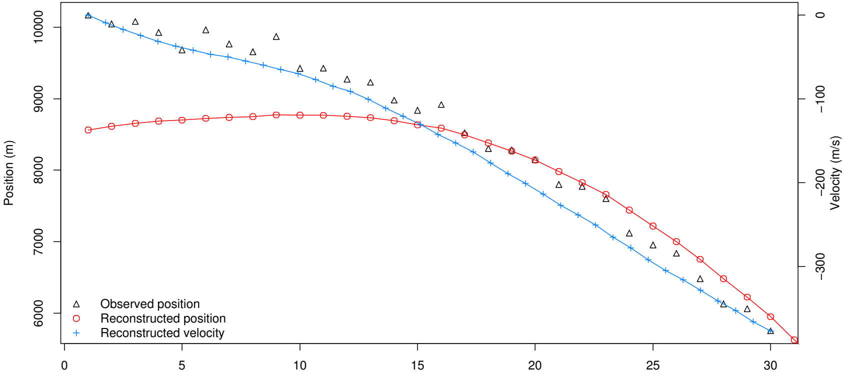
Prediction:

$$\widehat{\boldsymbol{X}}_{4|3} = \left[\begin{array}{c} 9955.94 \\ -29.41 \end{array}\right] \qquad \boldsymbol{\Sigma}^{xx}_{4|3} = \left[\begin{array}{cc} 15.79 & 5.4 \\ 5.4 & 3 \end{array}\right] \qquad \boldsymbol{\Sigma}^{yy}_{4|3} = \left[\begin{array}{c} 10015.79 \end{array}\right]$$

Falling body – the 10 first time points



Falling body – wrong initial state



Highlights

- State space model:

$$\begin{aligned} \text{System equation:} \quad & \boldsymbol{X}_t = \boldsymbol{A}\boldsymbol{X}_{t-1} + \boldsymbol{B}\boldsymbol{u}_{t-1} + \boldsymbol{e}_{1,t} \\ \text{Observation equation:} \quad & \boldsymbol{Y}_t = \boldsymbol{C}\boldsymbol{X}_t + \boldsymbol{e}_{2,t} \end{aligned}$$

- Sampling
- Observability

$$\text{rank}\left[\boldsymbol{C}^T \vdots (\boldsymbol{C}\boldsymbol{A})^T \vdots \cdots \vdots \left(\boldsymbol{C}\boldsymbol{A}^{m-1}\right)^T\right] = m.$$

- Kalman filter
  - Reconstruction
  - Prediction