13 - The Kalman filter

- Linear dynamical systems with sensor noise
- Estimating the initial state
- Lyapunov recursion
- Estimating the current state
- Kalman filter problem formulation
- Measurement update
- Time update
- Kalman filter recursion
- Riccati recursion
- Comparison with LQR
- Observer form
- Steady-state Kalman filter
- Example: mass-spring system

Markov Chains

We would like to model dynamical systems which are *driven* by randomness. For example

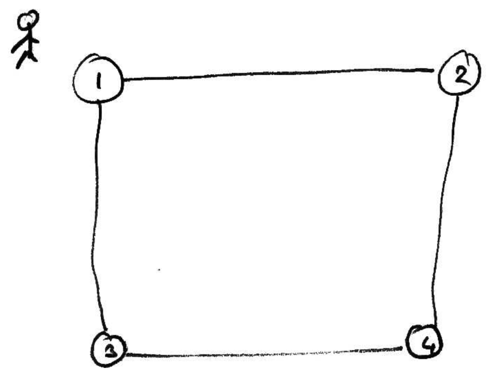
$$x(t+1) = Ax(t) + Bu(t) + w(t)$$

where

- $u(t)$ is a control input,
- $w(t)$ is a *disturbance*; often $w(0), w(1), \dots$ is an IID sequence of random variables

Markov Chains

Another example is a person randomly wandering around a town with four street corners



- At each time t he stands at corner $x(t) \in \{1,2,3,4\}$.
- Each time t he flips a coin; if it is heads he moves clockwise, and if it is tails he moves anticlockwise.
- So one representation is $x(t+1) = \big((x(t)+w(t)) \bmod 4\big) + 1$ where $w(t)$ is IID Bernoulli on $\{-1, 1\}$.
- Another is via the transition matrix given $\mathbf{Prob}(x(t+1) = j \mid x(t) = i)$.

The Joint Distribution

A sequence of random variables $x_0, x_1, x_2, \dots$ is called a *stochastic process*.

To model it, we need to know the joint pmf for any *initial subsequence*; i.e. for every t we want to know

$$J_t(z_0, z_1, \dots, z_t) = \mathbf{Prob}(x_0 = z_0, \dots, x_t = z_t)$$

The Chain Rule

From the definition of conditional pmf, we have

$$\begin{aligned}&\mathbf{Prob}(x_t = z_t, z_{t-1} = z_{t-1}, \dots, x_0 = z_0) \\ &\qquad = \mathbf{Prob}(x_t = z_t \,|\, x_{t-1} = z_{t-1}, \dots, x_0 = z_0)\, \mathbf{Prob}(x_{t-1} = z_{t-1}, \dots, x_0 = z_0)\end{aligned}$$

Temporarily abbreviating $x_t = z_t$ to just x_t, we can repeatedly apply this to give

$$\begin{aligned}&\mathbf{Prob}(x_t, x_{t-1}, \dots, x_0) \\ &\qquad = \mathbf{Prob}(x_t \,|\, x_{t-1}, \dots, x_0)\, \mathbf{Prob}(x_{t-1} \,|\, x_{t-2}, \dots, x_0) \dots \mathbf{Prob}(x_1 \,|\, x_0)\, \mathbf{Prob}(x_0)\end{aligned}$$

- Called the *chain rule* or *innovations decomposition*

Markov Process

A stochastic process is called *Markov* if

$$\boxed{\mathbf{Prob}(x_t = z_t \,|\, x_{t-1} = z_{t-1}, \dots, x_0 = z_0) = \mathbf{Prob}(x_t = z_t \,|\, x_{t-1} = z_{t-1})}$$

Call this *transition matrix* P_t, so that

$$P_t(a, b) = \mathbf{Prob}(x_t = a \,|\, x_{t-1} = b)$$

Using the chain rule, we have that the joint distribution is

$$\begin{aligned}&\mathbf{Prob}(x_0 = z_0, \dots, x_t = z_t)\\ &\qquad = P_t(z_t, z_{t-1})P_{t-1}(z_{t-1}, z_{t-2})P_{t-2}(z_{t-2}, z_{t-3}) \dots P_1(z_1, z_0)\,\mathbf{Prob}(x_0 = z_0)\end{aligned}$$

We have a *factorization* of the joint pdf; just as for recursive estimation, factorization makes inference easier.

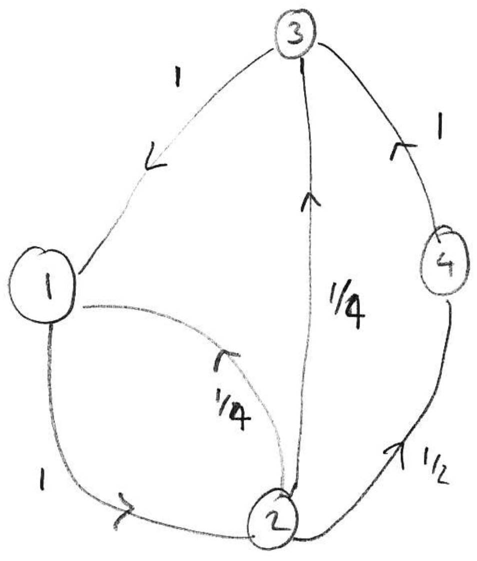
Markov Process

- A Markov process is often called a *Markov Chain*
- If P_t does not depend on t it is called *homogeneous* or *time-invariant*
- For $x_t : \Omega \rightarrow \{1, 2, \ldots, n\}$ the most common convention is to use a matrix whose rows sum to 1, as in

$$P_{ij} = P(j, i) = \mathbf{Prob}(x(t) = j \,|\, x(t-1) = i)$$

  It's also reasonable to use the transpose.
- Often represented by graphs, from which we can read the transition matrix.

$$P = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 1/4 & 0 & 1/4 & 1/2 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$



Marginals

For a Markov chain, we have

$$\begin{aligned}\mathbf{Prob}(x_{t+1} = j) &= \sum_i \mathbf{Prob}(x_{t+1} = j \mid x_t = i)\,\mathbf{Prob}(x_t = i) \\ &= \sum_{i=1}^n P_{ij}\,\mathbf{Prob}(x_t = i)\end{aligned}$$

So let $v(t) \in \mathbb{R}^n$ be the *distribution* of x_t, with

$$v(t)_i = \mathbf{Prob}(x_t = i)$$

then

$$\boxed{v(t+1) = P^T v(t)}$$

Dynamical Systems with Measurements

We are interested in systems of the form

$$\begin{aligned} x_{t+1} &= f(x_t, w_t) \\ y_t &= g(x_t) \end{aligned}$$

Called a *Hidden Markov Model*

- $x_0, w_0, w_1, \dots$ are independent
- We have measured $y_0, y_1, \dots, y_t$ and want to estimate x_t
- We'd like to do this recursively, so we don't store all our past measurements.

Hidden Markov Models

Consider the chain rule applied to the sequence

$$x_0, y_0, x_1, y_1, x_2, \ldots$$

We'll abbreviate again:

$$\begin{aligned}\mathbf{Prob}(y_t = q_t, x_t = z_t, y_{t-1} = q_{t-1}, x_{t-1} = z_{t-1}, \ldots, y_0 = q_0, x_0 = z_0)\\ = \mathbf{Prob}(y_t, x_t, \ldots, y_0, x_0)\end{aligned}$$

The chain rule gives

$$\begin{aligned}\mathbf{Prob}(y_t, x_t, \ldots, y_0, x_0) = & \mathbf{Prob}(y_t \,|\, x_t, y_{t-1}, x_{t-1}, \ldots, y_0, x_0)\\ & \mathbf{Prob}(x_t \,|\, y_{t-1}, x_{t-1}, \ldots, y_0, x_0)\\ & \mathbf{Prob}(y_{t-1} \,|\, x_{t-1}, y_{t-2}, x_{t-2}, \ldots, y_0, x_0)\\ & \quad \vdots\\ & \mathbf{Prob}(x_1 \,|\, y_0, x_0)\\ & \mathbf{Prob}(y_0 \,|\, x_0)\\ & \mathbf{Prob}(x_0)\end{aligned}$$

Hidden Markov Models

You can verify that

$$\mathbf{Prob}(y_t \,|\, x_t, y_{t-1}, x_{t-1}, \dots, y_0, x_0) = \mathbf{Prob}(y_t \,|\, x_t)$$

and

$$\mathbf{Prob}(x_t \,|\, y_{t-1}, x_{t-1}, \dots, y_0, x_0) = \mathbf{Prob}(x_t \,|\, x_{t-1})$$

Let

$$G_t(q_t, z_t) = \mathbf{Prob}(y_t = q_t \,|\, x_t = z_t)$$

and

$$P_{t+1}(z_{t+1}, z_t) = \mathbf{Prob}(x_{t+1} = z_{t+1} \,|\, x_t = z_t)$$

Hidden Markov Models

Then the chain rule gives

$$\begin{aligned}&\mathbf{Prob}(y_t = q_t, x_t = z_t, \dots, y_0 = q_0, x_0 = z_0)\\ &\qquad = G_t(q_t, z_t)P_t(z_t, z_{t-1})G_{t-1}(q_{t-1}, z_{t-1}) \dots G_0(q_0, z_0)\,\mathbf{Prob}(x_0 = z_0)\end{aligned}$$

The Kalman Filter

Define the $p_{t|t}$ to be the posterior probability of x_t given measurements of $y_0, y_1, \ldots, y_t$

$$p_{t|t}(z) = \mathbf{Prob}(x_t = z \,|\, y_0 = q_0, \ldots, y_t = q_t)$$

and $p_{t|t-1}$ to be the posterior probability of x_t given measurements of $y_0, y_1, \ldots, y_{t-1}$

$$p_{t|t-1}(z) = \mathbf{Prob}(x_t = z \,|\, y_0 = q_0, \ldots, y_{t-1} = q_{t-1})$$

- Note that $p_{t|t}$ also depends on $q_0, \ldots, q_t$ but we supress the dependence in the notation.
- In practice if the state space is finite, so x_t can take n possible values, then we store $p_{t|t}$ as a vector in $\mathbb{R}^n$.

The Kalman Filter

Since we know the joint distribution of $x_0, y_0, x_1, y_1, \dots, x_t, y_t$ we can immediately write down the conditional pmf from the joint pmf. This gives

$$p_{t+1|t}(z_{t+1}) = \frac{\sum_{z_0,\dots,z_t} P_{t+1}\dots P_0 G_t \dots G_0}{\sum_{z_0,\dots,z_{t+1}} P_{t+1}\dots P_0 G_t \dots G_0}$$

and

$$p_{t|t}(z_t) = \frac{\sum_{z_0,\dots,z_{t-1}} P_t\dots P_0 G_t \dots G_0}{\sum_{z_0,\dots,z_t} P_t\dots P_0 G_t \dots G_0}$$

- Here we have suppressed in the notation the arguments of the functions $P_t(z_t, z_{t-1})$ and $G_t(q_t, z_t)$
- The above expressions follow immediately from the definition of conditional pmf
- This is how one would compute posterior pmf for *batch* estimation; where we first measure all $y_0, \dots, y_t$ and then estimate x_t (or x_{t+1}.)

The Time Update

If we know $p_{t|t}$ then we can compute $p_{t+1|t}$ as follows.

$$\boxed{p_{t+1|t}(z_{t+1}) = \sum_{z_t} P_{t+1}(z_{t+1}, z_t) p_{t|t}(z_t)}$$

- This is called the *time update*.
- We don't have any more data, we just update our earlier posterior of x_t to give a new posterior of x_{t+1}
- We are just *propagating* the pmf under the transition matrix P_{t+1}.
- Easy to verify from definitions of $p_{t+1|t}$ and $p_{t|t}$

The Measurement Update

If we know $p_{t+1|t}$ then we can compute $p_{t+1|t+1}$ as follows.

$$\boxed{p_{t+1|t+1}(z_{t+1}) = \frac{G_{t+1}(q_{t+1}, z_{t+1})p_{t+1|t}(z_{t+1})}{\sum_a G_{t+1}(q_{t+1}, a)p_{t+1|t}(a)}}$$

- This is called the *measurement update*.
- We start with $p_{t+1|t}$, and use it as the prior for our next measurement of $y_{t+1} = q_{t+1}$.
- This is exactly the usual formula which we used for classification problems.
- Easy to verify from definitions of $p_{t+1|t}$ and $p_{t|t}$

The Kalman Filter

Now we have the Kalman Filter:

1. Start with $t = 0$ and

$$p_{0|-1}(z_0) = P_0(z_0) \qquad \text{prior on } x_0$$

2. Apply *measurement update*, *i.e.*, use $p_{t|t-1}$ and measured $y_t = q_t$ to construct $p_{t|t}$ by the formula

$$p_{t|t}(z_t) = \frac{G_t(q_t, z_t)p_{t|t-1}(z_t)}{\sum_a G_t(q_t, a)p_{t|t-1}(a)}$$

3. Apply *time update*, *i.e.*, use $p_{t|t}$ to construct $p_{t+1|t}$ according to

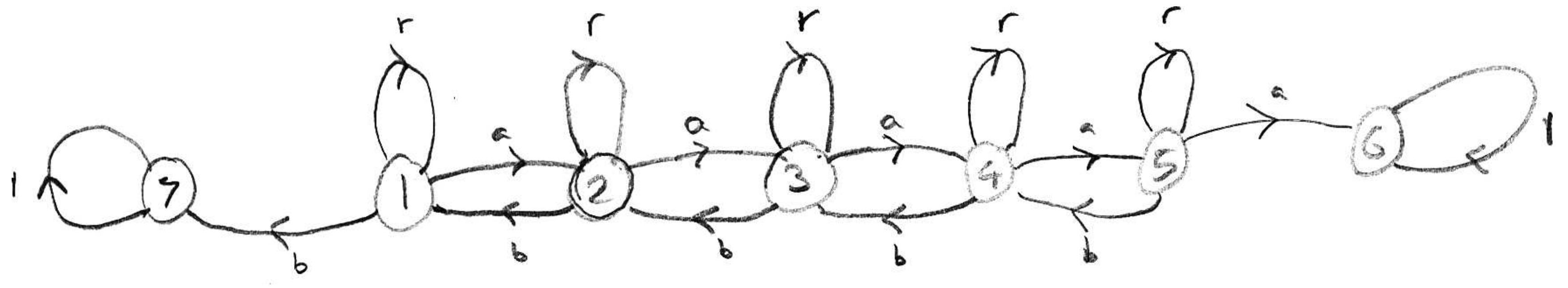
$$p_{t+1|t}(z_{t+1}) = \sum_{z_t} P_{t+1}(z_{t+1}, z_t)q_{t|t}(z_t)$$

4. $t \mapsto t + 1$. Goto 2.

Example

We have a gambler, who starts with 3. With each bet, he either gains a dollar, loses a dollar, or stays the same.

At each time t he tells us if he is doing better or worse than even. We have 7 states, as follows.



The dynamics is given by the transition matrix

$$P = \begin{bmatrix} r & a & & & & & b \\ b & r & a & & & & \\ & b & r & a & & & \\ & & b & r & a & & \\ & & & b & r & a & \\ & & & & & 1 & \\ & & & & & & 1 \end{bmatrix}$$

Note that state 7 means he has 0 dollars left, he stops when he has 6 dollars or 0 dollars.

Example

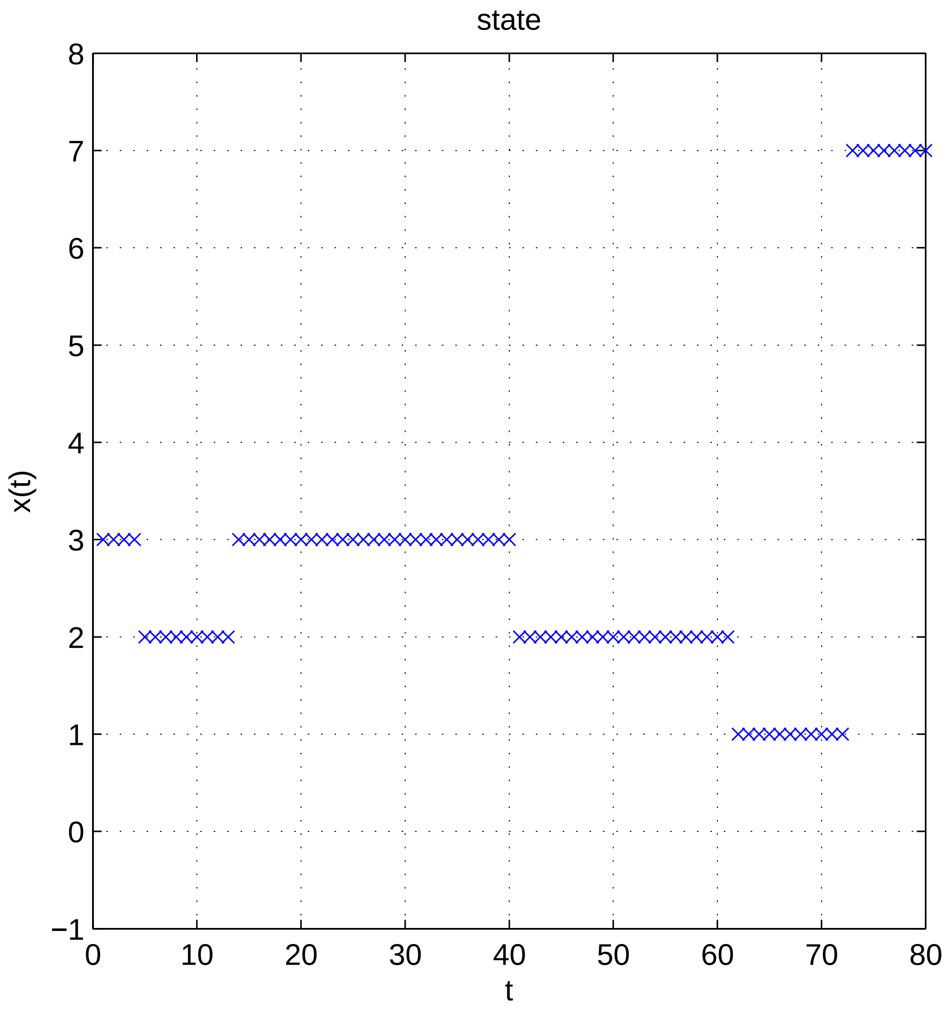
The measurment transition matrix is G, which is 2×7 is

$$G = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 1 & 1 & 0 \end{bmatrix}$$

We start with $x(0) = 3$, and parameters $a = b = 0.05$.

Example

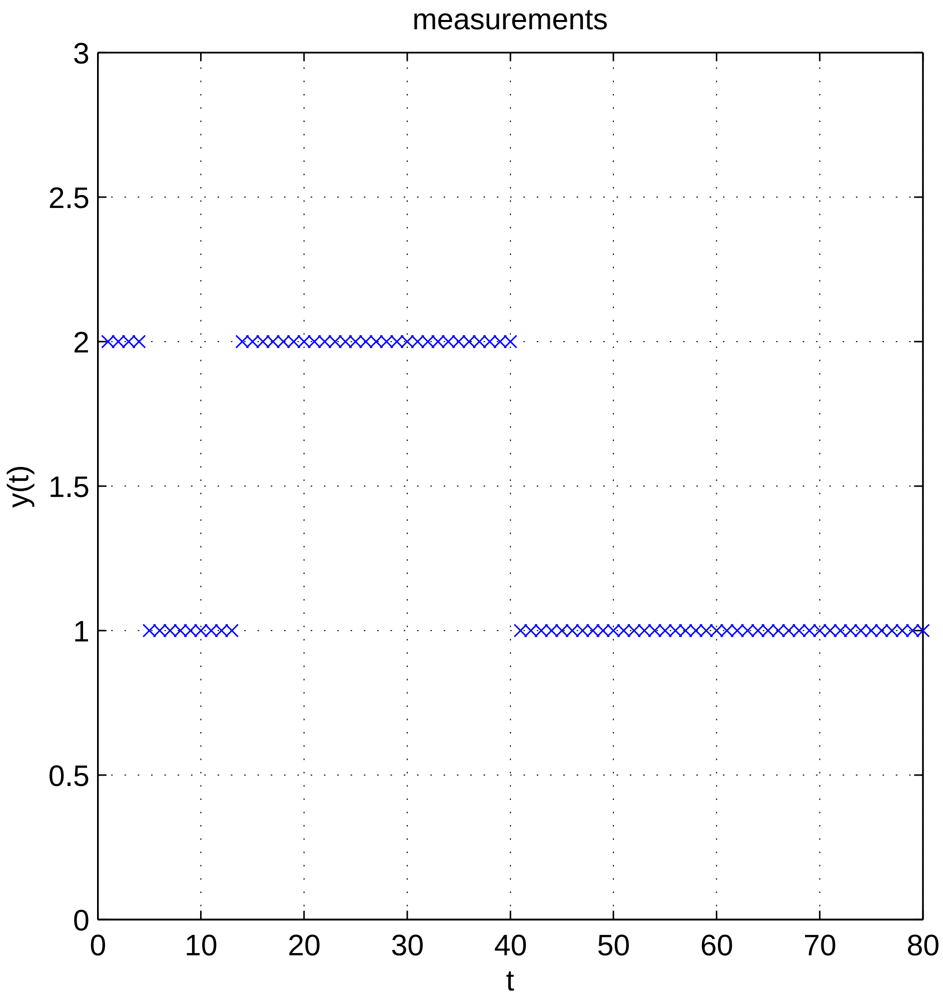
The state evolution looks like:



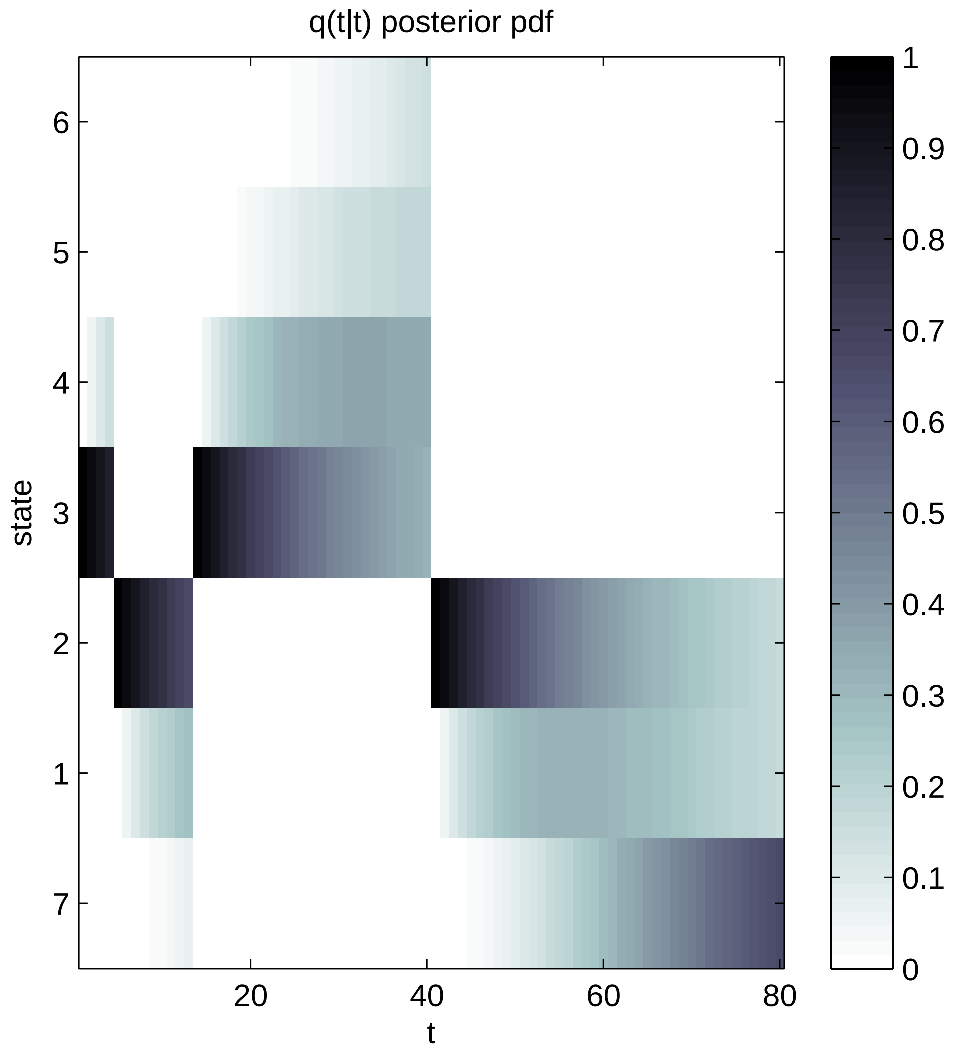
At time 72 the player runs out of money.

Example

The measurements are:



The posterior pdf



At times $0, 4, 13, 41$ we know the state exactly.

Linear systems driven by IID noise

consider the linear dynamical system

$$x(t+1) = Ax(t) + Bu(t) + v(t)$$

with

- the input $u(0), u(1), \dots$ is not random
- the disturbance $v(0), v(1), v(2), \dots$ is white Gaussian noise

$$\mathbf{E}\, v(t) = \mu_v(t) \qquad \mathbf{cov}\, v(t) = \Sigma_v$$

- the initial state is random $x(0) \sim \mathcal{N}(\mu_x(0), \Sigma_x(0))$, independent of $v(t)$ for all t

view this as *stochastic* simulation of the system

- what are the statistical properties (mean and covariance) of $x(t)$?

Evolution of Mean and Covariance

we have

$$x(t+1) = Ax(t) + Bu(t) + v(t)$$

taking the expectation of both sides, we have, as before

$$\boxed{\mu_x(t+1) = A\mu_x(t) + Bu(t) + \mu_v(t)}$$

taking the covariance of both sides, we have

$$\boxed{\Sigma_x(t+1) = A\Sigma_x(t)A^T + \Sigma_v}$$

i.e, the state covariance $\Sigma_x(t) = \mathbf{cov}(x(t))$ obeys a *Lyapunov recursion*

State Covariance

The solution to the Lyapunov recursion is

$$\boxed{\Sigma_x(t) = A^t \Sigma_x(0)(A^t)^T \; + \sum_{k=0}^{t-1} A^k \Sigma_v (A^k)^T}$$

Because the covariance of the state $\Sigma_x(t) = \mathbf{cov}\, x(t)$ is

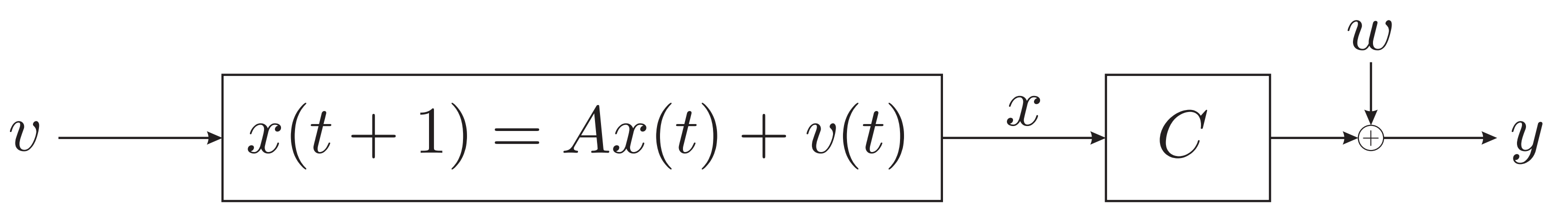
$$\begin{aligned}\Sigma_x(t) &= A^t \Sigma_x(0)(A^t)^T \; + \; \begin{bmatrix} A^{t-1} & \dots & A & I \end{bmatrix} \begin{bmatrix} \Sigma_v & & & \\ & \Sigma_v & & \\ & & \ddots & \\ & & & \Sigma_v \end{bmatrix} \begin{bmatrix} (A^{t-1})^T \\ \vdots \\ A^T \\ I \end{bmatrix} \\ &= A^t \Sigma_x(0)(A^t)^T \; + \sum_{k=0}^{t-1} A^k \Sigma_v (A^k)^T\end{aligned}$$

Linear Systems Estimation

Now let's add noisy measurements

$$\begin{aligned} x(t+1) &= Ax(t) + v(t) \\ y(t) &= Cx(t) + w(t) \end{aligned}$$

- $x(t) \in \mathbb{R}^n$ is the state
- $y(t) \in \mathbb{R}^p$ is the observed output
- $v(t) \in \mathbb{R}^n$ is called *process noise* or *disturbance*
- $w(t) \in \mathbb{R}^p$ is called *measurement noise*



assumptions

- $w(t) \sim \mathcal{N}(0, \Sigma_w)$ is white Gaussian noise
- $v(t) \sim \mathcal{N}(0, \Sigma_v)$ is white Gaussian noise
- $v(t)$ and $w(s)$ are independent for all s and t
- initial state is random $x(0) \sim \mathcal{N}(\mu_x(0), \Sigma_0)$, independent of $v(t)$

properties

the system is

$$\begin{aligned} x(t+1) &= Ax(t) + v(t) \\ y(t) &= Cx(t) + w(t) \end{aligned}$$

from the assumptions we can conclude

- sensor noise $w(t)$ is independent of $x(s)$ for all t, s
- disturbance $v(t)$ is independent of $x(0), \dots, x(t)$ and $y(0), \dots, y(t)$ i.e., past and current states and outputs
- *Markov property:* the process x is Markov, i.e.,

$$x(t) | x(0), \dots, x(t-1) = x(t) | x(t-1)$$

  roughly speaking: if you know $x(t-1)$ then knowledge of $x(t-2), \dots, x(0)$ doesn't give any more information

State Estimation

we use the notation

$$\hat{x}(t|s) = \mathbf{E}\big(x(t)|y(0),\dots,y(s)\big)$$

$$\Sigma_{t|s} = \mathbf{cov}\big(x(t)|y(0),\dots,y(s)\big)$$

- $\hat{x}(t|s)$ is the MMSE estimate of $x(t)$ based on $y(0),\dots,y(s)$
- $\Sigma_{t|s}$ is the conditional covariance of $x(t)$ given measurements of $y(0),\dots,y(s)$
- the random variable $x(t)|y(0),\dots,y(s)$ is Gaussian, with mean $\hat{x}(t|s)$ and covariance $\Sigma_{t|s}$

estimation problems

we first look at two state estimation problems:

- finding $\hat{x}(t|t)$, i.e., estimating the current state, based on the current and past observed outputs
- finding $\hat{x}(t+1|t)$, i.e., predicting the next state, based on the current and past observed outputs

since $x(t)$ and $y(t)$ are jointly Gaussian, we can use the standard formulae to find $\hat{x}(t|t)$ and similarly for $\hat{x}(t+1|t)$

$$\hat{x}(t|t) = \mu_x(t) + \mathbf{cov}\big(x(t), y_{\mathsf{seq}}(t)\big)\big(\mathbf{cov}(y_{\mathsf{seq}}(t))\big)^{-1}\big(y_{\mathsf{seq}}(t) - \mathbf{E}(y_{\mathsf{seq}}(t))\big)$$

where

$$y_{\mathsf{seq}}(t) = \begin{bmatrix} y(0) \\ \vdots \\ y(t) \end{bmatrix}$$

solution via large MMSE problem

$$y_{\mathsf{seq}}(t) = J_t x(0) \ + \ P_t v_{\mathsf{seq}}(t) \ + \ w_{\mathsf{seq}}(t)$$

$$x(t) = A^t x(0) \ + \ H_t v_{\mathsf{seq}}(t)$$

where

$$P_t = \begin{bmatrix} 0 & & & & & \\ C & 0 & & & & \\ CA & C & 0 & & & \\ CA^2 & CA & C & 0 & & \\ \vdots & & & & \ddots & \\ CA^{t-1} & & \dots & & & 0 \end{bmatrix} \qquad J_t = \begin{bmatrix} C \\ CA \\ CA^2 \\ CA^3 \\ \vdots \\ CA^t \end{bmatrix} \qquad H_t = \begin{bmatrix} A^{t-1} & A^{t-2} & \dots & A & I & 0 \end{bmatrix}$$

so

$$\mathbf{cov}(y_{\mathsf{seq}}(t)) = J_t \Sigma_0 J_t^T \ + \ P_t \Sigma_{v_{\mathsf{seq}}} P_t^T \ + \ \Sigma_{w_{\mathsf{seq}}}$$

$$\mathbf{cov}\big(x(t), y_{\mathsf{seq}}(t)\big) = A^t \Sigma_0 J_t^T + H_t \Sigma_{v_{\mathsf{seq}}} P_t^T$$

solution via large MMSE problem

so we can find $\hat{x}(t|t)$ from

$$\hat{x}(t|t) = \mu_x(t) + (A^t \Sigma_0 J_t^T + H_t \Sigma_{v_{\mathsf{seq}}} P_t^T) \times \left(J_t \Sigma_0 J_t^T + P_t \Sigma_{v_{\mathsf{seq}}} P_t^T + \Sigma_{w_{\mathsf{seq}}}\right)^{-1} \left(\begin{bmatrix} y(0) \\ \vdots \\ y(t) \end{bmatrix} - \begin{bmatrix} C\mu_x(0) \\ \vdots \\ C\mu_x(t) \end{bmatrix} \right)$$

the inverse is size $p(t+1) \times p(t+1)$, which grows with t

the *Kalman filter*

- is a recursive method for computing $\hat{x}(t|t)$ and $\hat{x}(t+1|t)$
- uses much more efficient computation
- doesn't require storing past measurements $y(0), \dots, y(t-1)$

The Kalman Filter

linear dynamical system

$$\begin{aligned} x(t+1) &= Ax(t) + v(t) \\ y(t) &= Cx(t) + w(t) \end{aligned}$$

assumptions

- $w(t) \sim \mathcal{N}(0, \Sigma_w)$, $v(t) \sim \mathcal{N}(0, \Sigma_v)$ are independent WGN
- initial state is random $x(0) \sim \mathcal{N}(\mu_x(0), \Sigma_0)$, independent of $v(t)$ and $w(t)$

notation: $\hat{x}(t|s)$ is the MMSE estimate of $x(t)$ based on $y(0), \ldots, y(s)$, $\Sigma_{t|s}$ is the error covariance

$$\hat{x}(t|s) = \mathbf{E}\big(x(t)|y(0), \ldots, y(s)\big)$$

$$\Sigma_{t|s} = \mathbf{cov}\big(x(t)|y(0), \ldots, y(s)\big)$$

Kalman filter recursion

We could just apply the (integral versions) of the Kalman filter formula to the linear system.

But we don't need to, since we know

- For measurement updates, if the prior is Gaussian, it is characterized by its mean and covariance.

  Then the posterior will also be Gaussian, and so to describe it we need only store its mean and covariance.

- For time update, if $p_{t|t}$ is Gaussian, and $x_{t+1} = Ax_t + v_t$, then x_{t+1} is Gaussian also.

Kalman filter recursion

start with prior mean and covariance of $x(0)$, given by

$$\hat{x}(0|-1) = \mu_0 \qquad \Sigma_{0|-1} = \Sigma_0$$

apply measurement update

$$\boxed{\begin{aligned} \hat{x}(t|t) &= \hat{x}(t|t-1) + \Sigma_{t|t-1}C^T(C\Sigma_{t|t-1}C^T + \Sigma_w)^{-1}\big(y(t) - C\hat{x}(t|t-1)\big) \\ \Sigma_{t|t} &= \Sigma_{t|t-1} - \Sigma_{t|t-1}C^T(C\Sigma_{t|t-1}C^T + \Sigma_w)^{-1}C\Sigma_{t|t-1} \end{aligned}}$$

apply time update

$$\boxed{\begin{aligned} \hat{x}(t+1|t) &= A\hat{x}(t|t) \\ \Sigma_{t+1|t} &= A\Sigma_{t|t}A^T + \Sigma_v \end{aligned}}$$

then repeat measurement and time update

Riccati recursion

to lighten notation, we'll use

$\hat{x}(t) = \hat{x}(t\|t-1)$	the predicted state
$\hat{\Sigma}_t = \Sigma_{t\|t-1}$	the covariance of the prediction error

applying the measurement and time updates to $\hat{\Sigma}_t$ gives

$$\boxed{\hat{\Sigma}_{t+1} = A\hat{\Sigma}_t A^T - A\hat{\Sigma}_t C^T (C\hat{\Sigma}_t C^T + \Sigma_w)^{-1} C\hat{\Sigma}_t A^T + \Sigma_v}$$

which is a Riccati recursion, with initial condition $\hat{\Sigma}_0 = \Sigma_0$

- $\hat{\Sigma}_t$ can be computed before any measurements are made
- so we can compute the estimation error covariance before we observe any data

Comparison with LQR

in LQR we have

$$P_{t-1} = A^T P_t A - A^T P_t B(B^T P_t B + R)^{-1} B^T P_t A + Q$$

- Riccati recursion for $P(t)$, determines minimum cost to go from time t
- runs *backwards* in time
- we can compute cost-to-go without knowing $x(t)$

in the Kalman filter

$$\hat{\Sigma}_{t+1} = A\hat{\Sigma}_t A^T - A\hat{\Sigma}_t C^T (C\hat{\Sigma}_t C^T + \Sigma_w)^{-1} C\hat{\Sigma}_t A^T + \Sigma_v$$

- Riccati recursion for $\hat{\Sigma}_t$, equal to the state prediction error covariance at time t
- runs *forwards* in time
- we can compute $\hat{\Sigma}_t$ before we take any measurements

Observer form

we can express the Kalman filter as

$$\begin{aligned}\hat{x}(t+1) &= A\hat{x}(t) + A\hat{\Sigma}_t C^T (C\hat{\Sigma}_t C^T + \Sigma_w)^{-1}\big(y(t) - C\hat{x}(t)\big) \\ &= A\hat{x}(t) + L_t\big(y(t) - \hat{y}(t)\big)\end{aligned}$$

where $L_t = A\hat{\Sigma}_t C^T (C\hat{\Sigma}_t C^T + \Sigma_w)^{-1}$ is the *observer gain*, and $\hat{y}(t) = \hat{y}(t|t-1)$

- $\hat{y}(t)$ is the output prediction, i.e. estimate of $y(t)$ based on $y(0), \dots, y(t-1)$
- $e(t) = y(t) - \hat{y}(t)$ is the output prediction error
- $A\hat{x}(t)$ is our prediction of $x(t+1)$ based on $y(0), \dots, y(t-1)$
- estimate of $x(t+1)$ is prediction based on $y(0), \dots, y(t-1)$ plus a linear function of the output prediction error

steady-state Kalman filter

- as in LQR, Riccati recursion for $\hat{\Sigma}_t$ converges to a steady-state value $\hat{\Sigma}$, provided (C, A) is observable and (A, Σ_v) is controllable
- $\hat{\Sigma}$ is the steady-state error covariance for estimating $x(t+1)$ given $y(0), \ldots, y(t)$
- note that state prediction error converges, even if system is unstable

$\hat{\Sigma}$ satisfies the ARE

$$\hat{\Sigma} = A\hat{\Sigma}A^T - A\hat{\Sigma}C^T(C\hat{\Sigma}C^T + \Sigma_w)^{-1}C\hat{\Sigma}A^T + \Sigma_v$$

which can be solved directly

the steady-state filter is a time-invariant observer

$$\boxed{\hat{x}(t+1) = A\hat{x}(t) + L\big(y(t) - \hat{y}(t)\big)}$$

where $L = A\hat{\Sigma}C^T(C\hat{\Sigma}C^T + \Sigma_w)^{-1}$ and $\hat{y}(t) = C\hat{x}(t)$

steady-state Kalman filter

define the estimation error $\tilde{x}(t) = x(t) - \hat{x}(t)$, then

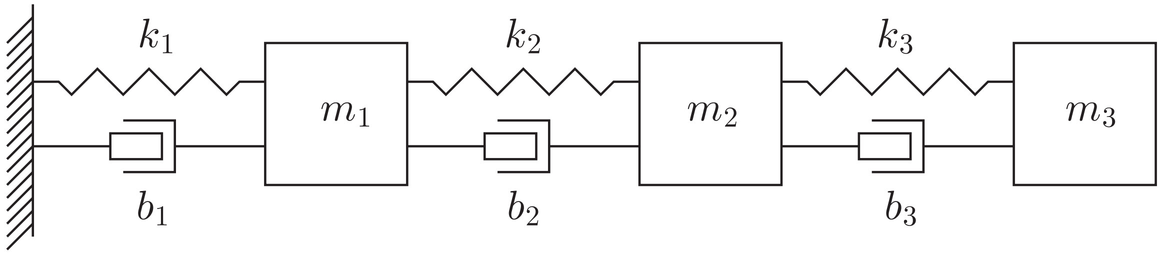
$$\begin{aligned} y(t) - \hat{y}(t) &= Cx(t) + w(t) - C\hat{x}(t) \\ &= C\tilde{x}(t) + w(t) \end{aligned}$$

and

$$\begin{aligned} \tilde{x}(t+1) &= x(t+1) - \hat{x}(t+1) \\ &= Ax(t) + v(t) - A\hat{x}(t) - L\big(C\tilde{x}(t) + w(t)\big) \\ &= (A - LC)\tilde{x}(t) + v(t) - Lw(t) \end{aligned}$$

- so the estimation error propagates according to a linear system with dynamics matrix $A - LC$
- this system is driven by the process $v(t) - Lw(t)$ which is IID with zero mean and covariance $\Sigma_V + L\Sigma_w L^T$
- if (A, Σ_v) is controllable and (C, A) is observable, then $A - LC$ is stable

example: mass-spring system



masses $m_i = 1$, springs $k_i = 2$, dampers $b_i = 0.1$, $\Sigma_w = 0.1I$, $\Sigma_v = 0.2I$

$$\begin{aligned}\dot{x}(t) &= A_c x(t) + B_{c1} v(t) + B_{c2} u(t) \\ y(t) &= C x(t) + w(t)\end{aligned}$$

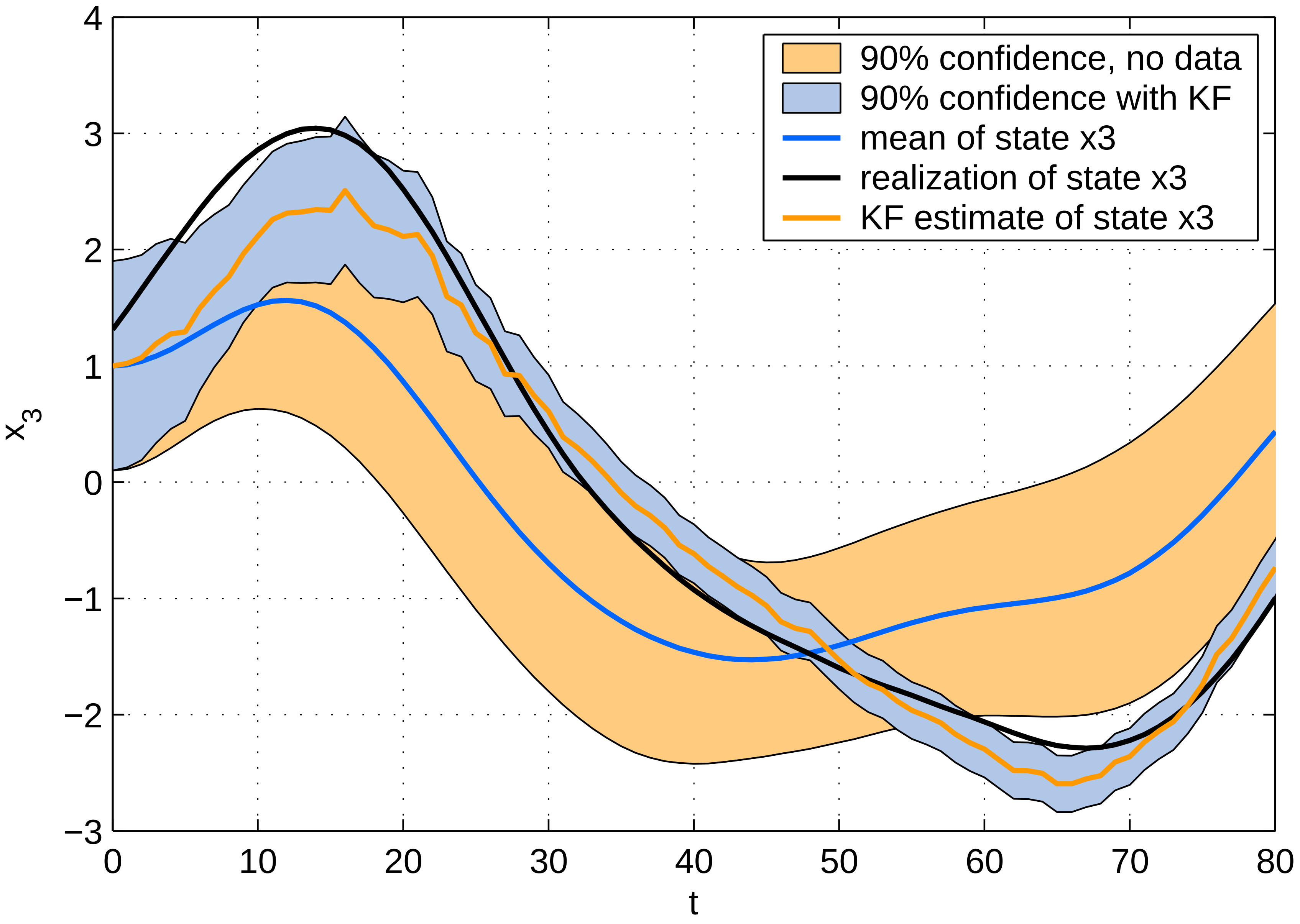
where

$$A_c = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ -4 & 2 & 0 & -0.2 & 0.1 & 0 \\ 2 & -4 & 2 & 0.1 & -0.2 & 0.1 \\ 0 & 2 & -2 & 0 & 0.1 & -0.1 \end{bmatrix} \qquad B_{c1} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad B_{c2} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix}$$

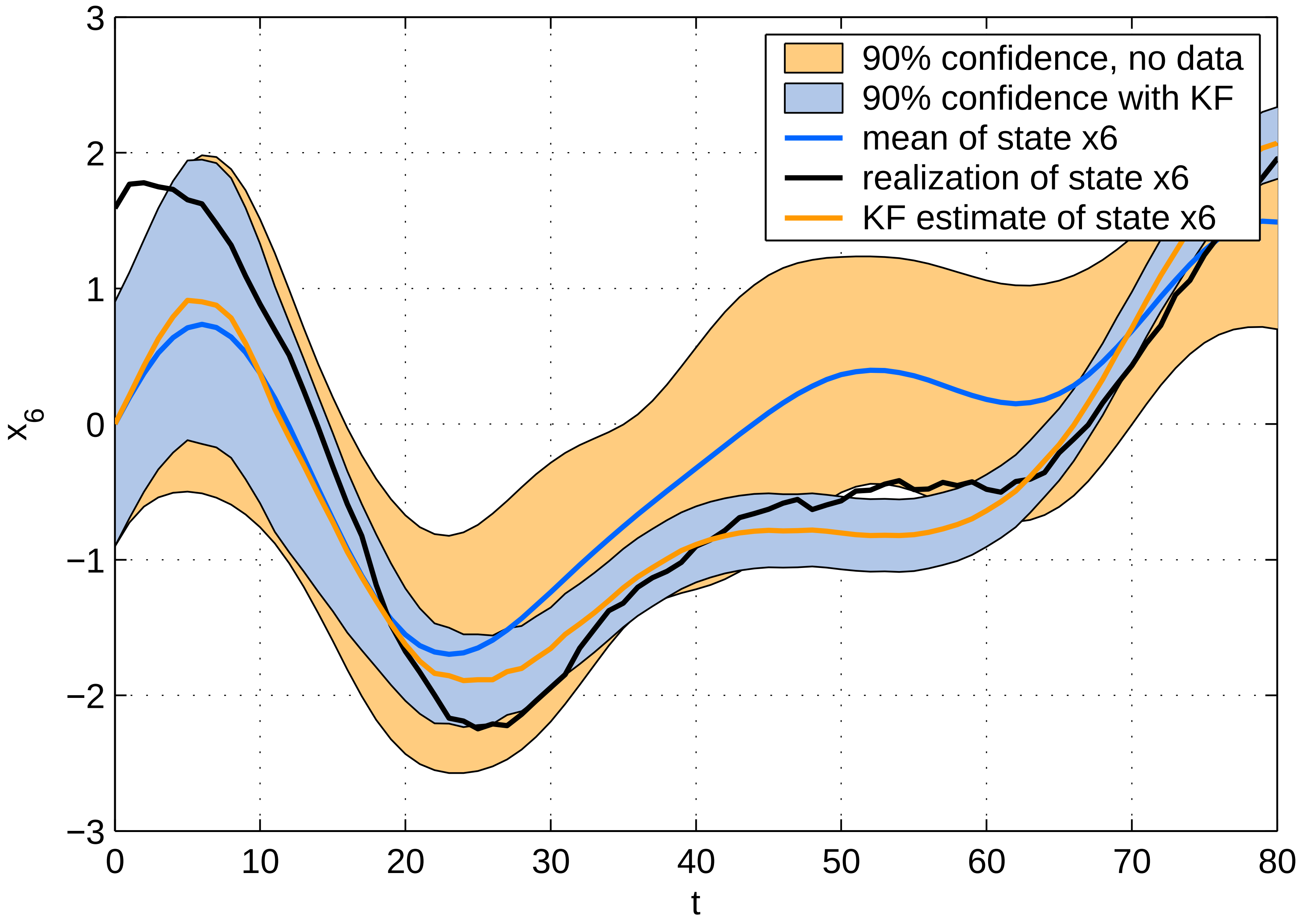
example: mass-spring system

position of mass 3



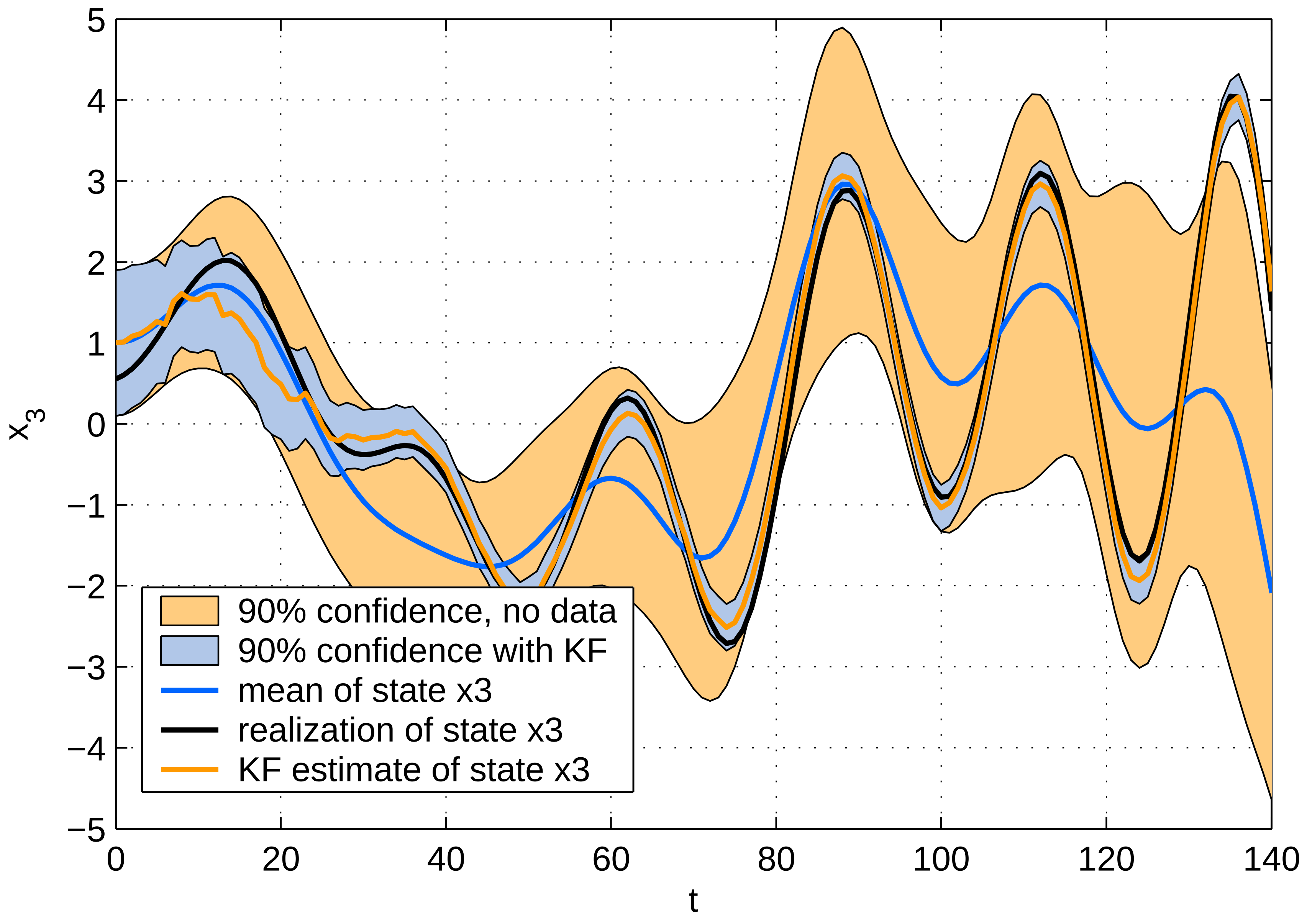
example: mass-spring system

velocity of mass 3



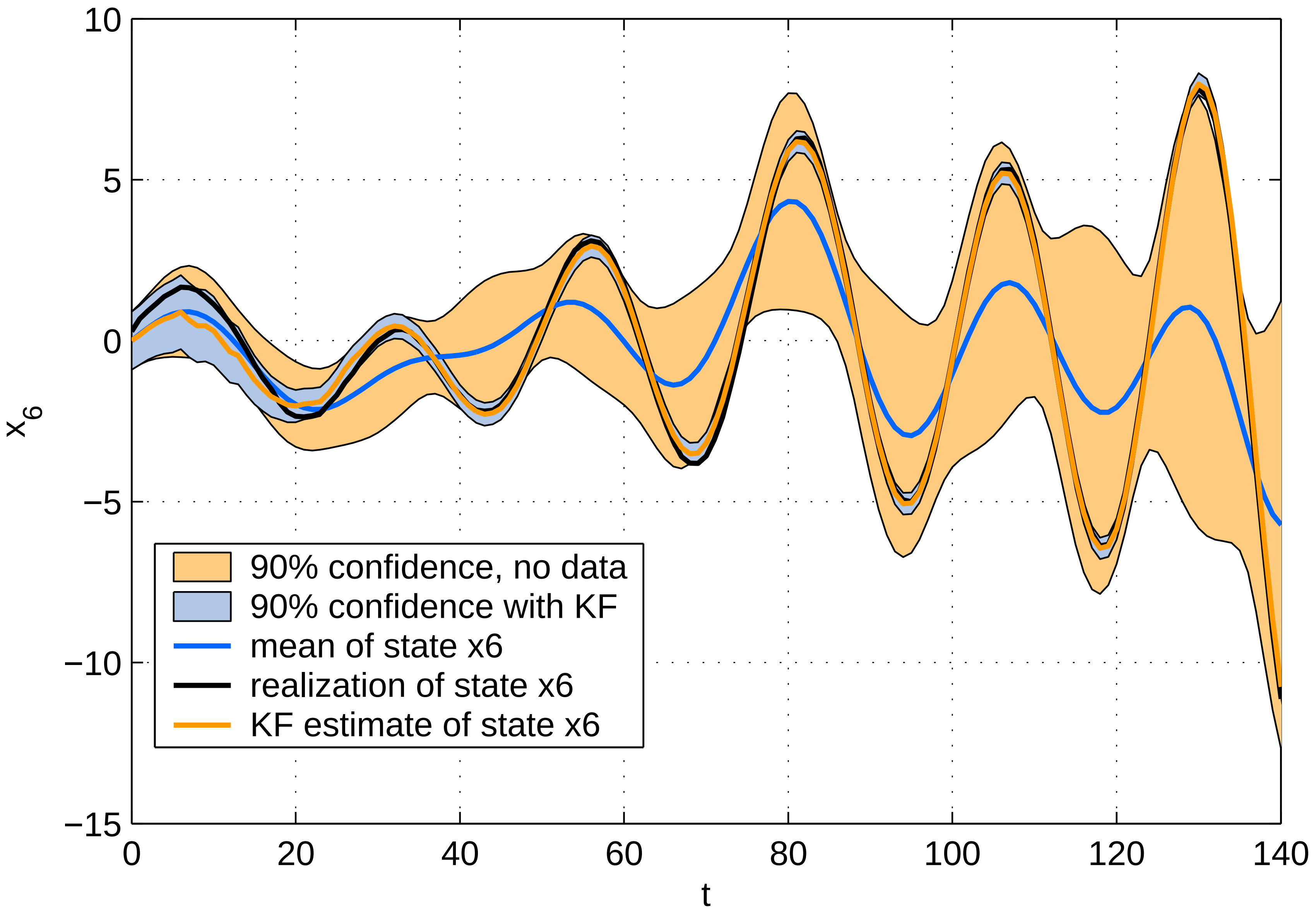
example: mass-spring system

unstable case, (changed sign of dampers) position of mass 3



example: mass-spring system

unstable case, (changed sign of dampers) velocity of mass 3


t
x6
90% confidence, no data
90% confidence with KF
mean of state x6
realization of state x6
KF estimate of state x6