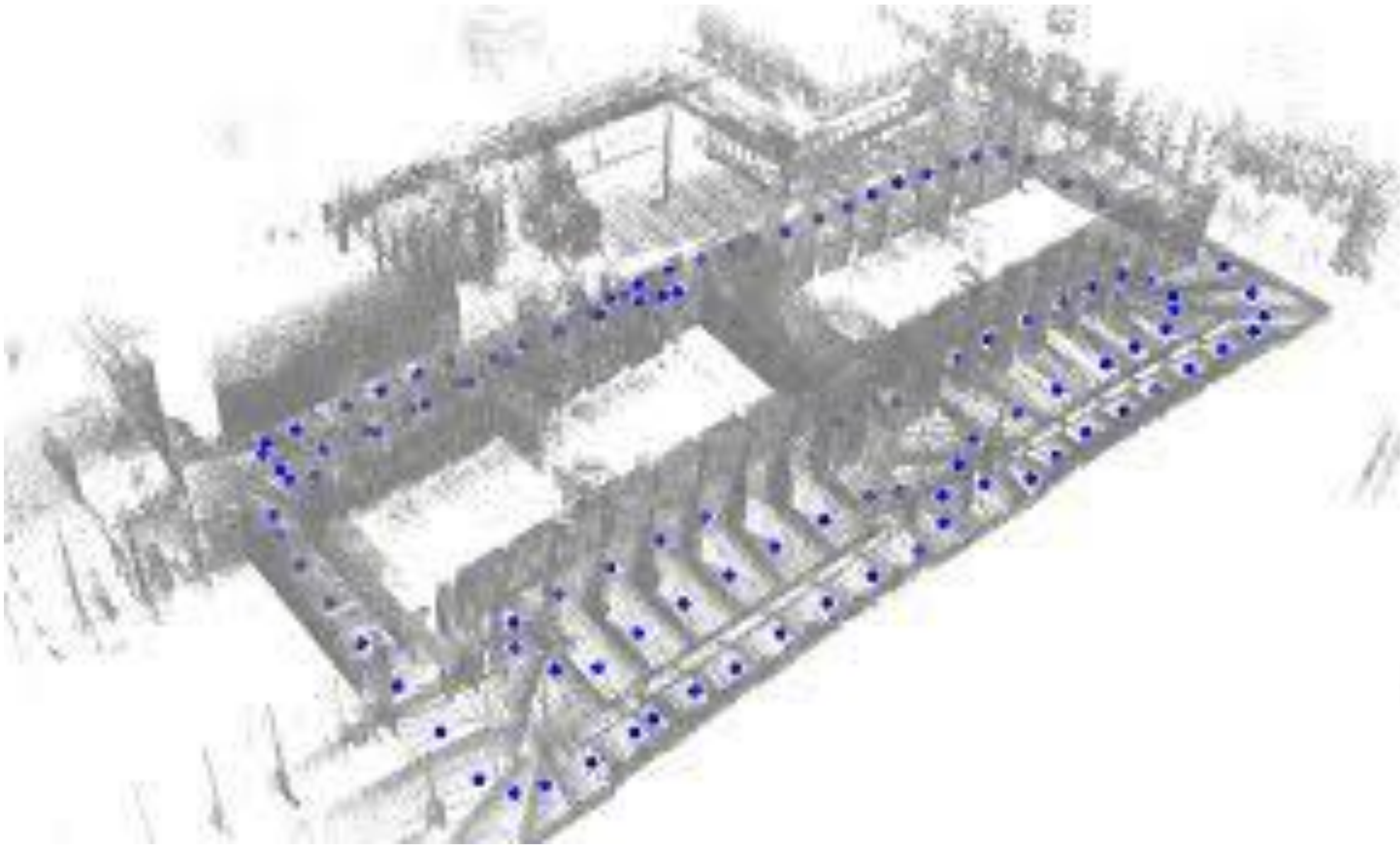
Temporal inference and SLAM



http://www.cs.cmu.edu/~16385/

16-385 Computer Vision
Spring 2018, Lecture 25

Course announcements

- Homework 7 will be posted and will be due on <u>Sunday</u> 6th.
    - You can use all of your remaining late days for it.

- RI Seminar this week: Vladlen Koltun, “Learning to drive”, Friday 3:30-4:30pm.
    - Very exciting speaker, you should all attend.
    - Make sure to go early as the room will be packed.
    - Do you want me to move my office hours so that you can make it to the talk?

Overview of today's lecture

- Temporal state models.
- Temporal inference.
- Kalman filtering.
- Extended Kalman filtering.
- Mono SLAM.

Slide credits

Most of these slides were adapted from:

- Kris Kitani (16-385, Spring 2017).

Temporal state models

Represent the 'world' as a set of random variables $\mathbf{X}$

$\boldsymbol{X} = \{\boldsymbol{x}, \boldsymbol{y}\}$ location on the ground plane

$\boldsymbol{X} = \{\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}\}$ position in the 3D world

$\boldsymbol{X} = \{\boldsymbol{x}, \dot{\boldsymbol{x}}\}$ position and velocity

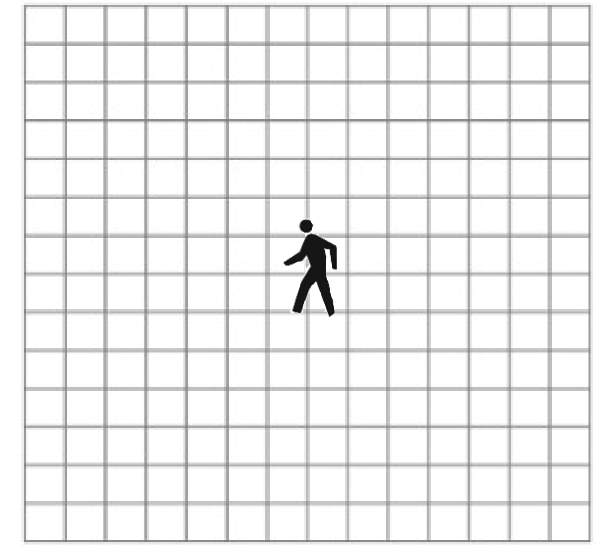
$\boldsymbol{X} = \{\boldsymbol{x}, \dot{\boldsymbol{x}}, \boldsymbol{f}_1, \ldots, \boldsymbol{f}_n\}$

position, velocity and location of landmarks

Object tracking (localization)

$$\boldsymbol{X} = \{\boldsymbol{x}, \boldsymbol{y}\}$$

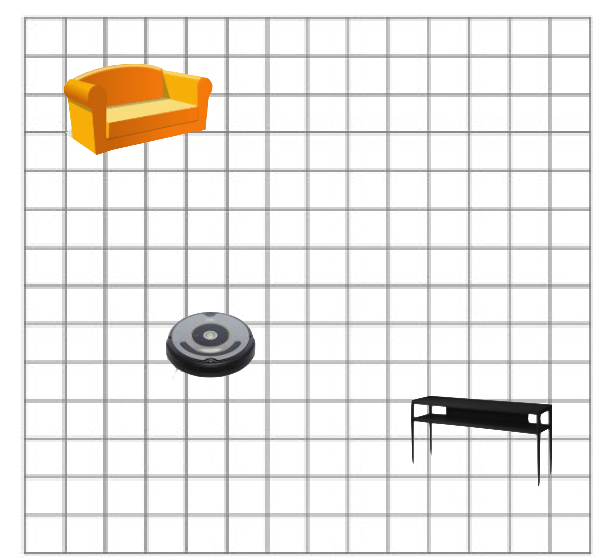
e.g., location on the ground plane



Object location and world landmarks (localization and mapping)

$$\boldsymbol{X} = \{\boldsymbol{x}, \dot{\boldsymbol{x}}, \boldsymbol{f}_1, \ldots, \boldsymbol{f}_n\}$$

e.g., position and velocity of robot
and location of landmarks



$$\boldsymbol{x}_t$$

The state of the world changes over time

$$\boldsymbol{X}_t$$

The state of the world changes over time

So we use a sequence of random variables:

$$\boldsymbol{X}_0, \boldsymbol{X}_1, \ldots, \boldsymbol{X}_t$$

$$\boldsymbol{X}_t$$

The state of the world changes over time

So we use a sequence of random variables:

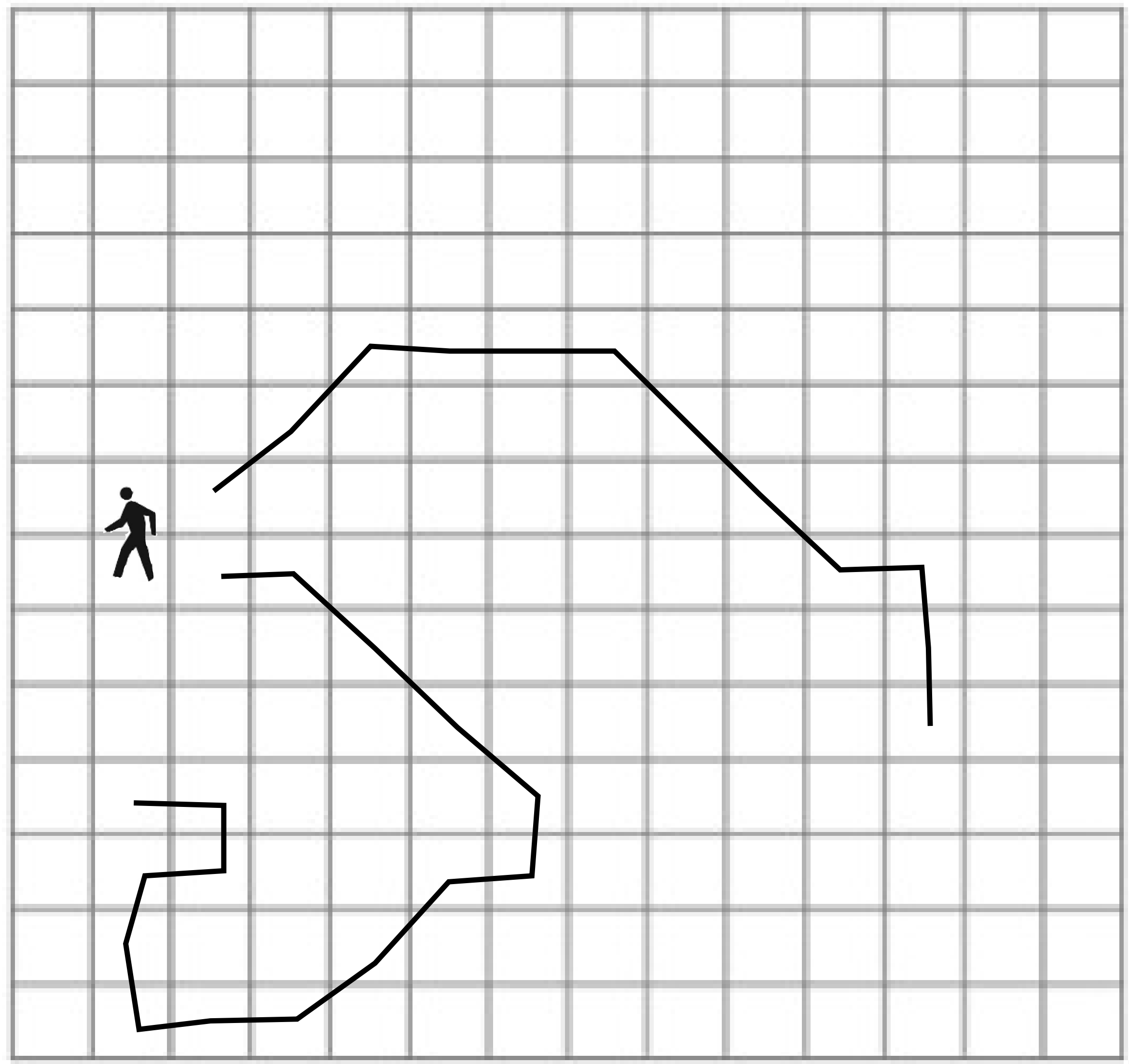
$$\boldsymbol{X}_0, \boldsymbol{X}_1, \ldots, \boldsymbol{X}_t$$

The state of the world is usually **uncertain** so we think in terms of a distribution

$$P(\boldsymbol{X}_0, \boldsymbol{X}_1, \ldots, \boldsymbol{X}_t)$$

How big is the space of this distribution?

If the state space is $\boldsymbol{X} = \{\boldsymbol{x}, \boldsymbol{y}\}$ the location on the ground plane



$$P(\boldsymbol{X}_0, \boldsymbol{X}_1, \ldots, \boldsymbol{X}_t)$$

is the probability over all possible trajectories through a room of length t+1

When we use a sensor (camera),
we don't have direct access to the state but noisy observations of the state

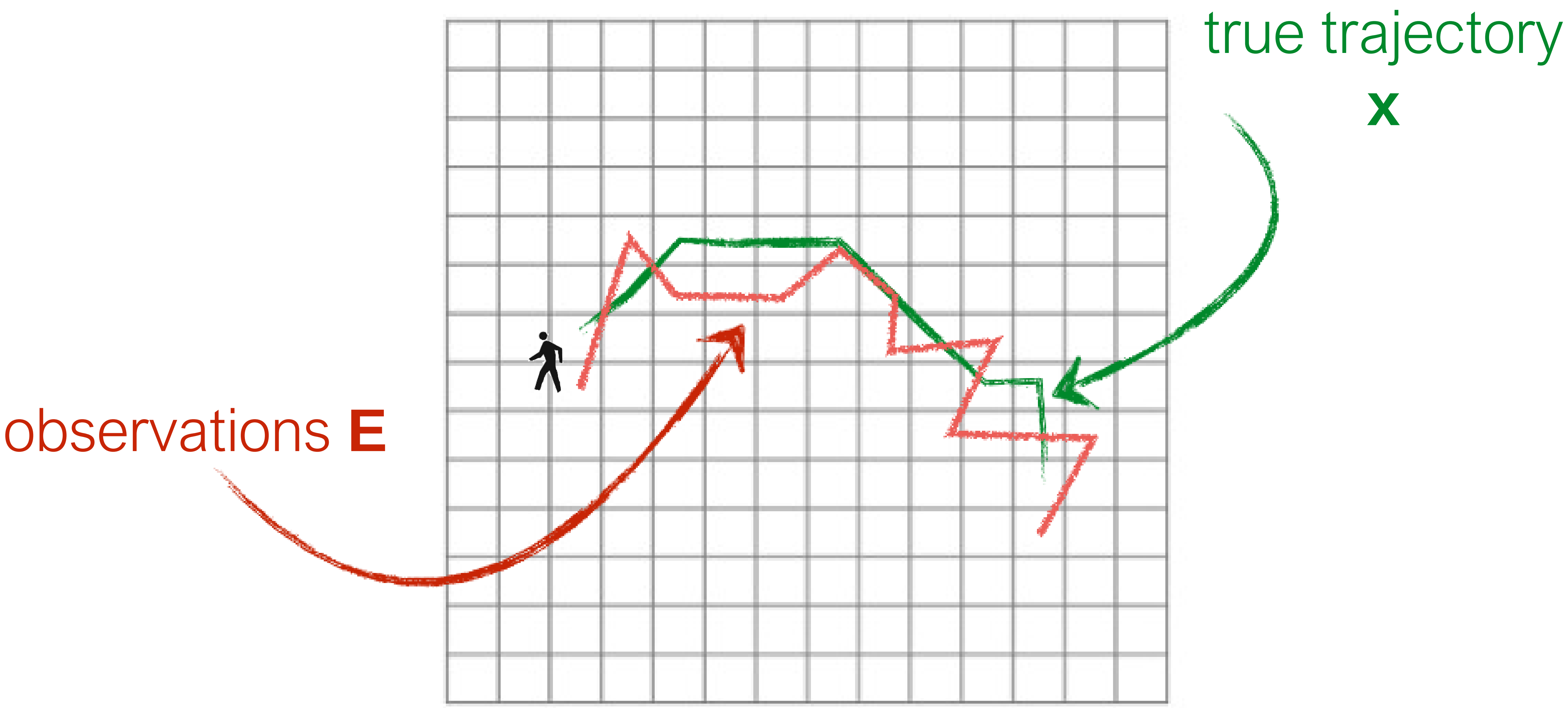
$$\boldsymbol{E}_t$$

$$\boldsymbol{X}_0, \boldsymbol{X}_1, \ldots, \boldsymbol{X}_t, \boldsymbol{E}_1, \boldsymbol{E}_2, \ldots, \boldsymbol{E}_t$$

(**all** possible ways of observing **all** possible trajectories)

How big is the space of this distribution?

all possible ways of observing **all** possible trajectories of length t



So we think of the world in terms of the distribution

$$P(\underbrace{\boldsymbol{X}_0, \boldsymbol{X}_1, \ldots, \boldsymbol{X}_t}_{\substack{\text{unobserved variables} \\ \text{(hidden state)}}}, \underbrace{\boldsymbol{E}_1, \boldsymbol{E}_2, \ldots, \boldsymbol{E}_t}_{\substack{\text{observed variables} \\ \text{(evidence)}}})$$

So we think of the world in terms of the distribution

$$P(\underbrace{\boldsymbol{X}_0, \boldsymbol{X}_1, \ldots, \boldsymbol{X}_t}_{\substack{\text{unobserved variables} \\ \text{(hidden state)}}}, \underbrace{\boldsymbol{E}_1, \boldsymbol{E}_2, \ldots, \boldsymbol{E}_t}_{\substack{\text{observed variables} \\ \text{(evidence)}}})$$

How big is the space of this distribution?

So we think of the world in terms of the distribution

$$P(\underbrace{\boldsymbol{X}_0, \boldsymbol{X}_1, \ldots, \boldsymbol{X}_t}_{\substack{\text{unobserved variables} \\ \text{(hidden state)}}}, \underbrace{\boldsymbol{E}_1, \boldsymbol{E}_2, \ldots, \boldsymbol{E}_t}_{\substack{\text{observed variables} \\ \text{(evidence)}}})$$

How big is the space of this distribution?

Can you think of a way to reduce the space?

Reduction 1. Stationary process assumption:

'a process of change that is governed by laws that do not themselves change over time.'

$$P(\boldsymbol{E}_t|\boldsymbol{X}_t) = P_t(\boldsymbol{E}_t|\boldsymbol{X}_t)$$

the model doesn't change over time

Reduction 1. Stationary process assumption:

'a process of change that is governed by laws that do not themselves change over time.'

$$P(\boldsymbol{E}_t|\boldsymbol{X}_t) = P_t(\boldsymbol{E}_t|\boldsymbol{X}_t)$$

the model doesn't change over time

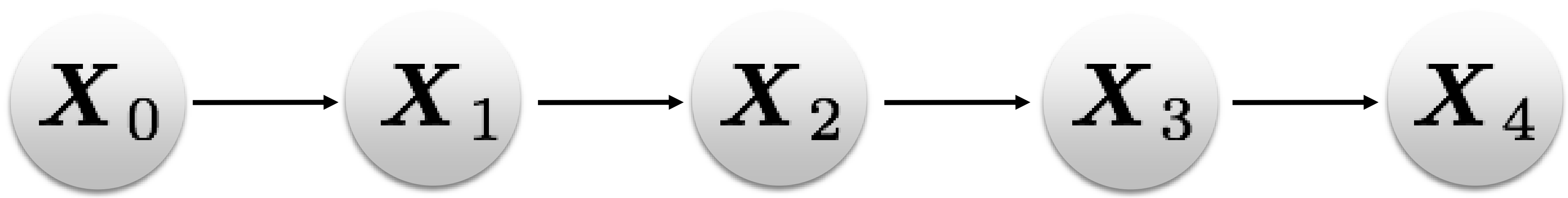
Only have to store **one** model.

Is this a reasonable assumption?

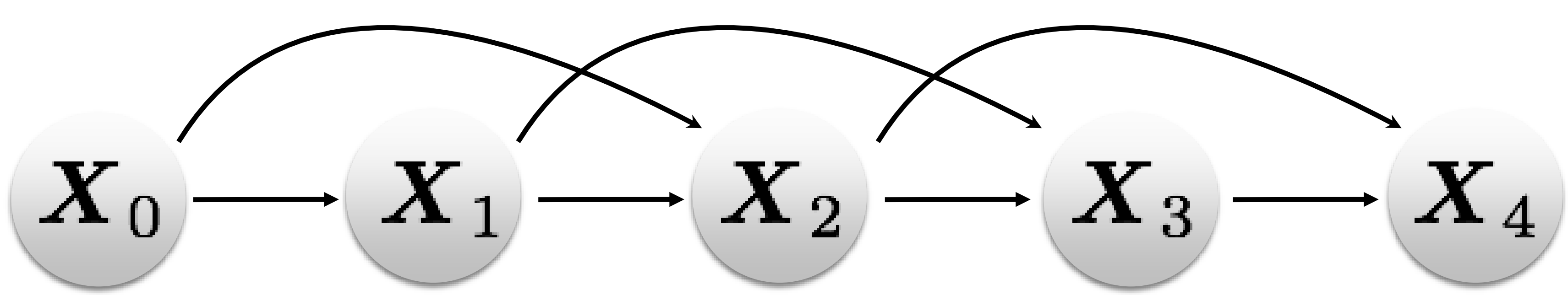
Reduction 2. Markov Assumption:

‘the current state only depends on a finite history of previous states.’

First-order Markov Model: $P(\boldsymbol{X}_t|\boldsymbol{X}_{t-1})$.



Second-order Markov Model: $P(\boldsymbol{X}_t|\boldsymbol{X}_{t-1}, \boldsymbol{X}_{t-2})$



(this relationship is called the **motion** model)

Reduction 2. Markov Assumption:

'the current observation only depends on current state.'

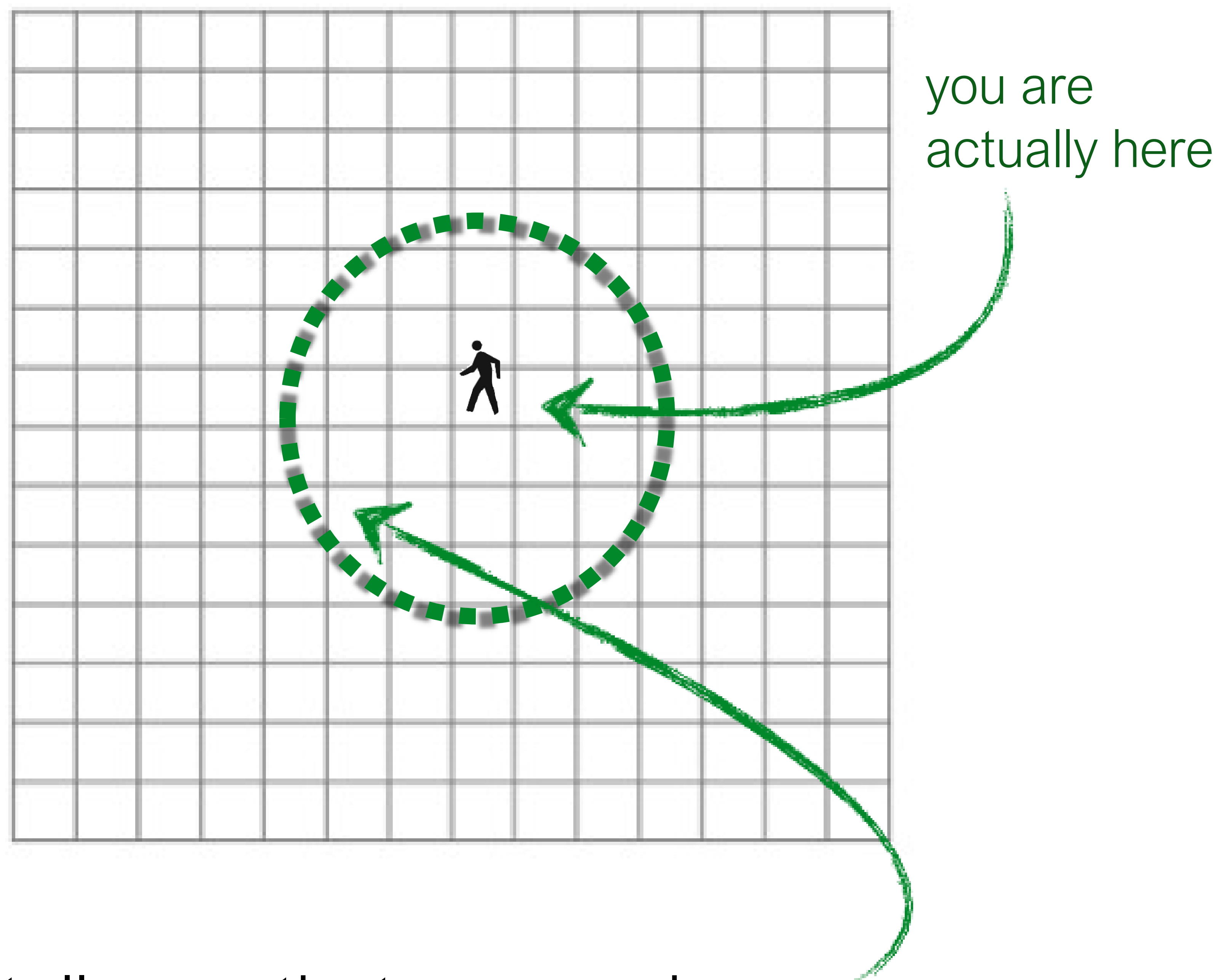
The current observation is usually
most influenced by the current state

$$P(\boldsymbol{E}_t|\boldsymbol{X}_t)$$

(this relationship is called the **observation** model)

Can you think of an observation of a state?

For example, GPS is a noisy observation of location.



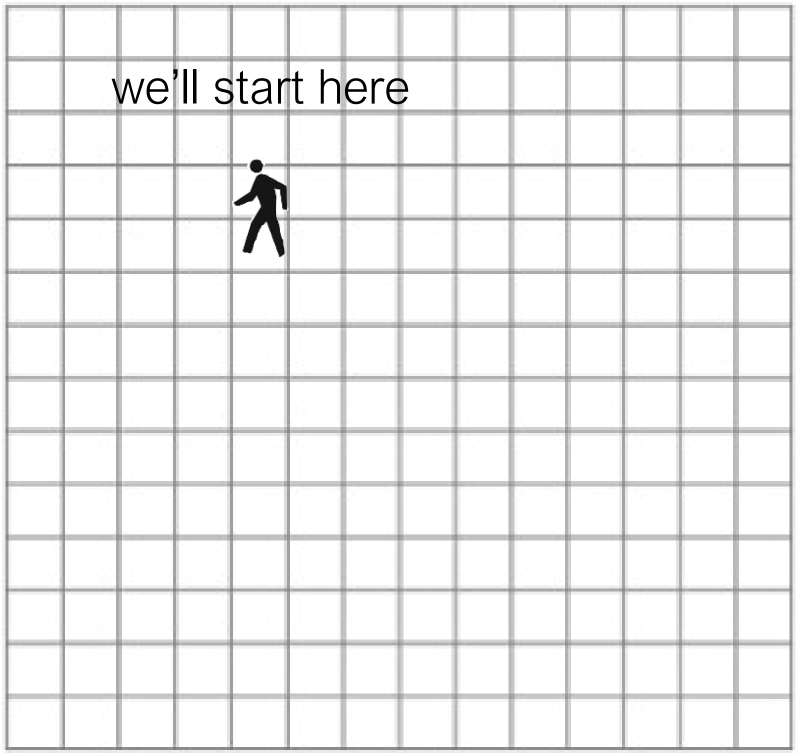
But GPS tells you that you are here

with probability $P(\boldsymbol{E}_t | \boldsymbol{X}_t)$

Reduction 3. Prior State Assumption:

‘we know where the process (probably) starts’

$$\boldsymbol{X}_0$$



Applying these assumptions,
we can decompose the joint probability:

$$P(\boldsymbol{X}_0\boldsymbol{X}_1,\ldots,\boldsymbol{X}_T,\boldsymbol{E}_1\boldsymbol{E}_1,\ldots,\boldsymbol{E}_T) = P(\boldsymbol{X}_0)\prod_{t=1}^{T} P(\boldsymbol{X}_t|\boldsymbol{X}_{t-1})P(\boldsymbol{E}_t|\boldsymbol{X}_t)$$

Stationary process assumption:
only have to store ____ models
(assuming only a single variable for state and observation)

Markov assumption:
This is a model of order ___

We have significantly reduced the number of parameters

Joint Probability of a Temporal Sequence

$$P(\boldsymbol{X}_0)\prod_{t=1}^{T}P(\boldsymbol{X}_t|\boldsymbol{X}_{t-1})P(\boldsymbol{E}_t|\boldsymbol{X}_t)$$

state prior prior	motion model transition model	sensor model observation model

Joint Probability of a Temporal Sequence

$$P(\boldsymbol{X}_0)\prod_{t=1}^{T} P(\boldsymbol{X}_t|\boldsymbol{X}_{t-1})P(\boldsymbol{E}_t|\boldsymbol{X}_t)$$

state prior
prior

motion model
transition model

sensor model
observation model

Joint Distribution for a Dynamic Bayesian Network

specific instances of a DBN
covered in this class

Hidden Markov Model

(typically taught as discrete but not necessarily)

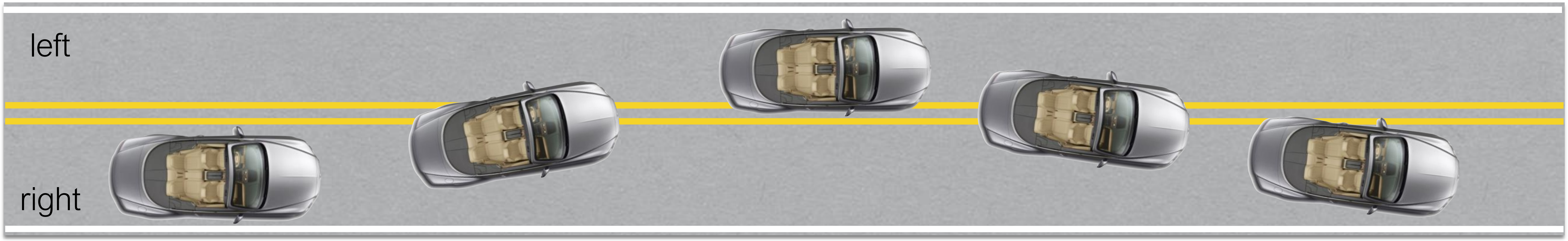
Kalman Filter

(Gaussian motion model, prior and observation model)

Hidden Markov Model

Hidden Markov Model example

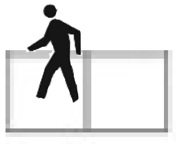
'In the trunk of a car of a sleepy driver' model



binary random variable (left lane or right lane)

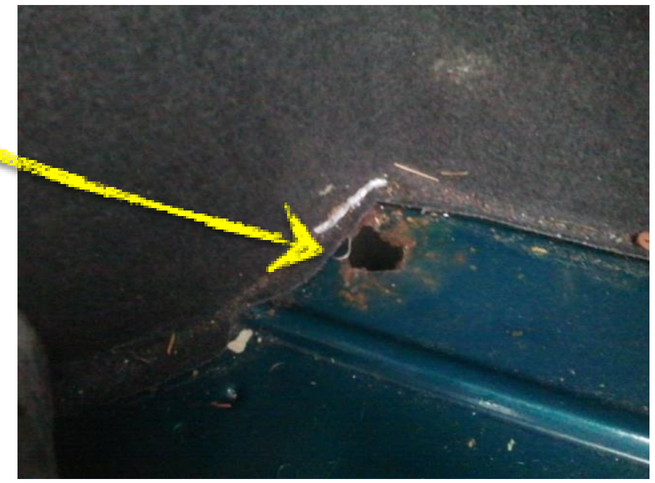
$$\boldsymbol{x}_0 \rightarrow \boldsymbol{x}_1 \rightarrow \boldsymbol{x}_2 \rightarrow \boldsymbol{x}_3 \rightarrow \boldsymbol{x}_4$$

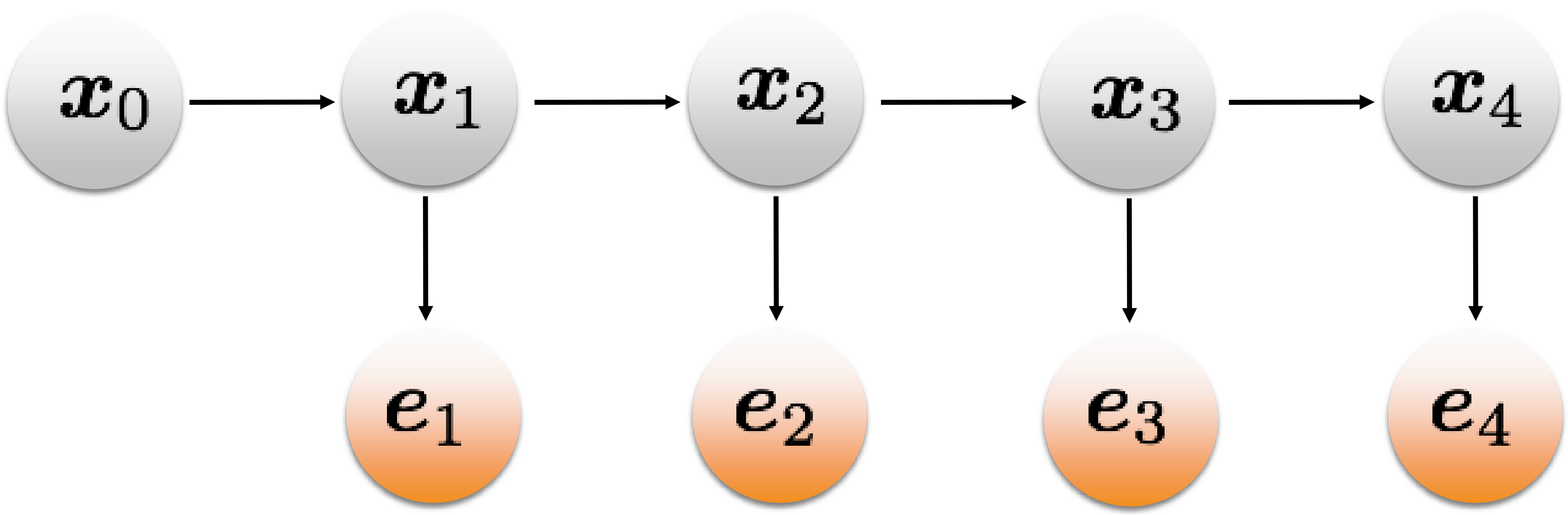
$$\boldsymbol{x} = \{x_{\text{left}}, x_{\text{right}}\}$$



two state world!

From a hole in the car you can see the ground





binary random variable (road is yellow or road is gray)

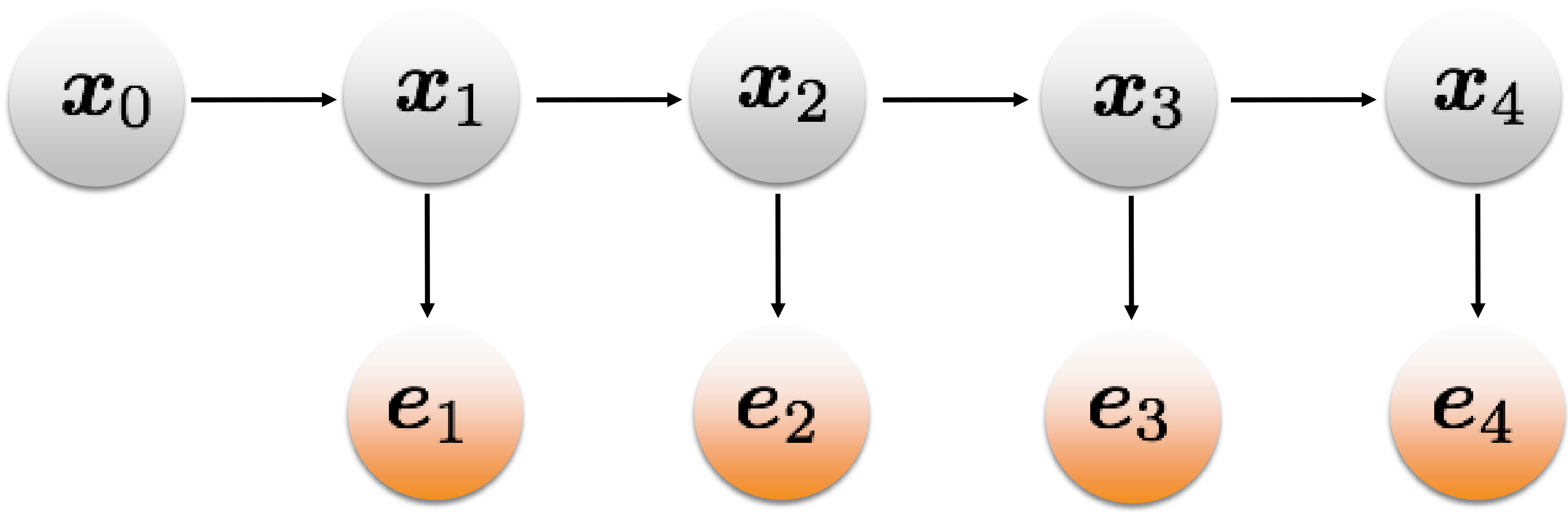
$$\boldsymbol{e} = \{e_{\text{gray}}, e_{\text{yellow}}\}$$

Nice visualization here: http://setosa.io/ev/markov-chains/?utm_content=buffer5e504

	x_{left}	x_{right}
$P(\boldsymbol{x}_0)$	0.5	0.5

$P(\boldsymbol{x}_t \mid \boldsymbol{x}_{t-1})$	x_{left}	x_{right}
x_{left}	0.7	0.3
x_{right}	0.3	0.7

What needs to sum to one?



What's the probability of staying in the left lane if I'm in the left lane?

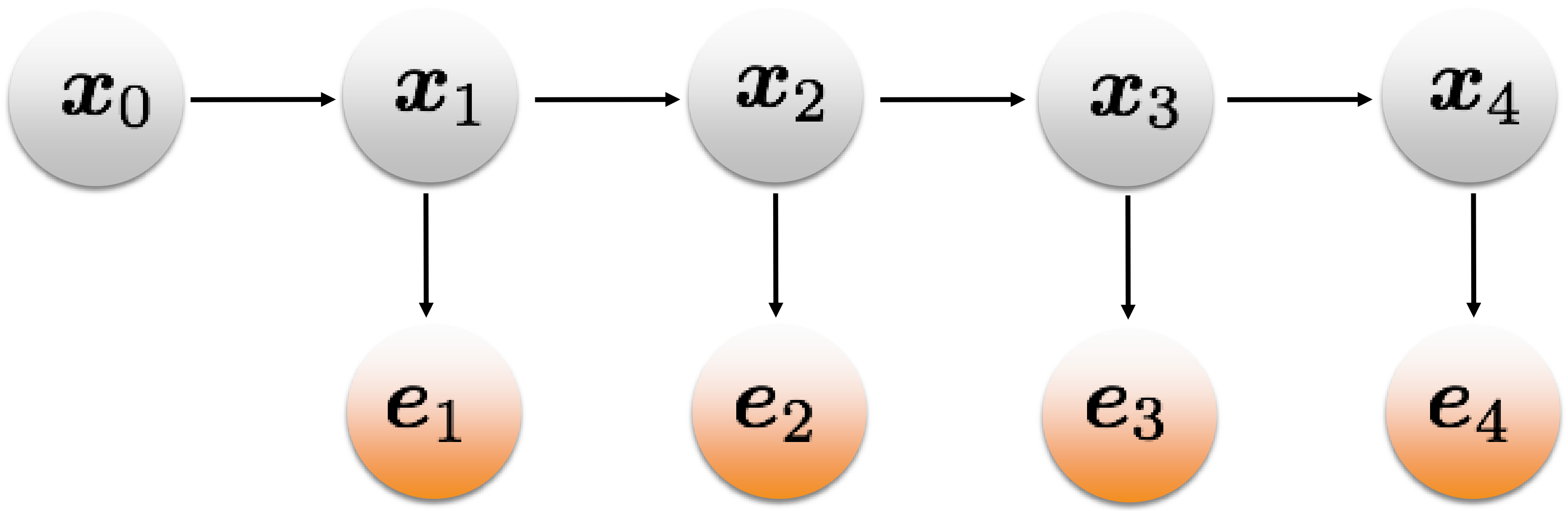
$P(\boldsymbol{e}_t \mid \boldsymbol{x}_t)$	x_{left}	x_{right}
e_{yellow}	0.9	0.2
e_{gray}	0.1	0.8

What lane am I in if I see yellow?

visualization of the motion model

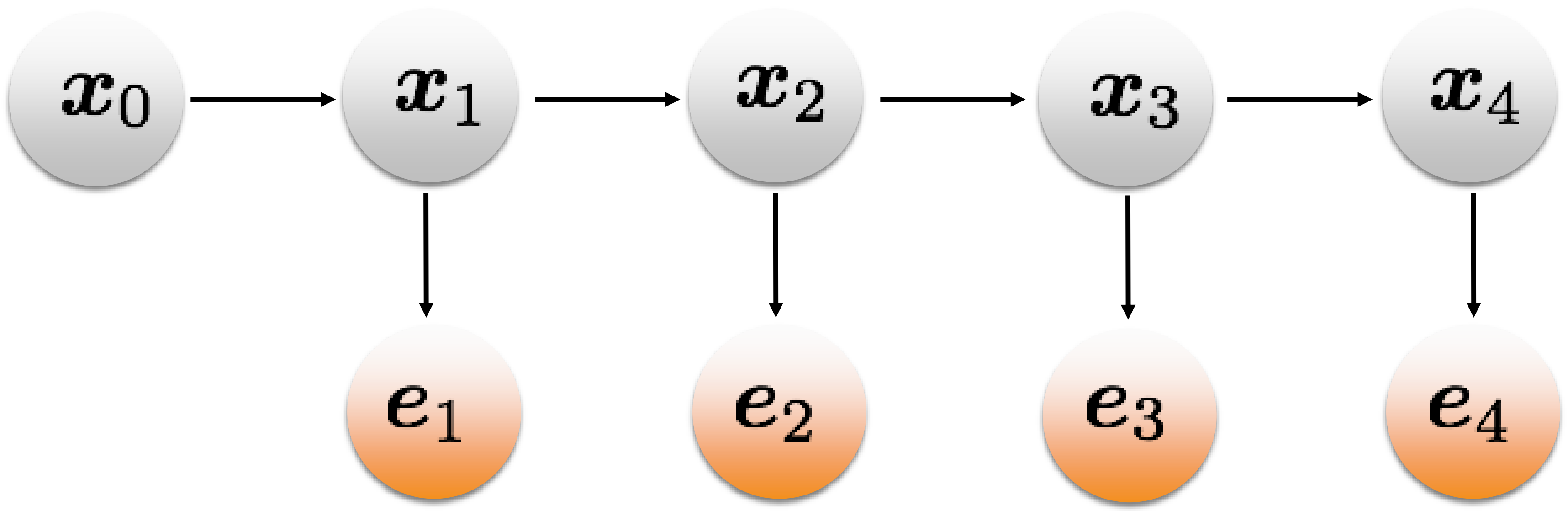
$P(\boldsymbol{x}_t \mid \boldsymbol{x}_{t-1})$	$x_{t-1} = R$	$x_{t-1} = S$
$x_t = R$	0.9	0.1
$x_t = S$	0.1	0.9

Is the stationary assumption true?



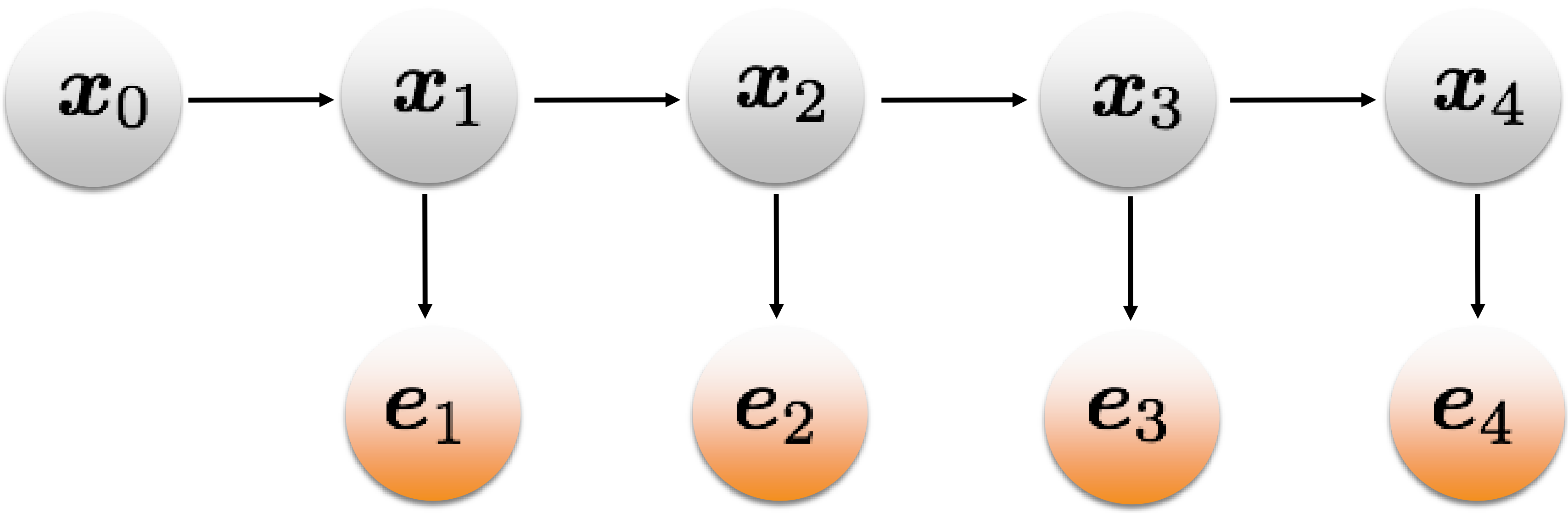
Is the stationary assumption true?

visibility at night?
visibility after a day in the car?
still swerving after one day of driving?



Is the stationary assumption true?

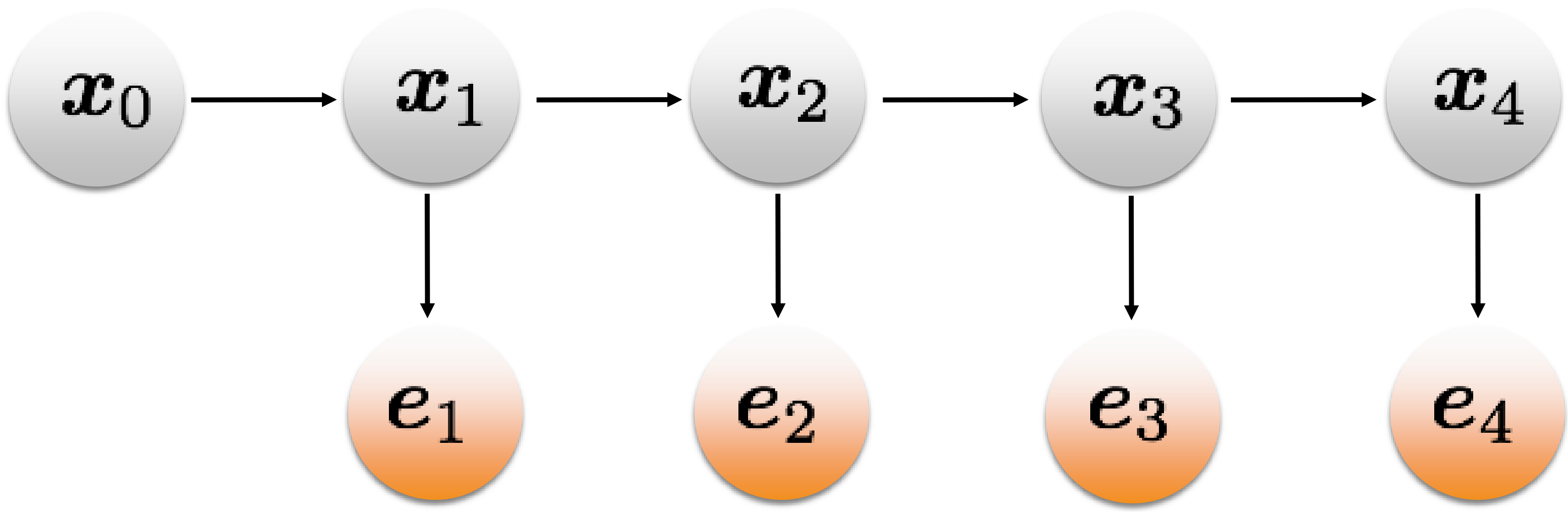
visibility at night?
visibility after a day in the car?
still swerving after one day of driving?



Is the Markov assumption true?

Is the stationary assumption true?

visibility at night?
visibility after a day in the car?
still swerving after one day of driving?



Is the Markov assumption true?

what can you learn with higher order models?
what if you have been in the same lane for the last hour?

In general, assumptions are not correct but they simplify the problem and work most of the time when designed appropriately

Temporal inference

Basic Inference Tasks

Filtering

$$P(\boldsymbol{X}_t|\boldsymbol{e}_{1:t})$$

Posterior probability over the **current** state, given all evidence up to present

Prediction

$$P(\boldsymbol{X}_{t+k}|\boldsymbol{e}_{1:t})$$

Posterior probability over a **future** state, given all evidence up to present

Smoothing

$$P(\boldsymbol{X}_k|\boldsymbol{e}_{1:t})$$

Posterior probability over a **past** state, given all evidence up to present

Best Sequence

$$\underset{\boldsymbol{X}_{1:t}}{\arg\max}\, P(\boldsymbol{X}_{1:t}|\boldsymbol{e}_{1:t})$$

Best state sequence given all evidence up to present

Filtering

$$P(\mathbf{X}_t|\mathbf{e}_{1:t})$$

Posterior probability over the **current** state, given all evidence up to present

Where am I now?

Filtering

Can be computed with recursion (Dynamic Programming)

$$P(\boldsymbol{X}_{t+1}|\boldsymbol{e}_{1:t+1}) \propto P(\boldsymbol{e}_{t+1}|\boldsymbol{X}_{t+1}) \sum_{\boldsymbol{X}_t} P(\boldsymbol{X}_{t+1}|\boldsymbol{X}_t) P(\boldsymbol{X}_t|\boldsymbol{e}_{1:t})$$

posterior observation model motion model prior

Filtering

Can be computed with recursion (Dynamic Programming)

$$P(\boldsymbol{X}_{t+1}|\boldsymbol{e}_{1:t+1}) \propto \underbrace{P(\boldsymbol{e}_{t+1}|\boldsymbol{X}_{t+1})}_{\text{observation model}} \sum_{\boldsymbol{X}_t} \underbrace{P(\boldsymbol{X}_{t+1}|\boldsymbol{X}_t)}_{\text{motion model}} \underline{P(\boldsymbol{X}_t|\boldsymbol{e}_{1:t})}$$

What is this?

Filtering

Can be computed with recursion (Dynamic Programming)

$$\underline{P(\boldsymbol{X}_{t+1}|\boldsymbol{e}_{1:t+1})} \propto P(\boldsymbol{e}_{t+1}|\boldsymbol{X}_{t+1}) \sum_{\boldsymbol{X}_t} P(\boldsymbol{X}_{t+1}|\boldsymbol{X}_t)\underline{P(\boldsymbol{X}_t|\boldsymbol{e}_{1:t})}$$

same type of 'message'

Filtering

Can be computed with recursion (Dynamic Programming)

$$\underbrace{P(\boldsymbol{X}_{t+1}|\boldsymbol{e}_{1:t+1})} \propto P(\boldsymbol{e}_{t+1}|\boldsymbol{X}_{t+1}) \sum_{\boldsymbol{X}_t} P(\boldsymbol{X}_{t+1}|\boldsymbol{X}_t)\underbrace{P(\boldsymbol{X}_t|\boldsymbol{e}_{1:t})}$$

same type of 'message'

called a **belief distribution**

sometimes people use this annoying notation instead: $Bel(x_t)$

a belief is a reflection of the systems (robot, tracker) knowledge about the state **X**

Filtering

Can be computed with recursion (Dynamic Programming)

$$\underline{P(\boldsymbol{X}_{t+1}|\boldsymbol{e}_{1:t+1})} \propto P(\boldsymbol{e}_{t+1}|\boldsymbol{X}_{t+1}) \sum_{\boldsymbol{X}_t} P(\boldsymbol{X}_{t+1}|\boldsymbol{X}_t)\underline{P(\boldsymbol{X}_t|\boldsymbol{e}_{1:t})}$$

Where does this equation come from?

(scary math to follow…)

Filtering

Can be computed with recursion (Dynamic Programming)

$$\underline{P(\boldsymbol{X}_{t+1}|\boldsymbol{e}_{1:t+1})} \propto P(\boldsymbol{e}_{t+1}|\boldsymbol{X}_{t+1}) \sum_{\boldsymbol{X}_t} P(\boldsymbol{X}_{t+1}|\boldsymbol{X}_t)\underline{P(\boldsymbol{X}_t|\boldsymbol{e}_{1:t})}$$

just splitting up the notation here

$$P(\boldsymbol{X}_{t+1}|\boldsymbol{e}_{1:t+1}) = P(\boldsymbol{X}_{t+1}|\boldsymbol{e}_{t+1}, \boldsymbol{e}_{1:t})$$

Filtering

Can be computed with recursion (Dynamic Programming)

$$\underline{P(\mathbf{X}_{t+1}|\mathbf{e}_{1:t+1})} \propto P(\mathbf{e}_{t+1}|\mathbf{X}_{t+1}) \sum_{\mathbf{X}_t} P(\mathbf{X}_{t+1}|\mathbf{X}_t)\underline{P(\mathbf{X}_t|\mathbf{e}_{1:t})}$$

$$P(\mathbf{X}_{t+1}|\mathbf{e}_{1:t+1}) = P(\mathbf{X}_{t+1}|\mathbf{e}_{t+1}, \mathbf{e}_{1:t})$$

Apply Bayes' rule (with evidence)

Filtering

Can be computed with recursion (Dynamic Programming)

$$\underline{P(\boldsymbol{X}_{t+1}|\boldsymbol{e}_{1:t+1})} \propto P(\boldsymbol{e}_{t+1}|\boldsymbol{X}_{t+1}) \sum_{\boldsymbol{X}_t} P(\boldsymbol{X}_{t+1}|\boldsymbol{X}_t)\underline{P(\boldsymbol{X}_t|\boldsymbol{e}_{1:t})}$$

$$\begin{aligned} P(\boldsymbol{X}_{t+1}|\boldsymbol{e}_{1:t+1}) &= P(\boldsymbol{X}_{t+1}|\boldsymbol{e}_{t+1}, \boldsymbol{e}_{1:t}) \\ &= \frac{P(\boldsymbol{e}_{t+1}|\boldsymbol{X}_{t+1}, \boldsymbol{e}_{1:t})P(\boldsymbol{X}_{t+1}|\boldsymbol{e}_{1:t})}{P(\boldsymbol{e}_{t+1}|\boldsymbol{e}_{1:t})} \end{aligned}$$

Apply Markov assumption on observation model

Filtering

Can be computed with recursion (Dynamic Programming)

$$\underline{P(\boldsymbol{X}_{t+1}|\mathbf{e}_{1:t+1})} \propto P(\mathbf{e}_{t+1}|\boldsymbol{X}_{t+1}) \sum_{\boldsymbol{X}_t} P(\boldsymbol{X}_{t+1}|\boldsymbol{X}_t)\underline{P(\boldsymbol{X}_t|\mathbf{e}_{1:t})}$$

$$\begin{aligned} P(\boldsymbol{X}_{t+1}|\mathbf{e}_{1:t+1}) &= P(\boldsymbol{X}_{t+1}|\mathbf{e}_{t+1}, \mathbf{e}_{1:t}) \\ &= \frac{P(\mathbf{e}_{t+1}|\boldsymbol{X}_{t+1}, \mathbf{e}_{1:t})P(\boldsymbol{X}_{t+1}|\mathbf{e}_{1:t})}{P(\mathbf{e}_{t+1}|\mathbf{e}_{1:t})} \\ &= \alpha P(\mathbf{e}_{t+1}|\boldsymbol{X}_{t+1})P(\boldsymbol{X}_{t+1}|\mathbf{e}_{1:t}) \end{aligned}$$

Condition on the previous state $\mathbf{X}_t$

Filtering

Can be computed with recursion (Dynamic Programming)

$$\underline{P(\boldsymbol{X}_{t+1}|\boldsymbol{e}_{1:t+1})} \propto P(\boldsymbol{e}_{t+1}|\boldsymbol{X}_{t+1}) \sum_{\boldsymbol{X}_t} P(\boldsymbol{X}_{t+1}|\boldsymbol{X}_t)\underline{P(\boldsymbol{X}_t|\boldsymbol{e}_{1:t})}$$

$$\begin{aligned}
P(\boldsymbol{X}_{t+1}|\boldsymbol{e}_{1:t+1}) &= P(\boldsymbol{X}_{t+1}|\boldsymbol{e}_{t+1}, \boldsymbol{e}_{1:t}) \\
&= \frac{P(\boldsymbol{e}_{t+1}|\boldsymbol{X}_{t+1}, \boldsymbol{e}_{1:t})P(\boldsymbol{X}_{t+1}|\boldsymbol{e}_{1:t})}{P(\boldsymbol{e}_{t+1}|\boldsymbol{e}_{1:t})} \\
&= \alpha P(\boldsymbol{e}_{t+1}|\boldsymbol{X}_{t+1})P(\boldsymbol{X}_{t+1}|\boldsymbol{e}_{1:t}) \\
&= \alpha P(\boldsymbol{e}_{t+1}|\boldsymbol{X}_{t+1}) \sum_{\boldsymbol{X}_t} P(\boldsymbol{X}_{t+1}|\boldsymbol{X}_t, \boldsymbol{e}_{1:t})P(\boldsymbol{X}_t|\boldsymbol{e}_{1:t})
\end{aligned}$$

Apply Markov assumption on motion model

Filtering

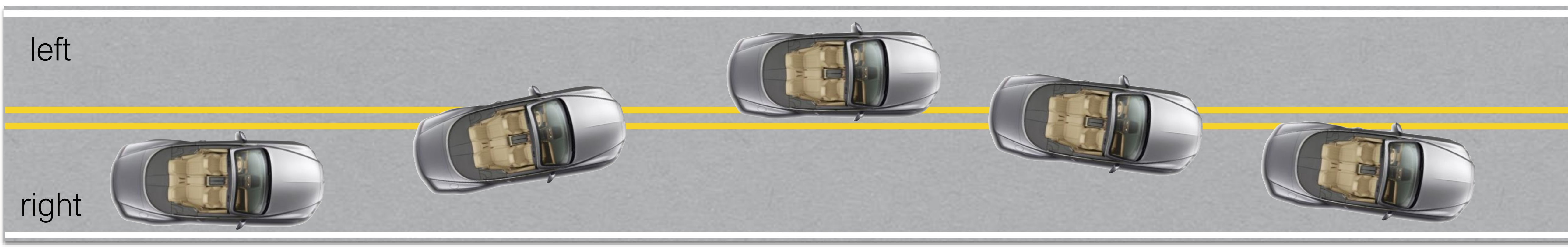
Can be computed with recursion (Dynamic Programming)

$$\underline{P(\boldsymbol{X}_{t+1}|\boldsymbol{e}_{1:t+1})} \propto P(\boldsymbol{e}_{t+1}|\boldsymbol{X}_{t+1}) \sum_{\boldsymbol{X}_t} P(\boldsymbol{X}_{t+1}|\boldsymbol{X}_t)\underline{P(\boldsymbol{X}_t|\boldsymbol{e}_{1:t})}$$

$$\begin{aligned}
P(\boldsymbol{X}_{t+1}|\boldsymbol{e}_{1:t+1}) &= P(\boldsymbol{X}_{t+1}|\boldsymbol{e}_{t+1}, \boldsymbol{e}_{1:t}) \\
&= \frac{P(\boldsymbol{e}_{t+1}|\boldsymbol{X}_{t+1}, \boldsymbol{e}_{1:t})P(\boldsymbol{X}_{t+1}|\boldsymbol{e}_{1:t})}{P(\boldsymbol{e}_{t+1}|\boldsymbol{e}_{1:t})} \\
&= \alpha P(\boldsymbol{e}_{t+1}|\boldsymbol{X}_{t+1})P(\boldsymbol{X}_{t+1}|\boldsymbol{e}_{1:t}) \\
&= \alpha P(\boldsymbol{e}_{t+1}|\boldsymbol{X}_{t+1}) \sum_{\boldsymbol{X}_t} P(\boldsymbol{X}_{t+1}|\boldsymbol{X}_t, \boldsymbol{e}_{1:t})P(\boldsymbol{X}_t|\boldsymbol{e}_{1:t}) \\
&= \alpha P(\boldsymbol{e}_{t+1}|\boldsymbol{X}_{t+1}) \sum_{\boldsymbol{X}_t} P(\boldsymbol{X}_{t+1}|\boldsymbol{X}_t)P(\boldsymbol{X}_t|\boldsymbol{e}_{1:t})
\end{aligned}$$

Hidden Markov Model example

'In the trunk of a car of a sleepy driver' model

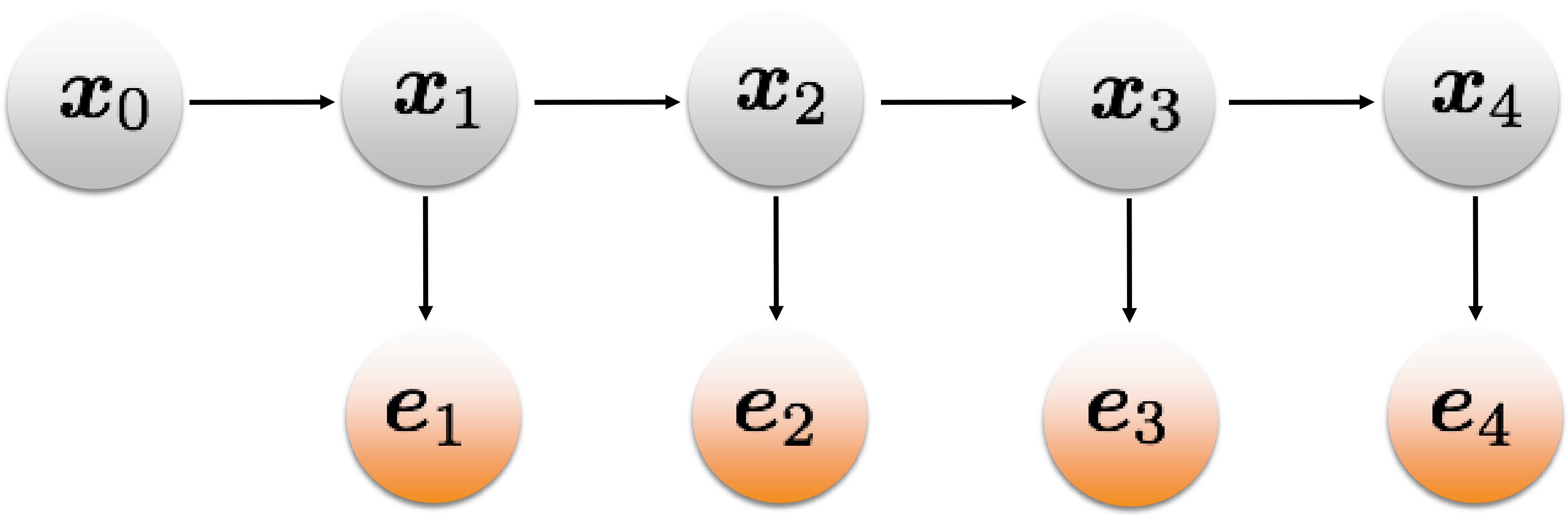


binary random variable (left lane or right lane)

$$\boldsymbol{x}_0 \rightarrow \boldsymbol{x}_1 \rightarrow \boldsymbol{x}_2 \rightarrow \boldsymbol{x}_3 \rightarrow \boldsymbol{x}_4$$

$$\boldsymbol{x} = \{x_{\text{left}}, x_{\text{right}}\}$$

From a hole in the car you can see the ground



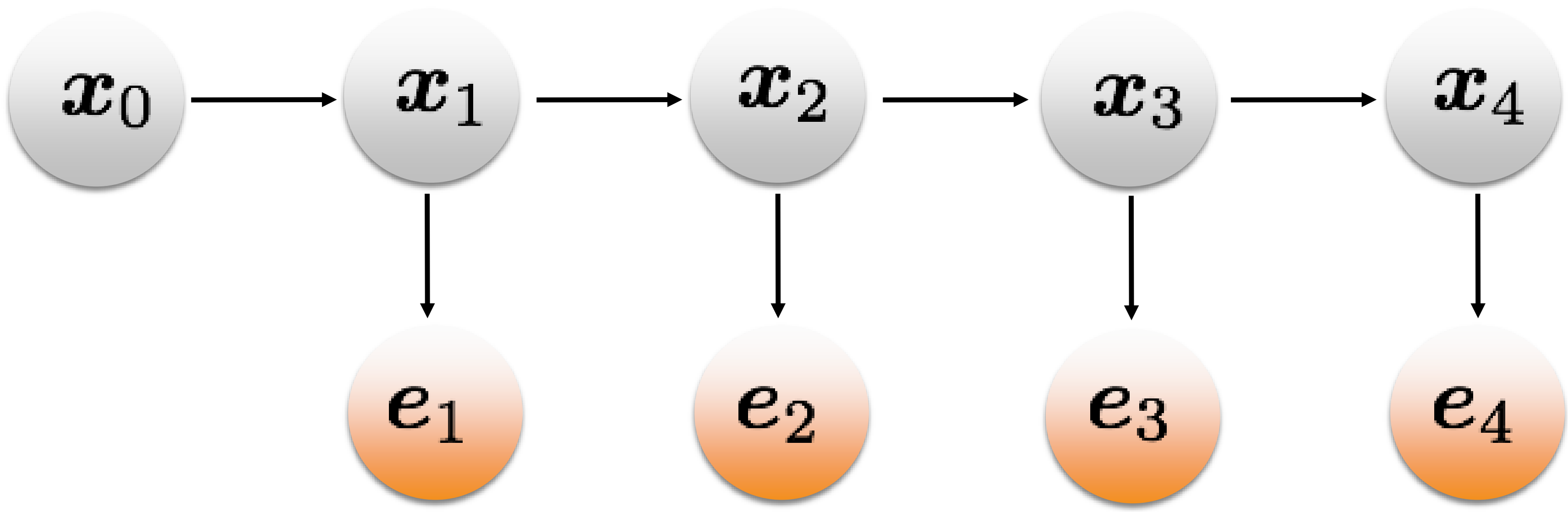
binary random variable (center lane is yellow or road is gray)

$$\boldsymbol{e} = \{e_{\text{gray}}, e_{\text{yellow}}\}$$

	x_{left}	x_{right}
$P(\boldsymbol{x}_0)$	0.5	0.5

$P(\boldsymbol{x}_t \mid \boldsymbol{x}_{t-1})$	x_{left}	x_{right}
x_{left}	0.7	0.3
x_{right}	0.3	0.7

What needs to sum to one?



$P(\boldsymbol{e}_t \mid \boldsymbol{x}_t)$	x_{left}	x_{right}
e_{yellow}	0.9	0.2
e_{gray}	0.1	0.8

This is filtering!

What's the probability of being in the left lane at t=4?

$P(\boldsymbol{x}_0)$	x_{left}	x_{right}
	0.5	0.5

$P(\boldsymbol{x}_t \vert \boldsymbol{x}_{t-1})$	x_{left}	x_{right}
x_{left}	0.7	0.3
x_{right}	0.3	0.7

$P(e_t \vert \boldsymbol{x}_t)$	x_{left}	x_{right}
e_{yellow}	0.9	0.2
e_{gray}	0.1	0.8

Filtering: $P(\boldsymbol{X}_{t+1}|\boldsymbol{e}_{1:t+1}) \propto P(\boldsymbol{e}_{t+1}|\boldsymbol{X}_{t+1}) \sum_{\boldsymbol{X}_t} P(\boldsymbol{X}_{t+1}|\boldsymbol{X}_t) P(\boldsymbol{X}_t|\boldsymbol{e}_{1:t})$

*What is the belief distribution if I see **yellow** at t=1* $p(\boldsymbol{x}_1|\boldsymbol{e}_1 = e_{\text{yellow}}) =?$

Prediction step: $p(\boldsymbol{x}_1) = \sum_{\boldsymbol{x}_0} p(\boldsymbol{x}_1|\boldsymbol{x}_0)p(\boldsymbol{x}_0)$

Update step: $p(\boldsymbol{x}_1|\boldsymbol{e}_1) = \alpha\, p(\boldsymbol{e}_1|\boldsymbol{x}_1)p(\boldsymbol{x}_1)$

$P(\boldsymbol{x}_0)$	x_{left}	x_{right}
	0.5	0.5

$P(\boldsymbol{x}_t \mid \boldsymbol{x}_{t-1})$	x_{left}	x_{right}
x_{left}	0.7	0.3
x_{right}	0.3	0.7

$P(e_t \mid \boldsymbol{x}_t)$	x_{left}	x_{right}
e_{yellow}	0.9	0.2
e_{gray}	0.1	0.8

Filtering: $$P(\boldsymbol{X}_{t+1} \mid \boldsymbol{e}_{1:t+1}) \propto P(\boldsymbol{e}_{t+1} \mid \boldsymbol{X}_{t+1}) \sum_{\boldsymbol{X}_t} P(\boldsymbol{X}_{t+1} \mid \boldsymbol{X}_t) P(\boldsymbol{X}_t \mid \boldsymbol{e}_{1:t})$$

What is the belief distribution if I see ***yellow*** *at t=1* $p(\boldsymbol{x}_1 \mid \boldsymbol{e}_1 = e_{\text{yellow}}) = ?$

Prediction step:

$$p(\boldsymbol{x}_1) = \sum_{\boldsymbol{x}_0} p(\boldsymbol{x}_1 \mid \boldsymbol{x}_0) p(\boldsymbol{x}_0)$$

$$= [0.7 \quad 0.3](0.5) + [0.3 \quad 0.7](0.5)$$

$$= \begin{bmatrix} 0.7 & 0.3 \\ 0.3 & 0.7 \end{bmatrix} \begin{bmatrix} 0.5 \\ 0.5 \end{bmatrix} = \begin{bmatrix} 0.5 \\ 0.5 \end{bmatrix}$$

$P(\boldsymbol{x}_0)$	x_{left}	x_{right}
	0.5	0.5

$P(\boldsymbol{x}_t \mid \boldsymbol{x}_{t-1})$	x_{left}	x_{right}
x_{left}	0.7	0.3
x_{right}	0.3	0.7

$P(\boldsymbol{e}_t \mid \boldsymbol{x}_t)$	x_{left}	x_{right}
e_{yellow}	0.9	0.2
e_{gray}	0.1	0.8

Filtering: $$P(\boldsymbol{X}_{t+1} \mid \boldsymbol{e}_{1:t+1}) \propto P(\boldsymbol{e}_{t+1} \mid \boldsymbol{X}_{t+1}) \sum_{\boldsymbol{X}_t} P(\boldsymbol{X}_{t+1} \mid \boldsymbol{X}_t) P(\boldsymbol{X}_t \mid \boldsymbol{e}_{1:t})$$

*What is the belief distribution if I see **yellow** at t=1* $p(\boldsymbol{x}_1 \mid \boldsymbol{e}_1 = e_{\text{yellow}}) = ?$

Update step: $p(\boldsymbol{x}_1 \mid \boldsymbol{e}_1) = \alpha \, p(\boldsymbol{e}_1 \mid \boldsymbol{x}_1) p(\boldsymbol{x}_1)$

$P(\boldsymbol{x}_0)$	x_{left}	x_{right}
	0.5	0.5

$P(\boldsymbol{x}_t \mid \boldsymbol{x}_{t-1})$	x_{left}	x_{right}
x_{left}	0.7	0.3
x_{right}	0.3	0.7

$P(\boldsymbol{e}_t \mid \boldsymbol{x}_t)$	x_{left}	x_{right}
e_{yellow}	0.9	0.2
e_{gray}	0.1	0.8

Filtering: $$P(\boldsymbol{X}_{t+1} \mid \boldsymbol{e}_{1:t+1}) \propto P(\boldsymbol{e}_{t+1} \mid \boldsymbol{X}_{t+1}) \sum_{\boldsymbol{X}_t} P(\boldsymbol{X}_{t+1} \mid \boldsymbol{X}_t) P(\boldsymbol{X}_t \mid \boldsymbol{e}_{1:t})$$

What is the belief distribution if I see ***yellow*** *at t=1* $p(\boldsymbol{x}_1 \mid \boldsymbol{e}_1 = e_{\text{yellow}}) = ?$

Update step: $$p(\boldsymbol{x}_1 \mid \boldsymbol{e}_1) = \alpha\, p(\boldsymbol{e}_1 \mid \boldsymbol{x}_1) p(\boldsymbol{x}_1)$$

$$= \alpha\, (0.9 \quad 0.2) .* (0.5 \quad 0.5)$$

observed yellow

$$= \alpha \begin{bmatrix} 0.9 & 0.0 \\ 0.0 & 0.2 \end{bmatrix} \begin{bmatrix} 0.5 \\ 0.5 \end{bmatrix} = \begin{bmatrix} 0.45 \\ 0.1 \end{bmatrix}$$

$$\approx \begin{bmatrix} 0.818 \\ 0.182 \end{bmatrix}$$

more likely to be in which lane?

$P(\boldsymbol{x}_0)$	x_{left}	x_{right}
	0.5	0.5

$P(\boldsymbol{x}_t \mid \boldsymbol{x}_{t-1})$	x_{left}	x_{right}
x_{left}	0.7	0.3
x_{right}	0.3	0.7

$P(\boldsymbol{e}_t \mid \boldsymbol{x}_t)$	x_{left}	x_{right}
e_{yellow}	0.9	0.2
e_{gray}	0.1	0.8

Filtering: $$P(\boldsymbol{X}_{t+1} \mid \boldsymbol{e}_{1:t+1}) \propto P(\boldsymbol{e}_{t+1} \mid \boldsymbol{X}_{t+1}) \sum_{\boldsymbol{X}_t} P(\boldsymbol{X}_{t+1} \mid \boldsymbol{X}_t) P(\boldsymbol{X}_t \mid \boldsymbol{e}_{1:t})$$

What is the belief distribution if I see ***yellow*** *at t=1* $p(\boldsymbol{x}_1 \mid \boldsymbol{e}_1 = e_{\text{yellow}}) = ?$

Summary

Prediction step: $$p(\boldsymbol{x}_1) = \sum_{\boldsymbol{x}_0} p(\boldsymbol{x}_1 \mid \boldsymbol{x}_0) p(\boldsymbol{x}_0)$$

$$= \begin{bmatrix} 0.5 \\ 0.5 \end{bmatrix}$$

Update step: $$p(\boldsymbol{x}_1 \mid \boldsymbol{e}_1) = \alpha \, p(\boldsymbol{e}_1 \mid \boldsymbol{x}_1) p(\boldsymbol{x}_1)$$

$$\approx \begin{bmatrix} 0.818 \\ 0.182 \end{bmatrix}$$

$P(\boldsymbol{x}_0)$	x_{left}	x_{right}
	0.5	0.5

$P(\boldsymbol{x}_t \mid \boldsymbol{x}_{t-1})$	x_{left}	x_{right}
x_{left}	0.7	0.3
x_{right}	0.3	0.7

$P(e_t \mid \boldsymbol{x}_t)$	x_{left}	x_{right}
e_{yellow}	0.9	0.2
e_{gray}	0.1	0.8

Filtering: $$P(\boldsymbol{X}_{t+1} \mid \boldsymbol{e}_{1:t+1}) \propto P(\boldsymbol{e}_{t+1} \mid \boldsymbol{X}_{t+1}) \sum_{\boldsymbol{X}_t} P(\boldsymbol{X}_{t+1} \mid \boldsymbol{X}_t) P(\boldsymbol{X}_t \mid \boldsymbol{e}_{1:t})$$

*What if you see **yellow** again at **t=2*** $p(\boldsymbol{x}_2 \mid e_1, e_2) = ?$

$P(\boldsymbol{x}_0)$	x_{left}	x_{right}
	0.5	0.5

$P(\boldsymbol{x}_t \vert \boldsymbol{x}_{t-1})$	x_{left}	x_{right}
x_{left}	0.7	0.3
x_{right}	0.3	0.7

$P(\boldsymbol{e}_t \vert \boldsymbol{x}_t)$	x_{left}	x_{right}
e_{yellow}	0.9	0.2
e_{gray}	0.1	0.8

Filtering: $$P(\boldsymbol{X}_{t+1}|\boldsymbol{e}_{1:t+1}) \propto P(\boldsymbol{e}_{t+1}|\boldsymbol{X}_{t+1}) \sum_{\boldsymbol{X}_t} P(\boldsymbol{X}_{t+1}|\boldsymbol{X}_t) P(\boldsymbol{X}_t|\boldsymbol{e}_{1:t})$$

*What if you see **yellow** again at **t=2*** $p(\boldsymbol{x}_2|e_1, e_2) = ?$

Prediction step: $$p(\boldsymbol{x}_2|e_1) = \sum_{\boldsymbol{x}_1} p(\boldsymbol{x}_2|\boldsymbol{x}_1) p(\boldsymbol{x}_1|e_1)$$

Update step: $$p(\boldsymbol{x}_1|e_1, e_2) = \alpha\, p(\boldsymbol{e}_1|\boldsymbol{x}_1) p(\boldsymbol{x}_1)$$

$P(\boldsymbol{x}_0)$	x_{left}	x_{right}
	0.5	0.5

$P(\boldsymbol{x}_t \mid \boldsymbol{x}_{t-1})$	x_{left}	x_{right}
x_{left}	0.7	0.3
x_{right}	0.3	0.7

$P(e_t \mid \boldsymbol{x}_t)$	x_{left}	x_{right}
e_{yellow}	0.9	0.2
e_{gray}	0.1	0.8

Filtering: $P(\boldsymbol{X}_{t+1} \mid \boldsymbol{e}_{1:t+1}) \propto P(\boldsymbol{e}_{t+1} \mid \boldsymbol{X}_{t+1}) \sum_{\boldsymbol{X}_t} P(\boldsymbol{X}_{t+1} \mid \boldsymbol{X}_t) P(\boldsymbol{X}_t \mid \boldsymbol{e}_{1:t})$

What if you see ***yellow*** *again at* ***t=2*** $p(\boldsymbol{x}_2 \mid e_1, e_2) = ?$

Prediction step: $p(\boldsymbol{x}_2 \mid e_1) = \sum_{\boldsymbol{x}_1} p(\boldsymbol{x}_2 \mid \boldsymbol{x}_1) p(\boldsymbol{x}_1 \mid e_1)$

$$= \begin{bmatrix} 0.7 & 0.3 \\ 0.3 & 0.7 \end{bmatrix} \begin{bmatrix} 0.818 \\ 0.182 \end{bmatrix} = \begin{bmatrix} 0.627 \\ 0.373 \end{bmatrix}$$

Why does the probability of being in the left lane go down?

$P(\boldsymbol{x}_0)$	x_{left}	x_{right}
	0.5	0.5

$P(\boldsymbol{x}_t \mid \boldsymbol{x}_{t-1})$	x_{left}	x_{right}
x_{left}	0.7	0.3
x_{right}	0.3	0.7

$P(e_t \mid \boldsymbol{x}_t)$	x_{left}	x_{right}
e_{yellow}	0.9	0.2
e_{gray}	0.1	0.8

Filering: $P(\boldsymbol{X}_{t+1} \mid \boldsymbol{e}_{1:t+1}) \propto P(\boldsymbol{e}_{t+1} \mid \boldsymbol{X}_{t+1}) \sum_{\boldsymbol{X}_t} P(\boldsymbol{X}_{t+1} \mid \boldsymbol{X}_t) P(\boldsymbol{X}_t \mid \boldsymbol{e}_{1:t})$

*What if you see **yellow** again at **t=2*** $p(\boldsymbol{x}_2 \mid e_1, e_2) = ?$

Update step:
$$
\begin{aligned}
p(\boldsymbol{x}_2 \mid e_1, e_2) &= \alpha \; p(e_2 \mid \boldsymbol{x}_2) p(\boldsymbol{x}_2 \mid e_1) \\
&= \alpha \begin{bmatrix} 0.9 & 0.0 \\ 0.0 & 0.2 \end{bmatrix} \begin{bmatrix} 0.627 \\ 0.373 \end{bmatrix} \\
&\approx \begin{bmatrix} 0.883 \\ 0.117 \end{bmatrix}
\end{aligned}
$$

Basic Inference Tasks

Filtering

$$P(\boldsymbol{X}_t|\boldsymbol{e}_{1:t})$$

Posterior probability over the **current** state, given all evidence up to present

Prediction

$$P(\boldsymbol{X}_{t+k}|\boldsymbol{e}_{1:t})$$

Posterior probability over a **future** state, given all evidence up to present

Smoothing

$$P(\boldsymbol{X}_k|\boldsymbol{e}_{1:t})$$

Posterior probability over a **past** state, given all evidence up to present

Best Sequence

$$\arg\max_{\boldsymbol{X}_{1:t}} P(\boldsymbol{X}_{1:t}|\boldsymbol{e}_{1:t})$$

Best state sequence given all evidence up to present

Prediction

$$P(\mathbf{X}_{t+k}|\mathbf{e}_{1:t})$$

Posterior probability over a **future** state, given all evidence up to present

Where am I going?

Prediction

same recursive form as filtering but…

$$P(\boldsymbol{X}_{t+k+1}|\boldsymbol{e}_{1:t}) = \sum_{\boldsymbol{x}_{t+k}} P(\boldsymbol{X}_{t+k+1}|\boldsymbol{x}_{t+k})P(\boldsymbol{x}_{t+k}|\boldsymbol{e}_{1:t})$$

no new evidence!

What happens as you try to predict further into the future?

Prediction

$$P(\boldsymbol{X}_{t+k+1}|\boldsymbol{e}_{1:t}) = \sum_{\boldsymbol{x}_{t+k}} P(\boldsymbol{X}_{t+k+1}|\boldsymbol{x}_{t+k})P(\boldsymbol{x}_{t+k}|\boldsymbol{e}_{1:t})$$

no new evidence

What happens as you try to predict further into the future?

Approaches its ‘stationary distribution’

Basic Inference Tasks

Filtering

$$P(\boldsymbol{X}_t|\boldsymbol{e}_{1:t})$$

Posterior probability over the **current** state, given all evidence up to present

Prediction

$$P(\boldsymbol{X}_{t+k}|\boldsymbol{e}_{1:t})$$

Posterior probability over a **future** state, given all evidence up to present

Smoothing

$$P(\boldsymbol{X}_k|\boldsymbol{e}_{1:t})$$

Posterior probability over a **past** state, given all evidence up to present

Best Sequence

$$\arg\max_{\boldsymbol{X}_{1:t}} P(\boldsymbol{X}_{1:t}|\boldsymbol{e}_{1:t})$$

Best state sequence given all evidence up to present

Smoothing

$$P(\mathbf{X}_k|\mathbf{e}_{1:t})$$

Posterior probability over a **past** state, given all evidence up to present

Wait, what did I do yesterday?

Smoothing

some time in the past

$$P(\boldsymbol{X}_k | \boldsymbol{e}_{1:t}) \qquad 1 \leq k < t$$

$$\begin{aligned} P(\boldsymbol{X}_k | \boldsymbol{e}_{1:t}) &= P(\boldsymbol{X}_k | \boldsymbol{e}_{1:k}, \boldsymbol{e}_{k+1:t}) \\ &= \alpha P(\boldsymbol{X}_k | \boldsymbol{e}_{1:k}) P(\boldsymbol{e}_{k+1:t} | \boldsymbol{X}_k, \boldsymbol{e}_{1:k}) \\ &= \alpha P(\boldsymbol{X}_k | \boldsymbol{e}_{1:k}) P(\boldsymbol{e}_{k+1:t} | \boldsymbol{X}_k) \end{aligned}$$

'forward' message

'backward' message

this is just filtering

this is backwards filtering
Let me explain…

Backward message

$$P(\boldsymbol{e}_{k+1:t}|\boldsymbol{X}_k) = \sum_{\boldsymbol{x}_{k+1}} P(\boldsymbol{e}_{k+1:t}|\boldsymbol{X}_k, \boldsymbol{x}_{k+1})P(\boldsymbol{x}_{k+1}|\boldsymbol{X}_k)$$

conditioning

copied from last slide

Backward message

$$P(\boldsymbol{e}_{k+1:t}|\boldsymbol{X}_k) = \sum_{\boldsymbol{x}_{k+1}} P(\boldsymbol{e}_{k+1:t}|\boldsymbol{X}_k, \boldsymbol{x}_{k+1}) P(\boldsymbol{x}_{k+1}|\boldsymbol{X}_k)$$

conditioning

copied from last slide

$$= \sum_{\boldsymbol{x}_{k+1}} P(\boldsymbol{e}_{k+1:t}|\boldsymbol{x}_{k+1}) P(\boldsymbol{x}_{k+1}|\boldsymbol{X}_k)$$

Markov Assumption

Backward message

$$P(\boldsymbol{e}_{k+1:t}|\boldsymbol{X}_k) = \sum_{\boldsymbol{x}_{k+1}} P(\boldsymbol{e}_{k+1:t}|\boldsymbol{X}_k, \boldsymbol{x}_{k+1}) P(\boldsymbol{x}_{k+1}|\boldsymbol{X}_k)$$ conditioning

copied from last slide

$$= \sum_{\boldsymbol{x}_{k+1}} P(\boldsymbol{e}_{k+1:t}|\boldsymbol{x}_{k+1}) P(\boldsymbol{x}_{k+1}|\boldsymbol{X}_k)$$ Markov Assumption

$$= \sum_{\boldsymbol{x}_{k+1}} P(\boldsymbol{e}_{k+1}, \boldsymbol{e}_{k+2:t}|\boldsymbol{x}_{k+1}) P(\boldsymbol{x}_{k+1}|\boldsymbol{X}_k)$$ split

Backward message

$$P(\mathbf{e}_{k+1:t}|\boldsymbol{X}_k) = \sum_{\boldsymbol{x}_{k+1}} P(\mathbf{e}_{k+1:t}|\boldsymbol{X}_k, \boldsymbol{x}_{k+1})P(\boldsymbol{x}_{k+1}|\boldsymbol{X}_k)$$

conditioning

copied from last slide

$$= \sum_{\boldsymbol{x}_{k+1}} P(\mathbf{e}_{k+1:t}|\boldsymbol{x}_{k+1})P(\boldsymbol{x}_{k+1}|\boldsymbol{X}_k)$$

Markov Assumption

$$= \sum_{\boldsymbol{x}_{k+1}} P(\mathbf{e}_{k+1}, \mathbf{e}_{k+2:t}|\boldsymbol{x}_{k+1})P(\boldsymbol{x}_{k+1}|\boldsymbol{X}_k)$$

split

$$= \sum_{\boldsymbol{x}_{k+1}} \underbrace{P(\mathbf{e}_{k+1}|\boldsymbol{x}_{k+1})}_{\text{observation model}} \underbrace{P(\mathbf{e}_{k+2:t}|\boldsymbol{x}_{k+1})}_{\text{recursive message}} \underbrace{P(\boldsymbol{x}_{k+1}|\boldsymbol{X}_k)}_{\text{motion model}}$$

This is just a 'backwards' version of filtering where

initial message $P(\mathbf{e}_{t-1:t}|\boldsymbol{X}_t) = \mathbf{1}$

Basic Inference Tasks

Filtering

$$P(\boldsymbol{X}_t|\mathbf{e}_{1:t})$$

Posterior probability over the **current** state, given all evidence up to present

Prediction

$$P(\boldsymbol{X}_{t+k}|\mathbf{e}_{1:t})$$

Posterior probability over a **future** state, given all evidence up to present

Smoothing

$$P(\boldsymbol{X}_k|\mathbf{e}_{1:t})$$

Posterior probability over a **past** state, given all evidence up to present

Best Sequence

$$\arg\max_{\boldsymbol{X}_{1:t}} P(\boldsymbol{X}_{1:t}|\mathbf{e}_{1:t})$$

Best state sequence given all evidence up to present

Best Sequence

$$\arg\max_{\boldsymbol{X}_{1:t}} P(\boldsymbol{X}_{1:t}|\boldsymbol{e}_{1:t})$$

Best state sequence given all evidence up to present

I must have done something right, right?

Best Sequence

‘Viterbi Algorithm’

$$\max_{\boldsymbol{x}_1,\ldots,\boldsymbol{x}_t} P(\boldsymbol{x}_1,\ldots,\boldsymbol{x}_t,\boldsymbol{X}_{t+1}|\mathbf{e}_{1:t+1})$$

$$= \alpha P(\mathbf{e}_{t+1}|\boldsymbol{X}_{t+1}) \max_{\boldsymbol{x}_t}\left[P(\boldsymbol{X}_{t+1}|\boldsymbol{x}_t) \underbrace{\max_{\boldsymbol{x}_1,\ldots,\boldsymbol{x}_{t-1}} P(\boldsymbol{x}_1,\ldots,\boldsymbol{x}_{t-1},\boldsymbol{X}_t|\mathbf{e}_{1:t})}_{\text{recursive message}}\right]$$

Identical to filtering but with a max operator

Recall: Filtering equation

$$P(\boldsymbol{X}_{t+1}|\mathbf{e}_{1:t+1}) \propto P(\mathbf{e}_{t+1}|\boldsymbol{X}_{t+1}) \sum_{\boldsymbol{X}_t} P(\boldsymbol{X}_{t+1}|\boldsymbol{X}_t) \underbrace{P(\boldsymbol{X}_t|\mathbf{e}_{1:t})}_{\text{recursive message}}$$

Now you know how to answer all the important questions in life:

Where am I now?

Where am I going?

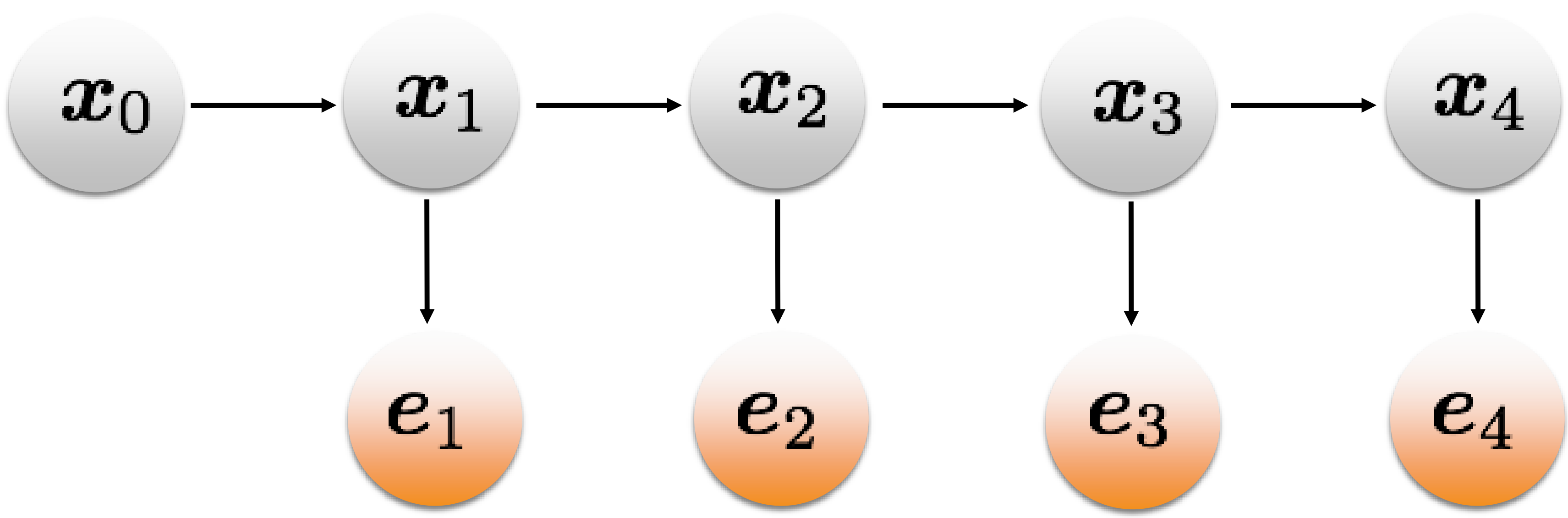
Wait, what did I do yesterday?

I must have done something right, right?

Kalman filtering

Examples up to now have been **discrete** (binary) random variables

Kalman 'filtering' can be seen as a special case of a temporal inference with continuous random variables



Everything is continuous…

$\boldsymbol{x}$ $\quad$ $\boldsymbol{e}$ $\quad$ $P(\boldsymbol{x}_0)$ $\quad$ $P(\boldsymbol{e}|\boldsymbol{x})$ $\quad$ $P(\boldsymbol{x}_t|\boldsymbol{x}_{t-1})$

probability distributions are no longer tables but functions

Making the connection to the 'filtering' equations

(Discrete) Filtering

Tables Tables Tables

$$P(\boldsymbol{X}_{t+1}|\boldsymbol{e}_{1:t+1}) \propto P(\boldsymbol{e}_{t+1}|\boldsymbol{X}_{t+1}) \sum_{\boldsymbol{X}_t} P(\boldsymbol{X}_{t+1}|\boldsymbol{X}_t) P(\boldsymbol{X}_t|\boldsymbol{e}_{1:t})$$

Kalman Filtering

Gaussian Gaussian Gaussian

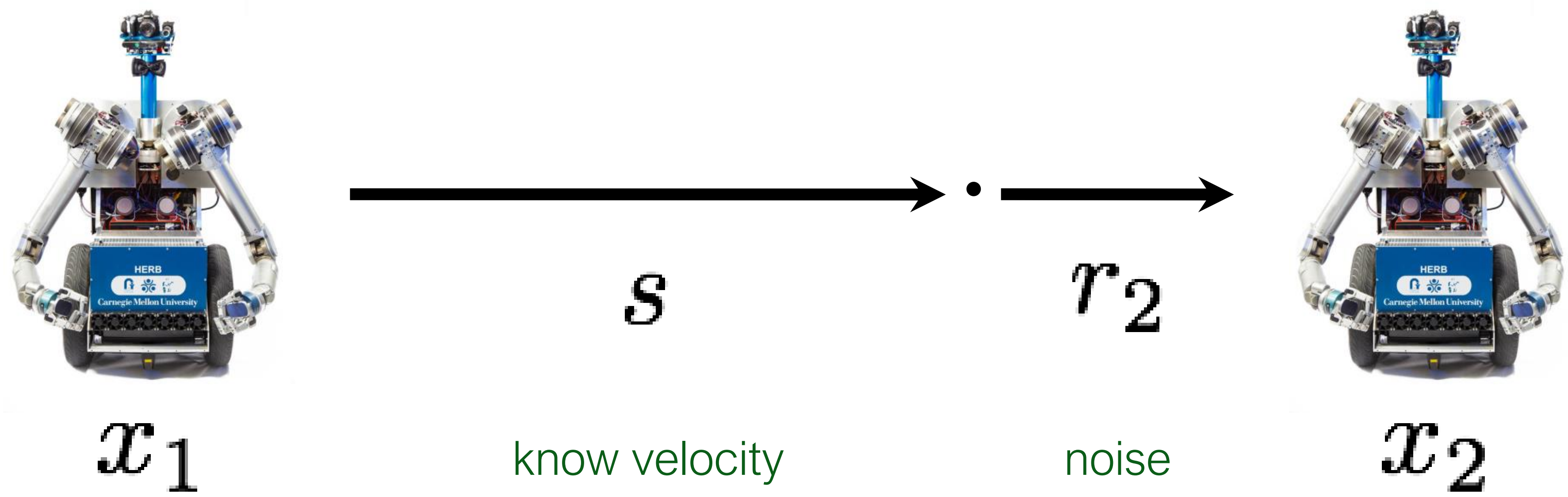
$$P(\boldsymbol{X}_{t+1}|\boldsymbol{e}_{1:t+1}) \propto P(\boldsymbol{e}_{t+1}|\boldsymbol{X}_{t+1}) \int_{\boldsymbol{x}_t} P(\boldsymbol{X}_{t+1}|\boldsymbol{x}_t) P(\boldsymbol{x}_t|\boldsymbol{e}_{1:t}) d\boldsymbol{x}_t$$

observation model motion model belief

integral because
continuous PDFs

Simple, 1D example…

x



$$x_t = x_{t-1} + s + r_t$$

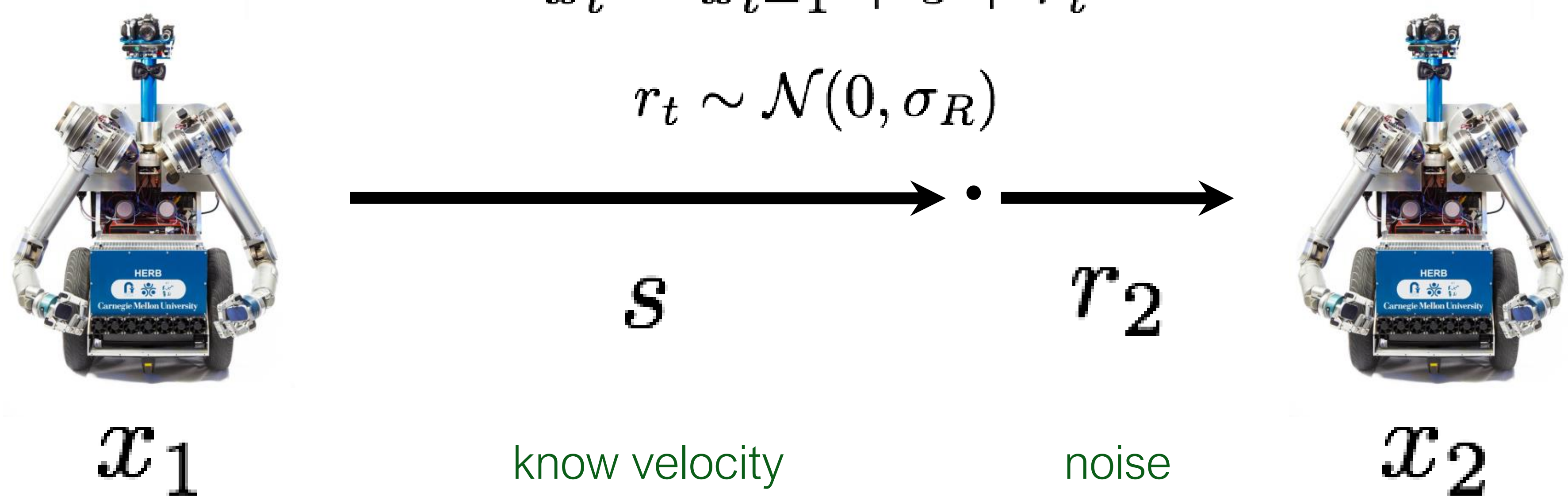
$$r_t \sim \mathcal{N}(0, \sigma_R)$$

'sampled from'

System (motion) model

$$x_t = x_{t-1} + s + r_t$$

$$r_t \sim \mathcal{N}(0, \sigma_R)$$

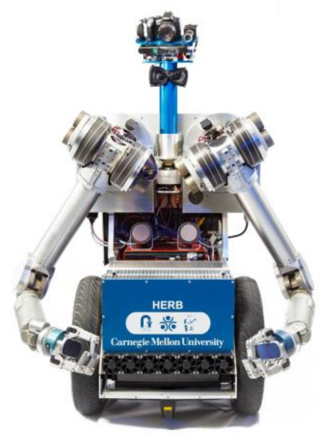


How do you represent the motion model?

$$P(x_t | x_{t-1})$$

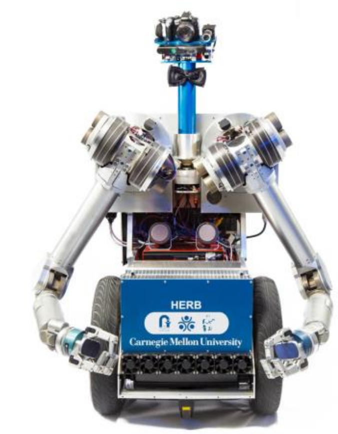
$$x_t = x_{t-1} + s + r_t$$

$$r_t \sim \mathcal{N}(0, \sigma_R)$$



s r_2

x_1 know velocity noise x_2



How do you represent the motion model?

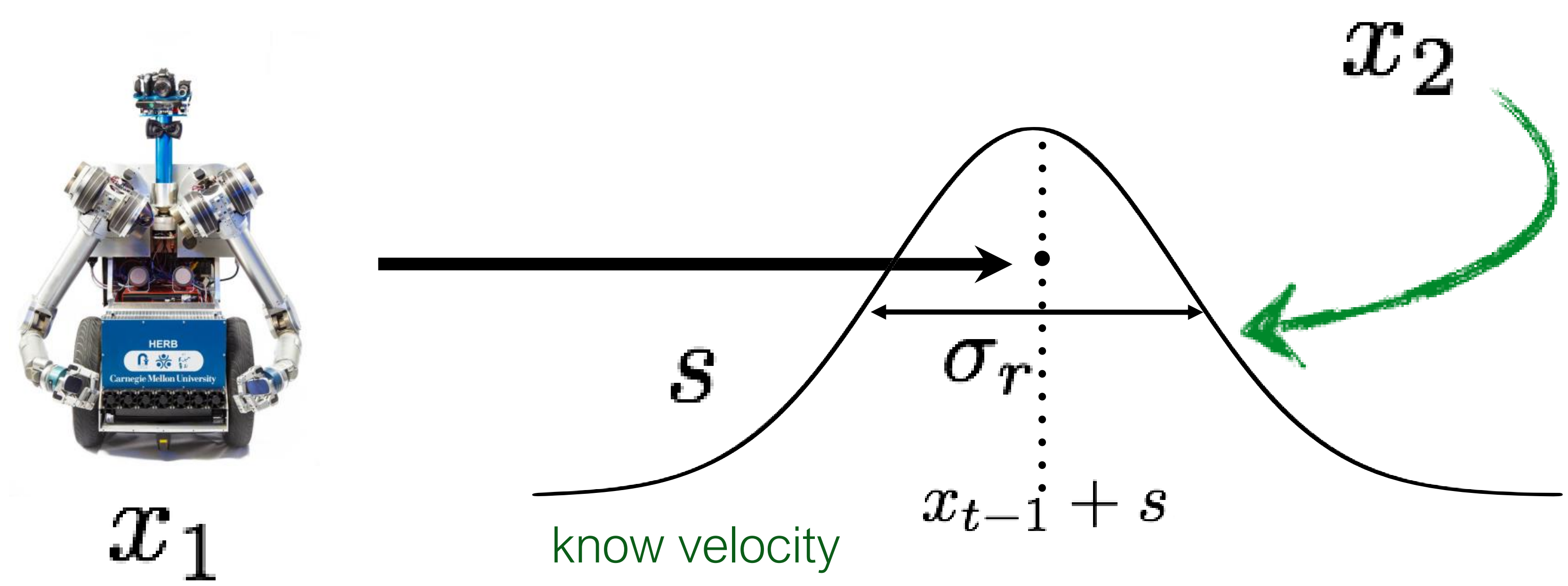
A linear Gaussian (continuous) transition model

$$P(x_t | x_{t-1}) = \mathcal{N}(x_t; x_{t-1} + s, \sigma_r)$$

mean

standard deviation

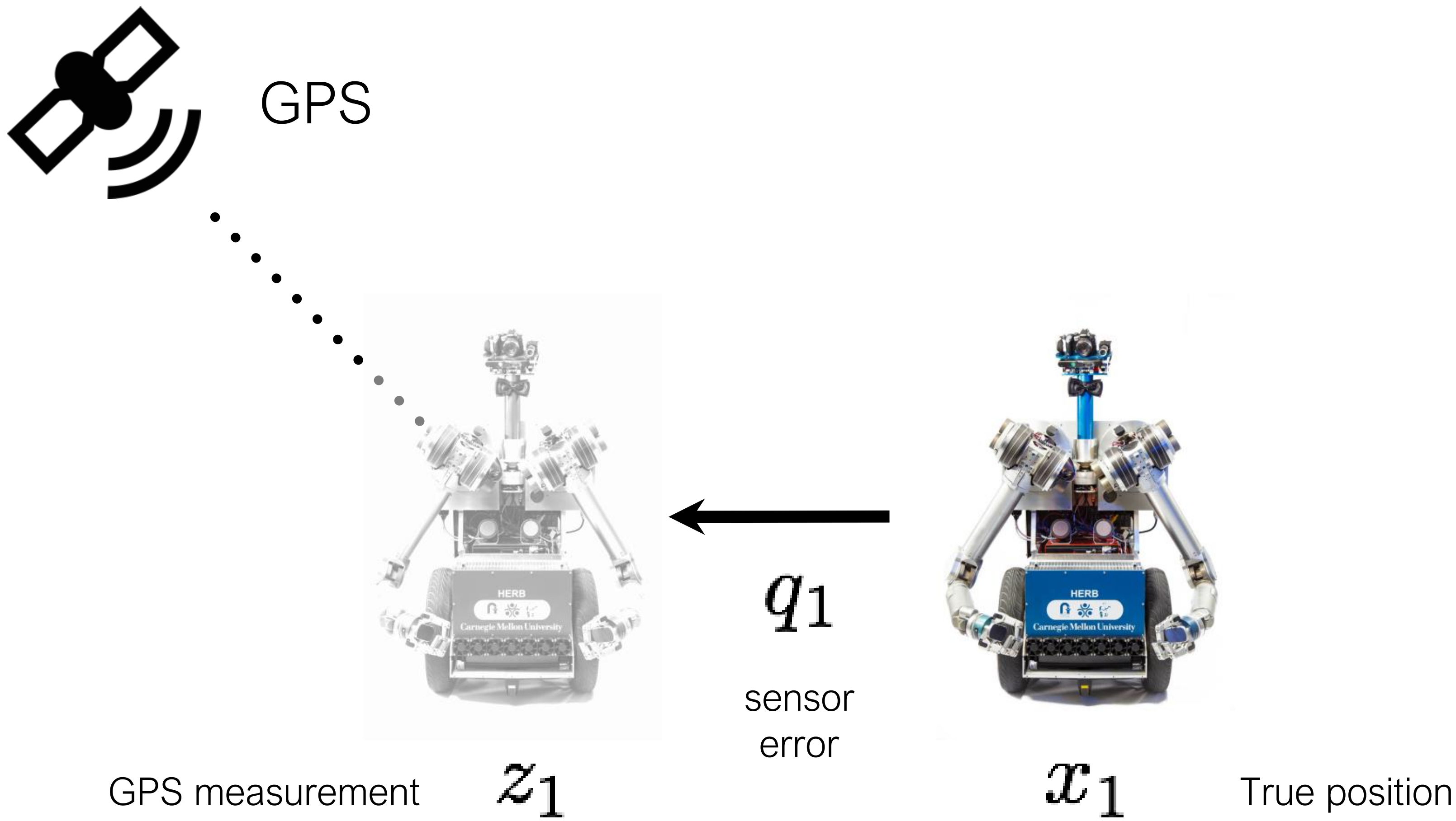
How can you visualize this distribution?



A linear Gaussian (continuous) transition model

$$P(x_t|x_{t-1}) = \mathcal{N}(x_t; x_{t-1} + s, \sigma_r)$$

Why don't we just use a table as before?

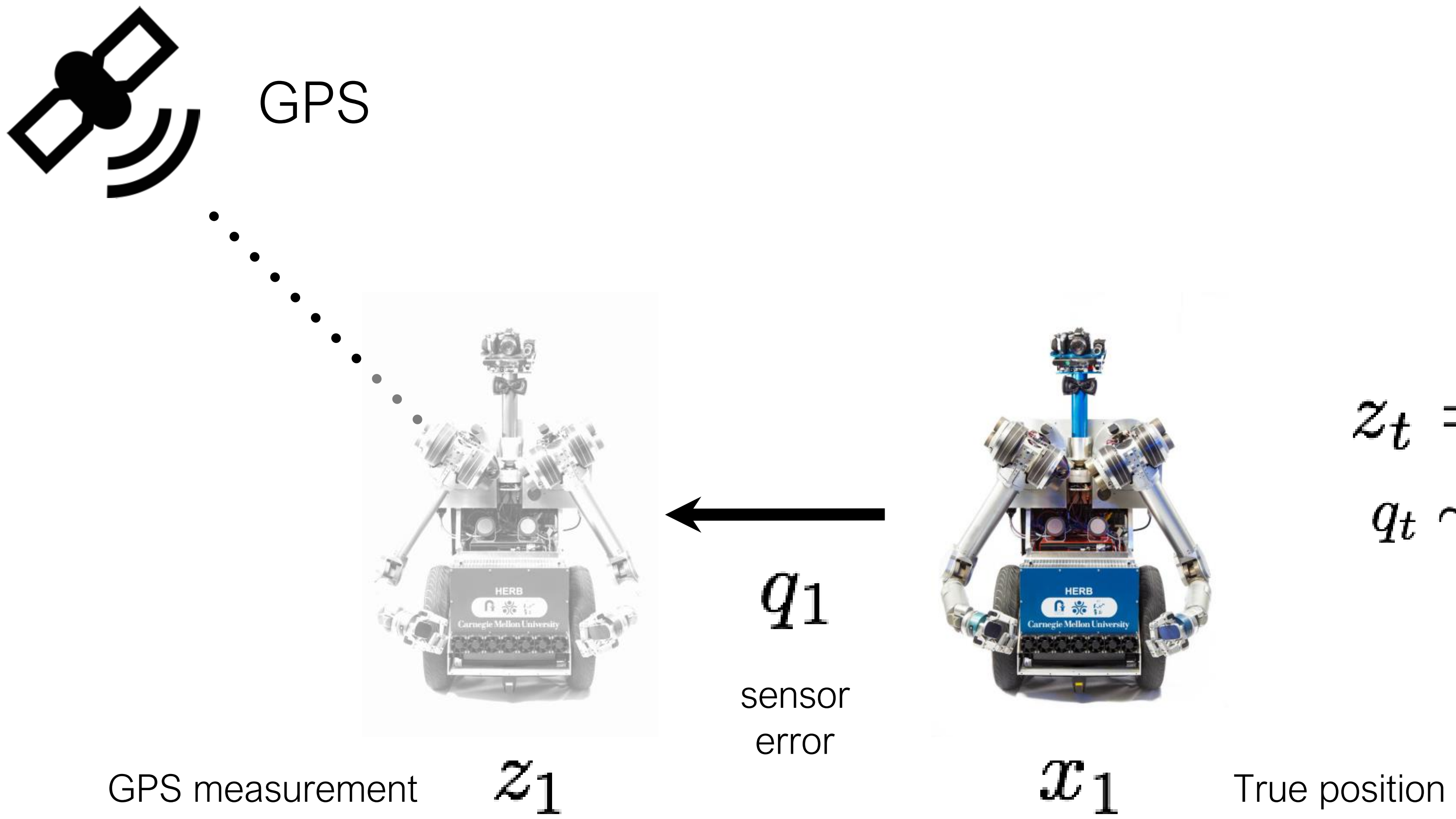


$$z_t = x_t + q_t$$

$$q_t \sim \mathcal{N}(0, \sigma_Q)$$

sampled from a Gaussian

Observation (measurement) model

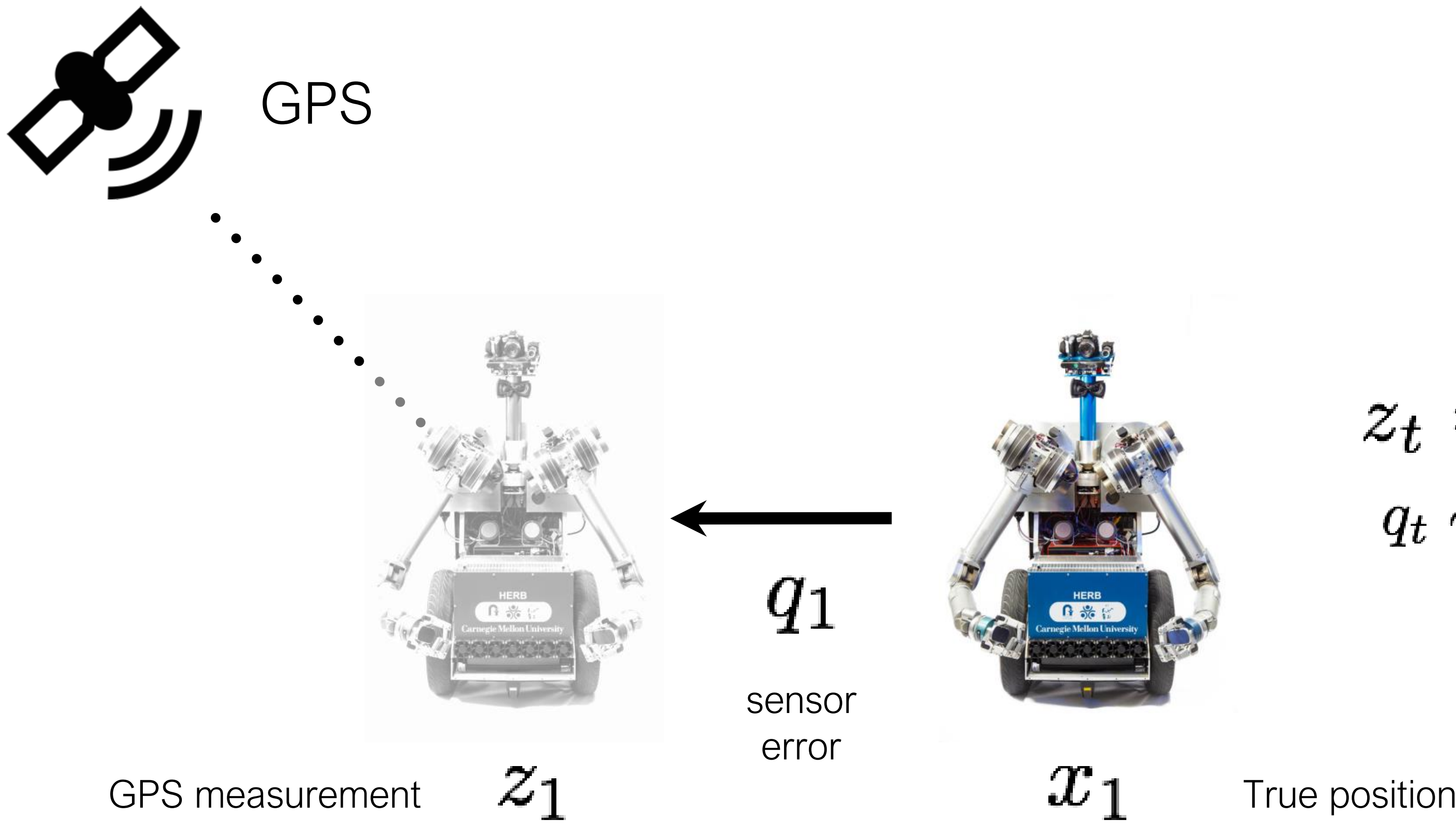


$$z_t = x_t + q_t$$
$$q_t \sim \mathcal{N}(0, \sigma_Q)$$

How do you represent the observation (measurement) model?

$$P(\mathbf{e}|\boldsymbol{x})$$

e represents z

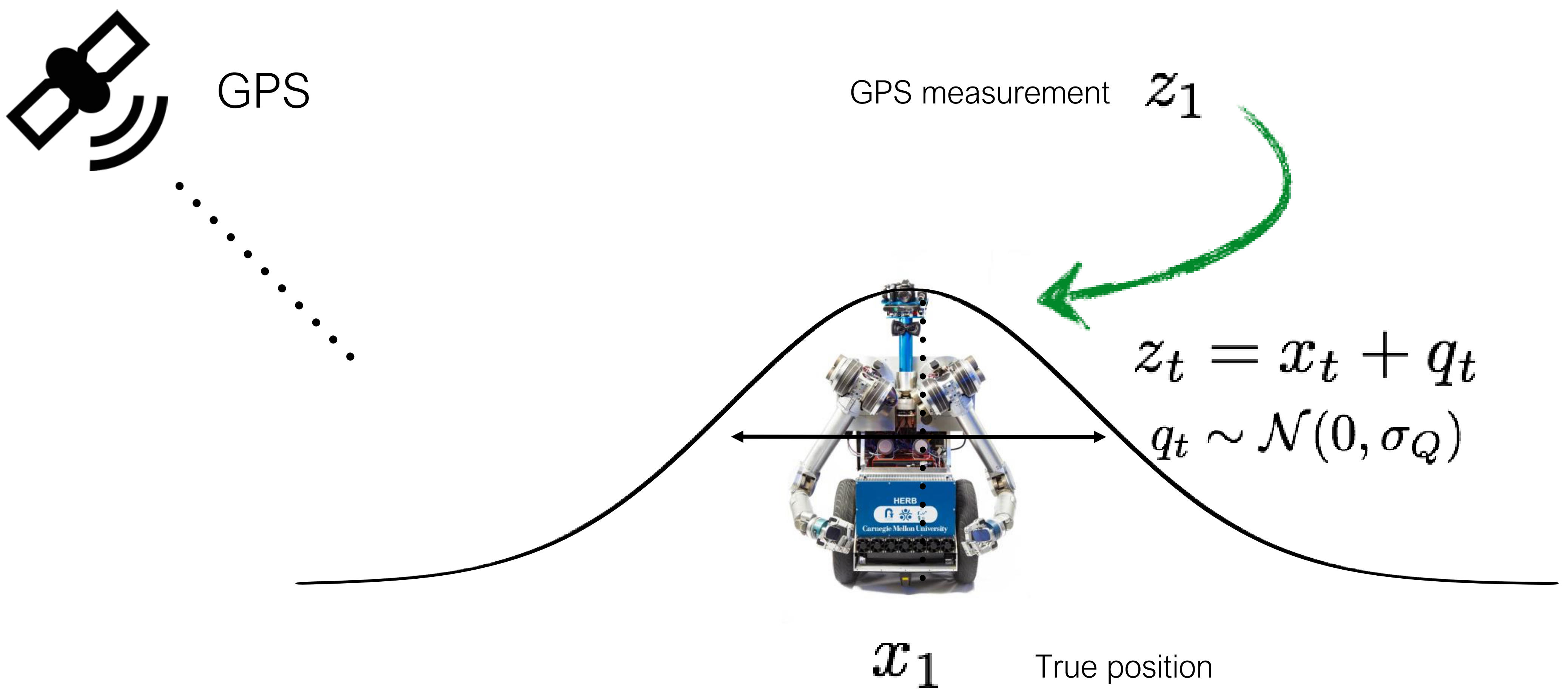


$$z_t = x_t + q_t$$
$$q_t \sim \mathcal{N}(0, \sigma_Q)$$

How do you represent the observation (measurement) model?

Also a linear Gaussian model

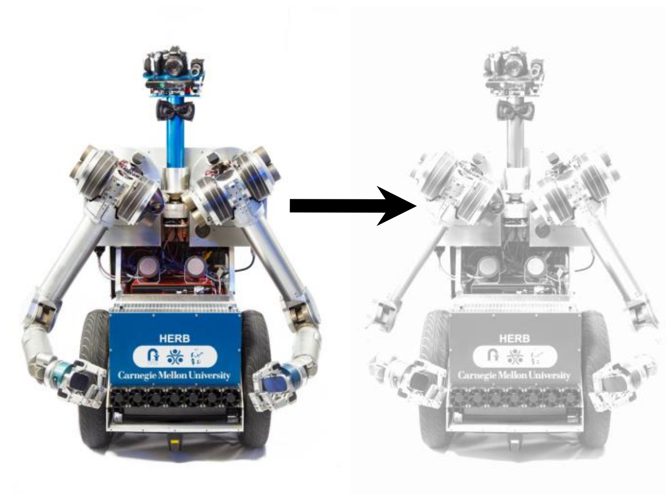
$$P(z_t | x_t) = \mathcal{N}(z_t; x_t, \sigma_Q)$$



How do you represent the observation (measurement) model?

Also a linear Gaussian model

$$P(z_t|x_t) = \mathcal{N}(z_t; x_t, \sigma_Q)$$

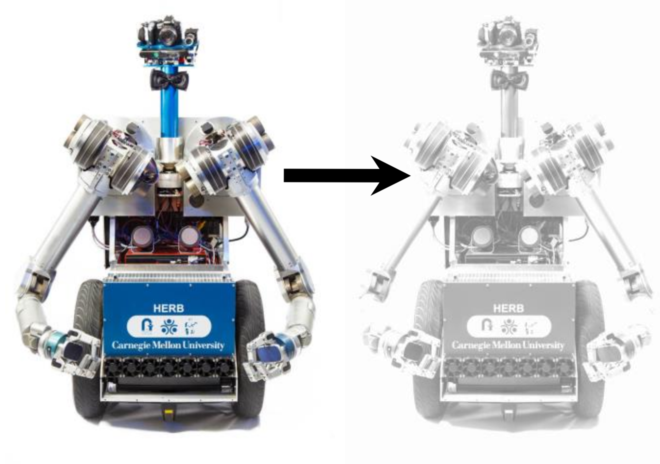


x_0 true position

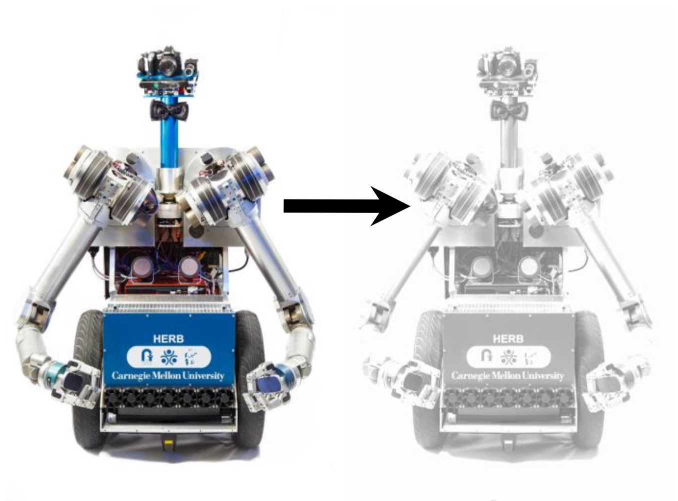
$\hat{x}_0$ initial estimate

initial estimate uncertainty σ_0

Prior (initial) State



How do you represent the prior state probability?



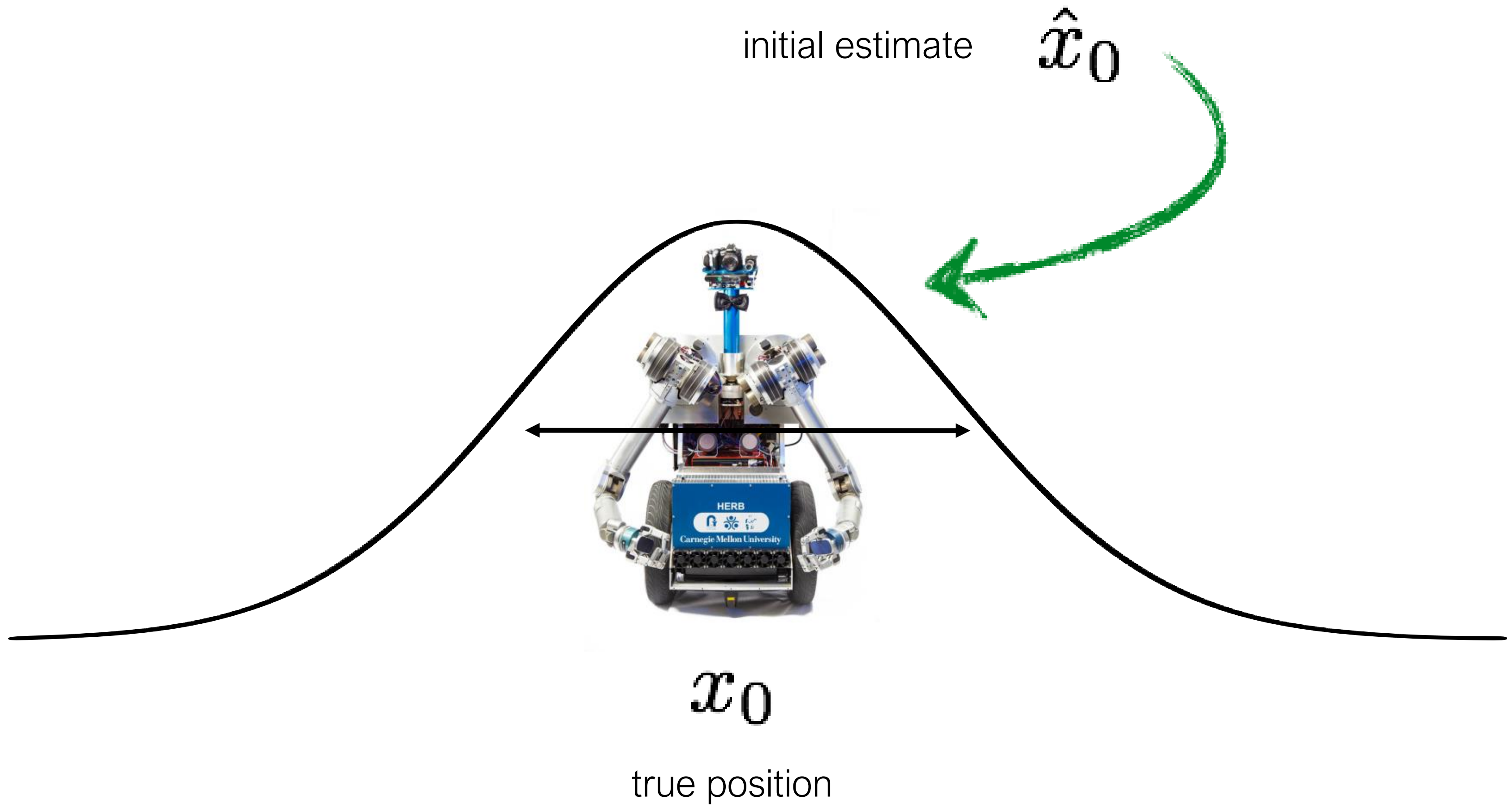
x_0 true position

$\hat{x}_0$ initial estimate

How do you represent the prior state probability?

Also a linear Gaussian model!

$$P(\hat{x}_0) = \mathcal{N}(\hat{x}_0; x_0, \sigma_0)$$



How do you represent the prior state probability?

Also a linear Gaussian model!

$$P(\hat{x}_0) = \mathcal{N}(\hat{x}_0; x_0, \sigma_0)$$

Inference

So how do you do temporal filtering with the KL?

Recall: the first step of filtering was the 'prediction step'

$$P(\boldsymbol{X}_{t+1}|\boldsymbol{e}_{1:t+1}) \propto P(\boldsymbol{e}_{t+1}|\boldsymbol{X}_{t+1}) \overbrace{\int_{\boldsymbol{x}_t} \underbrace{P(\boldsymbol{X}_{t+1}|\boldsymbol{x}_t)}_{\text{motion model}} \underbrace{P(\boldsymbol{x}_t|\boldsymbol{e}_{1:t})}_{\text{belief}} d\boldsymbol{x}_t}^{\text{prediction step}}$$

compute this!
It's just another Gaussian

need to compute the 'prediction' mean and variance…

Prediction

(Using the motion model)

How would you predict $\hat{x}_1$ *given* $\hat{x}_0$*?*

using this 'cap' notation to denote 'estimate'

$$\hat{x}_1 = \hat{x}_0 + s$$

(This is the mean)

$$\sigma_1^2 = \sigma_0^2 + \sigma_r^2$$

(This is the variance)

$$P(\boldsymbol{X}_{t+1}|\boldsymbol{e}_{1:t+1}) \propto P(\boldsymbol{e}_{t+1}|\boldsymbol{X}_{t+1}) \overbrace{\int_{\boldsymbol{x}_t} \underbrace{P(\boldsymbol{X}_{t+1}|\boldsymbol{x}_t)}_{\text{motion model}} \underbrace{P(\boldsymbol{x}_t|\boldsymbol{e}_{1:t})}_{\text{belief}} d\boldsymbol{x}_t}^{\text{prediction step}}$$

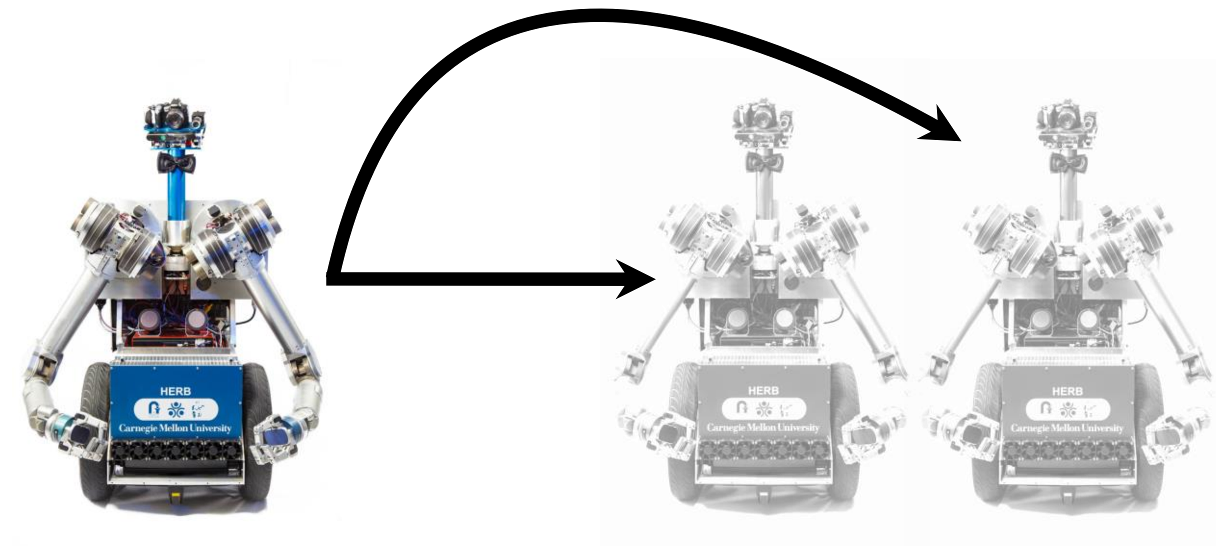
the second step after prediction is …

… update step!

$$P(\boldsymbol{X}_{t+1}|\boldsymbol{e}_{1:t+1}) \propto \underbrace{P(\boldsymbol{e}_{t+1}|\boldsymbol{X}_{t+1})}_{\text{observation}} \int_{\boldsymbol{x}_t} \underbrace{P(\boldsymbol{X}_{t+1}|\boldsymbol{x}_t)P(\boldsymbol{x}_t|\boldsymbol{e}_{1:t})d\boldsymbol{x}_t}_{\text{prediction}}$$

compute this
(using results of the prediction step)

In the **update step,** the **sensor measurement corrects** the system **prediction**

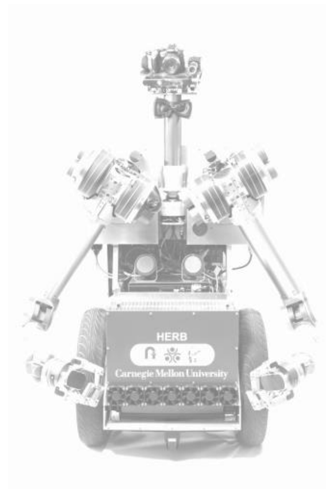


initial estimate	system prediction	sensor estimate
$\hat{x}_0$	$\hat{x}_1$	z_1
	σ_1^2	σ_q^2
	uncertainty	uncertainty

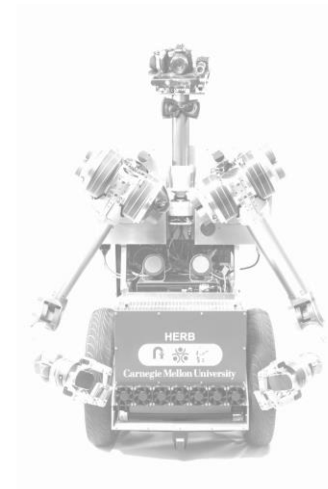
Which estimate is correct? Is there a way to know?

Is there a way to merge this information?

Intuitively, the smaller variance mean less uncertainty.



system prediction σ_1^2



sensor estimate σ_q^2

So we want a <u>weighted state estimate correction</u>

something like this…

$$\hat{x}_1^+ = \frac{\sigma_q^2}{\sigma_1^2 + \sigma_q^2}\hat{x}_1 + \frac{\sigma_1^2}{\sigma_1^2 + \sigma_q^2} z_1$$

This happens naturally in the Bayesian filtering (with Gaussians) framework!

Recall the filtering equation:

$$P(\boldsymbol{X}_{t+1} | \mathbf{e}_{1:t+1}) \propto \underbrace{\overbrace{\boxed{P(\mathbf{e}_{t+1} | \boldsymbol{X}_{t+1})}}^{\text{observation}}}_{\text{Gaussian}} \underbrace{\overbrace{\boxed{\int_{\boldsymbol{x}_t} P(\boldsymbol{X}_{t+1} | \boldsymbol{x}_t) P(\boldsymbol{x}_t | \mathbf{e}_{1:t}) d\boldsymbol{x}_t}}^{\text{one step motion prediction}}}_{\text{Gaussian}}$$

What is the product of two Gaussians?

Recall …

When we multiply the prediction (Gaussian) with the observation model (Gaussian) we get …

… a product of two Gaussians

$$\mu = \frac{\mu_1 \sigma_2^2 + \mu_2 \sigma_1^2}{\sigma_2^2 + \sigma_1^2}$$

$$\sigma = \frac{\sigma_1^2 \sigma_2^2}{\sigma_1^2 + \sigma_2^2}$$

applied to the filtering equation…

$$P(\boldsymbol{X}_{t+1}|\mathbf{e}_{1:t+1}) \propto P(\mathbf{e}_{t+1}|\boldsymbol{X}_{t+1}) \int_{\boldsymbol{x}_t} P(\boldsymbol{X}_{t+1}|\boldsymbol{x}_t) P(\boldsymbol{x}_t|\mathbf{e}_{1:t}) d\boldsymbol{x}_t$$

mean: z_1

variance: σ_q

mean: $\hat{x}_1$

variance: σ_1

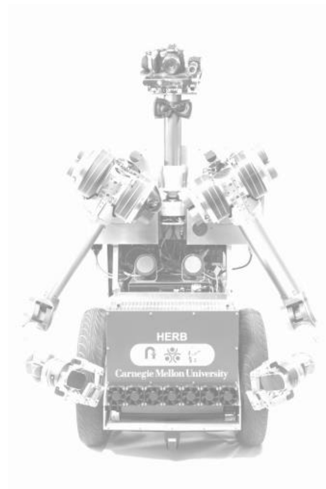
new mean:

$$\hat{x}_1^+ = \frac{\hat{x}_1 \sigma_q^2 + z_1 \sigma_1^2}{\sigma_q^2 + \sigma_1^2}$$

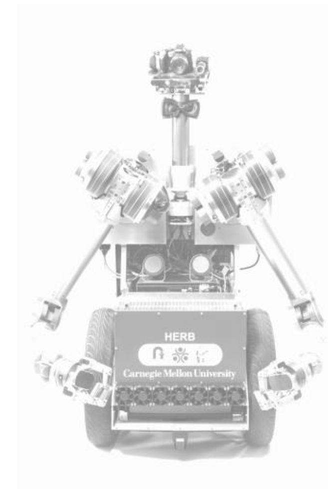
'plus' sign means post 'update' estimate

new variance:

$$\hat{\sigma}_1^{2+} = \frac{\sigma_q^2 \sigma_1^2}{\sigma_q^2 + \sigma_1^2}$$



system prediction σ_1^2



sensor estimate σ_q^2

With a little algebra…

$$\hat{x}_1^+ = \frac{\hat{x}_1\sigma_q^2 + z_1\sigma_1^2}{\sigma_q^2 + \sigma_1^2} = \hat{x}_1 \frac{\sigma_q^2}{\sigma_q^2 + \sigma_1^2} + z_1 \frac{\sigma_1^2}{\sigma_q^2 + \sigma_1^2}$$

We get a <u>weighted state estimate correction!</u>

Kalman gain notation

With a little algebra…

$$\hat{x}_1^+ = \hat{x}_1 + \frac{\sigma_1^2}{\sigma_q^2 + \sigma_1^2}(z_1 - \hat{x}_1) = \hat{x}_1 + K(z_1 - \hat{x}_1)$$

'Kalman gain' 'Innovation'

With a little algebra…

$$\sigma_1^+ = \frac{\sigma_1^2 \sigma_q^2}{\sigma_1^2 + \sigma_q^2} = \left(1 - \frac{\sigma_1^2}{\sigma_1^2 + \sigma_q^2}\right)\sigma_1^2 = (1 - \mathbf{K})\,\sigma_1^2$$

Summary (1D Kalman Filtering)

To solve this…

$$P(\boldsymbol{X}_{t+1}|\boldsymbol{e}_{1:t+1}) \propto P(\boldsymbol{e}_{t+1}|\boldsymbol{X}_{t+1}) \int_{\boldsymbol{x}_t} P(\boldsymbol{X}_{t+1}|\boldsymbol{x}_t) P(\boldsymbol{x}_t|\boldsymbol{e}_{1:t}) d\boldsymbol{x}_t$$

Compute this…

$$\hat{x}_1^+ = \hat{x}_1 + \boxed{\frac{\sigma_1^2}{\sigma_1^2 + \sigma_q^2}}(z_1 - \hat{x}_1) \qquad \sigma_1^{2+} = \sigma_1^2 - \boxed{\frac{\sigma_1^2}{\sigma_1^2 + \sigma_q^2}}\sigma_1^2$$

$$\boxed{K = \frac{\sigma_1^2}{\sigma_1^2 + \sigma_q^2}}$$

'Kalman gain'

$$\hat{x}_1^+ = \hat{x}_1 + K(z_1 - \hat{x}_1) \qquad \sigma_1^{2+} = \sigma_1^2 - K\sigma_1^2$$

mean of the new Gaussian

variance of the new Gaussian

Simple 1D Implementation

```
[x p] = KF(x,v,z)

  x = x + s;
  v = v + q;

  K = v/(v + r);

  x    = x + K * (z - x);
  p    = v - K * v;
```

Just 5 lines of code!

or just 2 lines

```
[x P] = KF(x,v,z)
  x   = (x+s)+(v+q)/((v+q)+r)*(z-(x+s));
  p   = (v+q)-(v+q)/((v+q)+r)*v;
```

Bare computations (algorithm) of Bayesian filtering:

$$\text{KalmanFilter}(\mu_{t-1}, \Sigma_{t-1}, u_t, z_t)$$

motion control

prediction mean

$$\bar{\mu}_t = A_t \mu_{t-1} + B u_t$$

'old' mean

Prediction

'old' covariance

prediction covariance

$$\bar{\Sigma}_t = A_t \Sigma_{t-1} A_t^\top + R$$

Gaussian noise

$$K_t = \bar{\Sigma}_t C_t^\top (C_t \bar{\Sigma}_t C_t^\top + Q_t)^{-1}$$

Gain

observation model

update mean

$$\mu_t = \bar{\mu}_t + K_t (z_t - C_t \bar{\mu}_t)$$

Update

update covariance

$$\Sigma_t = (I - K_t C_t) \bar{\Sigma}_t$$

Simple Multi-dimensional Implementation
(also 5 lines of code!)

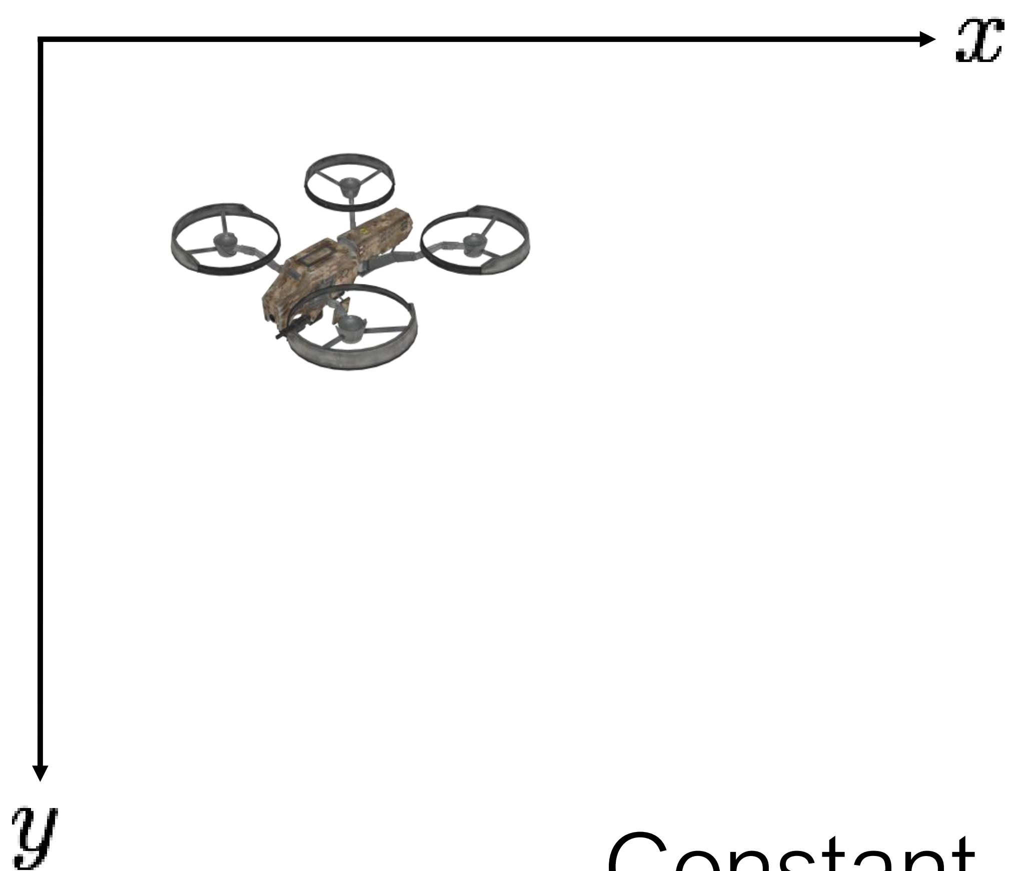
```
[x P] = KF(x,P,z)

  x = A*x;
  P = A*P*A' + Q;

  K = P*C'/(C*P*C' + R);

  x   = x + K*(z - C*x);
  P   = (eye(size(K,1))-K*C)*P;
```

2D Example



state

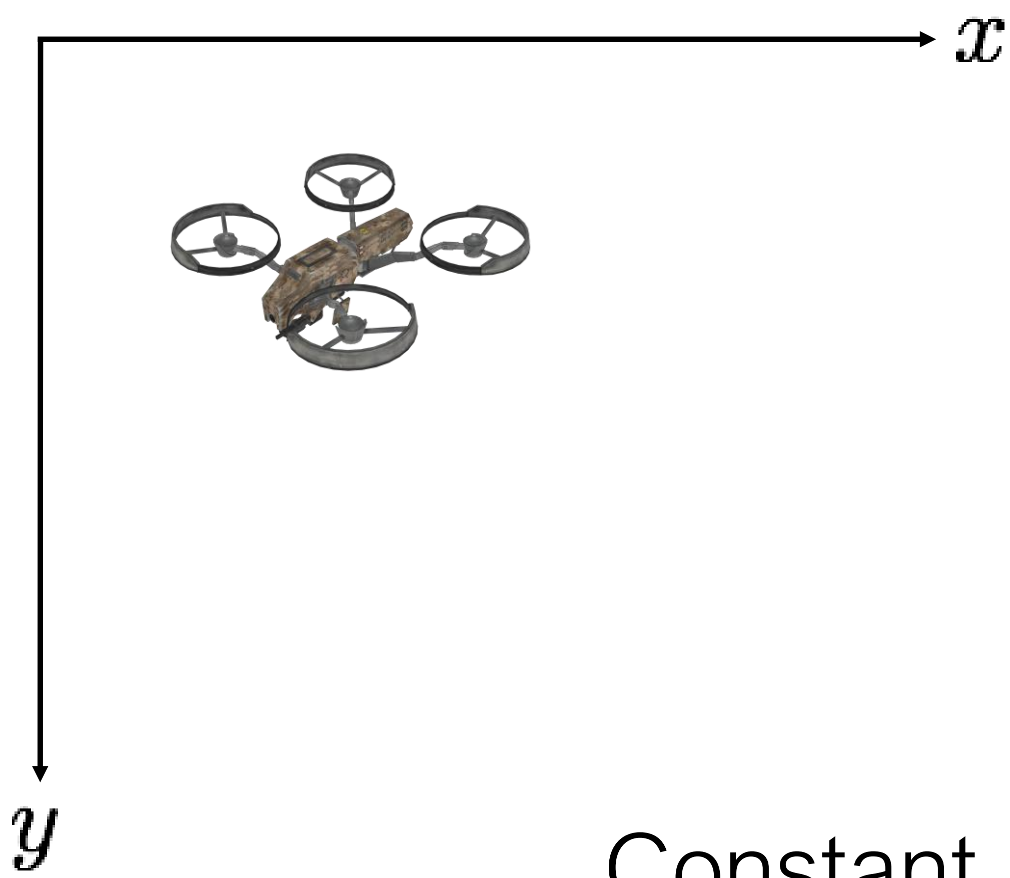
$$\boldsymbol{x} = \begin{bmatrix} x \\ y \end{bmatrix}$$

measurement

$$\boldsymbol{z} = \begin{bmatrix} x \\ y \end{bmatrix}$$

Constant position Motion Model

$$\boldsymbol{x}_t = A\boldsymbol{x}_{t-1} + B\boldsymbol{u}_t + \epsilon_t$$



state

$$\boldsymbol{x} = \left[\begin{array}{c} x \\ y \end{array} \right]$$

measurement

$$\boldsymbol{z} = \left[\begin{array}{c} x \\ y \end{array} \right]$$

Constant position Motion Model

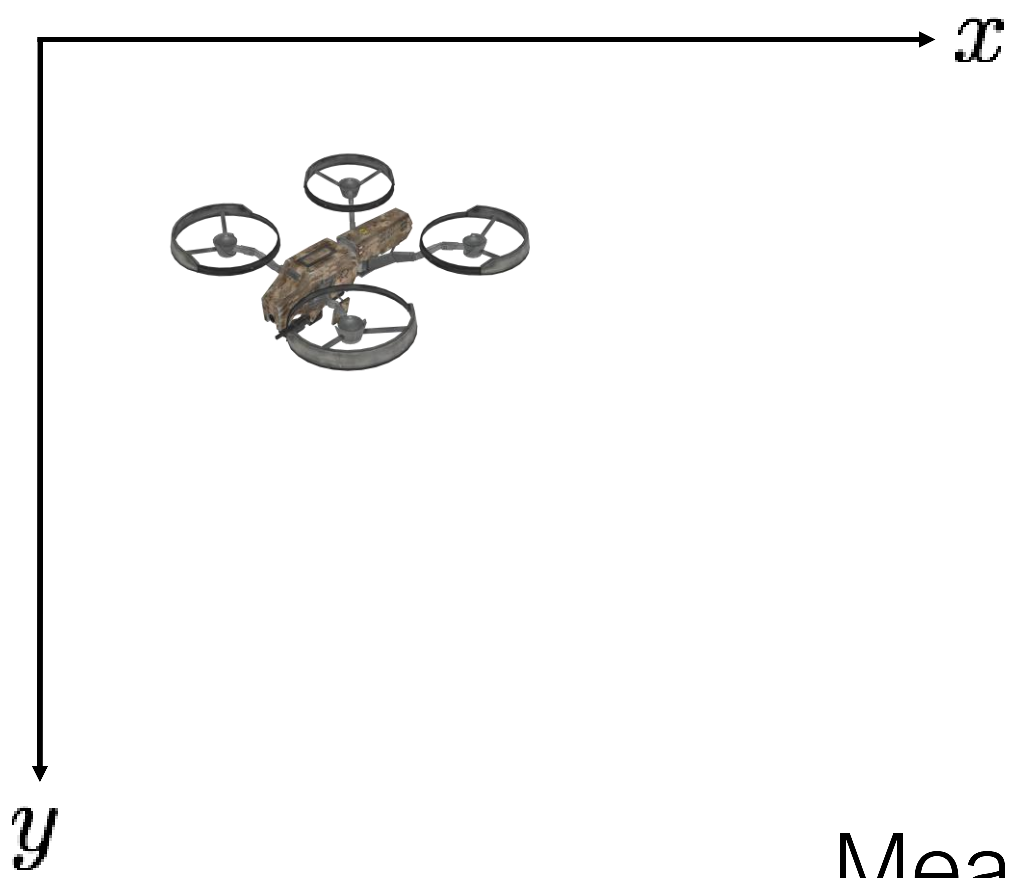
$$\boldsymbol{x}_t = A\boldsymbol{x}_{t-1} + B\boldsymbol{u}_t + \epsilon_t$$

system noise

$$\epsilon_t \sim \mathcal{N}(\mathbf{0}, R)$$

Constant position

$$A = \left[\begin{array}{cc} 1 & 0 \\ 0 & 1 \end{array} \right] \qquad B\boldsymbol{u} = \left[\begin{array}{c} 0 \\ 0 \end{array} \right] \qquad R = \left[\begin{array}{cc} \sigma_r^2 & 0 \\ 0 & \sigma_r^2 \end{array} \right]$$



state

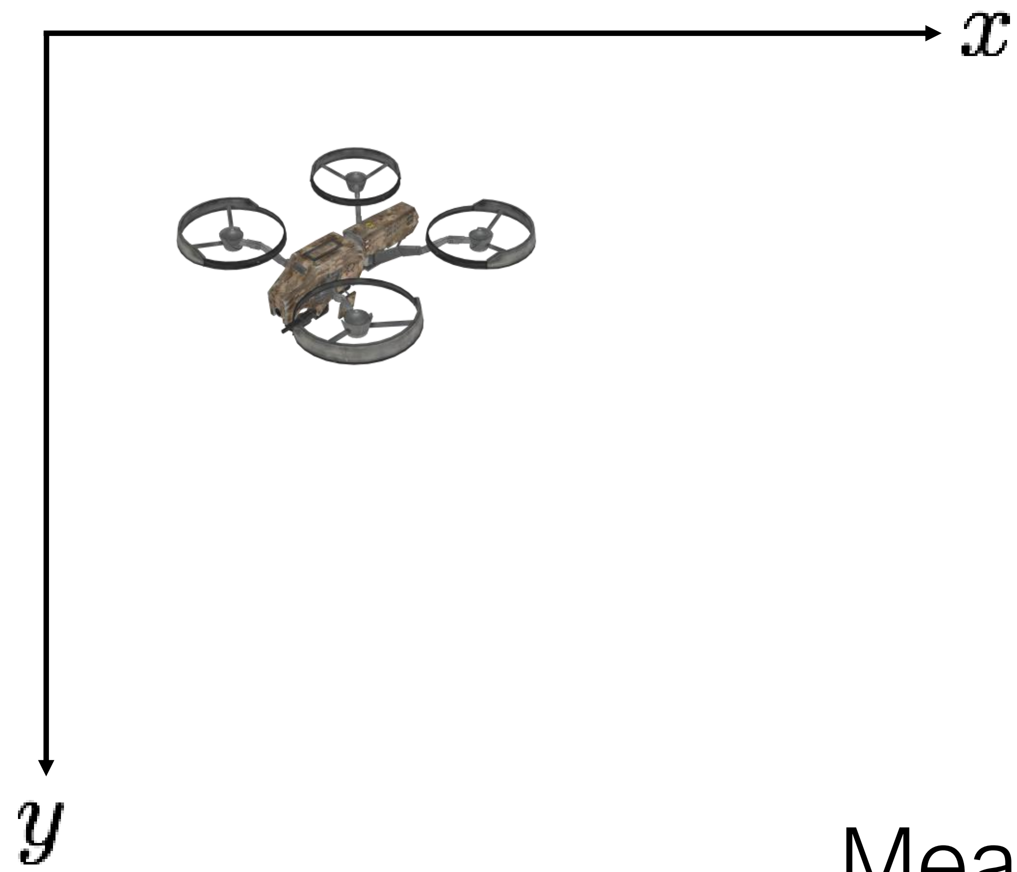
$$\boldsymbol{x} = \begin{bmatrix} x \\ y \end{bmatrix}$$

measurement

$$\boldsymbol{z} = \begin{bmatrix} x \\ y \end{bmatrix}$$

Measurement Model

$$\boldsymbol{z}_t = C_t \boldsymbol{x}_t + \delta_t$$



state

$$\boldsymbol{x} = \begin{bmatrix} x \\ y \end{bmatrix}$$

measurement

$$\boldsymbol{z} = \begin{bmatrix} x \\ y \end{bmatrix}$$

Measurement Model

$$\boldsymbol{z}_t = C_t \boldsymbol{x}_t + \delta_t$$

zero-mean measurement noise

$$C = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \qquad \delta_t \sim \mathcal{N}(\mathbf{0}, Q) \qquad Q = \begin{bmatrix} \sigma_q^2 & 0 \\ 0 & \sigma_q^2 \end{bmatrix}$$

Algorithm for the 2D object tracking example

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

motion model   observation model

General Case

$$\bar{\mu}_t = A_t \mu_{t-1} + B u_t$$
$$\bar{\Sigma}_t = A_t \Sigma_{t-1} A_t^\top + R$$
$$K_t = \bar{\Sigma}_t C_t^\top (C_t \bar{\Sigma}_t C_t^\top + Q_t)^{-1}$$
$$\mu_t = \bar{\mu}_t + K_t (z_t - C_t \bar{\mu}_t)$$
$$\Sigma_t = (I - K_t C_t) \bar{\Sigma}_t$$

Constant position Model

$$\bar{\boldsymbol{x}}_t = \boldsymbol{x}_{t-1}$$
$$\bar{\Sigma}_t = \Sigma_{t-1} + R$$
$$K_t = \bar{\Sigma}_t (\bar{\Sigma}_t + Q)^{-1}$$
$$\boldsymbol{x}_t = \bar{\boldsymbol{x}}_t + K_t (z_t - \bar{\boldsymbol{x}}_t)$$
$$\Sigma_t = (I - K_t) \bar{\Sigma}_t$$

Just 4 lines of code

```
[x P] = KF_constPos(x,P,z)

  P = P + Q;

  K = P / (P + R);

  x   = x + K * (z - x);
  P   = (eye(size(K,1))-K) * P;
```

Where did the 5th line go?

General Case

$$\bar{\mu}_t = A_t \mu_{t-1} + B u_t$$
$$\bar{\Sigma}_t = A_t \Sigma_{t-1} A_t^\top + R$$
$$K_t = \bar{\Sigma}_t C_t^\top (C_t \bar{\Sigma}_t C_t^\top + Q_t)^{-1}$$
$$\mu_t = \bar{\mu}_t + K_t(z_t - C_t \bar{\mu}_t)$$
$$\Sigma_t = (I - K_t C_t)\bar{\Sigma}_t$$

Constant position Model

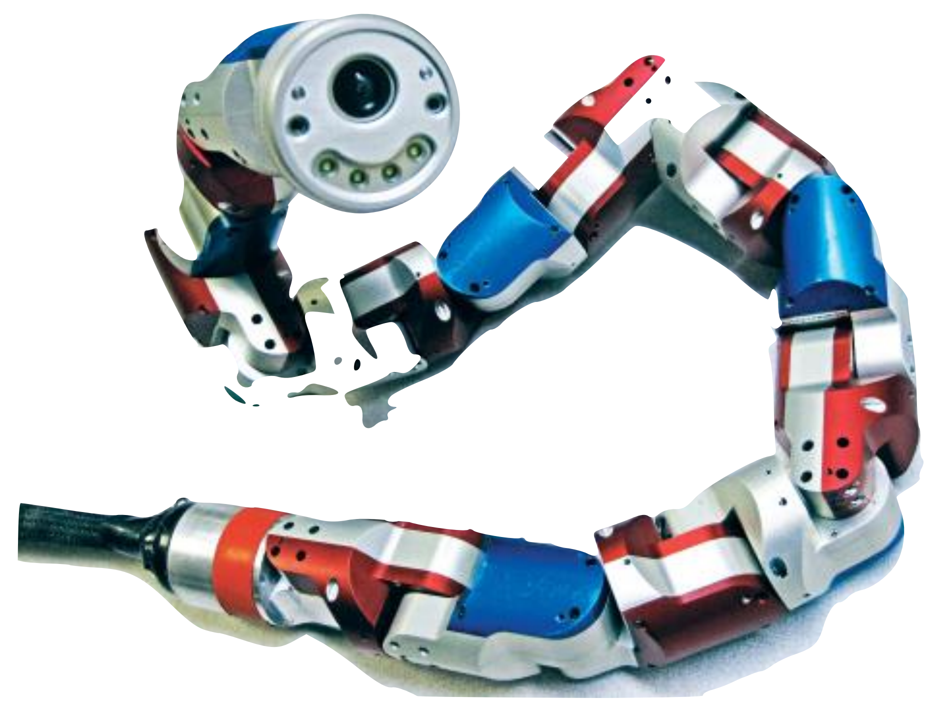
$$\bar{\boldsymbol{x}}_t = \boldsymbol{x}_{t-1}$$
$$\bar{\Sigma}_t = \Sigma_{t-1} + R$$
$$K_t = \bar{\Sigma}_t(\bar{\Sigma}_t + Q)^{-1}$$
$$\boldsymbol{x}_t = \bar{\boldsymbol{x}}_t + K_t(z_t - \bar{\boldsymbol{x}}_t)$$
$$\Sigma_t = (I - K_t)\bar{\Sigma}_t$$

Extended Kalman filter

Motion model of the Kalman filter is linear

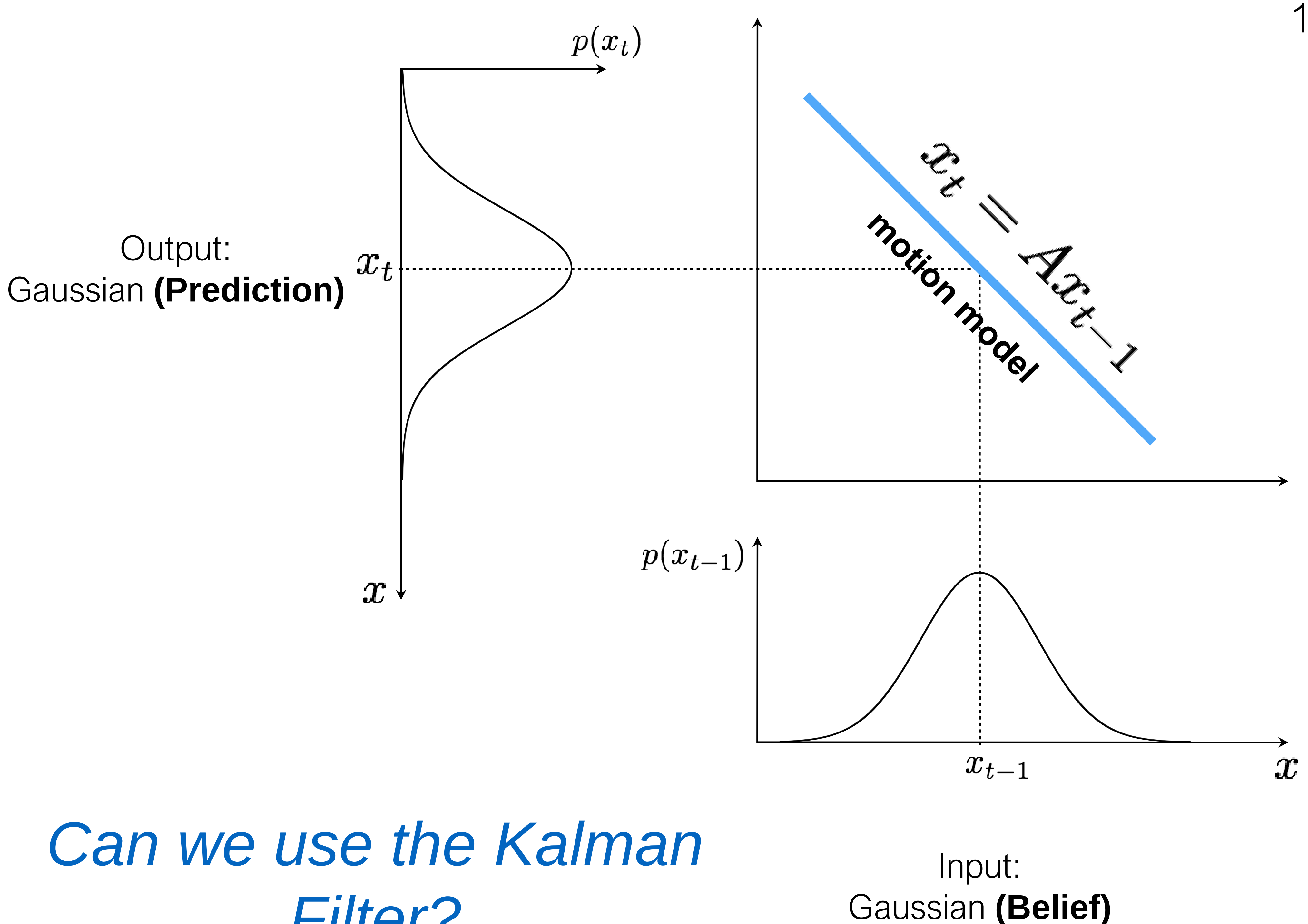
$$x_t = Ax_{t-1} + Bu_t + \epsilon_t$$

but motion is not always linear



Visualizing **linear** models

1D motion model example



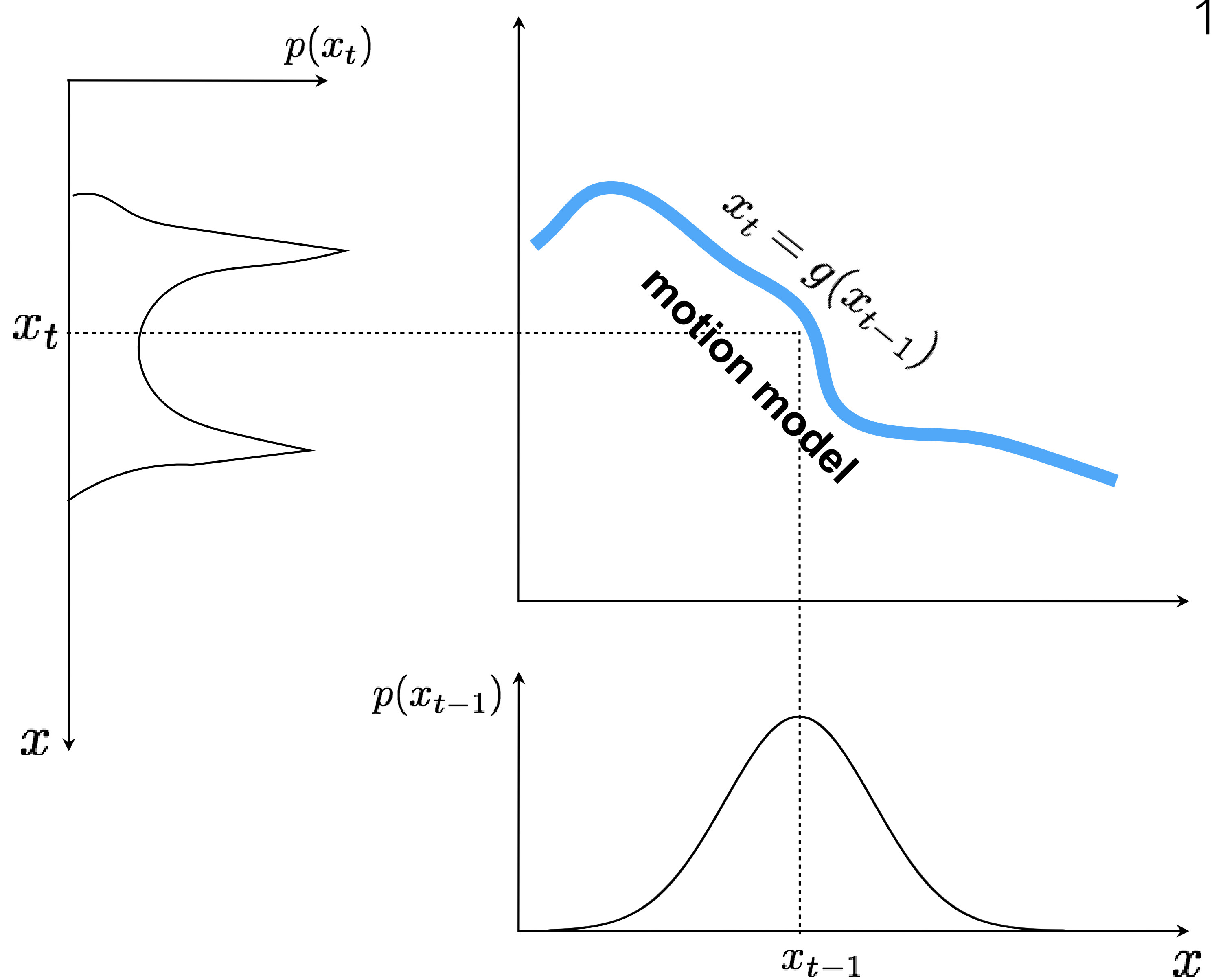
Can we use the Kalman Filter?

(motion model and observation model are linear)

Visualizing **non-linear** models

1D motion model example

Output:
NOT
Gaussian



Input:
Gaussian **(Belief)**

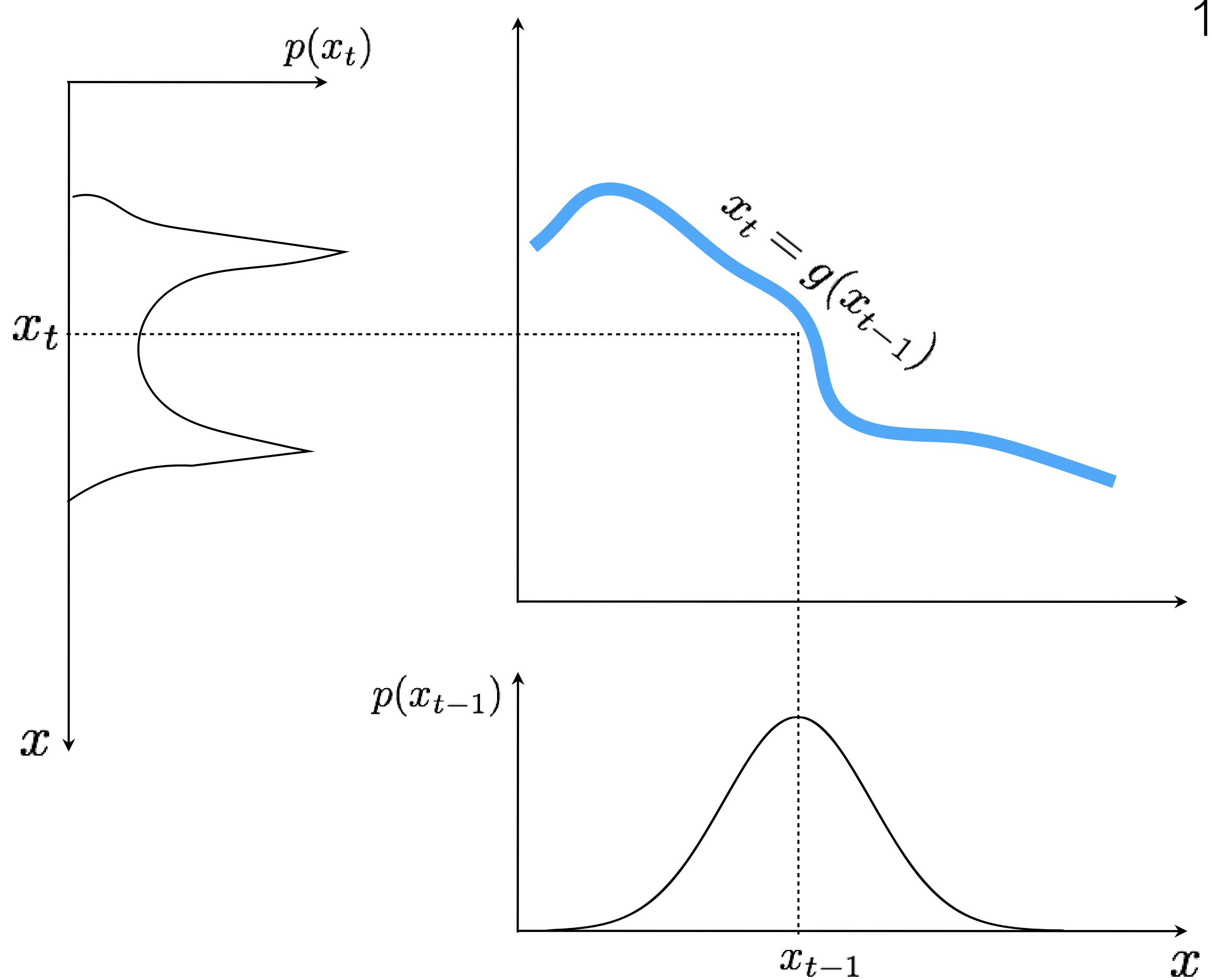
Can we use the Kalman Filter?

(motion model is not linear)

How do you deal with non-linear models?

1D motion model example

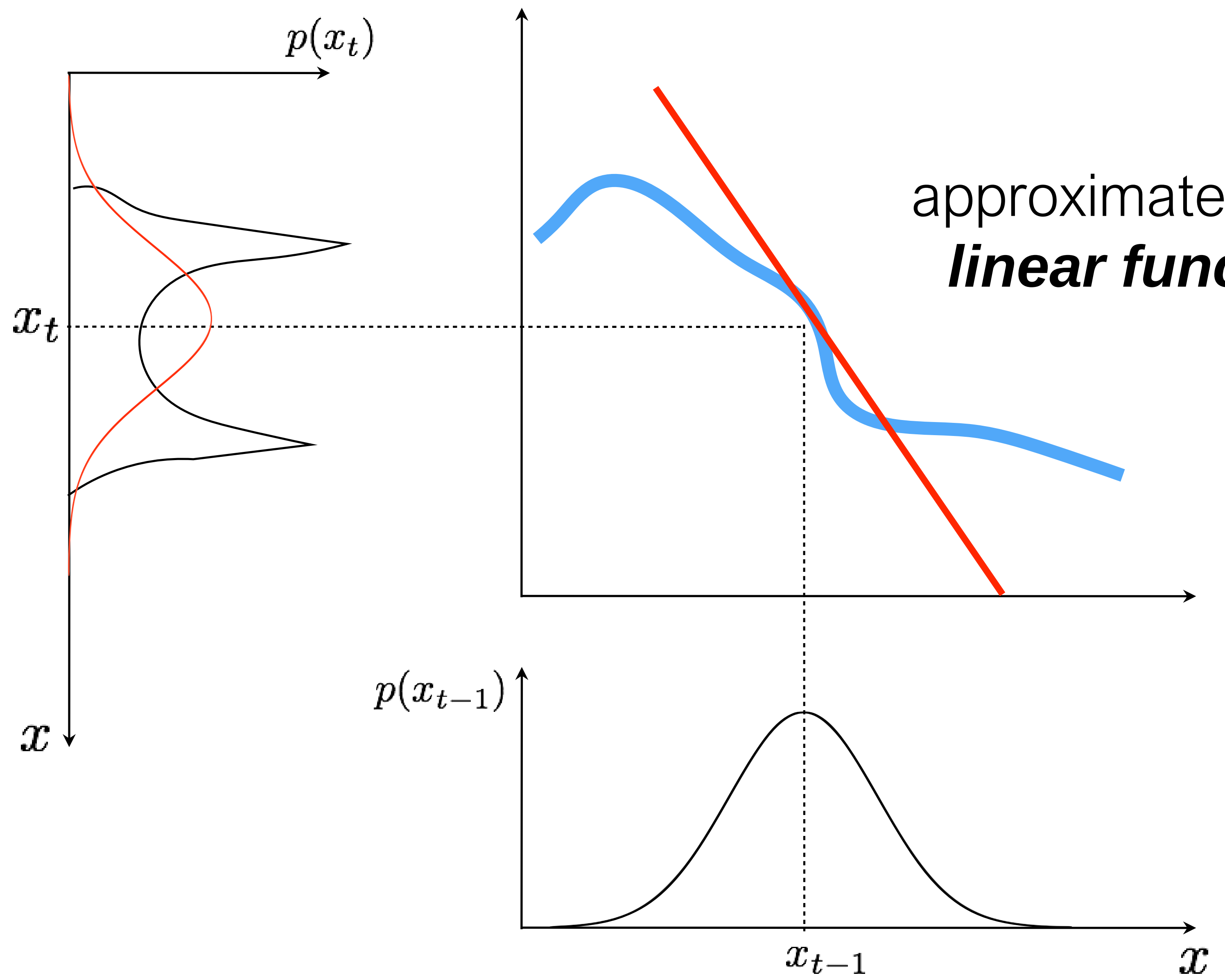
Output:
NOT
Gaussian



Input:
Gaussian

How do you deal with non-linear models?

Output:
Gaussian



Input:
Gaussian

When does this trick work?

Extended Kalman Filter

- Does not assume linear Gaussian models
- Assumes Gaussian noise
- Uses local linear approximations of model to keep the efficiency of the KF framework

Kalman Filter

linear motion model

$$x_t = Ax_{t-1} + Bu_t + \epsilon_t$$

linear sensor model

$$z_t = C_t x_t + \delta_t$$

Extended Kalman Filter

non-linear motion model

$$x_t = g(x_{t-1}, u_t) + \epsilon_t$$

non-linear sensor model

$$z_t = H(x_t) + \delta_t$$

Motion model linearization

$$g(x_{t-1}, u_t) \approx g(\mu_{t-1}, u_t) + \frac{\partial g(\mu_{t-1}, u_t)}{\partial x_{t-1}}(x_{t-1} - \mu_{t-1})$$

Taylor series expansion

Motion model linearization

$$
\begin{aligned}
g(x_{t-1}, u_t) &\approx g(\mu_{t-1}, u_t) + \frac{\partial g(\mu_{t-1}, u_t)}{\partial x_{t-1}}(x_{t-1} - \mu_{t-1}) \\
&\approx g(\mu_{t-1}, u_t) + \underset{\substack{\uparrow \\ \text{What's this called?}}}{G_t} (x_{t-1} - \mu_{t-1})
\end{aligned}
$$

Motion model linearization

$$
\begin{aligned}
g(x_{t-1}, u_t) &\approx g(\mu_{t-1}, u_t) + \frac{\partial g(\mu_{t-1}, u_t)}{\partial x_{t-1}} (x_{t-1} - \mu_{t-1}) \\
&\approx g(\mu_{t-1}, u_t) + \qquad G_t \qquad (x_{t-1} - \mu_{t-1})
\end{aligned}
$$

What's this called?

Jacobian Matrix

'the rate of change in x'
'slope of the function'

Motion model linearization

$$
\begin{aligned}
g(x_{t-1}, u_t) &\approx g(\mu_{t-1}, u_t) + \frac{\partial g(\mu_{t-1}, u_t)}{\partial x_{t-1}}(x_{t-1} - \mu_{t-1}) \\
&\approx g(\mu_{t-1}, u_t) + \qquad G_t \qquad (x_{t-1} - \mu_{t-1})
\end{aligned}
$$

Jacobian Matrix

'the rate of change in x'
'slope of the function'

Sensor model linearization

$$
\begin{aligned}
h(x_t) &\approx h(\bar{\mu}_t) + \frac{\partial h(\bar{\mu}_t)}{\partial x_t}(x_{t-1} - \bar{\mu}_t) \\
&\approx h(\bar{\mu}_t) + \quad H_t \quad (x_t - \bar{\mu}_t)
\end{aligned}
$$

New EKF Algorithm

(pretty much the same)

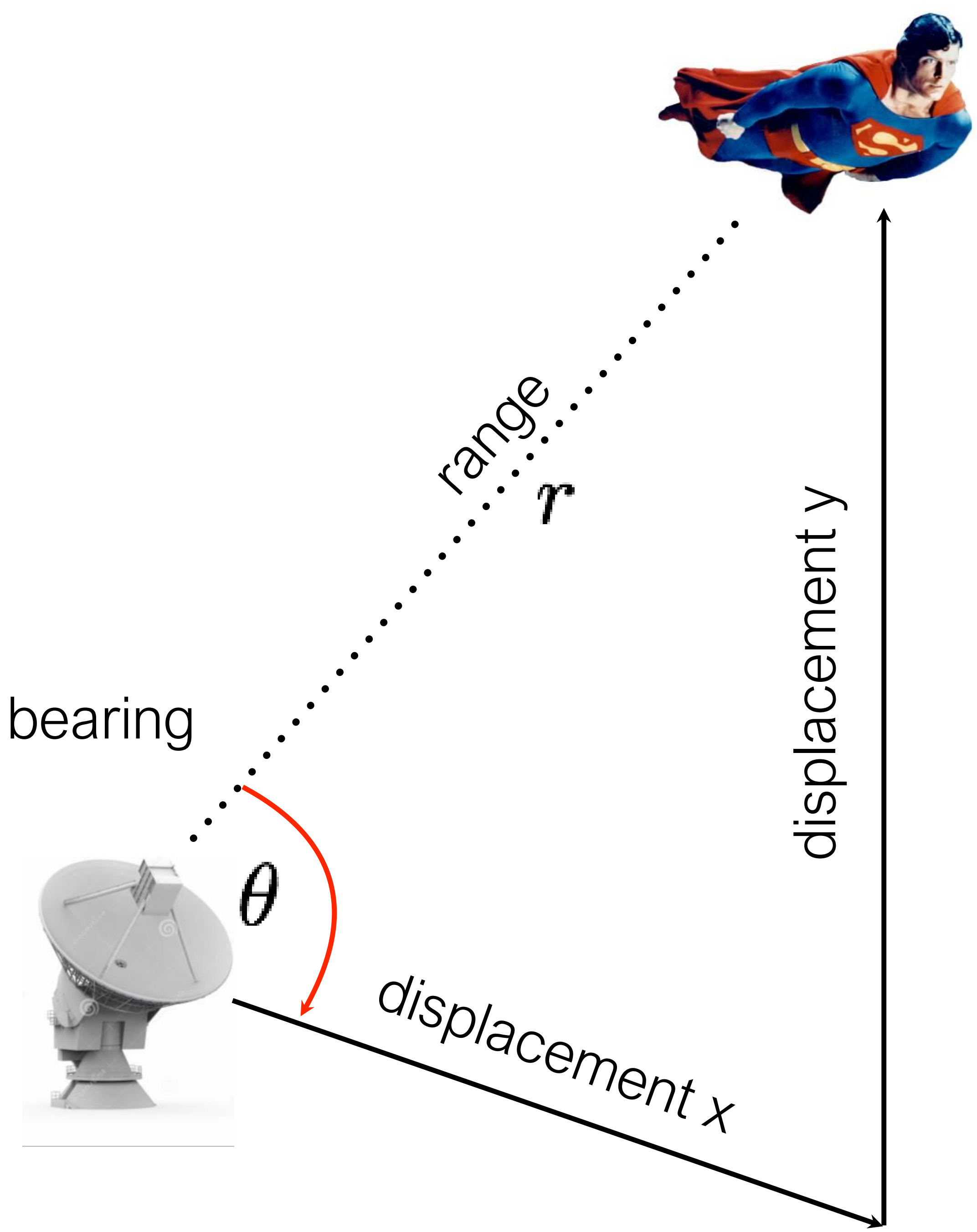
Kalman Filter

$$
\begin{aligned}
\bar{\mu}_t &= A_t \mu_{t-1} + B u_t \\
\bar{\Sigma}_t &= A_t \Sigma_{t-1} A_t^\top + R \\
K_t &= \bar{\Sigma}_t C_t^\top (C_t \bar{\Sigma}_t C_t^\top + Q_t)^{-1} \\
\mu_t &= \bar{\mu}_t + K_t (z_t - C_t \bar{\mu}_t) \\
\Sigma_t &= (I - K_t C_t) \bar{\Sigma}_t
\end{aligned}
$$

Extended KF

$$
\begin{aligned}
\bar{\mu}_t &= g(\mu_{t-1}, u_t) \\
\bar{\Sigma}_t &= G_t \bar{\Sigma}_{t-1} G_t^\top + R \\
K_t &= \bar{\Sigma}_t H_t^\top \left(H_t \bar{\Sigma}_t H^\top + Q \right)^{-1} \\
\mu_t &= \bar{\mu}_t + K_t (z_t - h(\bar{\mu}_t)) \\
\Sigma_t &= (I - K_t H_t) \bar{\Sigma}_t
\end{aligned}
$$

2D example



state: position-velocity

$$\boldsymbol{x} = \begin{array}{cl} \left[\begin{array}{c} x \\ \dot{x} \\ y \\ \dot{y} \end{array}\right] & \begin{array}{l} \text{position} \\ \text{velocity} \\ \text{position} \\ \text{velocity} \end{array} \end{array}$$

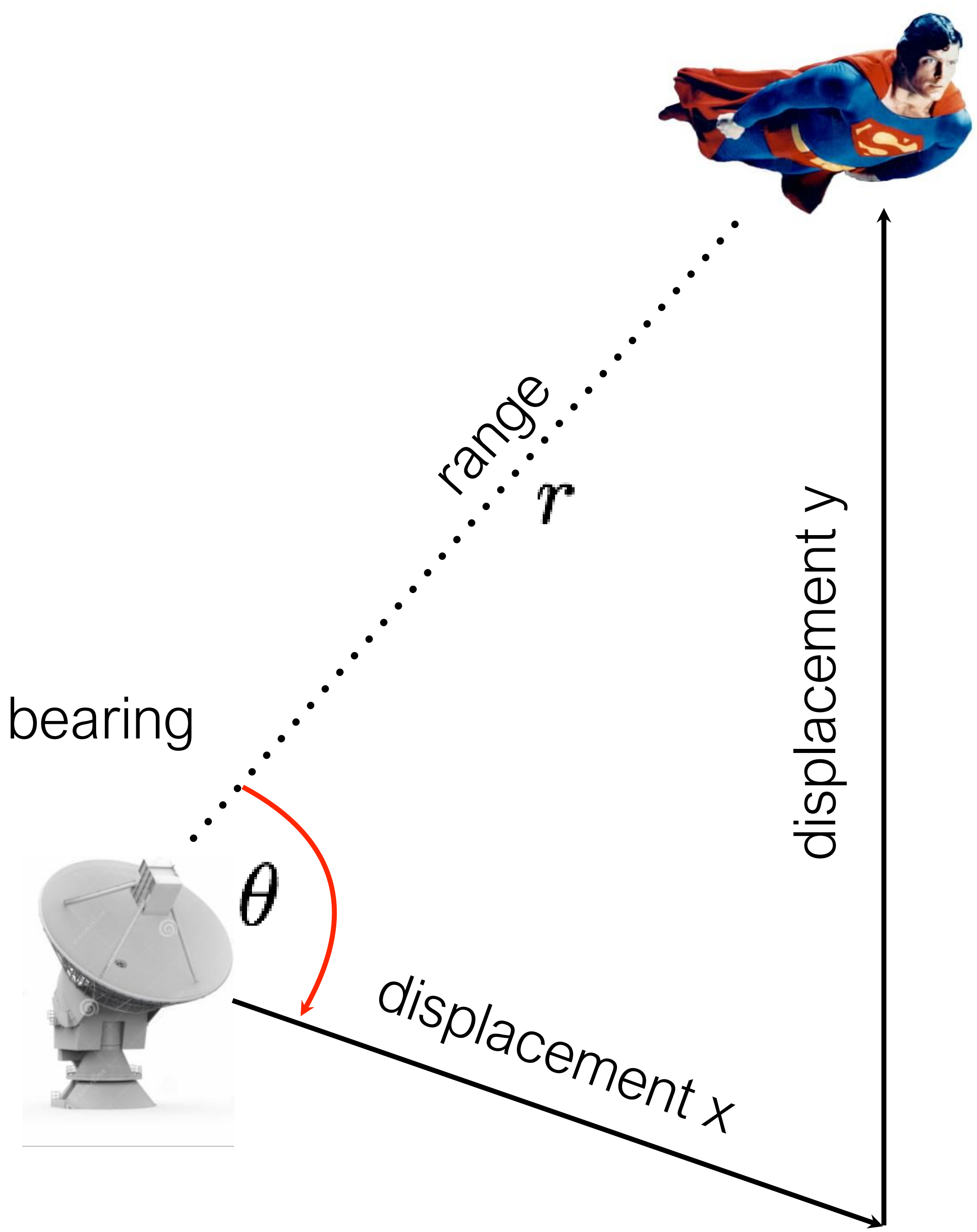
constant velocity motion model

$$A = \begin{bmatrix} 1 & \Delta t & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & \Delta t \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

with additive Gaussian noise

Motion model is linear but …

measurement: range-bearing



$$\boldsymbol{z} = \begin{bmatrix} r \\ \theta \end{bmatrix}$$

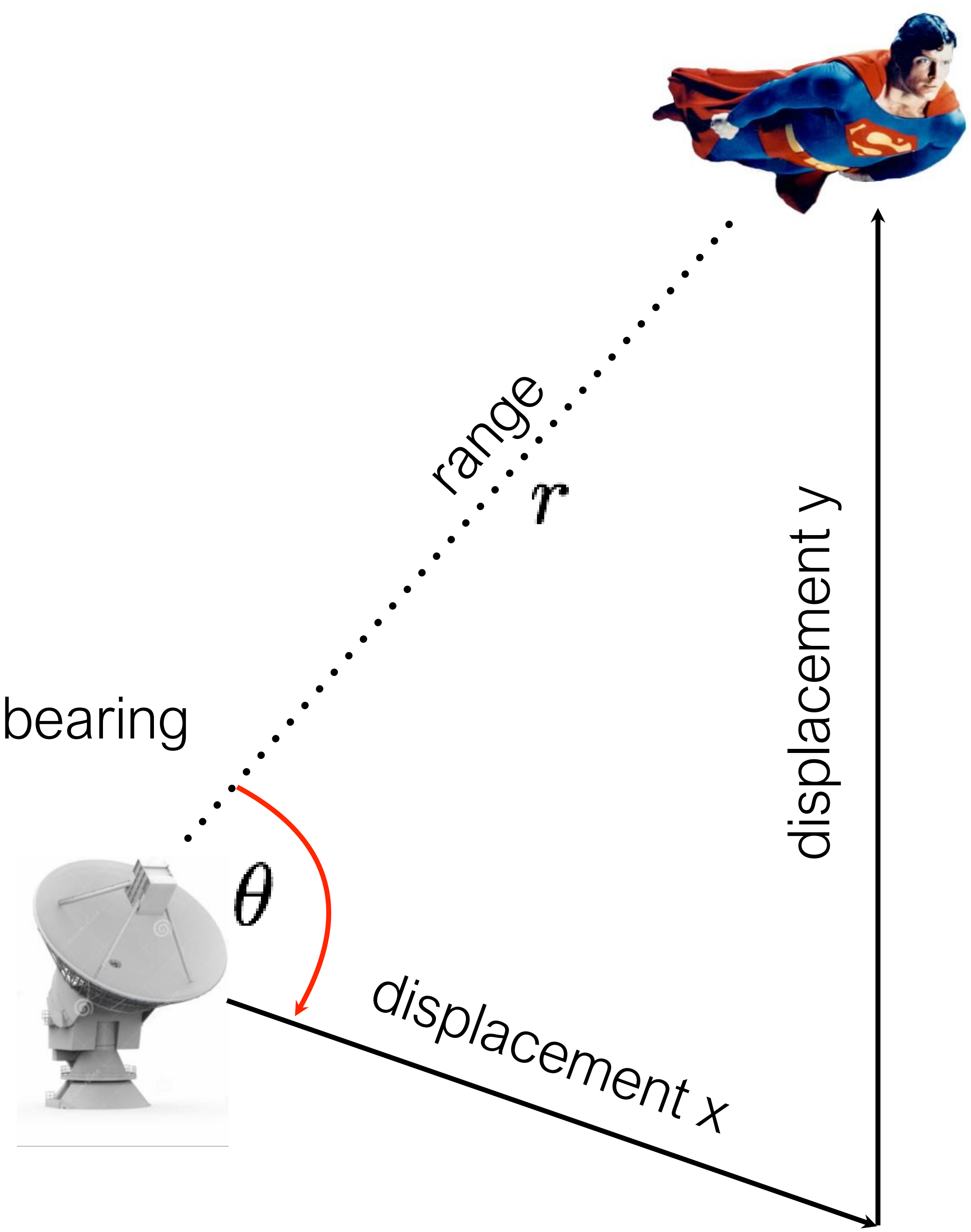
$$= \begin{bmatrix} \sqrt{x^2 + y^2} \\ \tan^{-1}(y/x) \end{bmatrix}$$

measurement model

Is the measurement model linear?

$$\boldsymbol{z} = h(r, \theta)$$

with additive Gaussian noise



measurement: range-bearing

$$z = \begin{bmatrix} r \\ \theta \end{bmatrix}$$

$$= \begin{bmatrix} \sqrt{x^2 + y^2} \\ \tan^{-1}(y/x) \end{bmatrix}$$

measurement model

Is the measurement model linear?

$$z = h(r, \theta)$$

with additive Gaussian noise

non-linear!

What should we do?

linearize the observation/measurement model!

$$\boldsymbol{z} = \begin{bmatrix} r \\ \theta \end{bmatrix} = \begin{bmatrix} \sqrt{x^2 + y^2} \\ \tan^{-1}(y/x) \end{bmatrix}$$

$$H = \frac{\partial \boldsymbol{z}}{\partial \boldsymbol{x}} = ?$$

What is the Jacobian?

$$H = \begin{bmatrix} \frac{\partial r}{\partial x} & \frac{\partial r}{\partial \dot{x}} & \frac{\partial r}{\partial y} & \frac{\partial r}{\partial \dot{y}} \\ \\ \frac{\partial \theta}{\partial x} & \frac{\partial \theta}{\partial \dot{x}} & \frac{\partial \theta}{\partial y} & \frac{\partial \theta}{\partial \dot{y}} \end{bmatrix} =$$

$$\boldsymbol{z} = \begin{bmatrix} r \\ \theta \end{bmatrix} = \begin{bmatrix} \sqrt{x^2+y^2} \\ \tan^{-1}(y/x) \end{bmatrix}$$

$$H = \frac{\partial \boldsymbol{z}}{\partial \boldsymbol{x}} = ?$$

What is the Jacobian?

Jacobian used in the Taylor series expansion looks like …

$$H = \begin{bmatrix} \frac{\partial r}{\partial x} & \frac{\partial r}{\partial \dot{x}} & \frac{\partial r}{\partial y} & \frac{\partial r}{\partial \dot{y}} \\ \\ \frac{\partial \theta}{\partial x} & \frac{\partial \theta}{\partial \dot{x}} & \frac{\partial \theta}{\partial y} & \frac{\partial \theta}{\partial \dot{y}} \end{bmatrix} = \begin{bmatrix} \cos(\theta) & 0 & \sin(\theta) & 0 \\ \\ -sin(\theta)/r & 0 & \cos(\theta)/r & 0 \end{bmatrix}$$

```
[x P] = EKF(x,P,z,dt)

r    = sqrt (x(1)^2+x(3)^2);
b    = atan2(x(3),x(1));
y    = [r; b];

H = [ cos(b)    0  sin(b)     0;
     -sin(b)/r 0  cos(b)/r   0];

x = F*x;
P = F*P*F' + Q;

K = P*H'/(H*P*H' + R);

x    = x + K*(z - y);
P    = (eye(size(K,1))-K*H)*P;
```

extra computation for the EKF measurement model Jacobian

Parameters:

```
Q = diag([0 .1 0 .1]);
R = diag([50^2 0.005^2]);
F = [ 1 dt 0 0;
      0 1 0 0;
      0 0 1 dt;
      0 0 0 1];
```

Problems with EKFs

Taylor series expansion = poor approximation of non-linear functions
success of linearization depends on limited uncertainty and amount of local non-linearity

Computing partial derivatives is a pain

Drifts when linearization is a bad approximation

Cannot handle multi-modal (multi-hypothesis) distributions

SLAM

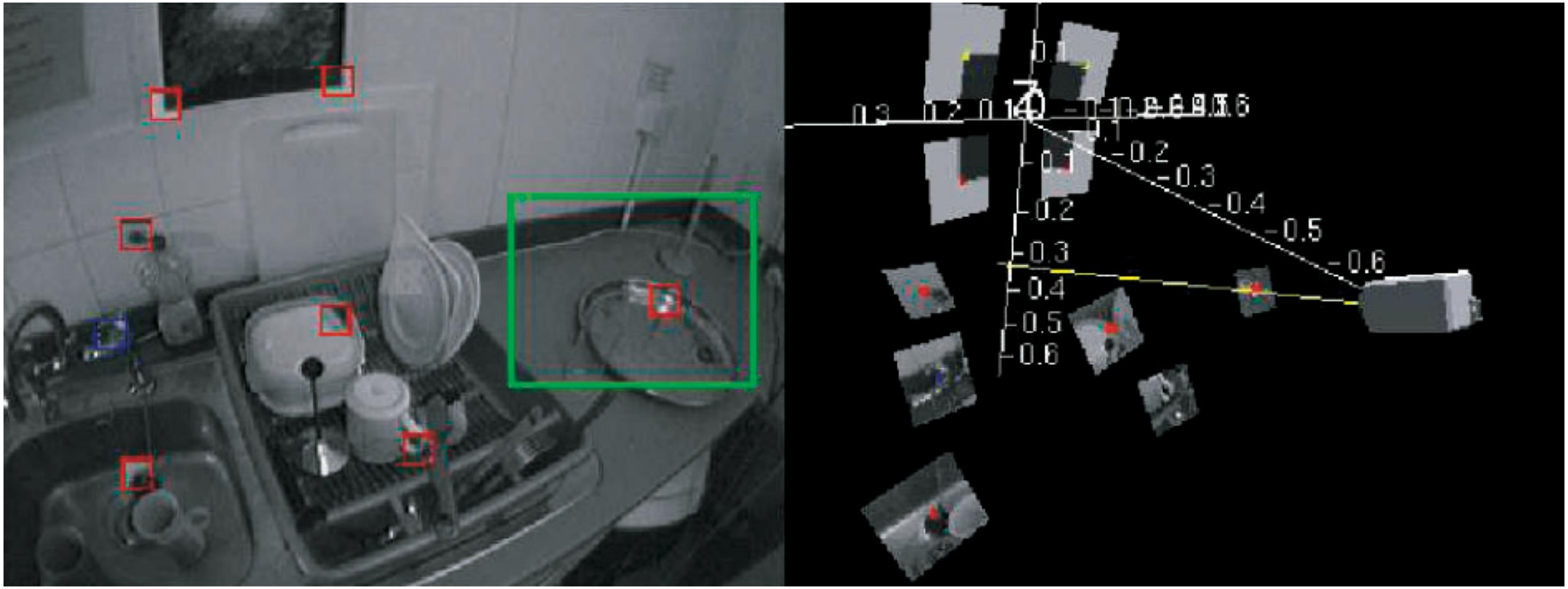
MonoSLAM: Real-Time Single Camera SLAM

Andrew J. Davison, Ian D. Reid, *Member*, *IEEE*, Nicholas D. Molton, and
Olivier Stasse, *Member*, *IEEE*

Abstract—We present a real-time algorithm which can recover the 3D trajectory of a monocular camera, moving rapidly through a previously unknown scene. Our system, which we dub *MonoSLAM*, is the first successful application of the SLAM methodology from mobile robotics to the "pure vision" domain of a single uncontrolled camera, achieving real time but drift-free performance inaccessible to Structure from Motion approaches. The core of the approach is the online creation of a sparse but persistent map of natural landmarks within a probabilistic framework. Our key novel contributions include an *active* approach to mapping and measurement, the use of a general motion model for smooth camera movement, and solutions for monocular feature initialization and feature orientation estimation. Together, these add up to an extremely efficient and robust algorithm which runs at 30 Hz with standard PC and camera hardware. This work extends the range of robotic systems in which SLAM can be usefully applied, but also opens up new areas. We present applications of *MonoSLAM* to real-time 3D localization and mapping for a high-performance full-size humanoid robot and live augmented reality with a hand-held camera.

Index Terms—Autonomous vehicles, 3D/stereo scene analysis, tracking.

Simultaneous Localization and Mapping



Given a **single camera** feed,
estimate the 3D **position of the camera** and
the 3D **positions of all landmark** points in the world

Real-Time
Camera Tracking
in Unknown Scenes

MonoSLAM is just EKF!

$$P(\boldsymbol{x}_t|\boldsymbol{z}_{1:t}) \propto P(\boldsymbol{z}_t|\boldsymbol{x}_t) \int_{\boldsymbol{x}_{t-1}} P(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}) P(\boldsymbol{x}_{t-1}|\boldsymbol{z}_{1:t-1}) d\boldsymbol{x}_{t-1}$$

Step 1: Prediction

$$P(\boldsymbol{x}_t|\boldsymbol{z}_{1:t-1}) = \int_{\boldsymbol{x}_{t-1}} P(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}) P(\boldsymbol{x}_{t-1}|\boldsymbol{z}_{1:t-1}) d\boldsymbol{x}_{t-1}$$

Step 2: Update:

$$P(\boldsymbol{x}_t|\boldsymbol{z}_{1:t}) = P(\boldsymbol{z}_t|\boldsymbol{x}_t) P(\boldsymbol{x}_t|\boldsymbol{z}_{1:t-1})$$

MonoSLAM is just EKF!

$$P(\boldsymbol{x}_t|\boldsymbol{z}_{1:t}) \propto P(\boldsymbol{z}_t|\boldsymbol{x}_t) \int_{\boldsymbol{x}_{t-1}} P(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}) P(\boldsymbol{x}_{t-1}|\boldsymbol{z}_{1:t-1}) d\boldsymbol{x}_{t-1}$$

What is the state representation?

Step 1: Prediction

$$P(\boldsymbol{x}_t|\boldsymbol{z}_{1:t-1}) = \int_{\boldsymbol{x}_{t-1}} P(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}) P(\boldsymbol{x}_{t-1}|\boldsymbol{z}_{1:t-1}) d\boldsymbol{x}_{t-1}$$

Step 2: Update:

$$P(\boldsymbol{x}_t|\boldsymbol{z}_{1:t}) = P(\boldsymbol{z}_t|\boldsymbol{x}_t) P(\boldsymbol{x}_t|\boldsymbol{z}_{1:t-1})$$

What is the camera (robot) state?

What are the dimensions?

$$\mathbf{x}_c = \begin{bmatrix} \mathbf{r} \\ \mathbf{q} \\ \mathbf{v} \\ \omega \end{bmatrix}$$

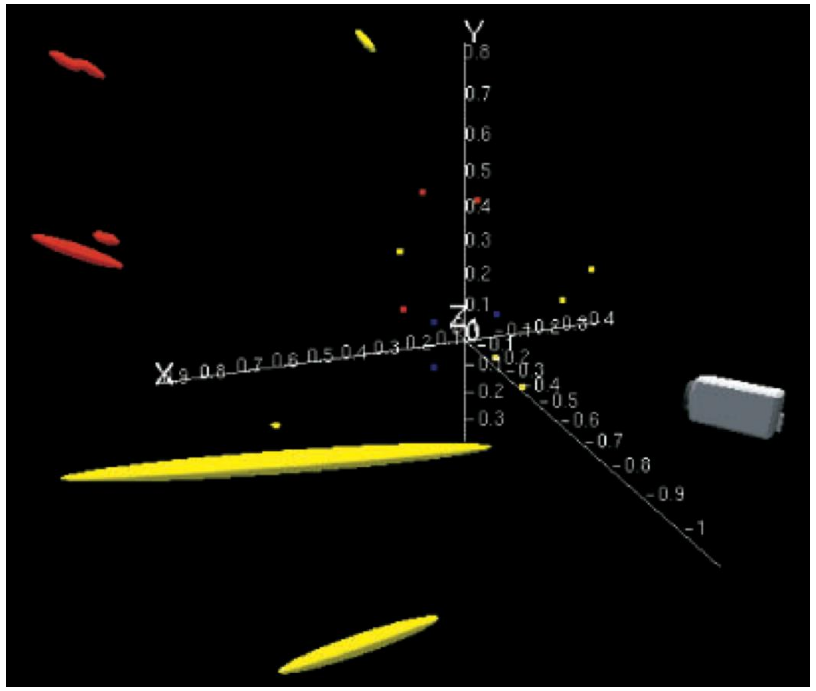
position

rotation (quaternion)

velocity

angular velocity

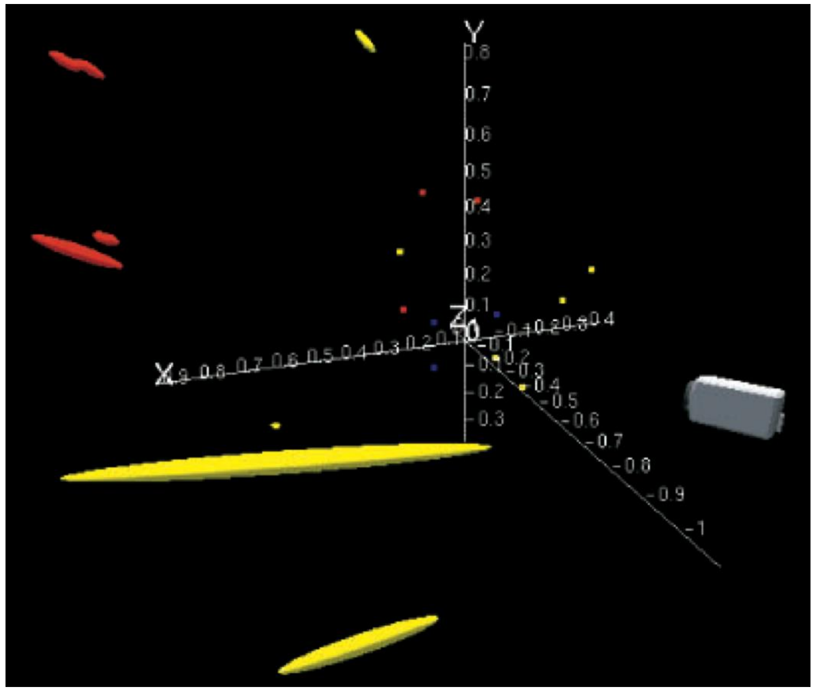
13 total



What is the camera (robot) state?

$$\mathbf{x}_c = \begin{bmatrix} \mathbf{r} \\ \mathbf{q} \\ \mathbf{v} \\ \omega \end{bmatrix}$$

	What are the dimensions?
position	3
rotation (quaternion)	4
velocity	3
angular velocity	3
	13 total



What is the world (robot+environment) state?

What are the dimensions?

$$\mathbf{x} = \begin{bmatrix} \mathbf{x}_c \\ \mathbf{y}_1 \\ \mathbf{y}_2 \\ \vdots \\ \mathbf{y}_N \end{bmatrix} \begin{array}{l} \text{state of the camera} \\ \text{location of feature 1} \\ \text{location of feature 2} \\ \\ \text{location of feature N} \end{array}$$

What is the world (robot+environment) state?

		What are the dimensions?
$$\mathbf{x} = \begin{bmatrix} \mathbf{x}_c \\ \mathbf{y}_1 \\ \mathbf{y}_2 \\ \vdots \\ \mathbf{y}_N \end{bmatrix}$$	state of the camera	13
	location of feature 1	3
	location of feature 2	3
	location of feature N	3
		13+3N total

What is the covariance (uncertainty) of the world state?

$$\boldsymbol{\Sigma} = \begin{bmatrix} \boldsymbol{\Sigma}_{\mathbf{x}_c\mathbf{x}_c} & \boldsymbol{\Sigma}_{\mathbf{x}_c\mathbf{y}_1} & \cdots & \boldsymbol{\Sigma}_{\mathbf{x}_c\mathbf{y}_N} \\ \boldsymbol{\Sigma}_{\mathbf{y}_1\mathbf{x}_c} & \boldsymbol{\Sigma}_{\mathbf{y}_1\mathbf{y}_1} & \cdots & \boldsymbol{\Sigma}_{\mathbf{y}_1\mathbf{y}_N} \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{\Sigma}_{\mathbf{y}_N\mathbf{x}_c} & \boldsymbol{\Sigma}_{\mathbf{y}_N\mathbf{y}_1} & \cdots & \boldsymbol{\Sigma}_{\mathbf{y}_N\mathbf{y}_N} \end{bmatrix}$$

What are the dimensions?

(13+3N) x (13+3N)

MonoSLAM is just EKF!

$$P(\boldsymbol{x}_t|\boldsymbol{z}_{1:t}) \propto P(\boldsymbol{z}_t|\boldsymbol{x}_t) \int_{\boldsymbol{x}_{t-1}} P(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}) P(\boldsymbol{x}_{t-1}|\boldsymbol{z}_{1:t-1}) d\boldsymbol{x}_{t-1}$$

What are the observations?

Step 1: Prediction

$$P(\boldsymbol{x}_t|\boldsymbol{z}_{1:t-1}) = \int_{\boldsymbol{x}_{t-1}} P(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}) P(\boldsymbol{x}_{t-1}|\boldsymbol{z}_{1:t-1}) d\boldsymbol{x}_{t-1}$$

Step 2: Update:

$$P(\boldsymbol{x}_t|\boldsymbol{z}_{1:t}) = P(\boldsymbol{z}_t|\boldsymbol{x}_t) P(\boldsymbol{x}_t|\boldsymbol{z}_{1:t-1})$$

MonoSLAM is just EKF!

$$P(\boldsymbol{x}_t|\boldsymbol{z}_{1:t}) \propto P(\boldsymbol{z}_t|\boldsymbol{x}_t) \int_{\boldsymbol{x}_{t-1}} P(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}) P(\boldsymbol{x}_{t-1}|\boldsymbol{z}_{1:t-1}) d\boldsymbol{x}_{t-1}$$

What are the observations?

Step 1: Prediction

$$P(\boldsymbol{x}_t|\boldsymbol{z}_{1:t-1}) = \int_{\boldsymbol{x}_{t-1}} P(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}) P(\boldsymbol{x}_{t-1}|\boldsymbol{z}_{1:t-1}) d\boldsymbol{x}_{t-1}$$

Step 2: Update:

$$P(\boldsymbol{x}_t|\boldsymbol{z}_{1:t}) = P(\boldsymbol{z}_t|\boldsymbol{x}_t) P(\boldsymbol{x}_t|\boldsymbol{z}_{1:t-1})$$

Observations are…

detected visual features of landmark points.
(e.g., Harris corners)

MonoSLAM is just EKF!

$$P(\boldsymbol{x}_t|\boldsymbol{z}_{1:t}) \propto P(\boldsymbol{z}_t|\boldsymbol{x}_t) \int_{\boldsymbol{x}_{t-1}} P(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}) P(\boldsymbol{x}_{t-1}|\boldsymbol{z}_{1:t-1}) d\boldsymbol{x}_{t-1}$$

Step 1: Prediction

$$\boxed{P(\boldsymbol{x}_t|\boldsymbol{z}_{1:t-1}) = \int_{\boldsymbol{x}_{t-1}} P(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}) P(\boldsymbol{x}_{t-1}|\boldsymbol{z}_{1:t-1}) d\boldsymbol{x}_{t-1}}$$

What does the prediction step look like?

Step 2: Update:

$$P(\boldsymbol{x}_t|\boldsymbol{z}_{1:t}) = P(\boldsymbol{z}_t|\boldsymbol{x}_t) P(\boldsymbol{x}_t|\boldsymbol{z}_{1:t-1})$$

What is the motion model? $P(\boldsymbol{x}_t|\boldsymbol{x}_{t-1})$

What is the form of the belief? $P(\boldsymbol{x}_t|\boldsymbol{z}_{1:t-1})$

What is the motion model? $P(\boldsymbol{x}_t | \boldsymbol{x}_{t-1})$

Landmarks:
constant position
(identity matrix)

Camera:
constant velocity
(not identity matrix and non-linear)

EKF!

What is the form of the belief? $P(\boldsymbol{x}_t | \boldsymbol{z}_{1:t-1})$

What is the motion model? $P(\boldsymbol{x}_t | \boldsymbol{x}_{t-1})$

Landmarks:
constant position
(identity matrix)

Camera:
constant velocity
(not identity matrix and non-linear)

What is the form of the belief? $P(\boldsymbol{x}_t | \boldsymbol{z}_{1:t-1})$

Gaussian!
(everything will be parametrized by a mean and variance)

Constant Velocity Motion Model

$$\mathbf{r}_t = \mathbf{r}_{t-1} + \mathbf{v}_{t-1}\Delta t$$ position

$$\mathbf{q}_t = \mathbf{q}_{t-1} \times [\mathbf{q}(\omega)\Delta t]$$ rotation (quaternion)

$$\mathbf{v}_t = \mathbf{v}_{t-1}$$ velocity

$$\omega_t = \omega_{t-1}$$ angular velocity

Gaussian noise uncertainty (only on velocity)

$$\mathbf{v}_t = \mathbf{v}_{t-1} + \mathbf{V}$$
$$\omega_t = \omega_{t-1} + \mathbf{\Omega}$$

$$\mathbf{V} \sim \mathcal{N}(\mathbf{0}, \begin{bmatrix} \sigma_v & 0 & 0 \\ 0 & \sigma_v & 0 \\ 0 & 0 & \sigma_v \end{bmatrix})$$

$$\mathbf{\Omega} \sim \mathcal{N}(\mathbf{0}, \begin{bmatrix} \sigma_w & 0 & 0 \\ 0 & \sigma_w & 0 \\ 0 & 0 & \sigma_w \end{bmatrix})$$

Prediction (mean of camera state):

$$P(\boldsymbol{x}_t|\boldsymbol{z}_{1:t-1}) = \int_{\boldsymbol{x}_{t-1}} P(\boldsymbol{x}_t|\boldsymbol{x}_{t-1})P(\boldsymbol{x}_{t-1}|\boldsymbol{z}_{1:t-1})d\boldsymbol{x}_{t-1}$$

$$\mathbf{f}_t = \begin{bmatrix} \mathbf{r}_t \\ \mathbf{q}_t \\ \mathbf{v}_t \\ \omega_t \end{bmatrix} = \begin{bmatrix} \mathbf{r}_{t-1} + \mathbf{v}_{t-1}\Delta t \\ \mathbf{q}_{t-1} + \mathbf{q}(\omega)_{t-1}\Delta t \\ \mathbf{v}_{t-1} \\ \omega_{t-1} \end{bmatrix}$$

Prediction (covariance of camera state):

$$P(\boldsymbol{x}_t|\boldsymbol{z}_{1:t-1}) = \int_{\boldsymbol{x}_{t-1}} P(\boldsymbol{x}_t|\boldsymbol{x}_{t-1})P(\boldsymbol{x}_{t-1}|\boldsymbol{z}_{1:t-1})d\boldsymbol{x}_{t-1}$$

$$\bar{\boldsymbol{\Sigma}}_{\mathbf{xx}} = \boxed{\frac{\partial \mathbf{f}_t}{\partial \mathbf{x}}} \boldsymbol{\Sigma}_{\mathbf{xx}} \frac{\partial \mathbf{f}_t}{\partial \mathbf{x}}^\top + \mathbf{Q}_t$$

new covariance | change around new state | old covariance | change around new state | system noise (process noise)

Where does this motion model approximation come from?

change in
position

$$\boxed{\frac{\partial \mathbf{f}_t}{\partial \mathbf{x}_{t-1}}} = \begin{bmatrix} \boxed{\frac{\partial \mathbf{r}_t}{\partial \mathbf{r}_{t-1}}} & \frac{\partial \mathbf{q}_t}{\partial \mathbf{r}_{t-1}} & \frac{\partial \mathbf{v}_t}{\partial \mathbf{r}_{t-1}} & \frac{\partial \omega_t}{\partial \mathbf{r}_{t-1}} \\ \frac{\partial \mathbf{r}_t}{\partial \mathbf{q}_{t-1}} & \frac{\partial \mathbf{q}_t}{\partial \mathbf{q}_{t-1}} & \frac{\partial \mathbf{v}_t}{\partial \mathbf{q}_{t-1}} & \frac{\partial \omega_t}{\partial \mathbf{q}_{t-1}} \\ \frac{\partial \mathbf{r}_t}{\partial \mathbf{v}_{t-1}} & \frac{\partial \mathbf{q}_t}{\partial \mathbf{v}_{t-1}} & \frac{\partial \mathbf{v}_t}{\partial \mathbf{v}_{t-1}} & \frac{\partial \omega_t}{\partial \mathbf{v}_{t-1}} \\ \frac{\partial \mathbf{r}_t}{\partial \omega_{t-1}} & \frac{\partial \mathbf{q}_t}{\partial \omega_{t-1}} & \frac{\partial \mathbf{v}_t}{\partial \omega_{t-1}} & \frac{\partial \omega_t}{\partial \omega_{t-1}} \end{bmatrix}$$

change in
camera state

What are the dimensions?

Skipping over many details…

$$\frac{\partial \mathbf{f}_t}{\partial \mathbf{x}_{t-1}} = \begin{bmatrix} \mathbf{I} & \mathbf{0} & \mathbf{I}\Delta t & \mathbf{0} \\ \mathbf{0} & \frac{\partial \mathbf{q}_t}{\partial \mathbf{q}_{t-1}} & \mathbf{0} & \frac{\partial \omega_t}{\partial \mathbf{q}_{t-1}} \\ \mathbf{0} & \mathbf{0} & \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I} \end{bmatrix}$$

Prediction (covariance of camera state):

$$P(\boldsymbol{x}_t|\boldsymbol{z}_{1:t-1}) = \int_{\boldsymbol{x}_{t-1}} P(\boldsymbol{x}_t|\boldsymbol{x}_{t-1})P(\boldsymbol{x}_{t-1}|\boldsymbol{z}_{1:t-1})d\boldsymbol{x}_{t-1}$$

$$\underbrace{\bar{\boldsymbol{\Sigma}}_{\mathbf{xx}}}_{\text{new covariance}} = \underbrace{\boxed{\frac{\partial \mathbf{f}_t}{\partial \mathbf{x}}}}_{\text{change around new state}} \underbrace{\boldsymbol{\Sigma}_{\mathbf{xx}}}_{\text{old covariance}} \underbrace{\frac{\partial \mathbf{f}_t}{\partial \mathbf{x}}^\top}_{\text{change around new state}} + \underbrace{\mathbf{Q}_t}_{\text{system noise (process noise)}}$$

Bit of a pain to compute this term…

We just covered the **prediction** step for the camera state

$$P(\boldsymbol{x}_t|\boldsymbol{z}_{1:t-1}) = \int_{\boldsymbol{x}_{t-1}} P(\boldsymbol{x}_t|\boldsymbol{x}_{t-1})P(\boldsymbol{x}_{t-1}|\boldsymbol{z}_{1:t-1})d\boldsymbol{x}_{t-1}$$

$$\mathbf{f}_t = \begin{bmatrix} \mathbf{r}_t \\ \mathbf{q}_t \\ \mathbf{v}_t \\ \omega_t \end{bmatrix} = \begin{bmatrix} \mathbf{r}_{t-1} + \mathbf{v}_{t-1} \\ \mathbf{q}_{t-1} + \mathbf{q}(\omega)_{t-1} \\ \mathbf{v}_{t-1} \\ \omega_{t-1} \end{bmatrix}$$

$$\bar{\boldsymbol{\Sigma}}_{\mathbf{xx}} = \frac{\partial \mathbf{f}_t}{\partial \mathbf{x}} \boldsymbol{\Sigma}_{\mathbf{xx}} \frac{\partial \mathbf{f}_t}{\partial \mathbf{x}}^\top + \mathbf{Q}_t$$

Now we need to do the **update** step!

General Filtering Equations

$$P(\boldsymbol{x}_t|\boldsymbol{z}_{1:t}) \propto P(\boldsymbol{z}_t|\boldsymbol{x}_t) \int_{\boldsymbol{x}_{t-1}} P(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}) P(\boldsymbol{x}_{t-1}|\boldsymbol{z}_{1:t-1}) d\boldsymbol{x}_{t-1}$$

Prediction:

$$P(\boldsymbol{x}_t|\boldsymbol{z}_{1:t-1}) = \int_{\boldsymbol{x}_{t-1}} P(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}) P(\boldsymbol{x}_{t-1}|\boldsymbol{z}_{1:t-1}) d\boldsymbol{x}_{t-1}$$

Update:

$$P(\boldsymbol{x}_t|\boldsymbol{z}_{1:t}) = P(\boldsymbol{z}_t|\boldsymbol{x}_t) P(\boldsymbol{x}_t|\boldsymbol{z}_{1:t-1})$$

Belief state — State observation — Predicted State

$$P(\boldsymbol{x}_t|\boldsymbol{z}_{1:t}) = P(\boldsymbol{z}_t|\boldsymbol{x}_t)P(\boldsymbol{x}_t|\boldsymbol{z}_{1:t-1})$$

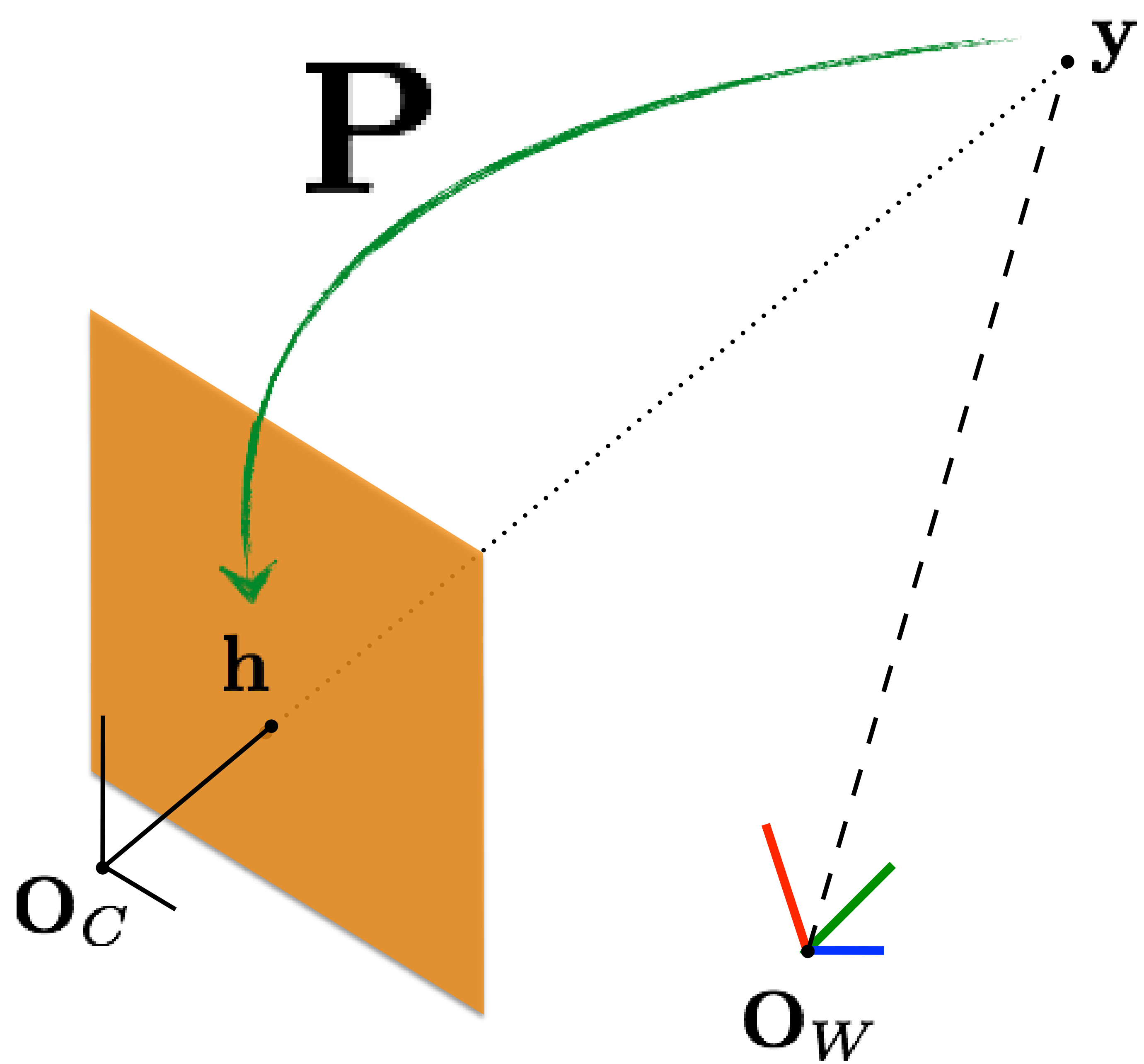
What are the observations?

2D projections of 3D landmarks

Recall, the state includes the 3D location of landmarks

$$\mathbf{x} = \begin{bmatrix} \mathbf{x}_c \\ \mathbf{y}_1 \\ \mathbf{y}_2 \\ \vdots \\ \mathbf{y}_N \end{bmatrix}$$

What is the projection from 3D point to 2D image point?



Observation Model

$$P(\boldsymbol{z}_t|\boldsymbol{x}_t)$$

If you know the 3D location of a landmark, what is the 2D projection?

Non-linear observation model

$$\mathbf{h} \sim \mathbf{P}\ \mathbf{y}$$

2D Image Point

Camera matrix

3D World Point

$$\mathbf{P} = \mathbf{K}\,[\mathbf{R}|\mathbf{T}]$$

*What do we know about **P**?*

How do we make the observation model linear?

$$H = \frac{\partial \mathbf{h}}{\partial \mathbf{x}}$$

(2n x 13)

n: number of visible points

I will spare you the pain of deriving the partial derivative…

$$P(\boldsymbol{x}_t|\boldsymbol{z}_{1:t}) = P(\boldsymbol{z}_t|\boldsymbol{x}_t)P(\boldsymbol{x}_t|\boldsymbol{z}_{1:t-1})$$

Update step (mean):

$$\underset{\text{Updated state}}{\mathbf{x}_t} = \underset{\text{Predicted state}}{\mathbf{x}_t} + \overset{\text{Kalman gain}}{\mathbf{K}_t}(\underset{\text{Matched 2D features}}{\mathbf{z}_t} - \underset{\text{2D projection of 3D point}}{\mathbf{h}(\mathbf{y};\mathbf{x}_t)})$$

Update step (covariance):

$$\underset{\text{Covariance (updated)}}{\mathbf{\Sigma}_t} = (\overset{\text{Identity}}{\mathbf{I}} - \overset{\text{Kalman gain}}{\mathbf{K}_t}\underset{\text{Jacobian}}{\mathbf{H}_t})\underset{\text{Covariance (predicted)}}{\mathbf{\Sigma}_t}$$

Kintinuous: Spatially Extended Kinect Fusion

Thomas Whelan, John McDonald
National University of Ireland Maynooth, Ireland

Michael Kaess, Maurice Fallon, Hordur Johannsson, John J. Leonard
Computer Science and Artificial Intelligence Laboratory, MIT, USA

NUI MAYNOOTH

CSAIL

References

Basic reading:

- Szeliski, Appendix B.