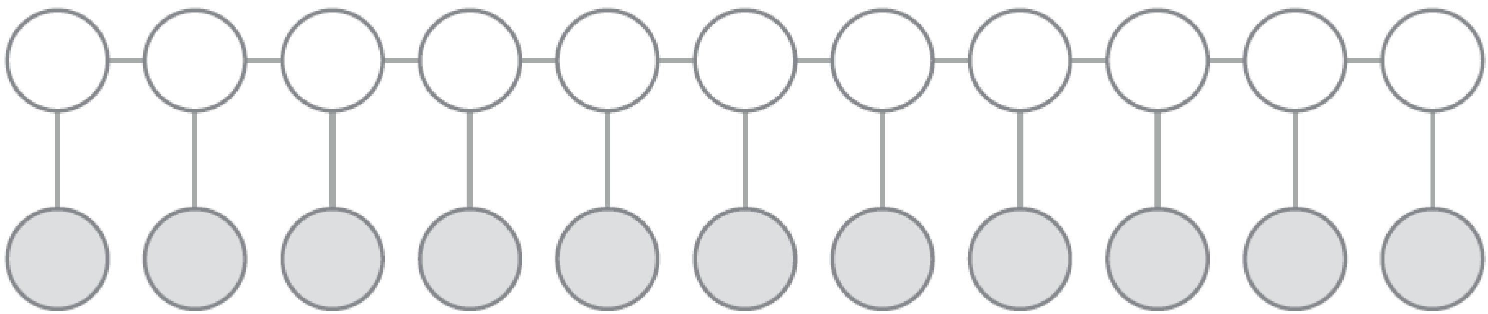


Temporal State Models

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

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

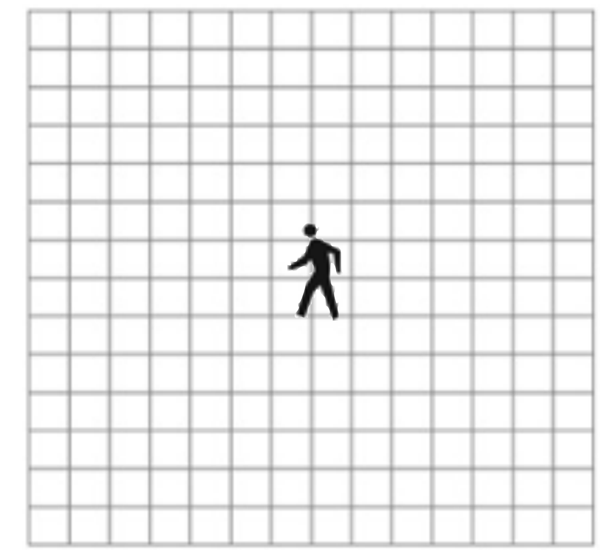
$\boldsymbol{X} = \{\boldsymbol{x}, \dot{\boldsymbol{x}}, \boldsymbol{f}_1, \ldots, \boldsymbol{f}_n\}$

position, velocity and
location of landmarks

Object tracking (localization)

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

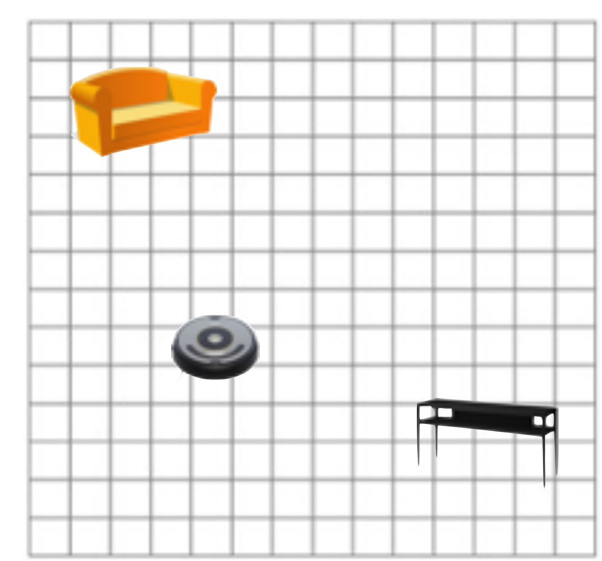
e.g., location on the ground plane



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:

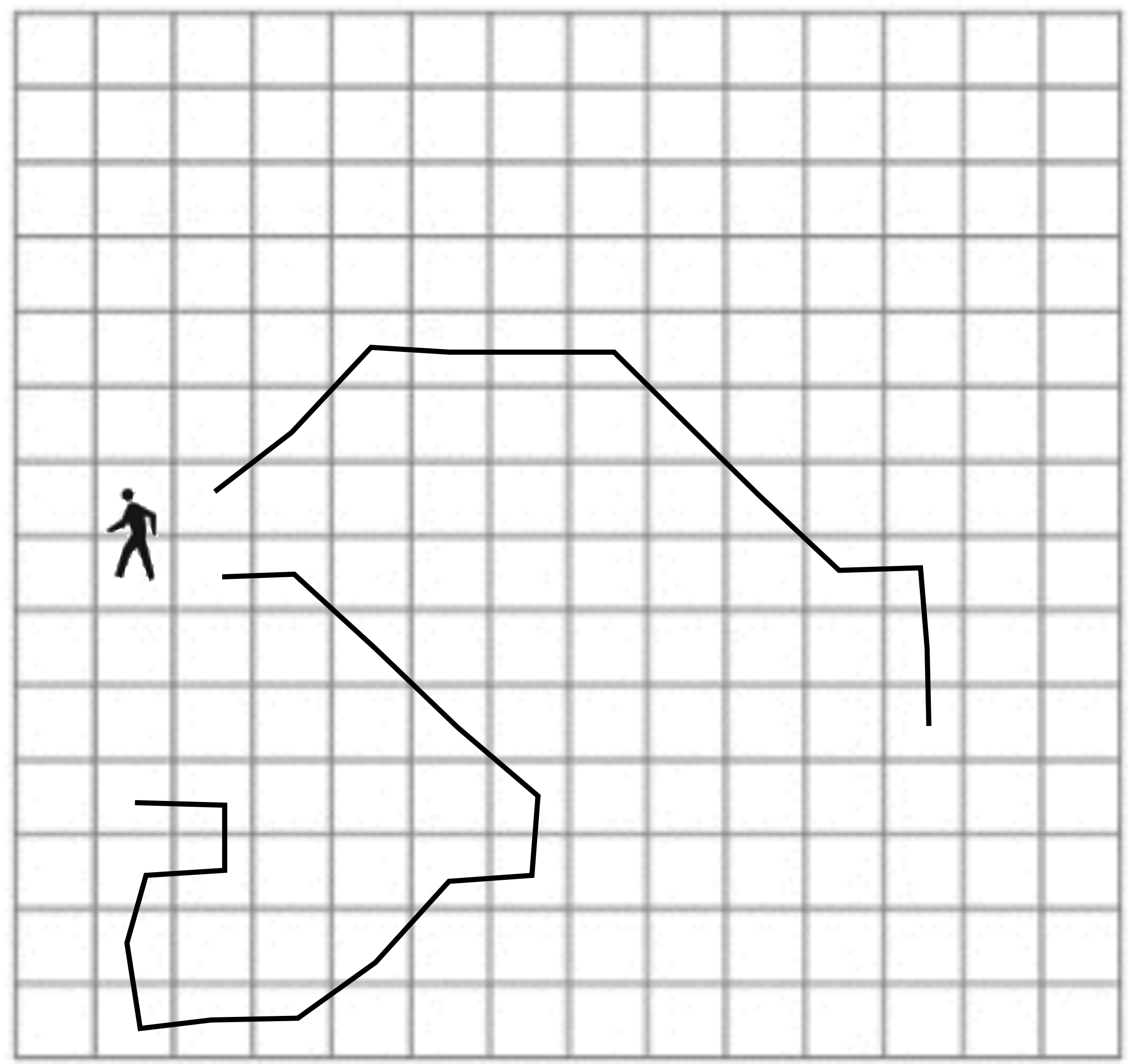
$$\boldsymbol{X}_0, \boldsymbol{X}_1, \ldots, \boldsymbol{X}_t$$

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

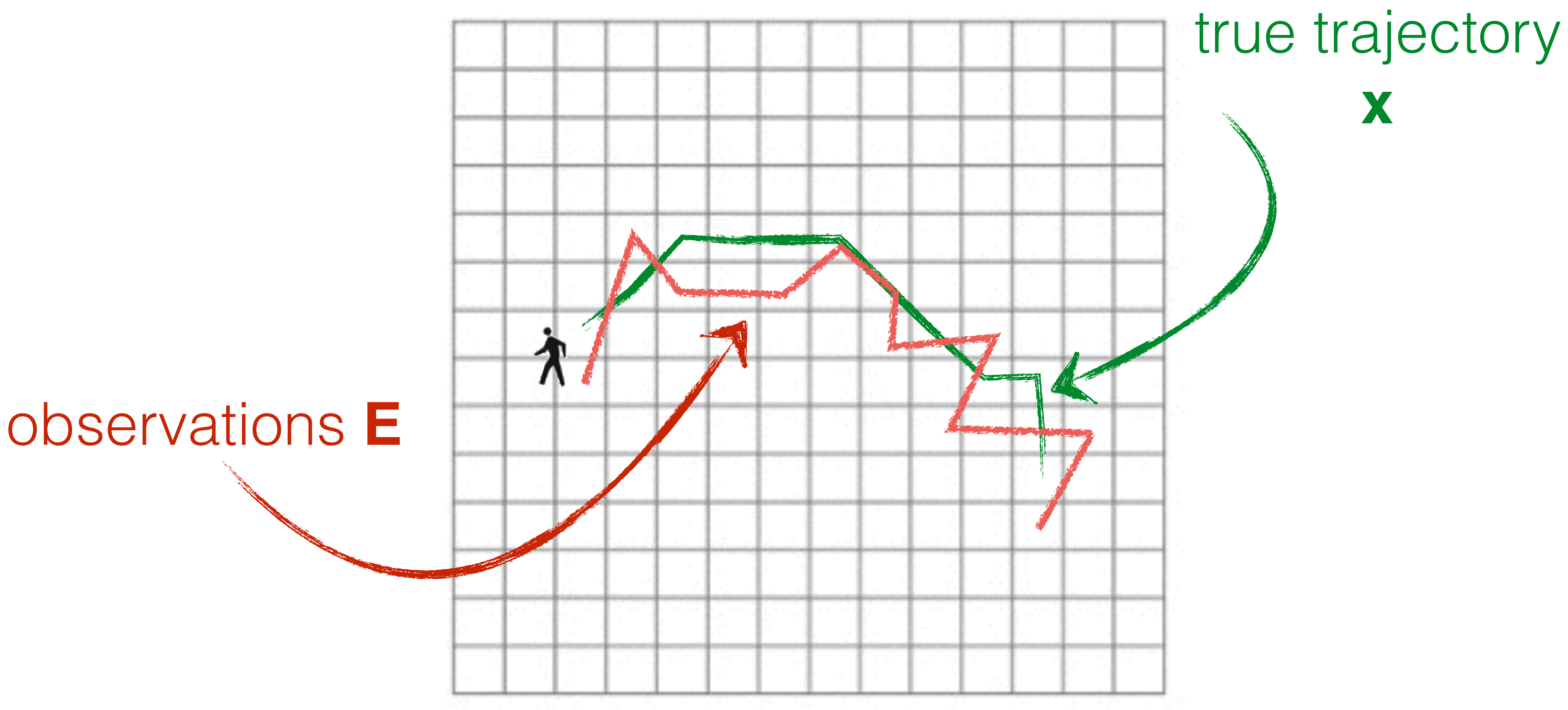
$$\boldsymbol{E}_t$$

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

(**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

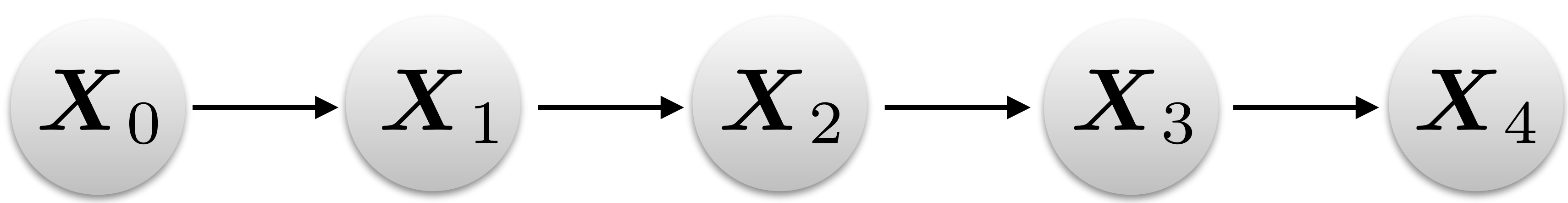
Only have to store **one** model.

Is this a reasonable assumption?

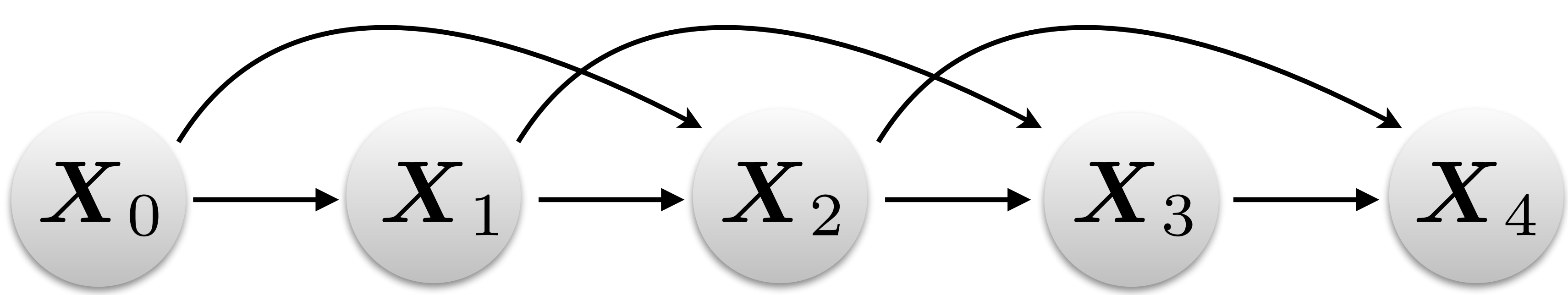
Reduction 2. Markov Assumption:

'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.’

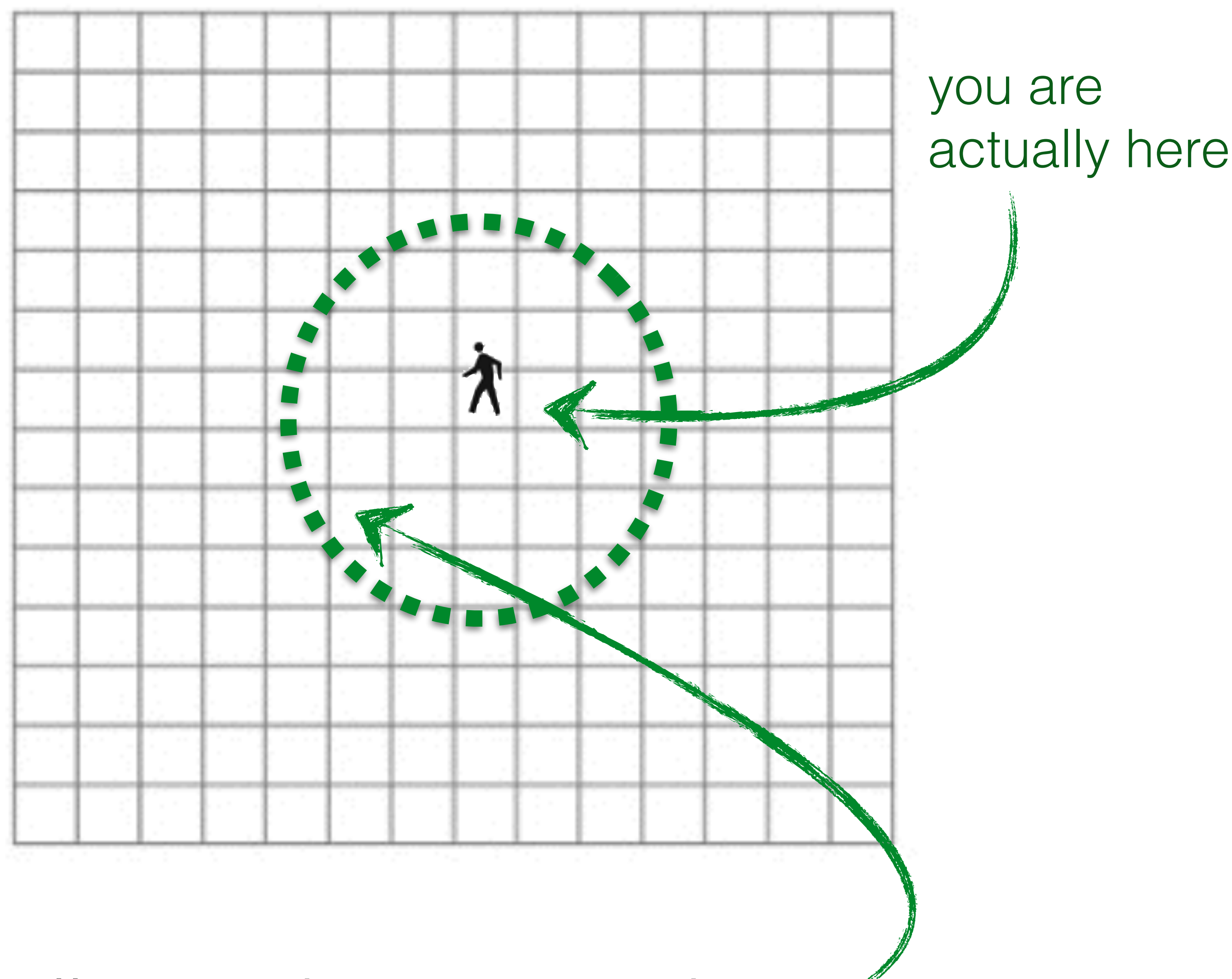
The current observation is usually
most influenced by the current state

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

(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.



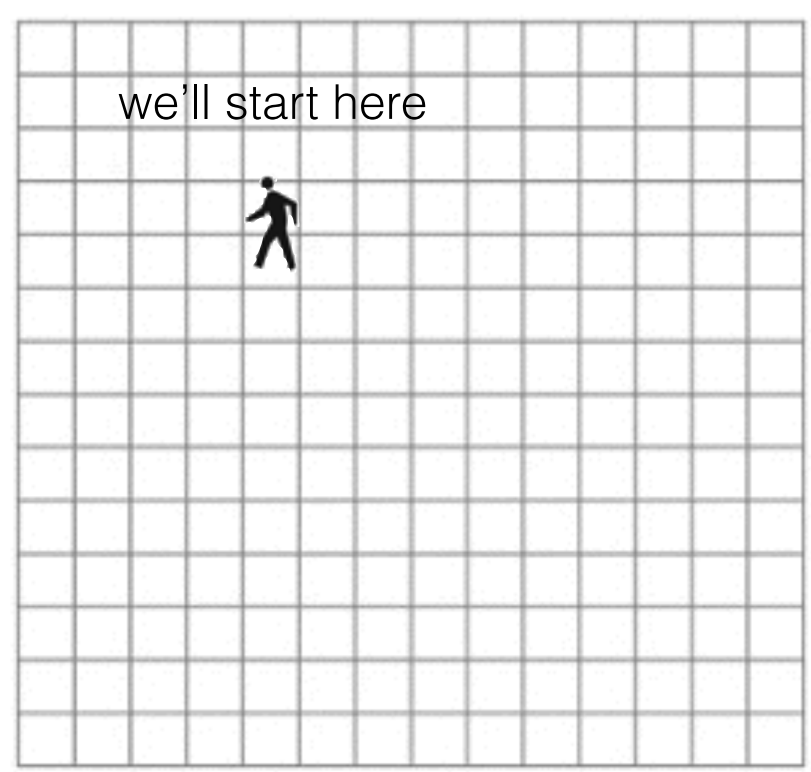
But GPS tells you that you are here

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

Reduction 3. Prior State Assumption:

‘we know where the process (probably) starts’

X_0



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	motion model	sensor model
prior	transition 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

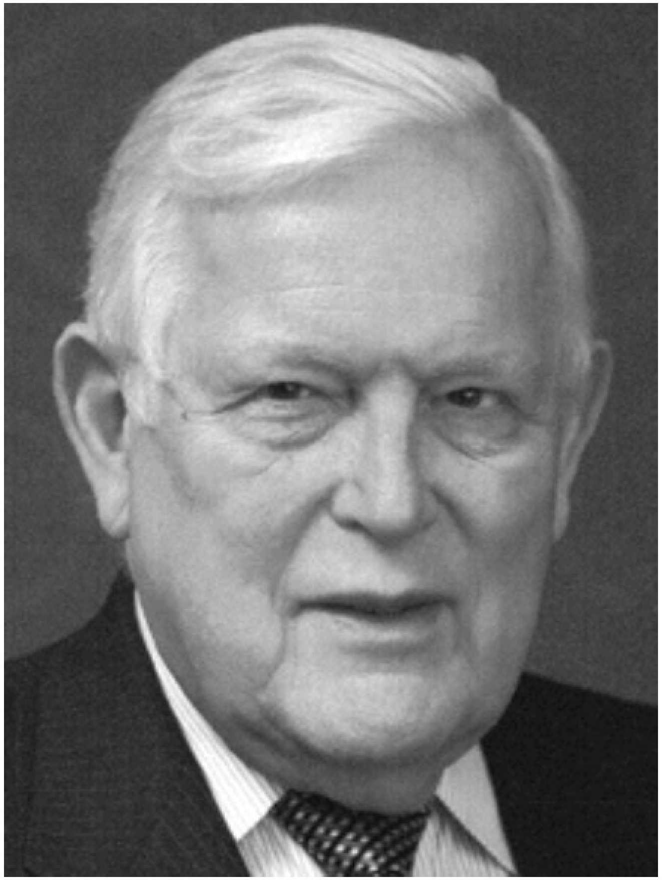
specific instances of a DBN
covered in this class

Hidden Markov Model

(typically taught as discrete but not necessarily)

Kalman Filter

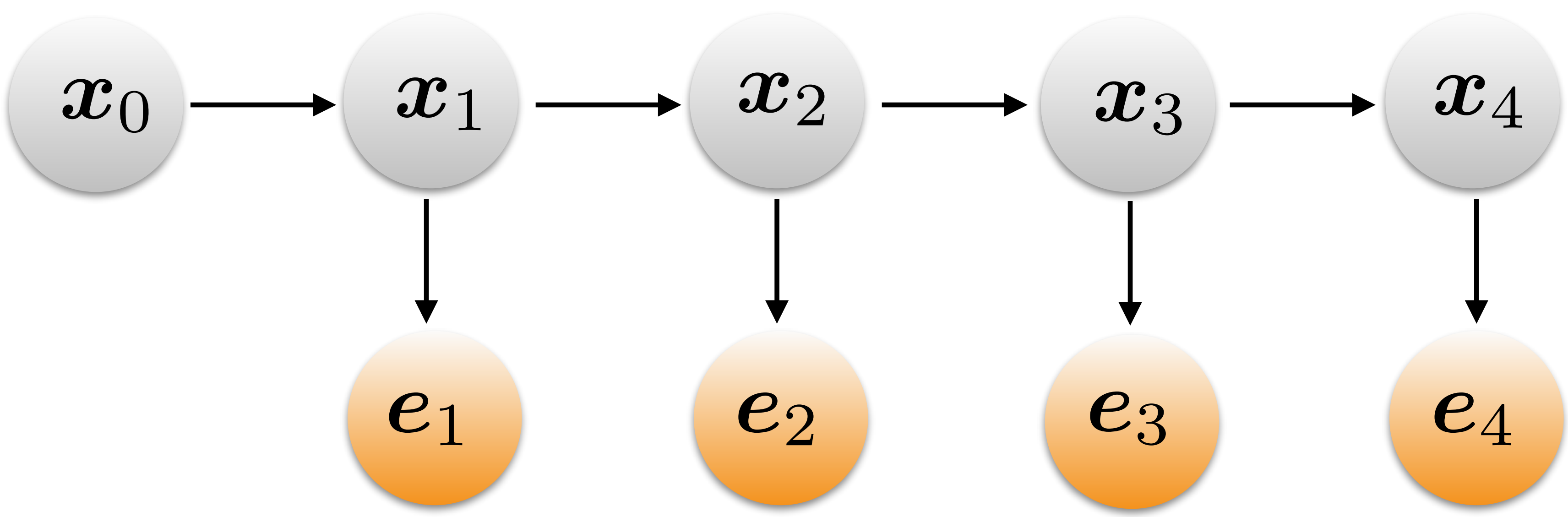
(Gaussian motion model, prior and observation model)



Kalman Filter

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

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

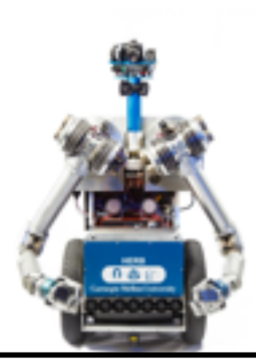
Kalman Filtering

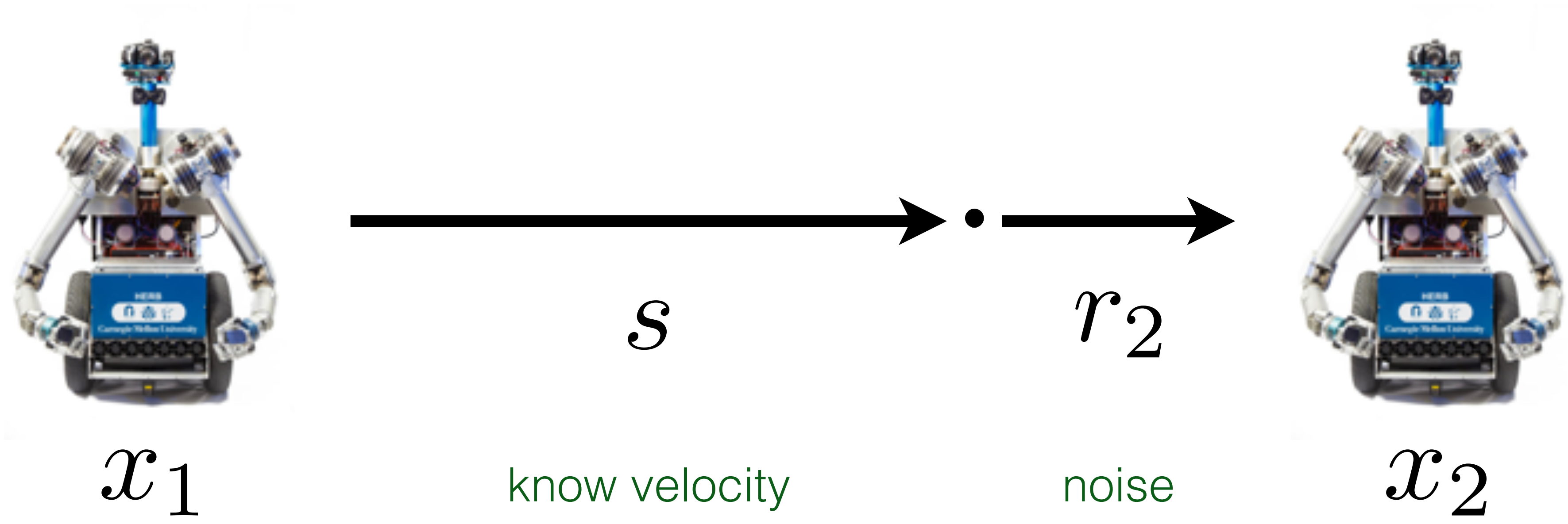
$$P(\boldsymbol{X}_{t+1}|\boldsymbol{e}_{1:t+1}) \propto \overbrace{\underbrace{P(\boldsymbol{e}_{t+1}|\boldsymbol{X}_{t+1})}_{\text{observation model}}}^{\text{Gaussian}} \int_{\boldsymbol{x}_t} \overbrace{\underbrace{P(\boldsymbol{X}_{t+1}|\boldsymbol{x}_t)}_{\text{motion model}}}^{\text{Gaussian}}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

integral because
continuous PDFs

Simple, 1D example…

x





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

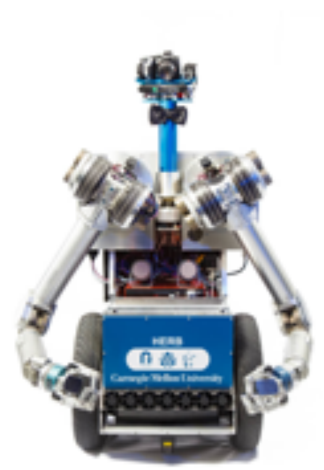
$$r_t \sim \mathcal{N}(0, \sigma_R)$$

'sampled from'

System (motion) model

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

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



x_1

s $\longrightarrow \bullet \longrightarrow$ r_2

know velocity

noise

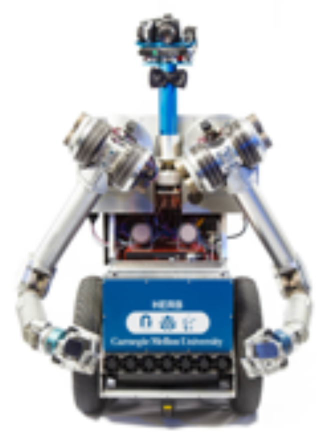
x_2

How do you represent the motion model?

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

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

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

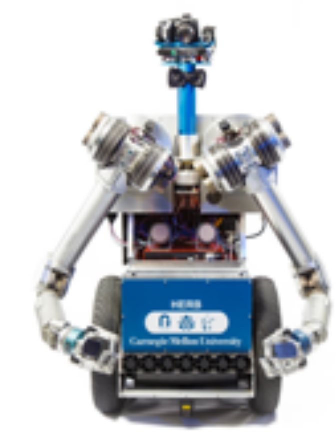


x_1

$\xrightarrow{s} \bullet \xrightarrow{r_2}$

know velocity

noise



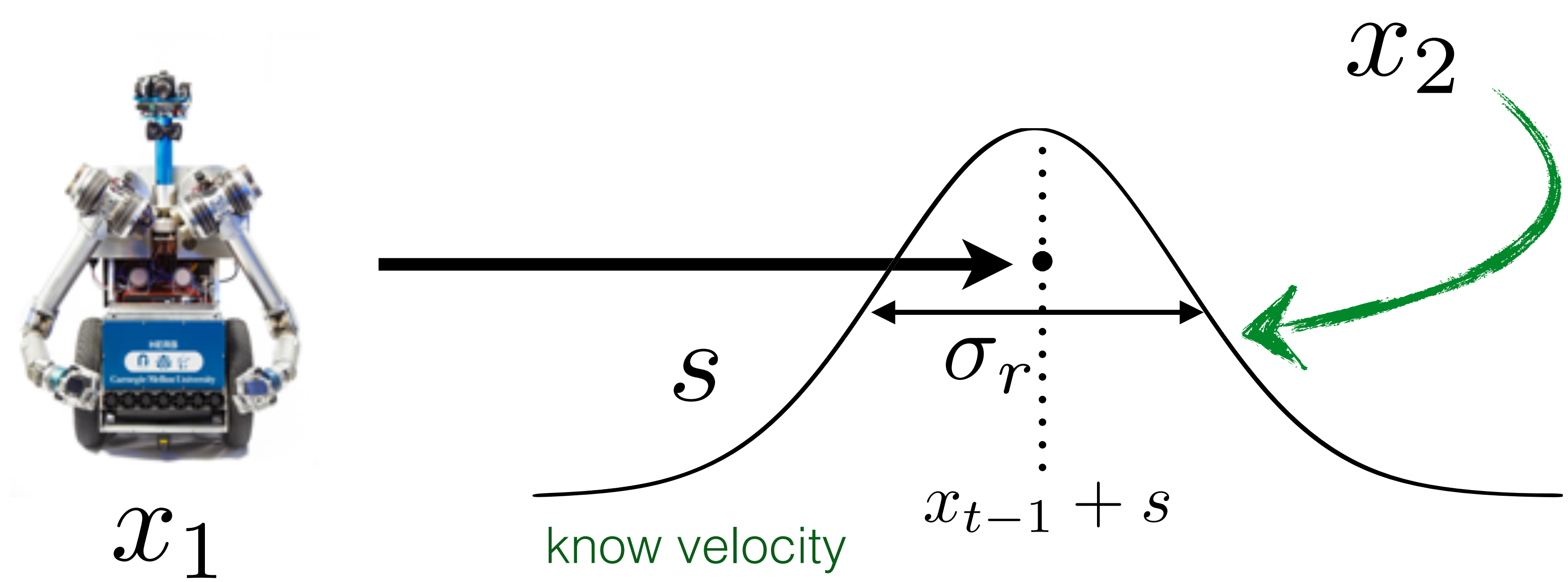
x_2

How do you represent the motion model?

A linear Gaussian (continuous) transition model

$$P(x_t | x_{t-1}) = \mathcal{N}(x_t; \underbrace{x_{t-1} + s}_{\text{mean}}, \underbrace{\sigma_r}_{\text{standard deviation}})$$

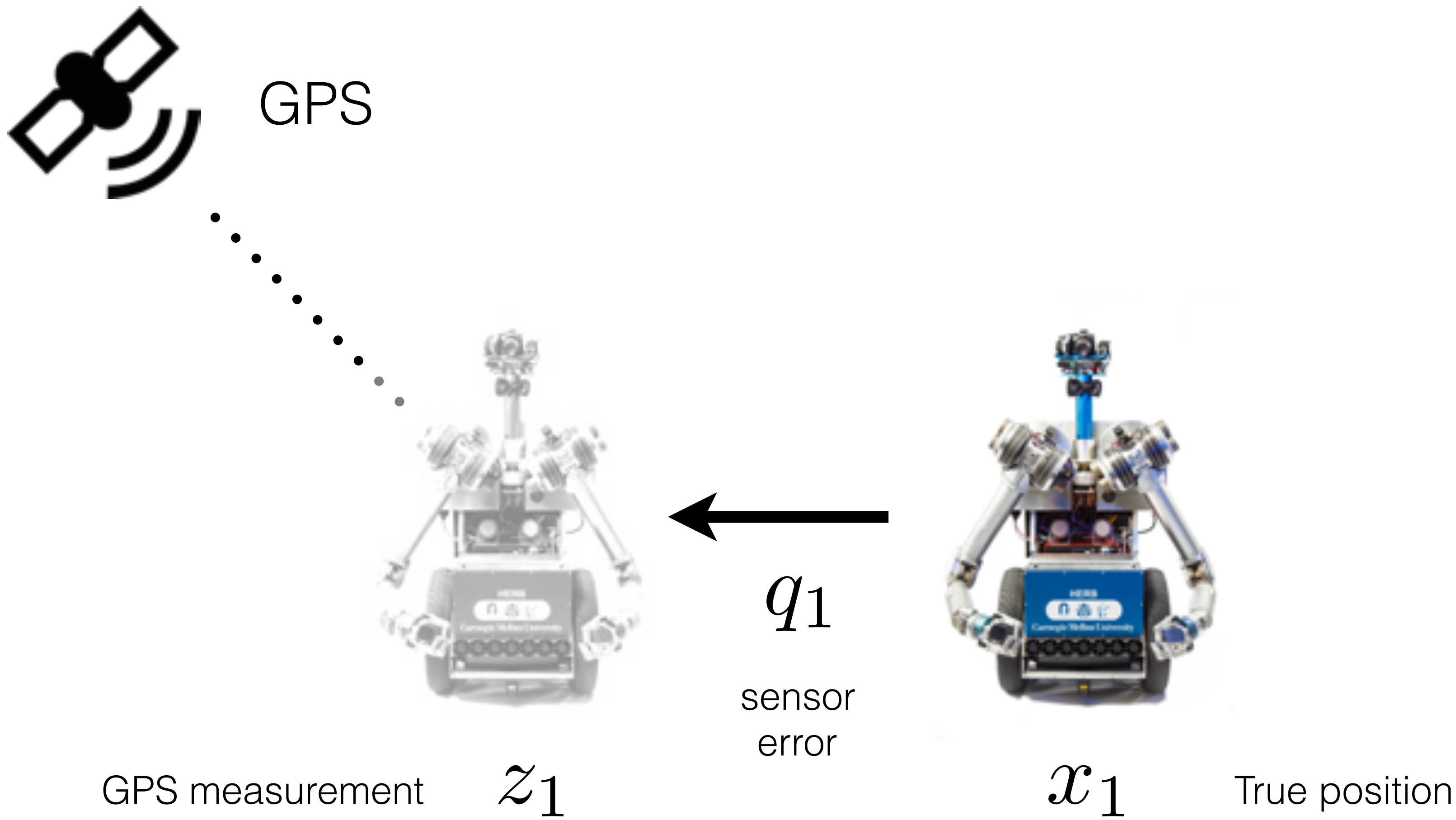
How can you visualize this distribution?



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?

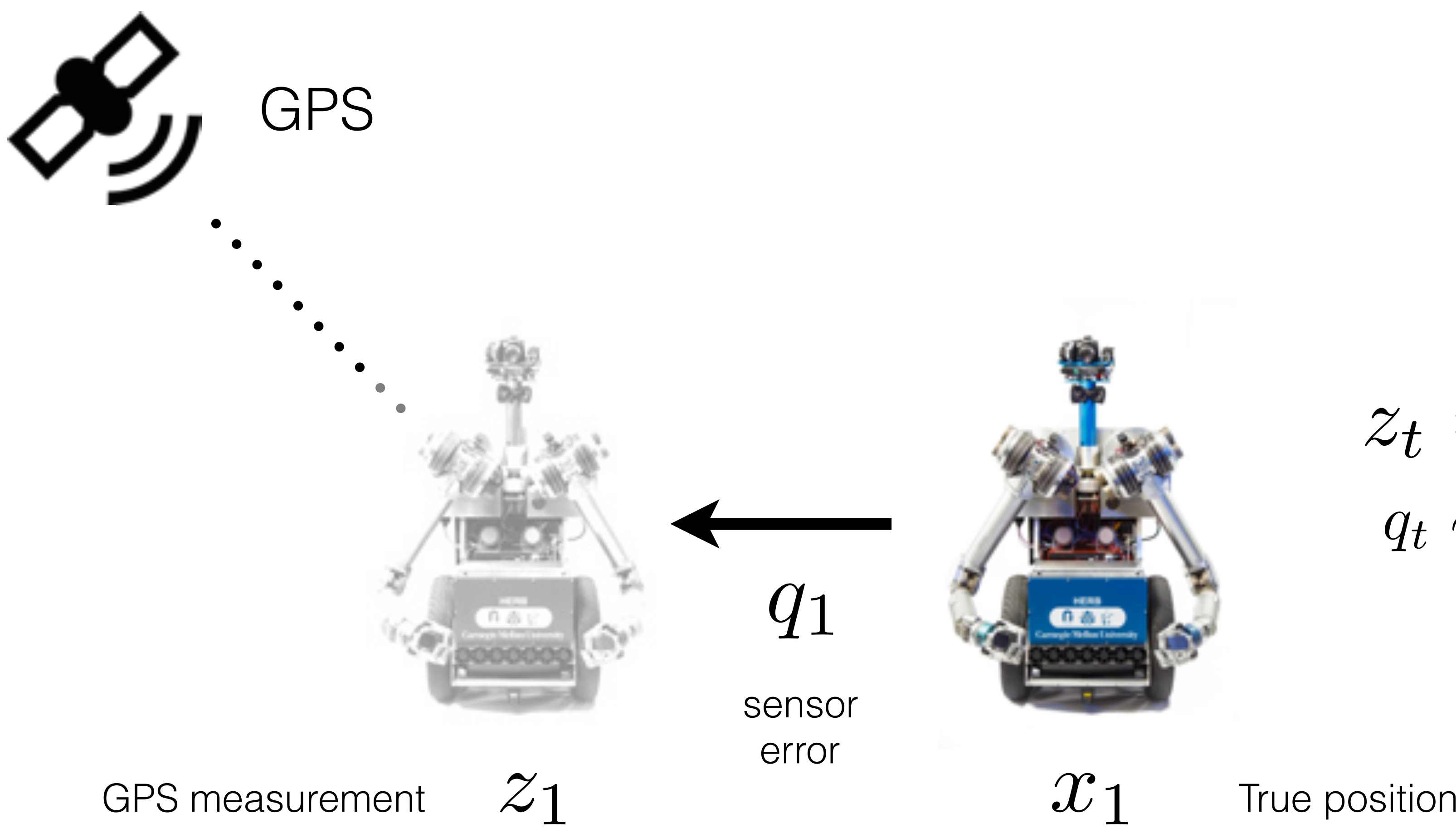


$$z_t = x_t + q_t$$

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

sampled from a Gaussian

Observation (measurement) model

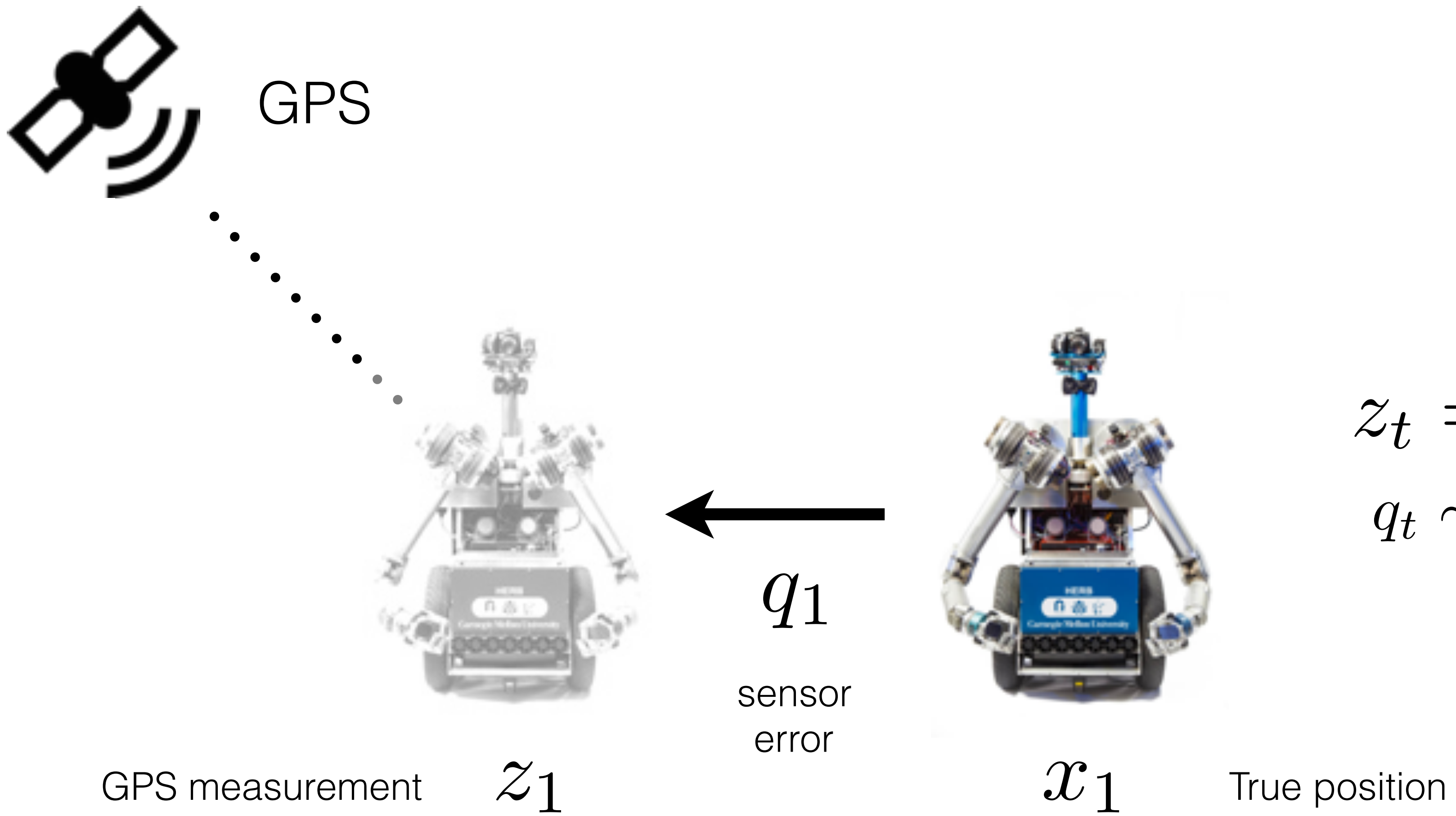


$$z_t = x_t + q_t$$
$$q_t \sim \mathcal{N}(0, \sigma_Q)$$

How do you represent the observation (measurement) model?

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

e represents z

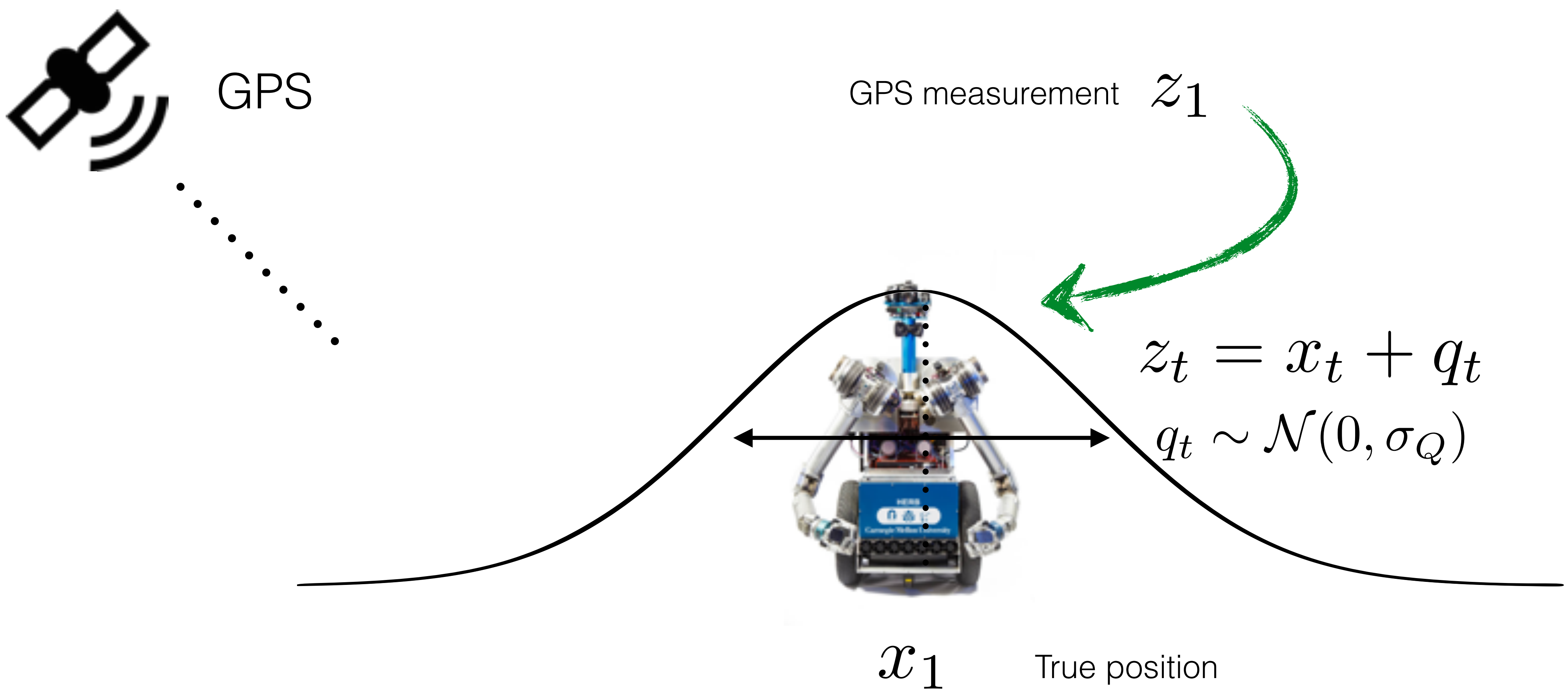


$$z_t = x_t + q_t$$
$$q_t \sim \mathcal{N}(0, \sigma_Q)$$

How do you represent the observation (measurement) model?

Also a linear Gaussian model

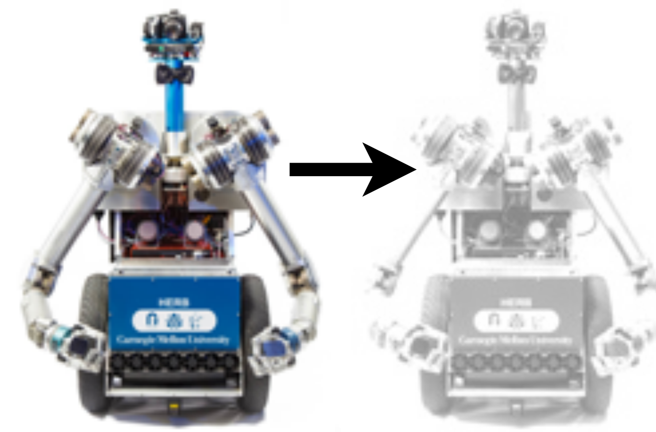
$$P(z_t | x_t) = \mathcal{N}(z_t; x_t, \sigma_Q)$$



How do you represent the observation (measurement) model?

Also a linear Gaussian model

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

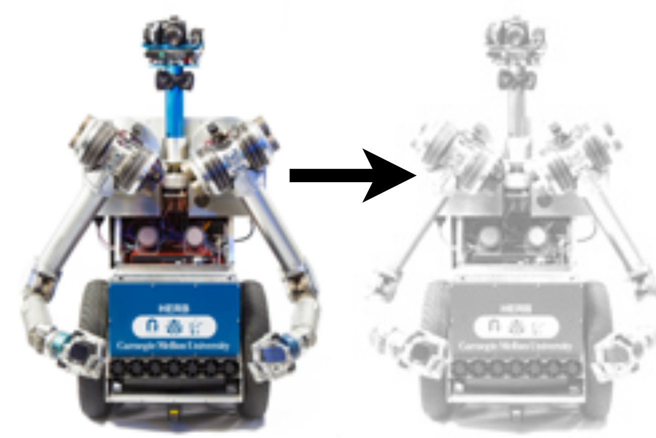


x_0 $\hat{x}_0$

true position initial estimate

initial estimate uncertainty σ_0

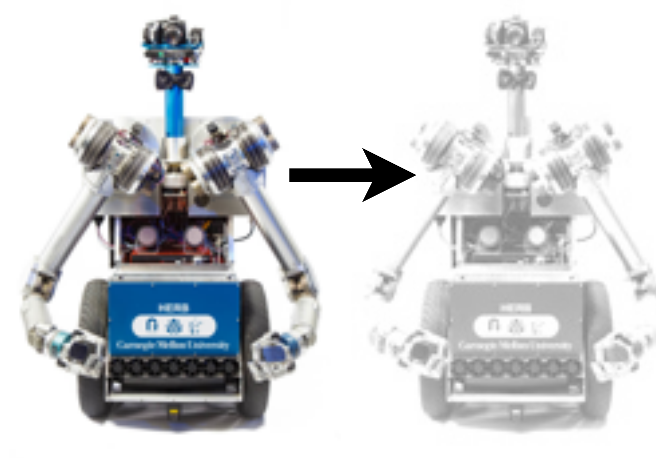
Prior (initial) State



x_0 $\hat{x}_0$

true position initial estimate

How do you represent the prior state probability?



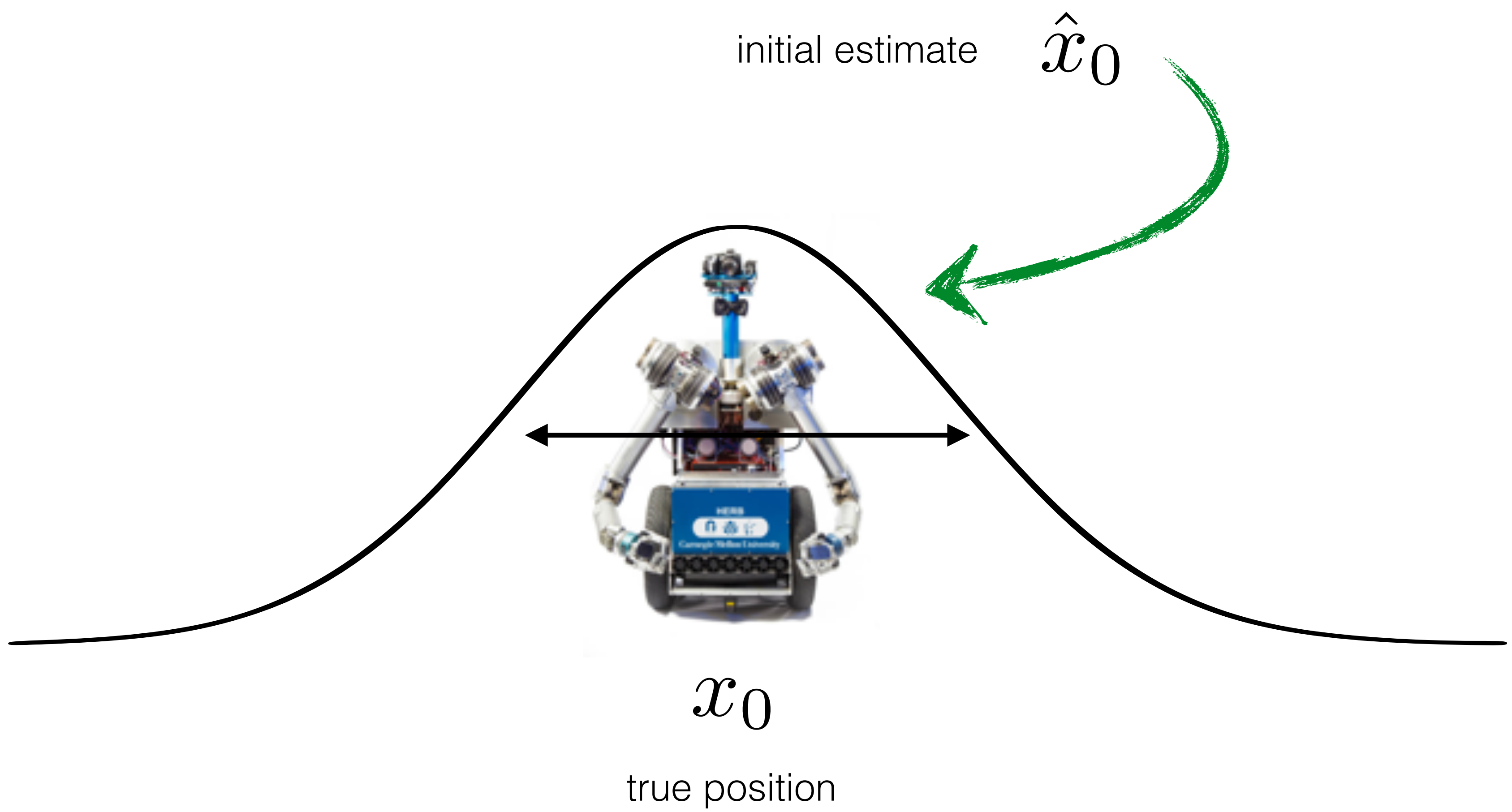
x_0 $\hat{x}_0$

true position 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'

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

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)

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

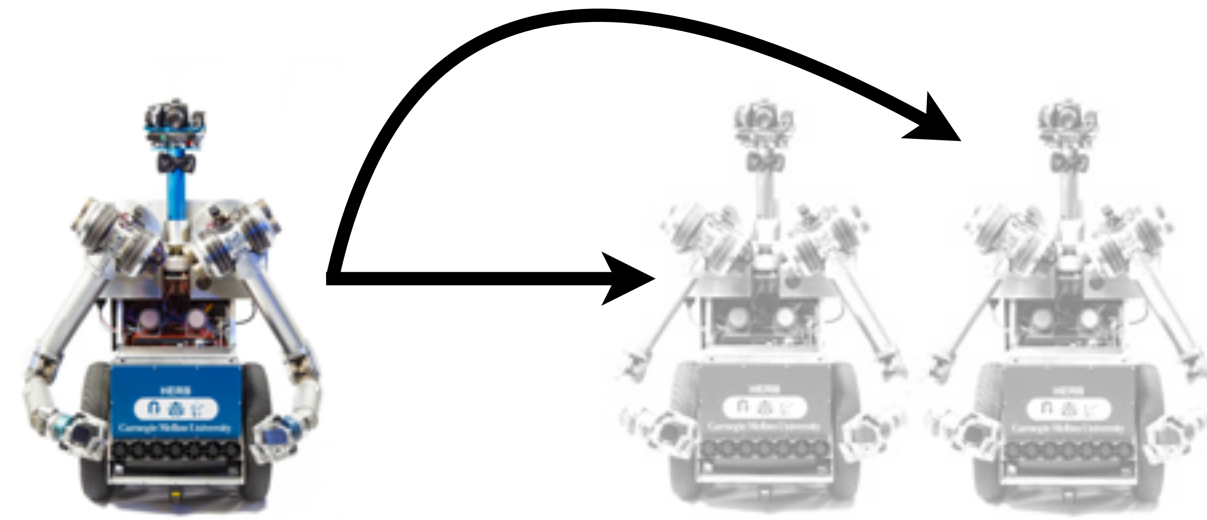
the second step after prediction is …

… update step!

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

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:

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

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…

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

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…

$$\hat{x}_1^+ = \hat{x}_1 + \frac{\sigma_1^2}{\sigma_q^2 + \sigma_1^2}(z_1 - \hat{x}_1) = \hat{x}_1 + \underset{\text{'Kalman gain'}}{K}\underset{\text{'Innovation'}}{(z_1 - \hat{x}_1)}$$

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 + r;

  K = v/(v + q);

  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+r)/((v+r)+q)*(z-(x+s));
  p   = (v+r)-(v+r)/((v+r)+q)*v;
```

Bare computations (algorithm) of Bayesian filtering:

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

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!)

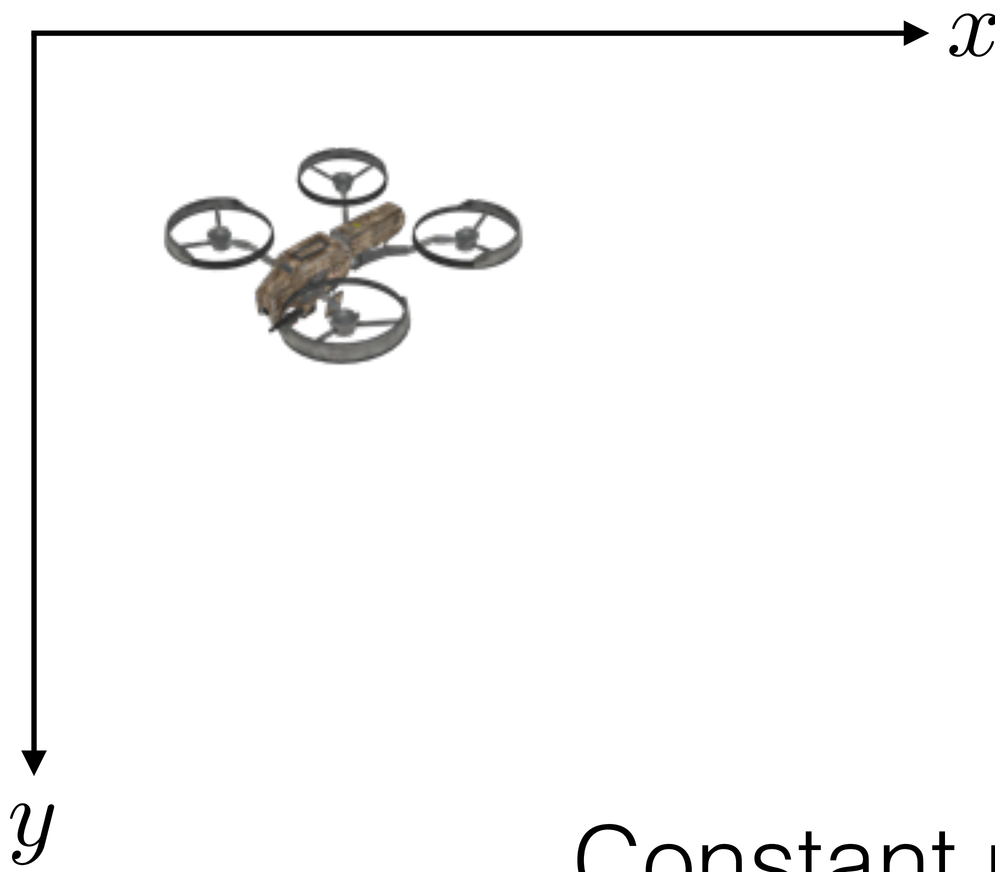
```
[x P] = KF(x,P,z)

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

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

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

2D Example



state

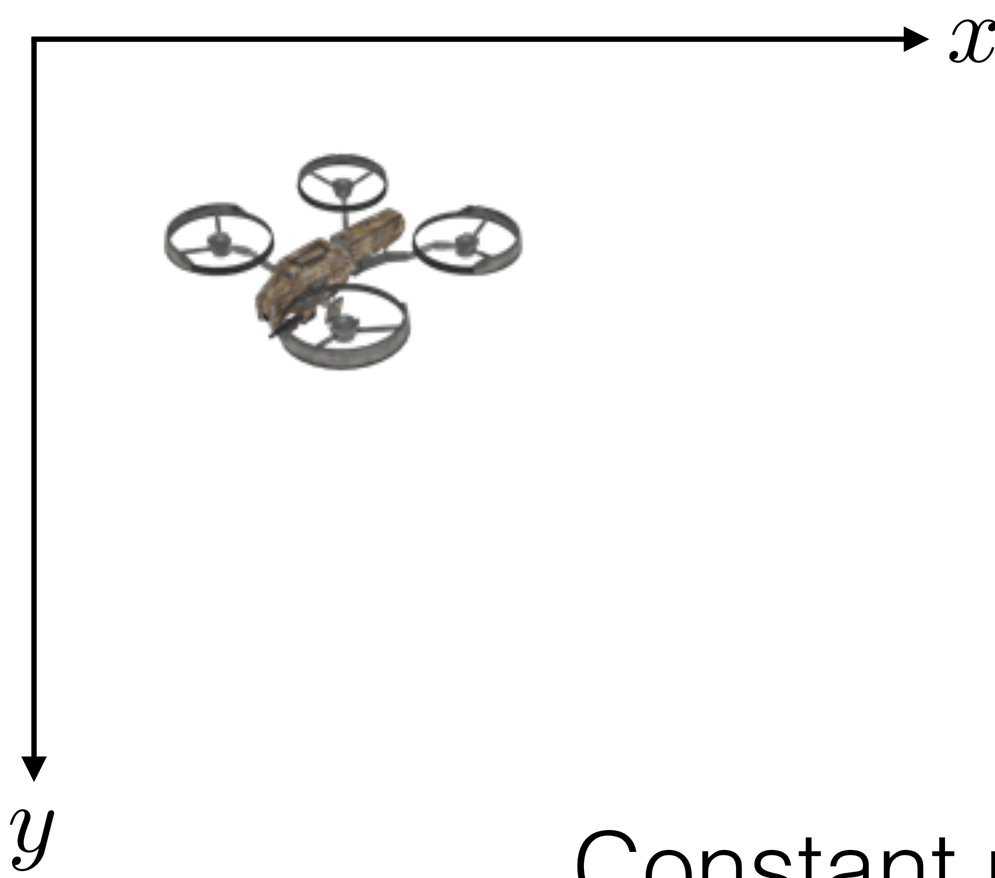
$$\boldsymbol{x} = \left[\begin{array}{c} x \\ y \end{array} \right]$$

measurement

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

Constant position Motion Model

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



state

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

measurement

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

Constant position Motion Model

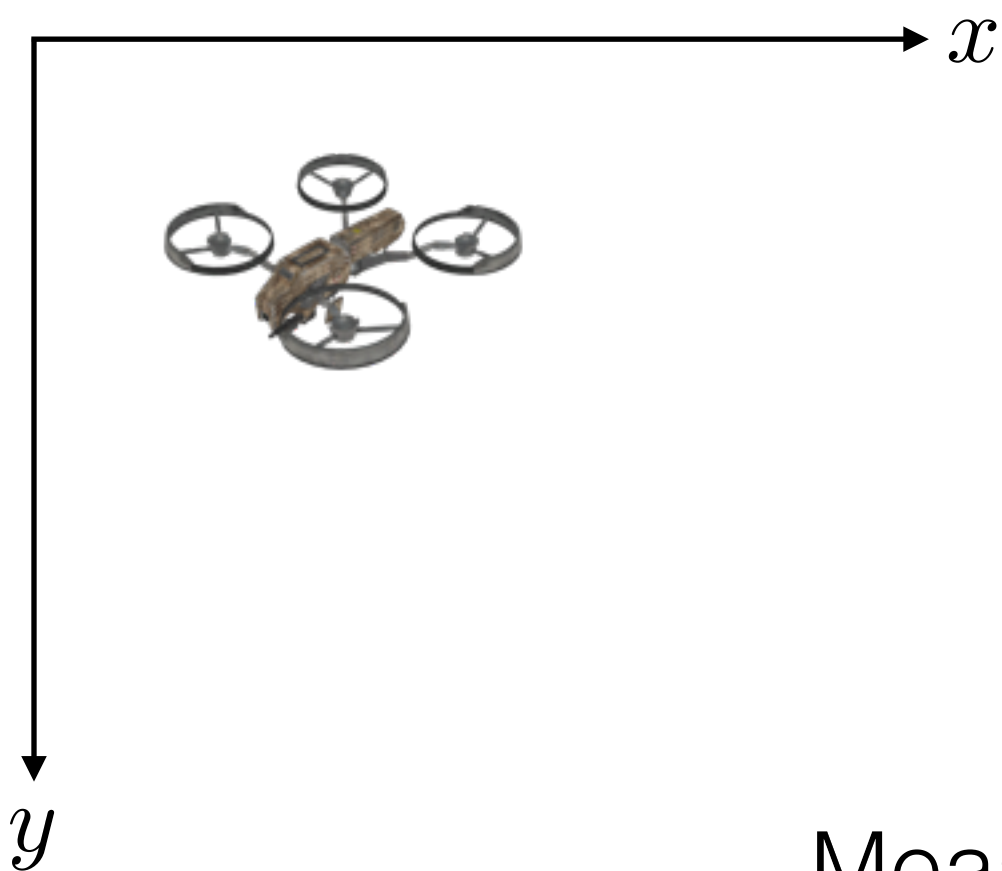
$$\boldsymbol{x}_t = A\boldsymbol{x}_{t-1} + B\boldsymbol{u}_t + \epsilon_t$$

system noise

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

Constant position

$$A = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \qquad B\boldsymbol{u} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \qquad R = \begin{bmatrix} \sigma_r^2 & 0 \\ 0 & \sigma_r^2 \end{bmatrix}$$



state

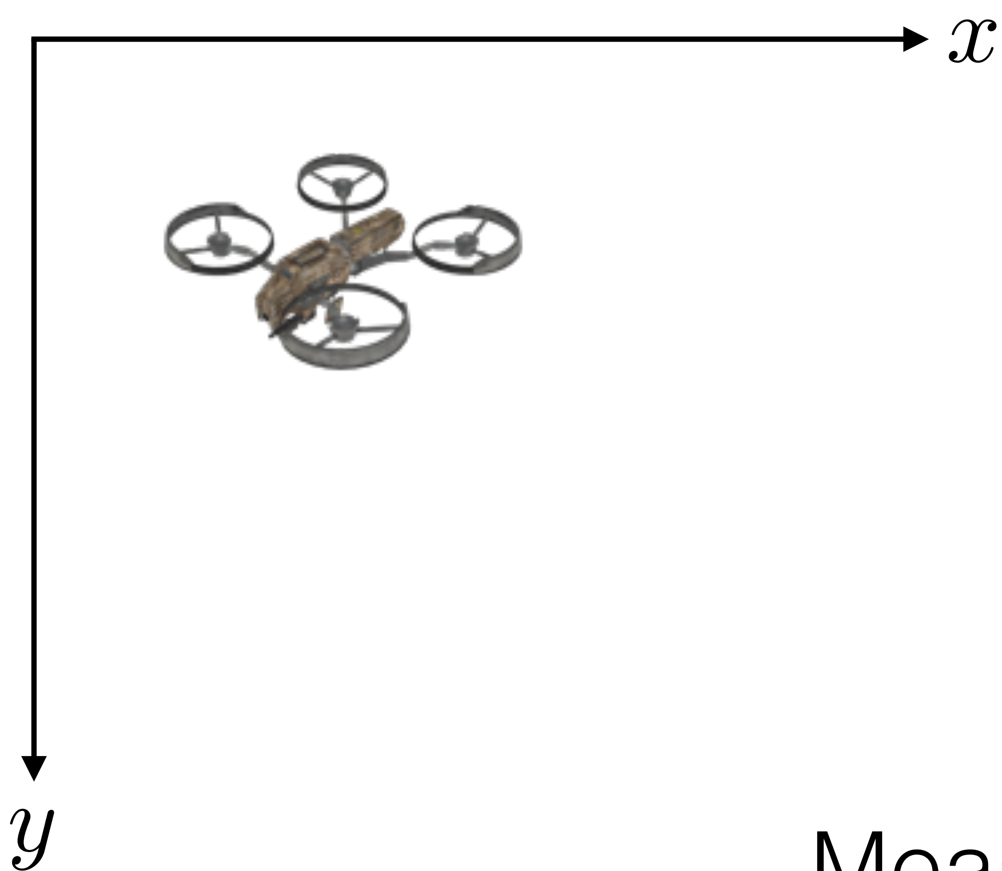
$$\boldsymbol{x} = \left[\begin{array}{c} x \\ y \end{array} \right]$$

measurement

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

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

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

motion model

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

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 + R;

  K = P / (P + Q);

  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

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