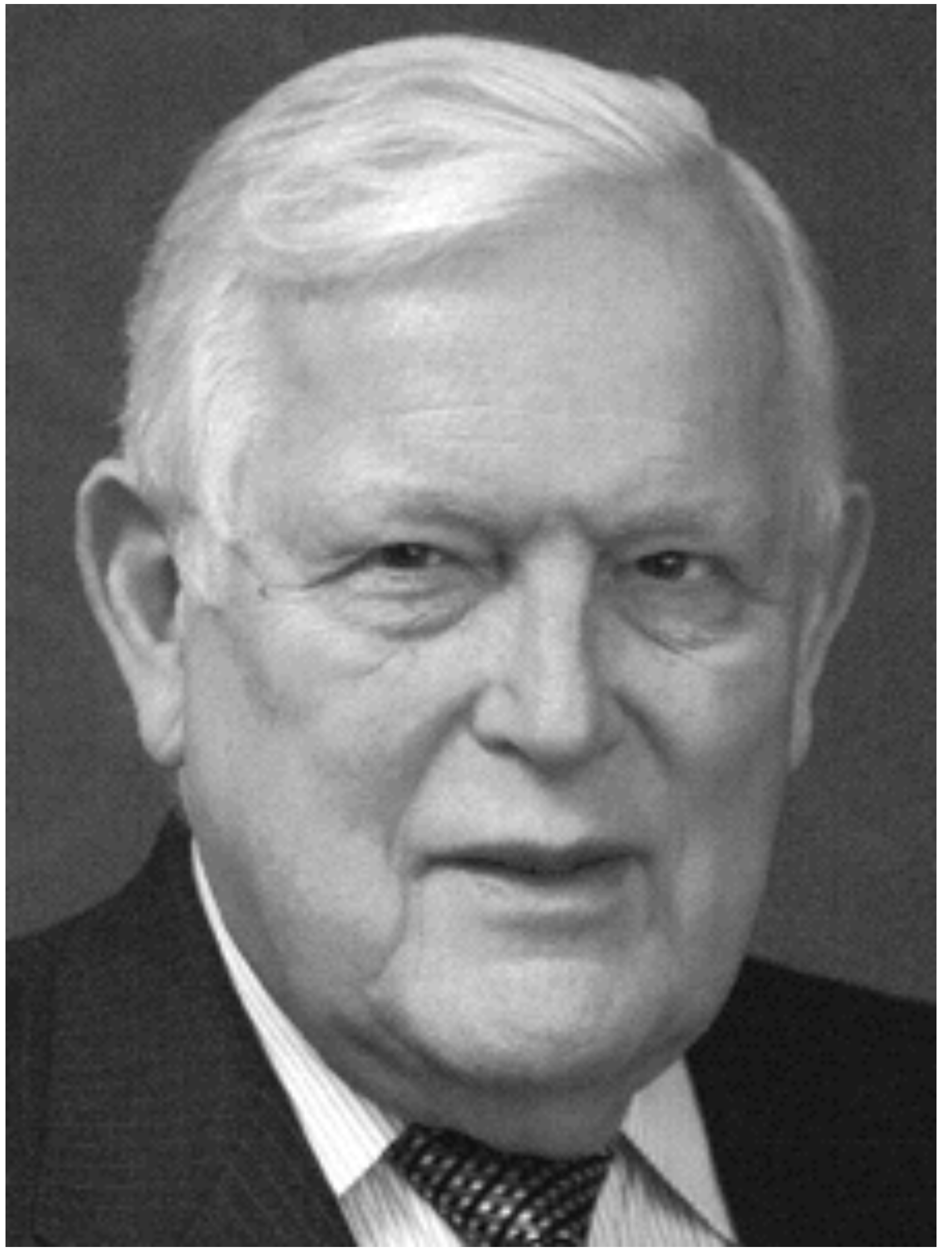
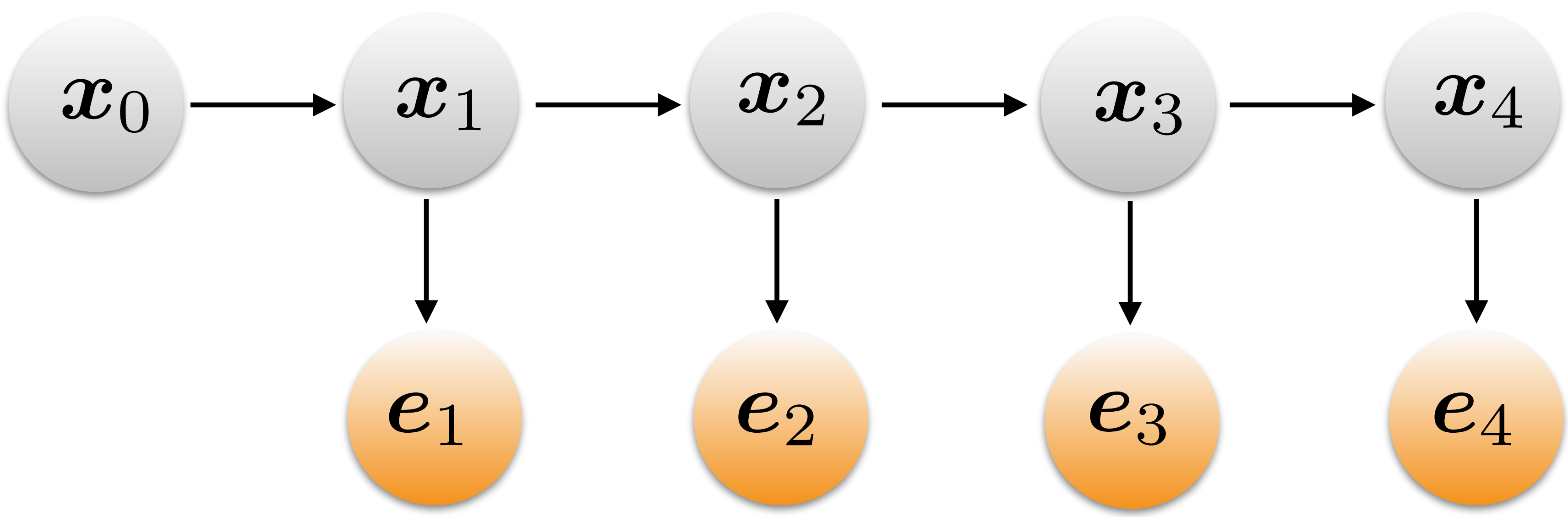


Kalman Filter

16-385 Computer Vision (Kris Kitani)
Carnegie Mellon University

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…

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

probability distributions are no longer tables but functions

Making the connection to the 'filtering' equations

(Discrete) Filtering

Tables | Tables | Tables

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

Kalman Filtering

Gaussian | Gaussian | Gaussian

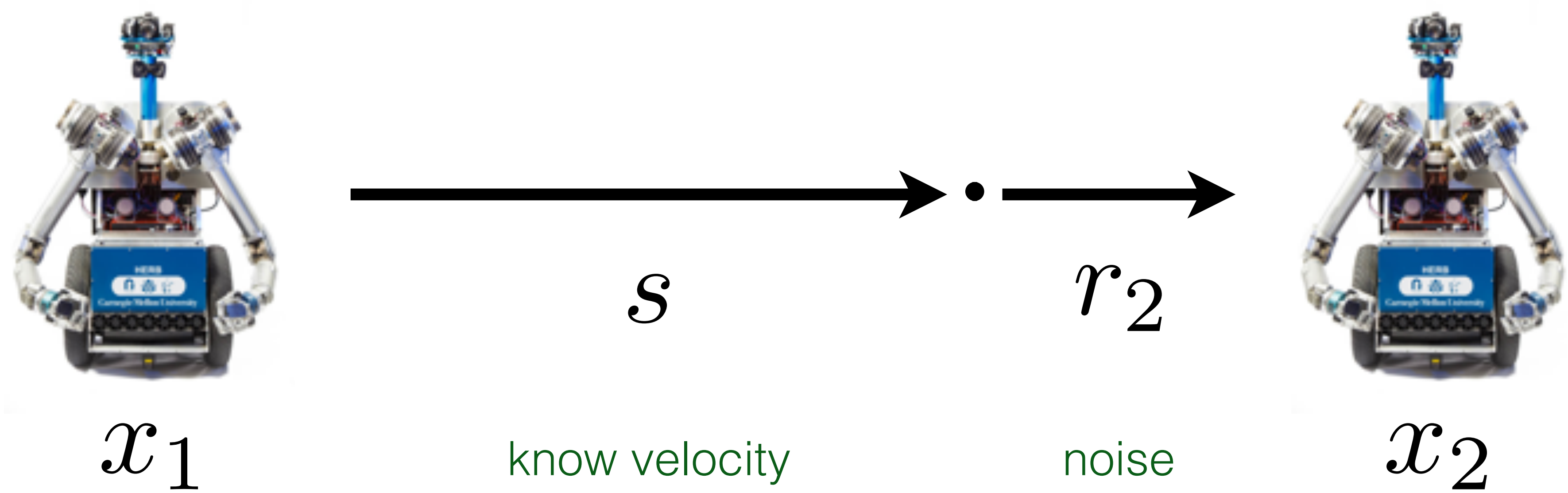
$$P(\boldsymbol{X}_{t+1}|\boldsymbol{e}_{1:t+1}) \propto P(\boldsymbol{e}_{t+1}|\boldsymbol{X}_{t+1}) \int_{\boldsymbol{x}_t} P(\boldsymbol{X}_{t+1}|\boldsymbol{x}_t) P(\boldsymbol{x}_t|\boldsymbol{e}_{1:t}) d\boldsymbol{x}_t$$

observation model | motion model | belief

integral because
continuous PDFs

Simple, 1D example…

x



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

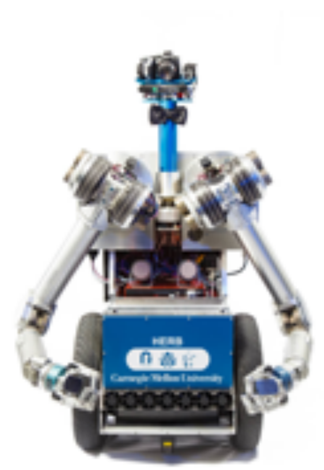
$$r_t \sim \mathcal{N}(0, \sigma_R)$$

'sampled from'

System (motion) model

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

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



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

x_1 | know velocity | noise | x_2

How do you represent the motion model?

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

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

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

s r_2

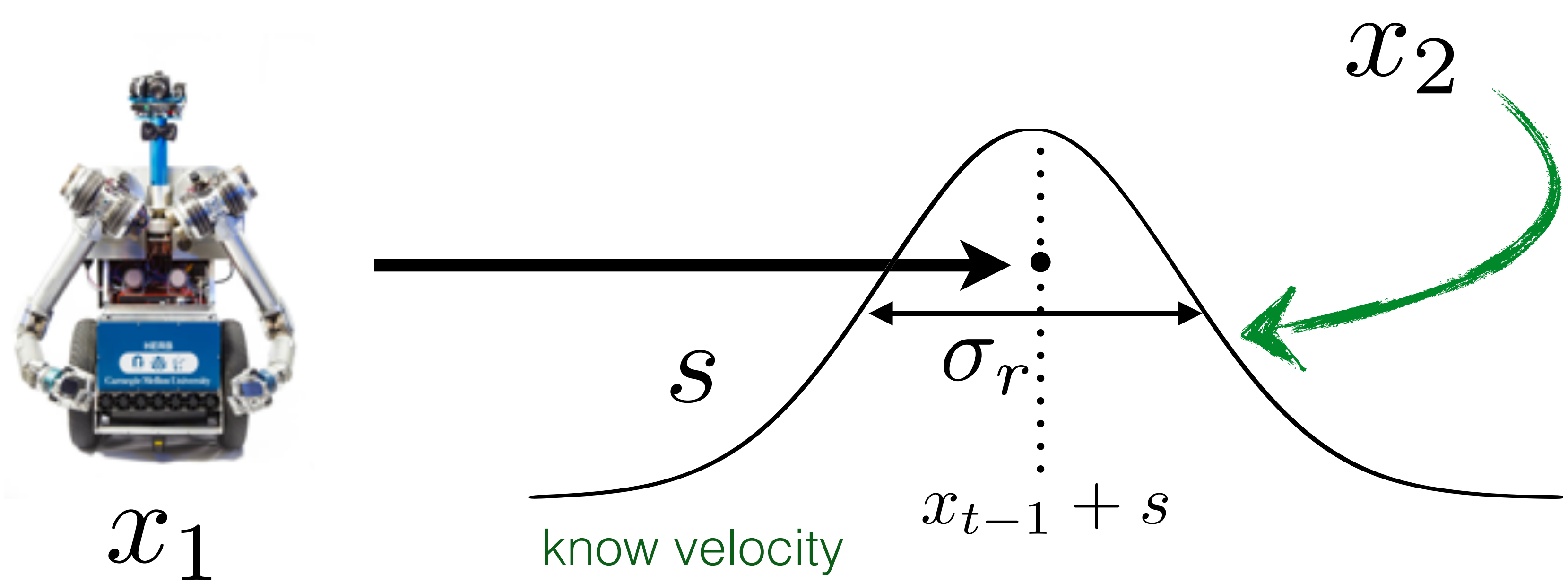
x_1 know velocity noise x_2

How do you represent the motion model?

A linear Gaussian (continuous) transition model

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

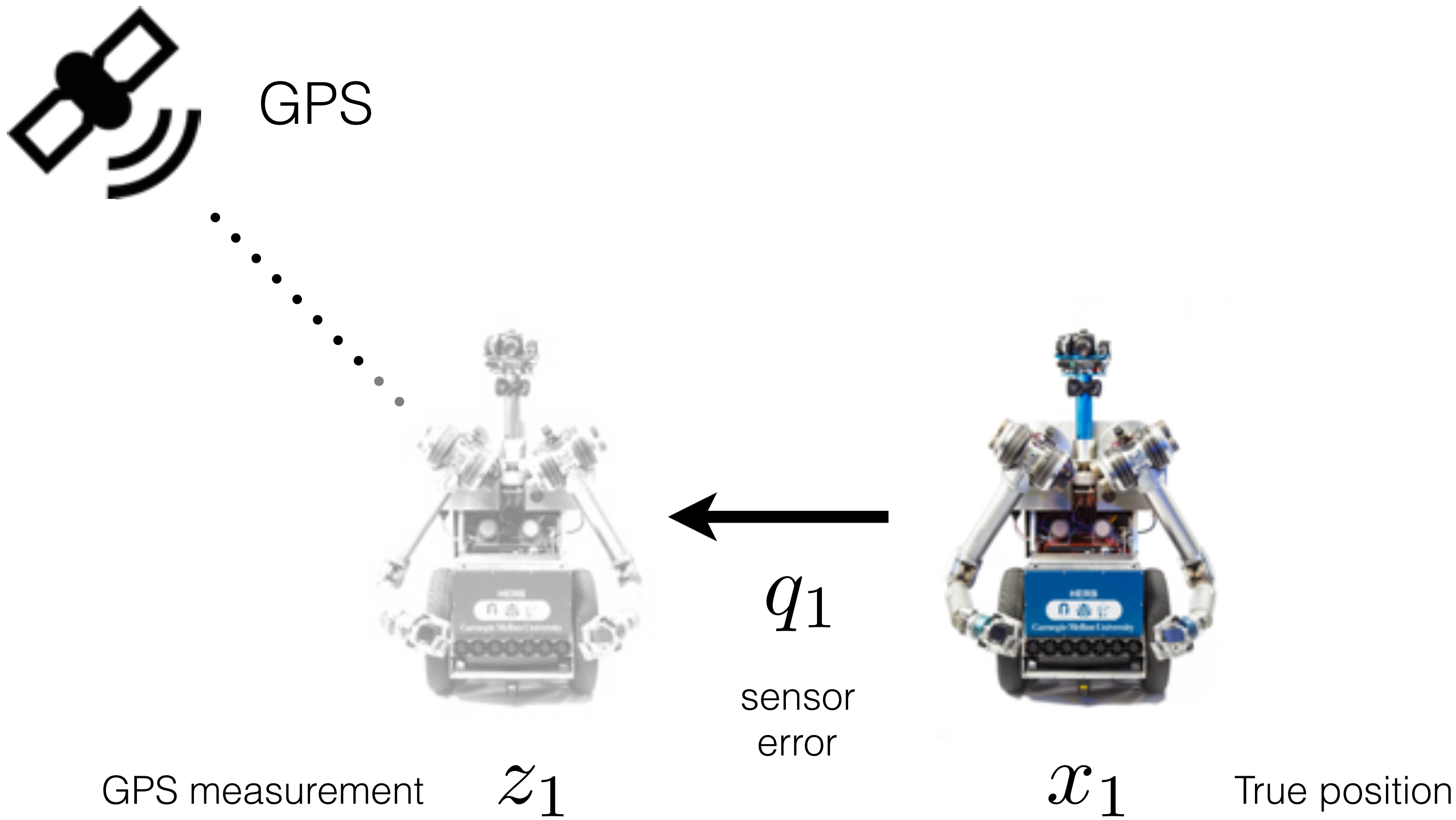
How can you visualize this distribution?



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?

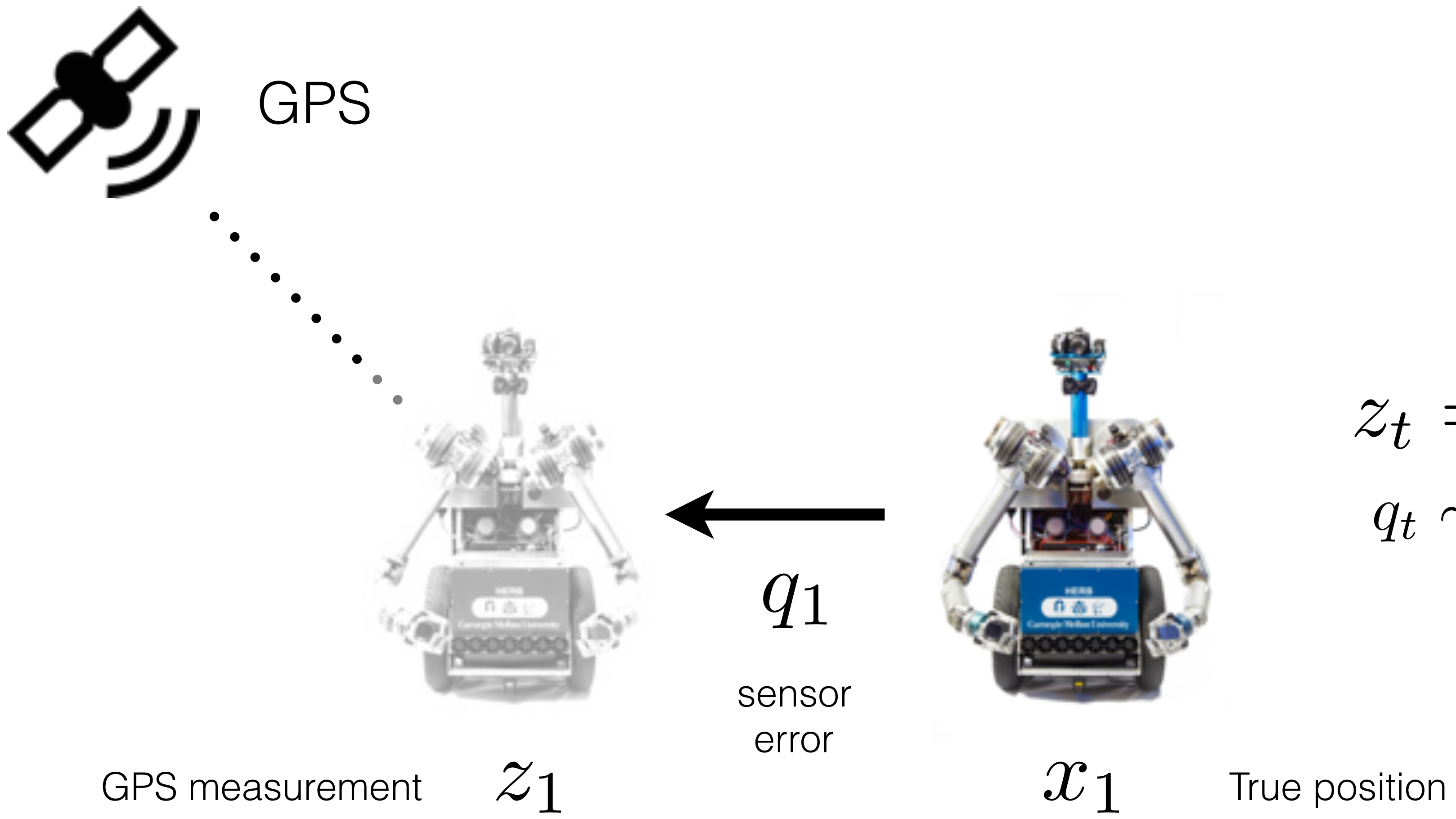


$$z_t = x_t + q_t$$

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

sampled from a Gaussian

Observation (measurement) model

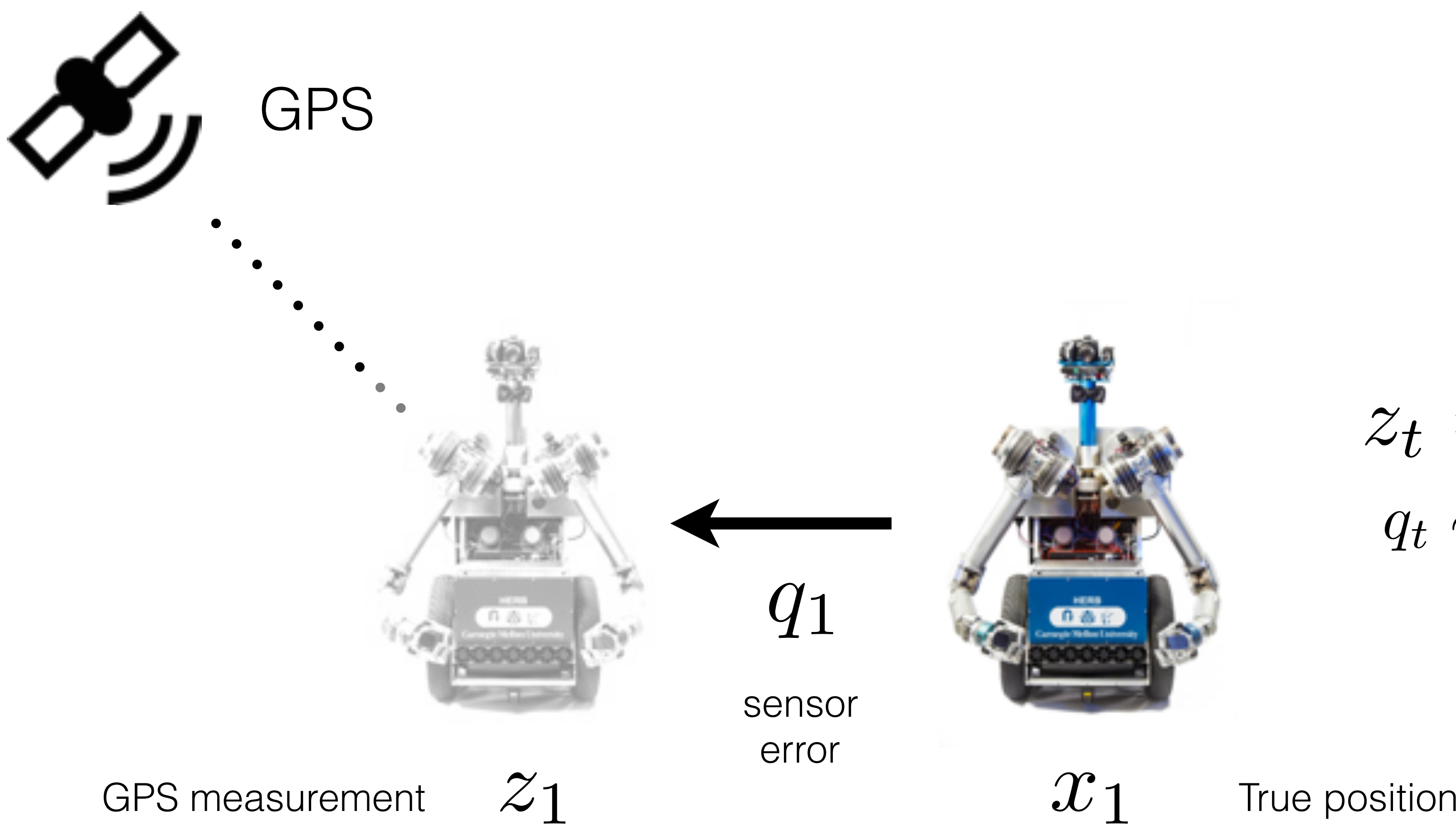


$$z_t = x_t + q_t$$
$$q_t \sim \mathcal{N}(0, \sigma_Q)$$

How do you represent the observation (measurement) model?

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

e represents z



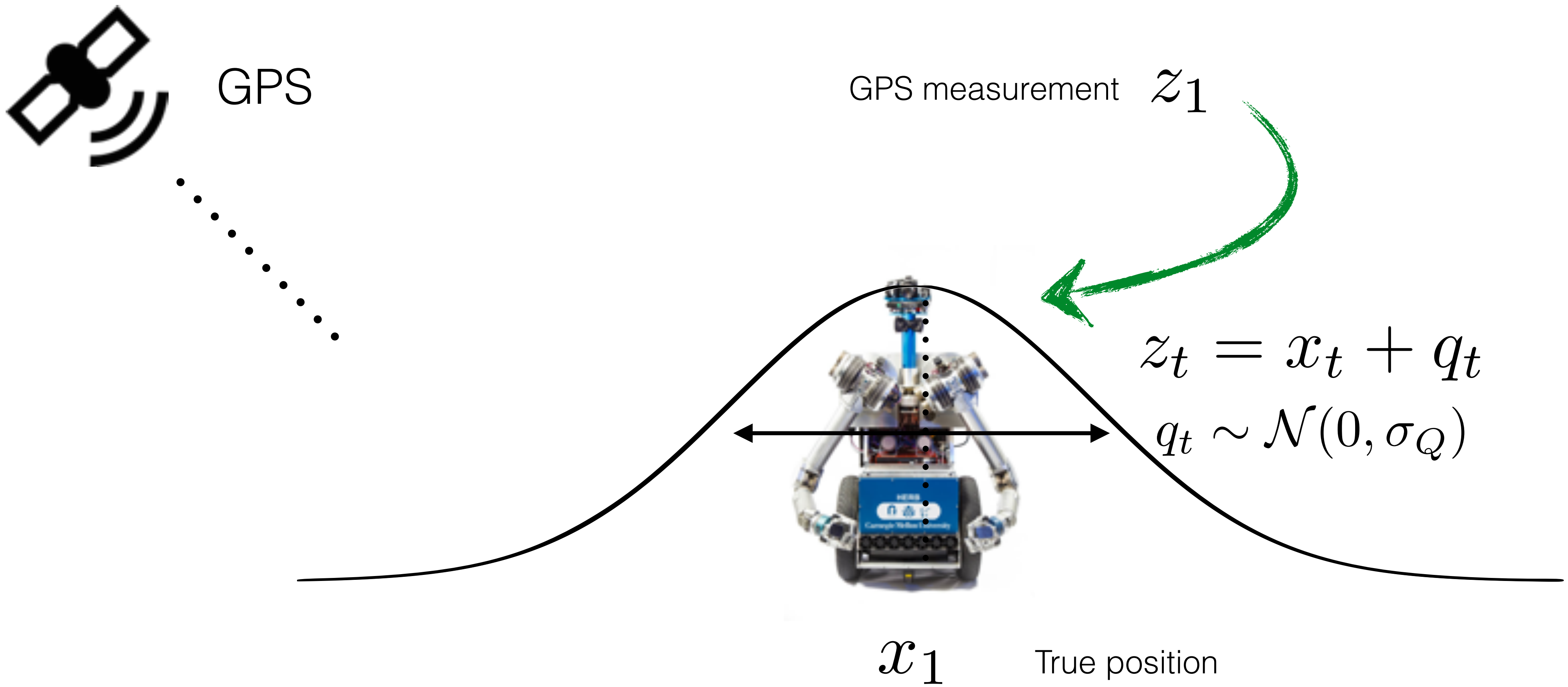
$$z_t = x_t + q_t$$

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

How do you represent the observation (measurement) model?

Also a linear Gaussian model

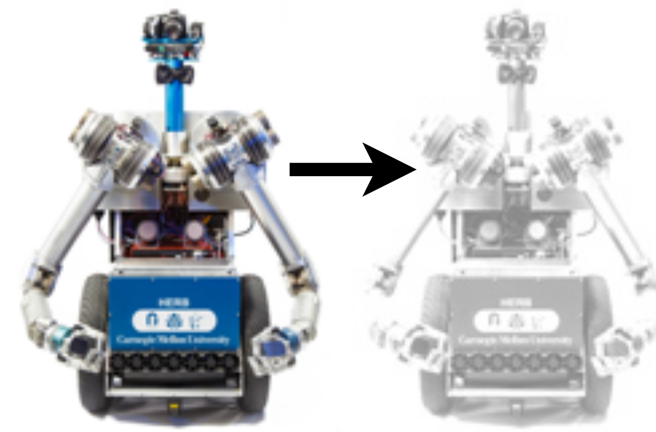
$$P(z_t | x_t) = \mathcal{N}(z_t; x_t, \sigma_Q)$$



How do you represent the observation (measurement) model?

Also a linear Gaussian model

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

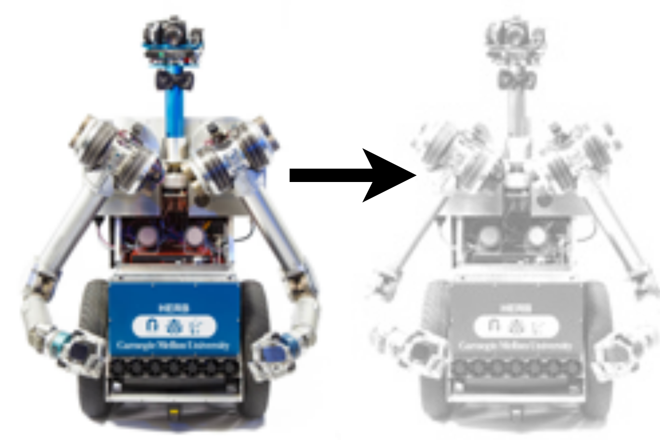


x_0 $\hat{x}_0$

true position initial estimate

initial estimate uncertainty σ_0

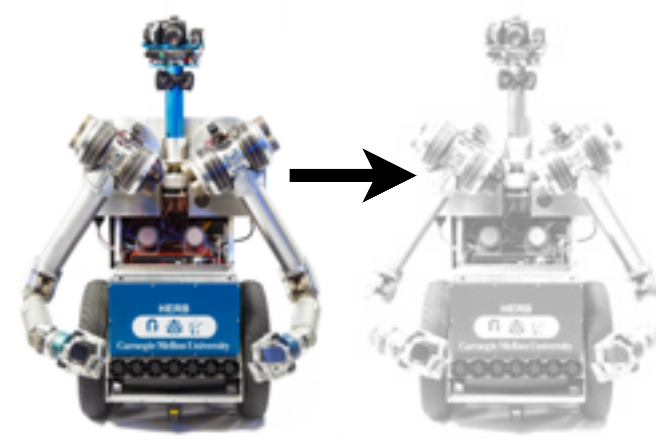
Prior (initial) State



x_0 $\hat{x}_0$

true position initial estimate

How do you represent the prior state probability?



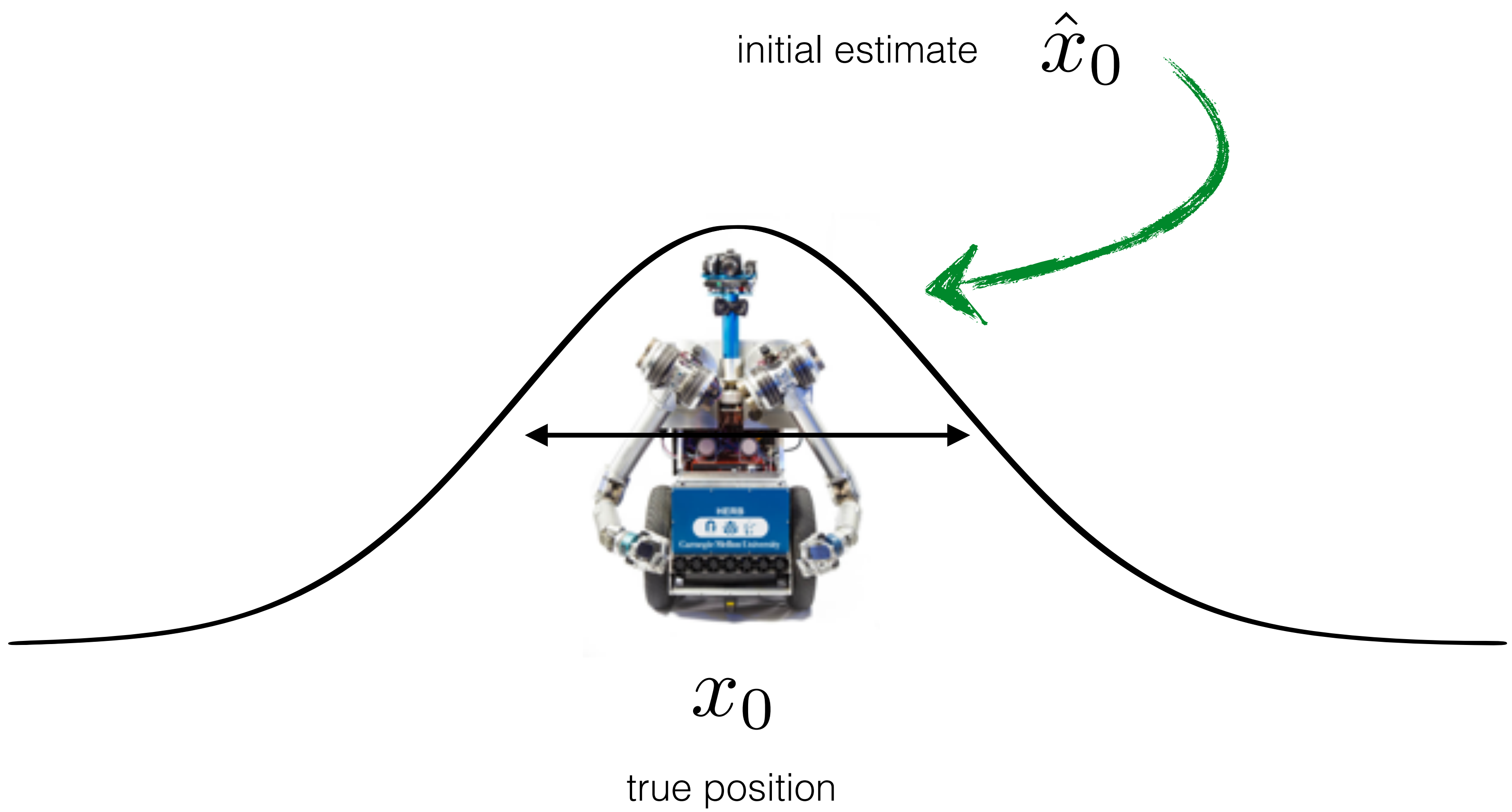
x_0 $\hat{x}_0$

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'

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

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)

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

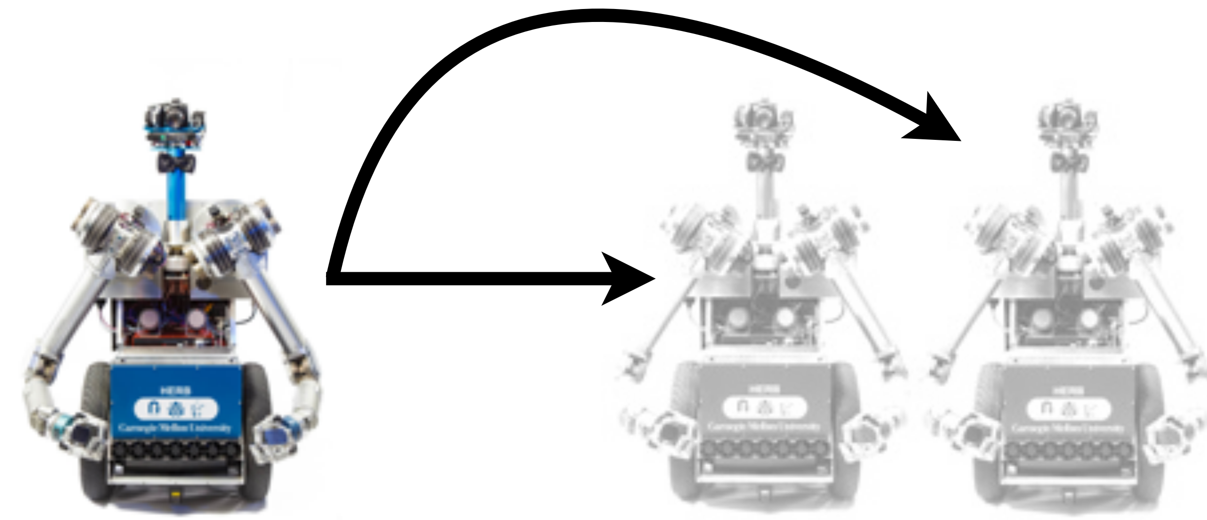
the second step after prediction is …

... 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:

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

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

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

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…

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

'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

'old' covariance

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

Prediction

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

Gain

observation model

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

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

Update

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

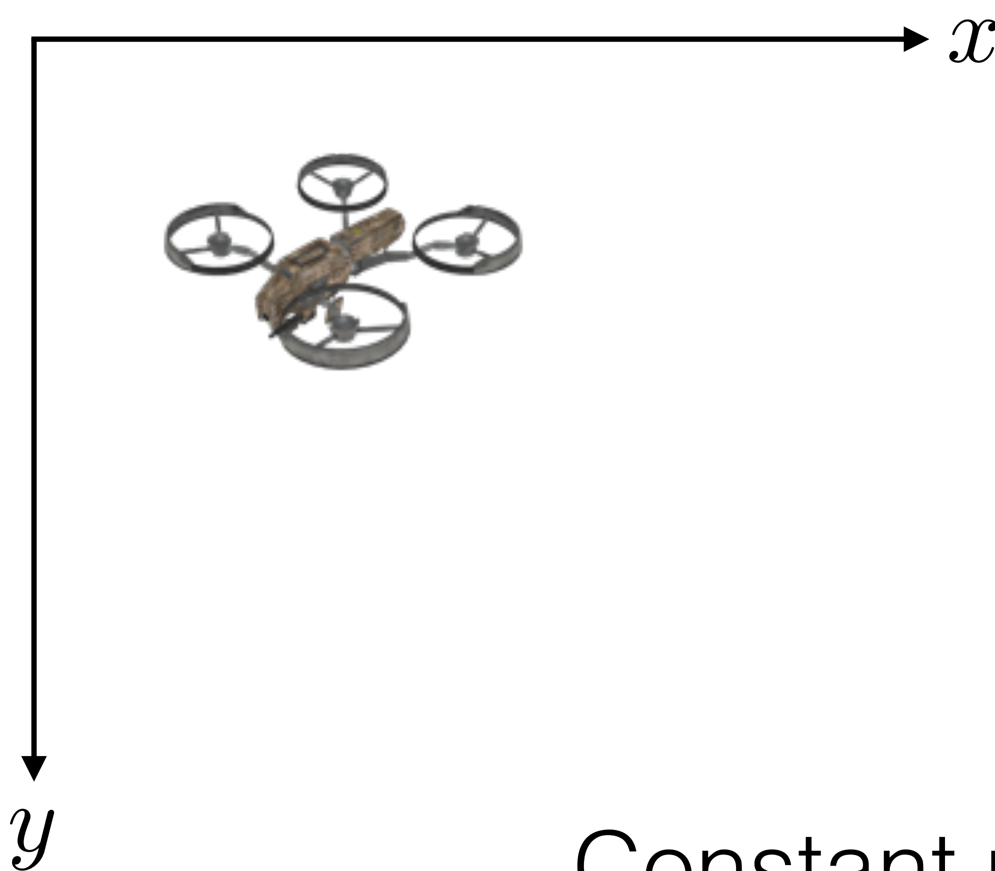
```
[x P] = KF(x,P,z)

  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

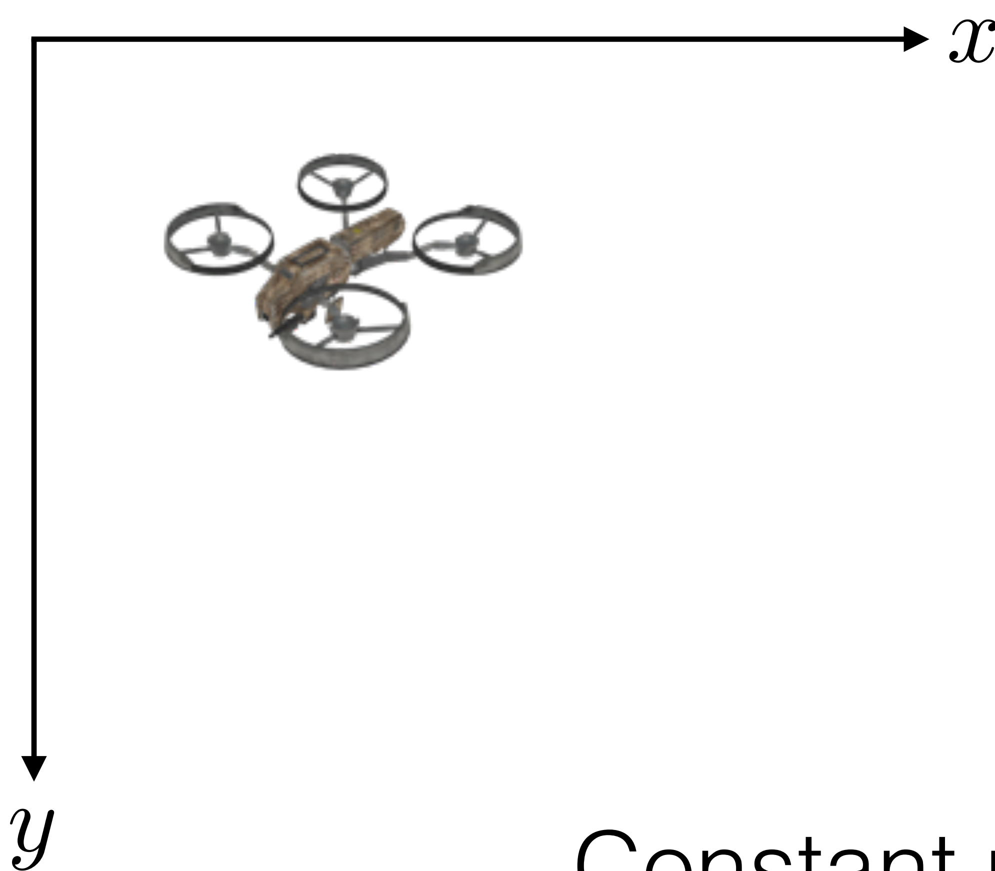
$$\boldsymbol{x} = \begin{bmatrix} x \\ y \end{bmatrix}$$

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

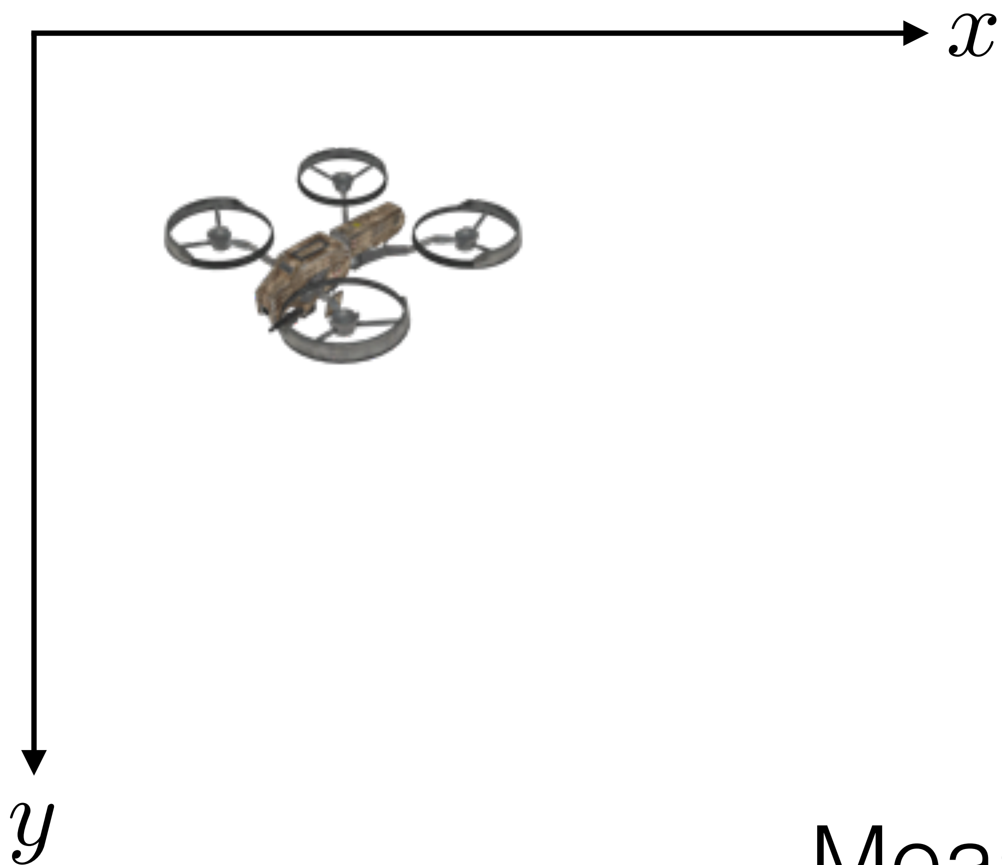
$$\boldsymbol{x}_t = A\boldsymbol{x}_{t-1} + B\boldsymbol{u}_t + \epsilon_t$$

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

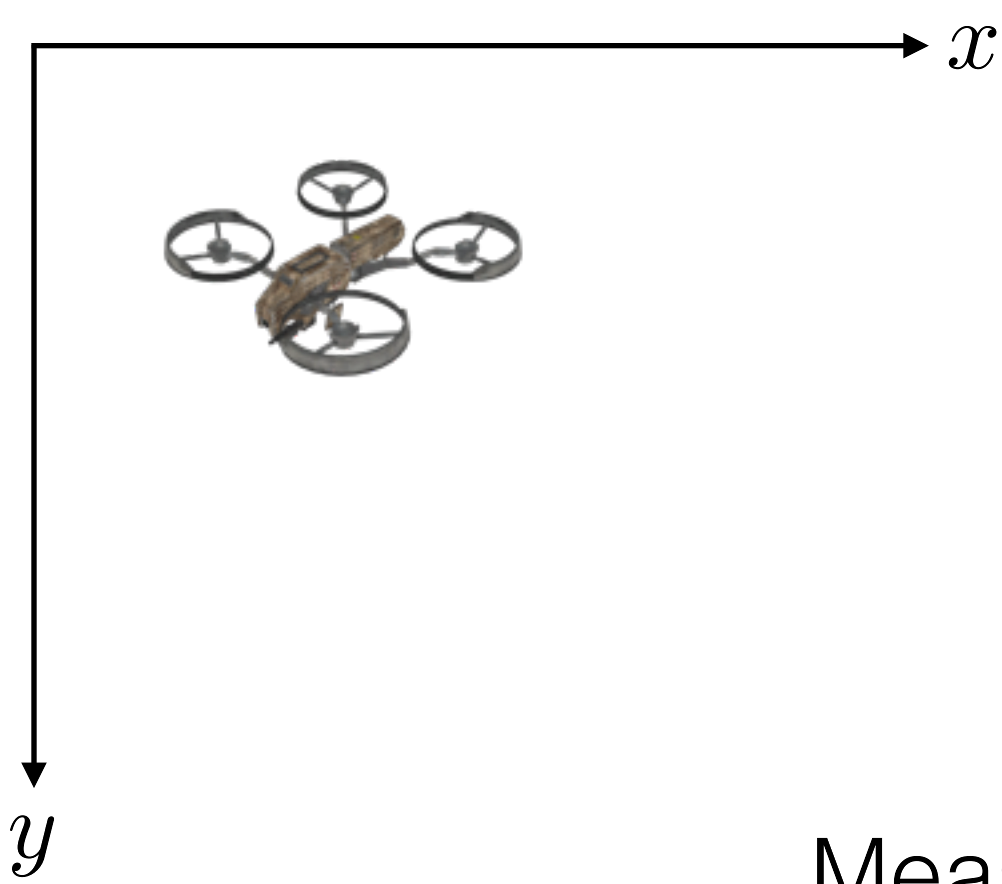
$$\boldsymbol{x} = \begin{bmatrix} x \\ y \end{bmatrix}$$

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

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

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 + 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

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