Localization, Mapping, SLAM and The Kalman Filter according to George

Robotics Institute 16-735

http://voronoi.sbp.ri.cmu.edu/~motion

Howie Choset

http://voronoi.sbp.ri.cmu.edu/~choset

The Problem

- What is the world around me (mapping)
  - sense from various positions
  - integrate measurements to produce map
  - assumes perfect knowledge of position
- Where am I in the world (localization)
  - sense
  - relate sensor readings to a world model
  - compute location relative to model
  - assumes a perfect world model
- Together, these are SLAM (Simultaneous Localization and Mapping)

Localization

Tracking: Known initial position

Global Localization: Unknown initial position

Re-Localization: Incorrect known position

(kidnapped robot problem)

SLAM

Mapping while tracking locally and globally

Challenges

- Sensor processing
- Position estimation
- Control Scheme
- Exploration Scheme
- Cycle Closure
- Autonomy
- Tractability
- Scalability

Representations for Robot Localization

Discrete approaches ('95)
- Topological representation ('95)
  - uncertainty handling (POMDPs)
  - occas. global localization, recovery
- Grid-based, metric representation ('96)
  - global localization, recovery

Kalman filters (late-80s?)
- Gaussians
- approximately linear models
- position tracking

Robotics

AI

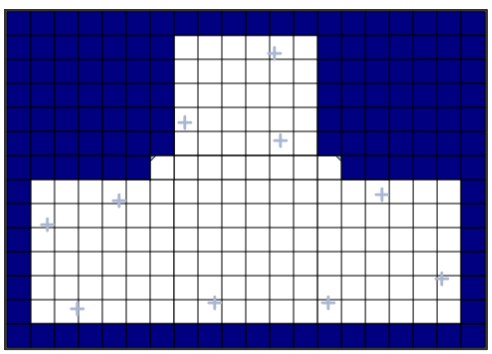
Particle filters ('99)
- sample-based representation
- global localization, recovery

Multi-hypothesis ('00)
- multiple Kalman filters
- global localization, recovery

Three Major Map Models

Grid-Based:

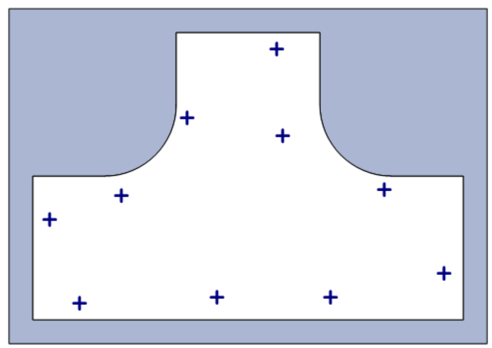
Collection of discretized obstacle/free-space pixels



Elfes, Moravec, Thrun, Burgard, Fox, Simmons, Koenig, Konolige, etc.

Feature-Based:

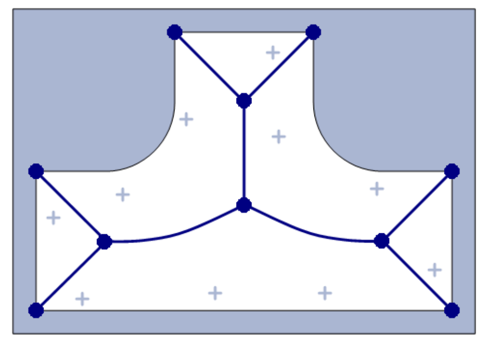
Collection of landmark locations and correlated uncertainty



Smith/Self/Cheeseman, Durrant–Whyte, Leonard, Nebot, Christensen, etc.

Topological:

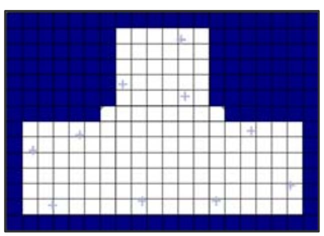
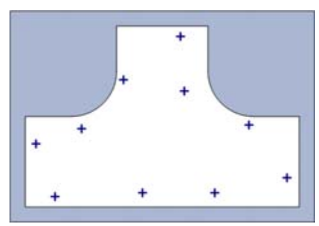
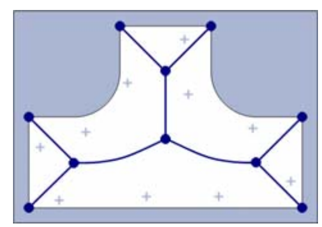
Collection of nodes and their interconnections



Kuipers/Byun, Chong/Kleeman, Dudek, Choset, Howard, Mataric, etc.

Three Major Map Models

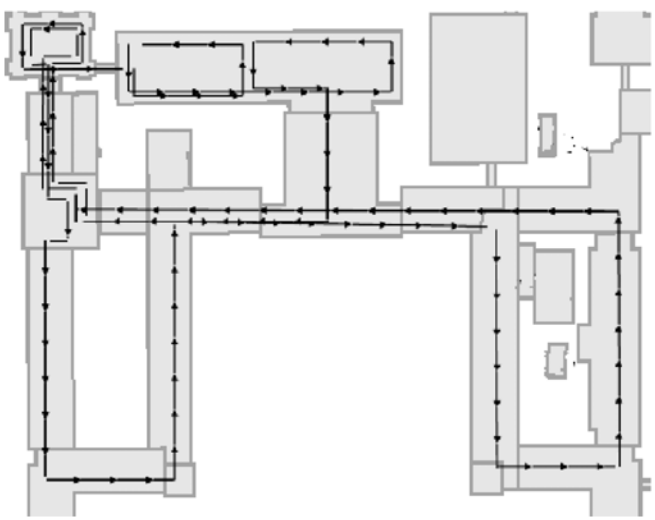
	Grid-Based	Feature-Based	Topological
Resolution vs. Scale	Discrete localization	Arbitrary localization	Localize to nodes
Computational Complexity	Grid size and resolution	Landmark covariance (N^2)	Minimal complexity
Exploration Strategies	Frontier-based exploration	No inherent exploration	Graph exploration

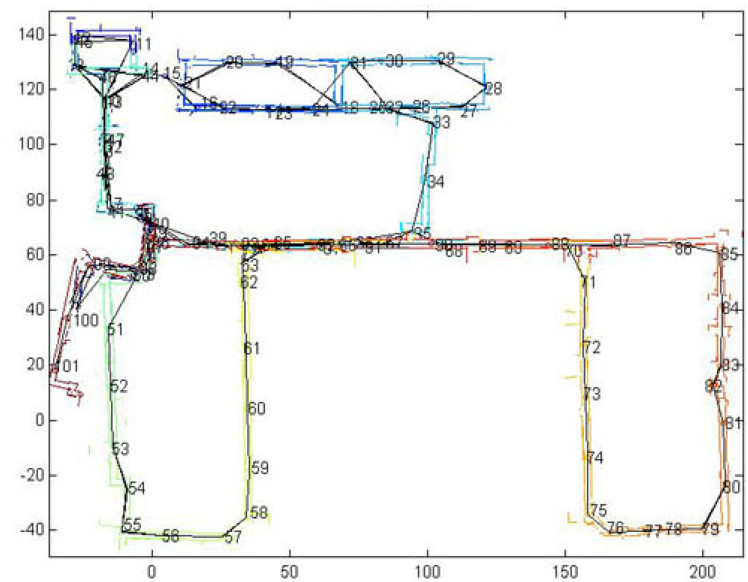




Atlas Framework

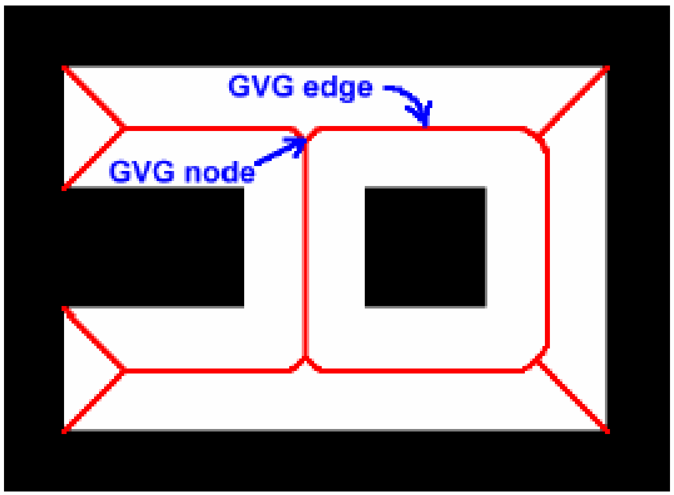
- **Hybrid Solution:**
  - Local features extracted from local grid map.
  - Local map frames created at complexity limit.
  - Topology consists of connected local map frames.

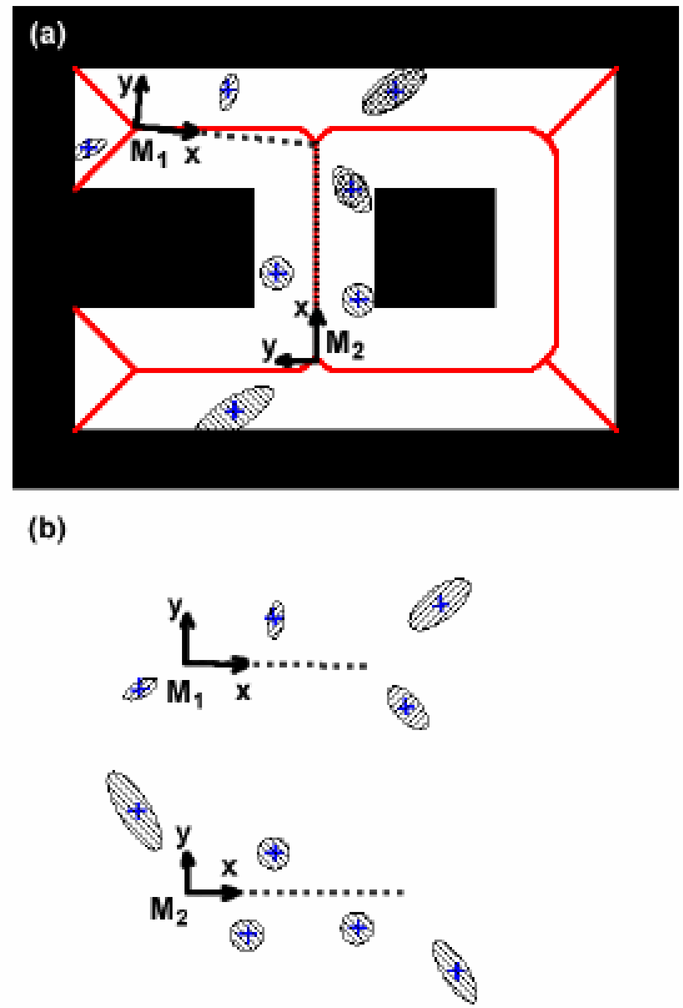
Authors: Chong, Kleeman; Bosse, Newman, Leonard, Soika, Feiten, Teller





H-SLAM





What does a Kalman Filter do, anyway?

Given the linear dynamical system:

$$x(k+1) = F(k)x(k) + G(k)u(k) + v(k)$$

$$y(k) = H(k)x(k) + w(k)$$

$x(k)$ is the n-dimensional state vector (unknown)

$u(k)$ is the m-dimensional input vector (known)

$y(k)$ is the p-dimensional output vector (known, measured)

$F(k), G(k), H(k)$ are appropriately dimensioned system matrices (known)

$v(k), w(k)$ are zero-mean, white Gaussian noise with (known)
covariance matrices $Q(k), R(k)$

the Kalman Filter is a recursion that provides the "best" estimate of the state vector *x*.

What's so great about that?

$$x(k+1) = F(k)x(k) + G(k)u(k) + v(k)$$
$$y(k) = H(k)x(k) + w(k)$$

- noise smoothing (improve noisy measurements)
- state estimation (for state feedback)
- recursive (computes next estimate using only most recent measurement)

How does it work?

$$x(k+1) = F(k)x(k) + G(k)u(k) + v(k)$$
$$y(k) = H(k)x(k) + w(k)$$

1. prediction based on last estimate:

$$\hat{x}(k+1\,|\,k) = F(k)\hat{x}(k\,|\,k) + G(k)u(k)$$
$$\hat{y}(k) = H(k)\hat{x}(k+1\,|\,k)$$

2. calculate correction based on prediction and current measurement:

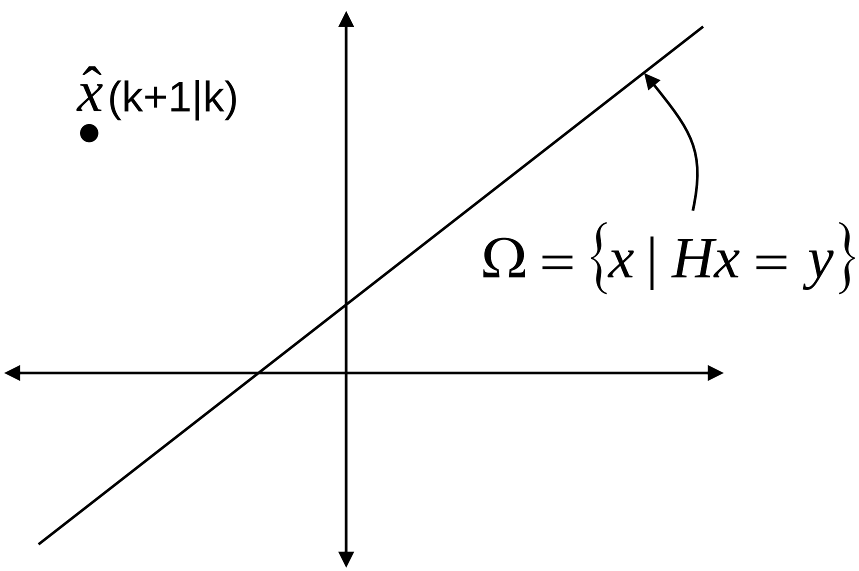
$$\Delta x = f\left(y(k+1), \hat{x}(k+1\,|\,k)\right)$$

3. update prediction: $\hat{x}(k+1\,|\,k+1) = \hat{x}(k+1\,|\,k) + \Delta x$

Finding the correction (no noise!)

$$y = Hx$$

Given prediction $\hat{x}(k+1|k)$ and output y, find Δx so that $\hat{x} = \hat{x}(k+1|k) + \Delta x$ is the "best" estimate of x.



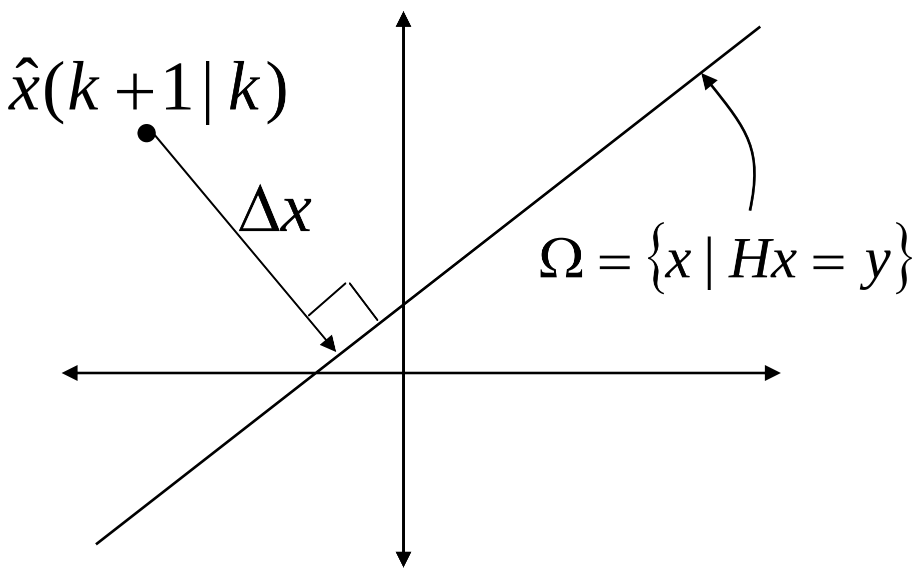
Want the best estimate to be consistent with sensor readings

"best" estimate comes from shortest Δx

Finding the correction (no noise!)

$$y = Hx$$

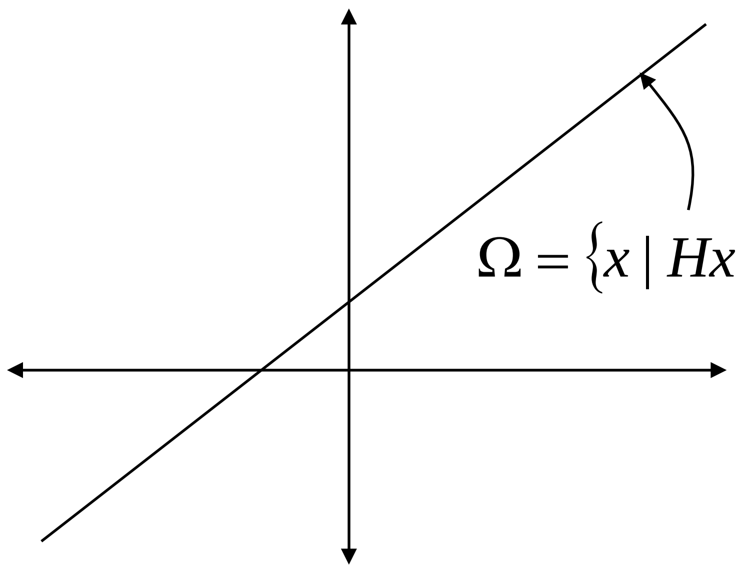
Given prediction $\hat{x}(k+1|1)$ and output y, find Δx so that $\hat{x} = \hat{x}(k+1|1) + \Delta x$ is the "best" estimate of x.



"best" estimate comes from shortest Δx

shortest Δx is perpendicular to Ω

Some linear algebra



$$\Omega = \{x \mid Hx = y\}$$

a is parallel to Ω if Ha = 0

$$Null(H) = \{a \neq 0 \mid Ha = 0\}$$

a is parallel to Ω if it lies in the null space of H

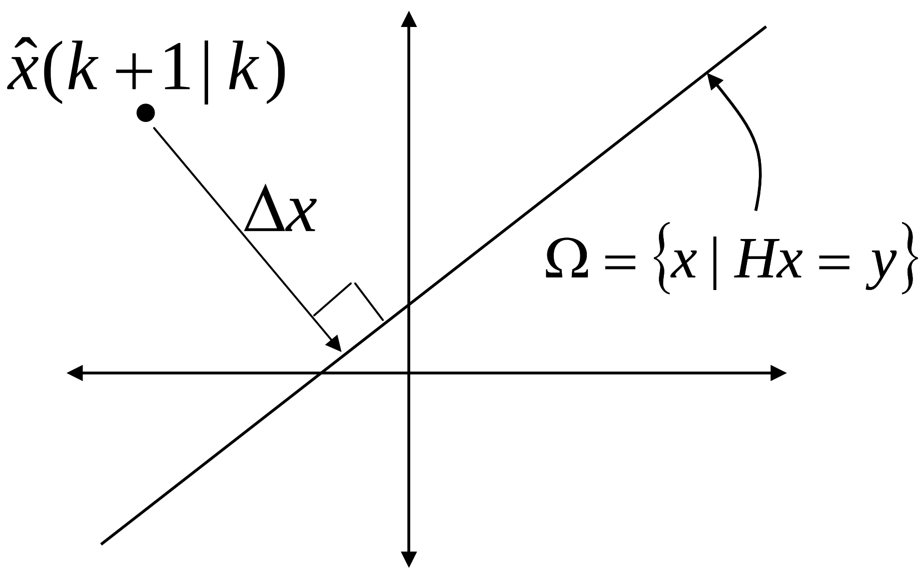
$$\text{for all } v \in Null(H), v \perp b \text{ if } b \in column(H^T)$$

Weighted sum of columns means $b = H\gamma$, the weighted sum of columns

Finding the correction (no noise!)

$$y = Hx$$

Given prediction $\hat{x}(k+1|k)$ and output y, find Δx so that $\hat{x} = \hat{x}(k+1|k) + \Delta x$ is the "best" estimate of x.



"best" estimate comes from shortest Δx

shortest Δx is perpendicular to Ω

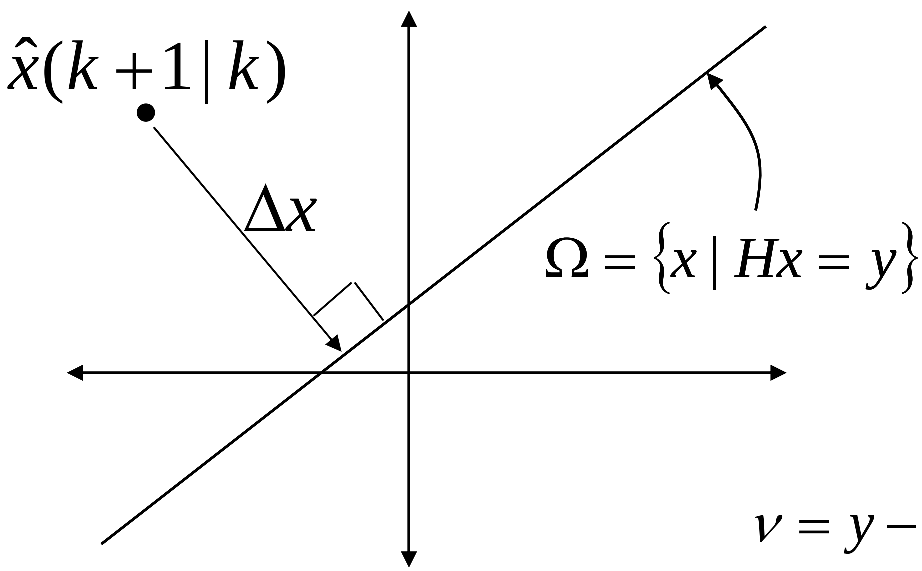
$$\Rightarrow \Delta x \in \text{null}(H)^{\perp} \Rightarrow \Delta x \in \text{column}(H^T)$$

$$\Rightarrow \Delta x = H^T \gamma$$

Finding the correction (no noise!)

$$y = Hx$$

Given prediction $\hat{x}(k+1|k)$ and output y, find Δx so that $\hat{x} = \hat{x}(k+1|k) + \Delta x$ is the "best" estimate of x.



"best" estimate comes from shortest Δx

shortest Δx is perpendicular to Ω

$$\Rightarrow \Delta x \in \text{null}(H)^{\perp} \Rightarrow \Delta x \in \text{column}(H^T)$$

$$\Rightarrow \Delta x = H^T \gamma$$

Real output – estimated output

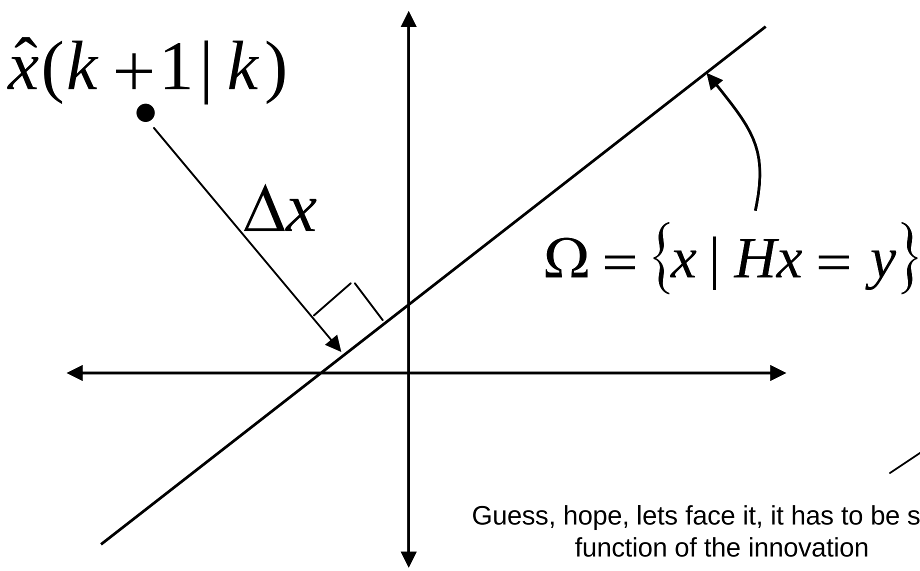
$$\nu = y - H(\hat{x}(k+1|k)) = H(x - \hat{x}(k+1|k))$$

innovation

Finding the correction (no noise!)

$$y = Hx$$

Given prediction $\hat{x}(k+1|k)$ and output y, find Δx so that $\hat{x} = \hat{x}(k+1|k) + \Delta x$ is the "best" estimate of x.



"best" estimate comes from shortest Δx

shortest Δx is perpendicular to Ω

$$\Rightarrow \Delta x \in \text{null}(H)^{\perp} \Rightarrow \Delta x \in \text{column}(H^T)$$

$$\Rightarrow \Delta x = H^T \gamma$$

assume γ is a linear function of ν

$$\Rightarrow \Delta x = H^T K \nu$$

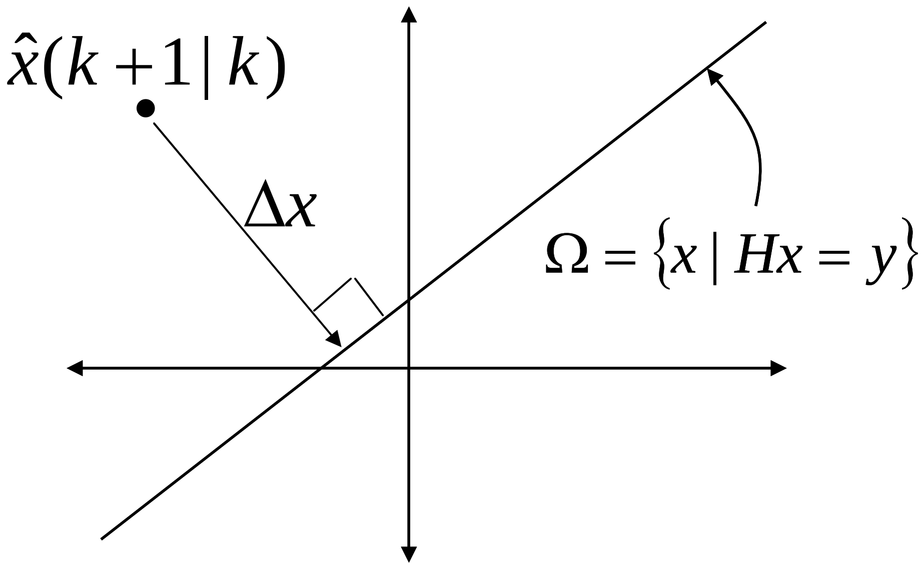
for some $m \times m$ matrix K

Guess, hope, lets face it, it has to be some function of the innovation

Finding the correction (no noise!)

$$y = Hx$$

Given prediction $\hat{x}_-$ and output y, find Δx so that $\hat{x} = \hat{x}(k+1|k) + \Delta x$ is the "best" estimate of x.



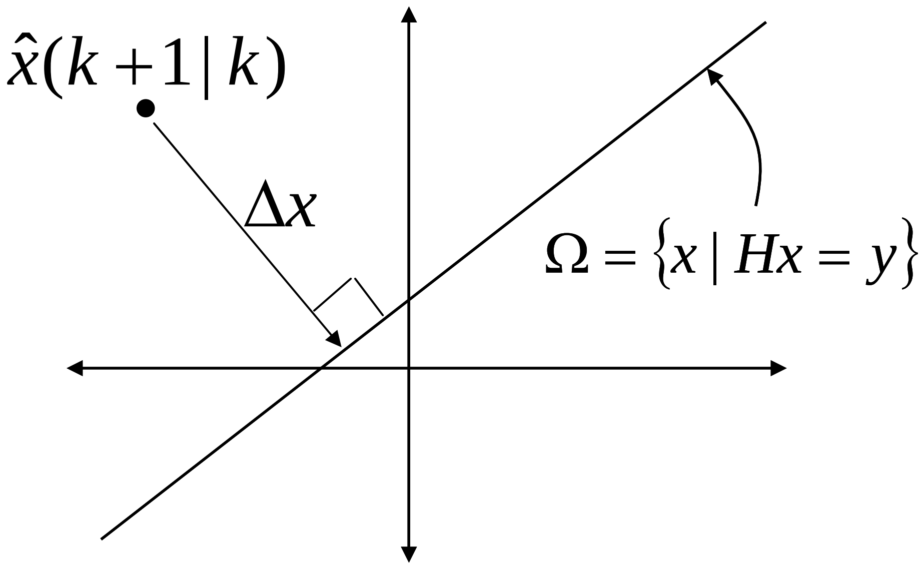
we require

$$H(\hat{x}(k+1|k) + \Delta x) = y$$

Finding the correction (no noise!)

$$y = Hx$$

Given prediction $\hat{x}_-$ and output y, find Δx so that $\hat{x} = \hat{x}(k+1|k) + \Delta x$ is the "best" estimate of x.



we require

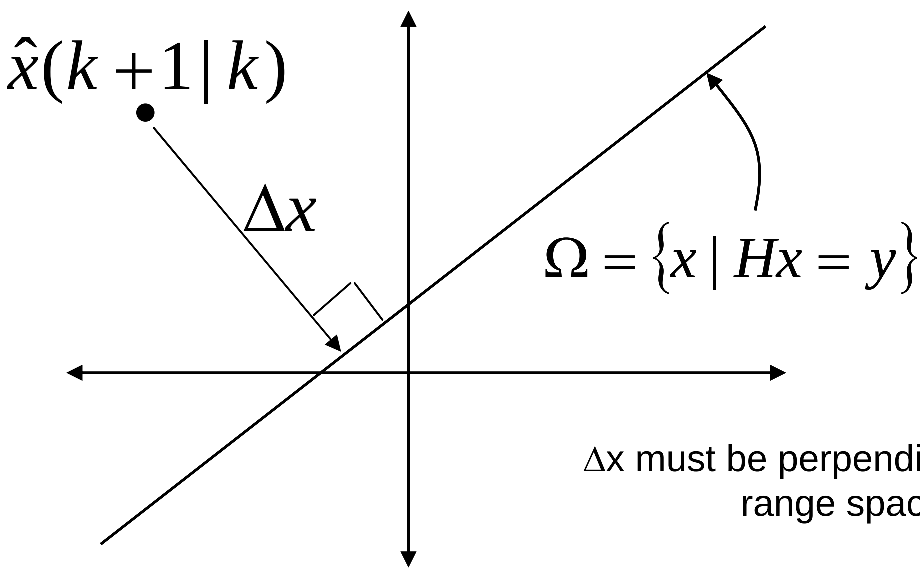
$$H(\hat{x}(k+1|k) + \Delta x) = y$$

$$\Rightarrow H\Delta x = y - H\hat{x}(k+1|k) = H(x - \hat{x}(k+1|k)) = \nu$$

Finding the correction (no noise!)

$$y = Hx$$

Given prediction $\hat{x}_-$ and output y, find Δx so that $\hat{x} = \hat{x}(k+1|k) + \Delta x$ is the "best" estimate of x.



we require

$$H(\hat{x}(k+1|k) + \Delta x) = y$$

$$\Rightarrow H\Delta x = y - H\hat{x}(k+1|k) = H(x - \hat{x}(k+1|k)) = \nu$$

substituting $\Delta x = H^T K \nu$ yields

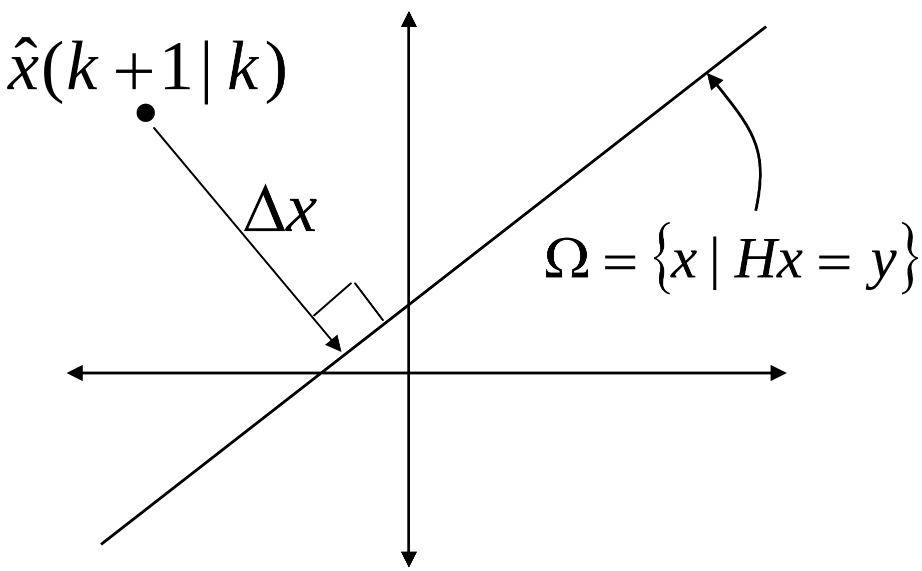
$$HH^T K \nu = \nu$$

Δx must be perpendicular to Ω b/c anything in the range space of H^T is perp to Ω

Finding the correction (no noise!)

$$y = Hx$$

Given prediction $\hat{x}_-$ and output y, find Δx so that $\hat{x} = \hat{x}(k+1|k) + \Delta x$ is the "best" estimate of x.



we require

$$H(\hat{x}(k+1|k) + \Delta x) = y$$

$$\Rightarrow H\Delta x = y - H\hat{x}(k+1|k) = H(x - \hat{x}(k+1|k)) = \nu$$

substituting $\Delta x = H^T K \nu$ yields

$$HH^T K \nu = \nu$$

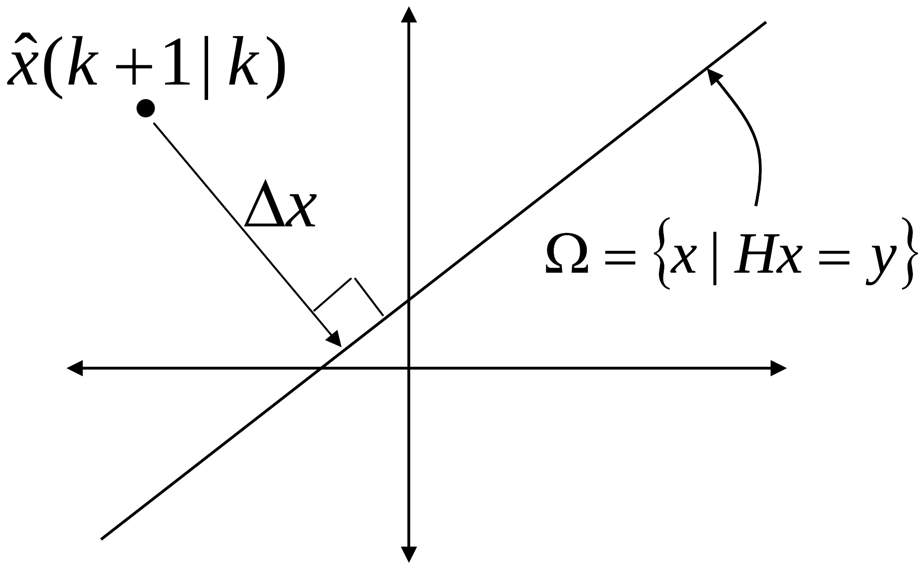
$$\Rightarrow K = \left(HH^T\right)^{-1}$$

The fact that the linear solution solves the equation makes assuming K is linear a kosher guess

Finding the correction (no noise!)

$$y = Hx$$

Given prediction $\hat{x}_{-}$ and output y, find Δx so that $\hat{x} = \hat{x}(k+1 \mid k) + \Delta x$ is the "best" estimate of x.



we require

$$H(\hat{x}(k+1 \mid k) + \Delta x) = y$$

$$\Rightarrow H\Delta x = y - H\hat{x}(k+1 \mid k) = H(x - \hat{x}(k+1 \mid k)) = \nu$$

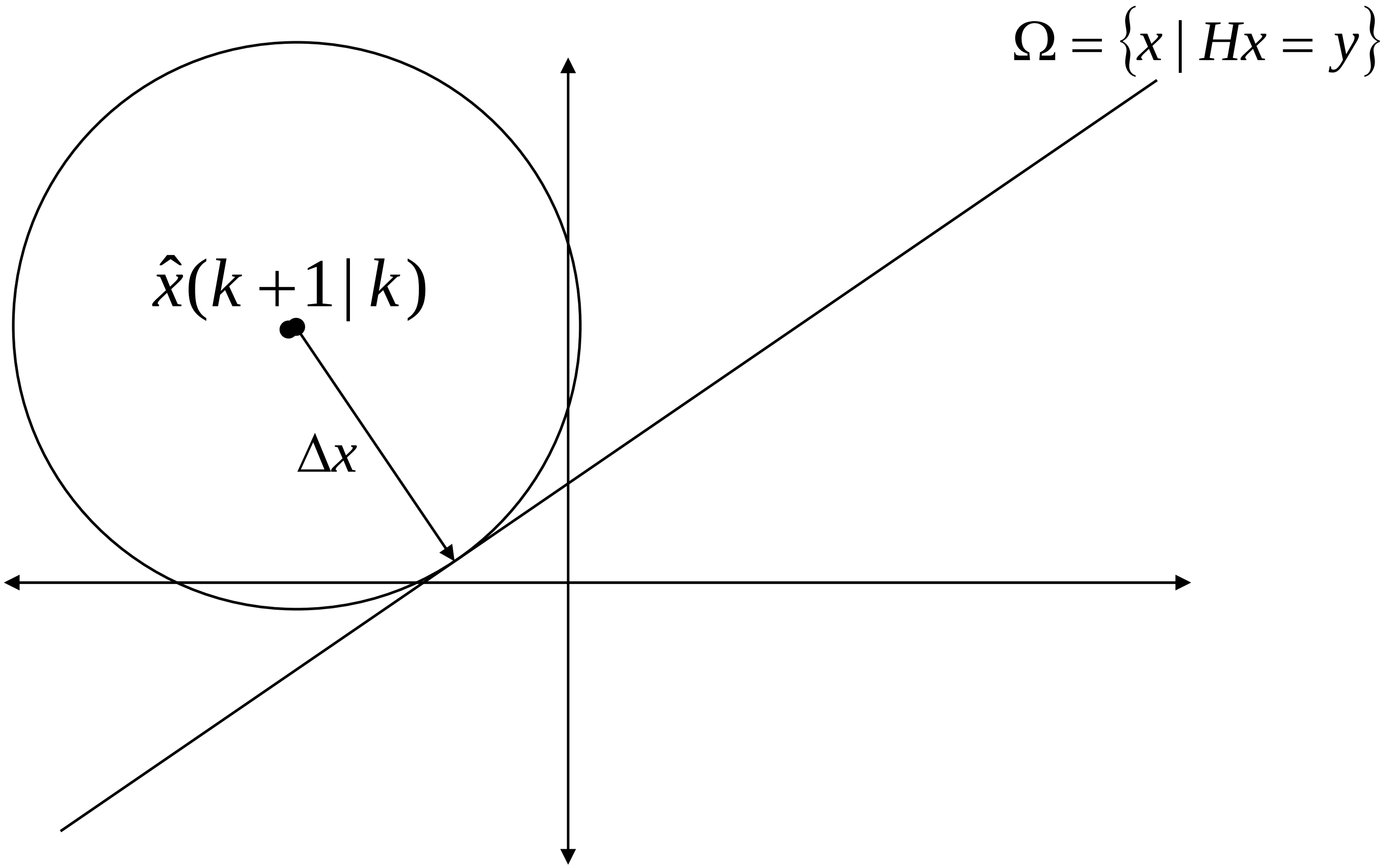
substituting $\Delta x = H^T K \nu$ yields

$$HH^T K \nu = \nu$$

$$\Rightarrow K = \left(HH^T\right)^{-1}$$

$$\Delta x = H^T \left(HH^T\right)^{-1} \nu$$

A Geometric Interpretation



A Simple State Observer

System:

$$x(k+1) = Fx(k) + Gu(k)$$

$$y(k) = Hx(k)$$

Observer:

1. prediction:

$$\hat{x}(k+1 \mid k) = F\hat{x}(k \mid k) + Gu(k)$$

2. compute correction:

$$\Delta x = H^T \left(HH^T\right)^{-1} \left(y(k+1) - H\hat{x}(k+1 \mid k)\right)$$

3. update:

$$\hat{x}(k+1 \mid k+1) = \hat{x}(k+1 \mid k) + \Delta x$$

Caveat #1

Note: The observer presented here is not a very good observer. Specifically, it is not guaranteed to converge for all systems. Still the intuition behind this observer is the same as the intuition behind the Kalman filter, and the problems will be fixed in the following slides.

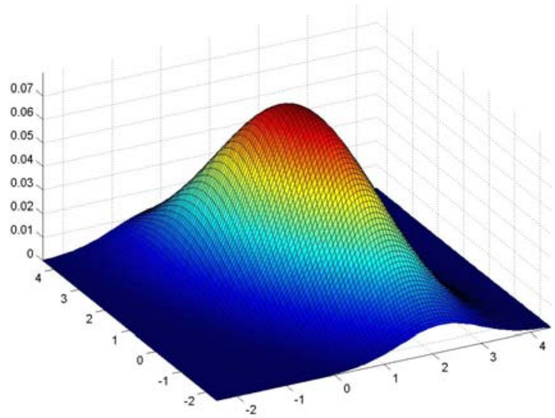
It really corrects only to the current sensor information, so if you are on the hyperplane but not at right place, you have no correction…. **I am waiving my hands here, look in book**

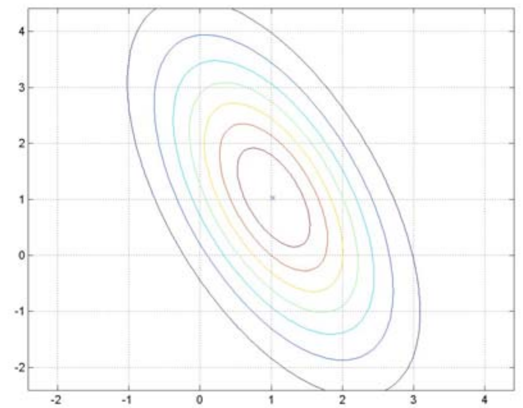
Estimating a *distribution* for *x*

Our estimate of *x* is not exact!
We can do better by estimating a joint Gaussian distribution *p(x)*.

$$p(x) = \frac{1}{(2\pi)^{n/2}|P|^{1/2}} e^{\frac{-1}{2}\left((x-\hat{x})^T P^{-1}(x-\hat{x})\right)}$$

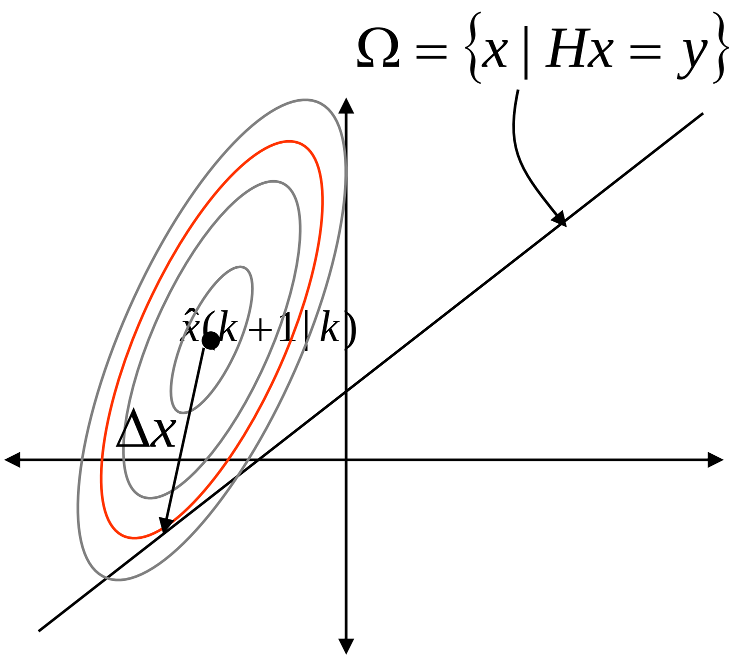
where $P = E\left((x-\hat{x})(x-\hat{x})^T\right)$ is the *covariance matrix*





Finding the correction (geometric intuition)

Given prediction $\hat{x}(k+1|k)$, covariance P, and output y, find Δx so that $\hat{x} = \hat{x}(k+1|k) + \Delta x$ is the "best" (i.e. most probable) estimate of x.



$$p(x) = \frac{1}{(2\pi)^{n/2}|P|^{1/2}} e^{\frac{-1}{2}\left((x-\hat{x})^T P^{-1} (x-\hat{x})\right)}$$

The most probable Δx is the one that :

1. satisfies $\hat{x}(k+1|k+1) = \hat{x}(k+1|k) + \Delta x$
2. minimizes $\Delta x^T P^{-1} \Delta x$

A new kind of distance

Suppose we define a new inner product on R^n to be :

$$\langle x_1, x_2 \rangle = x_1^T P^{-1} x_2 \qquad \text{(this replaces the old inner product } x_1^T x_2 \text{)}$$

Then we can define a new norm $\|x\|^2 = \langle x, x \rangle = x^T P^{-1} x$

The $\hat{x}$ in Ω that minimizes $\|\Delta x\|$ is the orthogonal projection of $\hat{x}(k+1 \mid k)$ onto Ω, so Δx is orthogonal to Ω.

$$\Rightarrow \langle \omega, \Delta x \rangle = 0 \text{ for } \omega \text{ in } \mathrm{T}\Omega = null(H)$$

$$\boxed{\langle \omega, \Delta x \rangle = \omega^T P^{-1} \Delta x = 0 \text{ iff } \Delta x \in column(PH^T)}$$

Finding the correction (for real this time!)

Assuming that Δx is linear in $\nu = y - H\hat{x}(k+1|k)$

$$\Delta x = PH^T K \nu$$

The condition $y = H(\hat{x}(k+1|k) + \Delta x) \Rightarrow H\Delta x = y - H\hat{x}(k+1|k) = \nu$

Substitution yields:

$$H\Delta x = \nu = HPH^T K \nu$$

$$\Rightarrow K = \left(HPH^T\right)^{-1}$$

$$\therefore \quad \Delta x = PH^T \left(HPH^T\right)^{-1} \nu$$

A Better State Observer

$$x(k+1) = Fx(k) + Gu(k) + v(k)$$
$$y(k) = Hx(k)$$

Sample of Guassian Dist. w/ COV Q

We can create a better state observer following the same 3. steps, but now we must also estimate the covariance matrix *P*.

We start with $x(k|k)$ and $P(k|k)$

Where did noise go?
Expected value…

Step 1: Prediction

$$\hat{x}(k+1 \mid k) = F\hat{x}(k \mid k) + Gu(k)$$

What about *P*? From the definition:

$$P(k \mid k) = E\left((x(k) - \hat{x}(k \mid k))(x(k) - \hat{x}(k \mid k))^T\right)$$

and

$$P(k+1 \mid k) = E\left((x(k+1) - \hat{x}(k+1 \mid k))(x(k+1) - \hat{x}(k+1 \mid k))^T\right)$$

Continuing Step 1

To make life a little easier, lets shift notation slightly:

$$P_{k+1}^{-} = E\left((x_{k+1} - \hat{x}_{k+1}^{-})(x_{k+1} - \hat{x}_{k+1}^{-})^T\right)$$

$$= E\left(\left(Fx_k + Gu_k + v_k - (F\hat{x}_k + Gu_k)\right)\left(Fx_k + Gu_k + v_k - (F\hat{x}_k + Gu_k)\right)^T\right)$$

$$= E\left(\left(F\left(x_k - \hat{x}_k\right) + v_k\right)\left(F\left(x_k - \hat{x}_k\right) + v_k\right)^T\right)$$

$$= E\left(F\left(x_k - \hat{x}_k\right)\left(x_k - \hat{x}_k\right)^T F^T + 2F\left(x_k - \hat{x}_k\right)v_k^T + v_k v_k^T\right)$$

$$= FE\left(\left(x_k - \hat{x}_k\right)\left(x_k - \hat{x}_k\right)^T\right)F^T + E\left(v_k v_k^T\right)$$

$$= FP_k F^T + Q$$

$$P(k+1 \mid k) = FP(k \mid k)F^T + Q$$

Step 2: Computing the correction

From step 1 we get $\hat{x}(k+1|k)$ and $P(k+1|k)$.

Now we use these to compute Δx :

$$\Delta x = P(k+1|k)H\left(HP(k+1|k)H^T\right)^{-1}\left(y(k+1) - H\hat{x}(k+1|k)\right)$$

For ease of notation, define W so that

$$\Delta x = W\nu$$

Step 3: Update

$$\hat{x}(k+1 \mid k+1) = \hat{x}(k+1 \mid k) + W\nu$$

$$P_{k+1} = E\left((x_{k+1} - \hat{x}_{k+1})(x_{k+1} - \hat{x}_{k+1})^T\right)$$

$$= E\left((x_{k+1} - \hat{x}_{k+1}^{-} - W\nu)(x_{k+1} - \hat{x}_{k+1}^{-} - W\nu)^T\right)$$

(just take my word for it…)

$$P(k+1 \mid k+1) = P(k+1 \mid k) - WHP(k+1 \mid k)H^T W^T$$

Just take my word for it…

$$P_{k+1} = E\left((x_{k+1} - \hat{x}_{k+1})(x_{k+1} - \hat{x}_{k+1})^T\right)$$

$$= E\left((x_{k+1} - \hat{x}^-_{k+1} - W\nu)(x_{k+1} - \hat{x}^-_{k+1} - W\nu)^T\right)$$

$$= E\left(\left((x_{k+1} - \hat{x}^-_{k+1}) - W\nu\right)\left((x_{k+1} - \hat{x}^-_{k+1}) - W\nu\right)^T\right)$$

$$= E\left((x_{k+1} - \hat{x}^-_{k+1})(x_{k+1} - \hat{x}^-_{k+1})^T - 2W\nu(x_{k+1} - \hat{x}^-_{k+1})^T + W\nu(W\nu)^T\right)$$

$$= P^-_{k+1} + E\left(-2WH(x_{k+1} - \hat{x}^-_{k+1})(x_{k+1} - \hat{x}^-_{k+1})^T + WH(x_{k+1} - \hat{x}^-_{k+1})(x_{k+1} - \hat{x}^-_{k+1})^T H^T W^T\right)$$

$$= P^-_{k+1} - 2WHP^-_{k+1} + WHP^-_{k+1}H^T W^T$$

$$= P^-_{k+1} - 2P^-_{k+1}H^T\left(HP^-_{k+1}H^T\right)^{-1}HP^-_{k+1} + WHP^-_{k+1}H^T W^T$$

$$= P^-_{k+1} - 2P^-_{k+1}H^T\left(HP^-_{k+1}H^T\right)^{-1}\left(HP^-_{k+1}H^T\right)\left(HP^-_{k+1}H^T\right)^{-1}HP^-_{k+1} + WHP^-_{k+1}H^T W^T$$

$$= P^-_{k+1} - 2WHP^-_{k+1}H^T W^T + WHP^-_{k+1}H^T W^T$$

Better State Observer Summary

System:
$$x(k+1) = Fx(k) + Gu(k) + v(k)$$
$$y(k) = Hx(k)$$

Observer

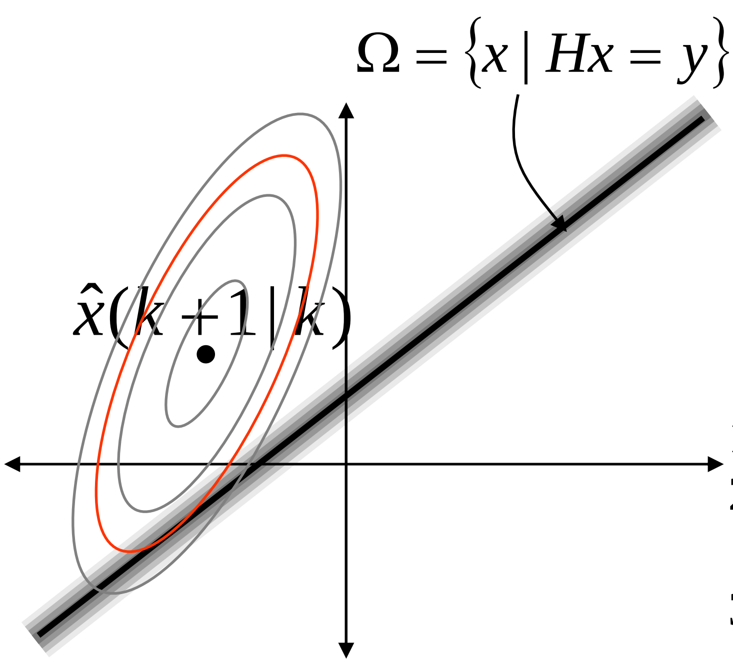
1. Predict
$$\hat{x}(k+1 \mid k) = F\hat{x}(k \mid k) + Gu(k)$$
$$P(k+1 \mid k) = FP(k \mid k)F^T + Q$$

2. Correction
$$W = P(k+1 \mid k)H\left(HP(k+1 \mid k)H^T\right)^{-1}$$
$$\Delta x = W\left(y(k+1) - H\hat{x}(k+1 \mid k)\right)$$

3. Update
$$\hat{x}(k+1 \mid k+1) = \hat{x}(k+1 \mid k) + W\nu$$
$$P(k+1 \mid k+1) = P(k+1 \mid k) - WHP(k+1 \mid k)H^T W^T$$

•Note: there is a problem with the previous slide, namely the covariance matrix of the estimate P will be singular. This makes sense because with perfect sensor measurements the uncertainty in some directions will be zero. There is no uncertainty in the directions perpendicular to Ω

Finding the correction (with output noise)



$$y = Hx + w$$

The previous results require that you know which hyperplane to aim for. Because there is now sensor noise, we don't know where to aim, so we can't directly use our method.

If we can determine which hyperplane aim for, then the previous result would apply.

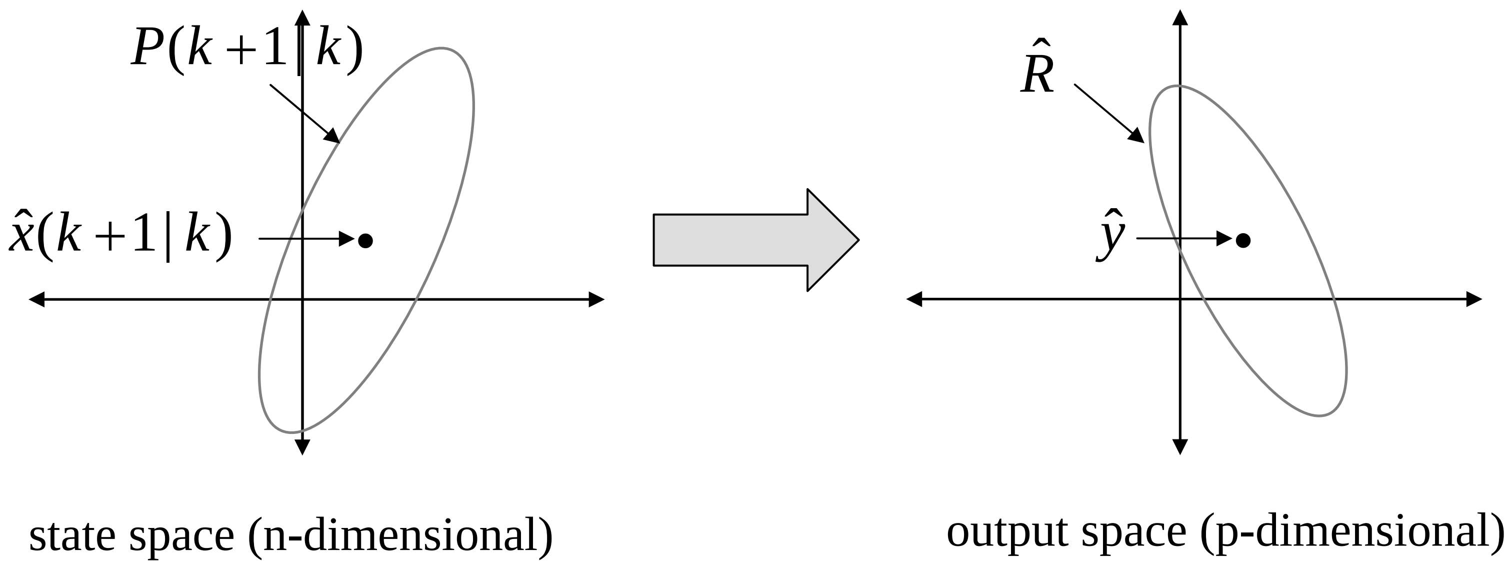
We find the hyperplane in question as follows:

1. project estimate into output space
2. find most likely point in output space based on measurement and projected prediction
3. the desired hyperplane is the preimage of this point

Projecting the prediction
(putting current state estimates into sensor space)

$$\hat{x}(k+1|k) \rightarrow \hat{y} = H\hat{x}(k+1|k)$$

$$P(k+1|k) \rightarrow \hat{R} = HP(k+1|k)H^T$$



Finding most likely output

ence independent, so multiply them because we want them both to be true at the same time

The objective is to find the most likely output that results from the independent Gaussian distributions

(y, R) measurement and associate covariance

$(\hat{y}, \hat{R})$ projected prediction (what you think your measurements should be and how confident you are)

Fact (Kalman Gains): The product of two Gaussian distributions given by mean/covariance pairs (x_1,C_1) and (x_2,C_2) is proportional to a third Gaussian with mean

$$x_3 = x_1 + K(x_2 - x_1)$$

and covariance

$$C_3 = C_1 - KC_1$$

where

$$K = C_1(C_1 + C_2)^{-1}$$

Strange, but true, this is symmetric

Most likely output (cont.)

Using the Kalman gains, the most likely output is

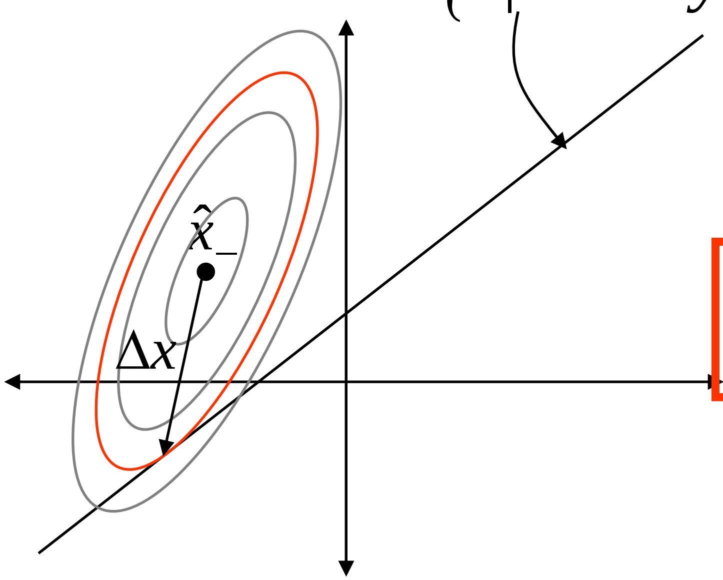
$$y^{*} = \hat{y} + \left(\hat{R}\left(\hat{R} + R\right)^{-1} \right)(\hat{y} - y)$$

$$= H\hat{x}(k+1|k) + \left(HPH^{T}\left(HPH^{T} + R\right)^{-1}\right)\left(H\hat{x}(k+1|k) - y\right)$$

Finding the Correction

Now we can compute the correction as we did in the noiseless case, this time using y* instead of y. In other words, y* tells us which hyperplane to aim for.

$$\Omega = \{x \mid Hx = y^*\}$$



The result is:

$$\Delta x = PH^T\left(HPH^T\right)^{-1}\left(y^* - H\hat{x}(k+1 \mid k)\right)$$

ot going all the way to y, but splitting the difference between how confident you are with your Sensor and process noise

Finding the Correction (cont.)

$$\Delta x = PH^T \left(HPH^T \right)^{-1} \left(\mathrm{y}^* - H\hat{x}(k+1 \mid k) \right)$$

$$= PH^T \left(HPH^T \right)^{-1} \left(H\hat{x} + HPH^T \left(HPH^T + R \right)^{-1} \left(y - H\hat{x}(k+1 \mid k) \right) - H\hat{x}(k+1 \mid k) \right)$$

$$= PH^T \left(HPH^T + R \right)^{-1} \left(y - H\hat{x}(k+1 \mid k) \right)$$

For convenience, we define

$$W = PH^T \left(HPH^T + R \right)^{-1}$$

So that

$$\boxed{\Delta x = W \left(y - H\hat{x}(k+1 \mid k) \right)}$$

Correcting the Covariance Estimate

The covariance error estimate correction is computed from the definition of the covariance matrix, in much the same way that we computed the correction for the "better observer". The answer turns out to be:

$$P(k+1|k+1) = P(k+1|k) - W\left(HP(k+1|k)H^T\right)W^T$$

LTI Kalman Filter Summary

System:

$$x(k+1) = Fx(k) + Gu(k) + v(k)$$
$$y(k) = Hx(k) + w(k)$$

Kalman Filter

1. Predict

$$\hat{x}(k+1|k) = F\hat{x}(k|k) + Gu(k)$$
$$P(k+1|k) = FP(k|k)F^T + Q$$

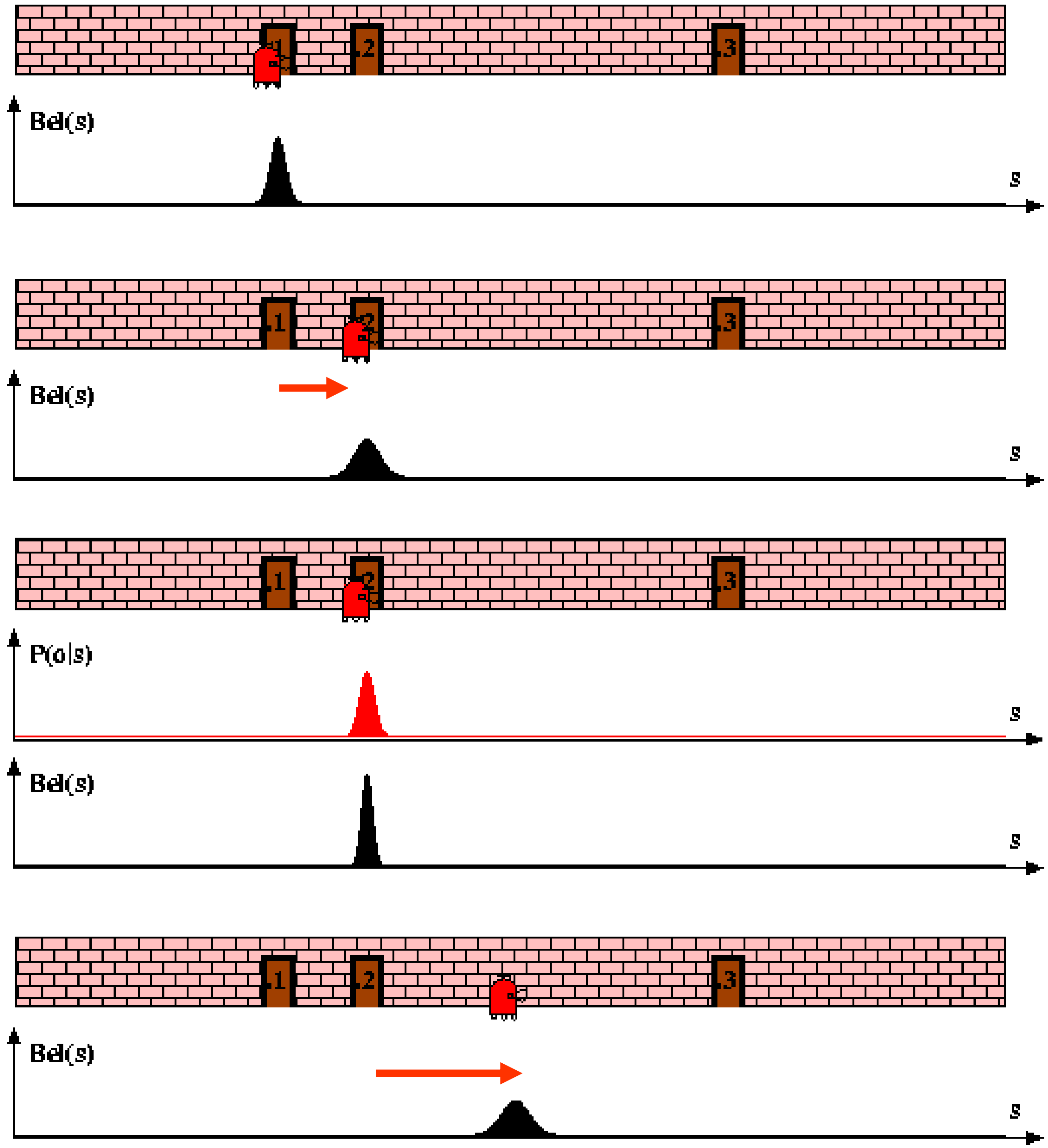
2. Correction

$$S = HP(k+1|k)H^T + \mathrm{R}$$
$$W = P(k+1|k)H^T S^{-1}$$
$$\Delta x = W\left(y(k+1) - H\hat{x}(k+1|k)\right)$$

3. Update

$$\hat{x}(k+1|k+1) = \hat{x}(k+1|k) + W\nu$$
$$P(k+1|k+1) = P(k+1|k) - WSW^T$$

Kalman Filters



Kalman Filter for Dead Reckoning

- Robot moves along a straight line with state $x = [x_r, v_r]^T$
- u is the force applied to the robot
- Newton tells us $\frac{dv_r}{dt} = \frac{u}{m}$ or $$\frac{v_r(k+1) - v_r(k)}{T} = \frac{u(k)}{m}$$

$$\begin{aligned} x(k+1) &= \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix} x(k) + \begin{bmatrix} 0 \\ \frac{T}{m} \end{bmatrix} u(k) + v(k) \\ &\triangleq Fx(k) + Gu(k) + v(k), \end{aligned}$$

← Process noise from a zero mean Gaussian V

- Robot has velocity sensor

$$\begin{aligned} y(k+1) &= \begin{bmatrix} 0 & 1 \end{bmatrix} x(k) + w(k) \\ &\triangleq Gx(k) + w(k), \end{aligned}$$

← Sensor noise from a zero mean Gaussian W

Set up

$$x(k+1) = \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix} x(k) + \begin{bmatrix} 0 \\ \frac{T}{m} \end{bmatrix} u(k) + v(k)$$

$$\triangleq Fx(k) + Gu(k) + v(k),$$

$$y(k+1) = \begin{bmatrix} 0 & 1 \end{bmatrix} x(k) + w(k)$$

$$\triangleq Gx(k) + w(k),$$

Assume

$m = 1$, $W = .5$, $T = 0.5$, and

$$V = \begin{bmatrix} 0.2 & 0.05 \\ 0.05 & 0.1 \end{bmatrix}$$

At some time *k* $u(k) = 0$

$$\hat{x}(k|k) = [2,\ 4]^T$$

$$P(k|k) = \begin{bmatrix} 1 & 0 \\ 0 & 2 \end{bmatrix}$$

PUT ELLIPSE FIGURE HERE

Observability

Recall from last time

$$
\begin{aligned}
x(k+1) &= Fx(k) + Gu(k) + v(k) \\
y(k) &= Hx(k) + w(k)
\end{aligned}
$$

is observable if and only if the observability matrix

$$
Q = \begin{bmatrix} H \\ HF \\ HF^2 \\ \vdots \\ HF^{(n-1)} \end{bmatrix}
$$

has rank n.

Actually, previous example is not observable but still nice to use Kalman filter

Extended Kalman Filter

- Life is not linear

$$\begin{aligned} x(k+1) &= f(x(k), u(k), k) + v(k) \\ y(k) &= h(x(k), k) + w(k), \end{aligned}$$

$$f\colon \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{Z}^+ \to \mathbb{R}^n$$

$$h\colon \mathbb{R}^n \times \mathbb{Z}^+ \to \mathbb{R}^p$$

- Predict

$$\begin{aligned} \hat{x}(k+1|k) &= f(\hat{x}(k|k), u(k), k) \\ P(k+1|k) &= F(k)P(k|k)F(k)^T + V(k) \end{aligned}$$

where

$$F(k) = \left.\frac{\partial f}{\partial x}\right|_{x=\hat{x}(k|k)} = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \cdots & \frac{\partial f_2}{\partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f_n}{\partial x_1} & \frac{\partial f_n}{\partial x_2} & \cdots & \frac{\partial f_n}{\partial x_n} \end{bmatrix}_{x=\hat{x}(k|k)} .$$

Extended Kalman Filter

- Update

$$
\begin{aligned}
\hat{x}(k+1|k+1) &= \hat{x}(k+1|k) + R\nu \\
P(k+1|k+1) &= P(k+1|k) - RH(k+1)P(k+1|k)
\end{aligned}
$$

$$
\begin{aligned}
\nu &= y(k+1) - h(x(k+1|k), k+1) \\
S &= H(k+1)P(k+1|k)H(k+1)^T + W(k+1) \\
R &= P(k+1|k)H(k+1)^T S^{-1}
\end{aligned}
$$

and

$$
H(k+1) = \left.\frac{\partial h}{\partial x}\right|_{x=\hat{x}(k+1|k)} = \begin{bmatrix} \frac{\partial h_1}{\partial x_1} & \frac{\partial h_1}{\partial x_2} & \cdots & \frac{\partial h_1}{\partial x_n} \\ \frac{\partial h_2}{\partial x_1} & \frac{\partial h_2}{\partial x_2} & \cdots & \frac{\partial h_2}{\partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial h_p}{\partial x_1} & \frac{\partial h_p}{\partial x_2} & \cdots & \frac{\partial h_p}{\partial x_n} \end{bmatrix}_{x=\hat{x}(k+1|k)} .
$$

EKF for Range-Bearing Localization

- State $x(k) = [x_r(k), y_r(k), \theta_r(k)]^T$ position and orientation
- Input $u(k) = [u_1(k), u_2(k)]^T$ forward and rotational velocity
- Process Model

$$x(k+1) = \begin{bmatrix} \cos\theta_r(k)u_1(k) + x_r(k) \\ \sin\theta_r(k)u_1(k) + y_r(k) \\ u_2(k) + \theta_r(k) \end{bmatrix} + v(k)$$

- n_l landmarks $(x_{\ell i}, y_{\ell i})$
- can only see *p(k)* of them at *k*

Association map

$$a : \{1, 2, \ldots, p(k)\} \rightarrow \{1, 2, \ldots, n_\ell\}$$

$$y(k) = \begin{bmatrix} h_1(x(k), a(1)) \\ h_2(x(k), a(2)) \\ \vdots \\ h_{p(k)}(x(k), a(p(k))) \end{bmatrix} + \begin{bmatrix} w_1(k) \\ w_2(k) \\ \vdots \\ w_{p(k)}(k) \end{bmatrix}$$

$$h_j(x(k), j) = \begin{bmatrix} \sqrt{(x_r(k) - x_{\ell j})^2 + (y_r(k) - y_{\ell j})^2} \\ \mathrm{atan2}(y_r(k) - y_{\ell j}, x_r(k) - x_{\ell j}) - \theta_r(k) \end{bmatrix}$$

Be wise, and linearize..

$$F(k) = \left.\frac{\partial f}{\partial x}\right|_{x=\hat{x}(k|k)} \qquad F = \begin{bmatrix} 1 & 0 & -\sin\theta_r(k)u_1(k) \\ 0 & 1 & \cos\theta_r(k)u_1(k) \\ 0 & 0 & 1 \end{bmatrix}$$

$$H(k+1) = \begin{bmatrix} H_1(k+1, a(1)) \\ H_2(k+1, a(2)) \\ \vdots \\ H_{p(k+1)}(k+1, a(p(k+1))) \end{bmatrix} = \begin{bmatrix} \left.\frac{\partial h_1}{\partial x}\right|_{x=\hat{x}(k+1|k)} \\ \left.\frac{\partial h_2}{\partial x}\right|_{x=\hat{x}(k+1|k)} \\ \vdots \\ \left.\frac{\partial h_{p(k+1)}}{\partial x}\right|_{x=\hat{x}(k+1|k)} \end{bmatrix}$$

$$H_i(k+1, j) = \begin{bmatrix} \frac{(\hat{x}_r(k+1|k) - x_{\ell j})}{\sqrt{(\hat{x}_r(k+1|k) - x_{\ell j})^2 + (\hat{y}_r(k+1|k) - y_{\ell j})^2}} & \frac{(\hat{y}_r(k+1|k) - y_{\ell j})}{\sqrt{(\hat{x}_r(k+1|k) - x_{\ell j})^2 + (\hat{y}_r(k+1|k) - y_{\ell j})^2}} & 0 \\ \frac{-(\hat{y}_r(k+1|k) - y_{\ell j})}{1 + \left(\frac{\hat{y}_r(k+1|k) - y_{\ell j}}{\hat{x}_r(k+1|k) - x_{\ell j}}\right)^2 \left(\hat{x}_r(k+1|k) - x_{\ell j}\right)^2} & \frac{1}{1 + \left(\frac{\hat{y}_r(k+1|k) - y_{\ell j}}{\hat{x}_r(k+1|k) - x_{\ell j}}\right)^2 (\hat{x}_r(k+1|k) - x_{\ell j})} & -1 \end{bmatrix}$$

Data Association

- BIG PROBLEM

Ith measurement corresponds to the jth landmark

$$\nu_{ij} \triangleq y_i(k+1) - h_i(\hat{x}(k+1|k), j)$$

innovation

$$\chi^2_{ij} = \nu_{ij} S^{-1}_{ij} \nu^T_{ij}$$

where

$$S_{ij} = H_i(k+1, j) P(k+1 \mid k) H_i(k+1, j)^T + W_i(k+1)$$

Pick the smallest

Kalman Filter for SLAM (simple)

state

$$x = \begin{bmatrix} x_r & y_r & x_{\ell 1} & y_{\ell 1} & x_{\ell 2} & y_{\ell 2} & \cdots & x_{\ell n_\ell} & y_{\ell n_\ell} \end{bmatrix}^T$$

Inputs are commands to x and y velocities, a bit naive

Process model

$$x(k+1) = x(k) + \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \\ 0 & 0 \\ \vdots & \vdots \\ 0 & 0 \end{bmatrix} \begin{bmatrix} u_x(k) \\ u_y(k) \end{bmatrix} + \begin{bmatrix} v_{rx}(k) \\ v_{ry}(k) \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

$$x(k+1) = Fx(k) + Gu(k) + v(k)$$

$$V(k) = \begin{bmatrix} V_r(k) & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 \end{bmatrix}$$

Kalman Filter for SLAM

$$y_i(k) = \begin{bmatrix} x_{\ell i}(k) - x_r(k) \\ y_{\ell i}(k) - y_r(k) \end{bmatrix} + w_i(k)$$

$$y_i(k) = H_i x(k) + w_i(k),$$

$$H_i = \begin{bmatrix} -1 & 0 & 0 & \cdots & 0 & 1 & 0 & 0 & \cdots & 0 \\ 0 & -1 & 0 & \cdots & 0 & 0 & 1 & 0 & \cdots & 0 \end{bmatrix}$$

$$y(k) = Hx(k) + w(k),$$

where

$$H = \begin{bmatrix} H_1 \\ H_2 \\ \vdots \\ H_{n_\ell} \end{bmatrix}, \quad \text{and} \quad w(k) = \begin{bmatrix} w_1(k) \\ w_2(k) \\ \vdots \\ w_{n_\ell}(k) \end{bmatrix},$$

and the covariance matrix associated with $w(k)$ is

$$W(k) = \begin{bmatrix} W_1(k) & 0 & \cdots & 0 \\ 0 & W_2(k) & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & W_{n_\ell} \end{bmatrix},$$

Range Bearing

Inputs are forward and rotational velocities

$$y_i(k) = \begin{bmatrix} \sqrt{(x_{\ell i}(k) - x_r(k))^2 + (y_{\ell i}(k) - y_r(k))^2} \\ \text{atan2}((y_{\ell i}(k) - y_r(k)), (x_{\ell i}(k) - x_r(k)) - \theta_r(k) \end{bmatrix} + w_i(k).$$

$$H_i = \begin{bmatrix} \frac{\partial y_i}{\partial x_r} \\ \frac{\partial y_i}{\partial y_r} \end{bmatrix} =$$

$$\begin{bmatrix} \frac{-x_{\ell i}(k)+x_r(k)}{\rho_i} & \frac{-y_{\ell i}(k)+y_r(k)}{\rho_i} & 0 & \dots & \frac{x_{\ell i}(k)-x_r(k)}{\rho_i} & \frac{y_{\ell i}(k)-y_r(k)}{\rho_i} & \dots \\ \frac{y_{\ell i}(k)-y_r(k)}{\rho_i^2} & \frac{-x_{\ell i}(k)+x_r(k)}{\rho_i^2} & -1 & \dots & \frac{-y_{\ell i}(k)+y_r(k)}{\rho_i^2} & \frac{x_{\ell i}(k)-x_r(k)}{\rho_i^2} & \dots \end{bmatrix}$$

ρ_i is the range of the landmark

$$\chi_{ij}^2 = (y(k)_i - h(k)_j)^T S_{ij} (y(k)_i - h(k)_j).$$

Greg's Notes: Some Examples

- Point moving on the line according to f = m a
  - state is position *and* velocity
  - input is force
  - sensing should be position
- Point in the plane under Newtonian laws
- Nonholonomic kinematic system (no dynamics)
  - state is workspace configuration
  - input is velocity command
  - sensing could be direction and/or distance to beacons
- Note that all of dynamical systems are "open-loop" integration
- Role of sensing is to "close the loop" and pin down state

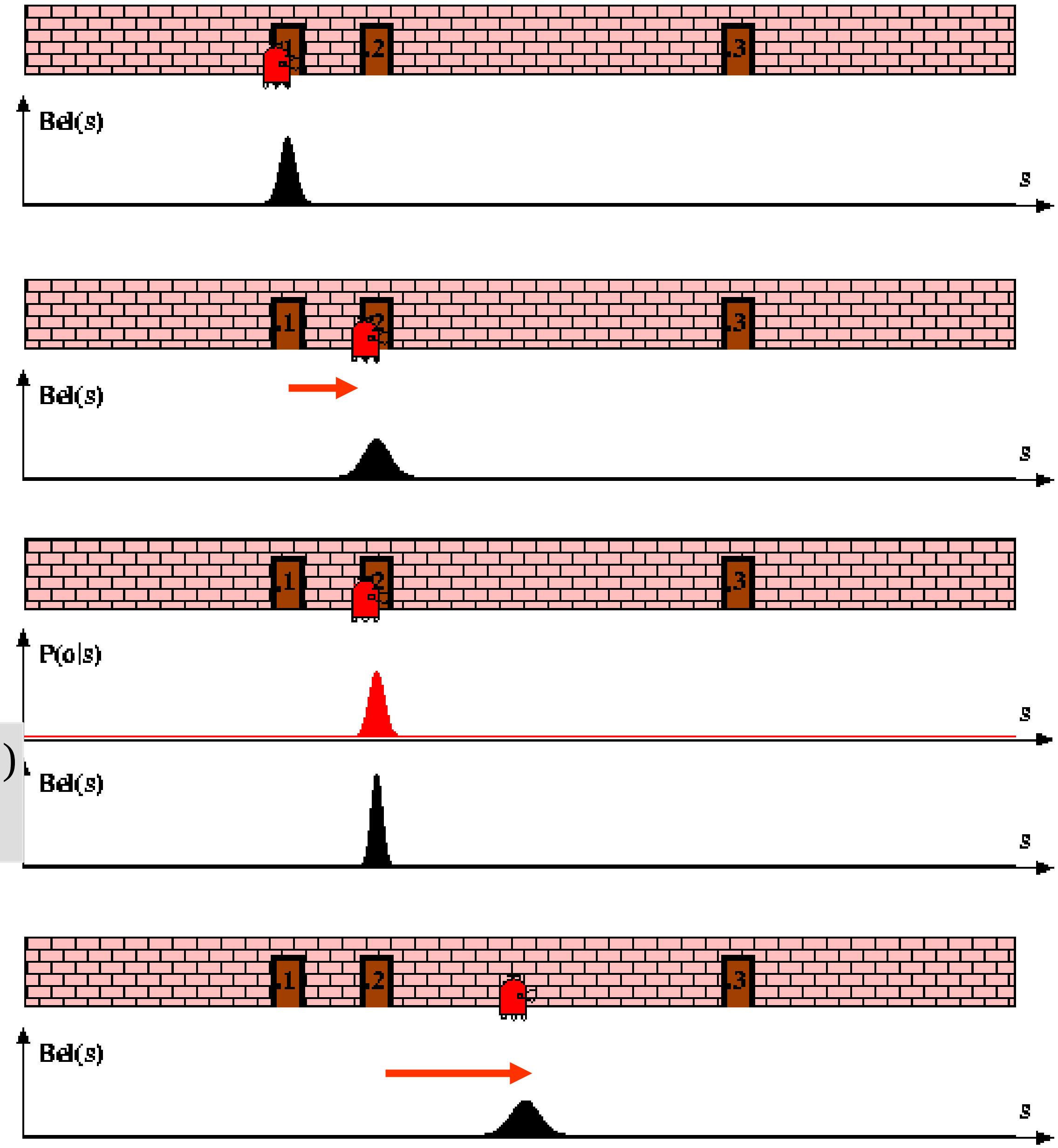
Kalman Filters

$$Bel(x_t) = N(\mu_t, \sigma_t^2)$$

$$Bel(x_{t+1}^-) = \begin{cases} \mu_{t+1}^- = \mu_t + Bu_t \\ \sigma^{-2}_{t+1} = A^2\sigma_t^2 + \sigma_{act}^2 \end{cases}$$

$$Bel(x_{t+1}) = \begin{cases} \mu_{t+1} = \mu_{t+1}^- + K_{t+1}(\mu_{z_t+1} - \mu_{t+1}^-) \\ \sigma_{t+1}^2 = (1 - K_{t+1})\sigma^{-2}_{t+1} \end{cases}$$

$$Bel(x_{t+2}^-) = \begin{cases} \mu_{t+2}^- = \mu_{t+1} + Bu_{t+1} \\ \sigma^{-2}_{t+2} = A^2\sigma_{t+1}^2 + \sigma_{act}^2 \end{cases}$$



Kalman Filter Algorithm

1. Algorithm **Kalman_filter**(<μ,Σ>, *d*):

2. If *d* is a perceptual data item *y* then
3. $K = \Sigma C^T \left(C \Sigma C^T + \Sigma_{obs} \right)^{-1}$
4. $\mu = \mu + K(z - C\mu)$
5. $\Sigma = (I - KC)\Sigma$
6. Else if *d* is an action data item *u* then
7. $\mu = A\mu + Bu$
8. $\Sigma = A \Sigma A^T + \Sigma_{act}$
9. Return <μ,Σ>

Limitations

- Very strong assumptions:
  - Linear state dynamics
  - Observations linear in state
- What can we do if system is not linear?
  - Non-linear state dynamics
  - Non-linear observations

$$X_{t+1}^{-} = AX_t + Bu_t + \Sigma_{act}$$

$$Z_t = CX_t + \Sigma_{obs}$$

$$X_{t+1}^{-} = f(X_t, u_t, \Sigma_{act})$$

$$Z_t = c(X_t, \Sigma_{obs})$$

- Linearize it!
  - Determine Jacobians of dynamics *f* and observation function *c* w.r.t the current state *x* and the noise.

$$A_{ij} = \frac{\partial f_i}{\partial x_j}(x_t, u_t, 0) \quad C_{ij} = \frac{\partial c_i}{\partial x_j}(x_t, 0) \quad W_{ij} = \frac{\partial f_i}{\partial \Sigma_{act\,j}}(x_t, u_t, 0) \quad V_{ij} = \frac{\partial c_i}{\partial \Sigma_{obs\,j}}(x_t, 0)$$

Extended Kalman Filter Algorithm

Algorithm **Extended_Kalman_filter**(<μ,Σ>, *d*):

If *d* is a perceptual data item *z* then

$$K = \Sigma C^T \left(C \Sigma C^T + V \Sigma_{obs} V^T \right)^{-1} \quad \longleftarrow \quad K = \Sigma C^T \left(C \Sigma C^T + \Sigma_{obs} \right)^{-1}$$

$$\mu = \mu + K\left(z - c(\mu, 0)\right) \quad \longleftarrow \quad \mu = \mu + K(z - C\mu)$$

$$\Sigma = (I - KC)\Sigma \quad \longleftarrow \quad \Sigma = (I - KC)\Sigma$$

Else if *d* is an action data item *u* then

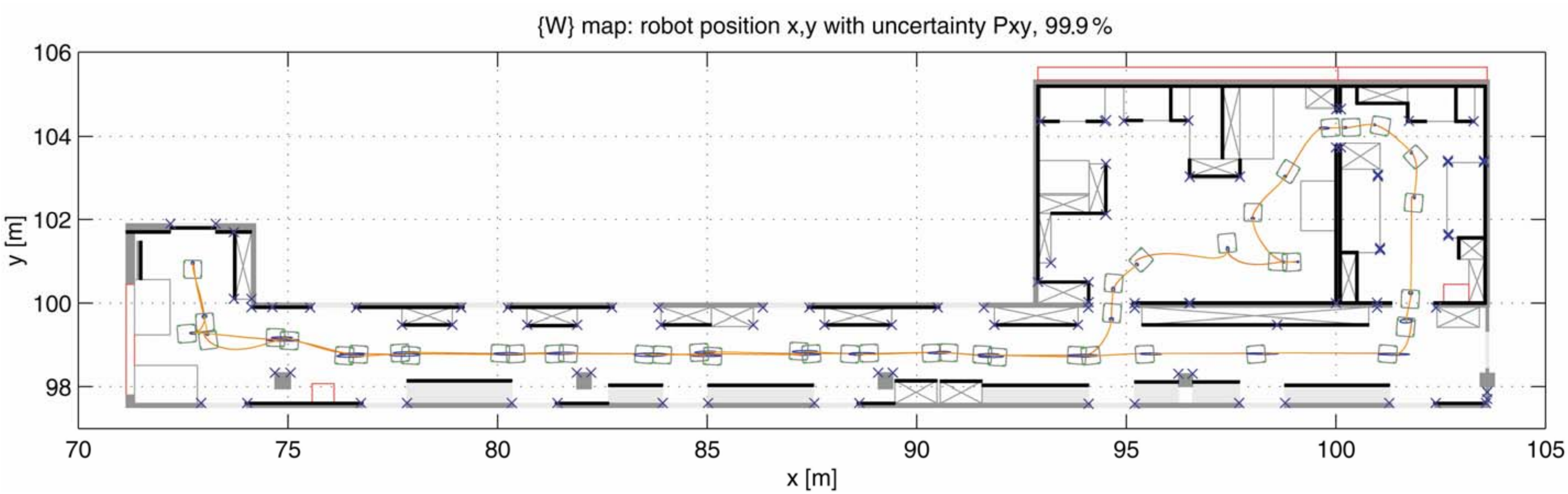
$$\mu = f(\mu, u, 0) \quad \longleftarrow \quad \mu = A\mu + Bu$$

$$\Sigma = A \Sigma A^T + W \Sigma_{act} W^T \quad \longleftarrow \quad \Sigma = A \Sigma A^T + \Sigma_{act}$$

Return <μ,Σ>

Kalman Filter-based Systems (2)

- [Arras et al. 98]:
  - Laser range-finder and vision
  - High precision (<1cm accuracy)



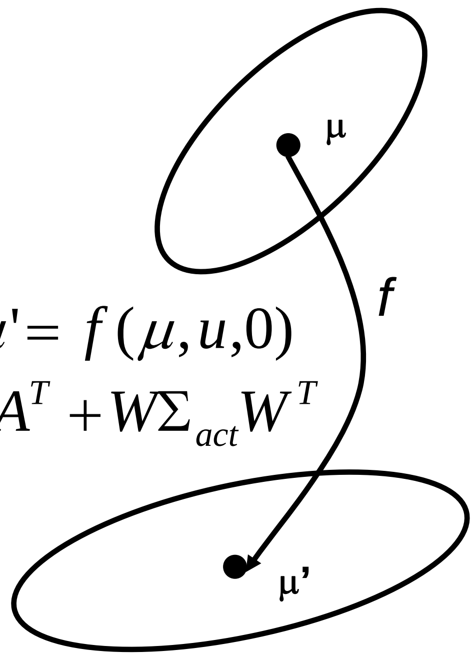
Courtesy of K. Arras

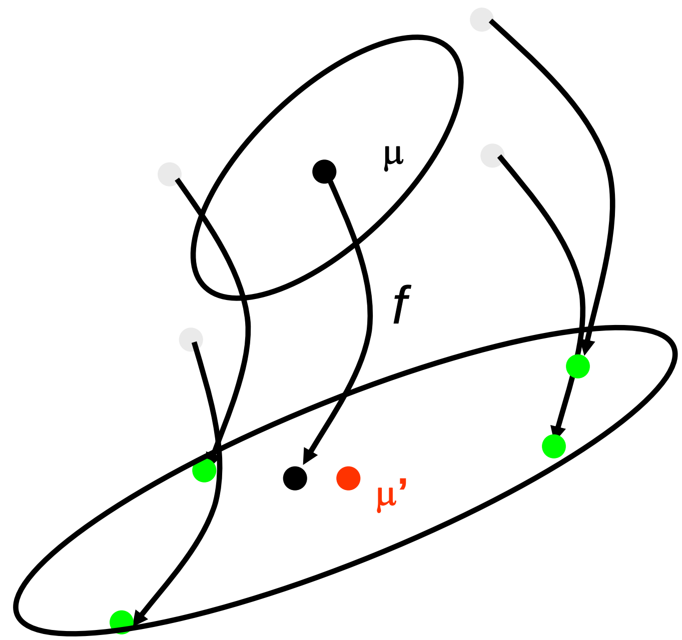
Unscented Kalman Filter

- Instead of linearizing, pass several points from the Gaussian through the non-linear transformation and re-compute a new Gaussian.
- Better performance (theory and practice).

$$\mu' = f(\mu, u, 0)$$

$$\Sigma = A \Sigma A^T + W \Sigma_{act} W^T$$





Kalman Filters and SLAM

- Localization: state is the location of the robot
- Mapping: state is the location of beacons
- SLAM: state combines both
- Consider a simple fully-observable holonomic robot
  - $x(k+1) = x(k) + u(k)\, dt + v$
  - $y_i(k) = p_i - x(k) + w$
- If the state is $(x(k), p_1, p_2 \ldots)$ then we can write a linear observation system
  - note that if we don't have some fixed beacons, our system is *unobservable* (we can't fully determine all unknown quantities)