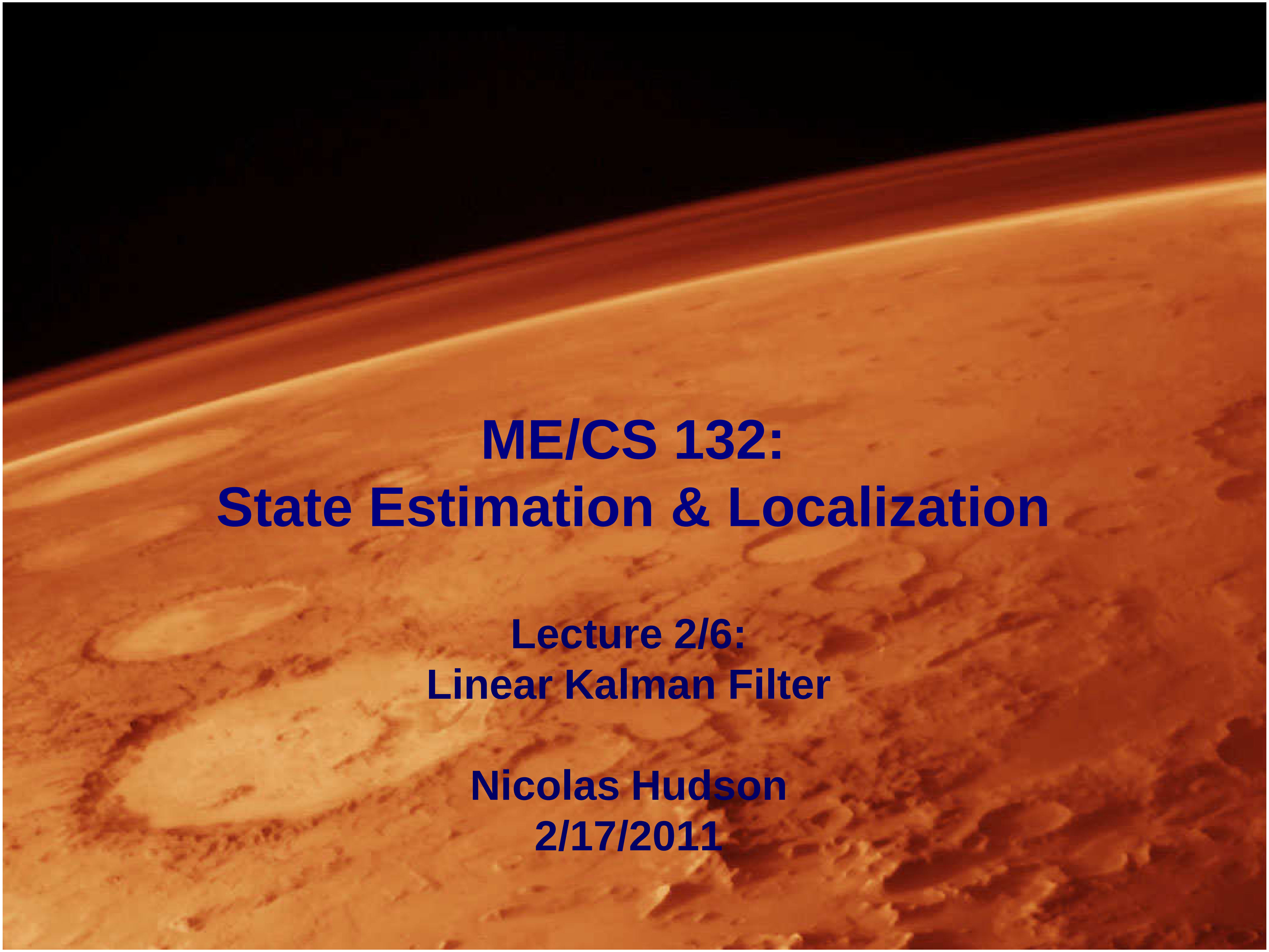


ME/CS 132: State Estimation & Localization

Lecture 2/6:
Linear Kalman Filter

Nicolas Hudson
2/17/2011

State Estimation & Localization Overview

- (1/6) Introduction

- (2/6) TODAY: Linear Kalman Filter
  - State Space Models (&Markov Systems)
  - Recursive Bayesian Filtering
  - Linear Kalman Filter

- (3/6) Extended Kalman Filter and Unscented Kalman Filter
- (4/6) Particle Filters
- (5/6) Simultaneous Localization and Mapping (SLAM)
- (6/6) Issues in SLAM

State Space Systems

Linear State Space Model (Gauss-Markov System**)**

Transition Function

$$x_k = A_k x_{k-1} + B_k u_k + q_k$$

$$z_k = C_k x_k + r_k$$

Measurement Model

Process Noise

$$q_k \sim N(0, Q)$$

$$r_k \sim N(0, R)$$

Measurement Noise

State	$x_k \in \mathbb{R}^n$
Measurement	$z_k \in \mathbb{R}^m$
Input	$u_k \in \mathbb{R}^l$
Parameters	$A_k \in \mathbb{R}^{n \times n}, B_k \in \mathbb{R}^{n \times l}, C_k \in \mathbb{R}^{m \times n}$

State Space Systems

Linear State Space Model (Gauss-Markov System**)**

Transition Function

Process Noise

$$x_k = A_k x_{k-1} + B_k u_k + q_k \qquad q_k \sim N(0,Q)$$

$$z_k = C_k x_k + r_k \qquad r_k \sim N(0,R)$$

Measurement Model

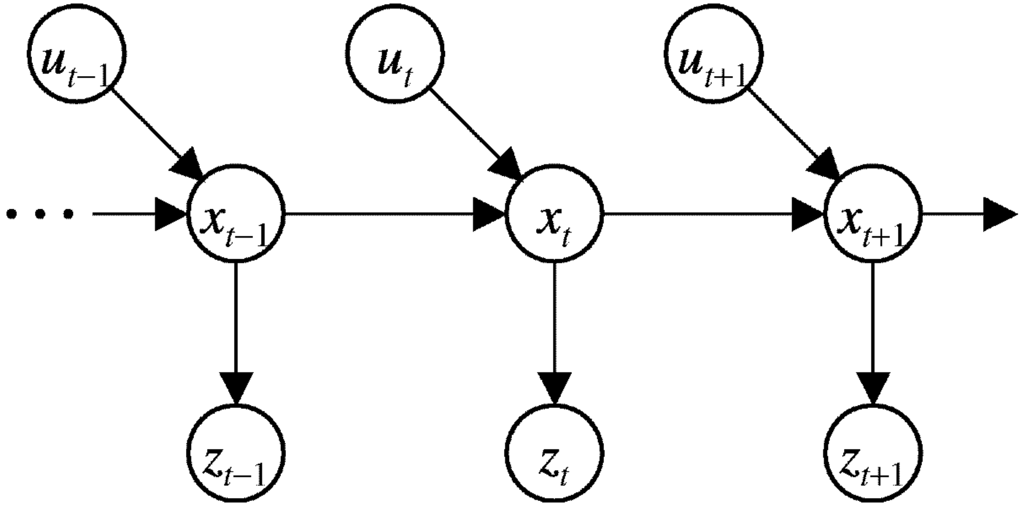
Measurement Noise

In probabilistic terms the model is

$$p(x_k \mid x_{k-1}) = N(A_k x_{k-1} + B_k u_k, Q)$$

$$p(z_k \mid x_k) = N(C_k x_k, R)$$

State Space is Markovian



$$x_k = A_k x_{k-1} + B_k u_k + q_k$$

$$z_k = C_k x_k + r_k$$

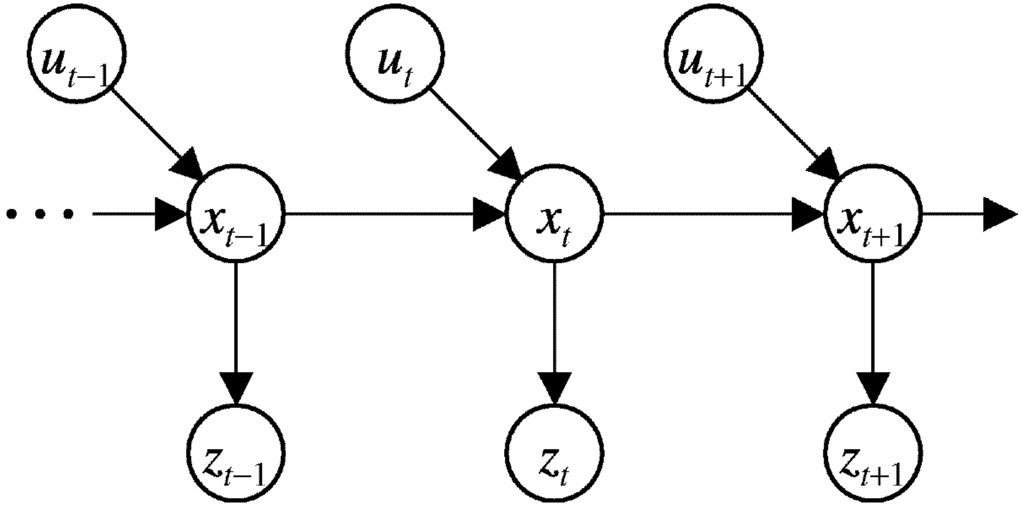
Markov property 1:

Future x_{k+1} is independent of the **past** given the **present** x_k

$$p(x_k \mid x_{1:t-1}, z_{1:k-1}, u_{1:k}) \;=\; p(x_k \mid x_{k-1}, u_k)$$

$$x_{1:k} \triangleq [x_1, x_2, ..., x_k]$$

State Space is Markovian



$$x_k = A_k x_{k-1} + B_k u_k + q_k$$

$$z_k = C_k x_k + r_k$$

Markov property 2:

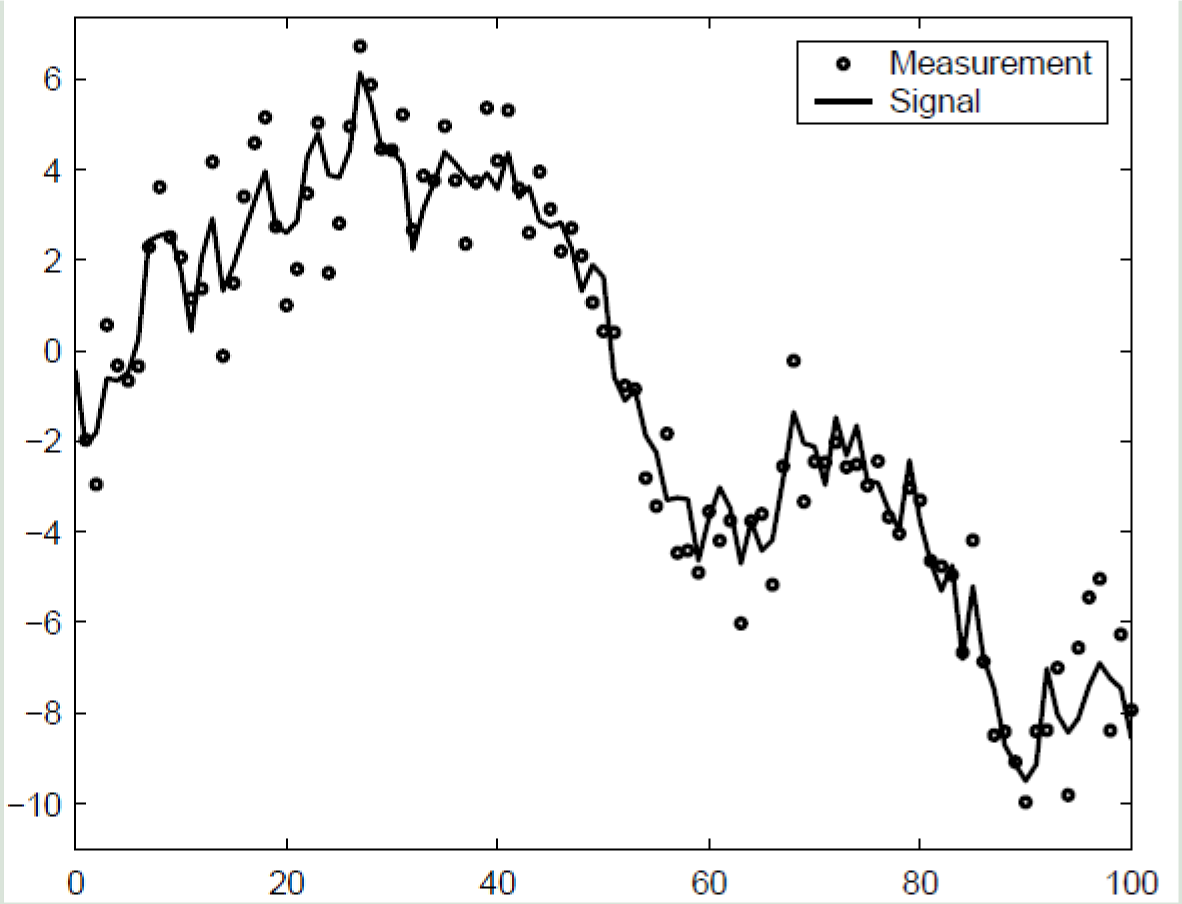
Measurements are **conditionally independent** given the current state

$$p(z_k \mid x_{1:k}, z_{1:k-1}, u_{1:k}) = p(z_k \mid x_k)$$

Example: Random Walk

$$x_k = x_{k-1} + q_k \qquad q_k \sim N(0,Q)$$

$$z_k = x_k + r_k \qquad r_k \sim N(0,R)$$



What are we trying to do?

Compute the distribution: $p(x_k \mid z_{1:k})$

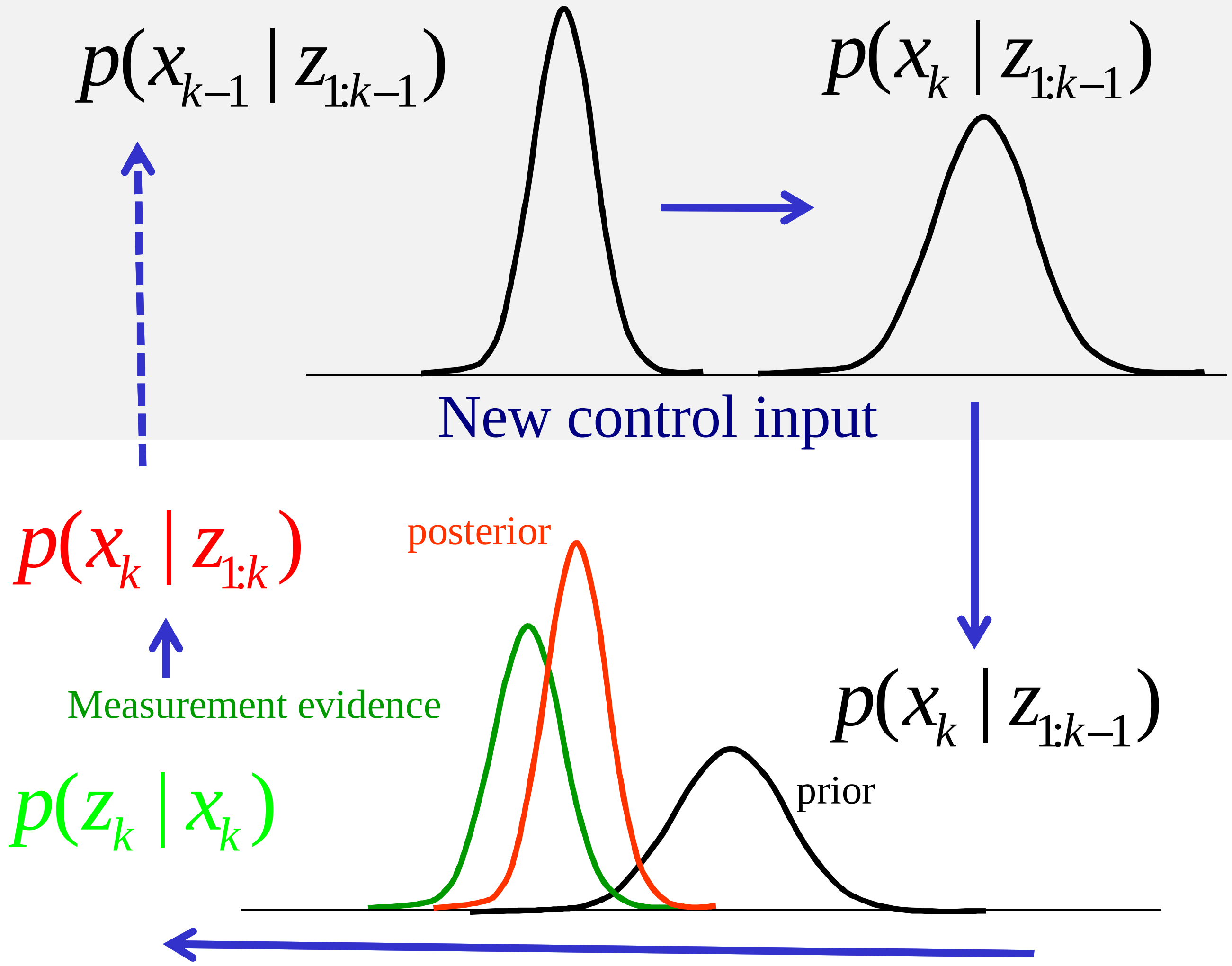
Note: have removed *u* from equations – not a stochastic variable & just complicates math

Given :

1. prior distribution $p(x_0)$
2. State space model
   $$x_k \sim p(x_k \mid x_{k-1})$$
   $$z_k \sim p(z_k \mid x_k)$$
3. Measurement sequence $z_{1:k} = [z_1, ..., z_k]$

Bayesian Recursion

Computation is done through a recursion:



Prediction Step:

$p(x_{k-1} \mid z_{1:k-1})$

$p(x_k \mid z_{1:k-1})$

New control input

Update Step:

$p(x_k \mid z_{1:k})$

posterior

Measurement evidence

$p(z_k \mid x_k)$

$p(x_k \mid z_{1:k-1})$

prior

New measurement

Prediction Step

Assuming we have the posterior distribution of the previous time step: $p(x_{k-1} | z_{1:k-1})$

$$p(x_{k-1} | z_{1:k-1}) \xrightarrow{\qquad} p(x_k | z_{1:k-1})$$

Prediction (to next step)

We can define the following joint distribution: [Product rule]

$$p(x_k, x_{k-1} | z_{1:k-1}) = p(x_k | x_{k-1}, z_{1:k-1}) p(x_{k-1} | z_{1:k-1})$$

$$= p(x_k | x_{k-1}) p(x_{k-1} | z_{1:k-1})$$

[Markov property]

Integrating the above pdf gives the **prediction step**:

$$\boxed{p(x_k | z_{1:k-1}) = \int p(x_k | x_{k-1}) p(x_{k-1} | z_{1:k-1}) dx_{k-1}}$$

[Sum rule]

[also called the Chapman-Kolmogorov equation]

Update Step

Assuming we have a prior from the prediction step:

$$p(x_k \mid z_{1:k-1})$$

And we have the measurement likelihood function:

$$p(z_k \mid x_k)$$

The posterior (or update step) is computed using Bayes' Rule:

$$p(x_k \mid z_{1:k}) = \frac{1}{\eta} p(z_k \mid x_k, z_{1:k-1}) p(x_k \mid z_{1:k-1})$$

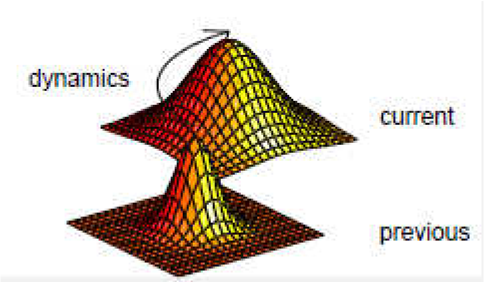
$$\eta = \int p(z_k \mid x_k, z_{1:k-1}) p(x_k \mid z_{1:k-1}) dx_k$$

Using the Markov Assumption, this results in the **update step:**

$$\boxed{p(x_k \mid z_{1:k}) = \frac{1}{\eta} p(z_k \mid x_k) p(x_k \mid z_{1:k-1})}$$

Optimal Bayesian Estimator:

Initialization: $p(x_0)$



Prediction:

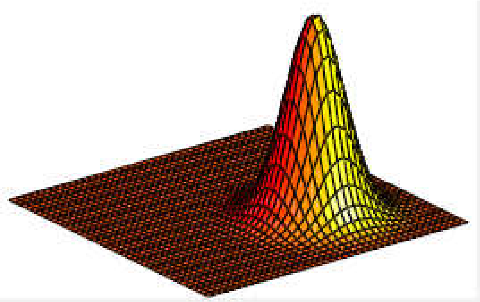
$$p(x_k \mid z_{1:k-1}) = \int p(x_k \mid x_{k-1})\, p(x_{k-1} \mid z_{1:k-1})\, dx_{k-1}$$

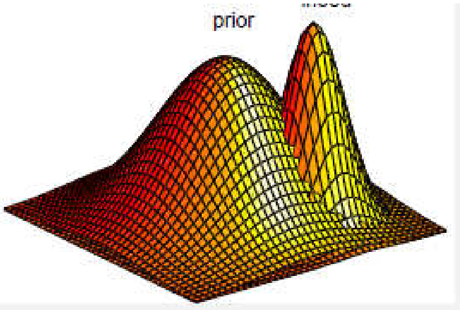
$k = k+1$

z_k

Update:

$$p(x_k \mid z_{1:k}) \propto p(z_k \mid x_k)\, p(x_k \mid z_{1:k-1})$$





Kalman Filter

- We will now show the Kalman Filter as a special case of the Optimal Bayesian Estimator

- First the Prediction step

- Then the Update step

- We will prove this on the board after we see the answer.

Kalman Filter: Prediction Step

Recall propagation equations:

$$x_k = A_k x_{k-1} + B_k u_k + q_k$$

And how this is represented as a pdf:

$$p(x_k \mid x_{k-1}) = N(A_k x_{k-1} + B_k u_k, Q)$$

And if we assume that prior $p(x_{k-1} \mid z_{1:k-1}) = N\left(\mu_{k-1}, \Sigma_{k-1}\right)$

update step:

$$\boxed{p(x_k \mid z_{1:k-1}) = \int p(x_k \mid x_{k-1}) p(x_{k-1} \mid z_{1:k-1}) dx_{k-1}}$$

$$N\left(A_k x_{k-1} + B_k u_k, Q_k\right)$$

Kalman Filter: Prediction Step

prediction step:

$$\boxed{p(x_k \mid z_{1:k-1}) = \int p(x_k \mid x_{k-1})\, p(x_{k-1} \mid z_{1:k-1}) dx_{k-1}}$$

$$N\left(A_k x_{k-1} + B_k u_k, Q_k\right) \qquad N\left(\mu_{k-1}, \Sigma_{k-1}\right)$$

Posterior is just another Gaussian!
(Proof: will show later)

....

$$\boxed{p(x \mid z_{1:k-1}) = N\left(A_k \mu_{k-1} + B_k u_k\, , A_k \Sigma_{k-1} A_k^T + Q_k\right)}$$

Kalman Filter: Update Step

Recall measurement equations:

$$z_k = C_k x_k + r_k \qquad r_k \sim N(0, R)$$

which can be written as the following pdf:

$$p(z_k \mid x_k) = N(C_k x_k, R)$$

update step:

$$\boxed{p(x_k \mid z_{1:k}) = \frac{1}{\eta} p(z_k \mid x_k) p(x_k \mid z_{1:k-1})}$$

$$N\left(A_k \mu_{k-1} + B_k u_k, A_k \Sigma_{k-1} A_k^T + Q_k\right) \triangleq N(\bar{\mu}_k, \bar{\Sigma}_k)$$

Output of prediction step

Kalman Filter: Update Step

update step:

$$\boxed{p(x_k \mid z_{1:k}) = \frac{1}{\eta} p(z_k \mid x_k) p(x_k \mid z_{1:k-1})}$$

$$N(C_k x_k, R) \qquad\qquad N\left(\bar{\mu}_k, \bar{\Sigma}_k\right)$$

The product of two Gaussians -> is another Gaussian

$$p(x_k \mid z_{1:k}) = N(\mu_k, \Sigma_k)$$

(Proof: will show later)

$$\begin{cases} \mu_k = \bar{\mu}_k + K_k (z_k - C_k \bar{\mu}_k) \\ \Sigma_k = (I - K_k C_k) \bar{\Sigma}_k \end{cases}$$

with $K_k = \bar{\Sigma}_k C_k^T (C_k \bar{\Sigma}_k C_k^T + Q_k)^{-1}$

Putting it all together : KF

Prior estimate

$$p(x_{k-1} \mid z_{1:k-1}) = N(\mu_{k-1}, \Sigma_{k-1})$$

Prediction step

$$\boxed{\begin{aligned} \bar{\mu}_k &= A_k \mu_{k-1} + B_k u_k \\ \bar{\Sigma}_k &= A_k \Sigma_{k-1} A_k^T + Q_k \end{aligned}}$$

Predicted estimate

$$p(x_k \mid z_{1:k-1}) = N(\bar{\mu}_k, \bar{\Sigma}_k)$$

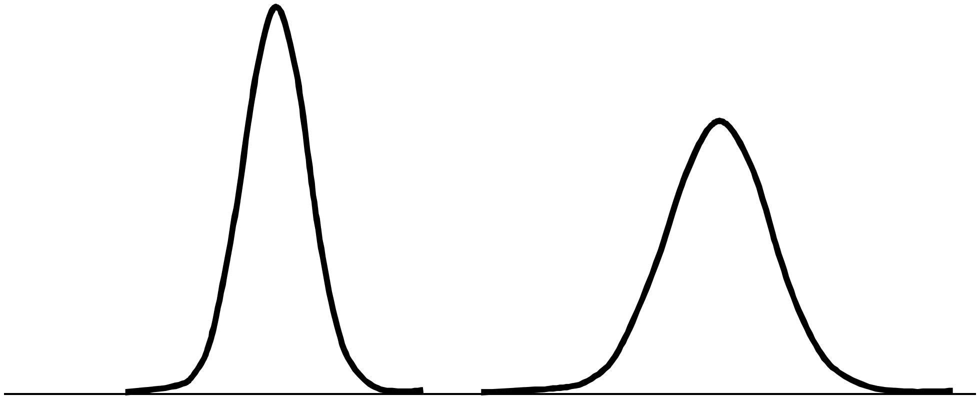
Update Step

$$\boxed{\begin{aligned} K_k &= \bar{\Sigma}_k C_k^T (C_k \bar{\Sigma}_k C_k^T + R_k)^{-1} \\ \mu_k &= \bar{\mu}_k + K_k (z_k - C_k \bar{\mu}_k) \\ \Sigma_k &= (I - K_k C_k) \bar{\Sigma}_k \end{aligned}}$$

Posterior estimate

$$p(x_k \mid z_{1:k}) = N(\mu_k, \Sigma_k)$$

So what do these terms mean?



$$p(x_{k-1} \mid z_{1:k-1}) = N(\mu_{k-1}, \Sigma_{k-1}) \longrightarrow p(x_k \mid z_{1:k-1}) = N(\bar{\mu}_k, \bar{\Sigma}_k)$$

Prior estimate

Prediction step

Predicted estimate

$$\bar{\mu}_k = A_k \mu_{k-1} + B_k u_k$$

$$\bar{\Sigma}_k = A_k \Sigma_{k-1} A_k^T + Q_k$$

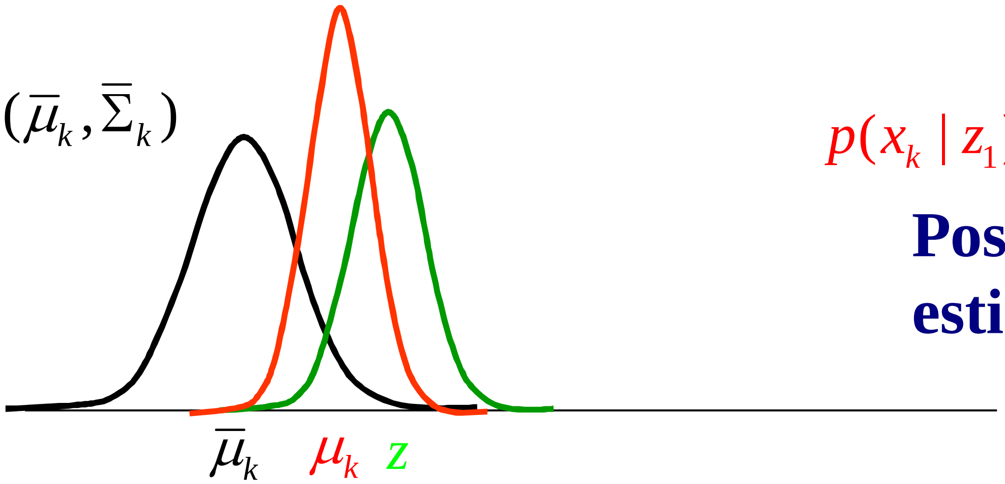
Covariance is skewed by Dynamics & grows with process noise

Mean is shifted by dynamics & control input

So what do these terms mean?

$p(x_k \mid z_{1:k-1}) = N(\bar{\mu}_k, \bar{\Sigma}_k)$

Predicted estimate



$p(x_k \mid z_1) = N(\mu_k, \Sigma_k)$

Posterior estimate

posterior mean

$$\mu_k = \bar{\mu}_k + K_k(\underbrace{z_k - C_k\bar{\mu}_k)}_{\text{“innovation”}}$$

“Kalman Gain”

$$K_k = \bar{\Sigma}_k C_k^T (C_k \bar{\Sigma}_k C_k^T + R_k)^{-1}$$

Is a tradeoff between covariance of prediction and measurement

Posterior covariance

$$\Sigma_k = \bar{\Sigma}_k - K_k C_k \bar{\Sigma}_k$$

Note: shrinking of covariance

Random Walk Example

State equations

$$x_k = x_{k-1} + q_k \qquad q_k \sim N(0,Q)$$

$$z_k = x_k + r_k \qquad r_k \sim N(0,R)$$

Prediction step

$$\left\{\begin{aligned} \bar{\mu}_k &= \mu_{k-1} \\ \bar{\Sigma}_k &= \Sigma_{k-1} + Q_k \end{aligned}\right.$$

Update

$$\left\{\begin{aligned} K_k &= \bar{\Sigma}_k (\bar{\Sigma}_k + R_k)^{-1} \\ \mu_k &= \bar{\mu}_k + K_k (z_k - \bar{\mu}_k) \\ \Sigma_k &= \bar{\Sigma}_k - K_k \bar{\Sigma}_k \end{aligned}\right.$$

The Kalman gain is a tradeoff

If the sensor is really noisy

$$R_k \to \infty, K_k \to 0$$

$$\Sigma_k = \bar{\Sigma}_k$$

$$\mu_k = \bar{\mu}_k$$

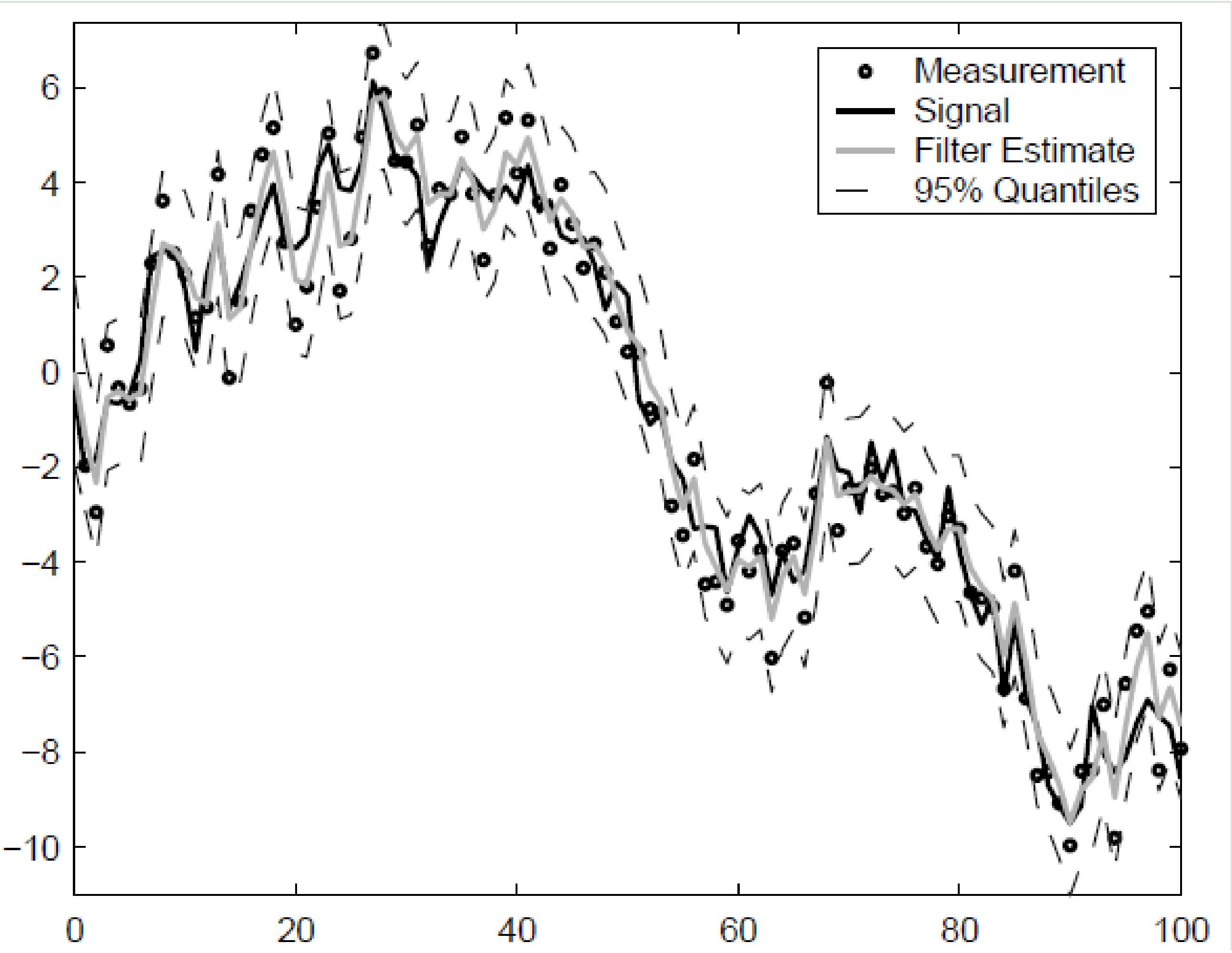
If the sensor is really good

$$R_k \to 0, K_k \to I$$

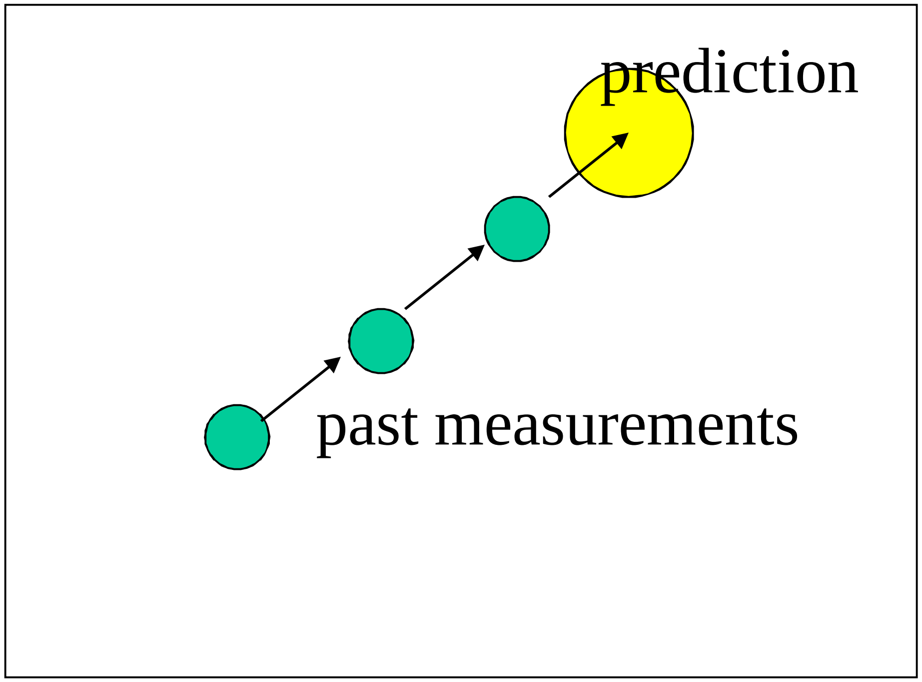
$$\Sigma_k = 0$$

$$\mu_k = z_k$$

Random Walk Example



Velocity estimation (from position)



Kalman Filter for 2D Tracking Example

State Propagation

$$\begin{pmatrix} x_{k+1} \\ y_{k+1} \\ \dot{x}_{k+1} \\ \dot{y}_{k+1} \end{pmatrix} = \begin{pmatrix} 1 & 0 & \Delta t & 0 \\ 0 & 1 & 0 & \Delta t \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_k \\ y_k \\ \dot{x}_k \\ \dot{y}_k \end{pmatrix} + N(0, Q)$$

$$Q = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & q^2 & 0 \\ 0 & 0 & 0 & q^2 \end{pmatrix}$$

Random walk
in velocity

Measurement

$$\begin{pmatrix} x_{obs} \\ y_{obs} \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix} \begin{pmatrix} x_k \\ y_k \\ \dot{x}_k \\ \dot{y}_k \end{pmatrix} + N(0, R)$$

$$R = \begin{pmatrix} r^2 & 0 \\ 0 & r^2 \end{pmatrix}$$

Only observe location

KF Updates

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

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

$$B = (\ \ )$$

$$x_k = A_k x_{k-1} + B_k u_k + q_k$$

$$z_k = C_k x_k + r_k$$

$$q_k \sim N(0, Q)$$

$$r_k \sim N(0, R)$$

$$\bar{\mu}_k = A\mu_{k-1}$$

$$\bar{\Sigma}_k = A\Sigma_{k-1}A^T + Q$$

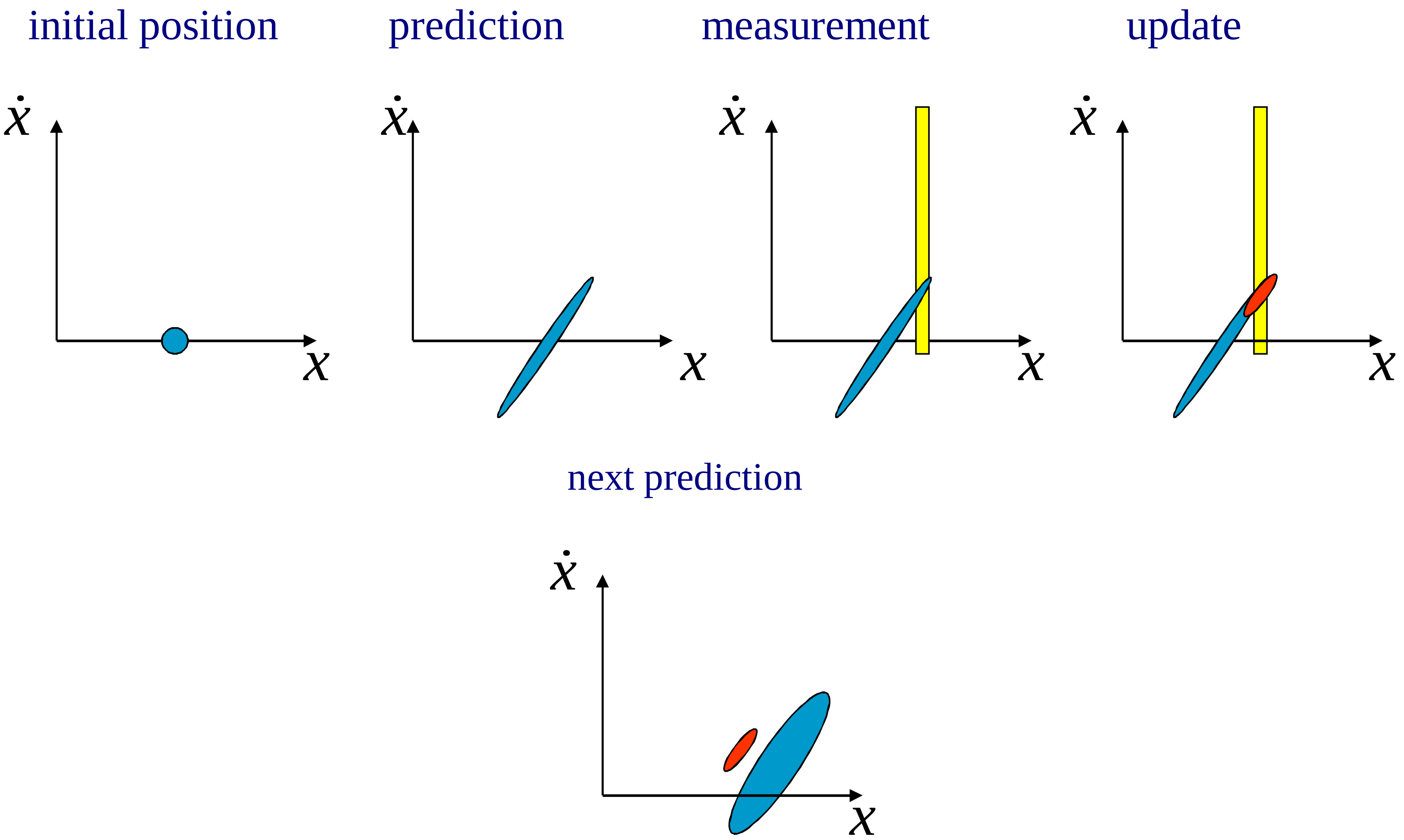
Trades off uncertainty in position (x,y) with R

$$K_k = \bar{\Sigma}_k C^T (C\ \bar{\Sigma}_k C^T + R\ )^{-1}$$

$$\mu_k = \bar{\mu}_k + K_k (z_k - C\ \bar{\mu}_k)$$

$$\Sigma_k = (I - K_k C\ )\bar{\Sigma}_k$$

Cartoon of $x, \dot{x}$ state



Observablility

- So even though velocity was not measured- it is observable:

$$O = \begin{pmatrix} C \\ CA \\ CA^2 \\ \vdots \\ CA^{(n-1)} \end{pmatrix}$$

Theorem: (Observability Test): the linear time invariant system: $x_k = Ax_{k-1} + Bu_k + q_k$

$$z_k = Cx_k + r_k$$

is observable if and only if the observability matrix O is has rank n

From the last example: $C = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}$

Which means O is rank n

$$CA = \begin{pmatrix} 1 & 0 & \Delta t & 0 \\ 0 & 1 & 0 & \Delta t \end{pmatrix}$$

Proof of KF Filter

- The Proof is shown in “Probabilistic Robotics” by S. Thrun pg. 45-54.
- Note that Q and R are reversed in the book

- See “Principles of Robot Motion: Theory, Algorithms, and Implementations” by Howie Choset for a geometric derivation SS 8.2. (*will post this page*)

- See “Lecture notes on Mobile Robotics” Paul Newman. (*will post this as well*)

- We will now derive the KF using a property of Gaussians.

Kalman Filter: Prove the following

1. prediction step:

$$\boxed{p(x_k \mid z_{1:k-1}) = \int \underbrace{p(x_k \mid x_{k-1})}_{N\left(A_k x_{k-1} + B_k u_k , Q_k\right)} \underbrace{p(x_{k-1} \mid z_{1:k-1})}_{N\left(\mu_{k-1}, \Sigma_{k-1}\right)} dx_{k-1}}$$

$$\boxed{p(x \mid z_{1:k-1}) = N\left(A_k \mu_{k-1} + B_k u_k \,, A_k \Sigma_{k-1} A_k^T + Q_k\right)}$$

2. update step:

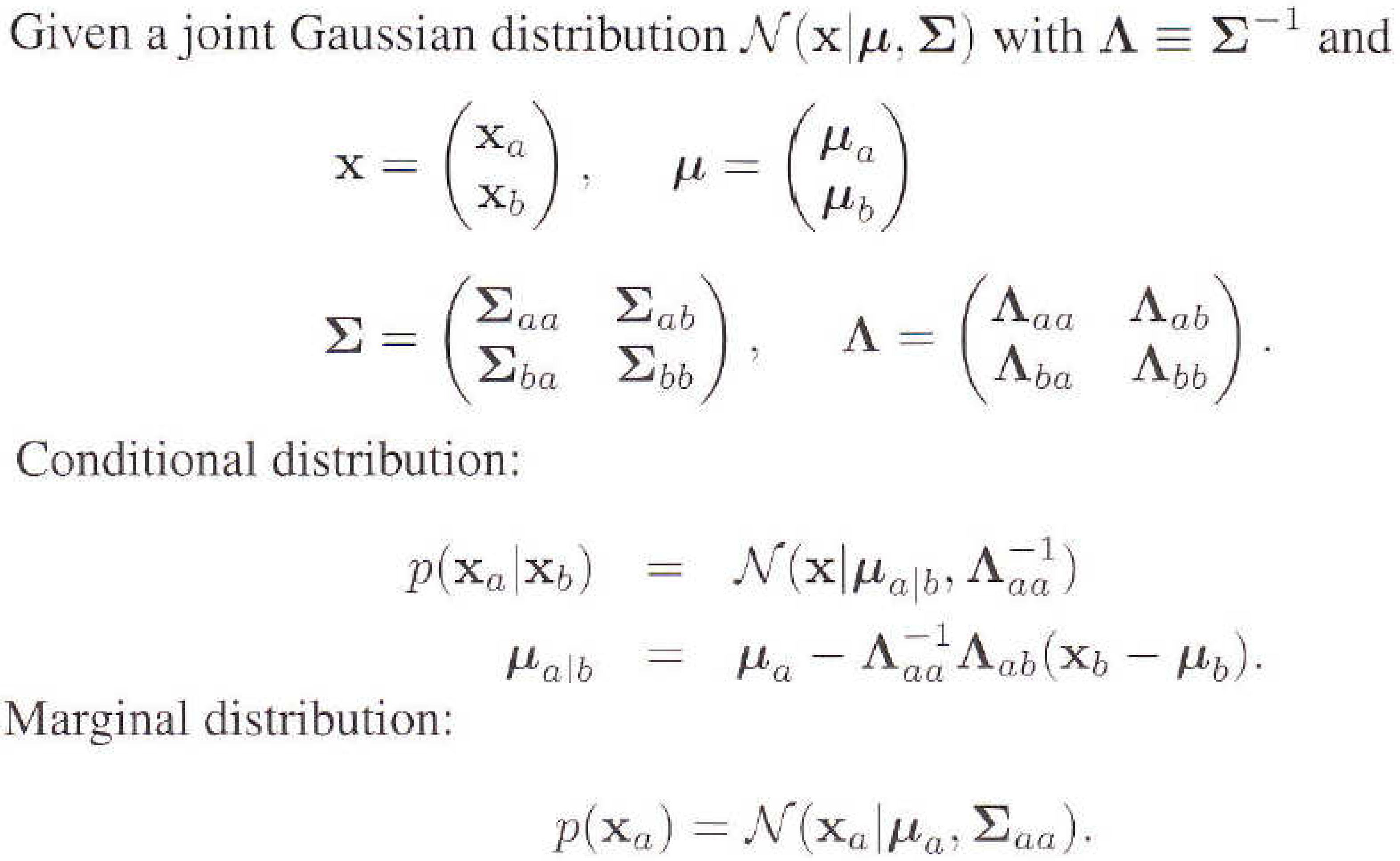
$$\boxed{p(x_k \mid z_{1:k}) = \frac{1}{\eta} \underbrace{p(z_k \mid x_k)}_{p(z_k \mid x_k) = N(C_k x_k, R_k)} \underbrace{p(x_k \mid z_{1:k-1})}_{N(\bar{\mu}_k, \bar{\Sigma}_k)}}$$

$$p(x_k \mid z_{1:k}) = N(\mu_k, \Sigma_k)$$

$$\begin{cases} \mu_t = \bar{\mu}_t + K_t(z_t - C_t \bar{\mu}_t) \\ \Sigma_t = (I - K_t C_t)\bar{\Sigma}_t \end{cases} \quad \text{with} \quad K_t = \bar{\Sigma}_t C_t^T (C_t \bar{\Sigma}_t C_t^T + Q_t)^{-1}$$

Gaussian Marginal and Conditional Densities

See “Pattern Recognition and Machine Learning” C. Bishop SS 2.3 for a review of Gaussians.



Given a joint Gaussian distribution $\mathcal{N}(\mathbf{x}|\boldsymbol{\mu}, \boldsymbol{\Sigma})$ with $\boldsymbol{\Lambda} \equiv \boldsymbol{\Sigma}^{-1}$ and

$$\mathbf{x} = \begin{pmatrix} \mathbf{x}_a \\ \mathbf{x}_b \end{pmatrix}, \qquad \boldsymbol{\mu} = \begin{pmatrix} \boldsymbol{\mu}_a \\ \boldsymbol{\mu}_b \end{pmatrix}$$

$$\boldsymbol{\Sigma} = \begin{pmatrix} \boldsymbol{\Sigma}_{aa} & \boldsymbol{\Sigma}_{ab} \\ \boldsymbol{\Sigma}_{ba} & \boldsymbol{\Sigma}_{bb} \end{pmatrix}, \qquad \boldsymbol{\Lambda} = \begin{pmatrix} \boldsymbol{\Lambda}_{aa} & \boldsymbol{\Lambda}_{ab} \\ \boldsymbol{\Lambda}_{ba} & \boldsymbol{\Lambda}_{bb} \end{pmatrix}.$$

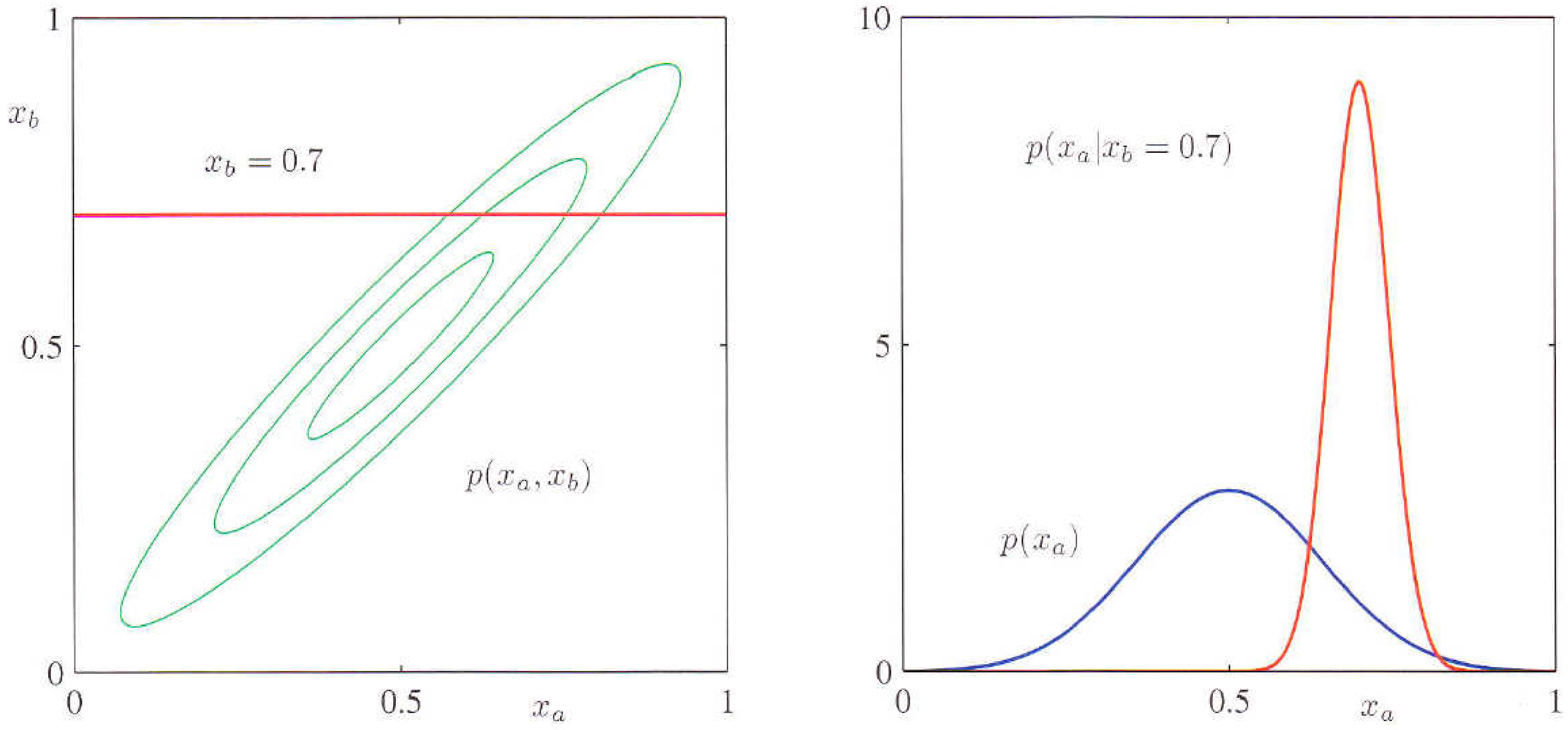
Conditional distribution:

$$\begin{aligned} p(\mathbf{x}_a|\mathbf{x}_b) &= \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_{a|b}, \boldsymbol{\Lambda}_{aa}^{-1}) \\ \boldsymbol{\mu}_{a|b} &= \boldsymbol{\mu}_a - \boldsymbol{\Lambda}_{aa}^{-1}\boldsymbol{\Lambda}_{ab}(\mathbf{x}_b - \boldsymbol{\mu}_b). \end{aligned}$$

Marginal distribution:

$$p(\mathbf{x}_a) = \mathcal{N}(\mathbf{x}_a|\boldsymbol{\mu}_a, \boldsymbol{\Sigma}_{aa}).$$

Intuition about Marginal and Conditional



Marginal $p(x_a) = \int p(x_a, x_b) dx_b$

Matrix Inversion Lemma & Matrix Derivatives

- Inversion Lemma

$$(\mathbf{A}+\mathbf{UBV})^{-1} = \mathbf{A}^{-1}-\mathbf{A}^{-1}\mathbf{U}(\mathbf{B}^{-1}+\mathbf{VA}^{-1}\mathbf{U})^{-1}\mathbf{VA}^{-1}$$

$$\begin{bmatrix} A & U \\ V & C \end{bmatrix}^{-1} = \begin{bmatrix} (A-UC^{-1}V)^{-1} & -(A-UC^{-1}V)^{-1}UC^{-1} \\ -C^{-1}V(A-UC^{-1}V)^{-1} & C^{-1}V(A-UC^{-1}V)^{-1}UC^{-1}+C^{-1} \end{bmatrix}$$

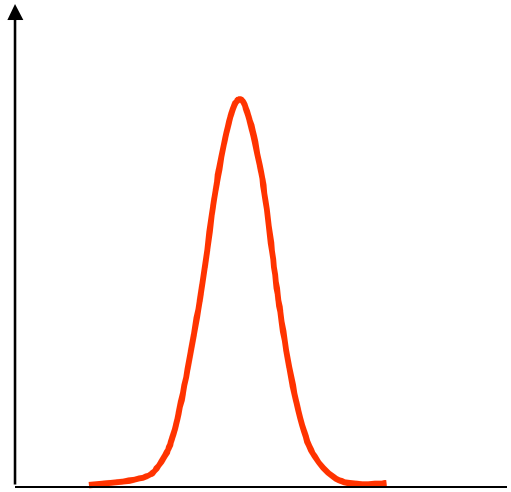
- Matrix Derivatives (see “matrix cookbook” for reference)

$$\frac{\partial \mathbf{x}^T\mathbf{a}}{\partial \mathbf{x}} = \frac{\partial \mathbf{a}^T\mathbf{x}}{\partial \mathbf{x}} = \mathbf{a} \qquad \frac{\partial \mathbf{x}^T\mathbf{B}\mathbf{x}}{\partial \mathbf{x}} = (\mathbf{B}+\mathbf{B}^T)\mathbf{x}$$

Assume $\mathbf{W}$ is symmetric, then

$$\frac{\partial}{\partial \mathbf{s}}(\mathbf{x}-\mathbf{As})^T\mathbf{W}(\mathbf{x}-\mathbf{As}) = -2\mathbf{A}^T\mathbf{W}(\mathbf{x}-\mathbf{As})$$

Review: Gaussians



$$p(x) \sim N(\mu,\Sigma)$$

$$p(x) \; \frac{1}{\det(2\pi\Sigma)^{-1/2}} \; \exp\left\{-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)\right\}$$

Next Time

- What happens when the Linear State Space model breaks down?

- How can we do this for nonlinear models?