Koller & Friedman: Chapter 16
Jordan: Chapters 13, 15
Uri Lerner's Thesis: Chapters 3,9

Dynamic models 1
Kalman filters, linearization,
Switching KFs, Assumed density filters

Probabilistic Graphical Models – 10708

Carlos Guestrin
Carnegie Mellon University

November 16th, 2005

Announcement

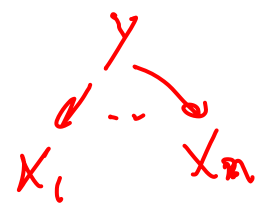
- Special recitation lectures
  - Pradeep will give two special lectures
  - Nov. 22 & Dec. 1: 5-6pm, during recitation
  - Covering: variational methods, loopy BP and their relationship
  - Don’t miss them!!!

Adventures of our BN hero

- Compact representation for probability distributions
- Fast inference
- Fast learning
- Approximate inference

- But… Who are the most popular kids?

1. Naïve Bayes



2 and 3.
Hidden Markov models (HMMs)
Kalman Filters

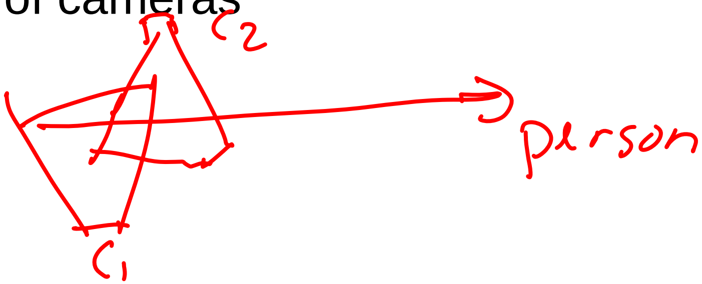
The Kalman Filter

- An HMM with Gaussian distributions
- Has been around for at least 50 years
- Possibly the most used graphical model ever
- It’s what
  - does your cruise control
  - tracks missiles
  - controls robots
  - …
- And it’s so simple…
  - Possibly explaining why it’s so used
- Many interesting models build on it…
  - Review and extensions today

Example of KF – SLAT
Simultaneous Localization and Tracking

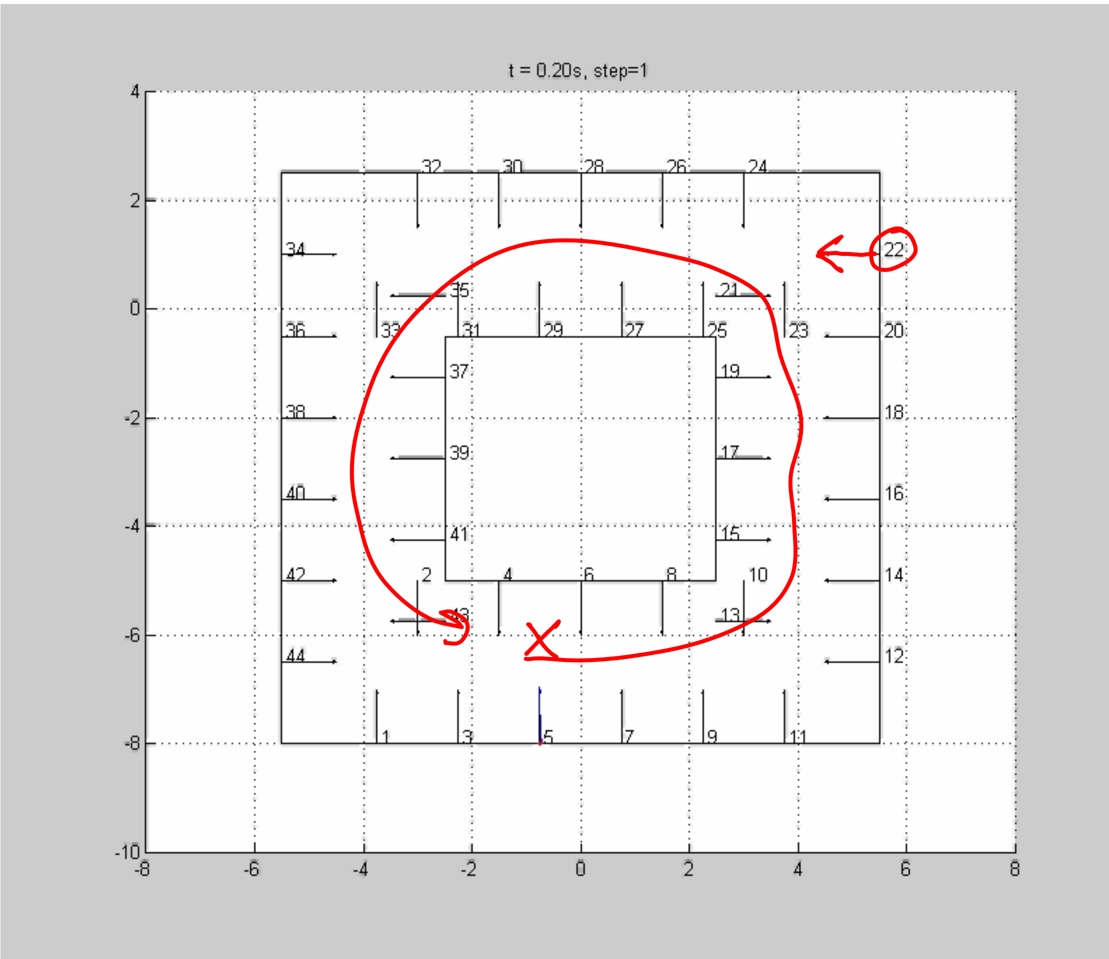
[Funiak, Guestrin, Paskin, Sukthankar ’05]

- Place some cameras around an environment, don’t know where they are
- Could measure all locations, but requires lots of grad. student (Stano) time
- Intuition:
  - A person walks around
  - If camera 1 sees person, then camera 2 sees person, learn about relative positions of cameras



Example of KF – SLAT
Simultaneous Localization and Tracking

[Funiak, Guestrin, Paskin, Sukthankar ’05]



C_i ← each camera i

L_t ← position at time t

$p(C, L_t)$

$p(C)$ ← "uniform"

$p(L_0)$

$p(O_i \mid C_j, L_i)$

$p(C, L_t \mid O_{1:t})$

Multivariate Gaussian

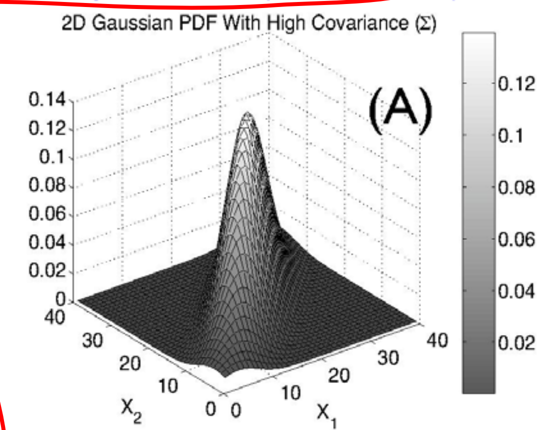
$$p(X_1,\ldots,X_n) = \frac{1}{(2\pi)^{n/2}|\Sigma|^{1/2}} \exp\left\{-\frac{1}{2}(\mathbf{x}-\mu)^T\Sigma^{-1}(\mathbf{x}-\mu)\right\}$$

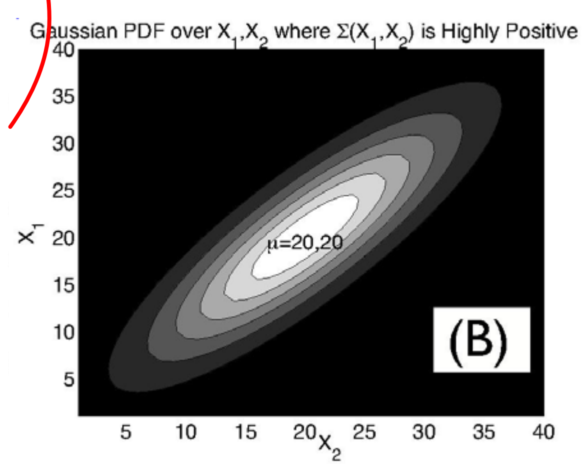
Mean vector:

$$\mu = \begin{pmatrix} \mu_1 \\ \vdots \\ \mu_d \end{pmatrix}$$

Covariance matrix:

$$\Sigma = \begin{pmatrix} \sigma_1^2 & \sigma_{12} & \cdots \\ \sigma_{12} & \sigma_2^2 & \\ \vdots & & \ddots \end{pmatrix}$$



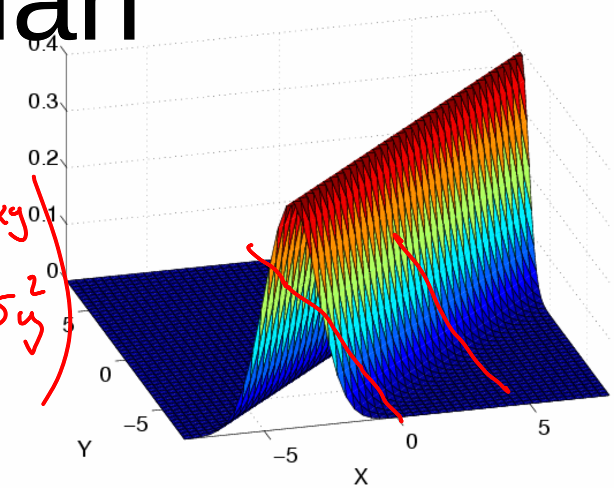


Conditioning a Gaussian

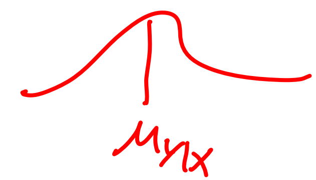
- Joint Gaussian:
  - p(X,Y) ~ $N(\mu;\Sigma)$

$$\mu = \begin{pmatrix} \mu_X \\ \mu_Y \end{pmatrix} \qquad \Sigma = \begin{pmatrix} \sigma_X^2 & \sigma_{XY} \\ \sigma_{XY} & \sigma_Y^2 \end{pmatrix}$$

- Conditional linear Gaussian:
  - p(Y|X) ~ $N(\mu_{Y|X}; \sigma^2)$ $= \frac{p(Y,X)}{p(x)}$



$$\mu_{Y|X} = \mu_Y + \frac{\sigma_{YX}}{\sigma_X^2}(x - \mu_x)$$



$$\sigma_{Y|X}^2 = \sigma_Y^2 - \frac{\sigma_{YX}^2}{\sigma_X^2}$$

doesn't depend on x

Gaussian is a "Linear Model"

- Conditional linear Gaussian:
  - p(Y|X) ~ $N(\beta_0+\beta X; \sigma^2_{Y|X})$

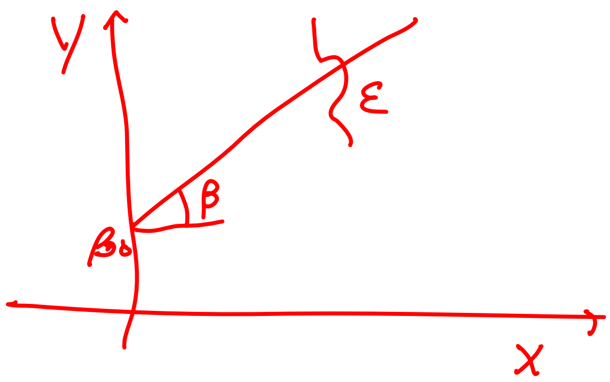
$$\mu_{Y|X} = \mu_Y + \frac{\sigma_{YX}}{\sigma_X^2}(x - \mu_x)$$

$$\sigma^2_{Y|X} = \sigma^2_Y - \frac{\sigma^2_{YX}}{\sigma^2_X}$$

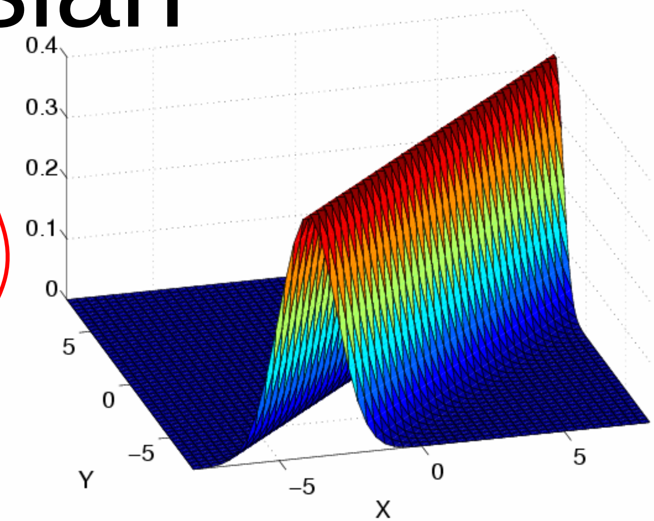
$$Y = \beta_0 + \beta X + \underbrace{\varepsilon}_{N(0, \sigma^2_{Y|X})}$$

$$\beta_0 = \mu_Y - \frac{\sigma_{YX}}{\sigma_X^2}\mu_X$$

$$\beta = \frac{\sigma_{YX}}{\sigma_X^2}$$



Conditioning a Gaussian



- Joint Gaussian:
  - p(X,Y) ~ $N(\mu;\Sigma)$

$$\Sigma = \begin{pmatrix} \Sigma_{XX} & \Sigma_{XY} \\ \Sigma_{YX} & \Sigma_{YY} \end{pmatrix}$$

- Conditional linear Gaussian:
  - p(Y|X) ~ $N(\mu_{Y|X};\ \Sigma_{YY|X})$

$$\mu_{Y|X} = \mu_Y + \Sigma_{YX}\Sigma_{XX}^{-1}(x - \mu_x)$$

$$\underbrace{\Sigma_{YY|X}}_{\text{doesn't depend on } x} = \Sigma_{YY} - \Sigma_{YX}\Sigma_{XX}^{-1}\Sigma_{XY}$$

Conditional Linear Gaussian (CLG) – general case

- Conditional linear Gaussian:
  - p(Y|X) ~ $N(\beta_0+BX; \Sigma_{YY|X})$

$$\mu_{Y|X} = \mu_Y + \Sigma_{YX}\Sigma_{XX}^{-1}(x-\mu_x)$$
$$\Sigma_{YY|X} = \Sigma_{YY} - \Sigma_{YX}\Sigma_{XX}^{-1}\Sigma_{XY}$$

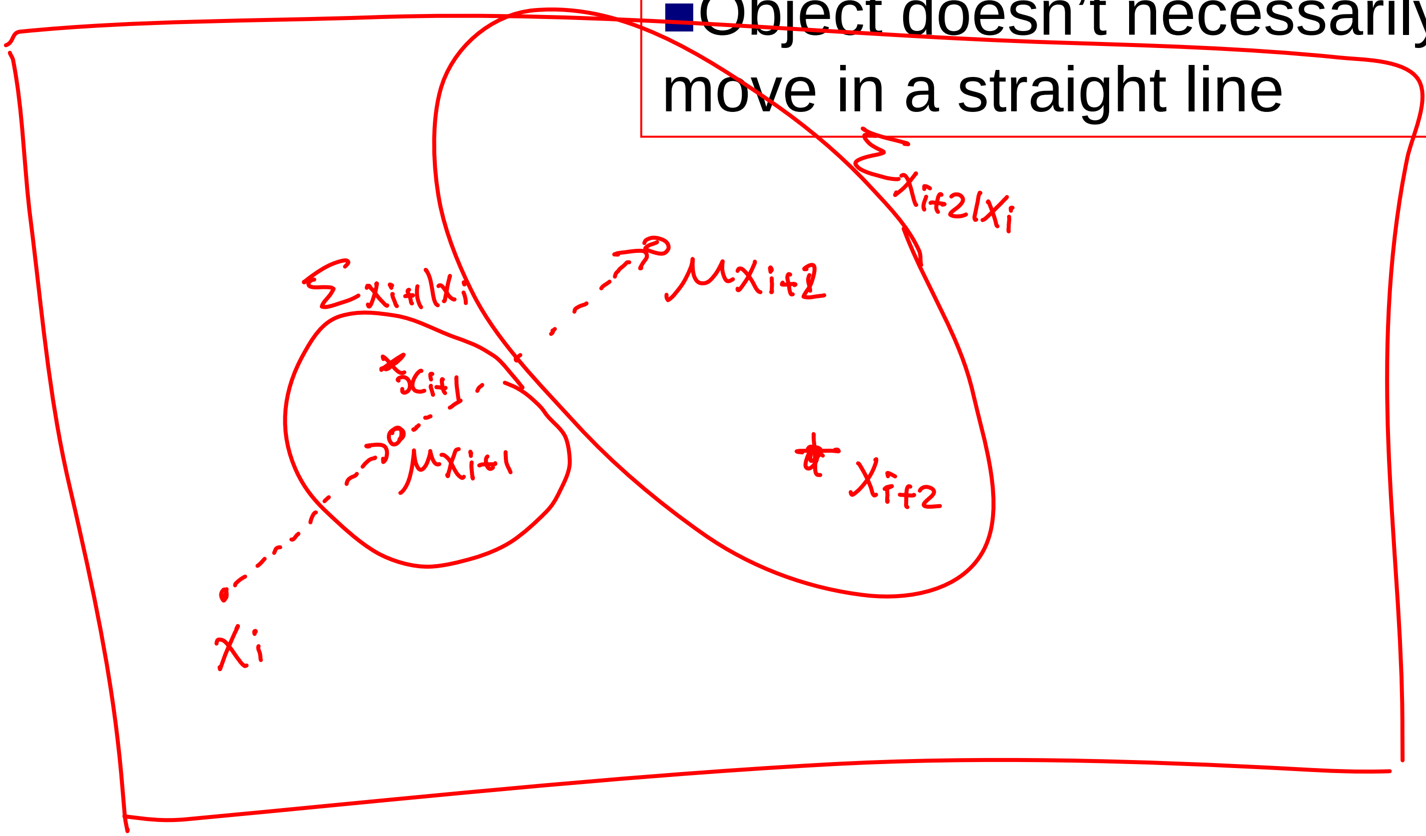
$\beta_0 = \mu_Y + \Sigma_{YX}\Sigma_{XX}^{-1}\mu_X$

$B = \Sigma_{YX}\Sigma_{XX}^{-1}$

Understanding a linear Gaussian – the 2d case

$P(X_{i+1} | X_i) = CLG$

- Variance increases over time (motion noise adds up)
- Object doesn't necessarily move in a straight line



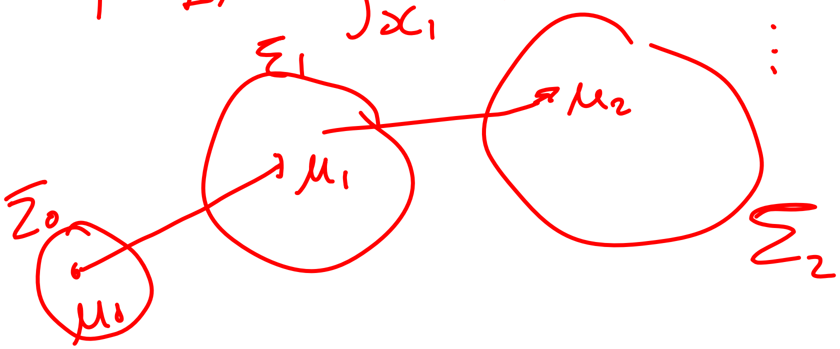
Tracking with a Gaussian 1

- $p(X_0) \sim N(\mu_0, \Sigma_0)$
- $p(X_{i+1}|X_i) \sim N(B X_i + \beta; \Sigma_{Xi+1|Xi})$

$$p(X_1) = \int_{x_0} p(x_0) \cdot p(X_1|x_0)\, dx_0$$

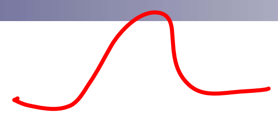
$$p(X_2) = \int_{x_1} p(x_1)\, p(X_2|x_1)\, dx_1$$

$\vdots$



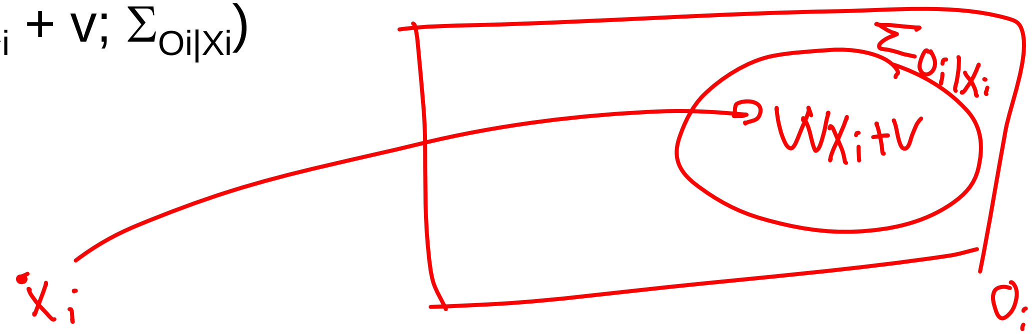
Tracking with Gaussians 2 – Making observations

- We have $p(X_i)$

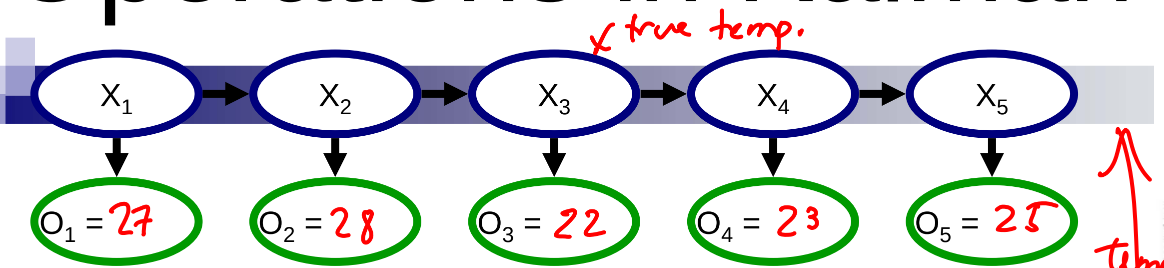


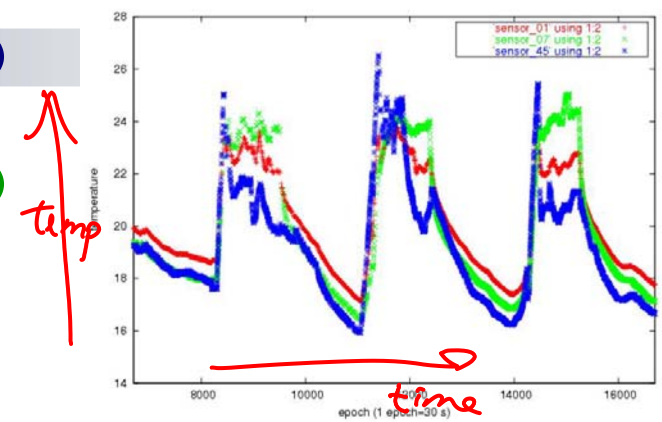
- Detector observes $O_i = o_i$
- Want to compute $p(X_i | O_i = o_i)$
- Use Bayes rule: $p(X_i | o_i) = \frac{p(X_i) \cdot p(o_i | X_i)}{p(o_i)}$

- Require a CLG observation model
  - $p(O_i | X_i) \sim N(W X_i + v; \Sigma_{Oi|Xi})$



Operations in Kalman filter





- Compute $p(X_t \mid O_{1:t} = o_{1:t})$
- Start with $p(X_0)$
- At each time step *t*:
  - **Condition** on observation
    $$p(X_t \mid o_{1:t}) \propto p(X_t \mid o_{1:t-1})p(o_t \mid X_t)$$
  - **Prediction** (Multiply transition model)
    $$p(X_{t+1}, X_t \mid o_{1:t}) = p(X_{t+1} \mid X_t)p(X_t \mid o_{1:t})$$
  - **Roll-up** (marginalize previous time step)
    $$p(X_{t+1} \mid o_{1:t}) = \int_{x_t} p(X_{t+1}, x_t \mid o_{1:t})dx_t$$
- I'll describe one implementation of KF, there are others
  - Information filter

Canonical form

$$
\begin{aligned}
p(X_1,\ldots,X_n) &= \frac{1}{(2\pi)^{n/2}|\Sigma|^{1/2}} \exp\left\{-\frac{1}{2}(\mathbf{x}-\mu)^T\Sigma^{-1}(\mathbf{x}-\mu)\right\} \\
&= K \exp\left\{\eta^T\mathbf{x} - \frac{1}{2}\mathbf{x}^T\Lambda^{-1}\mathbf{x}\right\}
\end{aligned}
$$

- Standard form and canonical forms are related:

$$
\begin{aligned}
\mu &= \Lambda^{-1}\underline{\eta} \\
\Sigma &= \underline{\Lambda}^{-1}
\end{aligned}
$$

- Conditioning is easy in canonical form
- Marginalization easy in standard form

Conditioning in canonical form

$$p(X_t \mid o_{1:t}) \propto p(X_t \mid o_{1:t-1})\,p(o_t \mid X_t)$$

- First multiply: $p(A, B) = \underbrace{p(A)}_{\Lambda_A}\underbrace{p(B \mid A)}_{\Lambda_{B|A}}$

$p(A)$: η_1, ${}^{K}\Lambda_1^{K}$

$p(B \mid A)$: η_2, ${}^{2K}\Lambda_2^{2K}$

$$\begin{pmatrix} \Lambda_1 & 0 \\ 0 & 0 \end{pmatrix} \quad \begin{pmatrix} \Lambda_2 \end{pmatrix}$$

$p(A, B)$: $\eta_3 = \eta_1 + \eta_2$, $\Lambda_3 = \Lambda_1 + \Lambda_2$

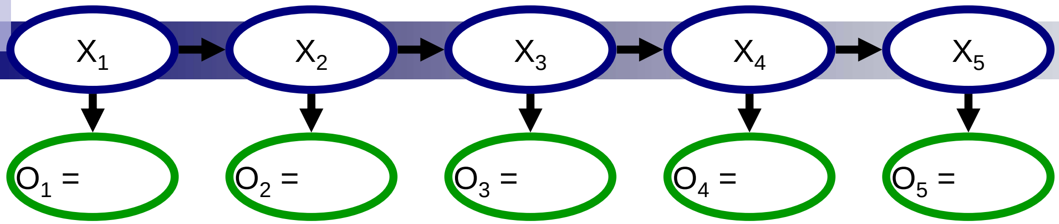
- Then, condition on value B = y $\quad p(A \mid B = y)$

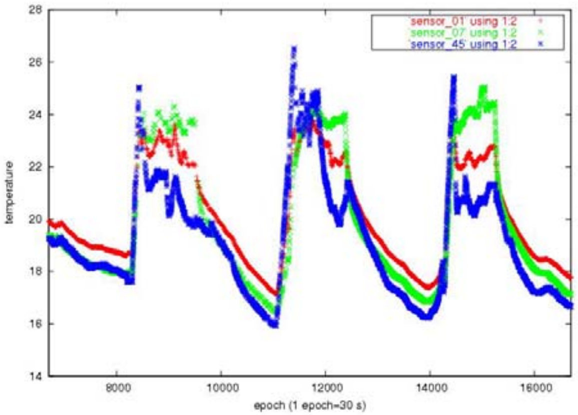
$$\eta_{A|B=y} = \eta_A - \Lambda_{AB}.y$$

$$\Lambda_{AA|B=y} = \Lambda_{AA}$$

$$\Lambda_3 = \begin{pmatrix} \Lambda_{AA} & \Lambda_{AB} \\ \Lambda_{AA} & \Lambda_{BB} \end{pmatrix}$$

Operations in Kalman filter





- Compute $p(X_t \mid O_{1:t} = o_{1:t})$
- Start with $p(X_0)$
- At each time step *t*:
  - **Condition** on observation
    $$p(X_t \mid o_{1:t}) \propto p(X_t \mid o_{1:t-1}) p(o_t \mid X_t)$$
    matrix ops
  - **Prediction** (Multiply transition model)
    $$p(X_{t+1}, X_t \mid o_{1:t}) = p(X_{t+1} \mid X_t) p(X_t \mid o_{1:t})$$
  - **Roll-up** (marginalize previous time step)
    $$p(X_{t+1} \mid o_{1:t}) = \int_{X_t} p(X_{t+1}, x_t \mid o_{1:t}) dx_t$$

Prediction & roll-up in canonical form

$$p(X_{t+1} \mid o_{1:t}) = \int_{X_t} p(X_{t+1} \mid x_t) p(x_t \mid o_{1:t}) dx_t$$

- First multiply: $p(A, B) = p(A)p(B \mid A)$

add matrices

- Then, marginalize X_t: $p(A) = \int_B p(A, b) db$

$$\Lambda_3 = \begin{pmatrix} \Lambda_{AA} & \Lambda_{AB} \\ \Lambda_{BA} & \Lambda_{BB} \end{pmatrix}$$

$$\begin{aligned} \eta_A^m &= \eta_A - \Lambda_{AB} \Lambda_{BB}^{-1} \eta_B \\ \Lambda_{AA}^m &= \Lambda_{AA} - \Lambda_{AB} \Lambda_{BB}^{-1} \Lambda_{BA} \end{aligned}$$

What if observations are not CLG?

- Often observations are not CLG
  - CLG if $O_i = B X_i + \beta_o + \varepsilon$ $\leftarrow N(0, \Sigma_\varepsilon)$
- Consider a motion detector
  - $O_i = 1$ if person is likely to be in the region

$$p(O_i = 1 \mid X_i) = \begin{cases} 0; & \text{if outside the box} \\ 1; & \text{if inside the box} \end{cases}$$

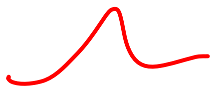
  - Posterior is not Gaussian

$p(X_i \mid O_i = 1)$

$p(X_i)$:

Linearization: incorporating non-linear evidence

- $p(O_i|X_i)$ not CLG, but…
- Find a Gaussian approximation of $p(X_i,O_i)= p(X_i)\ p(O_i|X_i)$
- Instantiate evidence $O_i=o_i$ and obtain a Gaussian for $p(X_i|O_i=o_i)$



- Why do we hope this would be any good?
  - Locally, Gaussian may be OK

Linearization as integration

$$\hat{\Sigma} = \begin{pmatrix} \hat{\sigma}_{o_i} & \hat{\sigma}_{o_i x_i} \\ \hat{\sigma}_{x_i o_i} & \sigma_{x_i}^2 \end{pmatrix}$$

$$\hat{\mu} = \begin{pmatrix} \hat{\mu}_{o_i} \\ \mu_{x_i} \end{pmatrix}$$

- Gaussian approximation of $p(X_i,O_i)= p(X_i)\ p(O_i|X_i)$
- Need to compute moments
  - $E[O_i]$ $= \int_{x_i,o_i} o_i \cdot p(x_i) \cdot p(o_i|x_i)\ do_i\, dx_i = \hat{\mu}_{o_i}$
  - $E[O_i^2]$ $= \int_{x_i o_i} o_i^2 \ \ldots \Rightarrow \hat{\sigma}_{o_i}^2 = \ldots - \hat{\mu}_{o_i}^2$
  - $E[O_i\ X_i]$ $= \int_{x_i o_i} o_i \cdot x_i \ \ldots \Rightarrow \hat{\sigma}_{o_i x_i} = \ldots - \hat{\mu}_{o_i} \mu_{x_i}$
- Note: Integral is product of a Gaussian with an arbitrary function

Linearization as numerical integration

- **Product of a Gaussian with arbitrary function**

gaussian

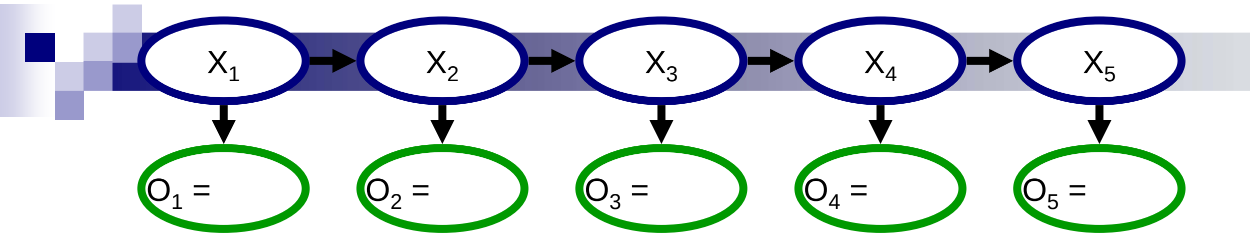
$$\int_x w(x)\, f(x)\, dx$$

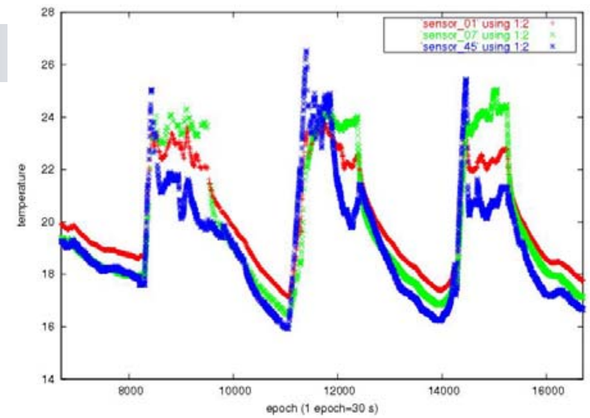
- Effective numerical integration with **Gaussian quadrature** method
  - Approximate integral as **weighted sum over integration points** $\to \{w_j, x_j\}$
  - Gaussian quadrature defines location of points and weights

$$\int_x w(x) f(x)\, dx \approx \sum_{j=1}^{N} w_j\, f(x_j)$$

- Exact if arbitrary function is **polynomial of bounded degree**
- **Number of integration points exponential** in number of dimensions *d*
- **Exact monomials** requires exponentially fewer points
  - For **2*d*+1 points**, this method is equivalent to effective **Unscented Kalman filter**
  - **Generalizes to many more points**

Operations in non-linear Kalman filter





- Compute $p(X_t \mid O_{1:t} = o_{1:t})$
- Start with $p(X_0)$
- At each time step *t*:
  - **Condition** on observation (use **numerical integration**)
    $$p(X_t \mid o_{1:t}) \propto p(X_t \mid o_{1:t-1}) p(o_t \mid X_t)$$
  - **Prediction** (Multiply transition model, use **numerical integration**)
    $$p(X_{t+1}, X_t \mid o_{1:t}) = p(X_{t+1} \mid X_t) p(X_t \mid o_{1:t})$$
  - **Roll-up** (marginalize previous time step)
    $$p(X_{t+1} \mid o_{1:t}) = \int_{X_t} p(X_{t+1}, x_t \mid o_{1:t}) dx_t$$

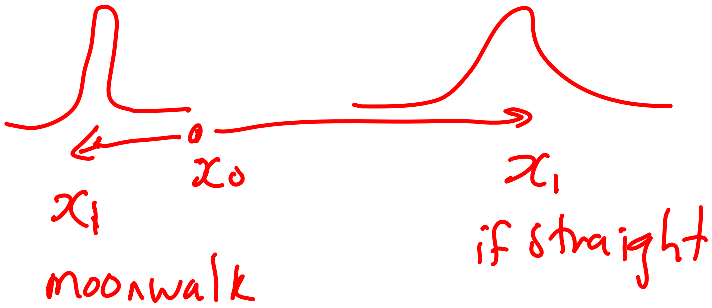
What if the person chooses different motion models?

- With probability θ, move more or less straight
- With probability $1-\theta$, do the “moonwalk”

The moonwalk

What if the person chooses different motion models?

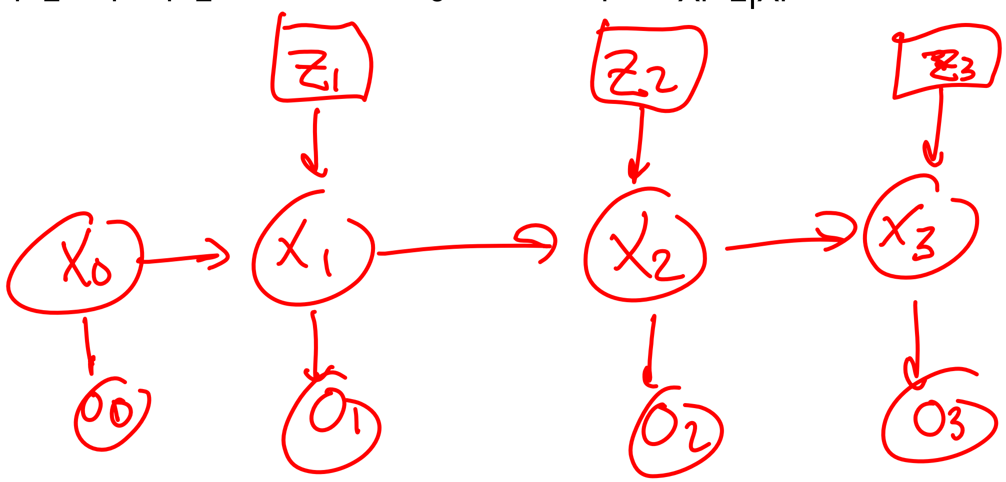
- With probability θ, move more or less straight
- With probability $1-\theta$, do the "moonwalk"



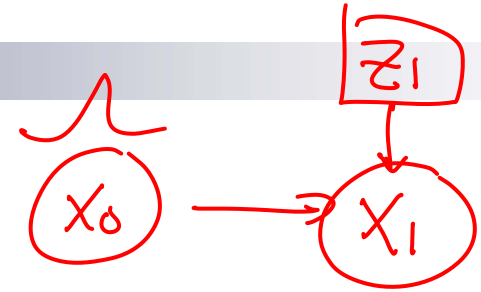
Switching Kalman filter

- At each time step, choose one of *k* motion models:
  - You never know which one!
- $p(X_{i+1}|X_i,Z_{i+1})$
  - CLG indexed by Z_i
  - $p(X_{i+1}|X_i,Z_{i+1}=j) \sim N(\beta^j_0 + B^j X_i; \Sigma^j_{X_{i+1}|X_i})$

depending on motion model



Inference in switching KF – one step



- Suppose
  - $p(X_0)$ is Gaussian
  - Z_1 takes one of two values
  - $p(X_1|X_o,Z_1)$ is CLG
- Marginalize X_0

$$p(x_1|z_1) = \int_{x_0} p(x_0) \cdot p(x_1|x_0,z_1)\, dx_0$$

$z_1 \rightarrow$ straight

$z_1 \rightarrow$ moonwalk

- Marginalize Z_1

$$p(x_1) = \sum_j p(x_1|z_1=j) \cdot P(z_1=j)$$

- Obtain mixture of two Gaussians!

Multi-step inference

Z_{i+1}

$X_i \to X_{i+1}$

- Suppose
  - $p(X_i)$ is a mixture of m Gaussians
  - Z_{i+1} takes one of two values
  - $p(X_{i+1}|X_i,Z_{i+1})$ is CLG

$$p(x_i) = \sum_{k=1}^{m} w_k \, N(\mu_k, \Sigma_k)$$

- Marginalize X_i

$$p(x_{i+1} \mid z_{i+1}=j) = \int_{x_i} p(x_{i+1} \mid x_i, z_i=j) \cdot p(x_i)\, dx_i$$

$$= \sum_{k=1}^{m} w_k \underbrace{\int_{x_i} p(x_{i+1} \mid x_i, z_i=j)\, N(\mu_k, \Sigma_k)\, dx_i}$$

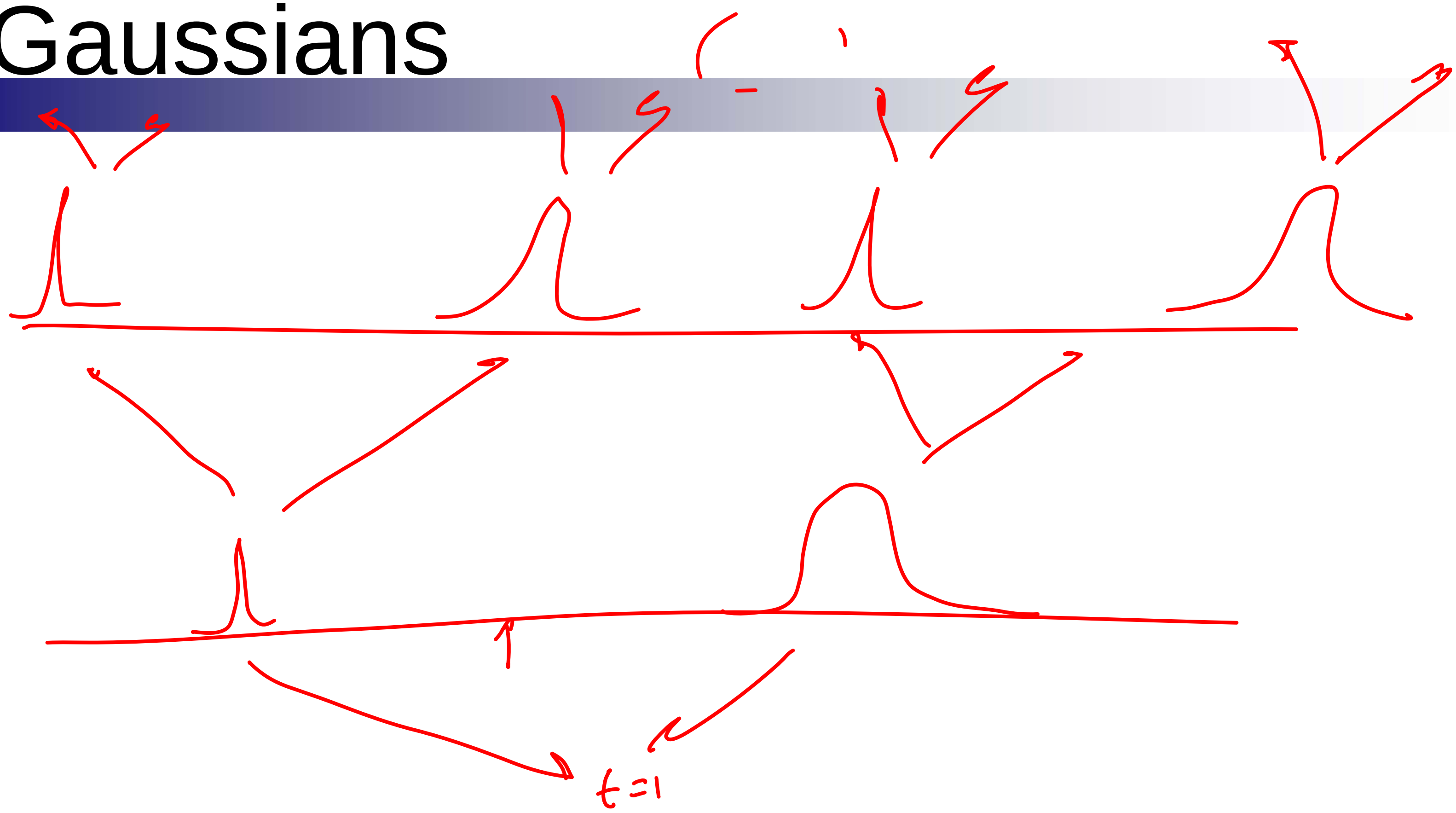
- Marginalize Z_i

$$p(x_{i+1}) = \sum_j p(z_{i+1}=j) \cdot p(x_{i+1} \mid z_{i+1}=j)$$

- Obtain mixture of 2*m* Gaussians!
  - Number of Gaussians grows exponentially!!!

m bumps

Visualizing growth in number of Gaussians



Computational complexity of inference in switching Kalman filters

- Switching Kalman Filter with (only) 2 motion models

- Query:

- Problem is NP-hard!!! [Lerner & Parr `01]
  - Why “!!!”?
  - Graphical model is a tree:
    - Inference efficient if all are discrete
    - Inference efficient if all are Gaussian
    - But not with hybrid model (combination of discrete and continuous)

Bounding number of Gaussians

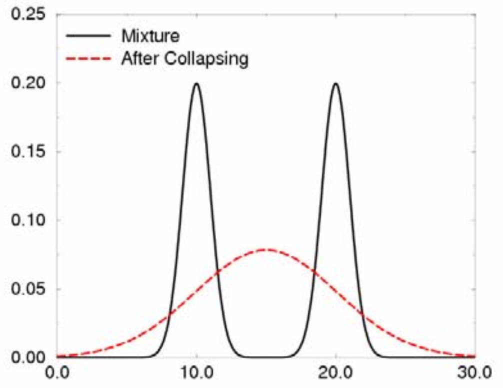
- $P(X_i)$ has 2^m Gaussians, but…
- usually, most are bumps have low probability and overlap:

- **Intuitive approximate inference**:
  - Generate *k.m* Gaussians
  - Approximate with *m* Gaussians

Collapsing Gaussians – Single Gaussian from a mixture

- Given mixture *P* <w_i;*N*(μ_i,Σ_i)>
- Obtain approximation *Q*~*N*(μ,Σ) as:

$$\mu = \sum_i w_i \mu_i$$
$$\Sigma = \sum_i w_i \Sigma_i + \sum_i w_i (\mu_i - \mu)(\mu_i - \mu)^T$$

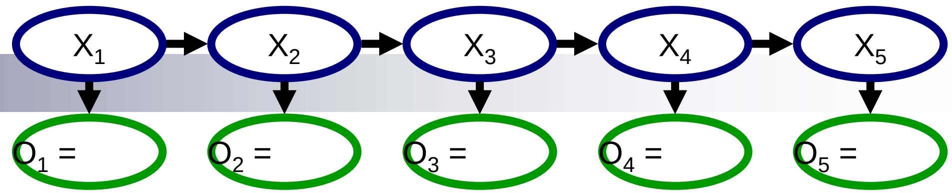


- **Theorem**:
  - *P* and *Q* have same first and second moments
  - **KL projection:** *Q* is single Gaussian with lowest KL divergence from *P*

Collapsing mixture of Gaussians into smaller mixture of Gaussians

- Hard problem!
  - Akin to clustering problem…
- Several heuristics exist
  - *c.f.*, Uri Lerner's Ph.D. thesis

Operations in non-linear switching Kalman filter



- Compute mixture of Gaussians for $p(X_t \mid O_{1:t} = o_{1:t})$
- Start with $p(X_0)$
- At each time step *t*:
  - For each of the *m* Gaussians in $p(X_i|o_{1:i})$:
    - **Condition** on observation (use **numerical integration**)
    - **Prediction** (Multiply transition model, use **numerical integration**)
      - Obtain *k* Gaussians
    - **Roll-up** (marginalize previous time step)
  - **Project** *k.m* Gaussians into *m'* Gaussians $p(X_i|o_{1:i+1})$

Assumed density filtering

- Examples of very important **assumed density filtering**:
  - Non-linear KF
  - Approximate inference in switching KF

- General picture:
  - Select an **assumed density**
    - e.g., single Gaussian, mixture of *m* Gaussians, …
  - After conditioning, prediction, or roll-up, **distribution no-longer representable with assumed density**
    - e.g., non-linear, mixture of *k.m* Gaussians,…
  - **Project** back into assumed density
    - e.g., numerical integration, collapsing,…

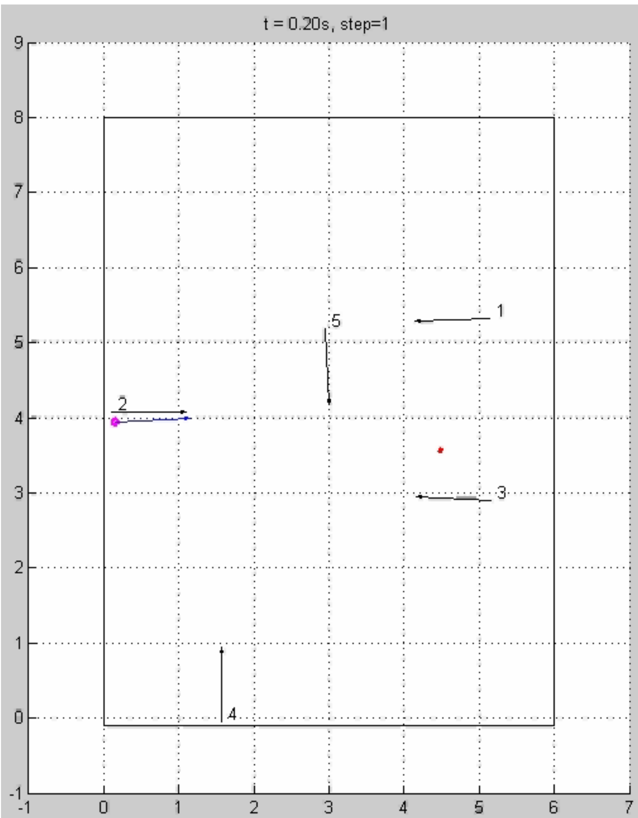
When non-linear KF is not good enough

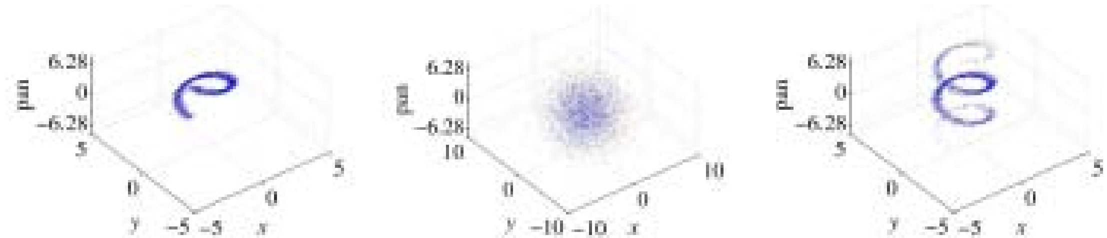
- Sometimes, distribution in non-linear KF is not approximated well as a single Gaussian
  - e.g., a banana-like distribution

- Assumed density filtering:
  - Solution 1: **reparameterize problem** and solve as a **single Gaussian**
  - Solution 2: more typically, **approximate as a mixture of Gaussians**

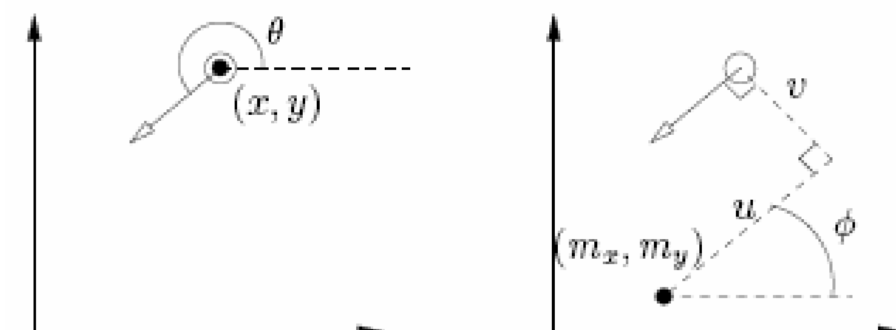
Reparameterized KF for SLAT

[Funiak, Guestrin, Paskin, Sukthankar '05]



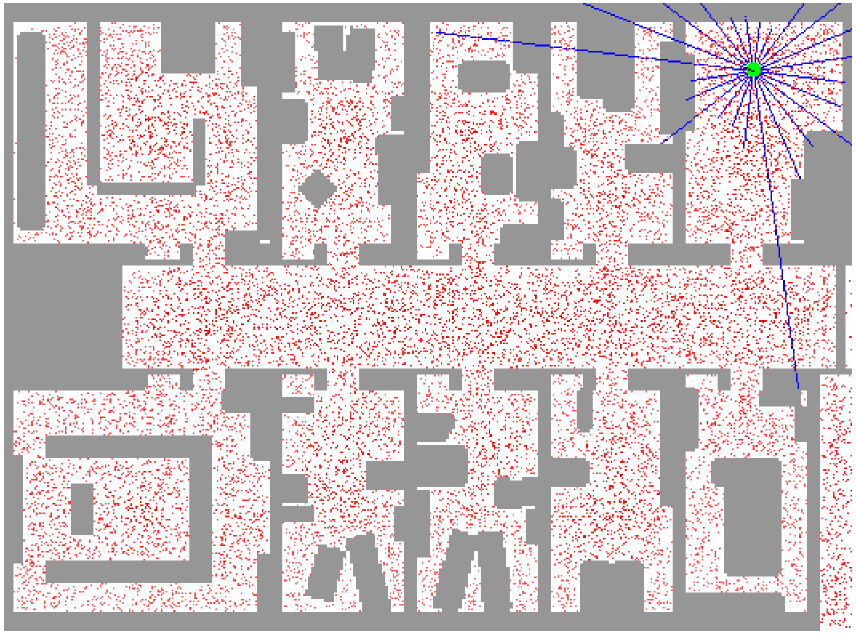


(a) true posterior (b) Gaussian in absolute parameters (c) Gaussian in relative parameters



(a) absolute parameters (b) ROP parameters

When a single Gaussian ain't good enough

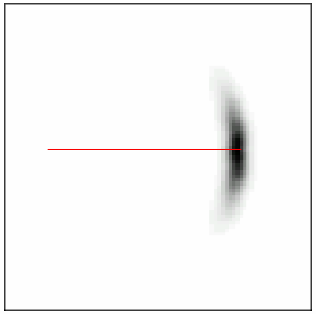


[Fox et al.]

- Sometimes, smart parameterization is not enough
  - Distribution has multiple hypothesis
- Possible solutions
  - Sampling – particle filtering
  - Mixture of Gaussians
  - …
- Quick overview of one such solution…

Approximating non-linear KF with mixture of Gaussians

- Robot example:



- $P(X_i)$ is a Gaussian, $P(X_{i+1})$ is a banana
- Approximate $P(X_{i+1})$ as a mixture of *m* Gaussians
  - e.g., using discretization, sampling,…
- Problem:
  - $P(X_{i+1})$ as a mixture of *m* Gaussians
  - $P(X_{i+2})$ is *m* bananas
- One solution:
  - Apply collapsing algorithm to project *m* bananas in *m*’ Gaussians

What you need to know

- **Kalman filter**
  - Probably most used BN
  - Assumes Gaussian distributions
  - Equivalent to linear system
  - Simple matrix operations for computations
- **Non-linear Kalman filter**
  - Usually, observation or motion model not CLG
  - Use numerical integration to find Gaussian approximation
- **Switching Kalman filter**
  - Hybrid model – discrete and continuous vars.
  - Represent belief as mixture of Gaussians
  - Number of mixture components grows exponentially in time
  - Approximate each time step with fewer components
- **Assumed density filtering**
  - Fundamental abstraction of most algorithms for dynamical systems
  - Assume representation for density
  - Every time density not representable, project into representation