**Koller & Friedman: Chapter 16**
**Boyen & Koller '98, '99**
**Uri Lerner's Thesis: Chapters 3,9**
**Paskin '03**

# Dynamic models 2

## Switching KFs continued, Assumed density filters, DBNs, BK, extensions

Probabilistic Graphical Models – 10708

Carlos Guestrin
Carnegie Mellon University

November 21st, 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!!!
- It’s FCE time!!!
  - Fill the forms online by Dec. 11
  - www.cmu.edu/fce
  - It will only take a few minutes
  - Please, please, please help us improve the course by providing feedback

# Last week in “Your BN Hero”

- Gaussian distributions reviewed
  - Linearity of Gaussians
  - Conditional Linear Gaussian (CLG)
- Kalman filter
  - HMMs with CLG distributions
  - Linearization of non-linear transitions and observations using numerical integration
- Switching Kalman filter
  - Discrete variable selects transition model depends
  - Mixture of Gaussians represents belief state
  - Number of mixture components grows exponentially in time

# The moonwalk

# Last week in “Your BN Hero”

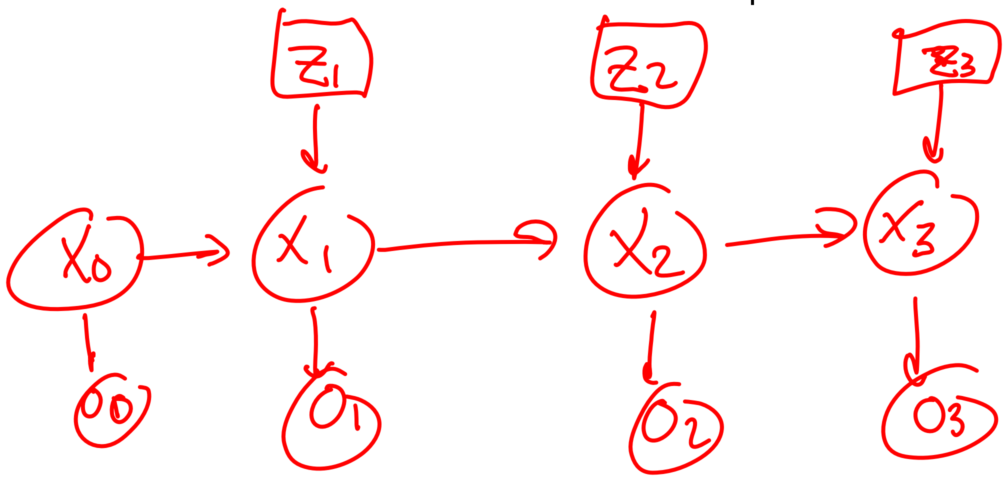
- Gaussian distributions reviewed
  - Linearity of Gaussians
  - Conditional Linear Gaussian (CLG)
- Kalman filter
  - HMMs with CLG distributions
  - Linearization of non-linear transitions and observations using numerical integration
- Switching Kalman filter
  - Discrete variable selects transition model depends
  - Mixture of Gaussians represents belief state
  - Number of mixture components grows exponentially in time

# Switching Kalman filter

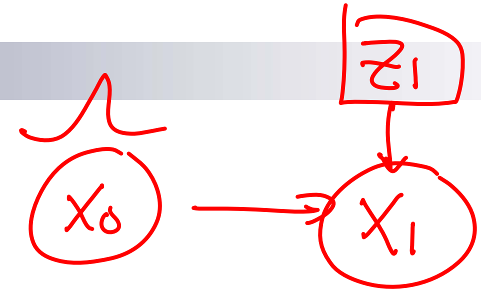
- At each time step, choose one of *k* motion models:
  - You never know which one!

depending on motion model

- $p(X_{i+1}|X_i,Z_{i+1})$
  - CLG indexed by $Z_i$
  - $p(X_{i+1}|X_i,Z_{i+1}=j) \sim N(\beta^j_0 + B^j X_i; \Sigma^j_{X_{i+1}|X_i})$



# 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 $X_0$

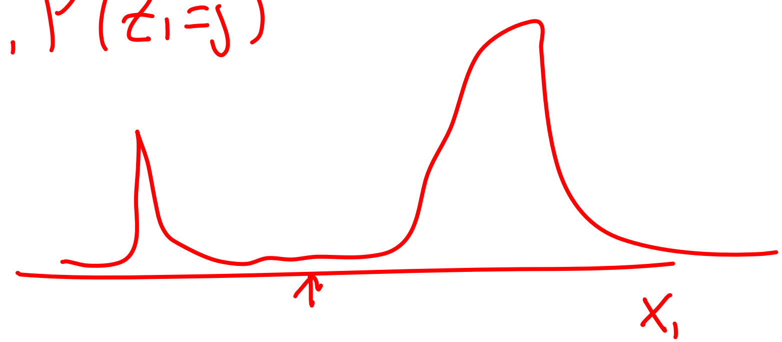
$$p(x_1|z_1) = \int_{x_0} p(x_0) \cdot p(x_1|x_0,z_1)\, dx_0$$

$z_1$ → straight
$z_1$ → moonwalk

- Marginalize $Z_1$

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

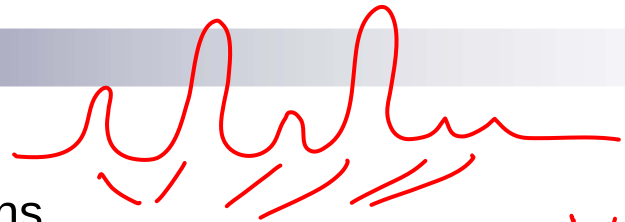
- Obtain mixture of two Gaussians!



# Multi-step inference

$X_i \to X_{i+1}$, $Z_{i+1}$

- 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

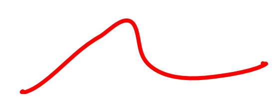


$$p(x_i) = \sum_{k=1}^{m} w_k N(\mu_k, \Sigma_k)$$

- Marginalize $X_i$

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

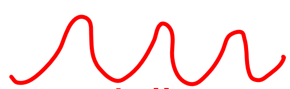
$$= \sum_{k=1}^{m} w_k \underbrace{\int_{x_i} p(x_{i+1} | x_i, z_i = j) N(\mu_k, \Sigma_k)\, dx_i}$$



- Marginalize $Z_i$

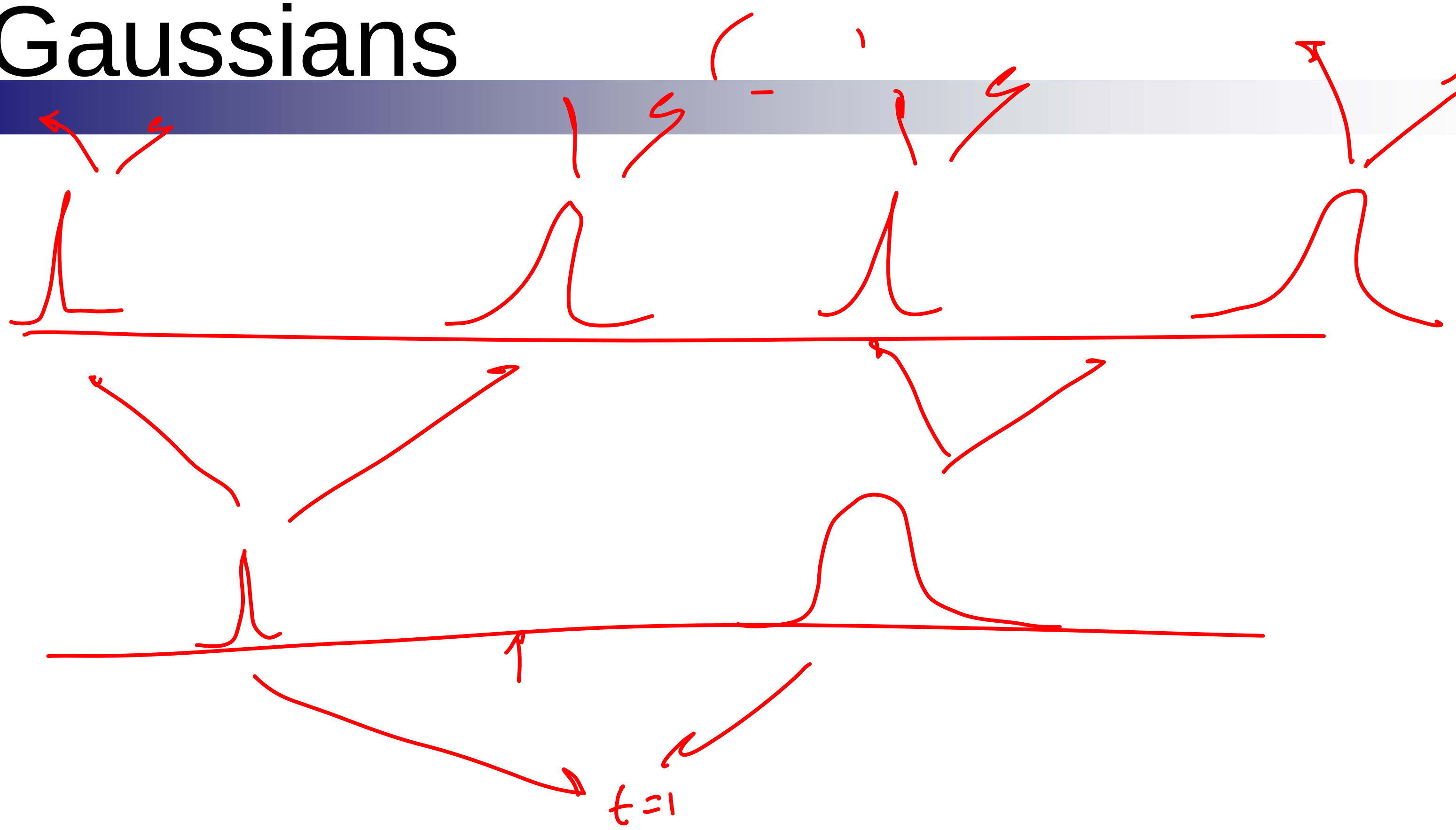
$$p(x_{i+1}) = \sum_j p(z_{i+1} = j) \cdot p(x_{i+1} | z_{i+1} = j)$$

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



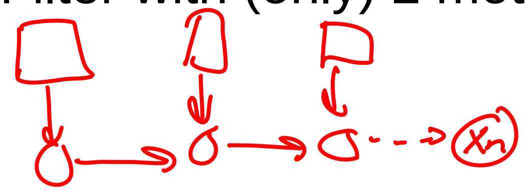
m bumps

# Visualizing growth in number of Gaussians



# Computational complexity of inference in switching Kalman filters

- Switching Kalman Filter with (only) 2 motion models



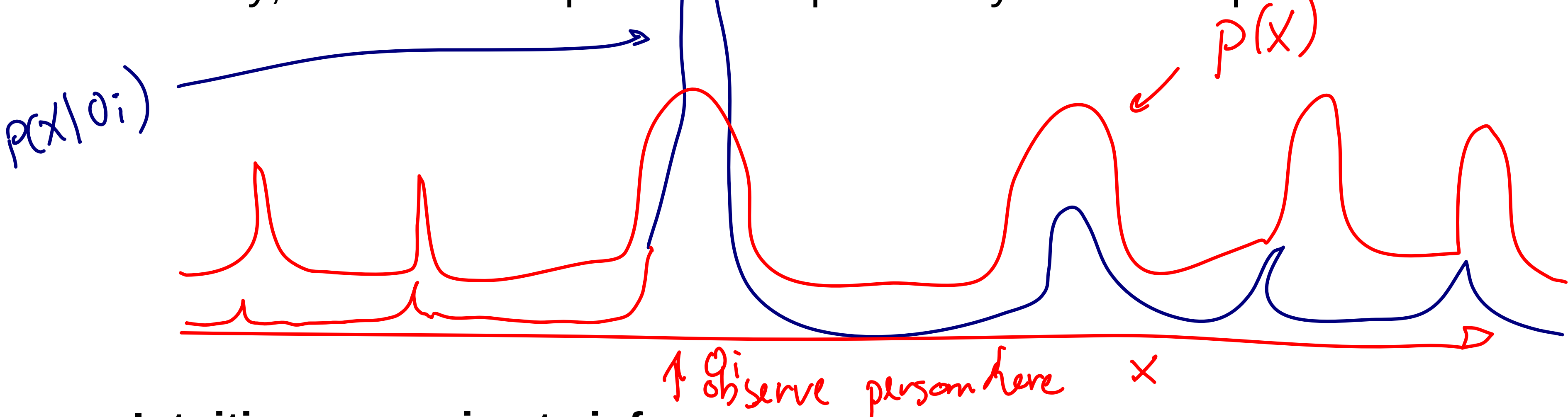

- Query:

$p(x_n)$

- 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:



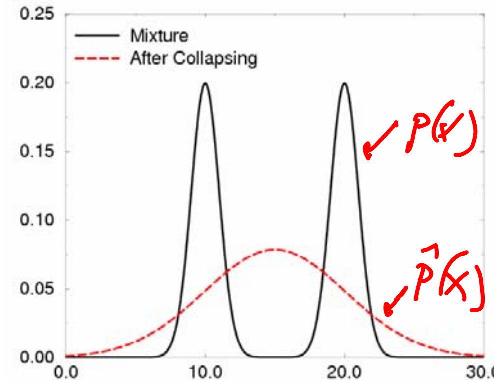

- **Intuitive approximate inference**:
  - Generate *k.m* Gaussians
  - Approximate with *m'* Gaussians

# Collapsing Gaussians – Single Gaussian from a mixture

- Given mixture $P\ \langle w_i; N(\mu_i, \Sigma_i) \rangle$ ↙ true
- Obtain approximation $Q \sim N(\mu, \Sigma)$ as: ↖ approx

$$\mu = \sum_i w_i \mu_i \quad \leftarrow \text{weighted sum}$$

$$\Sigma = \underbrace{\sum_i w_i \Sigma_i}_{\text{weighted sum of } \Sigma_i} + \underbrace{\sum_i w_i (\mu_i - \mu)(\mu_i - \mu)^T}_{\text{how far from new mean.}}$$

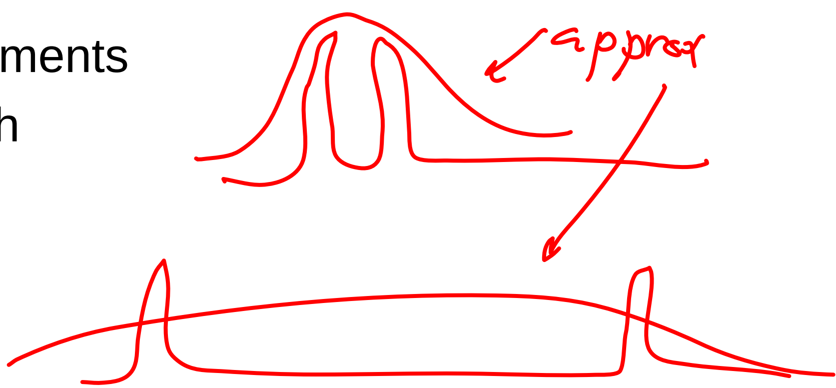



P(x)

$\hat{P}(x)$

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

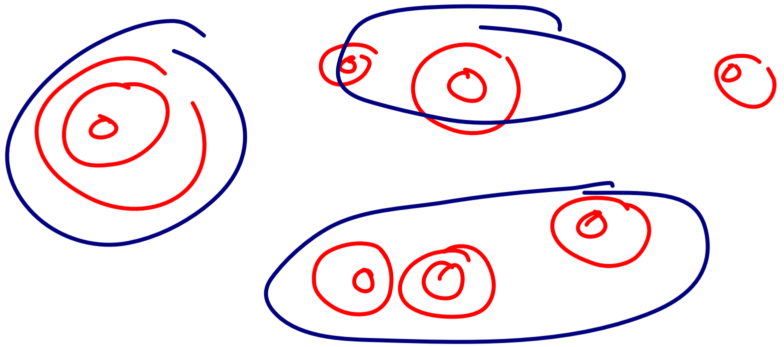
$$Q = \arg\min_{\hat{Q} \sim N} KL(P \| \hat{Q})$$



# Collapsing mixture of Gaussians into smaller mixture of Gaussians

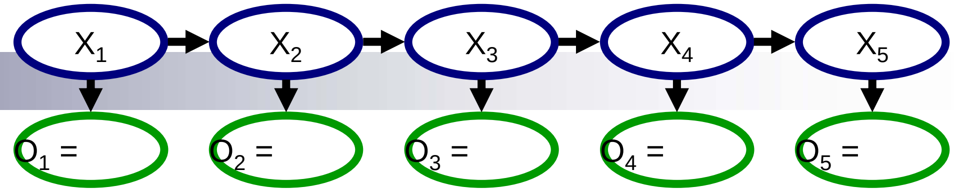
- Hard problem!
  - Akin to clustering problem…

Similar to fitting mixture of k-Gaussians to data



- Several heuristics exist
  - *c.f.*, Uri Lerner's Ph.D. thesis

# Operations in non-linear switching Kalman filter

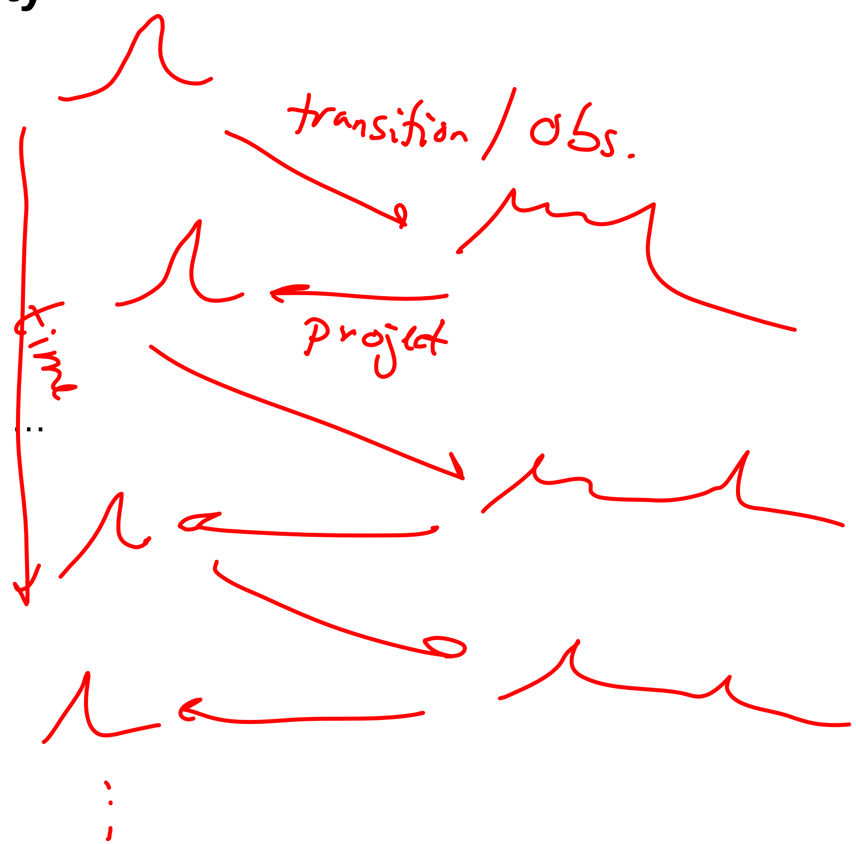



- Compute mixture of Gaussians for $p(X_t \mid O_{1:t} = o_{1:t})$
- Start with $p(X_0)$ mixt. Gauss.
- At each time step *t*:
  - For each of the *m* Gaussians in $p(X_t|o_{1:t})$:
    - **Condition** on observation (use **numerical integration**)
    - **Prediction** (Multiply transition model, use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

# Assumed density filtering

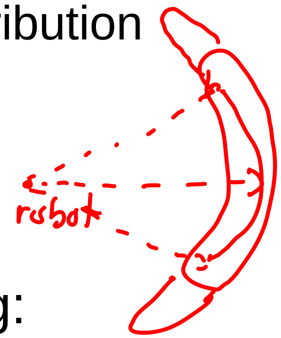
- Examples of very important **assumed density filtering**:
  - Non-linear KF
  - Approximate inference in switching KF

- 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,…

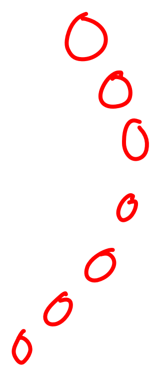


# When non-linear KF is not good enough

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



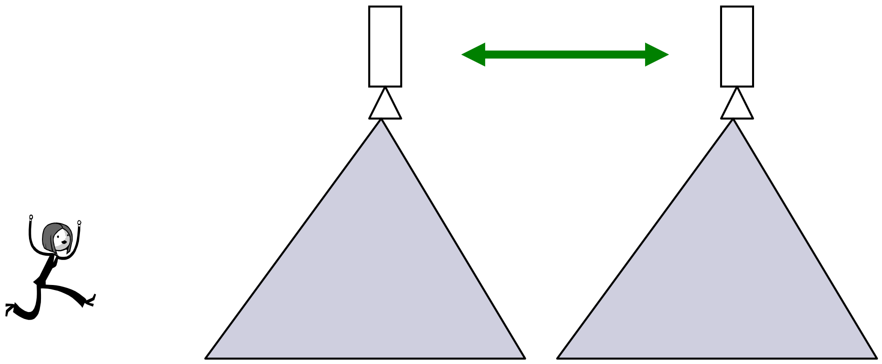

- Assumed density filtering:
  - Solution 1: **reparameterize problem** and solve as a **single Gaussian**
  - Solution 2: more typically, **approximate as a mixture of Gaussians**



# Distributed Simultaneous Localization and Tracking

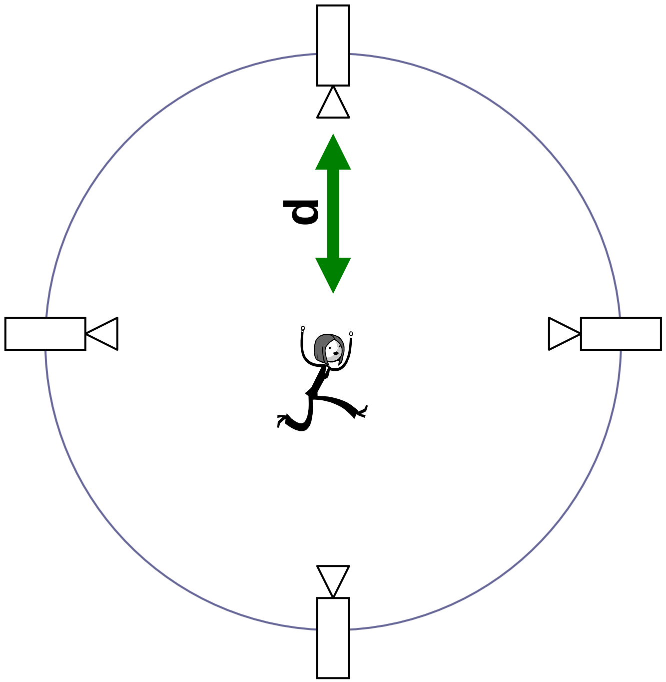
[Funiak, Guestrin, Paskin, Sukthankar '05]

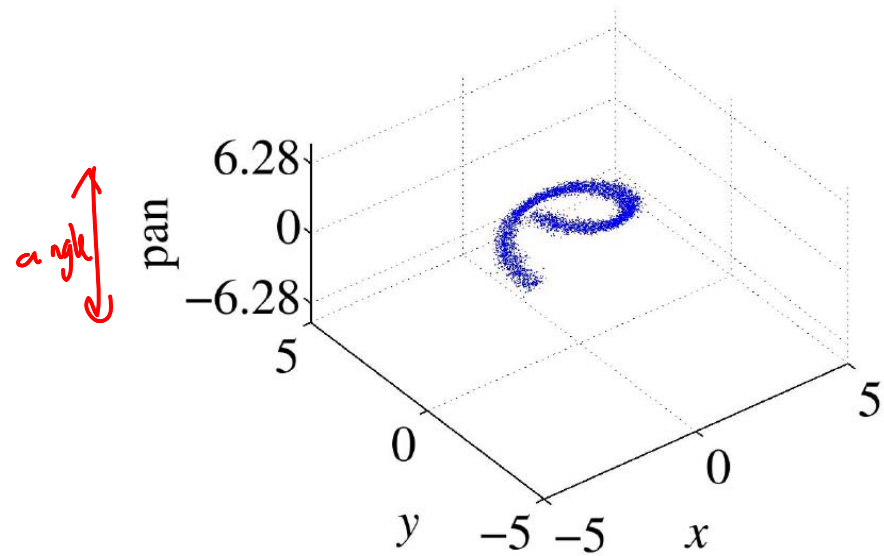
- Place cameras around an environment, don't know where they are
- Could measure all locations, but requires lots of grad. student time
- Intuition:
  - A person walks around
  - If camera 1 sees person, then camera 2 sees person, learn about relative positions of cameras



# Donut and Banana distributions

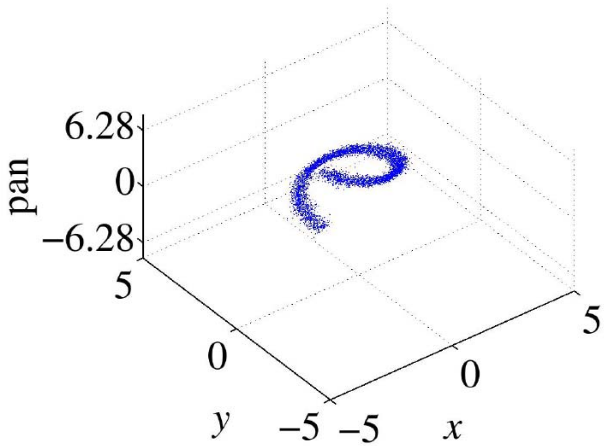
- Observe person at distance d
- Camera could be anywhere in a ring



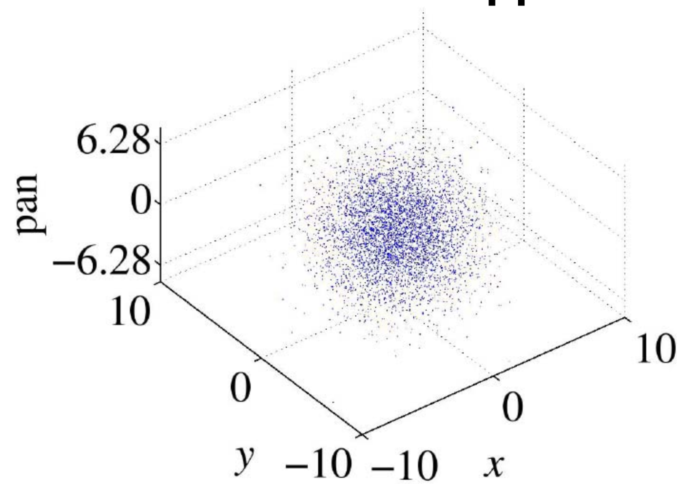



# Gaussians represent “balls”

**True distribution**

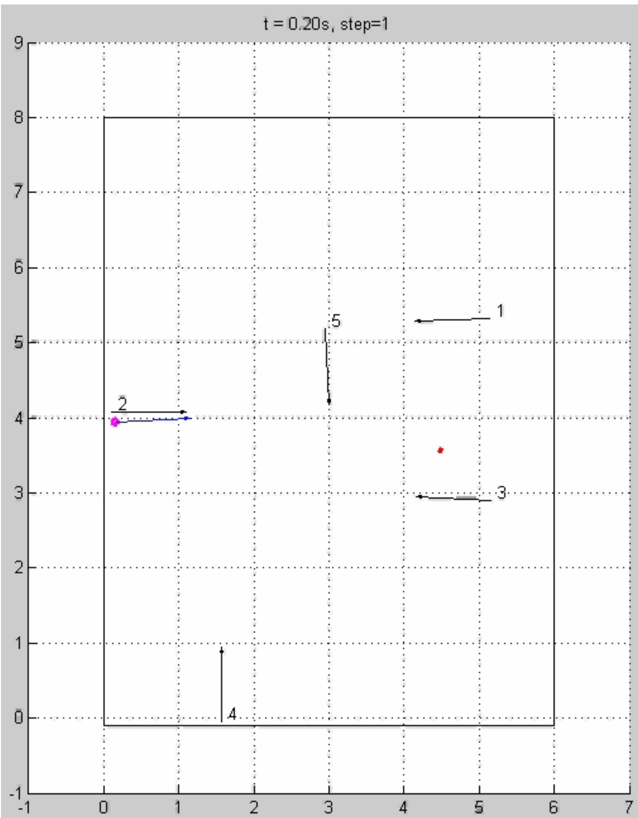


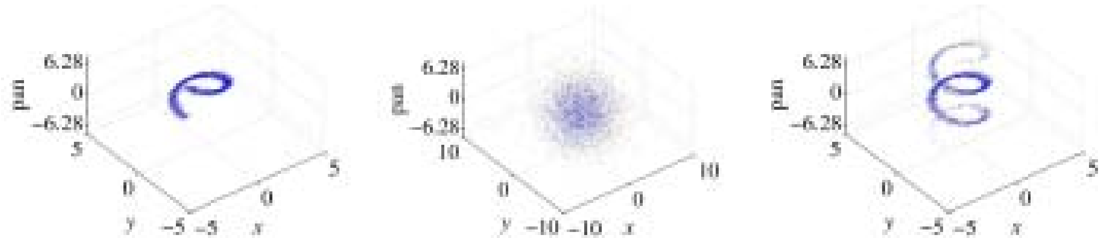

**Gaussian approximation**



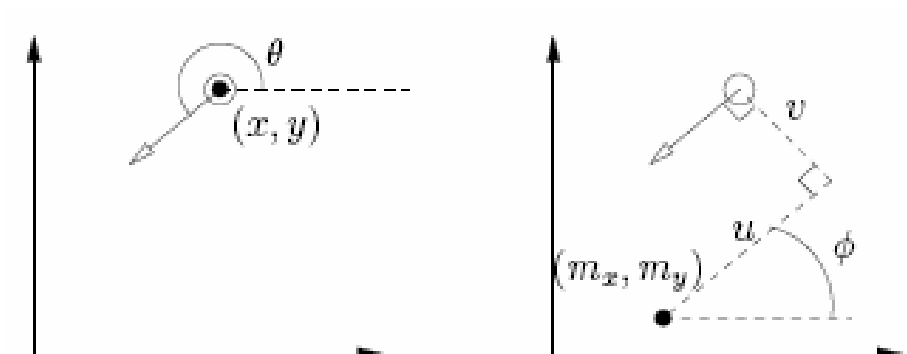

- **Gaussian approximation leads to poor results**
- **Can’t apply standard Kalman filter ☹**
- **Or can we… ☺**

# Reparameterized KF for SLAT



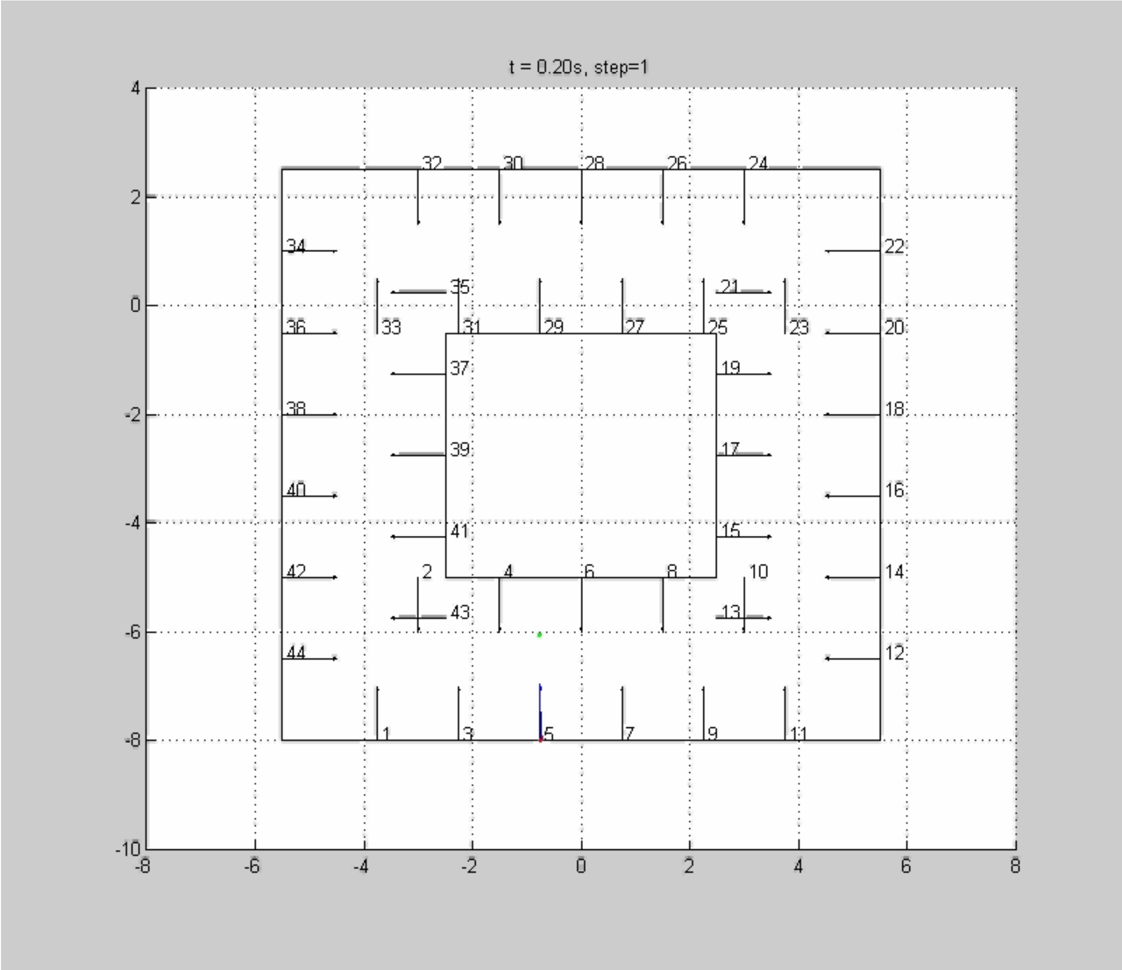



(a) true posterior (b) Gaussian in absolute parameters (c) Gaussian in relative parameters

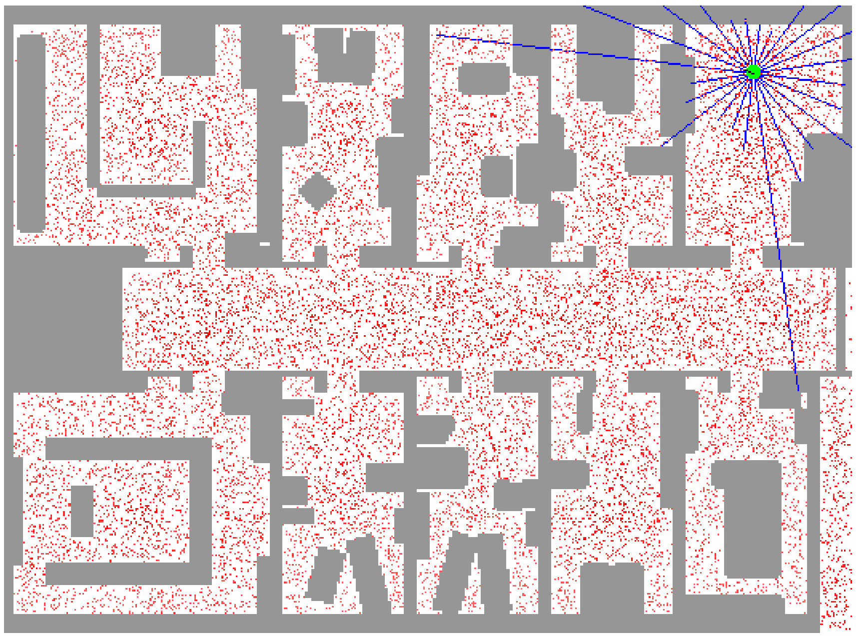



(a) absolute parameters (b) ROP parameters

# Example of KF – SLAT
## Simultaneous Localization and Tracking



# When a single Gaussian ain't good enough

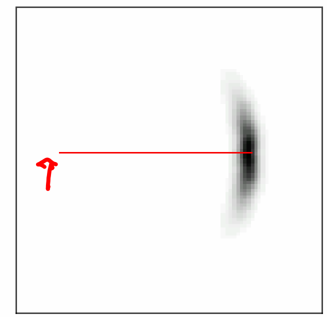


[Fox et al.]

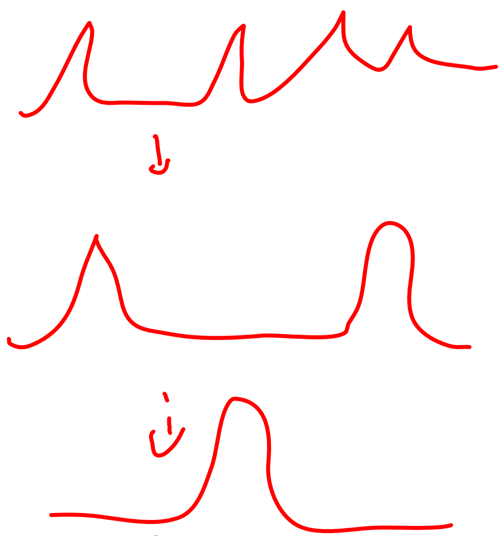
- Sometimes, smart parameterization is not enough
  - Distribution has multiple hypothesis
- Possible solutions
  - Sampling – particle filtering
  - Mixture of Gaussians
  - …
- Quick overview of one such solution…

# 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

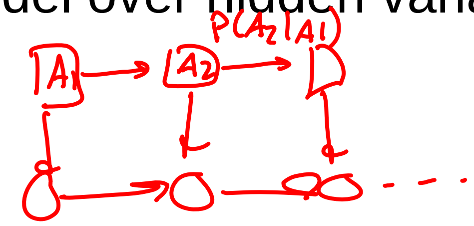


# What you need to know about switching Kalman filters

- **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

# More than just a switching KF

- Switching KF selects among *k* motion models
- Discrete variable can depend on past
  - Markov model over hidden variable




- What if *k* is really large?
  - Generalize HMMs to large number of variables

# Dynamic Bayesian network (DBN)

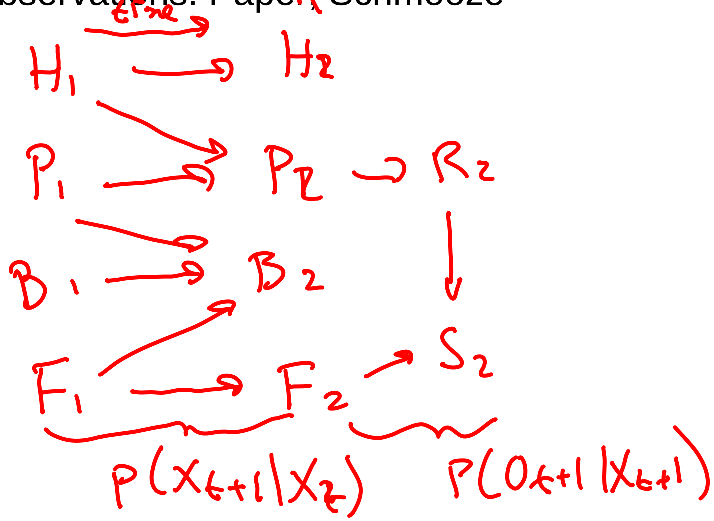
- HMM defined by
  - Transition model $P(X_{t+1}|X_t)$
  - Observation model $P(O_t|X_t)$
  - Starting state distribution $P(X_0)$
- DBN – Use Bayes net to represent each of these compactly
  - Starting state distribution $P(X_0)$ is a BN
  - (silly) e.g, performance in grad. school DBN
    - Vars: Happiness, Productivity, Hirablility, Fame
    - Observations: Paper, Schmooze

$x_t \quad x_{t+1}$

$O((10^4)^2)$

10 values each

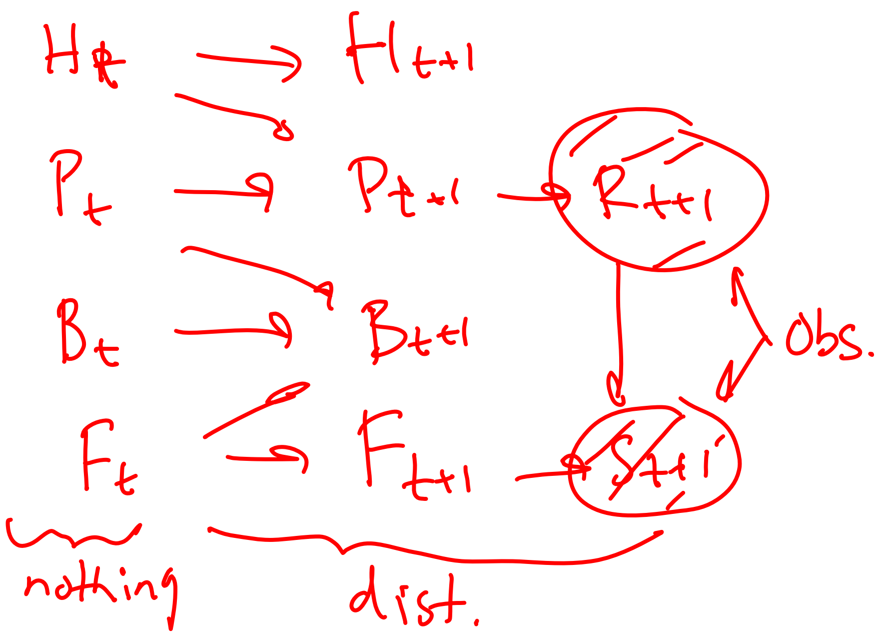
many fewer params.




$P(X_0)$:

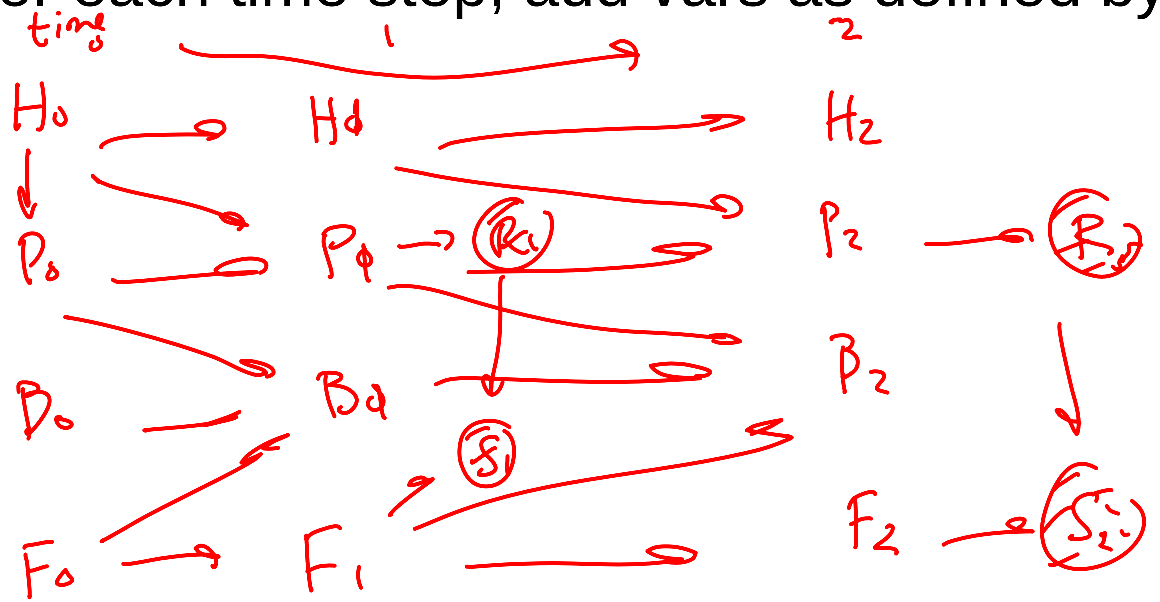
# Transition Model: Two Time-slice Bayes Net (2-TBN)

- Process over vars. **X**
- 2-TBN: represents transition and observation models $P(\mathbf{X}_{t+1}, \mathbf{O}_{t+1} | \mathbf{X}_t)$
  - $\mathbf{X}_t$ are *interface variables* (don't represent distribution over these variables)
  - As with BN, exponential reduction in representation complexity



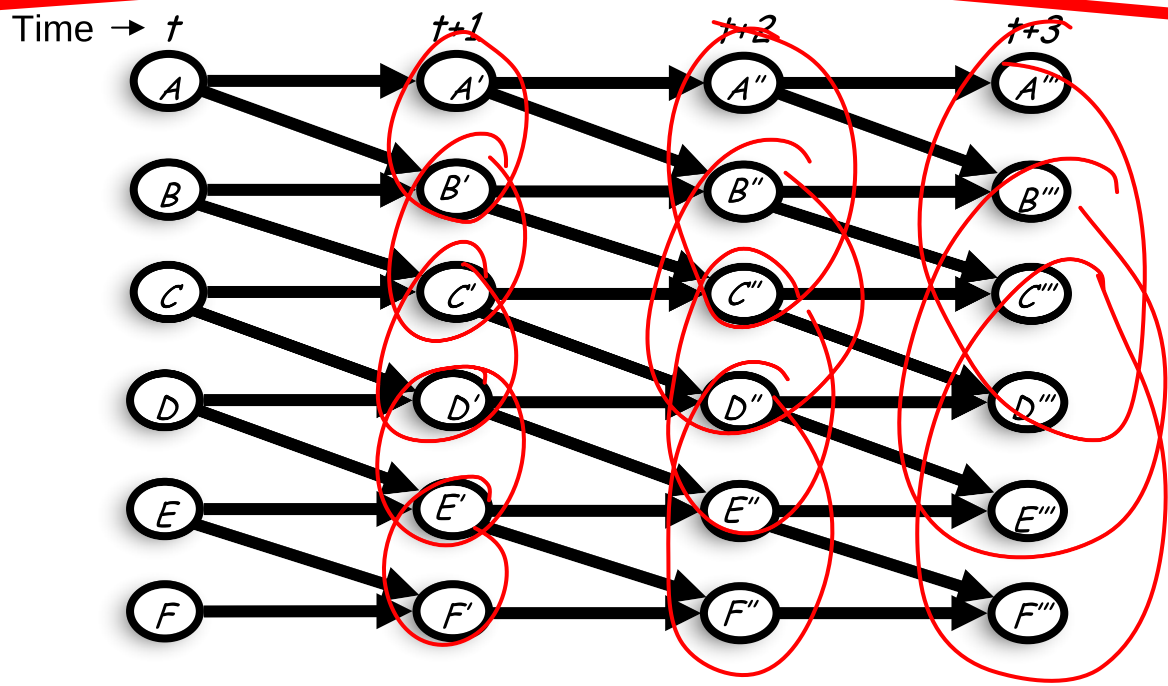
# Unrolled DBN

- Start with $P(X_0)$
- For each time step, add vars as defined by 2-TBN



# “Sparse” DBN and fast inference

~~“Sparse” DBN ☹ ⇒ Fast inference~~



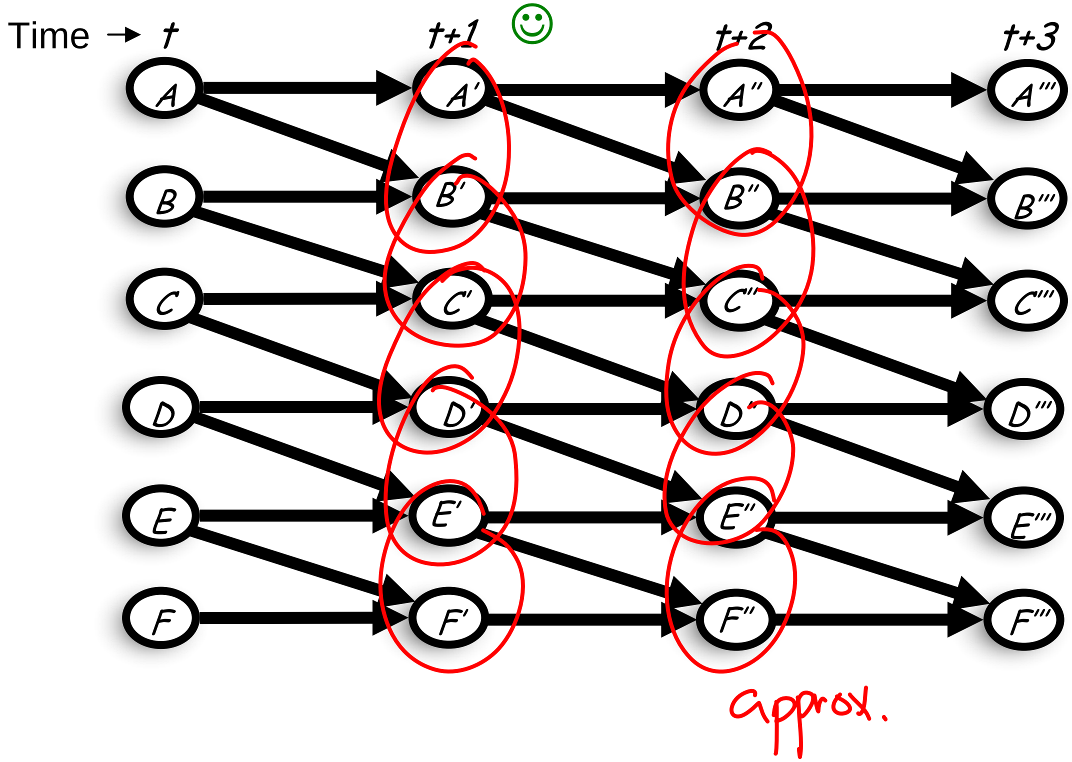
# "Sparse" DBN and fast inference 1

**Structured representation of belief often yields good approximate**

**Almost!**

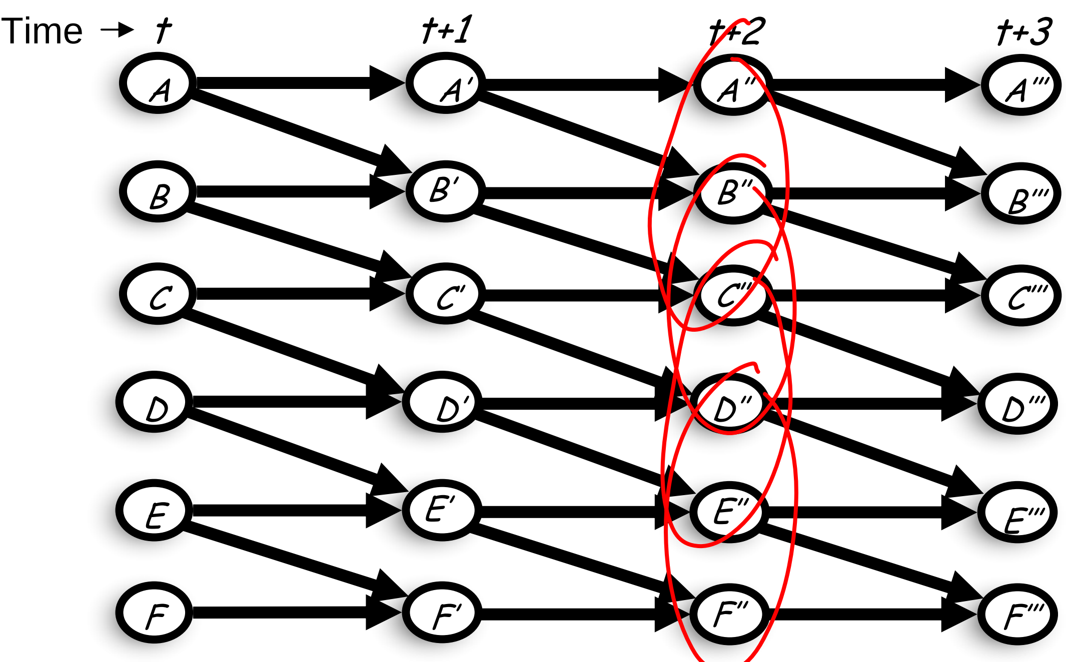
"Sparse" DBN **?** Fast inference

☺



# BK Algorithm for approximate DBN inference [Boyen, Koller '98]

- Assumed density filtering:
  - Choose a factored representation $\hat{b}$ for the belief state
  - Every time step, belief not representable with $\hat{b}$, project into representation




Project $\hat{b}$:

ABC — BCD — CDE — DE — DEF

P(CDE)

P(DEF)

project →

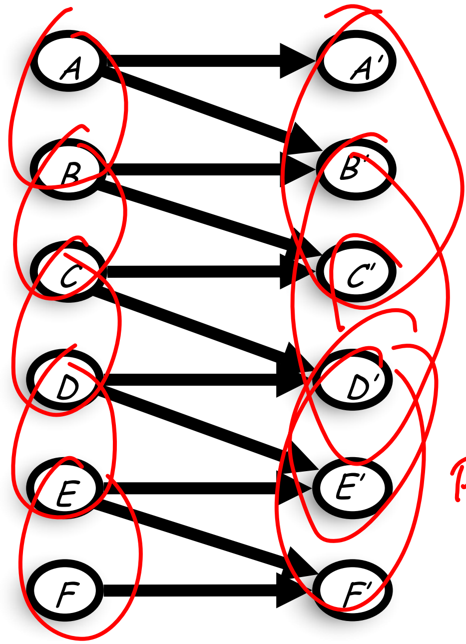
same

AB — BC — CD — DE — EF

P(DE)

# Computing factored belief state in the next time step

- Introduce observations in current time step
  - Use J-tree to calibrate time *t* beliefs
- Compute *t+1* belief, project into approximate belief state
  - marginalize into desired factors
  - corresponds to KL projection
- Equivalent to computing marginals over factors directly
  - For each factor in *t+1* step belief
    - Use variable elimination




JT t:

ABC - multiply obs.

|

BCD - calibrate

|

CDE

Project into

AB

|

BC → apply transition model

|

CD

|

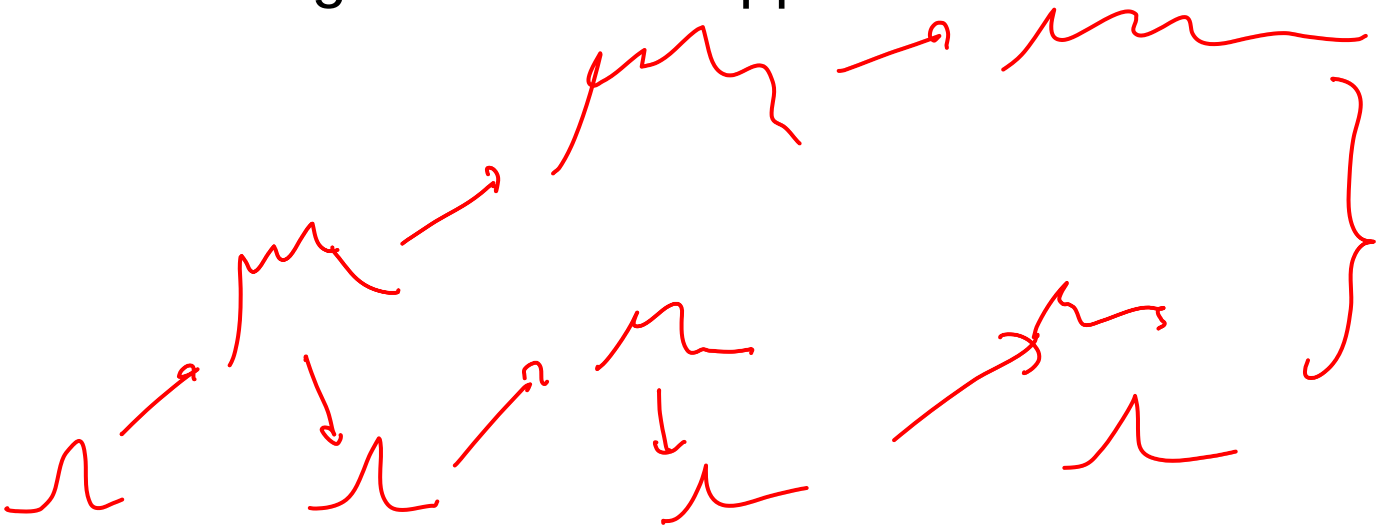
DE

P(ABC) → AB − BC

→ $\frac{P(AB).P(BC)}{P(B)}$

# Error accumulation

- Each time step, projection introduces error
- Will error add up?
  - causing unbounded approximation error as $t \to \infty$

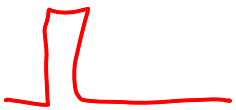


error → ∞
as $t \to \infty$

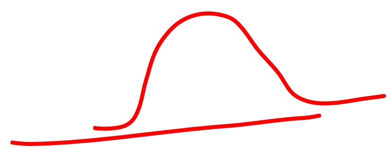
# Contraction in Markov process

at time $t$ $\Delta t$ $t+1$

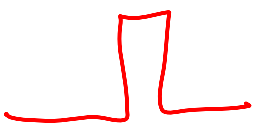
$p$ :



closer after transition



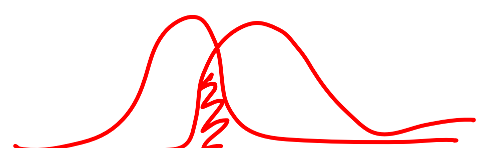
$q$ :



mixing rate $\gamma < 1$

diff. high

$\gamma\, d(p_t, q_t) \geq d(p_{t+1}, q_{t+1})$



overlap

# BK Theorem

- Error does not grow unboundedly!

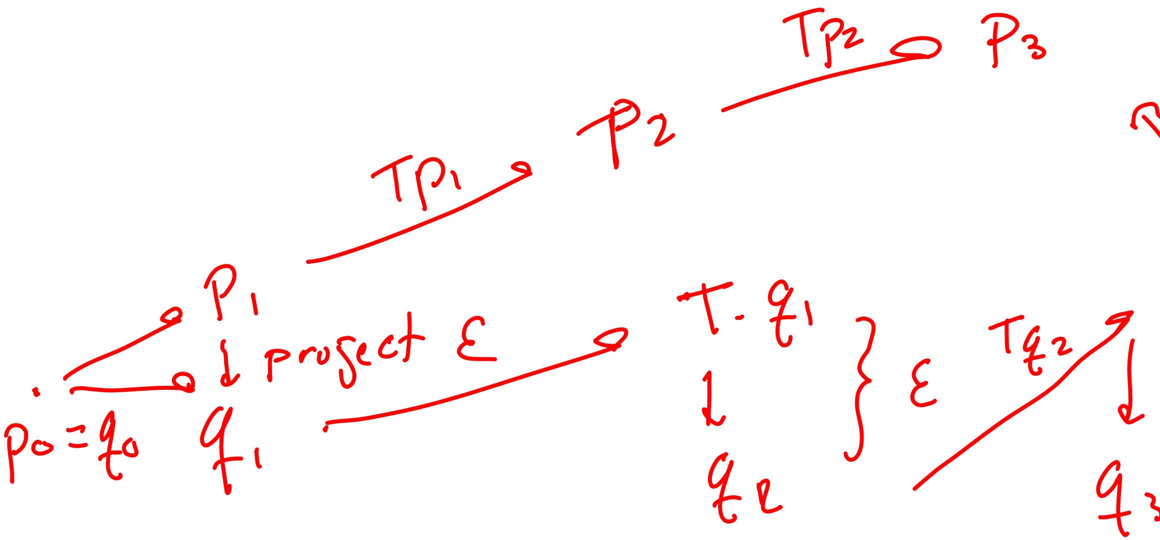
$p_t \to$ truth

$q_t \to$ approx. at $t$

| | |
|---|---|
| $t=1$ | $\varepsilon$ |
| $t=2$ | $\gamma\varepsilon + \varepsilon$ |
| $t=3$ | $\gamma(\gamma\varepsilon+\varepsilon) + \varepsilon$ |

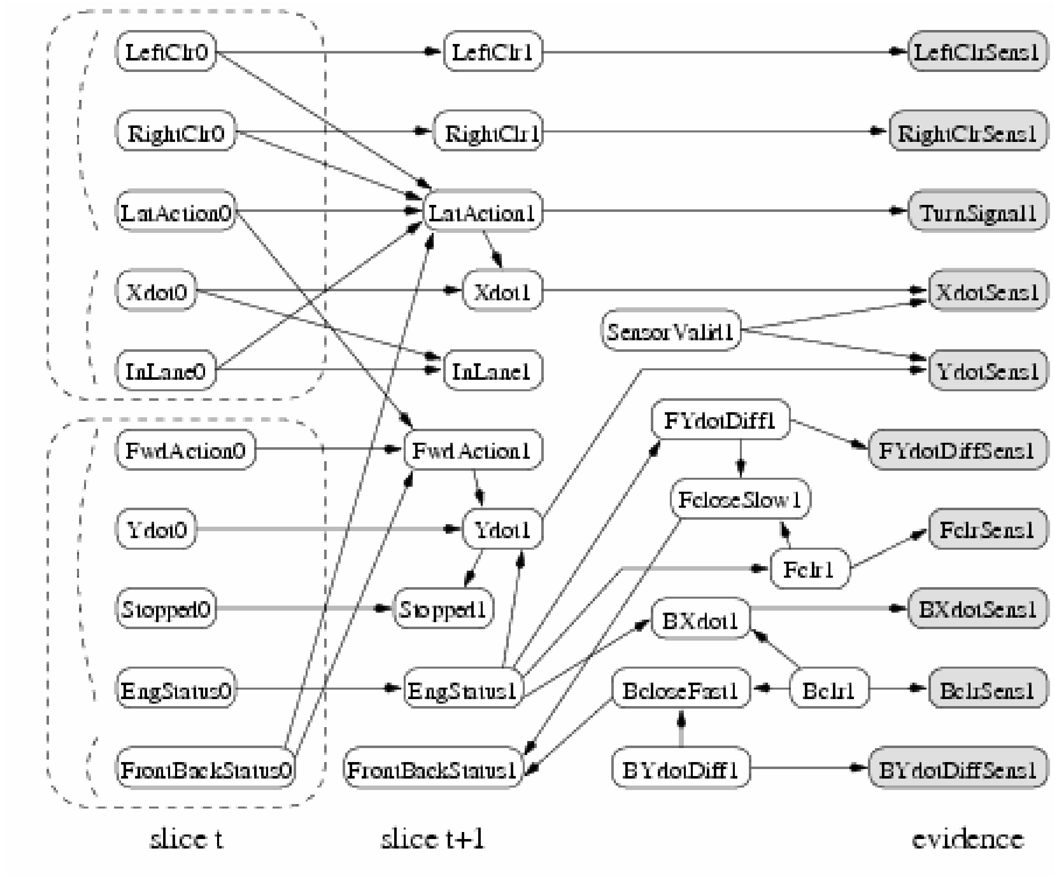
$\vdots$

$\leq \boxed{\frac{\varepsilon}{1-\gamma}}$

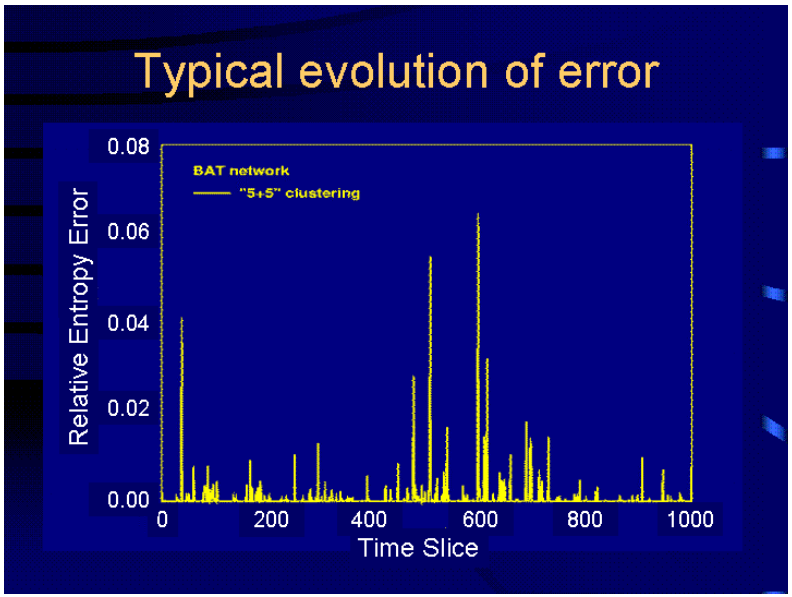


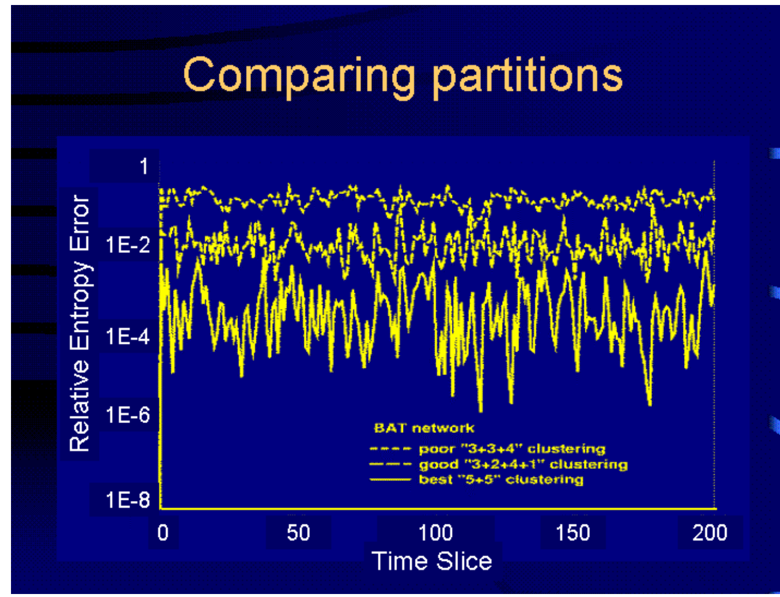

$$d(p_2; q_2) = \underbrace{d(Tp_1, Tq_1)}_{\gamma d(p_1, q_1)} + \overbrace{d(Tq_1, q_2)}^{\varepsilon}$$

# Example – BAT network [Forbes et al.]



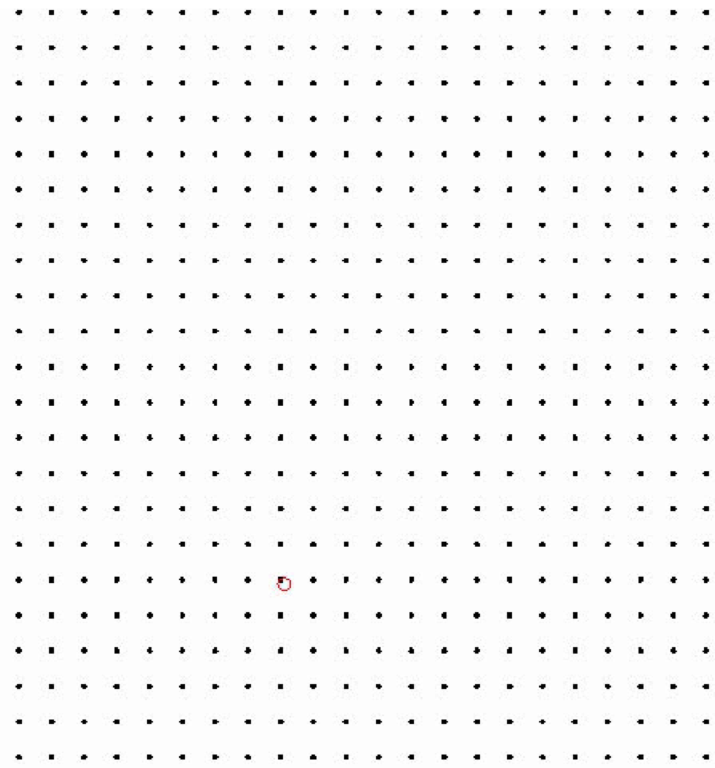
# BK results [Boyen, Koller '98]



# Thin Junction Tree Filters [Paskin '03]

- BK assumes fixed approximation clusters
- TJTF adapts clusters over time
  - attempt to minimize projection error



# Hybrid DBN (many continuous and discrete variables)

- DBN with large number of discrete and continuous variables
- # of mixture of Gaussian components blows up in one time step!
- Need many smart tricks…
  - e.g., see Lerner Thesis

Figure 10.1: The prototype RWGS system

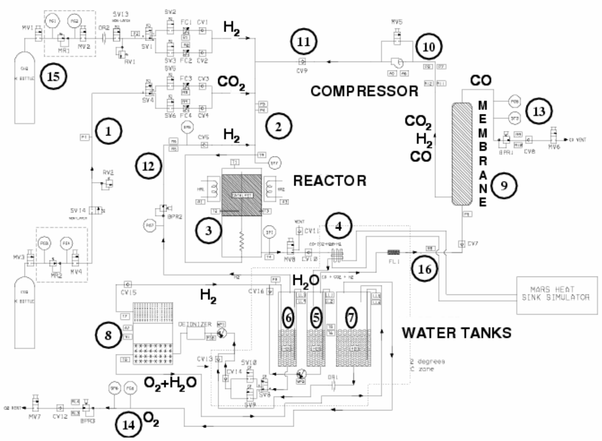



Figure 10.2: The RWGS schematic

Reverse Water Gas Shift System (RWGS) [Lerner et al. '02]

# DBN summary

- DBNs
  - factored representation of HMMs/Kalman filters
  - sparse representation does not lead to efficient inference
- Assumed density filtering
  - BK – factored belief state representation is assumed density
  - Contraction guarantees that error does blow up (but could still be large)
  - Thin junction tree filter adapts assumed density over time
  - Extensions for hybrid DBNs