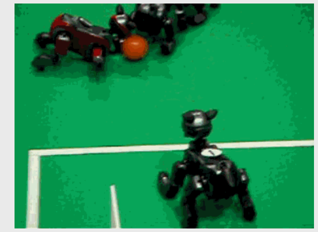
CSE-571
Robotics

Rao-Blackwelized Particle Filters for State Estimation

Ball Tracking in RoboCup



- Extremely noisy (nonlinear) motion of observer
- Inaccurate sensing, limited processing power
- Interactions between target and
t

Goal: Unified framework for modeling the ball and its interactions.

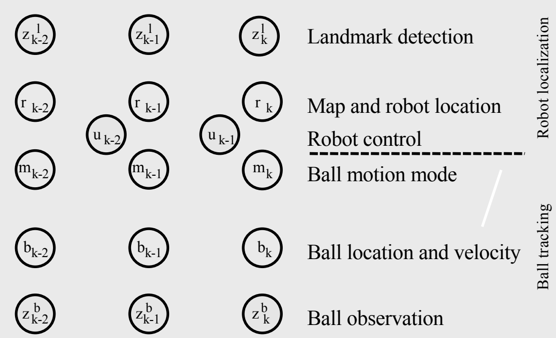
Dieter Fox CSE-571: Probabilistic Robotics 2

Tracking Techniques

- Kalman Filter
  - Highly efficient, robust (even for nonlinear)
  - Uni-modal, limited handling of nonlinearities
- Particle Filter
  - Less efficient, highly robust
  - Multi-modal, nonlinear, non-Gaussian
- Rao-Blackwellised Particle Filter, MHT
  - Combines PF with KF
  - Multi-modal, highly efficient

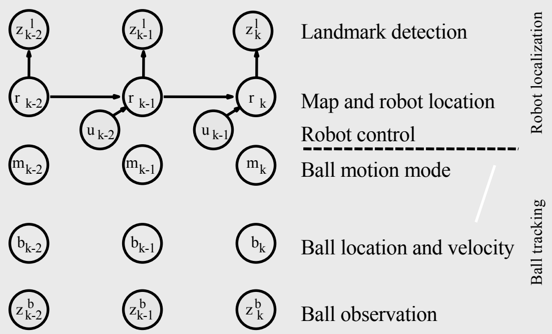
Dieter Fox CSE-571: Probabilistic Robotics 3

Dynamic Bayes Network for Ball Tracking

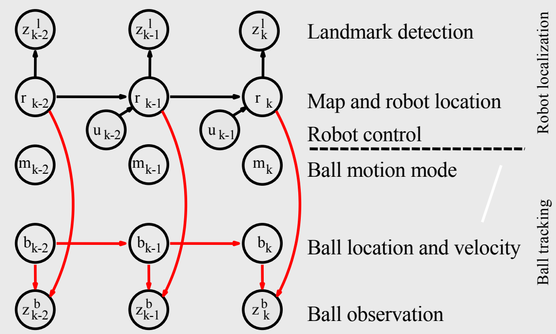


Dieter Fox CSE-571: Probabilistic Robotics 4

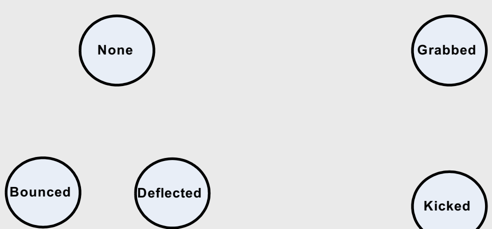
Robot Location



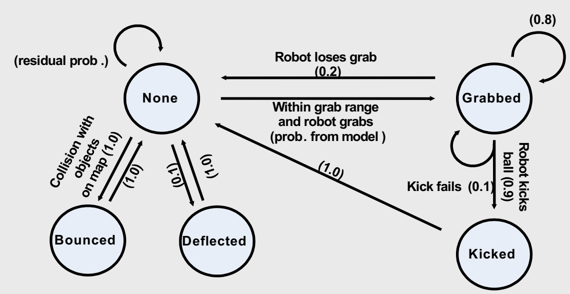
Robot and Ball Location (and velocity)



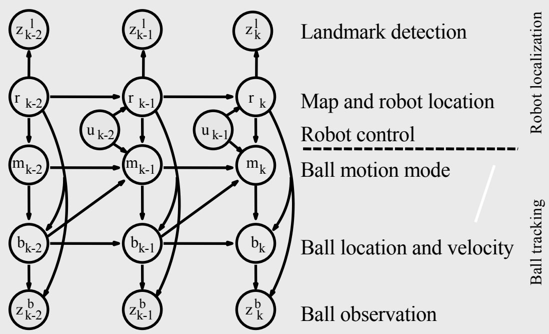
Ball-Environment Interactions



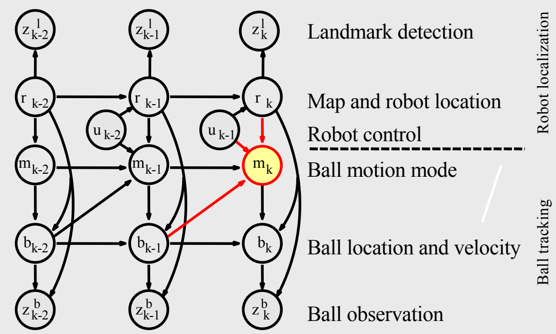
Ball-Environment Interactions



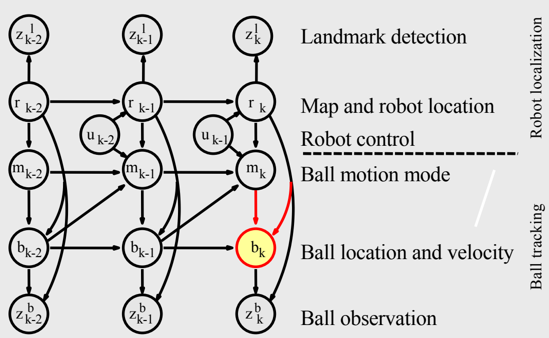
Integrating Discrete Ball Motion Mode



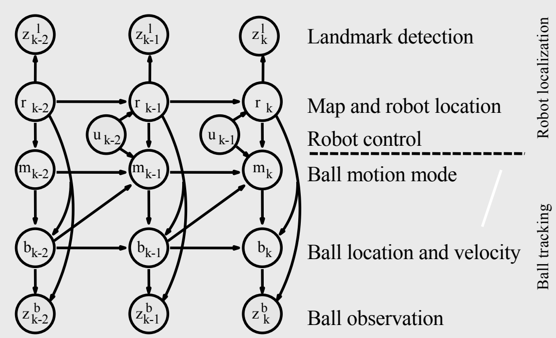
Grab Example (1)



Grab Example (2)



Inference: Posterior Estimation



$$\boxed{p(b_k, m_k, r_k \mid z^b_{1:k}, z^l_{1:k}, u_{1:k-1})}$$

Rao-Blackwellised PF for Inference

- Represent posterior by random samples
- Each sample

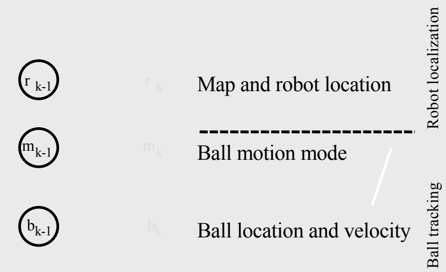
$$s_i = \langle r_i, m_i, b_i \rangle \quad = \quad \langle \langle x, y, \theta \rangle_i, m_i, \langle \mu, \Sigma \rangle_i \rangle$$

contains robot location, ball mode, ball Kalman filter

- Generate individual components of a particle stepwise using the factorization

$$p(b_k, m_{1:k}, r_{1:k} \mid z_{1:k}, u_{1:k-1}) =$$
$$p(b_k \mid m_{1:k}, r_{1:k}, z_{1:k}, u_{1:k-1})\, p(m_{1:k} \mid r_{1:k}, z_{1:k}, u_{1:k-1}) \cdot p(r_{1:k} \mid z_{1:k}, u_{1:k-1})$$

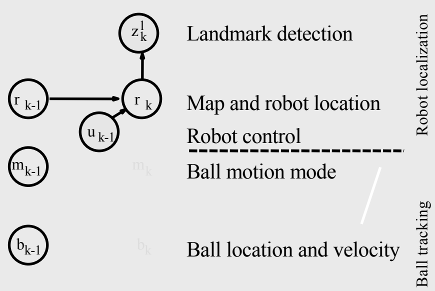
Rao-Blackwellised Particle Filter for Inference



- Draw a sample from the previous sample set:

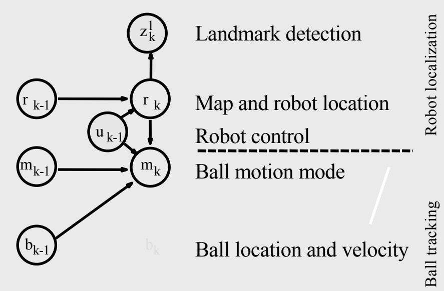
$$\langle r_{k-1}^{(i)}, m_{k-1}^{(i)}, b_{k-1}^{(i)} \rangle$$

Generate Robot Location



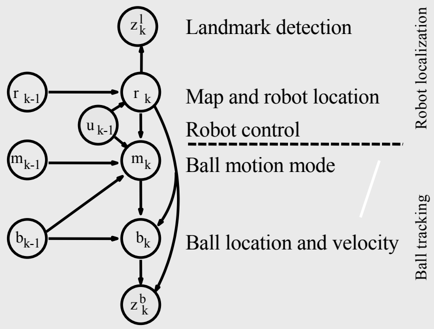
$$r_k^{(i)} \sim p(r_k \mid r_{k-1}^{(i)}, m_{k-1}^{(i)}, b_{k-1}^{(i)}, z_k, u_{k-1}) \quad \Rightarrow \quad \langle r_k^{(i)}, \_, \_ \rangle$$

Generate Ball Motion Model



$$m_k^{(i)} \sim p(m_k \mid r_k^{(i)}, m_{k-1}^{(i)}, b_{k-1}^{(i)}, z_k, u_{k-1}) \quad \Rightarrow \quad \langle r_k^{(i)}, m_k^{(i)}, \_ \rangle$$

Update Ball Location and Velocity



$$b_k^{(i)} \sim p(b_k \mid r_k^{(i)}, m_k^{(i)}, b_{k-1}^{(i)}, z_k) \;\Rightarrow\; \left\langle r_k^{(i)}, m_k^{(i)}, b_k^{(i)} \right\rangle$$

Importance Resampling

- Weight sample by

$$w_k^{(i)} \propto p(z_k^l \mid r_k^{(i)})$$

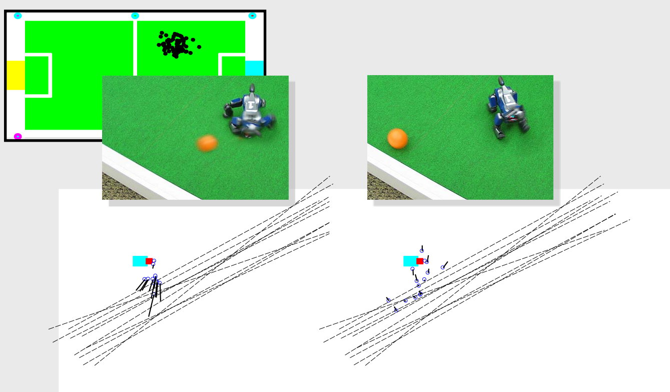
if observation is landmark detection and by

$$w_k^{(i)} \propto p(z_k^b \mid m_k^{(i)}, r_k^{(i)}, b_{k-1}^{(i)})$$
$$= \int p(z_k^b \mid m_k^{(i)}, r_k^{(i)}, b_k^{(i)}) p(b_k^{(i)} \mid m_k^{(i)}, r_k^{(i)}, b_{k-1}^{(i)}) \, \mathrm{d}b_k$$

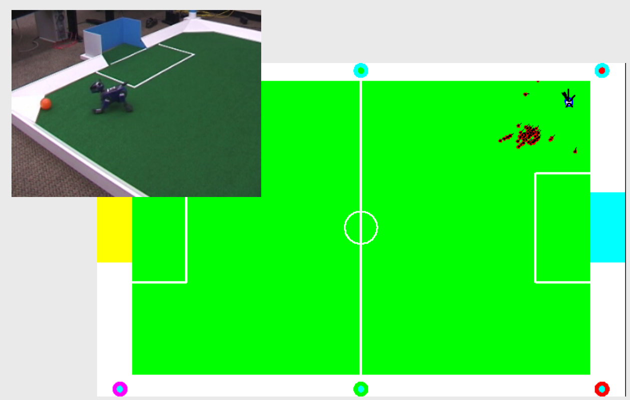
if observation is ball detection.

- Resample

Ball-Environment Interaction

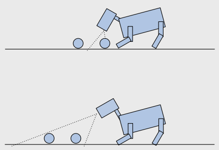


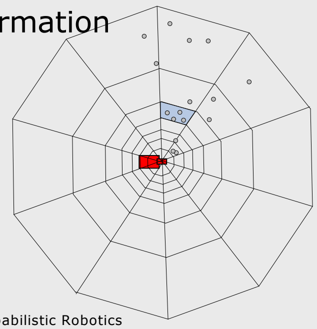
Ball-Environment Interaction



Tracking and Finding the Ball

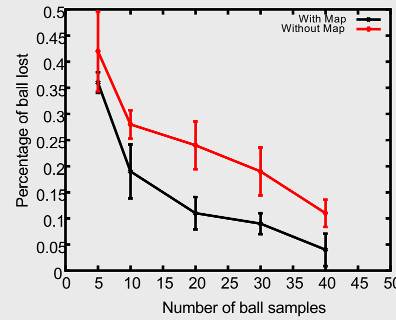
- Cluster ball samples by discretizing pan / tilt angles
- Uses negative information



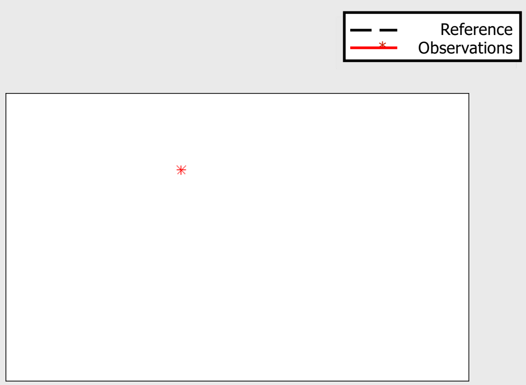


Experiment: Real Robot

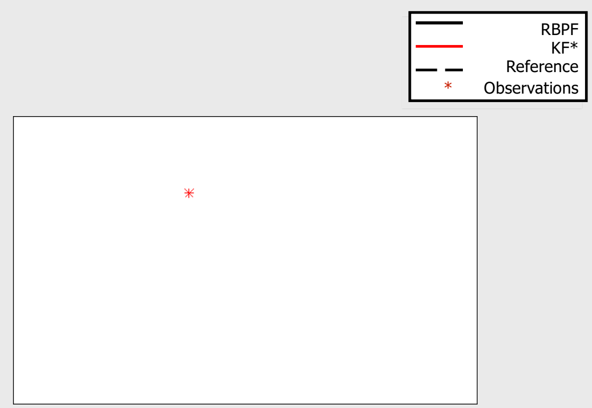
- Robot kicks ball 100 times, tries to find it afterwards
- Finds ball in 1.5 seconds on average



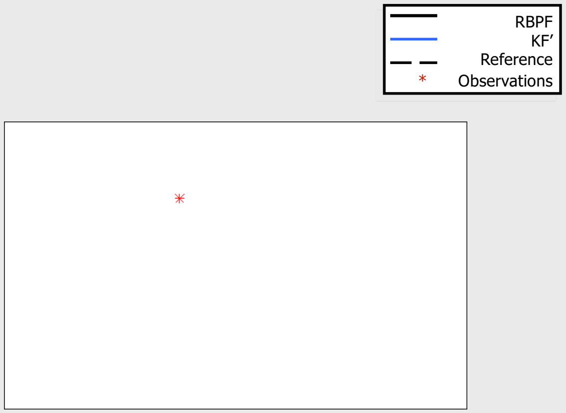
Simulation Runs



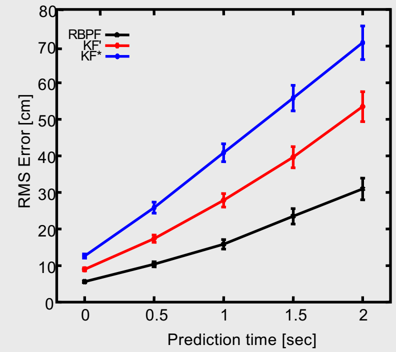
Comparison to KF* (optimized for straight motion)



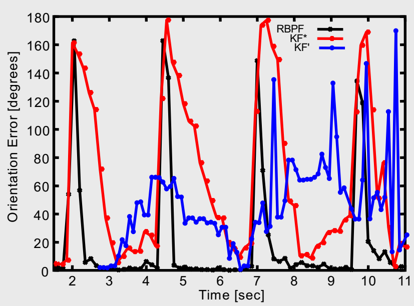
Comparison to KF’ (inflated prediction noise)



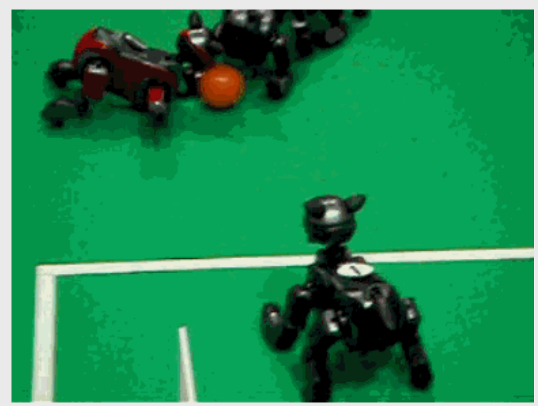
Error vs. Prediction Time



Orientation Errors



Goalie



Discussion

- Particle filters are intuitive and simple
  - Support point-wise thinking (reduced uncertainty)
  - Good for test implementation if system behavior is not well known
- Inefficient compared to Kalman filter
- Rao-Blackwellization
  - Only sample discrete / highly non-linear parts of state space
  - Solve remaining part analytically (KF,discrete)