Mapping with limited sensing

Kris Beevers

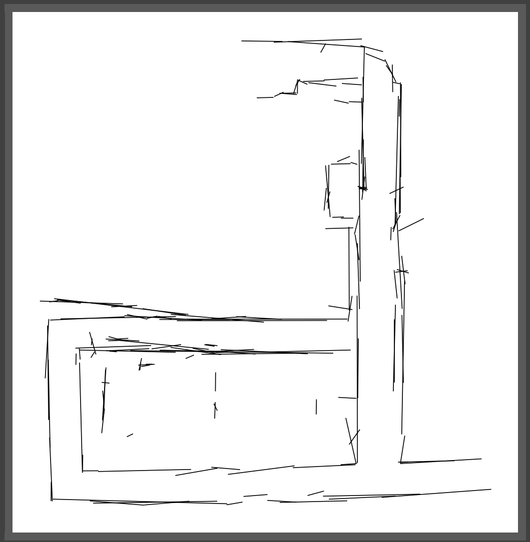
Department of Computer Science
Rensselaer Polytechnic Institute

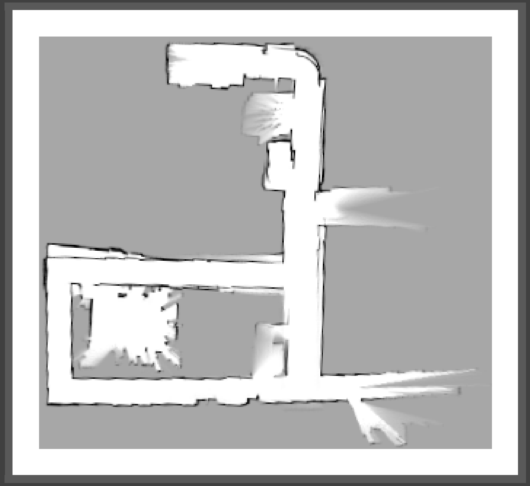
January 15, 2007

Robot mapping

Basic problem: build a “map” using a robot’s sensors

- Context for this thesis:
  - 2D static environments
  - Passive mapping
  - **Low-fidelity range-bearing sensors**
- Some issues in designing algorithms:
  - Environment model (**landmarks**, **occupancy**, topological)
  - Feature extraction and data association
  - Managing and reducing uncertainty
  - Computational feasibility





Thesis contributions

- An **analysis of mapping sensors** and **bounds on map error** for a simple range-bearing sensor model
- A Rao-Blackwellized particle filtering (RBPF) algorithm for **simultaneous localization and mapping (SLAM) with sparse sensing**
- Techniques for **incorporating prior information** in RBPF SLAM
- Two new **sampling strategies** for RBPF SLAM
- An **implementation of RBPF on a 16 MHz microcontroller**
- Full **software implementations** of all the algorithms in the thesis (and other standard algorithms)

Thesis contributions

- An **analysis of mapping sensors** and **bounds on map error** for a simple range-bearing sensor model

- A Rao-Blackwellized particle filtering (RBPF) algorithm for **simultaneous localization and mapping (SLAM) with sparse sensing**

- Techniques for **incorporating prior information** in RBPF SLAM

- Two new **sampling strategies** for RBPF SLAM

- An **implementation of RBPF on a 16 MHz microcontroller**

- Full **software implementations** of all the algorithms in the thesis (and other standard algorithms)

Sensing and map quality

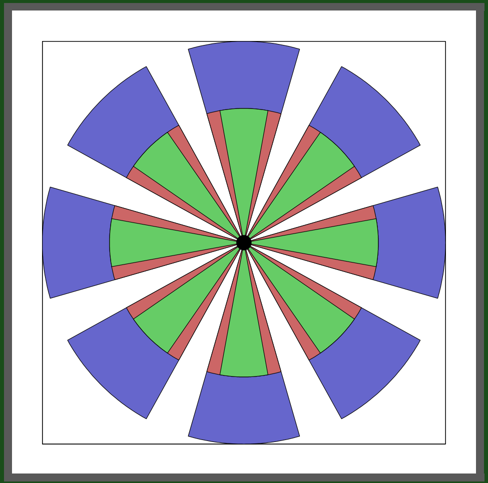
- **Question:** how can we relate "sensing capabilities" to map quality?
- **Related work**: for every kind of sensor, either **design a specific algorithm or prove no algorithm exists** (localization, O'Kane and LaValle, 2006):
  - Binary characterization (**can or can't** localize)
  - Compass + contact sensor: **can** localize
  - Angular odometer + contact sensor: **can't** localize
- An alternative approach: **fix** the mapping algorithm and define a broad sensor model
  - Encompasses many types of practical mapping sensors
  - Characterize **which sensors can build a map**
  - Give **quality bounds** on the map for a given sensor

Models and mapping algorithm

- **Environment**:
  - Occupancy grid model
  - Cells *independently occupied* with a given probability (“density”)

- **Motion assumption**: poses drawn uniformly at random

- **Sensor**:
  - Ring of beams of non-zero beam width
  - Bounded uncertainty model
  - Beam reports range to first cell *detected* as occupied
  - Model incorporates false negatives/positives



- **Mapping algorithm**:
  - **Increase** occupancy belief for cells **at** reported range ($\pm$ error)
  - **Decrease** occupancy belief for closer cells

Bound on expected map error

- **Error:** $\nu = \sum_{ij} \nu_{ij}$
  - $\nu_{ij} = 1$ if the ML estimate for cell m_{ij} is **incorrect**; $\nu_{ij} = 0$ otherwise
- **Chernoff bound:**

$$\boxed{E[\nu] \leq M^2 \exp\left\{-2E[o_{ab}]\left(\frac{1}{2} - p_{\text{inc}}\right)^2\right\}}$$

- $E[o_{ab}]$: expected number of updates of any cell

$$\boxed{E[o_{ab}] \geq \frac{2TF\rho(\Delta_\beta + \sigma_\beta)}{M^2} \sum_{\tau=0}^{\left\lceil \frac{r^+ + \sigma_r}{\delta} \right\rceil} \tau \cdot \mathcal{E}_{\mathrm{E}}^{\Delta_\beta \tau^2}}$$

- p_{inc}: probability that an update of a cell is incorrect
  - Mainly related to range/bearing uncertainties, false pos./neg. rates

Bound on expected map error

- **Error:** $\nu = \sum_{ij} \nu_{ij}$
  - $\nu_{ij} = 1$ if the ML estimate for cell m_{ij} is **incorrect**; $\nu_{ij} = 0$ otherwise
- **Chernoff bound:**

$$\boxed{E[\nu] \leq M^2 \exp\left\{-2E[o_{ab}]\left(\frac{1}{2} - p_{\text{inc}}\right)^2\right\}}$$

- $E[o_{ab}]$: expected number of updates of any cell

$$\boxed{E[o_{ab}] \geq \frac{2TF\rho(\Delta_\beta+\sigma_\beta)}{M^2} \sum_{\tau=0}^{\left\lceil \frac{r^+ + \sigma_r}{\delta} \right\rceil} \tau \cdot \mathcal{E}_{\mathrm{E}}^{\Delta_\beta \tau^2}}$$

- p_{inc}: probability that an update of a cell is incorrect
  - Mainly related to range/bearing uncertainties, false pos./neg. rates

Bound on expected map error

- **Error:** $\nu = \sum_{ij} \nu_{ij}$
  - $\nu_{ij} = 1$ if the ML estimate for cell m_{ij} is **incorrect**; $\nu_{ij} = 0$ otherwise
- **Chernoff bound:**

$$\boxed{E[\nu] \leq M^2 \exp\left\{-2E[o_{ab}]\left(\frac{1}{2} - p_{\mathrm{inc}}\right)^2\right\}}$$

- $E[o_{ab}]$: expected number of updates of any cell

$$\boxed{E[o_{ab}] \geq \frac{2TF\rho(\Delta_\beta + \sigma_\beta)}{M^2} \sum_{\tau=0}^{\left\lceil \frac{r^+ + \sigma_r}{\delta} \right\rceil} \tau \cdot \mathcal{E}_{\mathrm{E}}^{\Delta_\beta \tau^2}}$$

- p_{inc}: probability that an update of a cell is incorrect
  - Mainly related to range/bearing uncertainties, false pos./neg. rates

Bound on expected map error

- **Error:** $\nu = \sum_{ij} \nu_{ij}$
  - $\nu_{ij} = 1$ if the ML estimate for cell m_{ij} is **incorrect**; $\nu_{ij} = 0$ otherwise
- **Chernoff bound:**

$$\boxed{E[\nu] \leq M^2 \exp\left\{-2E[o_{ab}]\left(\frac{1}{2} - p_{\text{inc}}\right)^2\right\}}$$

- $E[o_{ab}]$: expected number of updates of any cell

$$\boxed{E[o_{ab}] \geq \frac{2TF\rho(\Delta_\beta+\sigma_\beta)}{M^2} \sum_{\tau=0}^{\left\lceil \frac{r^+ + \sigma_r}{\delta} \right\rceil} \tau \cdot \mathcal{E}_{\mathrm{E}}^{\Delta_\beta \tau^2}}$$

- p_{inc}: probability that an update of a cell is incorrect
  - Mainly related to range/bearing uncertainties, false pos./neg. rates

Bound on expected map error

- **Error:** $\nu = \sum_{ij} \nu_{ij}$
  - $\nu_{ij} = 1$ if the ML estimate for cell m_{ij} is **incorrect**; $\nu_{ij} = 0$ otherwise
- **Chernoff bound:**

$$\boxed{E[\nu] \leq M^2 \exp\left\{-2E[o_{ab}]\left(\frac{1}{2} - p_{\text{inc}}\right)^2\right\}}$$

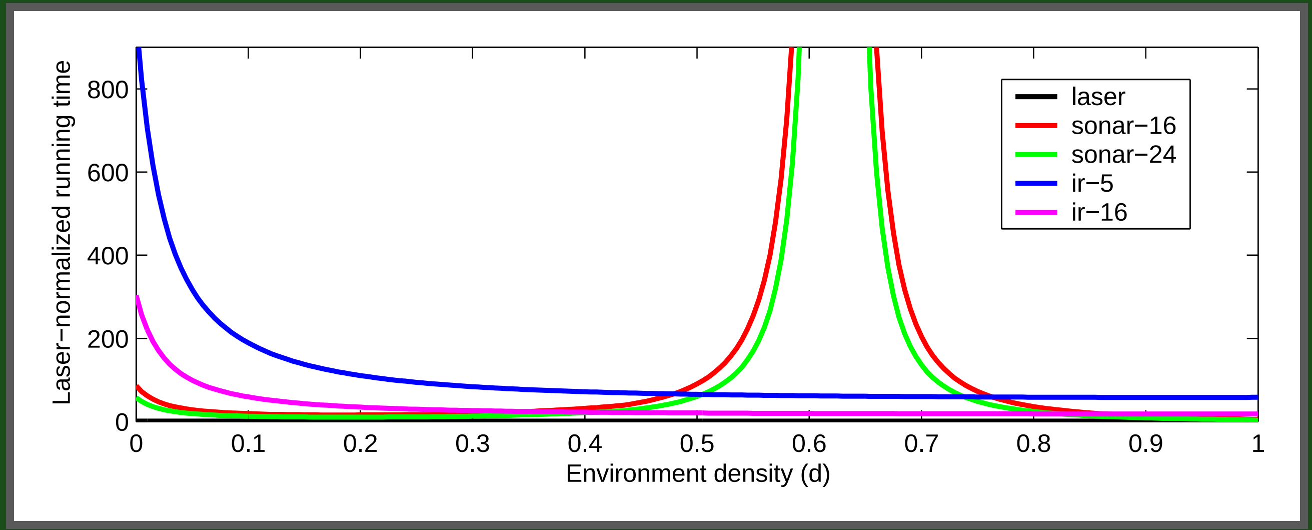
- $E[o_{ab}]$: expected number of updates of any cell

$$\boxed{E[o_{ab}] \geq \frac{2TF\rho(\Delta_\beta + \sigma_\beta)}{M^2} \sum_{\tau=0}^{\left\lceil \frac{r^+ + \sigma_r}{\delta} \right\rceil} \tau \cdot \mathcal{E}_{\mathrm{E}}^{\Delta_\beta \tau^2}}$$

- p_{inc}: probability that an update of a cell is incorrect
  - Mainly related to range/bearing uncertainties, false pos./neg. rates

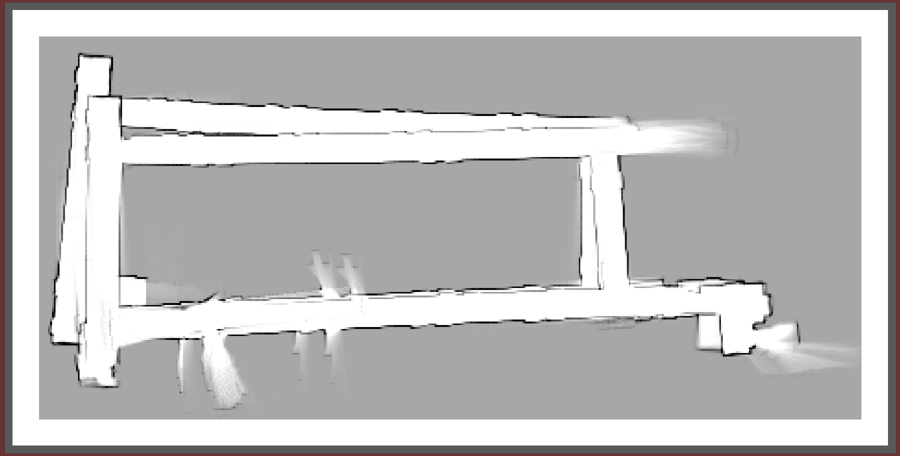
Application: comparing real sensors

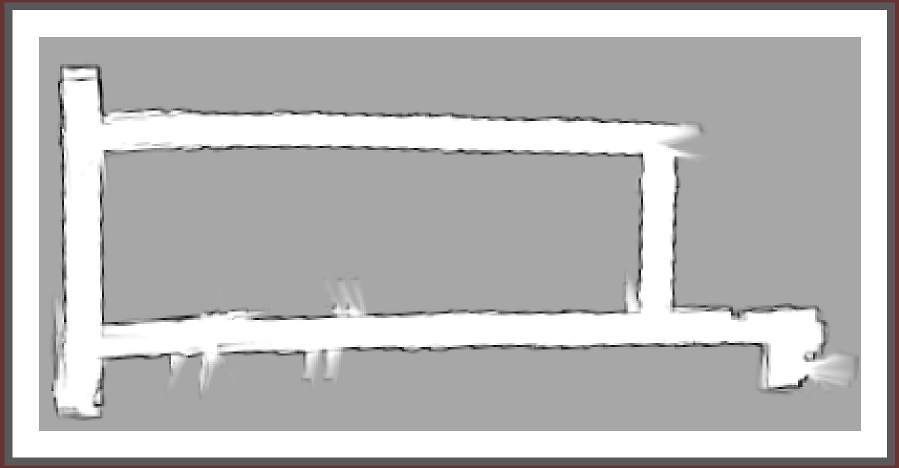
- We obtained model parameters for three real sensors used in mapping:
  - SICK LMS 200-30106 scanning laser rangefinder
  - Polaroid 6500 series SONAR ranging module
  - Sharp GPD12 infrared rangefinder
- **“Laser-normalized” running time**
  - Extra work (time) required for a sensor to build a map of (expected) quality equivalent to that built by the scanning laser rangefinder
  - Depends only on sensor characteristics and environment density



Simultaneous localization and mapping (SLAM)

- **Odometry** is notoriously noisy!
  - Cannot simply build map based on odometry-estimated trajectory
  - GPS is often not available (e.g., indoors)
- SLAM: Alternate mapping and localization steps:
  1. Use sensor returns to improve pose estimate **based on current map**
  2. Update the map with the sensor returns

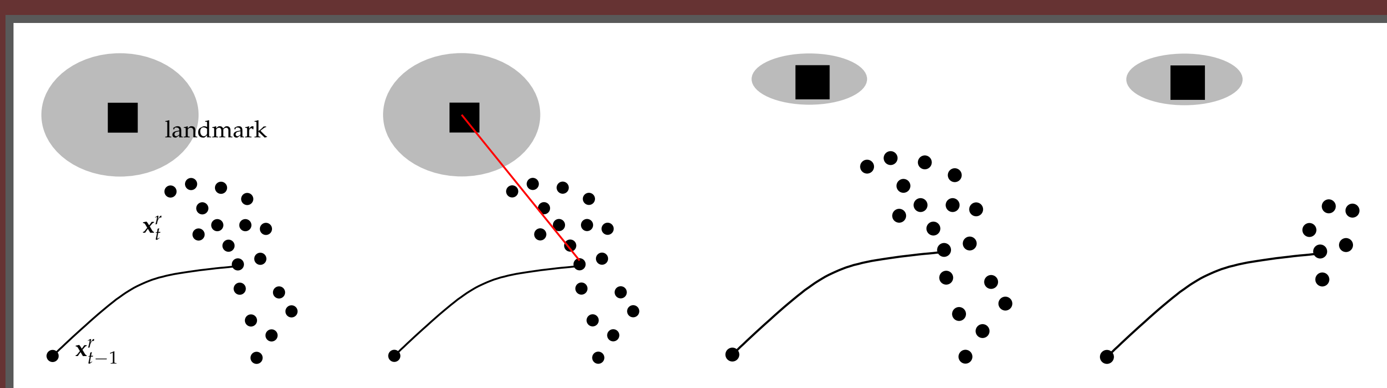


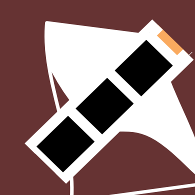


$$\underbrace{p(\mathbf{x}_t|\mathbf{u}_{1:t},\mathbf{z}_{1:t},\mathbf{n}_{1:t})}_{\text{posterior}} = \eta \underbrace{p(\mathbf{z}_t|\mathbf{x}_t,\mathbf{n}_t)}_{\text{measurement}} \int \underbrace{p(\mathbf{x}_t|\mathbf{x}_{t-1},\mathbf{u}_t)}_{\text{motion}} \underbrace{p(\mathbf{x}_{t-1}|\mathbf{u}_{1:t-1},\mathbf{z}_{1:t-1},\mathbf{n}_{1:t-1})}_{\text{prior}} \, d\mathbf{x}_{t-1}$$

Landmark based particle filtering SLAM

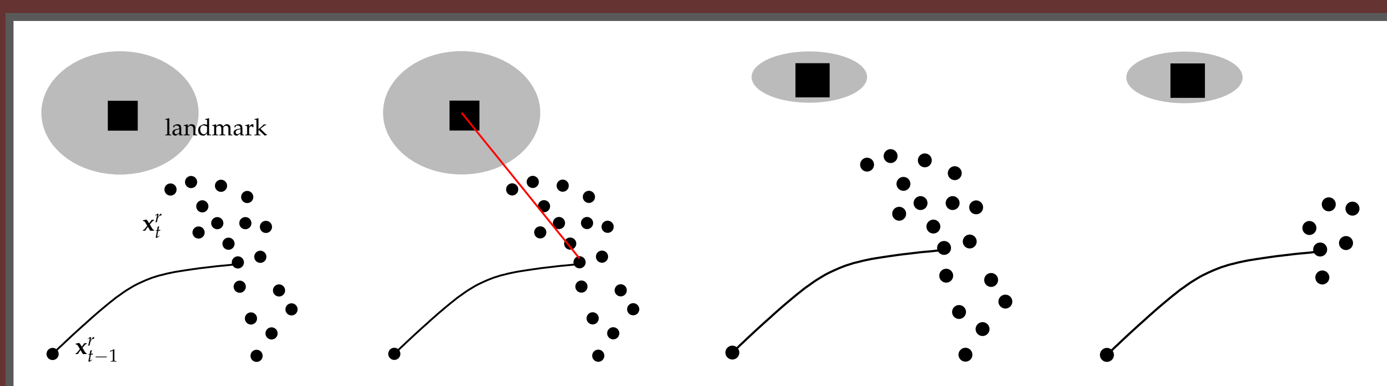
loop
  Move / sense / extract features
  for all particles ϕ^i **do**
    Project forward: $\mathbf{x}_t^{r,i} \sim p(\mathbf{x}_t^r | \mathbf{x}_{t-1}^{r,i}, \mathbf{u}_t)$
    Do data association (compute $\mathbf{n}_t^i$), update map
    Compute weight: $\omega_t^i = \omega_{t-1}^i \times p(\mathbf{z}_t | \mathbf{x}_t^{r,i}, \mathbf{x}^{m,i}, \mathbf{n}_t^i)$
  end for
  Resample (with replacement) according to ω_t^is
end loop

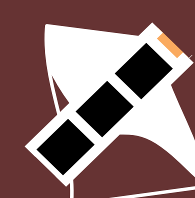




Landmark based particle filtering SLAM

loop
  Move / sense / extract features
  for all particles ϕ^i **do**
    Project forward: $\mathbf{x}_t^{r,i} \sim p(\mathbf{x}_t^r | \mathbf{x}_{t-1}^{r,i}, \mathbf{u}_t)$
    Do data association (compute $\mathbf{n}_t^i$), update map
    Compute weight: $\omega_t^i = \omega_{t-1}^i \times p(\mathbf{z}_t | \mathbf{x}_t^{r,i}, \mathbf{x}^{m,i}, \mathbf{n}_t^i)$
  end for
  Resample (with replacement) according to ω_t^is
end loop

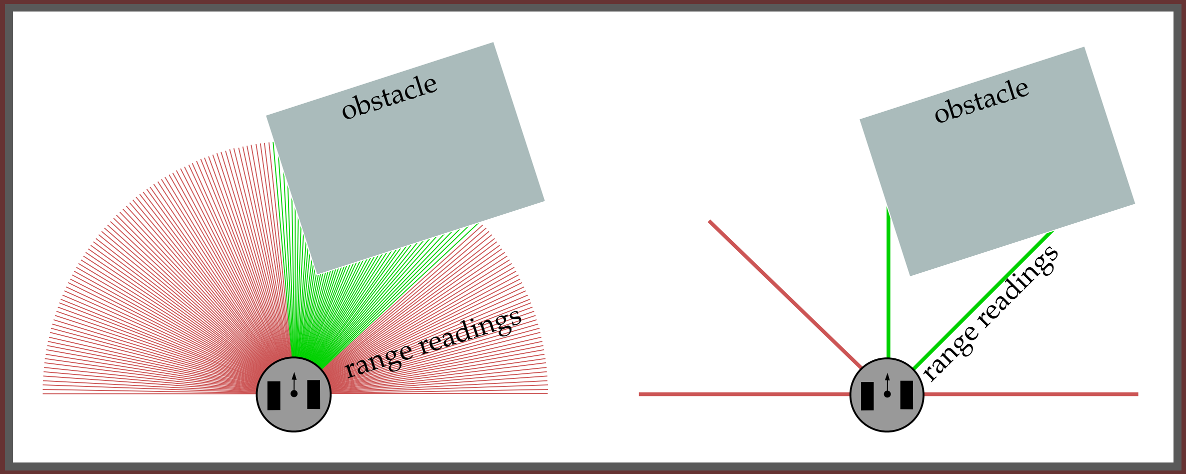




Thesis contributions

- An **analysis of mapping sensors** and **bounds on map error** for a simple range-bearing sensor model
- A Rao-Blackwellized particle filtering (RBPF) algorithm for **simultaneous localization and mapping (SLAM) with sparse sensing**
- Techniques for **incorporating prior information** in RBPF SLAM
- Two new **sampling strategies** for RBPF SLAM
- An **implementation of RBPF on a 16 MHz microcontroller**
- Full **software implementations** of all the algorithms in the thesis (and other standard algorithms)

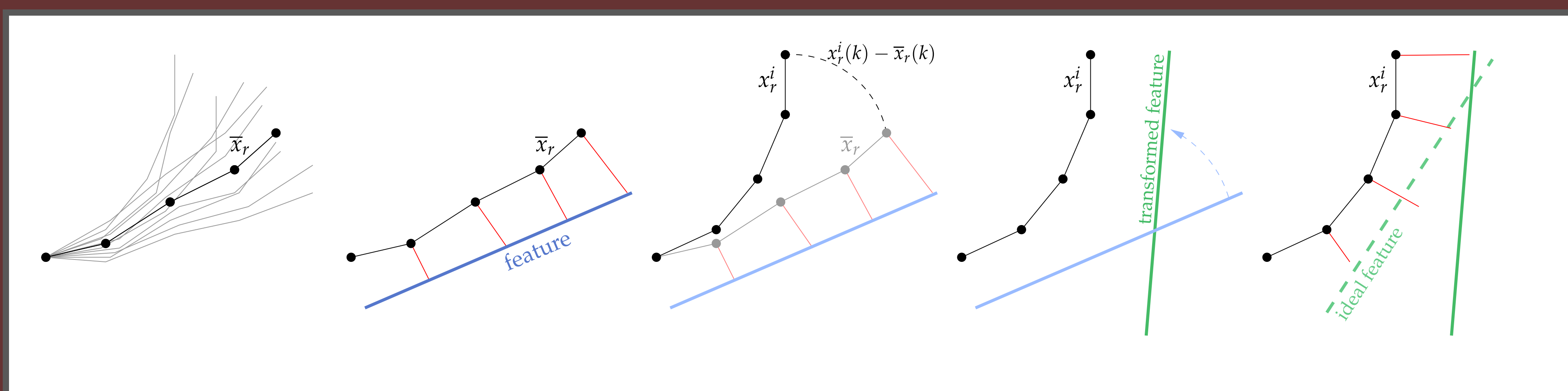
RBPF SLAM with sparse sensing



- **Related work**: partial observability
  - Bearing-only SLAM (cameras) — accurate data association
  - Leonard et al. (2002) — EKF, SONAR, stores recent trajectory in state
- **Approach**: extract features using *multiscans*
  - Data from multiple poses
  - Feature extraction is conditioned on trajectory
  - Naïve RBPF implementation: **per-particle feature extraction**

SLAM with multiscans

loop
  for m time steps: move and collect sparse scans
  Extract features with *multiscan* data from last m steps
  for all particles ϕ^i **do**
    for $k=t-m+1$ to t: project pose forward: $\mathbf{x}_k^{r,i} \sim p(\mathbf{x}_k^r | \mathbf{x}_{k-1}^{r,i}, \mathbf{u}_k)$
    Do data association, update map
    Compute weight ω_t^i
  end for
  Resample (with replacement) according to ω_t^is
end loop



SLAM with multiscans

loop

  for m time steps: move and collect sparse scans

  Extract features with *multiscan* data from last m steps

  for all particles ϕ^i **do**

    for $k = t - m + 1$ to t: project pose forward: $\mathbf{x}_k^{r,i} \sim p(\mathbf{x}_k^r | \mathbf{x}_{k-1}^{r,i}, \mathbf{u}_k)$

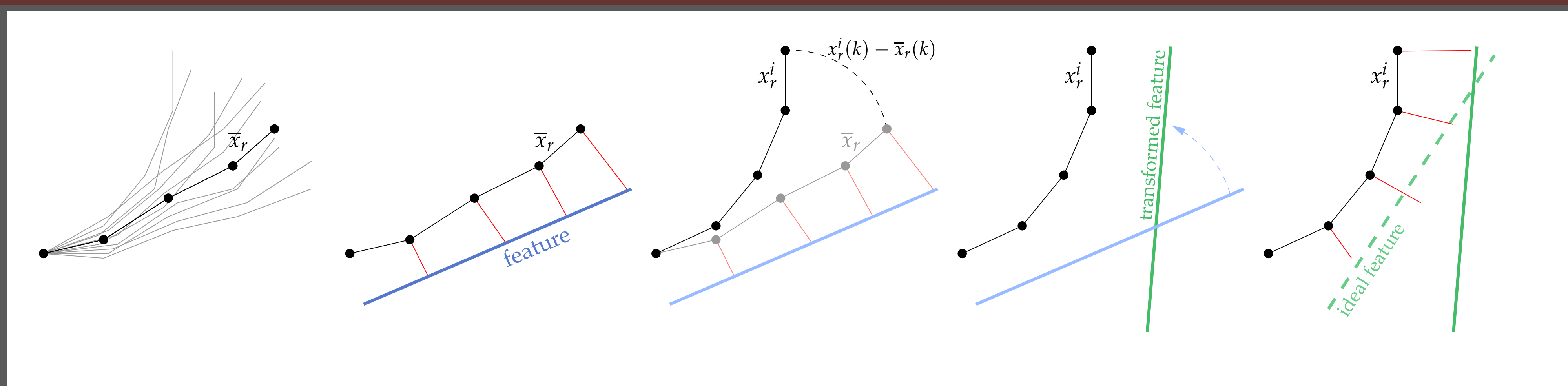
    Do data association, update map

    Compute weight ω_t^i

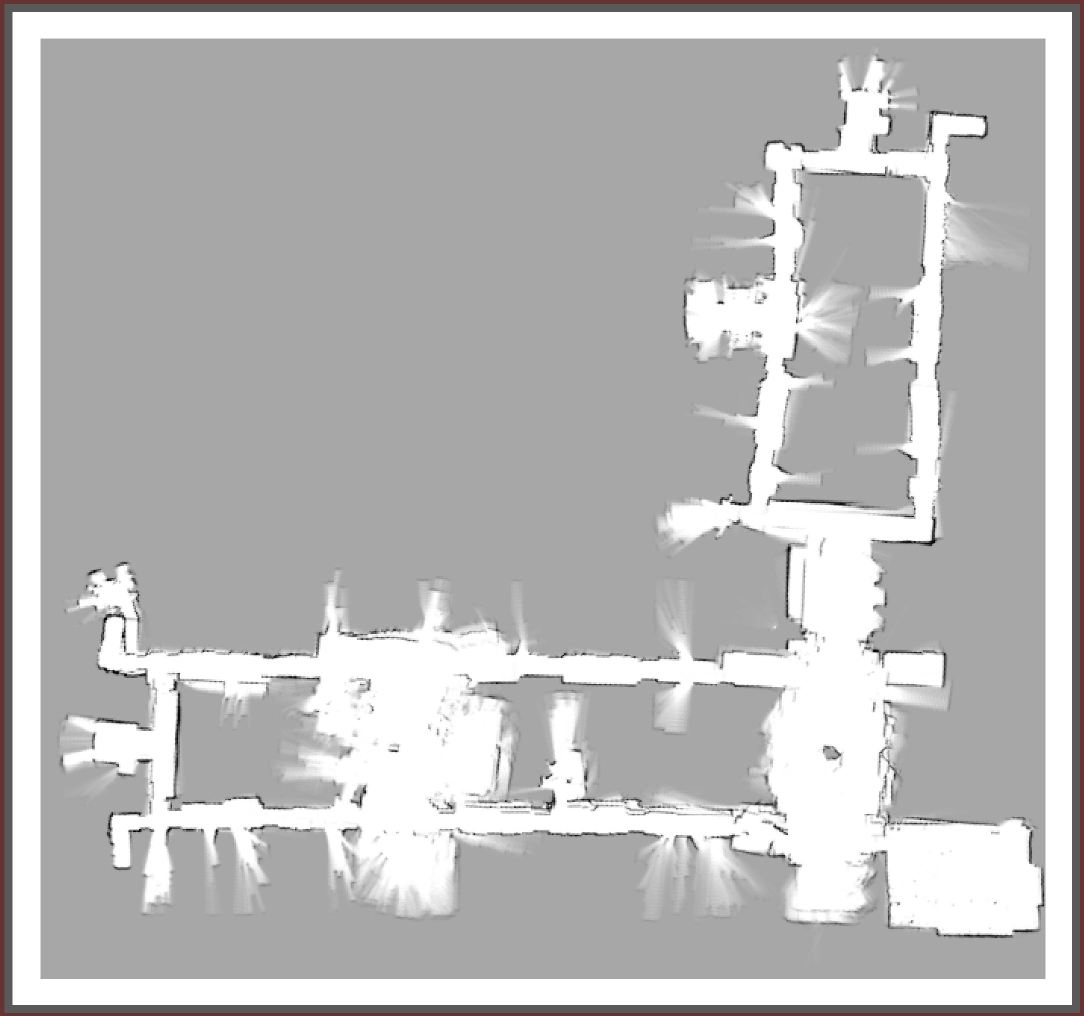
  end for

  Resample (with replacement) according to ω_t^is

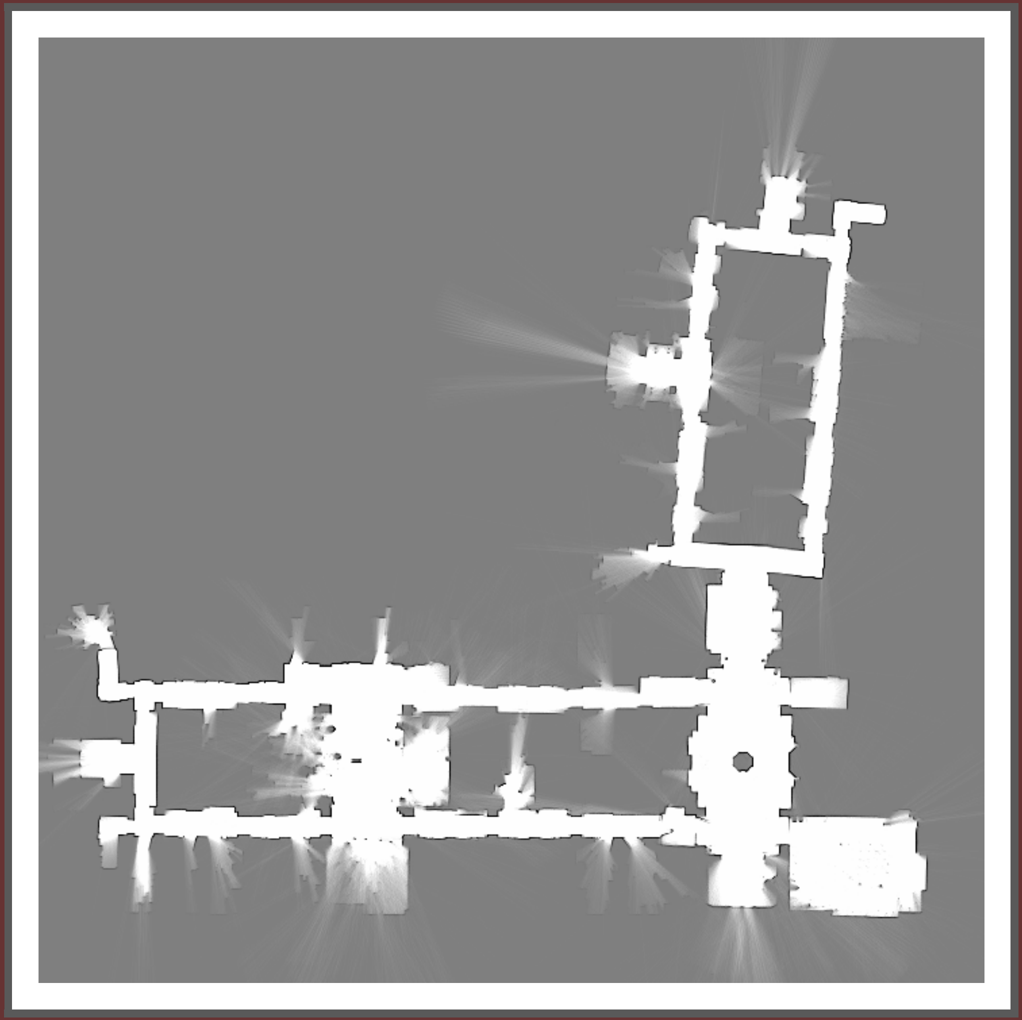
end loop



Multiscan SLAM result: Stanford



Multiscan SLAM



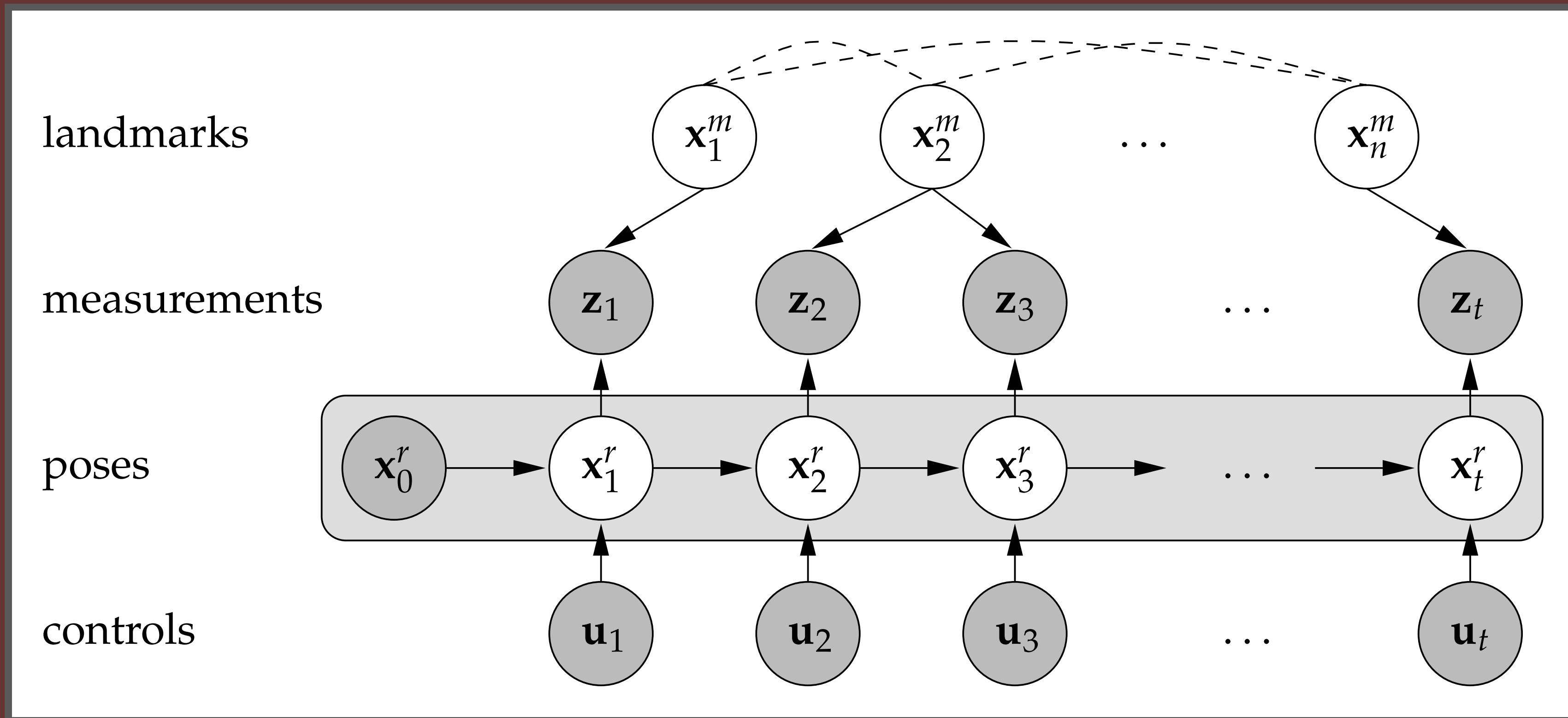
Scan-matching SLAM

Scan-matching result courtesy of Brian Gerkey.

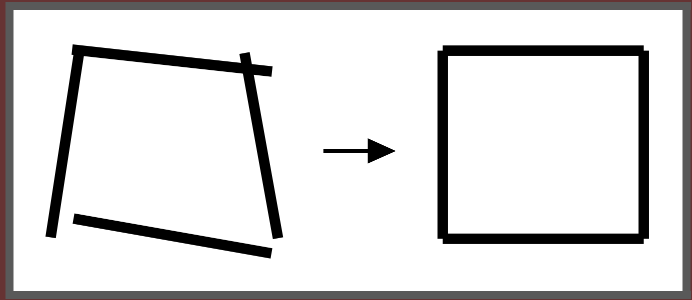
Thesis contributions

- An **analysis of mapping sensors** and **bounds on map error** for a simple range-bearing sensor model
- A Rao-Blackwellized particle filtering (RBPF) algorithm for **simultaneous localization and mapping (SLAM) with sparse sensing**
- Techniques for **incorporating prior information** in RBPF SLAM
- Two new **sampling strategies** for RBPF SLAM
- An **implementation of RBPF on a 16 MHz microcontroller**
- Full **software implementations** of all the algorithms in the thesis (and other standard algorithms)

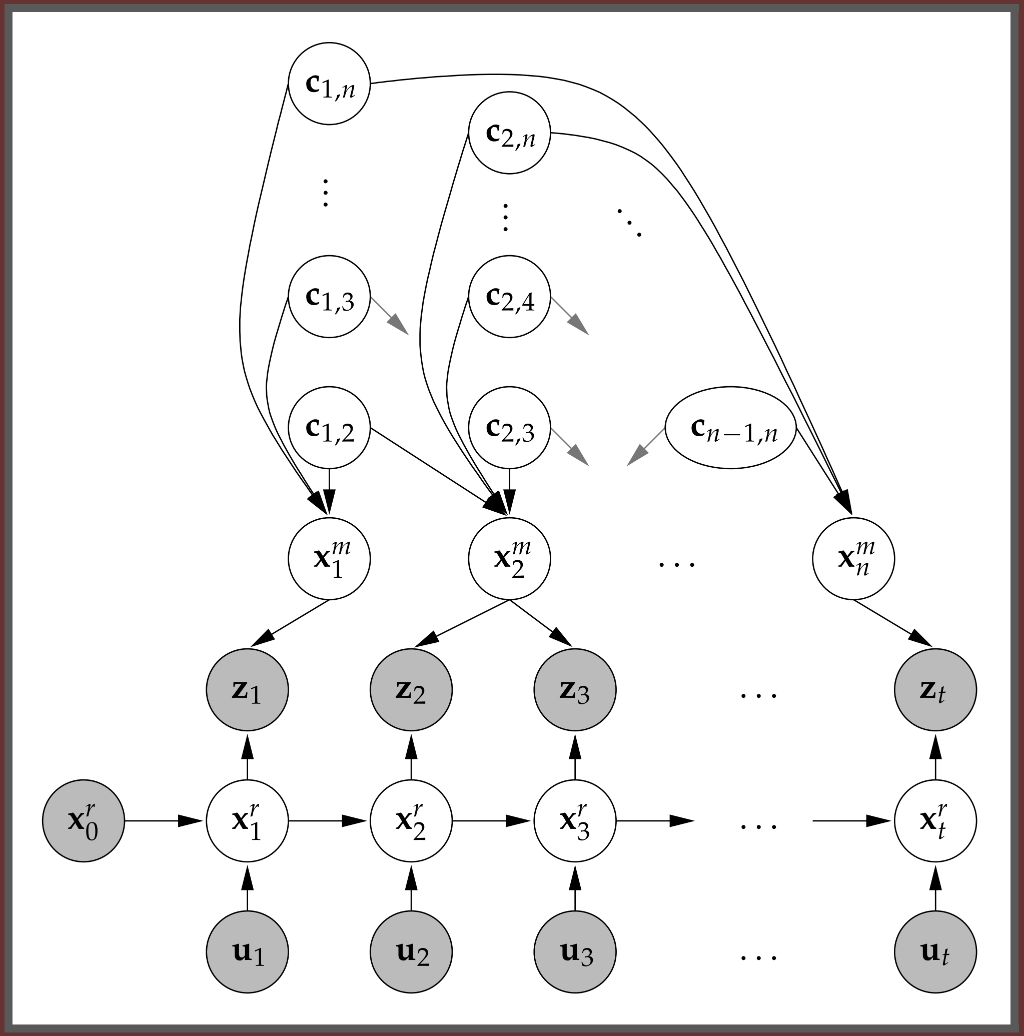
Prior knowledge in SLAM



- Typical RBPF SLAM algorithms ignore **environment structure**
- Often, measurements of a landmark inform you about other landmarks
- Example: rectilinearity



Prior information as pairwise constraints



- Write landmark relationships as *pairwise relative constraints*
- Use prior knowledge to do inference on constraint parameters (in particle filter)
- Enforce constraints separately for each particle

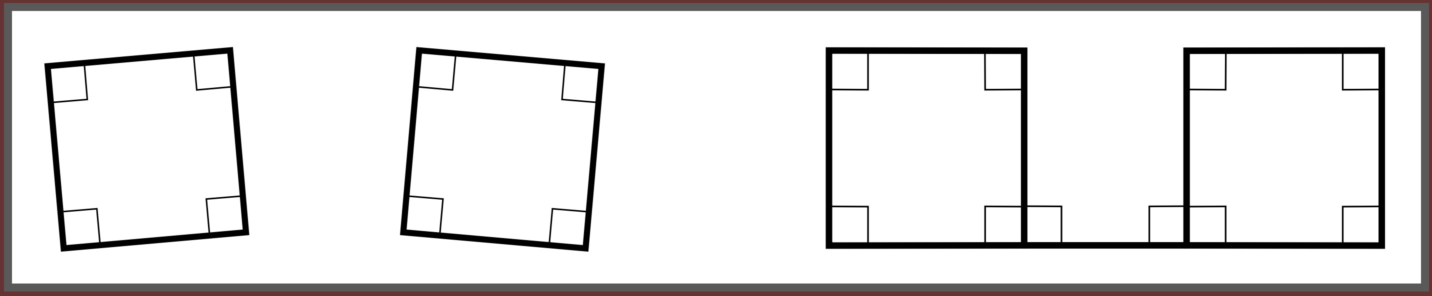
Related work — constrained SLAM

- **Constraint inference**: Rodriguez-Losada et al. (2006) — thresholding
- Enforcing *a priori* known constraints in EKF estimation:
  - Durrant-Whyte (1988); Smith et al. (1990); Wen and Durrant-Whyte (1992): **constraints as zero-uncertainty measurements**
  - Csorba and Durrant-Whyte (1997); Newman (1999); Deans and Hebert (2000b): relative maps, constraints enforce map consistency
  - Simon and Chia (2002); Simon and Simon (2003): **project unconstrained state onto constraint surface**
- Our work: **first to enforce constraints in particle filtering SLAM**

Rao-Blackwellized constraint filter — overview

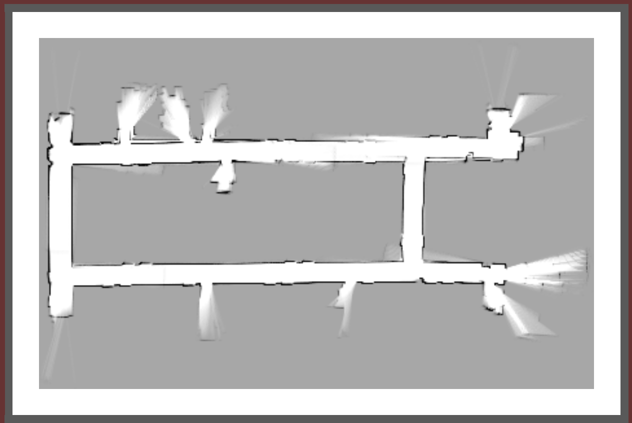
Landmark initialization — **for each particle**:

1: **Inference**: should new landmark be constrained with respect to any other landmarks?
2: If so, create a **superlandmark** with the new landmark and all constrained landmarks
3: Compute max. likelihood constrained parameter values
4: **Condition** unconstrained parameters on ML values of constrained parameters

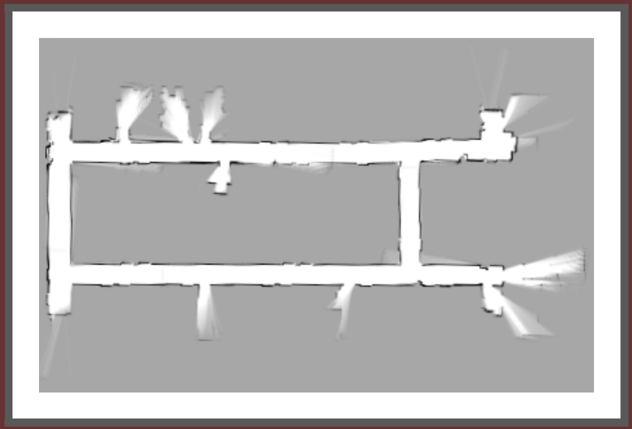


- Running time: asymptotically same as standard RBPF (linear in N)

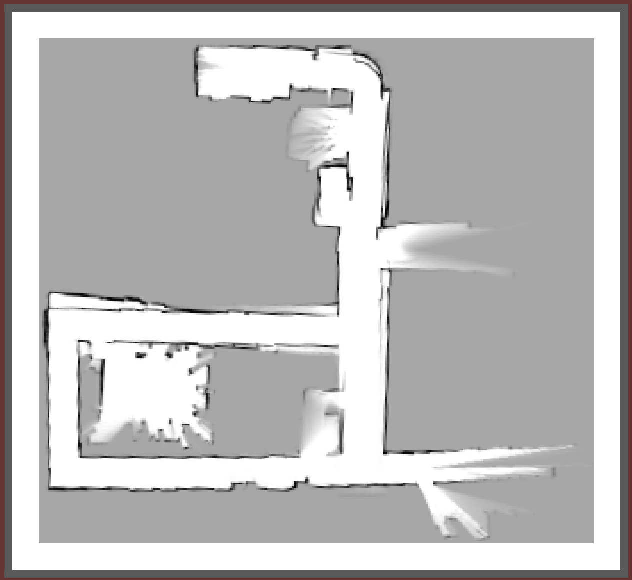
Real-world results: rectilinearity



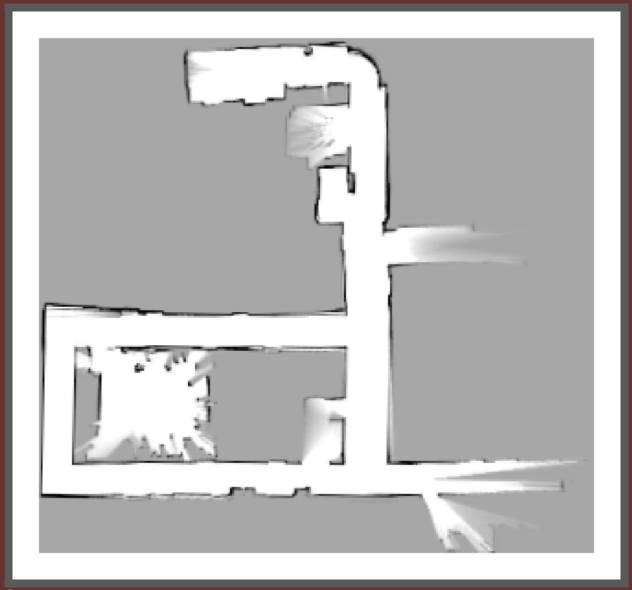
Unconstrained: 100 particles



Constrained: 20 particles



Unconstrained: 600 particles



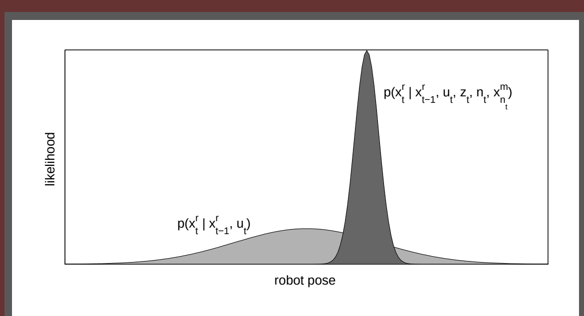
Constrained: 40 particles

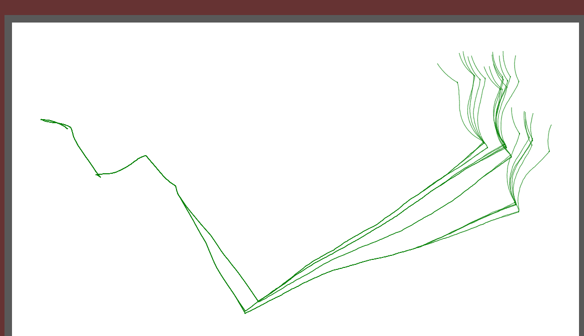
Thesis contributions

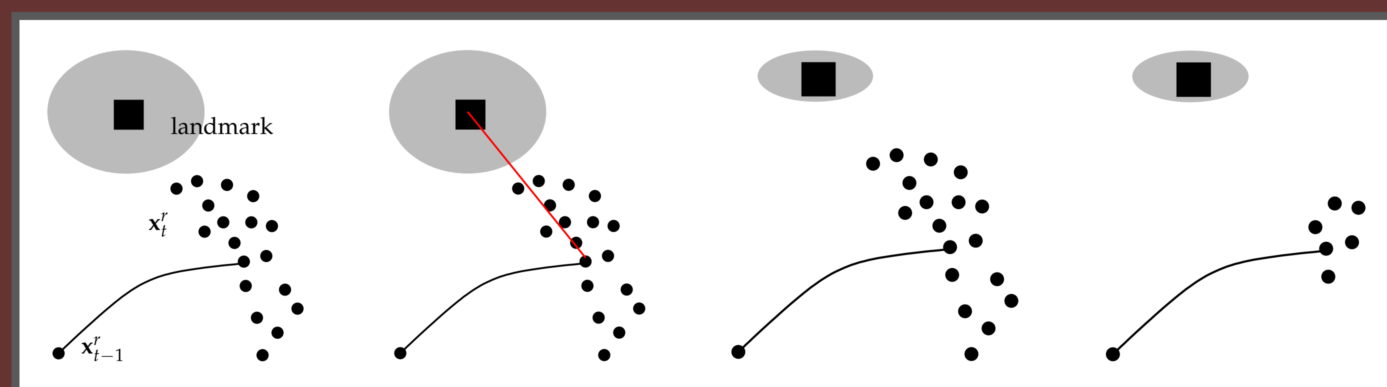
- An **analysis of mapping sensors** and **bounds on map error** for a simple range-bearing sensor model
- A Rao-Blackwellized particle filtering (RBPF) algorithm for **simultaneous localization and mapping (SLAM) with sparse sensing**
- Techniques for **incorporating prior information** in RBPF SLAM
- Two new **sampling strategies** for RBPF SLAM
- An **implementation of RBPF on a 16 MHz microcontroller**
- Full **software implementations** of all the algorithms in the thesis (and other standard algorithms)

Estimation consistency in RBPF SLAM

- A SLAM filter is **inconsistent** if it significantly **underestimates pose and map uncertainty**
- Inconsistency in a particle filter:
  - Occurs when samples poorly represent the target distribution
  - Bailey et al. (2006): RBPF SLAM algorithms are inconsistent in general
  - Due mainly to frequent resampling and poor proposal distributions

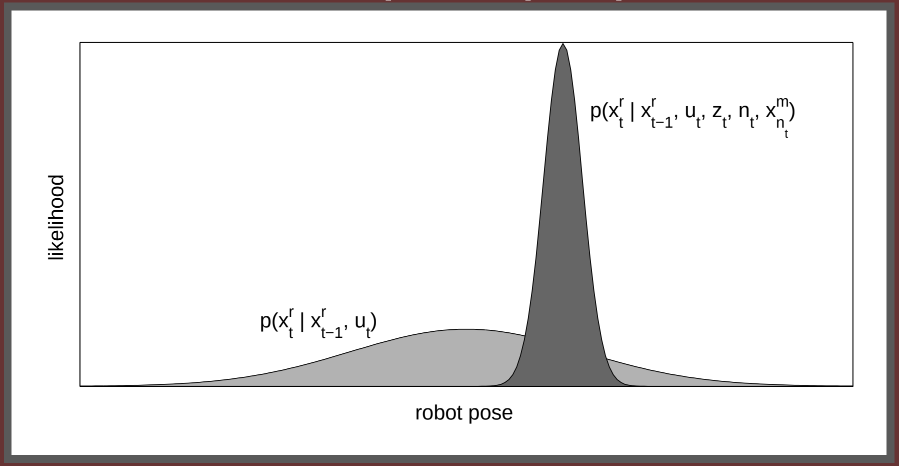






Related work — improving RBPF consistency

- **Improved proposal**: Montemerlo (2003); Grisetti et al. (2005)
  - "FastSLAM 2"
  - Use current measurement to compute proposal distribution



- **Effective sample size**: Liu and Chen (1995); Grisetti et al. (2005)
  - Only resample when particle weights are highly skewed
- **Backup state**: Stachniss et al. (2005)
  - Detect entry into a loop and store current particle set
  - After traversing loop, restore saved particle set to recover diversity

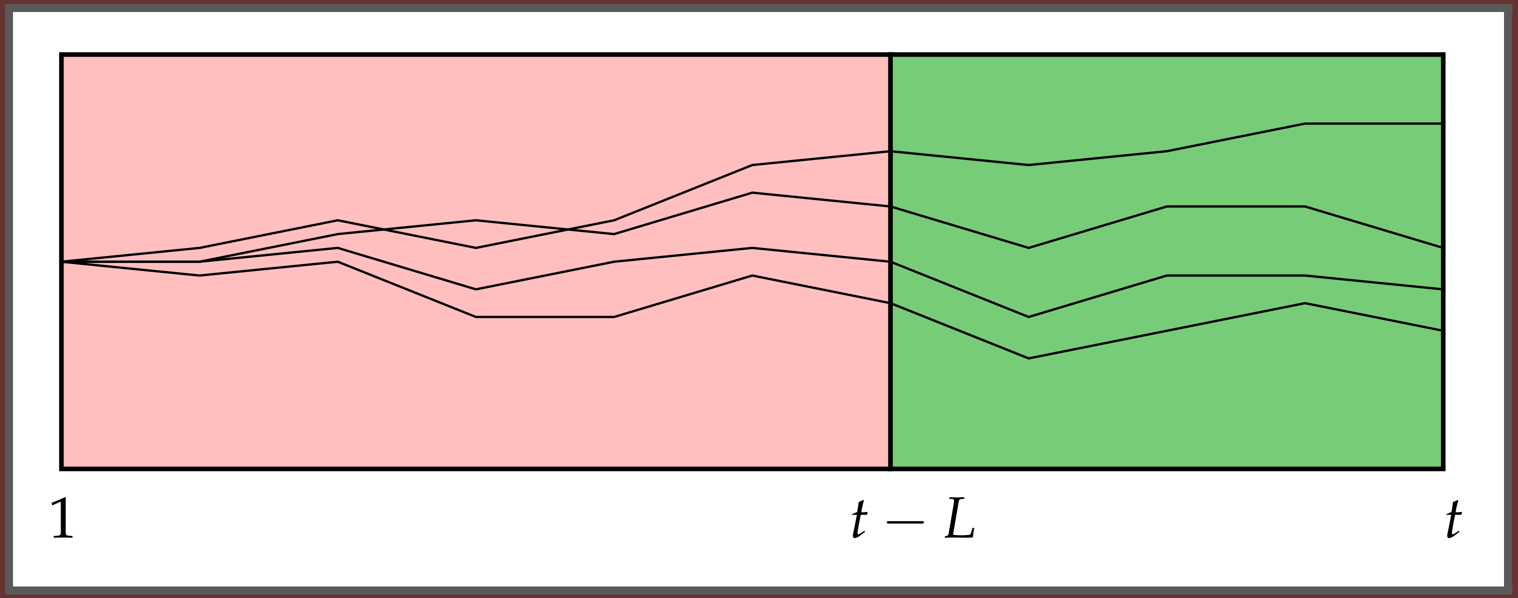
Updating the pose history

- **Key idea**: a new measurement tells you something about the *pose history*, not just the current pose
- Can't update the entire pose history (computationally infeasible)
- But we *can* draw new pose samples over a **fixed lag time**
  - Draw new samples for $\mathbf{x}^r_{t-L+1:t}$
  - Update maps from $t - L$ conditioned on new samples
- **Contribution** — two new techniques for RBPF SLAM:
  - **Fixed-lag roughening**: MCMC moves of pose samples over fixed lag
  - **Block proposal**: optimal joint distribution for poses over the lag time
  - Efficient; main implementation difficulty is extra book-keeping

Fixed-lag roughening

- After resampling, apply an MCMC move step to $\{\mathbf{x}^{r,i}_{t-L+1:t}\}$
- Fixed-lag Gibbs sampler for RBPF SLAM:

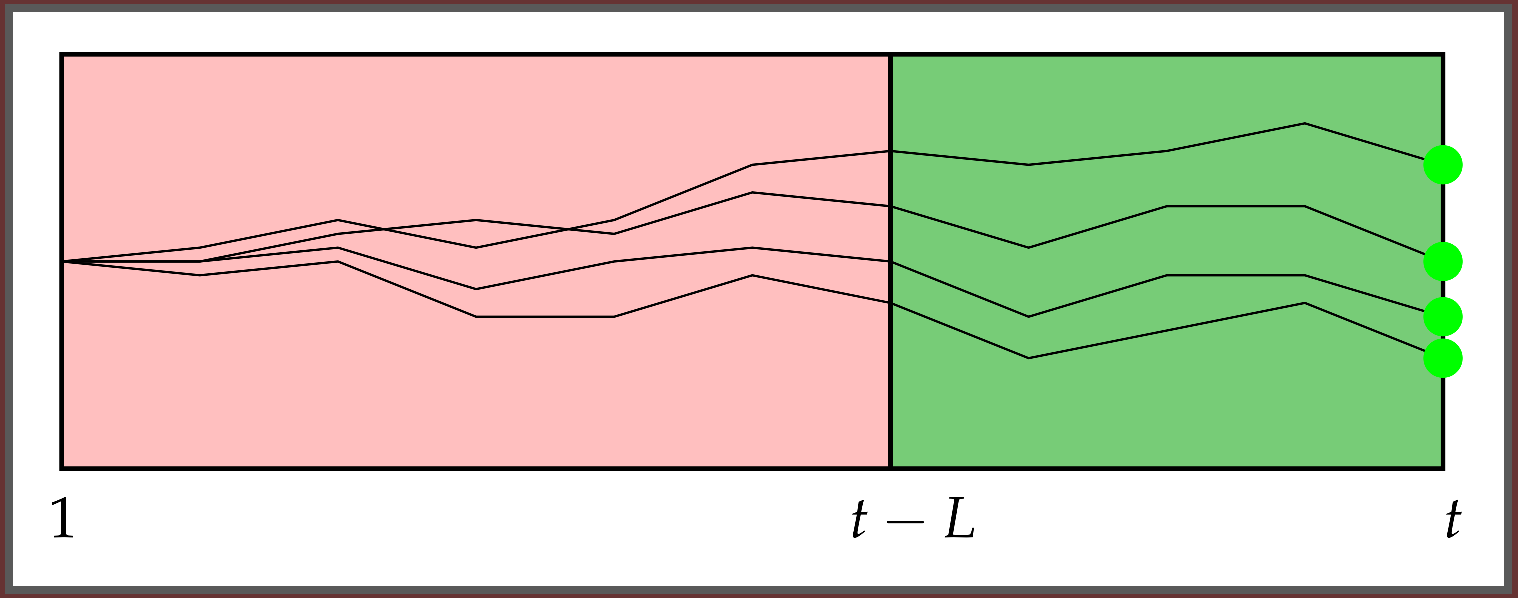
$$\boxed{\mathbf{x}^{r,i}_k \sim p(\mathbf{x}^r_k | \mathbf{x}^{r,i}_{1:k-1,k+1:t}, \mathbf{u}_{1:t}, \mathbf{z}_{1:t}, \mathbf{n}_{1:t})}$$



Fixed-lag roughening

- After resampling, apply an MCMC move step to $\{\mathbf{x}^{r,i}_{t-L+1:t}\}$
- Fixed-lag Gibbs sampler for RBPF SLAM:

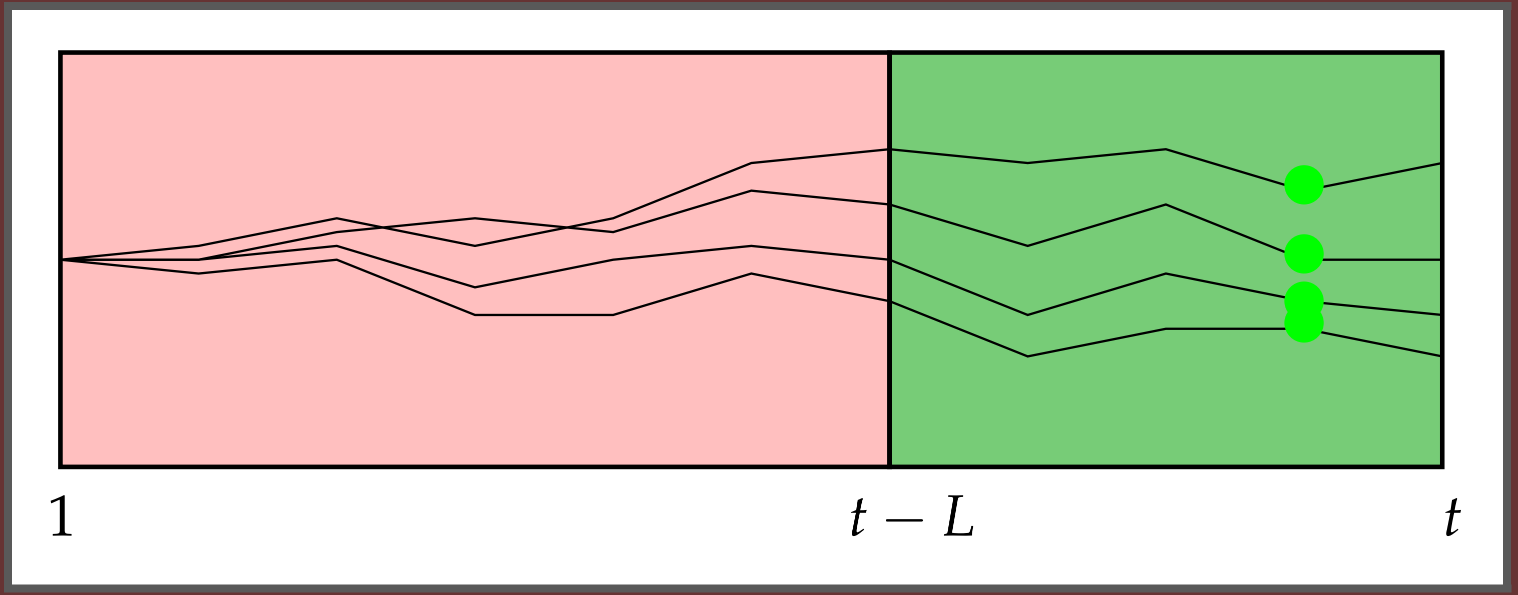
$$\boxed{\mathbf{x}^{r,i}_k \sim p(\mathbf{x}^r_k | \mathbf{x}^{r,i}_{1:k-1,k+1:t}, \mathbf{u}_{1:t}, \mathbf{z}_{1:t}, \mathbf{n}_{1:t})}$$



Fixed-lag roughening

- After resampling, apply an MCMC move step to $\{\mathbf{x}^{r,i}_{t-L+1:t}\}$
- Fixed-lag Gibbs sampler for RBPF SLAM:

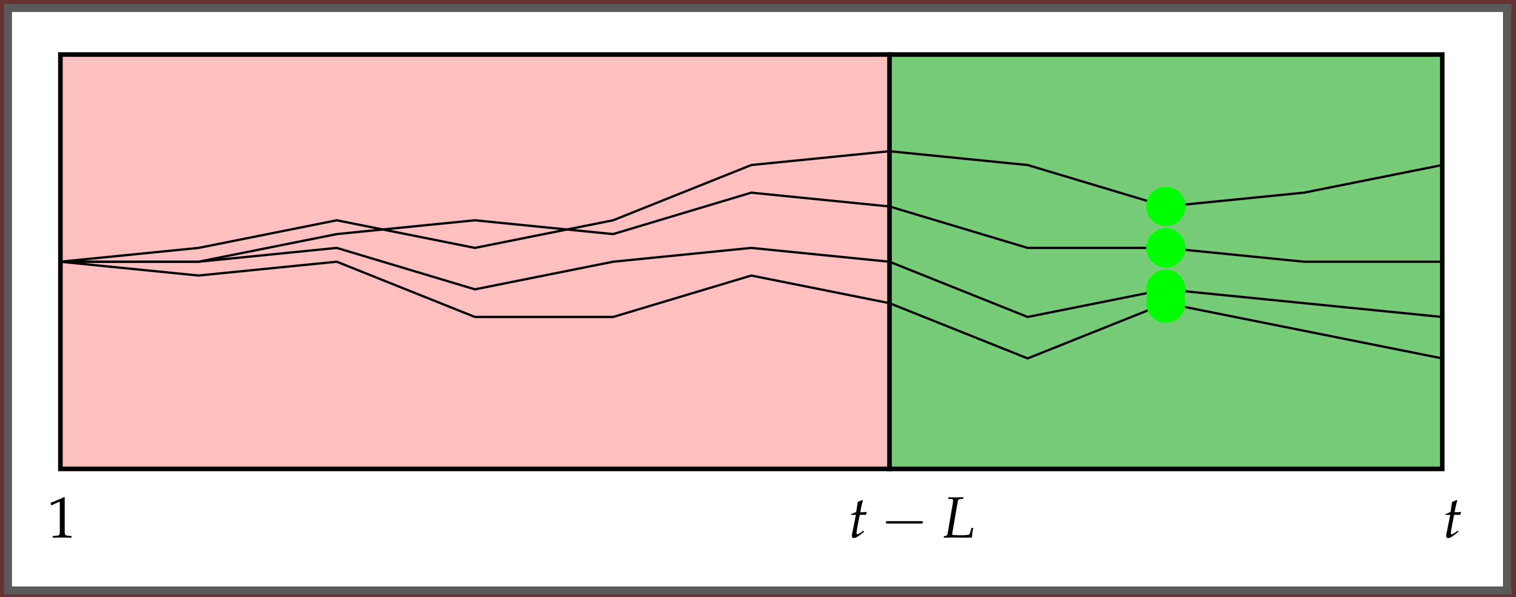
$$\boxed{\mathbf{x}^{r,i}_k \sim p(\mathbf{x}^r_k | \mathbf{x}^{r,i}_{1:k-1,k+1:t}, \mathbf{u}_{1:t}, \mathbf{z}_{1:t}, \mathbf{n}_{1:t})}$$



Fixed-lag roughening

- After resampling, apply an MCMC move step to $\{\mathbf{x}^{r,i}_{t-L+1:t}\}$
- Fixed-lag Gibbs sampler for RBPF SLAM:

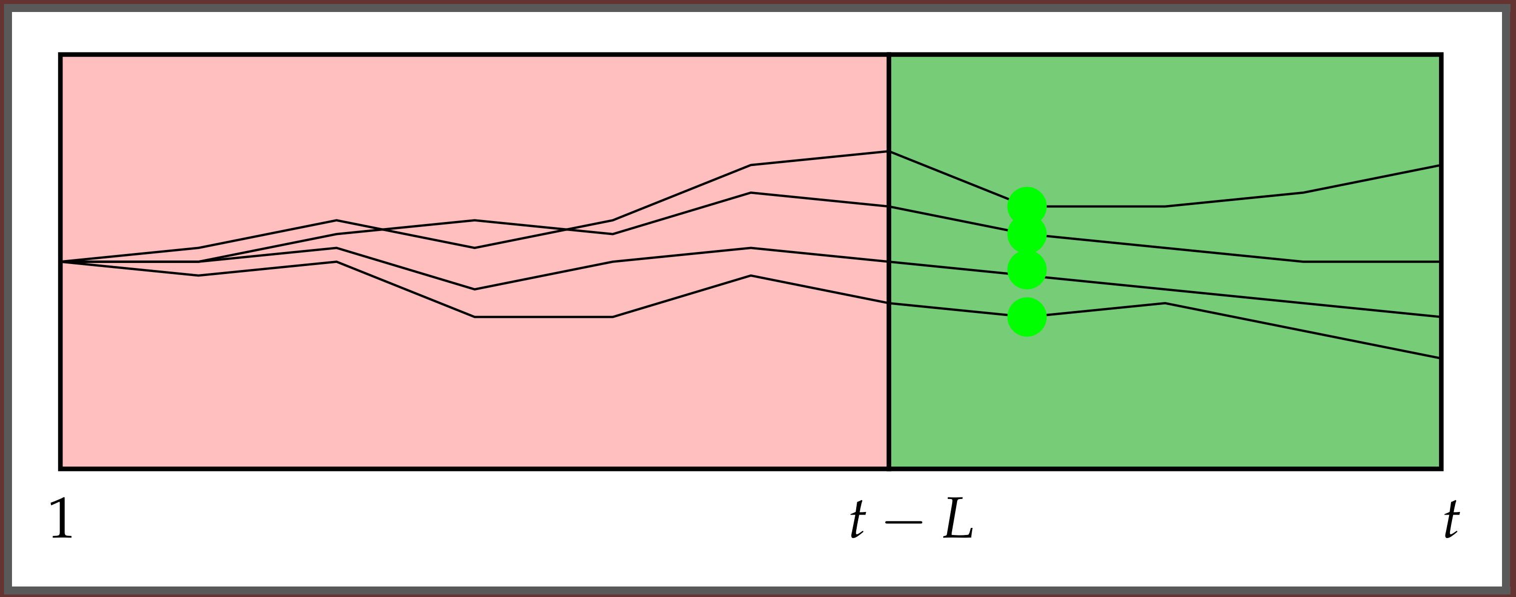
$$\boxed{\mathbf{x}^{r,i}_k \sim p(\mathbf{x}^r_k | \mathbf{x}^{r,i}_{1:k-1,k+1:t}, \mathbf{u}_{1:t}, \mathbf{z}_{1:t}, \mathbf{n}_{1:t})}$$



Fixed-lag roughening

- After resampling, apply an MCMC move step to $\{\mathbf{x}^{r,i}_{t-L+1:t}\}$
- Fixed-lag Gibbs sampler for RBPF SLAM:

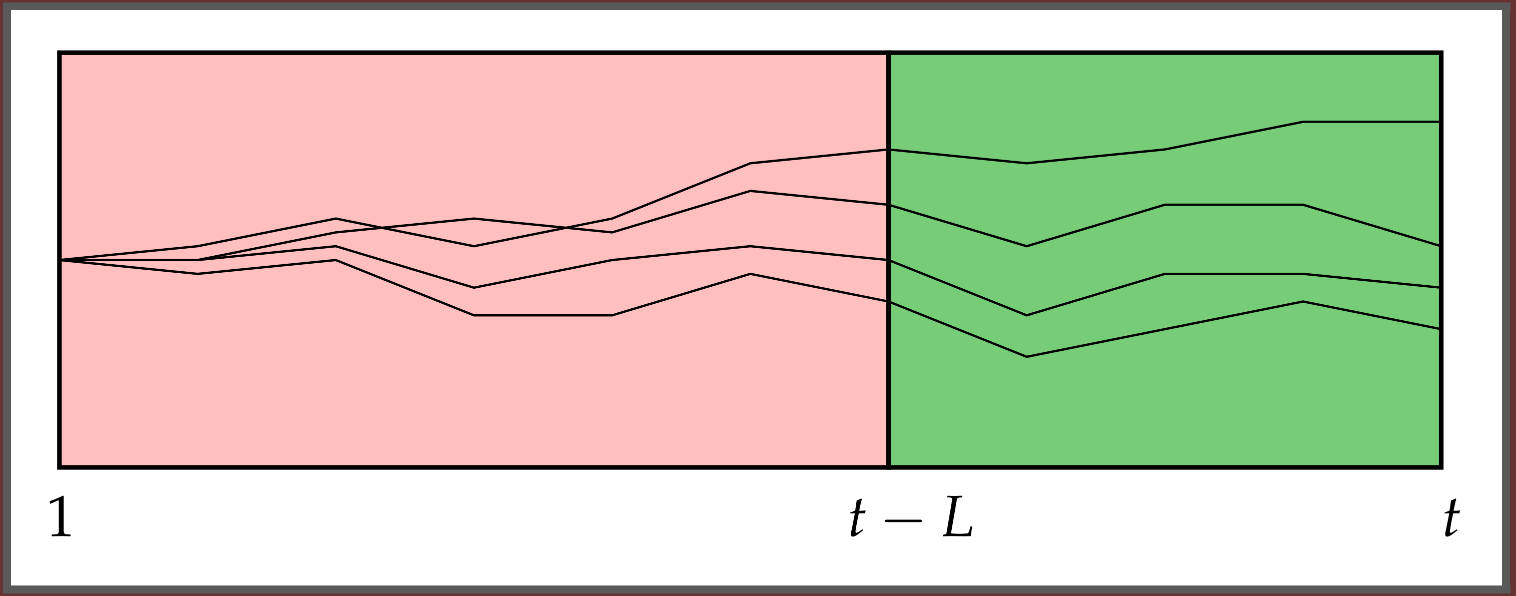
$$\boxed{\mathbf{x}^{r,i}_k \sim p(\mathbf{x}^r_k | \mathbf{x}^{r,i}_{1:k-1,k+1:t}, \mathbf{u}_{1:t}, \mathbf{z}_{1:t}, \mathbf{n}_{1:t})}$$



Block proposal

- Draw $\{\mathbf{x}^{r,i}_{t-L+1:t}\}$ from fully joint "optimal block proposal" distribution:

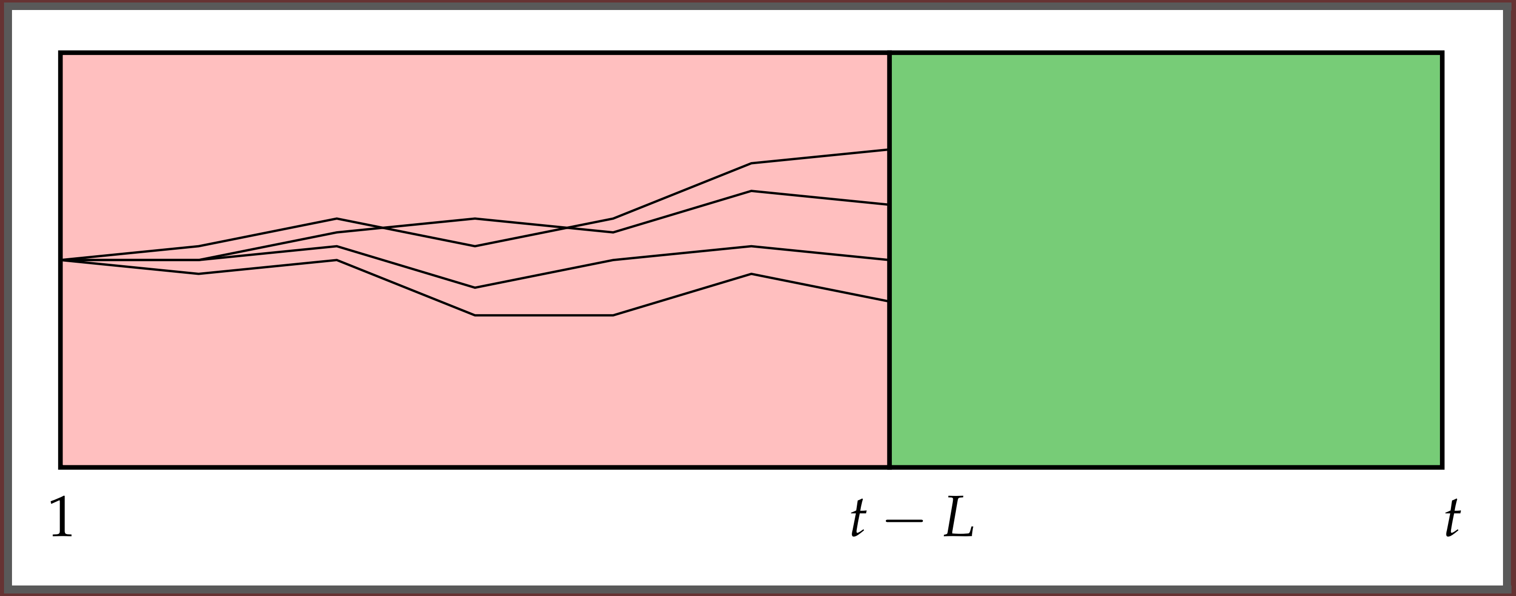
$$\boxed{p(\mathbf{x}^{r}_{t-L+1:t} | \mathbf{u}_{1:t}, \mathbf{z}_{1:t}, \mathbf{n}_{1:t}, \mathbf{x}^{r,i}_{t-L})}$$



Block proposal

- Draw $\{\mathbf{x}^{r,i}_{t-L+1:t}\}$ from fully joint "optimal block proposal" distribution:

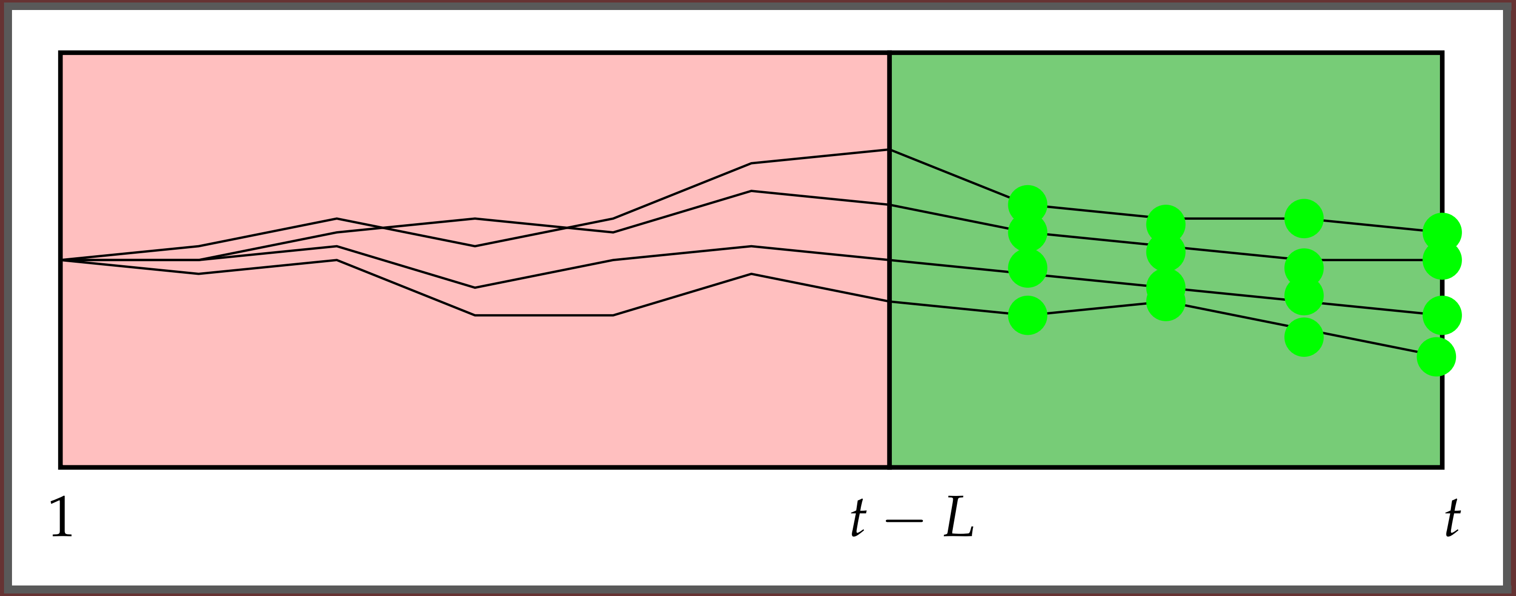
$$\boxed{p(\mathbf{x}^r_{t-L+1:t} | \mathbf{u}_{1:t}, \mathbf{z}_{1:t}, \mathbf{n}_{1:t}, \mathbf{x}^{r,i}_{t-L})}$$



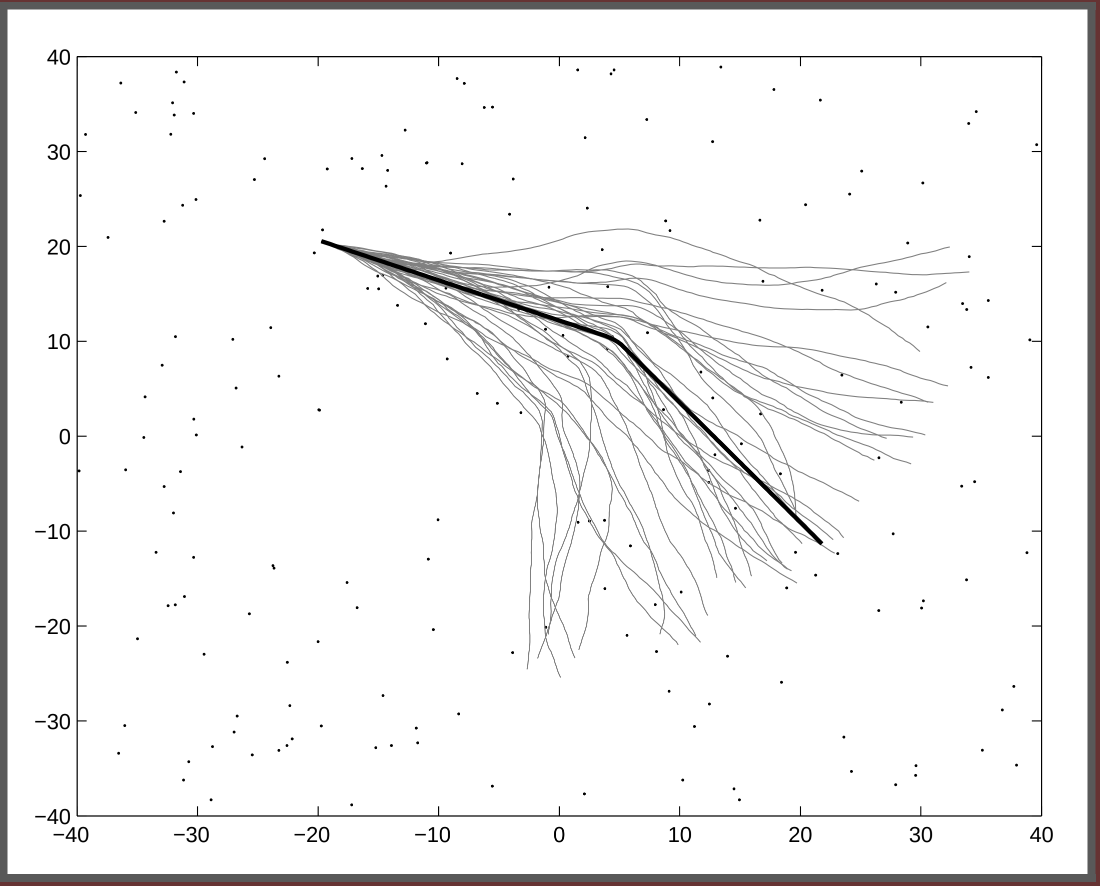
Block proposal

- Draw $\{\mathbf{x}^{r,i}_{t-L+1:t}\}$ from fully joint "optimal block proposal" distribution:

$$\boxed{p(\mathbf{x}^r_{t-L+1:t} | \mathbf{u}_{1:t}, \mathbf{z}_{1:t}, \mathbf{n}_{1:t}, \mathbf{x}^{r,i}_{t-L})}$$



Simulation results: sparse environment



- 27 sec., no loops
- 50 Monte Carlo trials averaged for all results

Norm. est. error sq. (NEES): $(\mathbf{x}^r_t - \hat{\mathbf{x}}^r_t)(\hat{\mathbf{P}}^r_t)^{-1}(\mathbf{x}^r_t - \hat{\mathbf{x}}^r_t)^T$

FS2

FLR(1)

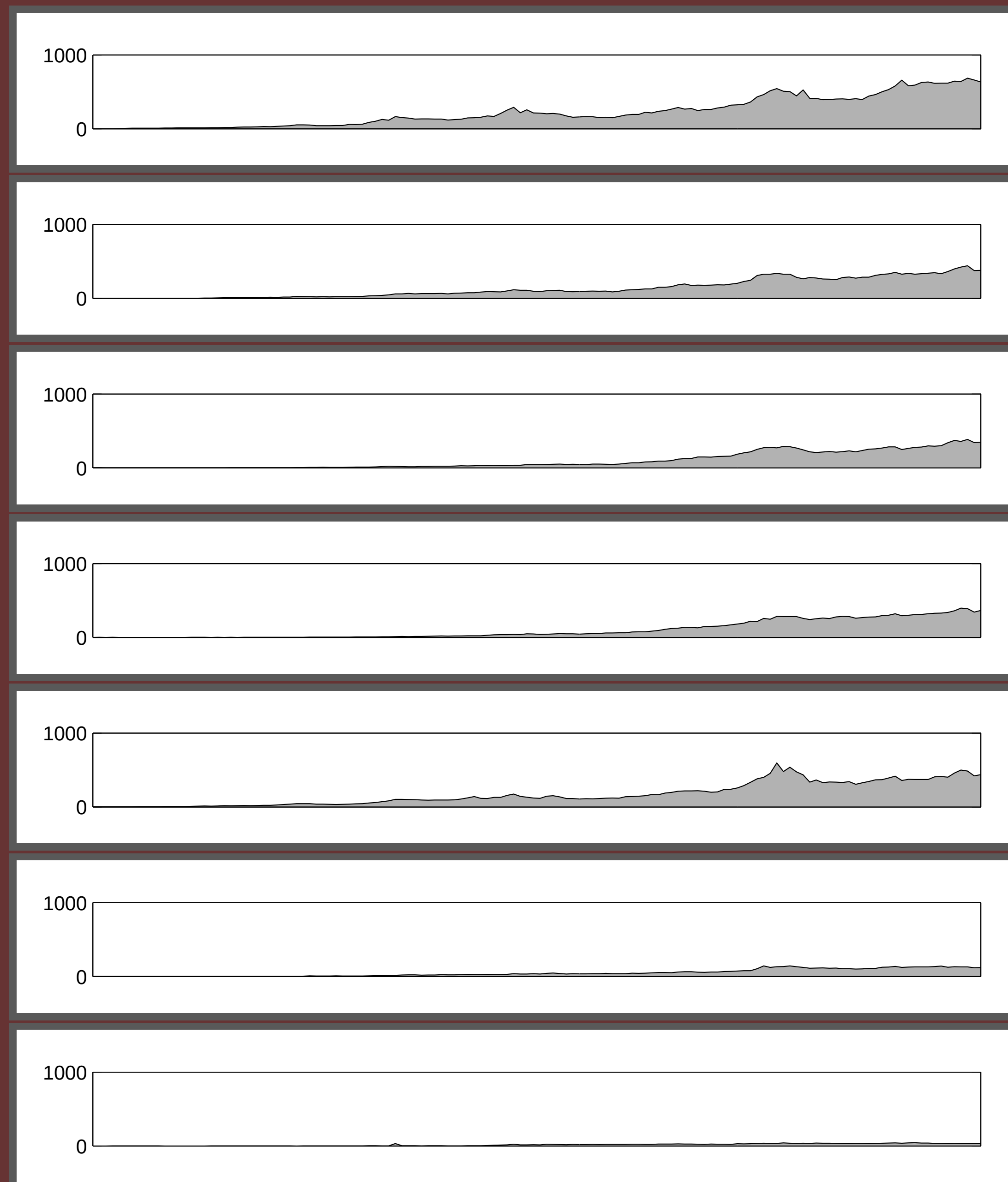
FLR(5)

FLR(10)

BP(1)

BP(5)

BP(10)



Unique samples of each pose: $|\{\mathbf{x}_k^{r,i}|\mathbf{u}_{1:t}, \mathbf{z}_{1:t}, \mathbf{n}_{1:t}\}|, k = 1 \ldots t$

FS2

FLR(1)

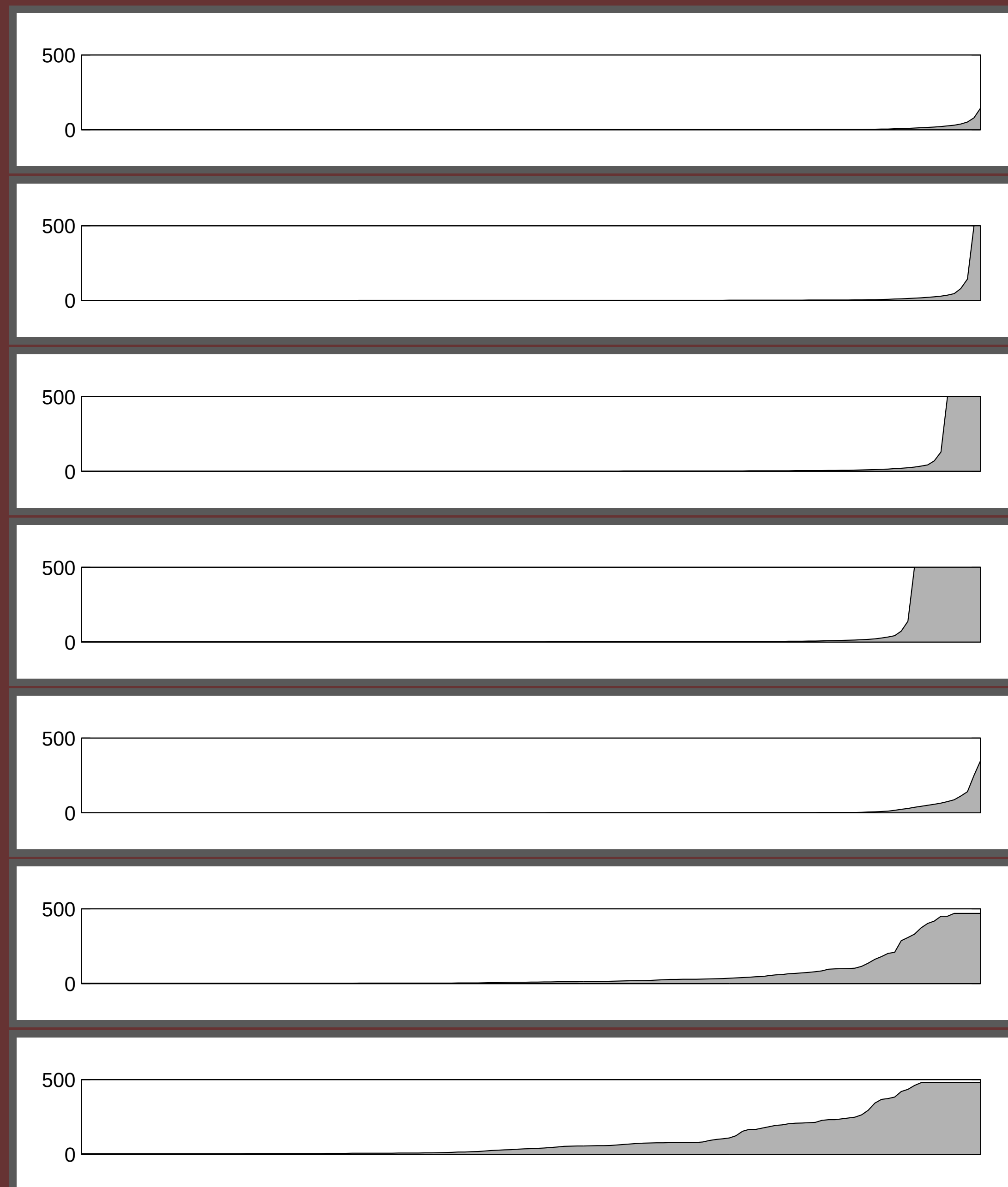
FLR(5)

FLR(10)

BP(1)

BP(5)

BP(10)

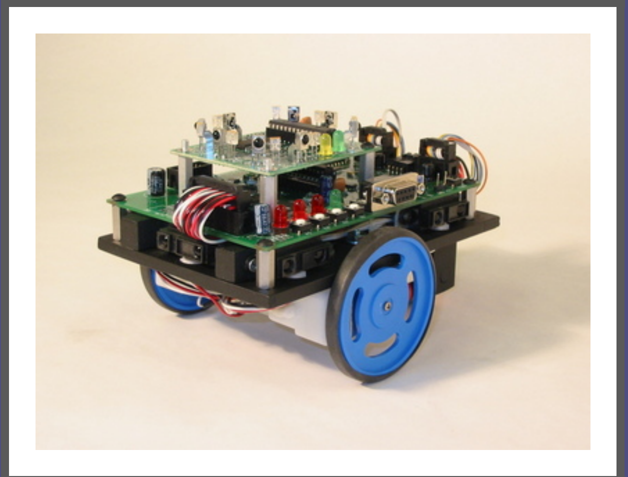


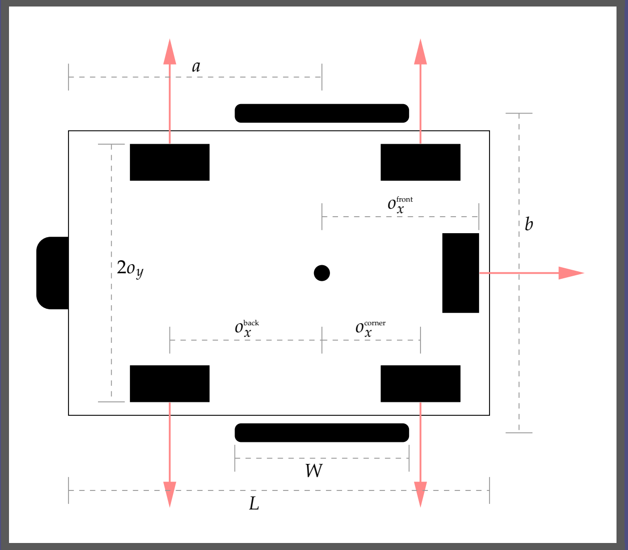
Thesis contributions

- An **analysis of mapping sensors** and **bounds on map error** for a simple range-bearing sensor model
- A Rao-Blackwellized particle filtering (RBPF) algorithm for **simultaneous localization and mapping (SLAM) with sparse sensing**
- Techniques for **incorporating prior information** in RBPF SLAM
- Two new **sampling strategies** for RBPF SLAM
- An **implementation of RBPF on a 16 MHz microcontroller**
- Full **software implementations** of all the algorithms in the thesis (and other standard algorithms)

SLAM on the Ratbots

- Ratbot mobile robot platform:
  - Inexpensive (several hundred US$)
  - Atmel ATMEGA64 8-bit, 16 MHz microcontroller
  - 64 KB program memory, 4 KB SRAM, 64 KB extended RAM
  - Five Sharp GPD12 IR rangefinders
- RBPF SLAM implementation details:
  - Fixed-point numbers
  - Lookup tables
  - Hand optimization of linear algebra
  - Multiscan SLAM— overlap data collection and processing
  - Efficient resampling
- Experiments in progress





Summary of contributions

- **Theoretical**:
  - Generalized sensor model, generic occupancy grid mapping
  - Bounds on ML map error in terms of sensor characteristics
  - Analytical comparison of mapping capabilities of laser, SONAR, IR
- **Particle filtering mapping algorithms**:
  - Multiscan particle filter for sparse arrays of range sensors
  - Rao-Blackwellized constraint filter: inference and enforcement of pairwise constraints in RBPF
  - Rectilinearity constraints
  - Fixed-lag roughening and the block proposal distribution: sample the *pose history* over a fixed lag time
- **Implementations**:
  - SLAM on a microcontroller (Ratbots)
  - All algorithms implemented in full in a unified framework

Future directions

- **Sensing requirements for mapping**:
  - Realistic trajectories, pose uncertainty, structured environments
  - More thorough definition and analysis of the “space of mapping sensors”
- **Exploration**:
  - Active mapping vs. passive mapping
  - What is the best exploration strategy for a given sensor?
  - Exploit motion to simulate a high-fidelity sensor with a low-fidelity one
- **Filtering algorithms**:
  - Exploit relationship between SLAM and general filtering problem
  - Practical scenarios: flexibility in trading off efficiency and accuracy
- **Practical implementations**:
  - What computational shortcuts can we take?

Final thoughts

- Right now, many mapping problems are “solved” if you throw enough $ at them

BUT

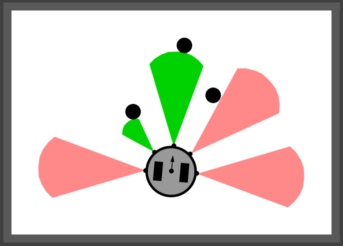
- Fundamental questions about the requirements for mapping are important to answer
- Practical mapping with inexpensive robots — must handle limitations in:
  - sensing
  - computing
  - memory
  - energy

Thanks for coming!

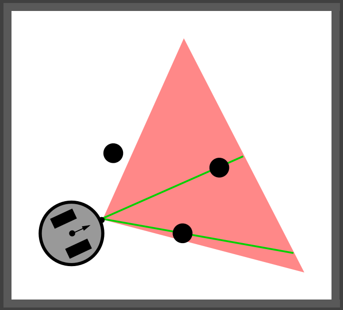
Related work: limited sensing

- **Topological mapping**: Acar et al. (2001); Tovar et al. (2003); Huang and Beevers (2005)

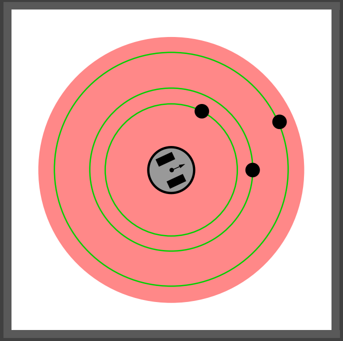
- **SONAR-based geometrical mapping**: Wijk and Christensen (2000); Zunino and Christensen (2001); Leonard et al. (2002); Tardós et al. (2002)



- **Bearing-only SLAM**: Deans and Hebert (2000a); Bailey (2003); Solá et al. (2005)

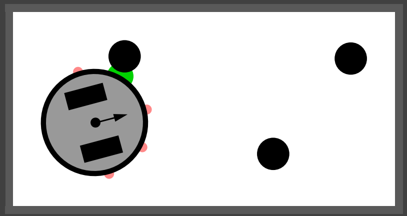


- **Range-only SLAM** (with RF beacons): Kantor and Singh (2002); Kurth (2004); Djugash et al. (2005)



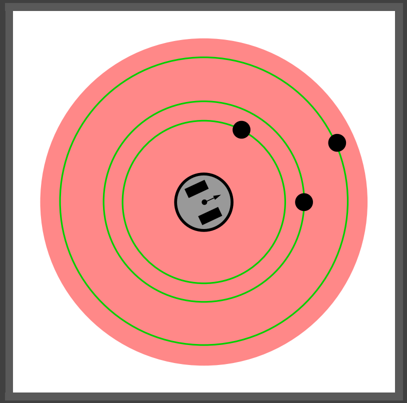
Sensors for mapping

Contact sensor array



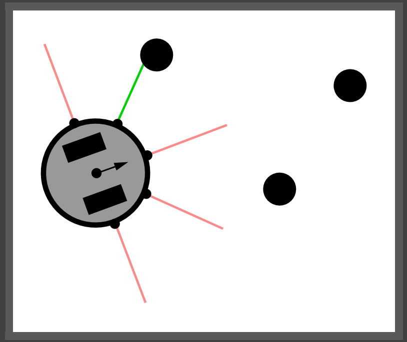
Zero-range, low-res, accurate, cheap

RF signal strength



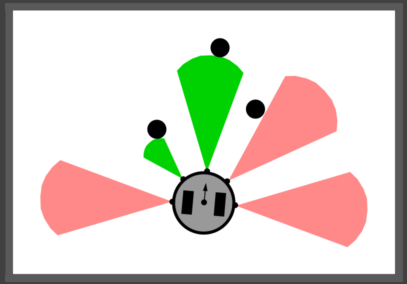
Mid-range, no-res, inaccurate, medium-cost
no bearing information (range only)

Infrared array



Short-range, low-res, accurate, cheap

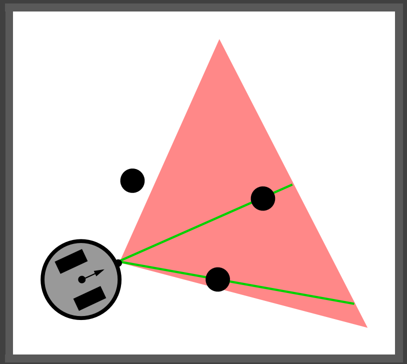
SONAR array



Mid-range, low-res, inaccurate, medium-cost

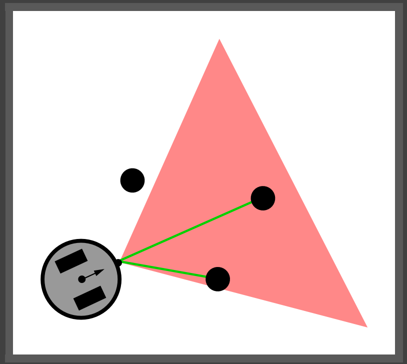
Sensors for mapping (cont.)

Monocular camera



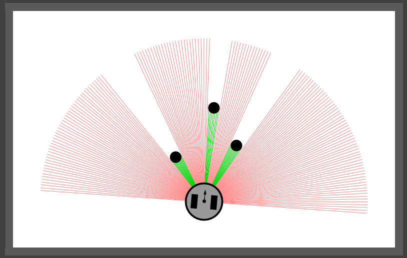
Long-range, high-res, accurate, medium-cost
no range information (bearing only)

Stereo camera



Long-range, high-res, accurate, high-cost

Laser rangefinder



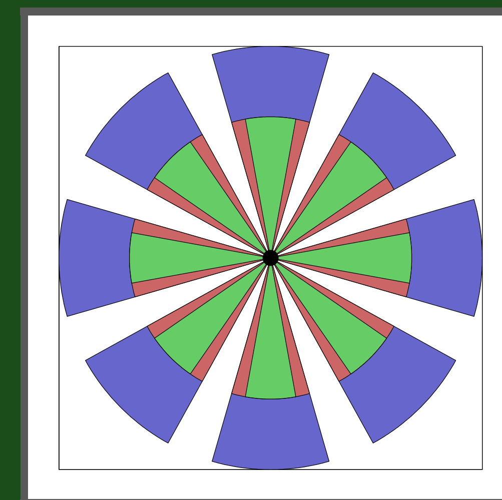
Long-range, high-res, accurate, high-cost

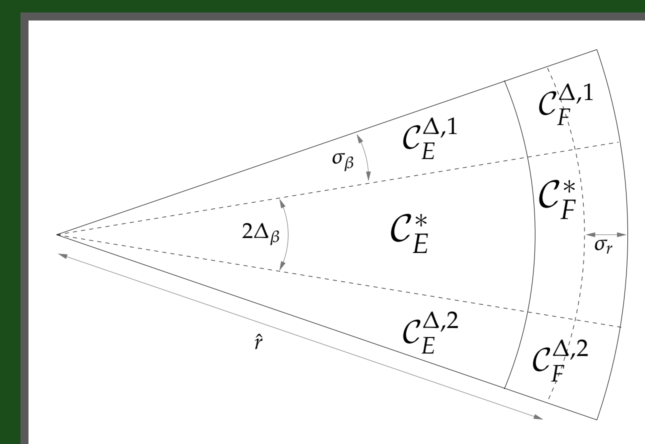
Basic algorithm (landmark based mapping)

```
loop
    Move; update pose estimate based on odometry
    Sense the environment
    Extract features from the raw sensing data
    Match features with the current map
    Based on matches, update pose and map estimates
    Add unmatched features to the map
end loop
```

Sensor and environment models

- Environment: $M \times M$ grid of cells m_{ij}; cells occupied ($\mathtt{F}$) at rate d, $\mathtt{E}$ otherwise
- Trajectory: $\mathbf{x}_t^r, t \in [0, T]$; *assumption: poses drawn uniformly at random*
- Sensor:
  - Ring: ρ beams, angles $\beta_i = i\frac{2\pi}{\rho} + U[-\sigma_\beta, \sigma_\beta]$
  - Firing frequency F
  - Beam: goes until *detecting* an occupied cell
  - False negative rate $\varepsilon_{\mathtt{E}}$, false positive rate $\varepsilon_{\mathtt{F}}$
- Mapping: **occupancy grid**; cell measurements depend on “region” in beam
  - $m_{ij} \in \mathcal{C}_{\mathtt{F}}$: bel$(m_{ij} = \mathtt{F})$+=$p_0$
  - $m_{ij} \in \mathcal{C}_{\mathtt{E}}$: bel$(m_{ij} = \mathtt{E})$+=$p_0$





Obtaining a bound on expected map error

Bound expected # observations of a cell

Compute likelihood that an observation is incorrect

Conditions for map convergence

Bound expected error in ML map

Bound on expected # observations

Let:

$$\mathcal{E}_{\mathrm{E}} = ((1-d)(1-\varepsilon_{\mathrm{E}}) + d\varepsilon_{\mathrm{F}}) \quad p(\text{some cell in a beam registers as } \texttt{E})$$
$$\mathcal{E}_{\mathrm{F}} = (d(1-\varepsilon_{\mathrm{F}}) + (1-d)\varepsilon_{\mathrm{E}}) \quad p(\text{some cell in a beam registers as } \texttt{F})$$

Expected # o_{ab} of times any cell m_{ab} is updated:

$$\boxed{E[o_{ab}] \geq \frac{2TF\rho(\Delta_\beta+\sigma_\beta)}{M^2} \sum_{\tau=0}^{\left\lceil \frac{r^+ + \sigma_r}{\delta} \right\rceil} \tau \cdot p_{\mathrm{obs}}}$$

where:

$$\boxed{p_{\mathrm{obs}} \geq \begin{cases} \mathcal{E}_{\mathrm{E}}^{\Delta_\beta \tau^2} & \text{if } \tau\delta > \sigma_r \\ 1 & \text{otherwise} \end{cases}}$$

Likelihood of an incorrect observation

Let: $$\boxed{p_{\mathrm{f}} = \min\left\{1, \frac{\Delta_\beta \mathcal{E}_{\mathrm{F}}}{\delta^2}\left((\tau\delta + \sigma_r)^2 - \max\{0, \tau\delta - \sigma_r\}^2\right)\right\}}$$

If cell m_{ij} is unoccupied ($\mathtt{E}$) the likelihood that any update to m_{ij} is incorrect is:

$$\boxed{p(\text{inc}|m_{ij} = \mathtt{E}) \leq \sum_{\tau=0}^{\left\lceil \frac{r^+ + \sigma_r}{\delta} \right\rceil} p_{\mathrm{obs}} \cdot p_{\mathrm{f}} \cdot \frac{(\tau\delta + \sigma_r)^2 - \max\{0, \tau\delta - \sigma_r\}^2}{(\tau\delta + \sigma_r)^2}}$$

If cell m_{ij} is occupied ($\mathtt{F}$) the likelihood that any update to m_{ij} is incorrect is:

$$\boxed{p(\text{inc}|m_{ij} = \mathtt{F}) \leq \sum_{\tau=0}^{\left\lceil \frac{r^+ + \sigma_r}{\delta} \right\rceil} p_{\mathrm{obs}} \cdot p_{\mathrm{f}} \cdot \frac{\max\{0, \tau\delta - \sigma_r\}^2}{(\tau\delta + \sigma_r)^2}}$$

Bound on expected ML map error

$\boxed{\text{The map converges if } p_{\text{inc}} < 1/2}$

Let $\nu = \sum_{ij} \nu_{ij}$, where $\nu_{ij} = 1$ if the ML estimate for cell m_{ij} is **incorrect**, and $\nu_{ij} = 0$ otherwise.

If $p_{\text{inc}} < 1/2$:

$$\boxed{E[\nu] \leq M^2 \exp\left\{ -2E[o_{ab}] \left(\frac{1}{2} - p_{\text{inc}} \right)^2 \right\}}$$

(Chernoff bound)

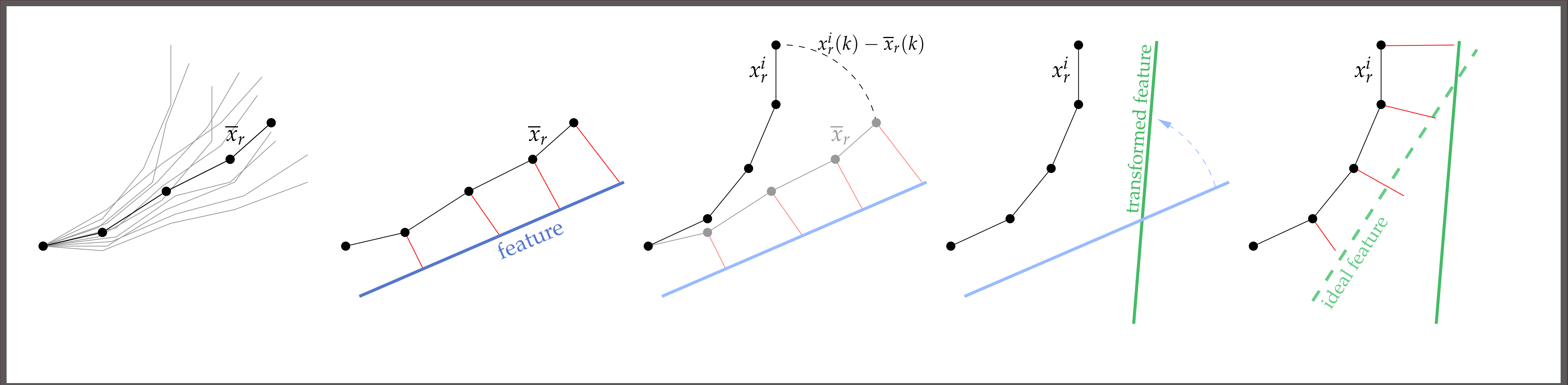
Approach of Leonard et al. (2002)

- Incorporate trajectory into state vector:

$$\mathbf{x}_t = [\mathbf{x}^r_{t-m+1:t} \;\; \mathbf{x}^m_t]^T = [\mathbf{x}^r_{t-m+1} \;\; \mathbf{x}^r_{t-m+2} \;\; \ldots \;\; \mathbf{x}^r_t \;\; \mathbf{x}^m_t]^T$$

- Keep measurements from last m timesteps
- At each timestep, do feature extraction using $\mathbf{z}_{t-m+1:t}$
- Discard data when:
  - It becomes too old
  - It is used to extract a particular feature
- Advantage: features extracted as soon as enough data available
- Main disadvantage: computational

Multiscan SLAM approximations



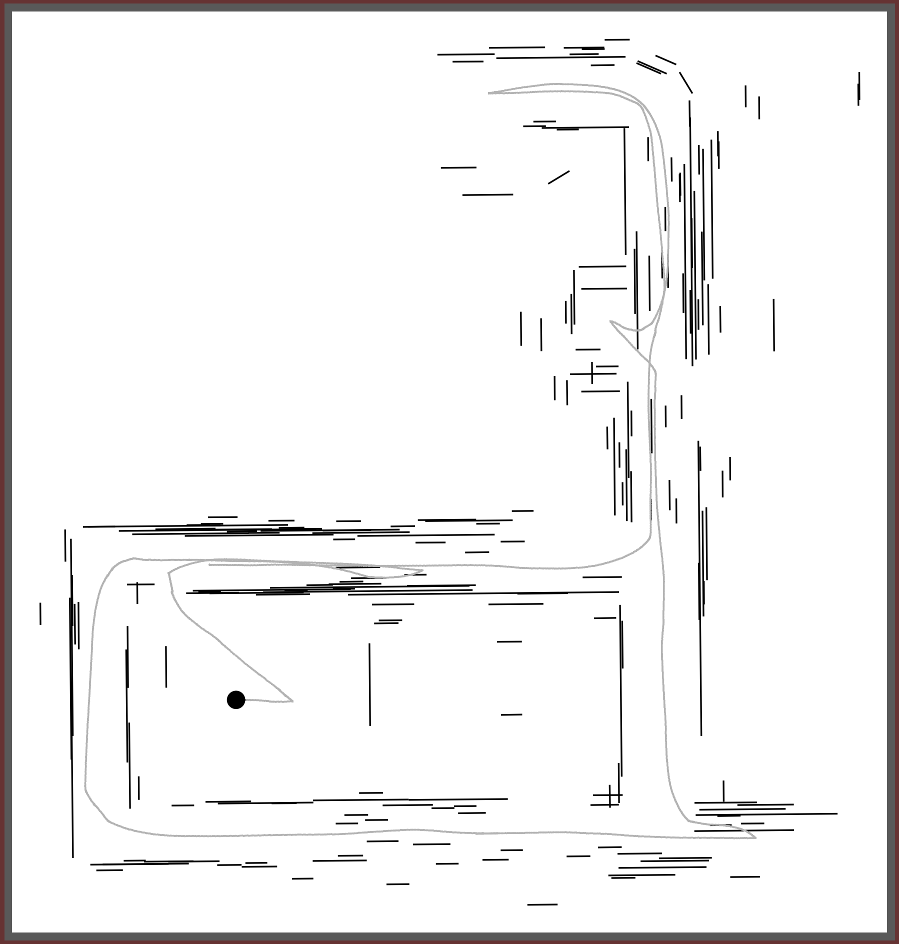
- Two big computationally motivated approximations:
  1. Ignore correlations between measurements from multiple poses
  2. Extract features using *expected* trajectory (picture)
- These hinge on pose uncertainty being small over m consecutive poses
- Alternatives:
  - Adaptively choose m
  - Extract features per-particle: only with very few particles
  - Extract features “per-stratum”

Enforcing relative constraints

- Problem: $(r_i, \theta_i), (r_j, \theta_j)$ are not independent if $c_{ij} \neq \star$
  - Group constrained landmarks: $\boxed{L_i = [r_1 \;\; \theta_1 \;\; r_2 \;\; \theta_2 \;\; \ldots \;\; r_n \;\; \theta_n]^T}$
  - Rewrite, e.g.: $\boxed{L_i = [r_1 \;\; \theta_1 \;\; r_2 \;\; g_2(c_{1,2}; \theta_1) \;\; \ldots \;\; r_n \;\; g_n(c_{1,n}; \theta_1)]^T}$
  - Filter on reduced state: $\boxed{L_i = [r_1 \;\; r_2 \;\; \ldots \;\; r_n \;\; \theta_1]^T}$
  - Conditioned on θ_1, the r_is are independent

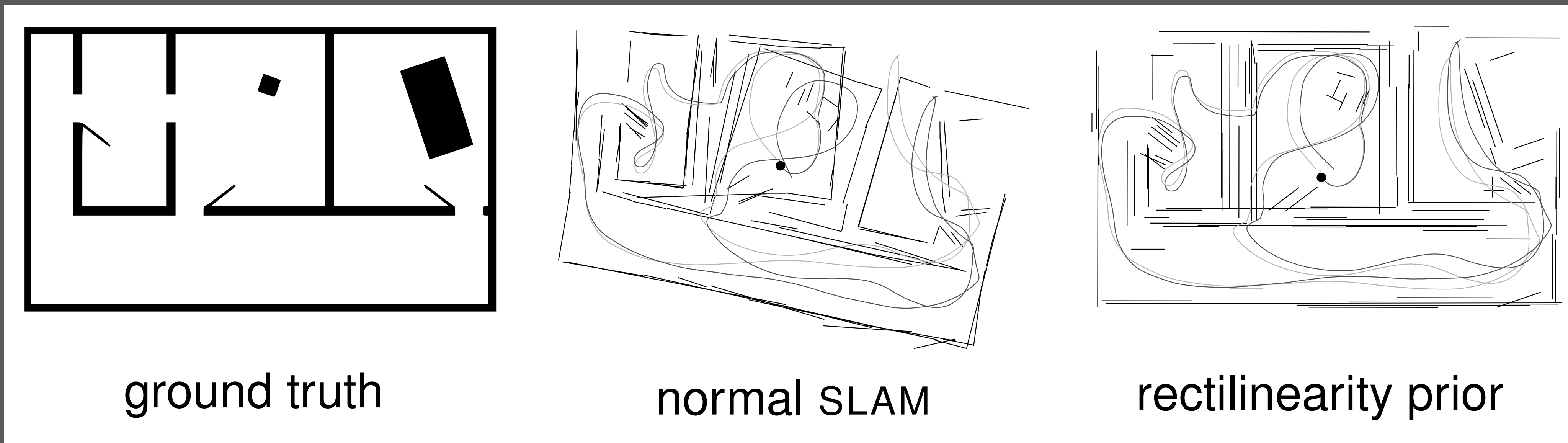
$$\boxed{p(\mathbf{x}^r_{1:t}, \mathbf{x}^m | \mathbf{u}_{1:t}, \mathbf{z}_{1:t}, \mathbf{n}_{1:t}) = p(\mathbf{x}^r_{1:t}, \mathbf{x}^{m,\mathrm{c}} | \mathbf{u}_{1:t}, \mathbf{z}_{1:t}, \mathbf{n}_{1:t}) \prod_{i=1}^{|\mathbf{x}^{m,\mathrm{f}}|} p(\mathbf{x}^{m,\mathrm{f}}_i | \mathbf{x}^r_{1:t}, \mathbf{x}^{m,\mathrm{c}}, \mathbf{z}_{1:t}, \mathbf{n}_{1:t})}$$

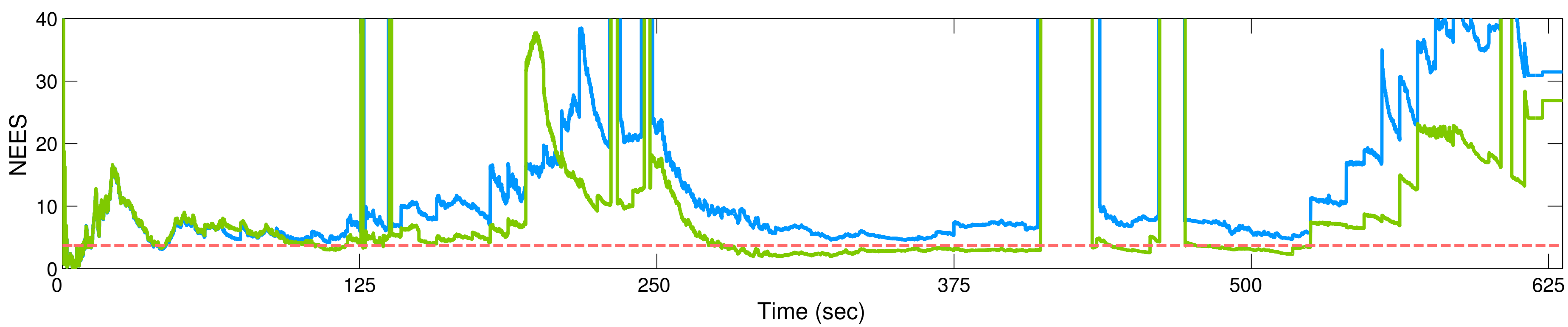
Sensitivity to cross-covariance



- Our approach:
  1. Sample values for constrained variables
  2. Condition unconstrained variables on sampled values
- Conditioning: sensitive to landmark estimation inaccuracy
- Cross-covariance of Gaussian PDFs must be accurately estimated
- Are EKFs good enough? (How non-Gaussian are landmark PDFs?)

Map constraints improve trajectory estimation





trajectory estimation error

Fixed-lag roughening

- After resampling, apply an MCMC move step to $\{\mathbf{x}^{r,i}_{t-L+1:t}\}$
- Fixed-lag Gibbs sampler for RBPF SLAM:

$$\boxed{\begin{aligned} \mathbf{x}^{r,i}_{t-L+1} &\sim p(\mathbf{x}^{r}_{t-L+1}|\mathbf{x}^{r,i}_{1:t-L,t-L+2:t}, \mathbf{u}_{1:t}, \mathbf{z}_{1:t}, \mathbf{n}_{1:t}) \\ & \ldots \\ \mathbf{x}^{r,i}_{k} &\sim p(\mathbf{x}^{r}_{k}|\mathbf{x}^{r,i}_{1:k-1,k+1:t}, \mathbf{u}_{1:t}, \mathbf{z}_{1:t}, \mathbf{n}_{1:t}) \\ & \ldots \\ \mathbf{x}^{r,i}_{t} &\sim p(\mathbf{x}^{r}_{t}|\mathbf{x}^{r,i}_{1:t-1}, \mathbf{u}_{1:t}, \mathbf{z}_{1:t}, \mathbf{n}_{1:t}) \end{aligned}}$$

$$p(\mathbf{x}^{r}_{k}|\mathbf{x}^{r,i}_{1:k-1,k+1:t}, \mathbf{u}_{1:t}, \mathbf{z}_{1:t}, \mathbf{n}_{1:t}) =$$

$$\eta \int \underbrace{p(\mathbf{z}_k|\mathbf{x}^{r,i}_{k}, \mathbf{n}_k, \mathbf{x}^{m}_{\mathbf{n}_k})}_{\text{measurement}} \underbrace{p(\mathbf{x}^{m}_{\mathbf{n}_k}|\mathbf{x}^{r,i}_{1:k-1,k+1:t}, \mathbf{z}_{1:k-1,k+1:t}, \mathbf{n}_{1:t})}_{\text{landmark}} \, d\mathbf{x}^{m}_{\mathbf{n}_k}$$

$$\underbrace{p(\mathbf{x}^{r}_{k}|\mathbf{x}^{r,i}_{k-1}, \mathbf{u}_k)}_{\text{forward}} \underbrace{p(\mathbf{x}^{r}_{k}|\mathbf{x}^{r,i}_{k+1}, \mathbf{u}_{k+1})}_{\text{backward}}$$

Block proposal

- Draw $\{\mathbf{x}_{t-L+1:t}^{r,i}\}$ from joint "L-optimal block proposal" distribution:

$$\boxed{p(\mathbf{x}_{t-L+1:t}^{r} | \mathbf{u}_{1:t}, \mathbf{z}_{1:t}, \mathbf{n}_{1:t}, \mathbf{x}_{t-L}^{r,i})}$$

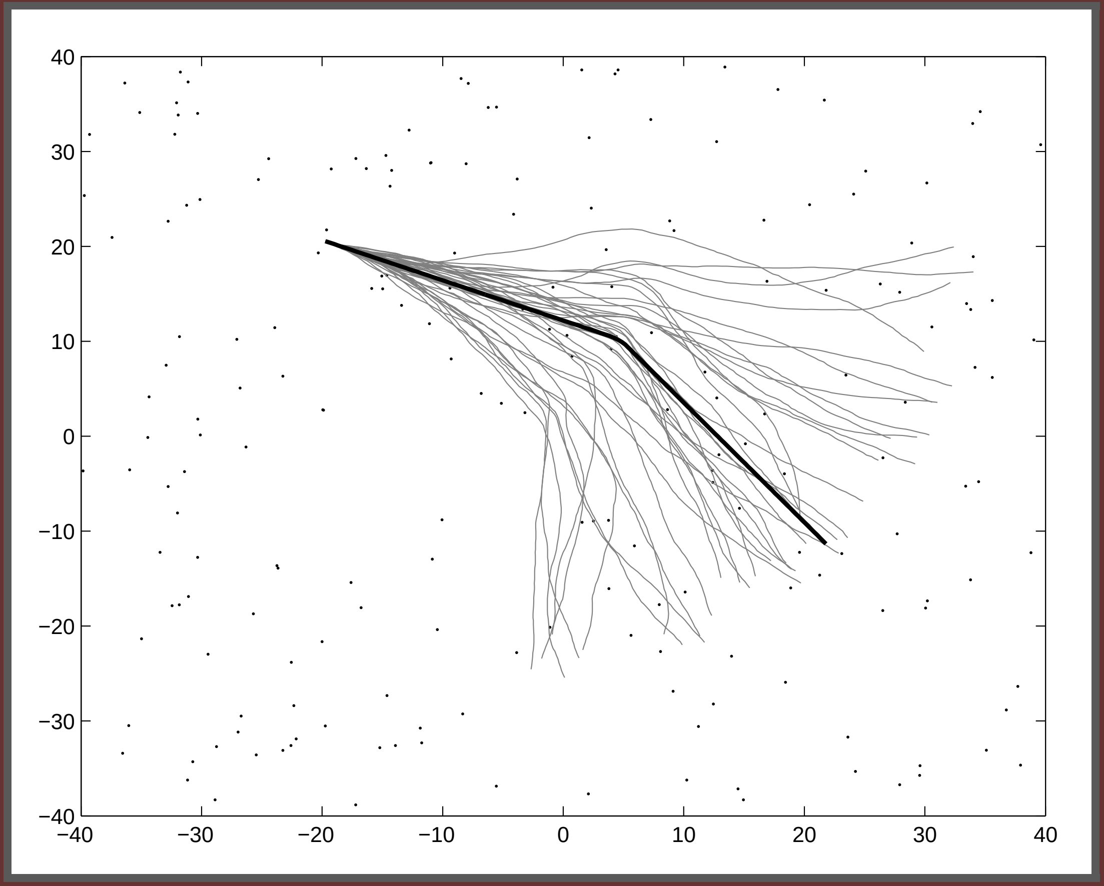
- How to do it: *forward filtering/backward sampling* (Chib, 1996)

$$\boxed{\underbrace{p(\mathbf{x}_{k}^{r} | \mathbf{u}_{1:t}, \mathbf{z}_{1:t}, \mathbf{n}_{1:t}, \mathbf{x}_{t-L}^{r,i}, \mathbf{x}_{k+1:t}^{r,i})}_{\text{sampling distribution}} \propto \underbrace{p(\mathbf{x}_{k}^{r} | \mathbf{u}_{1:k}, \mathbf{z}_{1:k}, \mathbf{n}_{1:k}, \mathbf{x}_{t-L}^{r,i})}_{\text{forward filtering}} \underbrace{p(\mathbf{x}_{k+1}^{r} | \mathbf{x}_{k}^{r,i}, \mathbf{u}_{k+1})}_{\text{backward model}}}$$

  - Filter forward using an EKF
  - Sample $\mathbf{x}_{t}^{r,i} \sim p(\mathbf{x}_{t}^{r} | \mathbf{u}_{1:t}, \mathbf{z}_{1:t}, \mathbf{n}_{1:t}, \mathbf{x}_{t-L}^{r,i})$
  - Compute sampling distribution for $\mathbf{x}_{t-1}^{r,i}$ and sample
  - Continue back to $t - L + 1$

- Need to reweight particles: $\omega_{t}^{i} = \omega_{t-1}^{i}\, p(\mathbf{z}_{t} | \mathbf{x}_{1:t-L}^{r,i}, \mathbf{u}_{1:t}, \mathbf{z}_{1:t-1}, \mathbf{n}_{1:t})$

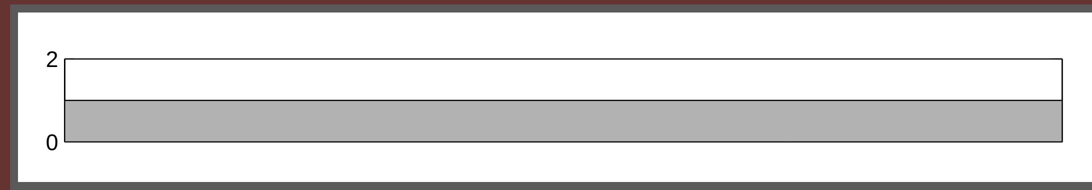
Extra simulation results: sparse environment



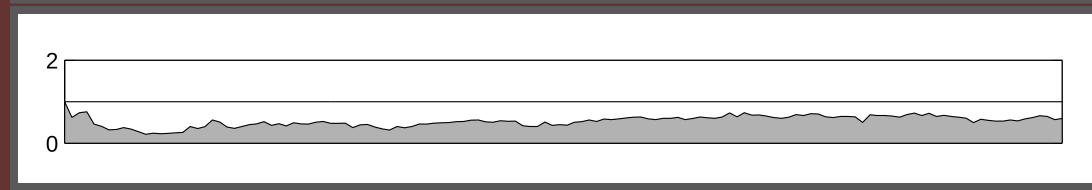
- 27 sec., no loops
- 50 Monte Carlo trials averaged for all results

NEES ratio: NEES(alg) / NEES(FS2)

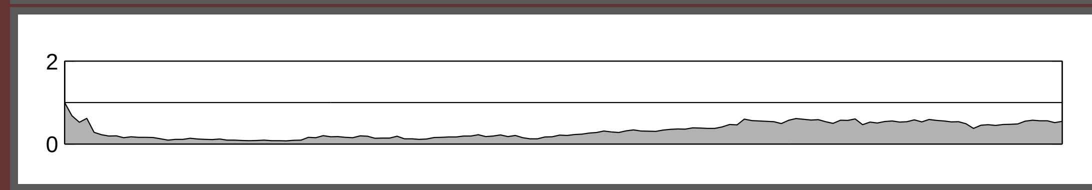
FS2



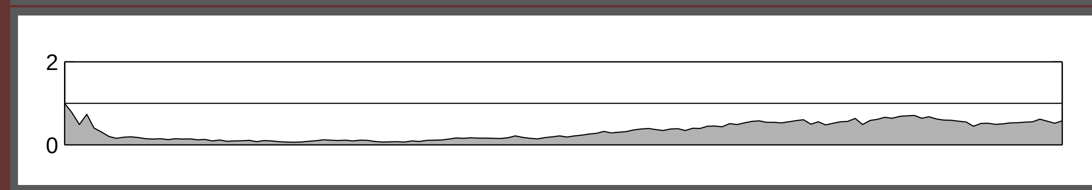
FLR(1)



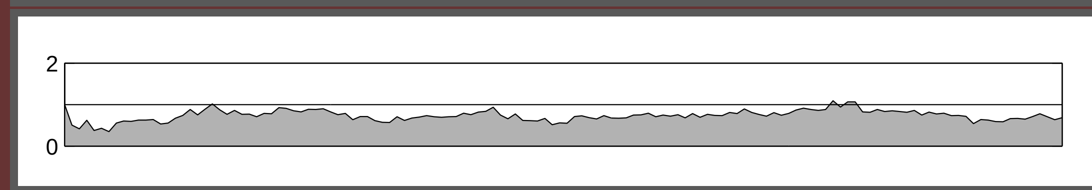
FLR(5)



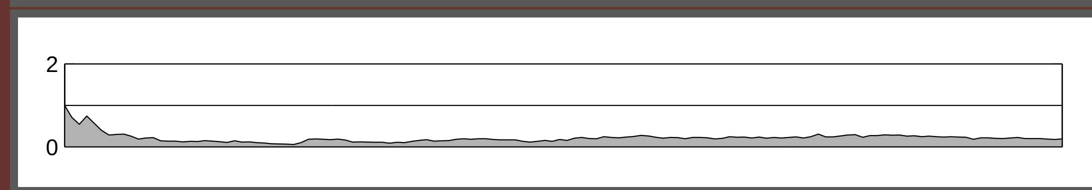
FLR(10)



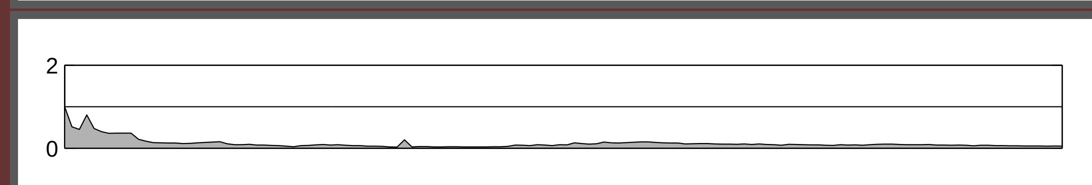
BP(1)



BP(5)



BP(10)



effective particles ($\hat{N}_{\text{eff}}$): $1/\sum_{i=1}^{N} (\omega_t^i)^2$

FS2

FLR(1)

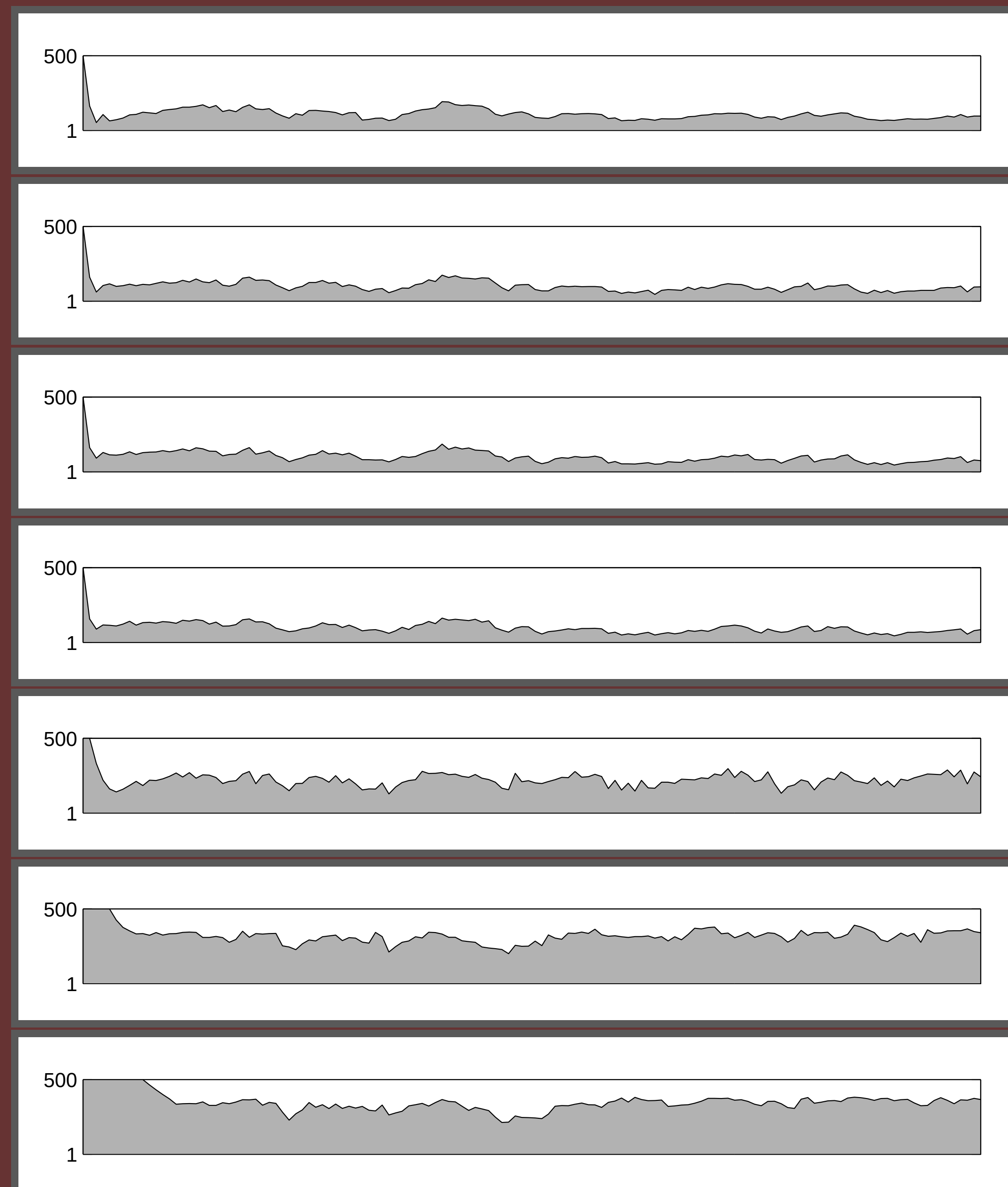
FLR(5)

FLR(10)

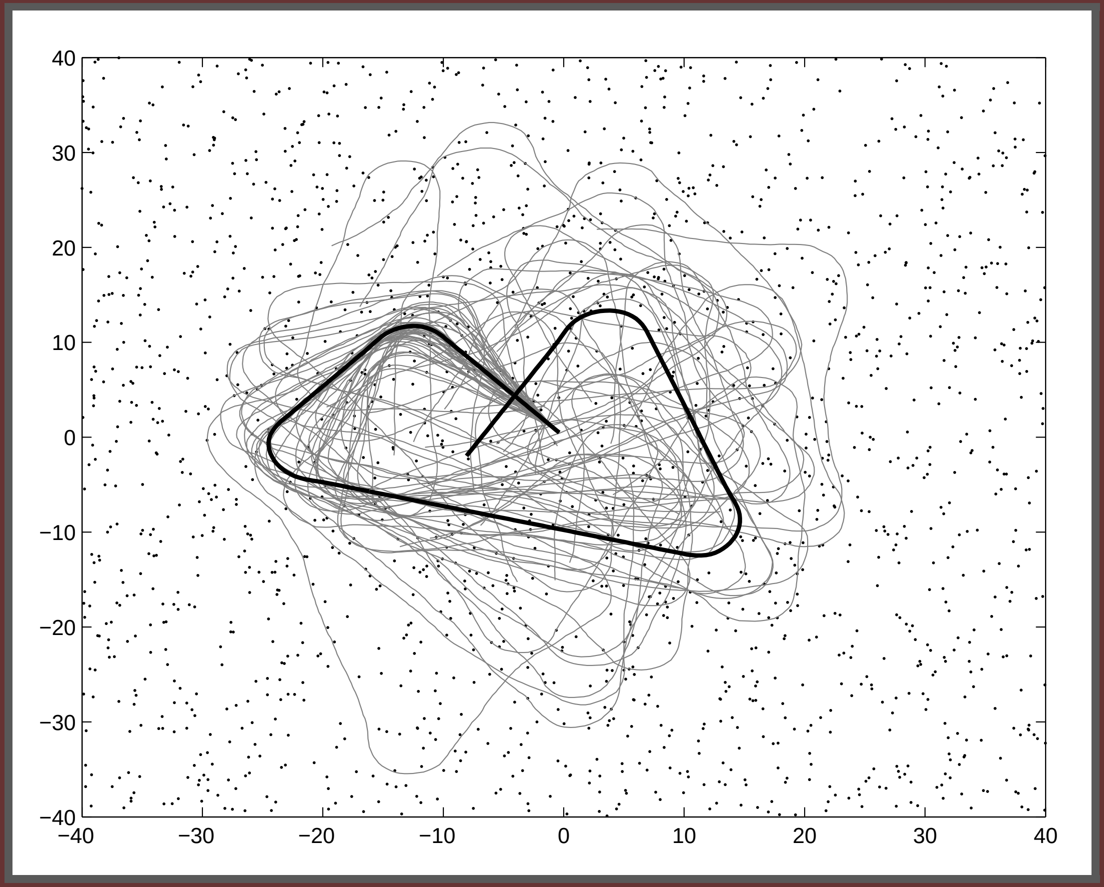
BP(1)

BP(5)

BP(10)



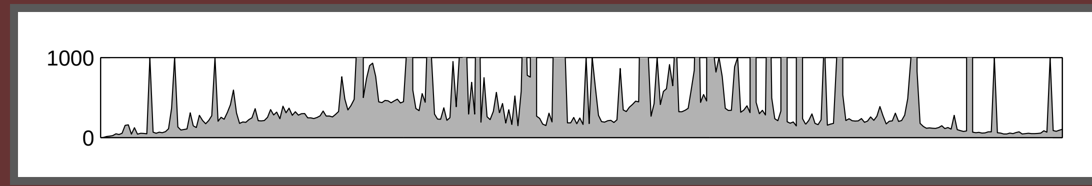
Simulation results: dense environment



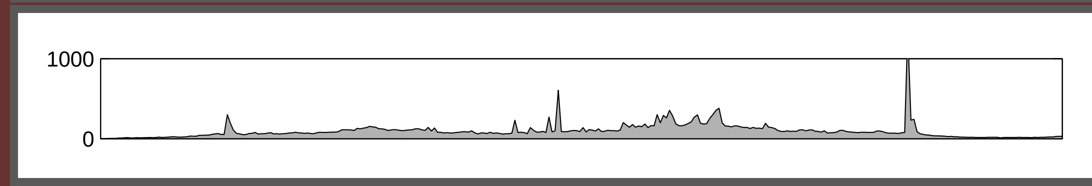
- 63 sec., loop
- 50 Monte Carlo trials averaged for all results

Norm. est. error sq. (NEES): $(\mathbf{x}_t^r - \hat{\mathbf{x}}_t^r)(\hat{\mathbf{P}}_t^r)^{-1}(\mathbf{x}_t^r - \hat{\mathbf{x}}_t^r)^T$

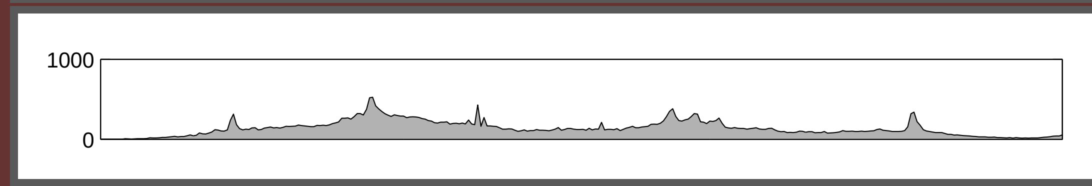
FS2



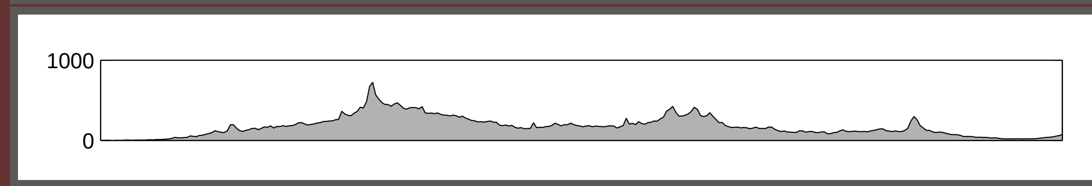
FLR(1)



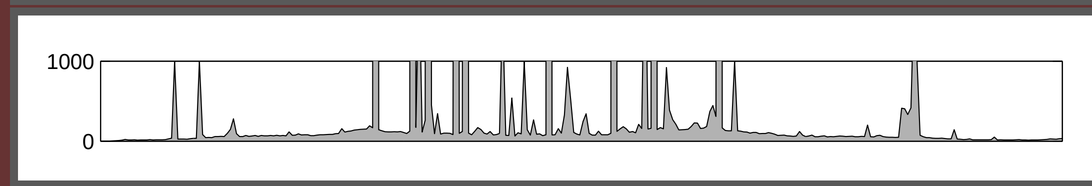
FLR(5)



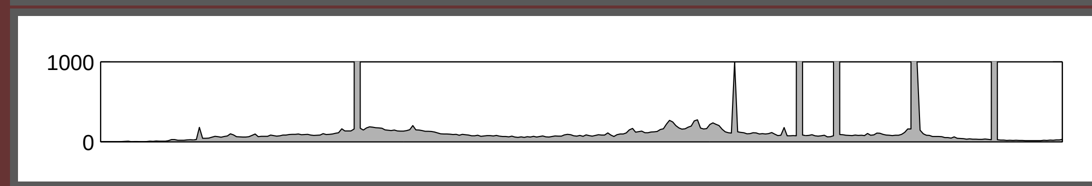
FLR(10)



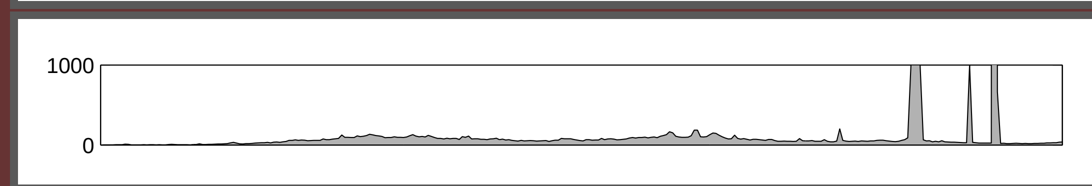
BP(1)



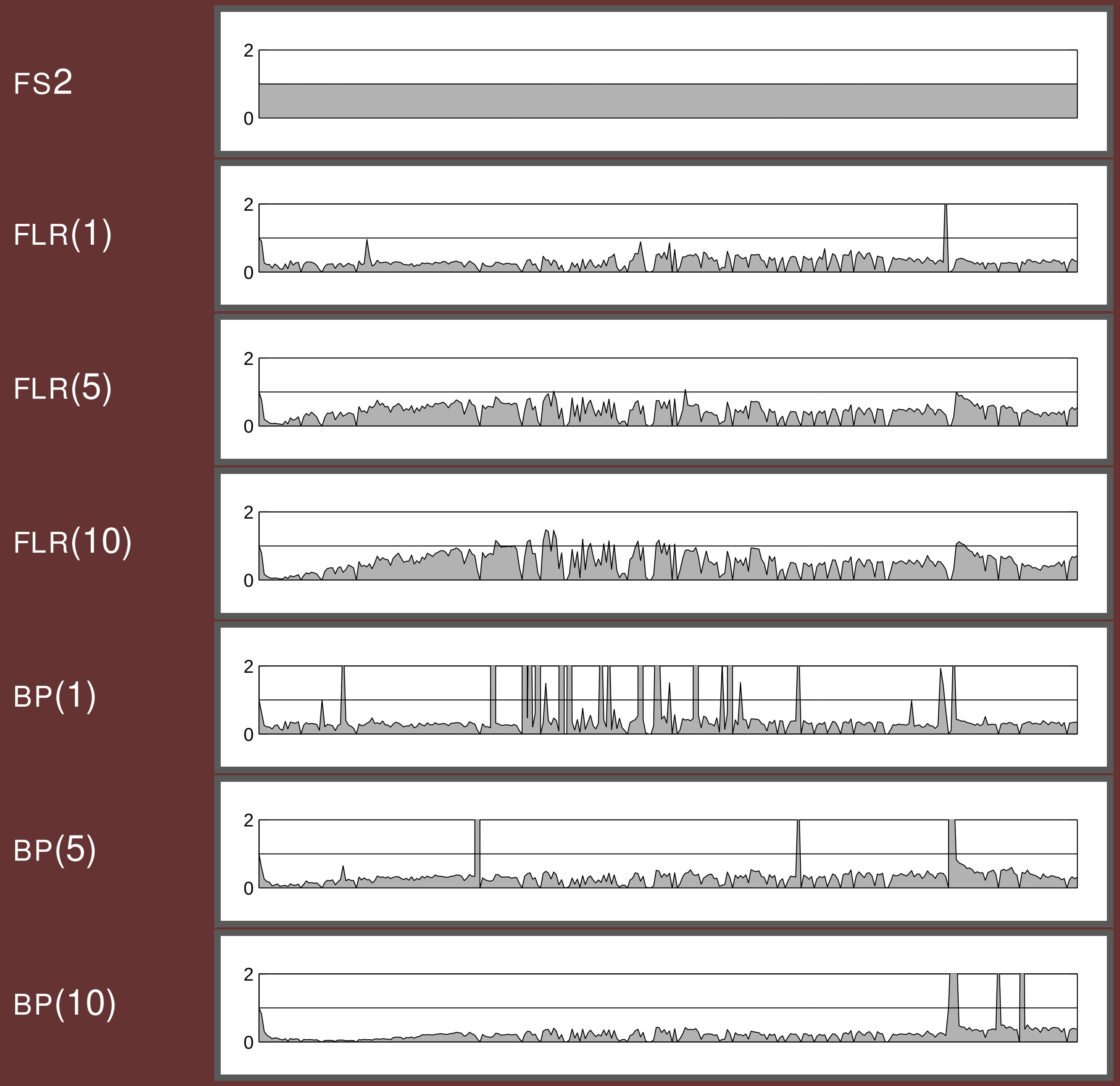
BP(5)



BP(10)



NEES ratio: NEES(alg) / NEES(FS2)



effective particles ($\hat{N}_{\text{eff}}$): $1/\sum_{i=1}^{N}\left(\omega_t^i\right)^2$

FS2

FLR(1)

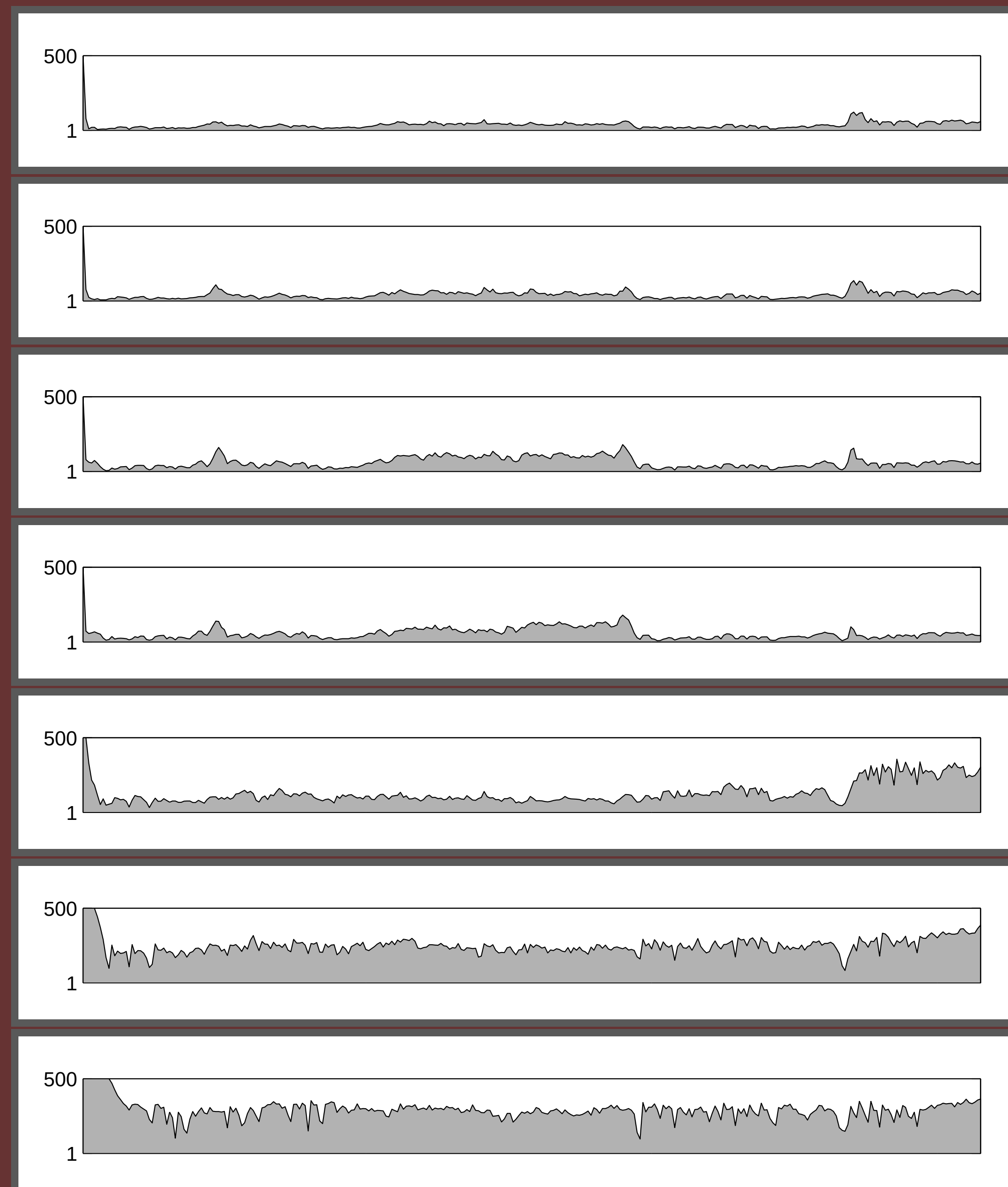
FLR(5)

FLR(10)

BP(1)

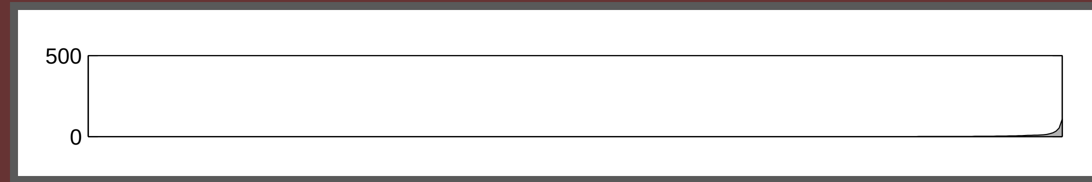
BP(5)

BP(10)

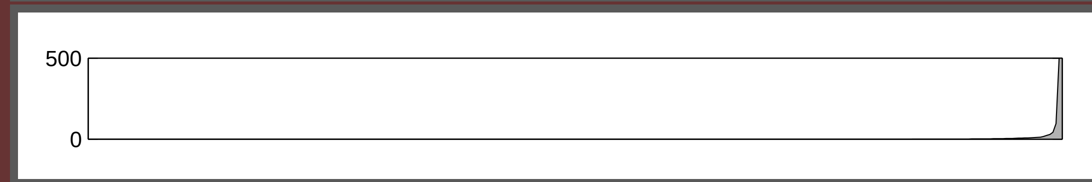


Unique samples of each pose: $|\{\mathbf{x}_k^{r,i} | \mathbf{u}_{1:t}, \mathbf{z}_{1:t}, \mathbf{n}_{1:t}\}|, k = 1 \dots t$

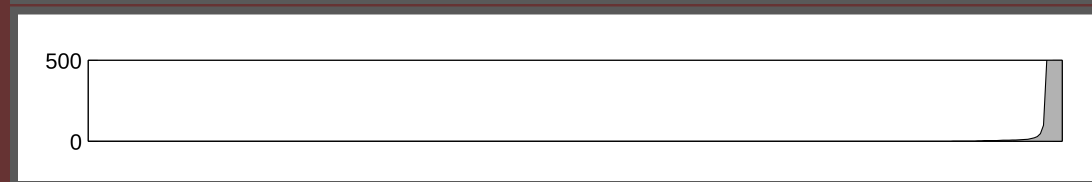
FS2



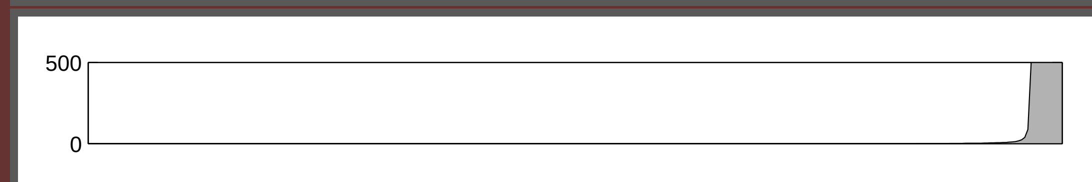
FLR(1)



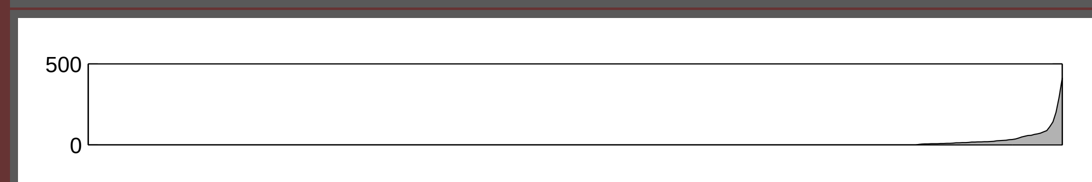
FLR(5)



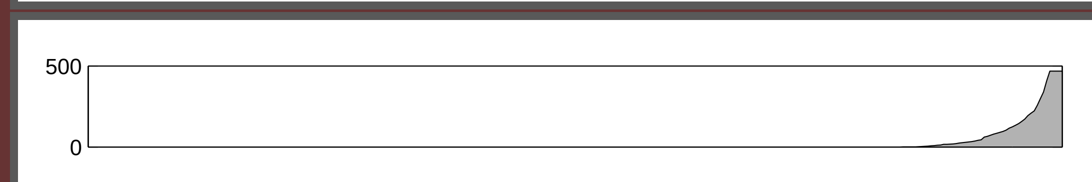
FLR(10)



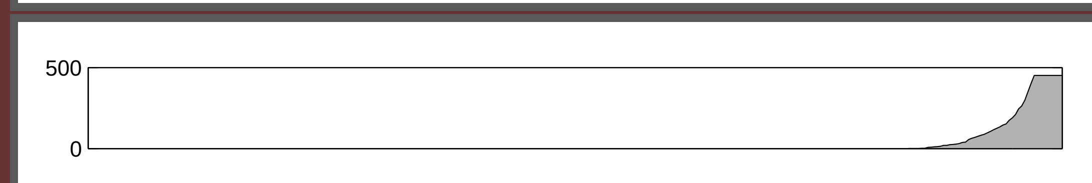
BP(1)



BP(5)



BP(10)



References

E.U. Acar, H. Choset, and P.N. Atkar. Complete sensor-based coverage with extended-range detectors: A hierarchical decomposition in terms of critical points and Voronoi diagrams. In *Proceedings of the 2001 IEEE/RSJ International Conference on Intelligent Robots & Systems*, pages 1305–1311, October 2001.

T. Bailey. Constrained initialisation for bearing-only SLAM. In *Proceedings of the 2003 IEEE International Conference on Robotics & Automation*, pages 1966–1971, September 2003.

T. Bailey, J. Nieto, and E. Nebot. Consistency of the FastSLAM algorithm. In *IEEE Intl. Conf. on Robotics and Automation*, pages 424–427, 2006.

S. Chib. Calculating posterior distributions and modal estimates in Markov mixture models. *Journal of Econometrics*, 75(1):79–97, 1996.

M. Csorba and H.F. Durrant-Whyte. New approach to map building using relative position estimates. *SPIE Navigation and Control Technologies for Unmanned Systems II*, 3087 (1):115–125, 1997.

M. Deans and M. Hebert. Experimental comparison of techniques for localization and mapping using a bearings only sensor. In *Proceedings of the Seventh International Symposium on Experimental Robotics (ISER)*, pages 395–404. Springer-Verlag, December 2000a.

M.C. Deans and M. Hebert. Invariant filtering for simultaneous localization and mapping. In *Proc. IEEE Intl. Conf. Robotics and Automation*, pages 1042–1047, 2000b.

J. Djugash, S. Singh, and P. Corke. Further results with localization and mapping using range from radio. In *Proceedings of the Fifth International Conference on Field and Service Robotics*, Australia, July 2005.

H.F. Durrant-Whyte. Uncertain geometry in robotics. *IEEE Journal of Robotics and Automation*, 4(1):23–31, 1988.

Giorgio Grisetti, Cyrill Stachniss, and Wolfram Burgard. Improving grid-based SLAM with Rao-Blackwellized particle filters by adaptive proposals and selective resampling. In *Proc. of the IEEE Intl. Conf. on Robotics and Automation*, pages 2443–2448, 2005.

Wesley H. Huang and Kristopher R. Beevers. Topological mapping with sensing-limited robots. In M. Erdmann et al., editors, *Algorithmic Foundations of Robotics VI*, pages 235–250. Springer, 2005.

G.A. Kantor and S. Singh. Preliminary results in range-only localization and mapping. In *Proceedings of the 2002 IEEE International Conference on Robotics & Automation*, volume 2, pages 1818–1823, May 2002.

D. Kurth. Range-only robot localization and SLAM with radio. Master’s thesis, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, May 2004.

J.J. Leonard, R. Rikoski, P. Newman, and M. Bosse. Mapping partially observable features from multiple uncertain vantage points. *Intl. Journal of Robotics Research*, 21(10):943–975, October 2002.

J.S. Liu and R. Chen. Blind deconvolution via sequential imputations. *Journal of the American Statistical Association*, 90(430):567–576, June 1995.

M. Montemerlo. *FastSLAM: a factored solution to the simultaneous localization and mapping problem with unknown data association*. PhD thesis, Carnegie Mellon University, Pittsburgh, PA, 2003.

P.M. Newman. *On the structure and solution of the simultaneous localization and mapping problem*. PhD thesis, University of Sydney, Australia, 1999.

D. Rodriguez-Losada, F. Matia, A. Jimenez, and R. Galan. Consistency improvement for SLAM – EKF for indoor environments. In *IEEE Intl. Conf. on Robotics and Automation*, pages 418–423, 2006.

D. Simon and T.L. Chia. Kalman filtering with state equality constraints. *IEEE Transactions on Aerospace and Electronic Systems*, 39:128–136, 2002.

D. Simon and D.L. Simon. Aircraft turbofan engine health estimation using constrained Kalman filtering. In *Proc. ASME Turbo Expo 2003*, number GT2003-38584, 2003.

R. Smith, M. Self, and P. Cheeseman. Estimating uncertain spatial relationships in robotics. In I. Cox and G. Wilfong, editors, *Autonomous Robot Vehicles*. Springer-Verlag, 1990.

J. Solá, A. Monin, M. Devy, and T. Lemaire. Undelayed initialization in bearing only SLAM. In *Proceedings of the 2005 IEEE/RSJ International Conference on Intelligent Robots & Systems*, pages 2499–2504, August 2005.

C. Stachniss, G. Grisetti, and W. Burgard. Recovering particle diversity in a Rao-Blackwellized particle filter for SLAM after actively closing loops. In *Proc. of the IEEE Intl. Conf. on Robotics and Automation*, pages 667–672, 2005.

J.D. Tardós, J. Neira, P.M. Newman, and J.J. Leonard. Robust mapping and localization in indoor environments using SONAR data. *Intl. Journal of Robotics Research*, 21(4): 311–330, April 2002.

B. Tovar, S. LaValle, and R. Murrieta. Optimal navigation and object finding without geometric maps or localization. In *Proceedings of the 2003 IEEE International Conference on Robotics & Automation*, Taipei, Taiwan, September 2003.

W. Wen and H.F. Durrant-Whyte. Model-based multi-sensor data fusion. In *Proc. IEEE Intl. Conf. Robotics and Automation*, pages 1720–1726, 1992.

O. Wijk and H.I. Christensen. Triangulation based fusion of sonar data with application in robot pose tracking. *IEEE Transactions on Robotics and Automation*, 16(6):740–752, 2000.

G. Zunino and H.I. Christensen. Navigation in realistic environments. In M. Devy, editor, *9th Intl. Symp. on Intelligent Robotic Systems*, Toulouse, France, 2001.