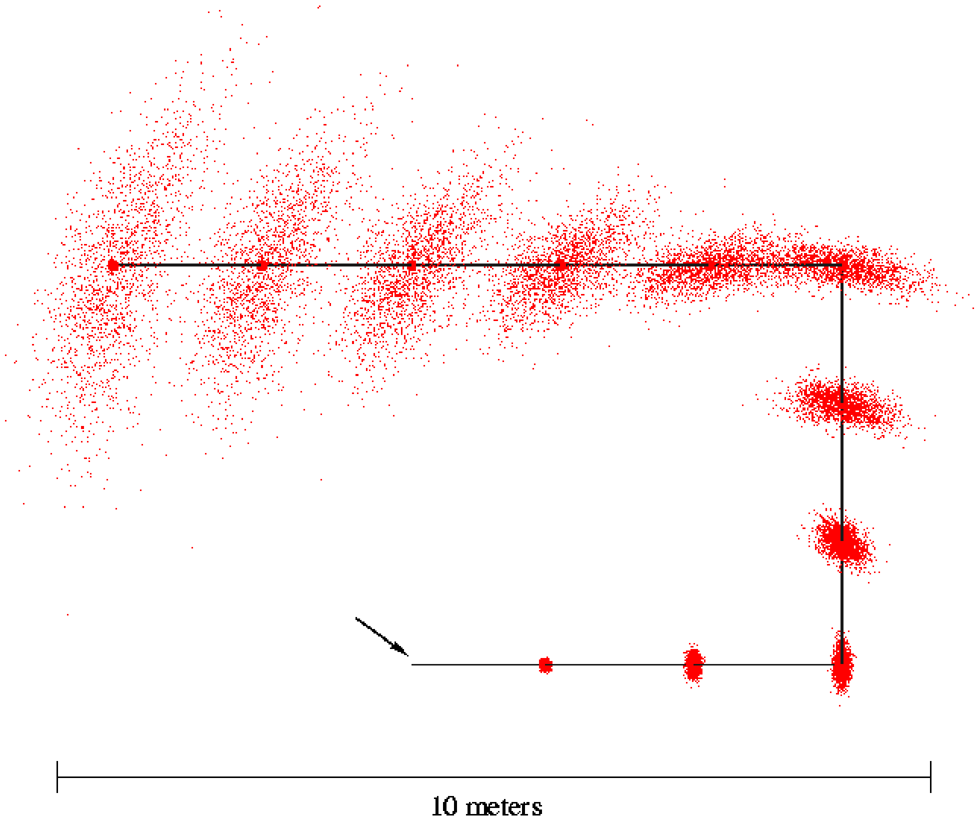
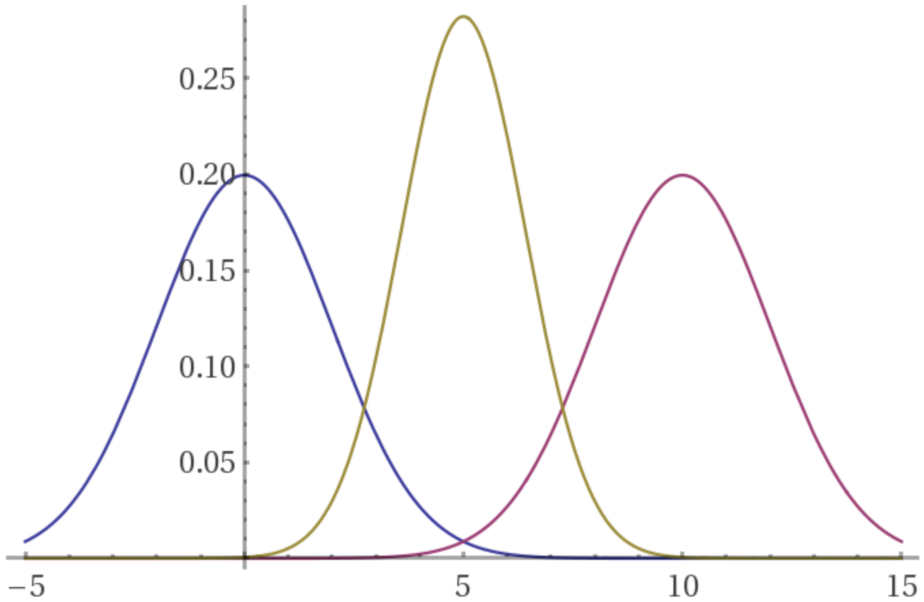
Two Popular Bayesian Estimators: Particle and Kalman Filters

McGill COMP 765

Sept 14th, 2017





Recall: Bayes Filters

z = observation
u = action
x = state

$$\boxed{Bel(x_t)} = P(x_t \mid u_1, z_1 \ldots, u_t, z_t)$$

Bayes $$= \eta \; P(z_t \mid x_t, u_1, z_1, \ldots, u_t) \; P(x_t \mid u_1, z_1, \ldots, u_t)$$

Markov $$= \eta \; P(z_t \mid x_t) \; P(x_t \mid u_1, z_1, \ldots, u_t)$$

Total prob. $$= \eta \; P(z_t \mid x_t) \int P(x_t \mid u_1, z_1, \ldots, u_t, x_{t-1}) \; P(x_{t-1} \mid u_1, z_1, \ldots, u_t) \; dx_{t-1}$$

Markov $$= \eta \; P(z_t \mid x_t) \int P(x_t \mid u_t, x_{t-1}) \; P(x_{t-1} \mid u_1, z_1, \ldots, u_t) \; dx_{t-1}$$

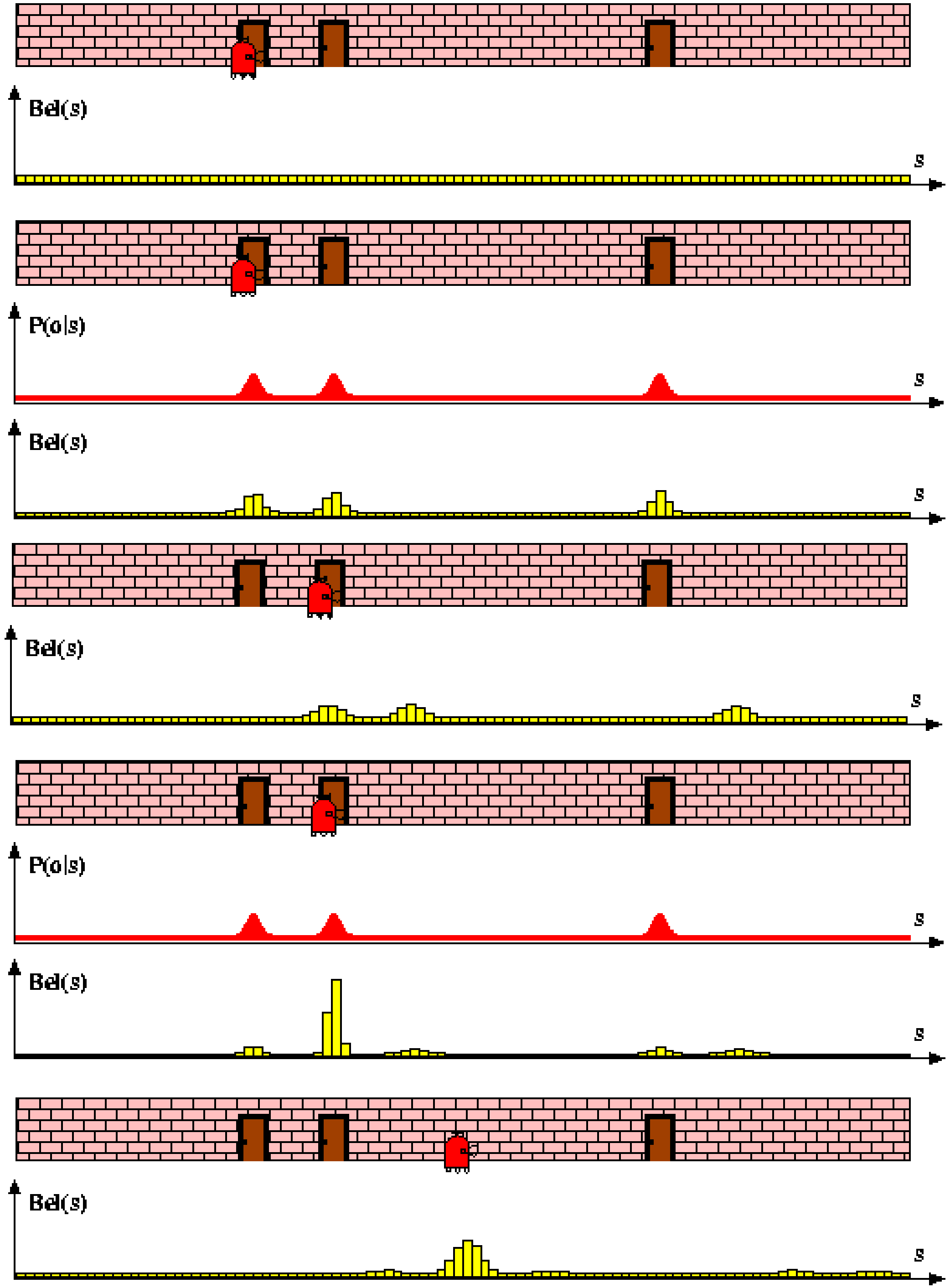
Markov $$= \eta P(z_t \mid x_t) \int P(x_t \mid u_t, x_{t-1}) \, P(x_{t-1} \mid u_1, z_1, \ldots, z_{t-1}) \; dx_{t-1}$$

$$\boxed{= \eta \; P(z_t \mid x_t) \int P(x_t \mid u_t, x_{t-1}) \; Bel(x_{t-1}) \; dx_{t-1}}$$

Discrete Bayes Filter Algorithm

1. Algorithm **Discrete_Bayes_filter**(*Bel(x),d*):
2. η=0
3. If *d* is a perceptual data item *z* then
4. For all *x* do
5. $$Bel'(x) = P(z \mid x) Bel(x)$$
6. $$\eta = \eta + Bel'(x)$$
7. For all *x* do
8. $$Bel'(x) = \eta^{-1} Bel'(x)$$
9. Else if *d* is an action data item *u* then
10. For all *x* do
11. $$Bel'(x) = \sum_{x'} P(x \mid u, x')\, Bel(x')$$
12. Return *Bel'(x)*

Piecewise Constant Bel(x)



Problem Statement

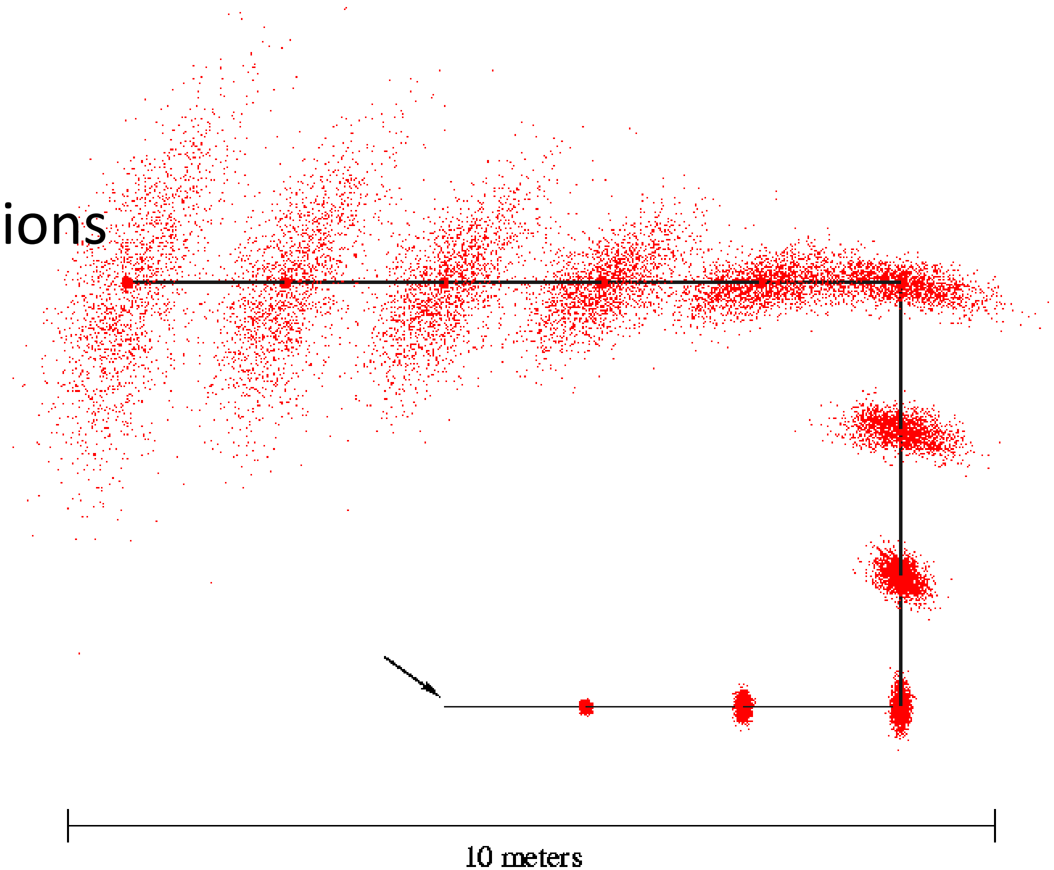
- What are representations for Bel(x) and matching update rules work well in practice?

$$Bel(x_t) = \eta \; P(z_t \mid x_t) \int P(x_t \mid u_t, x_{t-1}) \; Bel(x_{t-1}) \; dx_{t-1}$$

- Desirable:
  - Accuracy and correctness
  - Time and space usage scales well with size of state and # dimensions
  - Represent realistic range of motion and measurement models

Part 1: Particle Filters

- Intuition: track Bel(x) with adaptively located discrete samples
- Potentials:
  - Better accuracy/computation trade-off
  - Particles can take shape of arbitrary distributions
- Uses:
  - Indoor robotics
  - Self driving cars
  - Computer vision
  - General tool in learning



Probabilistic Algorithms and the Interactive Museum Tour-Guide Robot Minerva

S. Thrun[1], M. Beetz[3], M. Bennewitz[2], W. Burgard[2], A.B. Cremers[3], F. Dellaert[1]
D. Fox[1], D. Hähnel[2], C. Rosenberg[1], N. Roy[1], J. Schulte[1], D. Schulz[3]

[1]School of Computer Science
Carnegie Mellon University
Pittsburgh, PA

[2]Computer Science Dept.
University of Freiburg
Freiburg, Germany

[3]Computer Science Dept. III
University of Bonn
Bonn, Germany

Figure 2: (a) Minerva. (b) Minerva gives a tour in the Smithsonian's National Museum of American History. (c) Interaction with museum visitors.

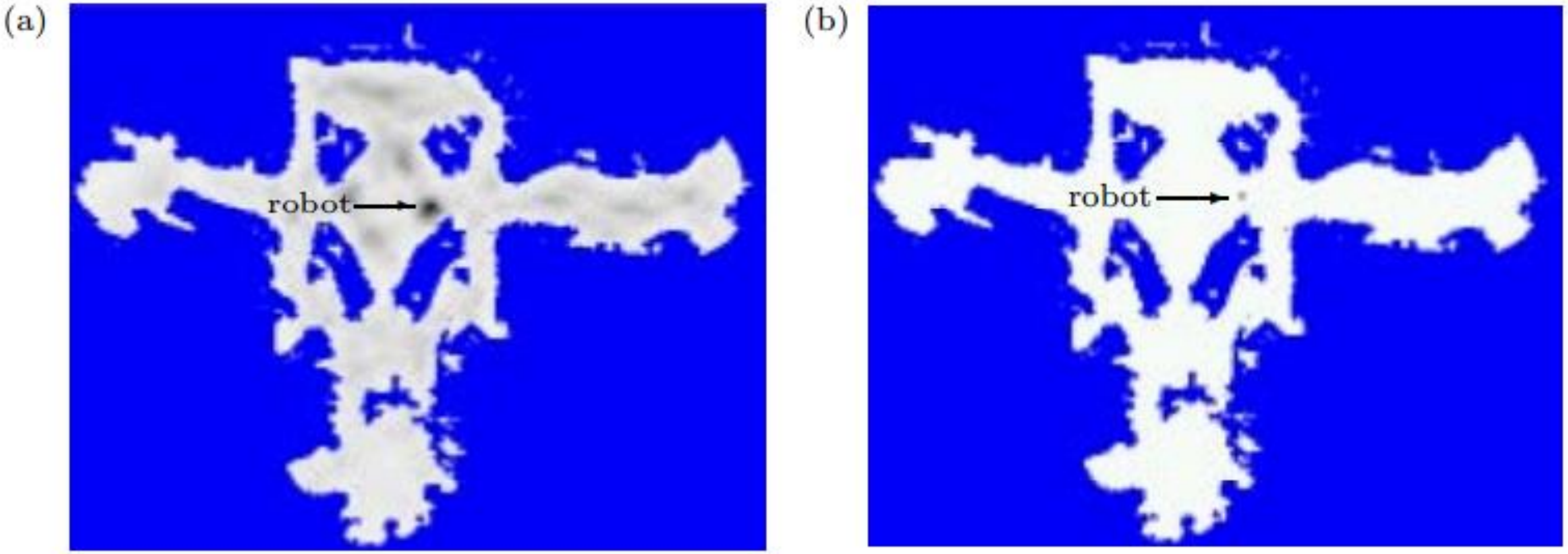
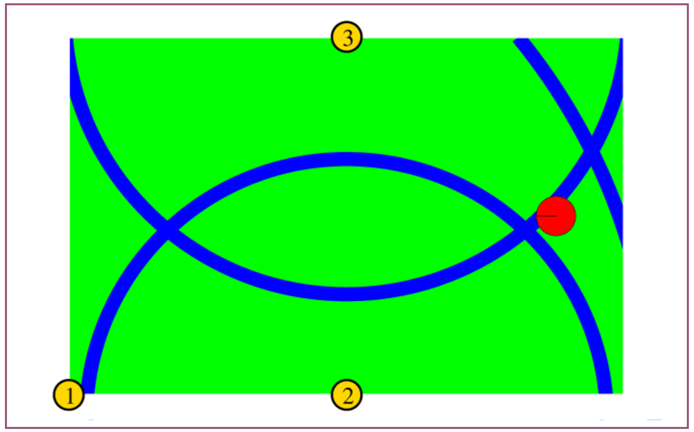


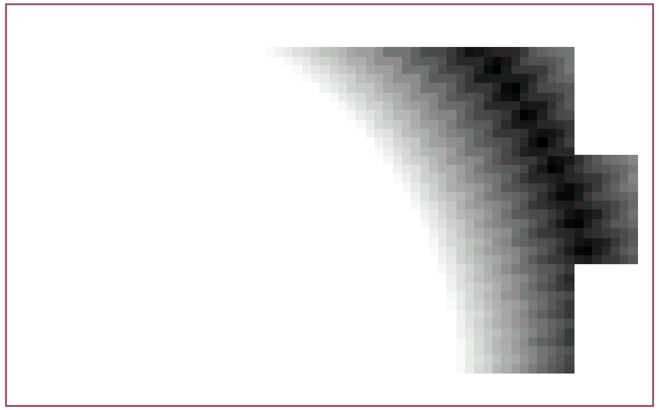
Figure 5: Global localization: (a) Pose posterior $b_t(s_t)$ after integrating a first laser scan (projected into 2D). The darker a pose, the more likely it is. (b) shows $b_t(s_t)$ after integrating a second sensor scan. Now the robot knows its pose with high certainty/accuracy.

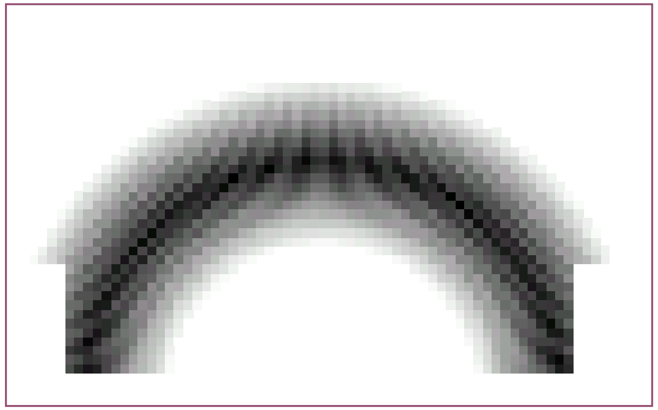
Intuitive Example: Localizing During Robocup

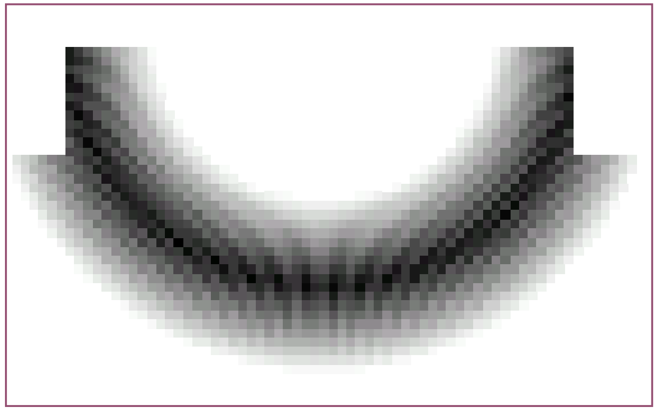
Distributions

Consider distributions to each $p(x|\ z_i)$ *only.* Are these related to our answer?

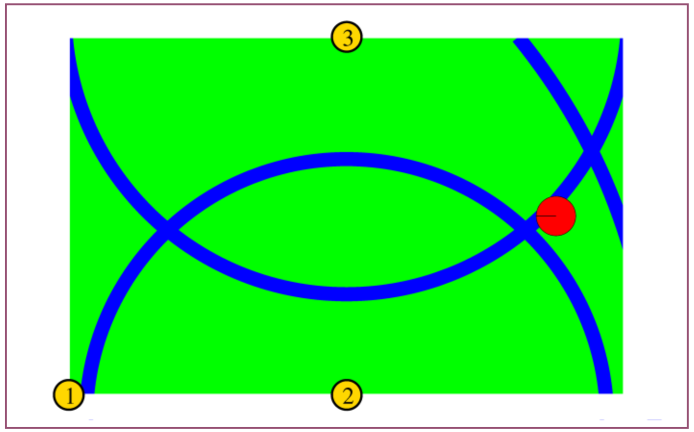


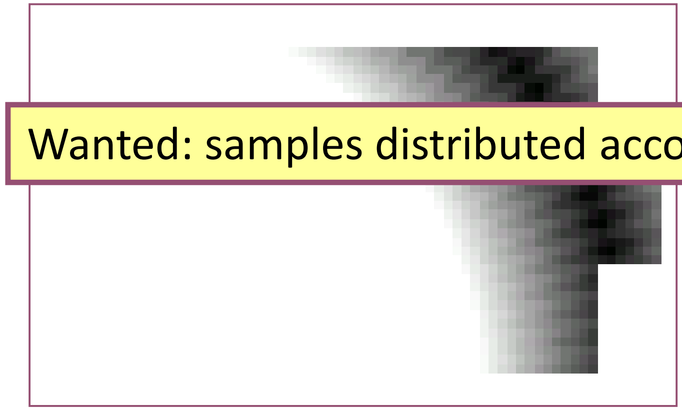


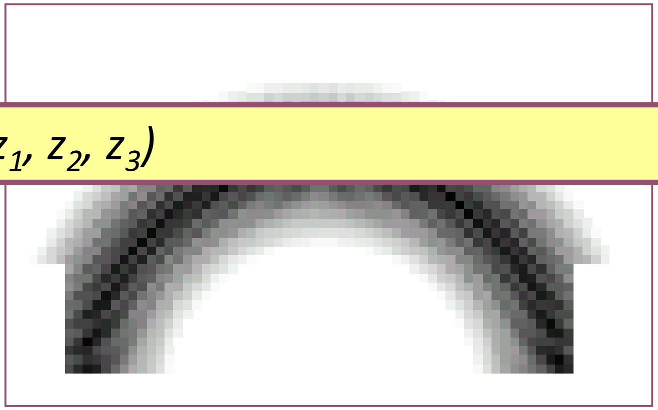




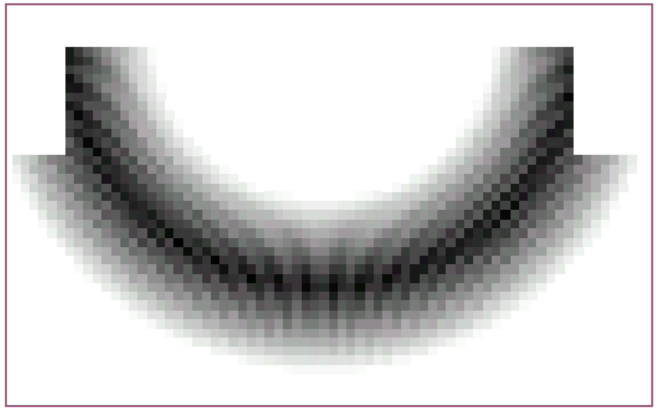
Distributions





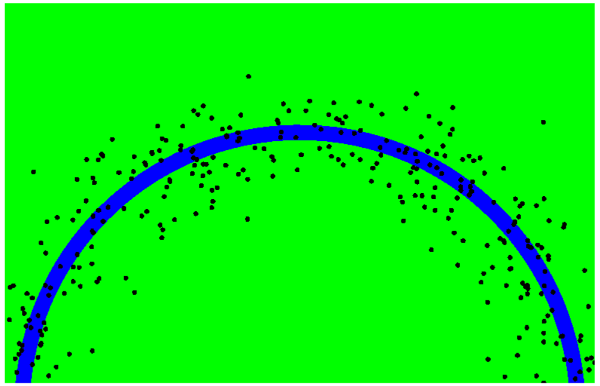


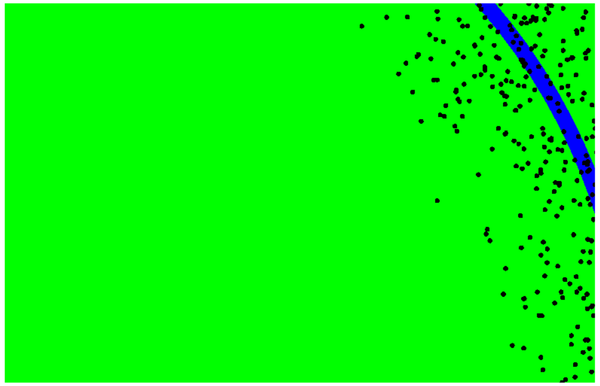
Wanted: samples distributed according to $p(x|\ z_1, z_2, z_3)$

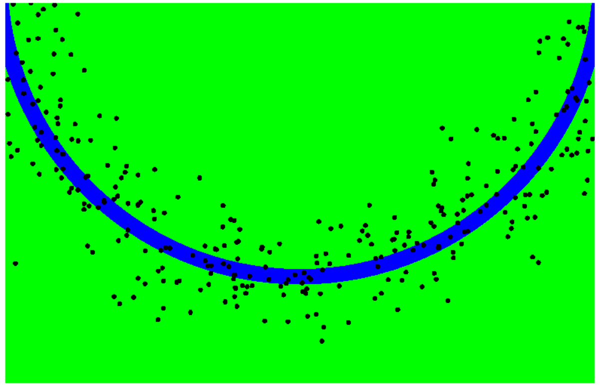


This is Easy!

We can draw samples from $p(x|z_l)$ by adding noise to the detection parameters.

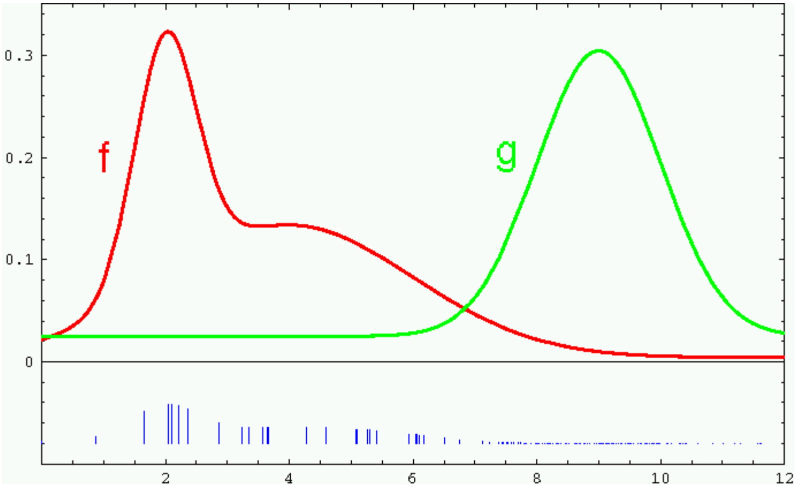






Importance Sampling

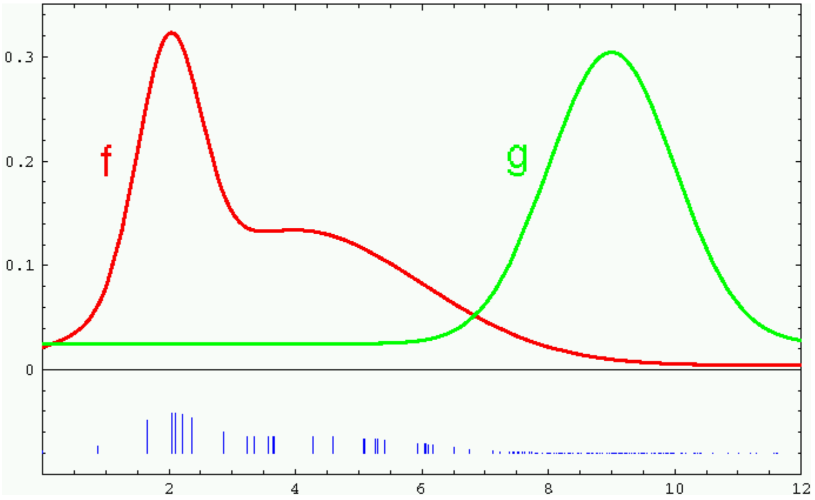
- As seen, it is often easy to draw samples from one portion of our Bayes filter
- Main trick: **importance sampling**, i.e. how to estimate properties/statistics of one distribution (f) given samples from another distribution (g)



For example, suppose we want to estimate the expected value of f given only samples from g.

Importance Sampling

- As seen, it is often easy to draw samples from one portion of our Bayes filter
- Main trick: **importance sampling**, i.e. how to estimate properties/statistics of one distribution (f) given samples from another distribution (g)

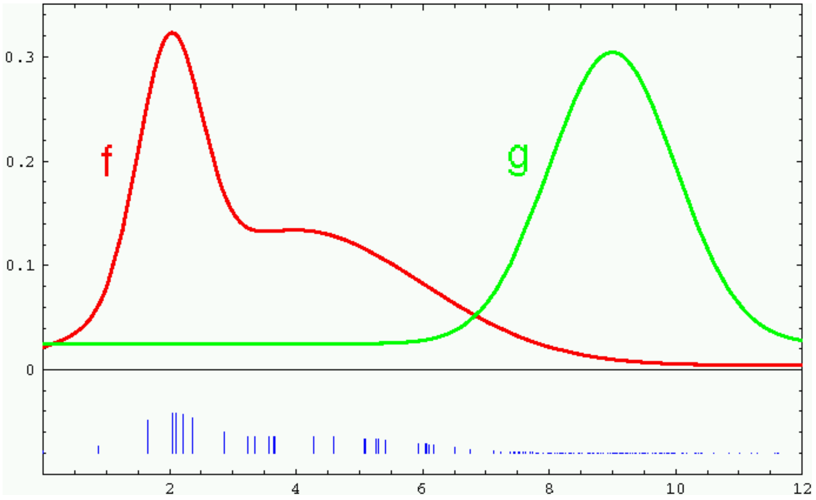


$$
\begin{aligned}
\mathbb{E}_{x\sim f(x)}[x] &= \int x f(x) dx \\
&= \int \frac{g(x)}{g(x)} x f(x) dx \\
&= \int \frac{x f(x)}{g(x)} g(x) dx \\
&= \mathbb{E}_{x\sim g(x)}[x \frac{f(x)}{g(x)}] \\
&= \mathbb{E}_{x\sim g(x)}[x \; w(x)]
\end{aligned}
$$

Importance Sampling

- As seen, it is often easy to draw samples from one portion of our Bayes filter

- Main trick: **importance sampling**, i.e. how to estimate properties/statistics of one distribution (f) given samples from another distribution (g)



$$
\begin{aligned}
\mathbb{E}_{x\sim f(x)}[x] &= \int x f(x) dx \\
&= \int \frac{g(x)}{g(x)} x f(x) dx \\
&= \int \frac{x f(x)}{g(x)} g(x) dx \\
&= \mathbb{E}_{x\sim g(x)}[x \frac{f(x)}{g(x)}] \\
&= \mathbb{E}_{x\sim g(x)}[x\ w(x)]
\end{aligned}
$$

Weights describe the mismatch between the two distributions, i.e. how to reweigh samples to obtain statistics of f from samples of g

Importance Sampling for Robocup

$$\text{Target distribution f}: p(x \mid z_1, z_2, \ldots, z_n) = \frac{\prod_k p(z_k \mid x) \quad p(x)}{p(z_1, z_2, \ldots, z_n)}$$

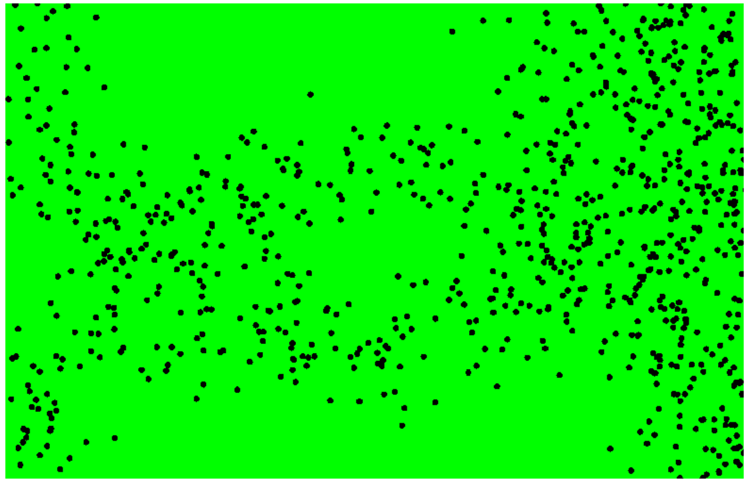
$$\text{Sampling distribution g}: p(x \mid z_l) = \frac{p(z_l \mid x)\, p(x)}{p(z_l)}$$

$$\text{Importance weights w}: \frac{f}{g} = \frac{p(x \mid z_1, z_2, \ldots, z_n)}{p(x \mid z_l)} = \frac{p(z_l) \prod_{k \neq l} p(z_k \mid x)}{p(z_1, z_2, \ldots, z_n)}$$

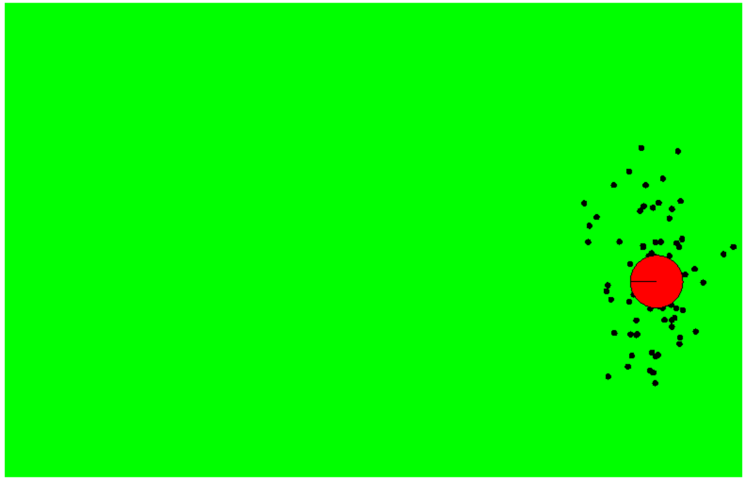
Importance Sampling

Here are all of our p(x|zi) samples, now with w attached (not shown).

If we re-draw from these samples, weighted by w, we get…



Weighted samples



After resampling

Importance Sampling for Bayes Filter

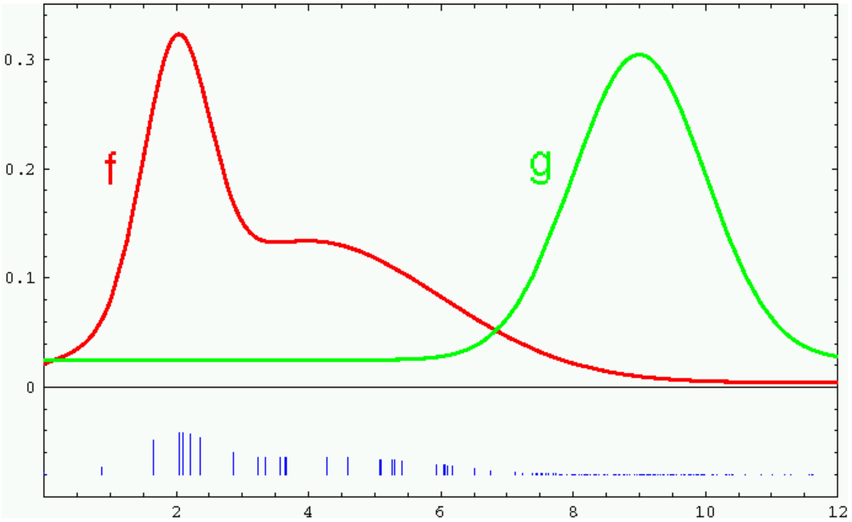
- What is are the proposal distribution and weighting computations?

$$g(x_t) = p(x_t|z_{0:t-1}, u_{0:t-1}) = \overline{bel}(x_t)$$

Sample from propagation, before update

$$f(x_t) = p(x_t|z_{0:t}, u_{0:t-1}) = bel(x_t)$$

Want posterior belief after update



Recall: weighting to remove sample bias

$$\begin{aligned}
\mathbb{E}_{x\sim f(x)}[x] &= \int x f(x) dx \\
&= \int \frac{g(x)}{g(x)} x f(x) dx \\
&= \int \frac{x f(x)}{g(x)} g(x) dx \\
&= \mathbb{E}_{x\sim g(x)}[x \frac{f(x)}{g(x)}] \\
&= \mathbb{E}_{x\sim g(x)}[x \; w(x)]
\end{aligned}$$

Importance Sampling for Bayes Filter

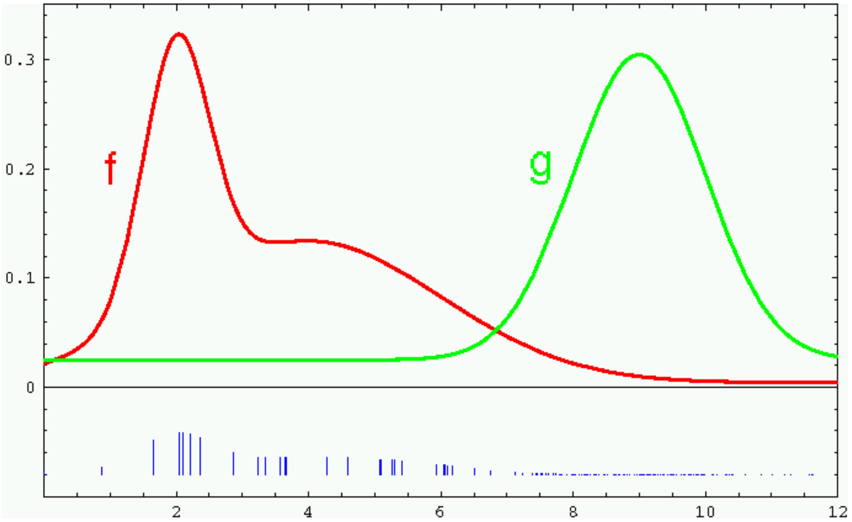
- What is are the proposal distribution and weighting computations?

$$g(x_t) = p(x_t|z_{0:t-1}, u_{0:t-1}) = \overline{bel}(x_t)$$

Sample from propagation, before update

$$f(x_t) = p(x_t|z_{0:t}, u_{0:t-1}) = bel(x_t)$$

Want posterior belief after update



$$\begin{aligned} w(x_t^{[m]}) &= \frac{f(x_t^{[m]})}{g(x_t^{[m]})} \\ &\propto \frac{p(z_t|x_t^{[m]})\, p(x_t^{[m]}|x_{t-1}^{[m]}, u_{t-1})\, bel(x_{t-1}^{[m]})}{p(x_t^{[m]}|x_{t-1}^{[m]}, u_{t-1})\, bel(x_{t-1}^{[m]})} \\ &\propto p(z_t|x_t^{[m]}) \end{aligned}$$

This algorithm is known as a particle filter.

Particle Filter Algorithm

Actual observation and control received

$\text{ParticleFilter}(\bar{z}_t, u_{t-1})$

$\bar{S}_t = \{\} \quad \bar{W}_t = \{\}$

for particle index $m = 1 \ldots M$

sample $x_t^{[m]} \sim p(x_t | x_{t-1}^{[m]}, u_{t-1})$

$w_t^{[m]} = p(\bar{z}_t | x_t^{[m]})$

$\bar{S}_t.\text{append}(x_t^{[m]})$

$\bar{W}_t.\text{append}(w_t^{[m]})$

$S_t = \{\}$

for particle index $m = 1 \ldots M$

sample particle i from $\bar{S}_t$ with probability $\propto w_t^{[i]}$

$S_t.\text{append}(x_t^{[m]})$

return S_t

Particle Filter Algorithm

ParticleFilter$(\bar{z}_t, u_{t-1})$

$\bar{S}_t = \{\} \quad \bar{W}_t = \{\}$

for particle index $\quad m = 1 \ldots M$

sample $\quad x_t^{[m]} \sim p(x_t | x_{t-1}^{[m]}, u_{t-1})$

$w_t^{[m]} = p(\bar{z}_t | x_t^{[m]})$

$\bar{S}_t.\text{append}(x_t^{[m]})$

$\bar{W}_t.\text{append}(w_t^{[m]})$

$S_t = \{\}$

for particle index $\quad m = 1 \ldots M$

sample particle i from $\bar{S}_t$ with probability $\propto w_t^{[i]}$

$S_t.\text{append}(x_t^{[m]})$

return S_t

Particle propagation/prediction:
noise needs to be added in order to make particles differentiate from each other.

If propagation is deterministic then particles are going to collapse to a single particle after a few resampling steps.

Particle Filter Algorithm

ParticleFilter$(\bar{z}_t, u_{t-1})$

$\bar{S}_t = \{\} \quad \bar{W}_t = \{\}$

for particle index $\quad m = 1 \ldots M$

sample $\quad x_t^{[m]} \sim p(x_t | x_{t-1}^{[m]}, u_{t-1})$

$w_t^{[m]} = p(\bar{z}_t | x_t^{[m]})$

Weight computation as measurement likelihood.
For each particle we compute the probability of the
actual observation given the state is at that particle.

$\bar{S}_t.\text{append}(x_t^{[m]})$

$\bar{W}_t.\text{append}(w_t^{[m]})$

$S_t = \{\}$

for particle index $\quad m = 1 \ldots M$

sample particle i from $\bar{S}_t$ with probability $\propto w_t^{[i]}$

$S_t.\text{append}(x_t^{[m]})$

return S_t

Particle Filter Algorithm

$\text{ParticleFilter}(\bar{z}_t, u_{t-1})$

$\bar{S}_t = \{\} \quad \bar{W}_t = \{\}$

for particle index $m = 1...M$

sample $x_t^{[m]} \sim p(x_t | x_{t-1}^{[m]}, u_{t-1})$

$w_t^{[m]} = p(\bar{z}_t | x_t^{[m]})$

$\bar{S}_t.\text{append}(x_t^{[m]})$

$\bar{W}_t.\text{append}(w_t^{[m]})$

$S_t = \{\}$

for particle index $m = 1...M$

sample particle i from $\bar{S}_t$ with probability $\propto w_t^{[i]}$

$S_t.\text{append}(x_t^{[m]})$

return S_t

Resampling step

Note: particle deprivation heuristics are not shown here

Particle Filter Algorithm

$\text{ParticleFilter}(\bar{z}_t, u_{t-1})$

$\bar{S}_t = \{\} \quad \bar{W}_t = \{\}$

for particle index $m = 1 \ldots M$

sample $x_t^{[m]} \sim p(x_t | x_{t-1}^{[m]}, u_{t-1})$

$w_t^{[m]} = p(\bar{z}_t | x_t^{[m]})$

$\bar{S}_t.\text{append}(x_t^{[m]})$

$\bar{W}_t.\text{append}(w_t^{[m]})$

$S_t = \{\}$

for particle index $m = 1 \ldots M$

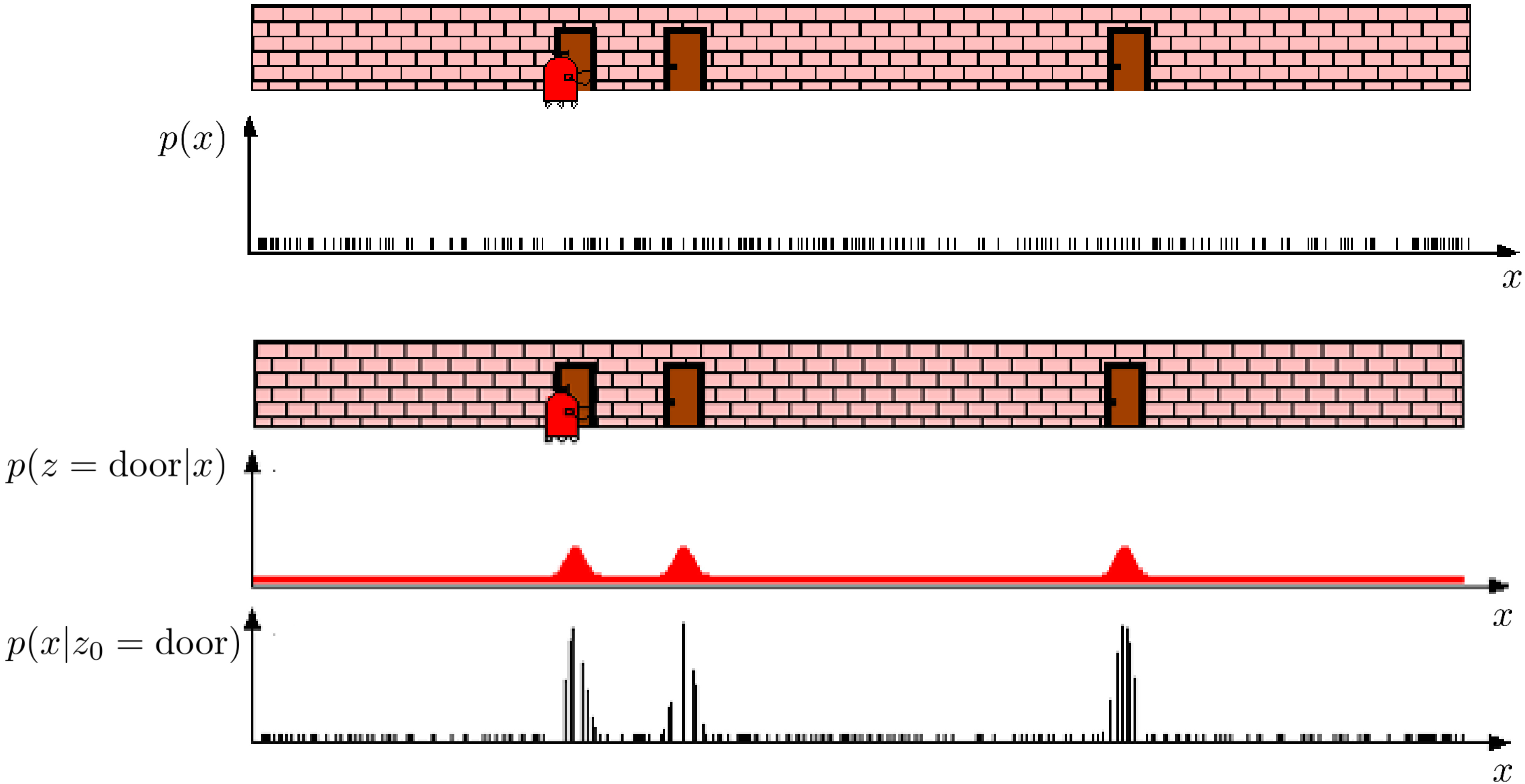
Resampling: The particle locations now have a chance to adapt according to the weights. More likely particles persist, while unlikely choices are removed.

sample particle i from $\bar{S}_t$ with probability $\propto w_t^{[i]}$

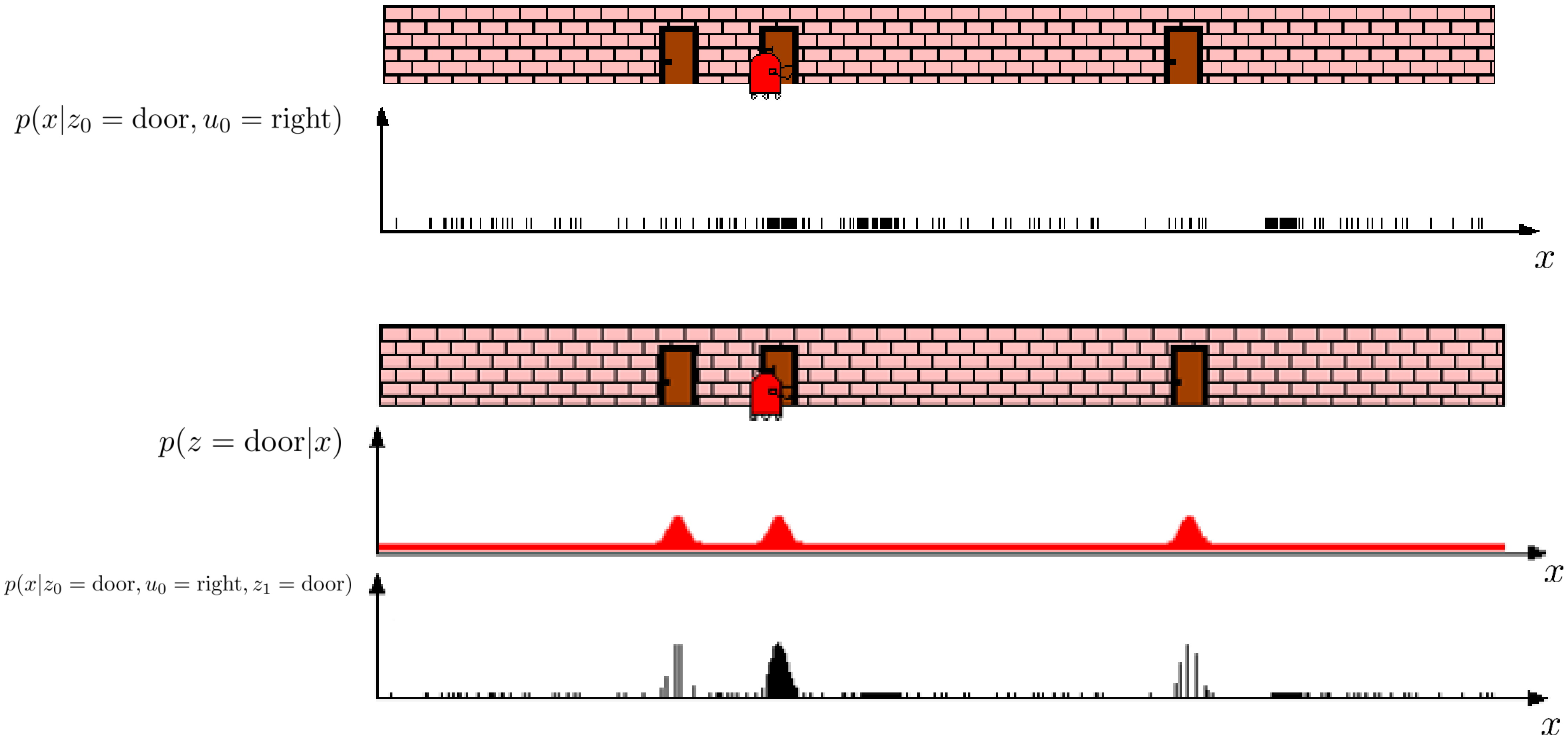
$S_t.\text{append}(x_t^{[m]})$

return S_t

Examples: 1D Localization



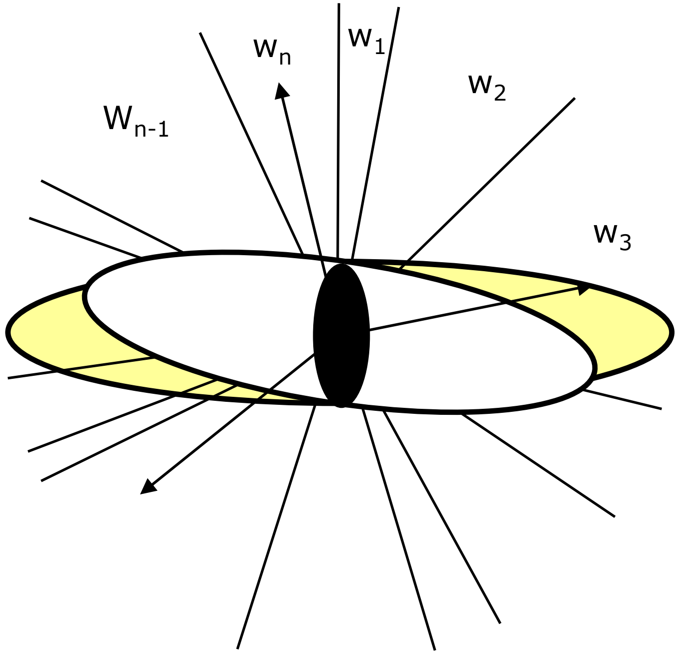
Examples: 1D Localization

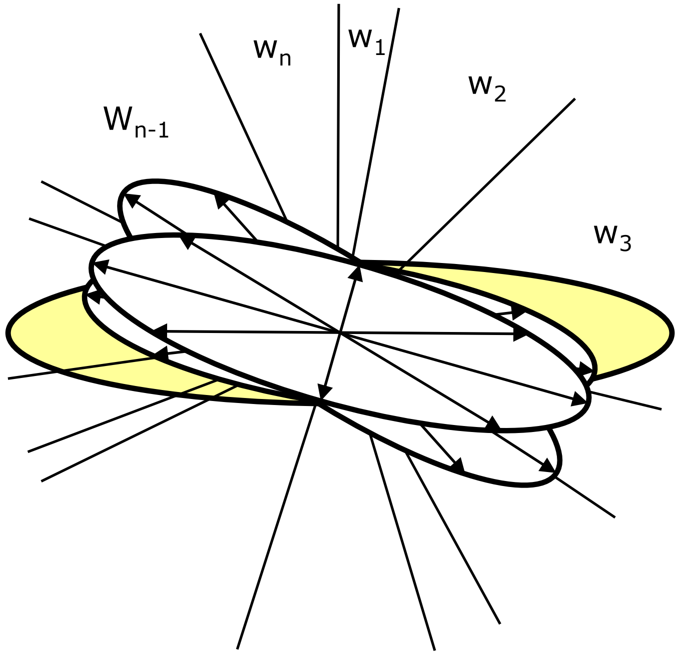


Resampling

- **Given**: Set S of weighted samples.
- **Wanted** : Random sample, where the probability of drawing x_i is given by w_i.
- Typically done n times with replacement to generate new sample set S'.

Resampling Carefully





- Roulette wheel
- Binary search, n log n

- Stochastic universal sampling
- Systematic resampling
- Linear time complexity
- Easy to implement, low variance

Resampling Algorithm

Algorithm **systematic_resampling**(S,n):

$S' = \emptyset, c_1 = w^1$

For $i = 2 \ldots n$ — *Generate cdf*

$\quad c_i = c_{i-1} + w^i$

$u_1 \sim U]0, n^{-1}], i = 1$ — *Initialize threshold*

For $j = 1 \ldots n$ — *Draw samples …*

$\quad$ **While** ($u_j > c_i$) — *Skip until next threshold reached*

$\quad\quad i = i + 1$

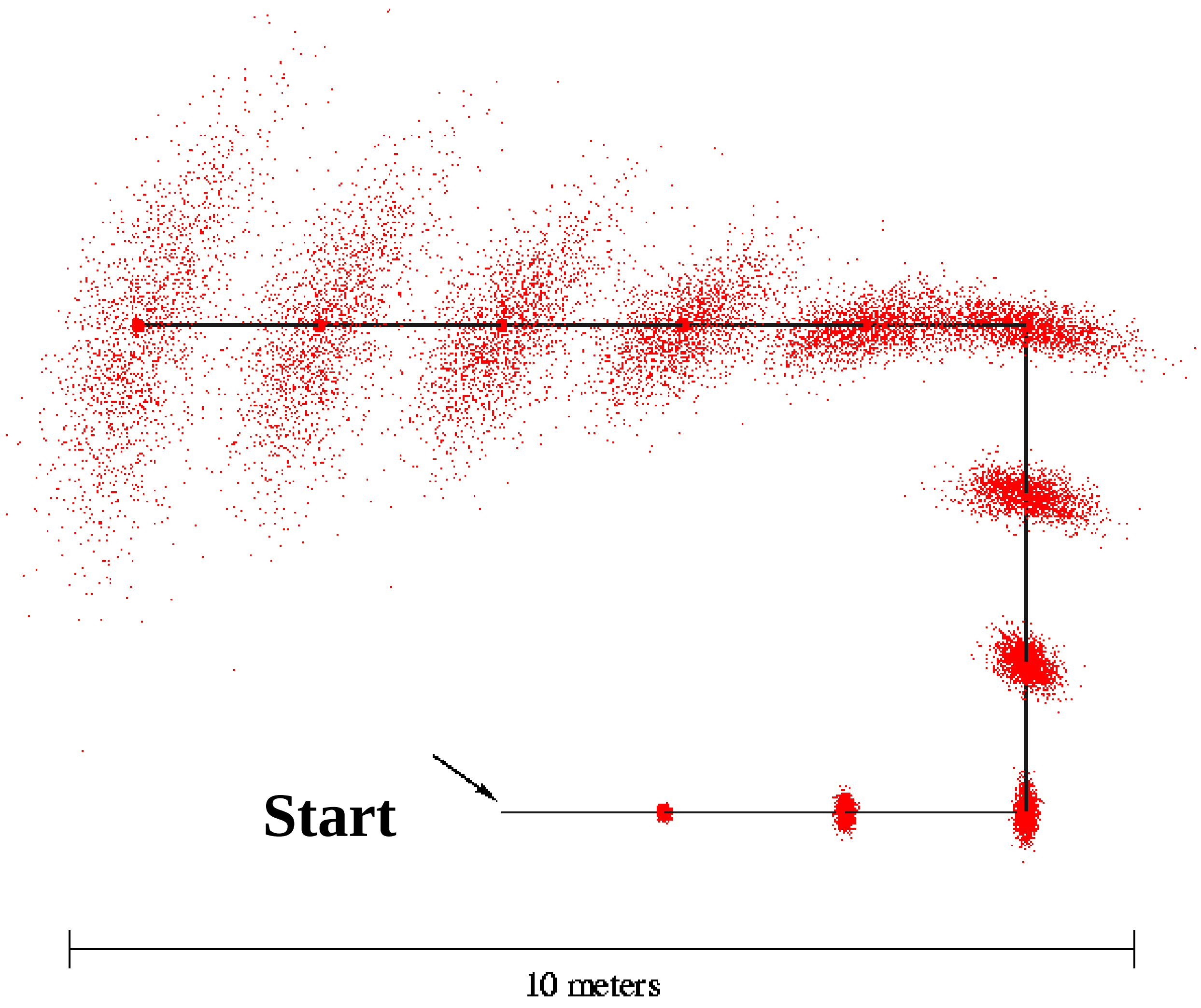
$\quad S' = S' \cup \left\{ < x^i, n^{-1} > \right\}$ — *Insert*

$\quad u_{j+1} = u_j + n^{-1}$ — *Increment threshold*

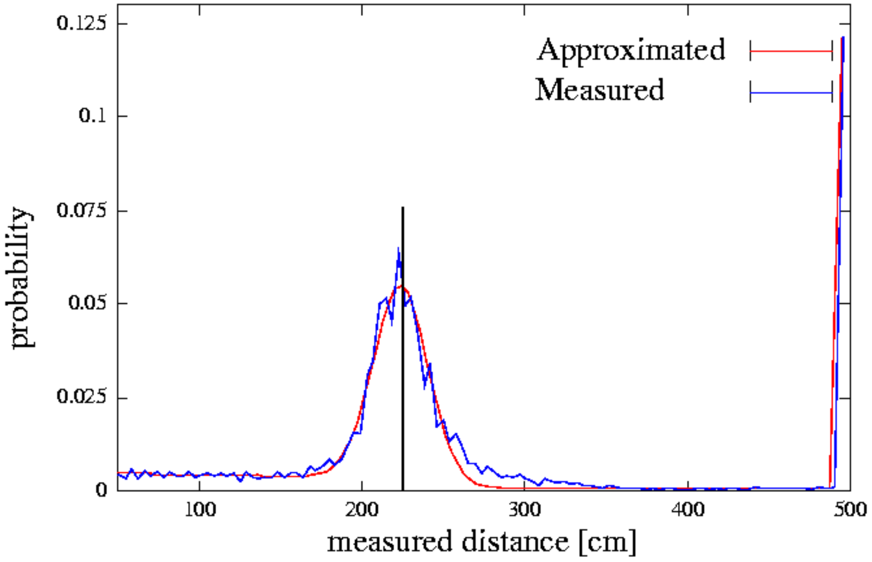
Return S'

Also called **stochastic universal sampling**

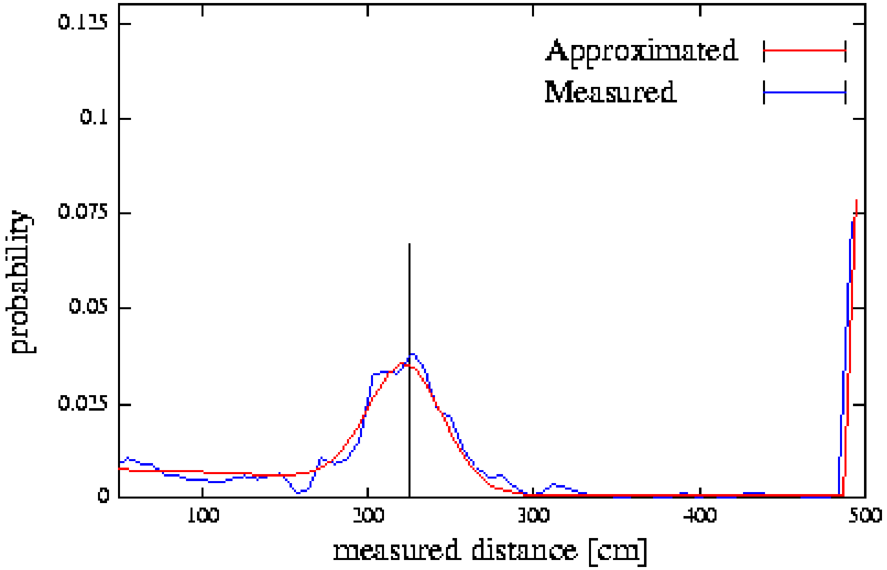
Particle Motion Model



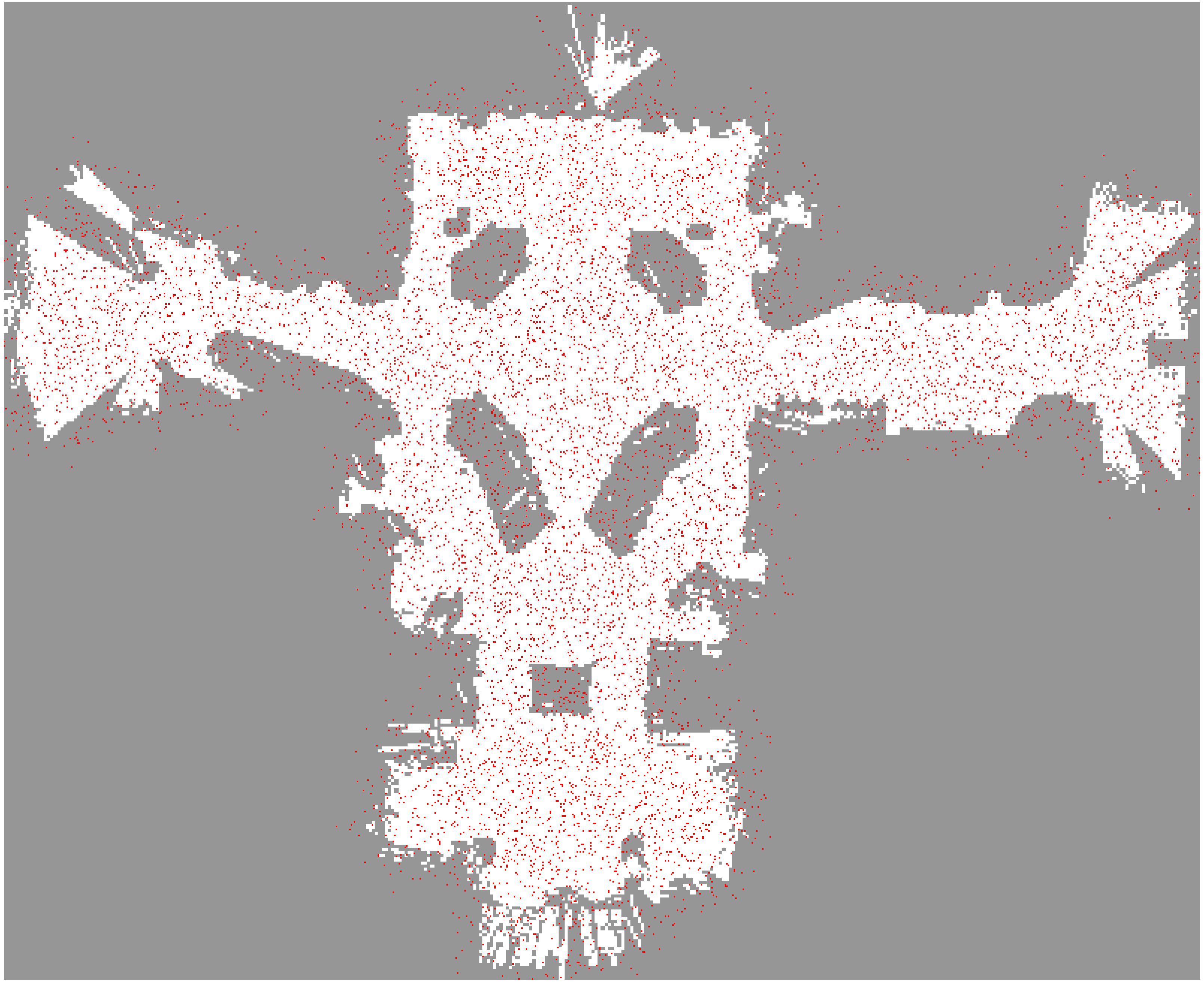
Proximity Sensor Model Reminder

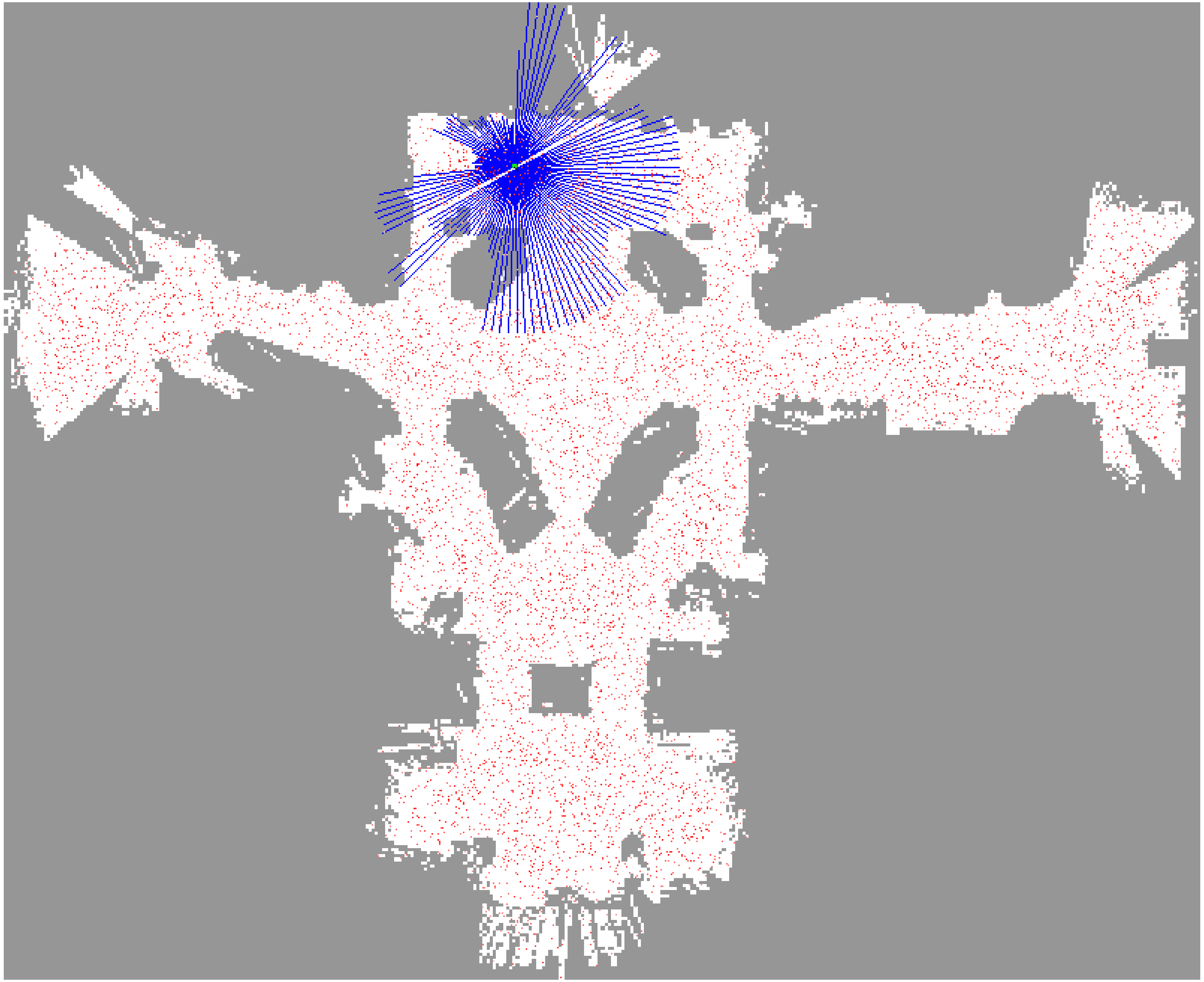


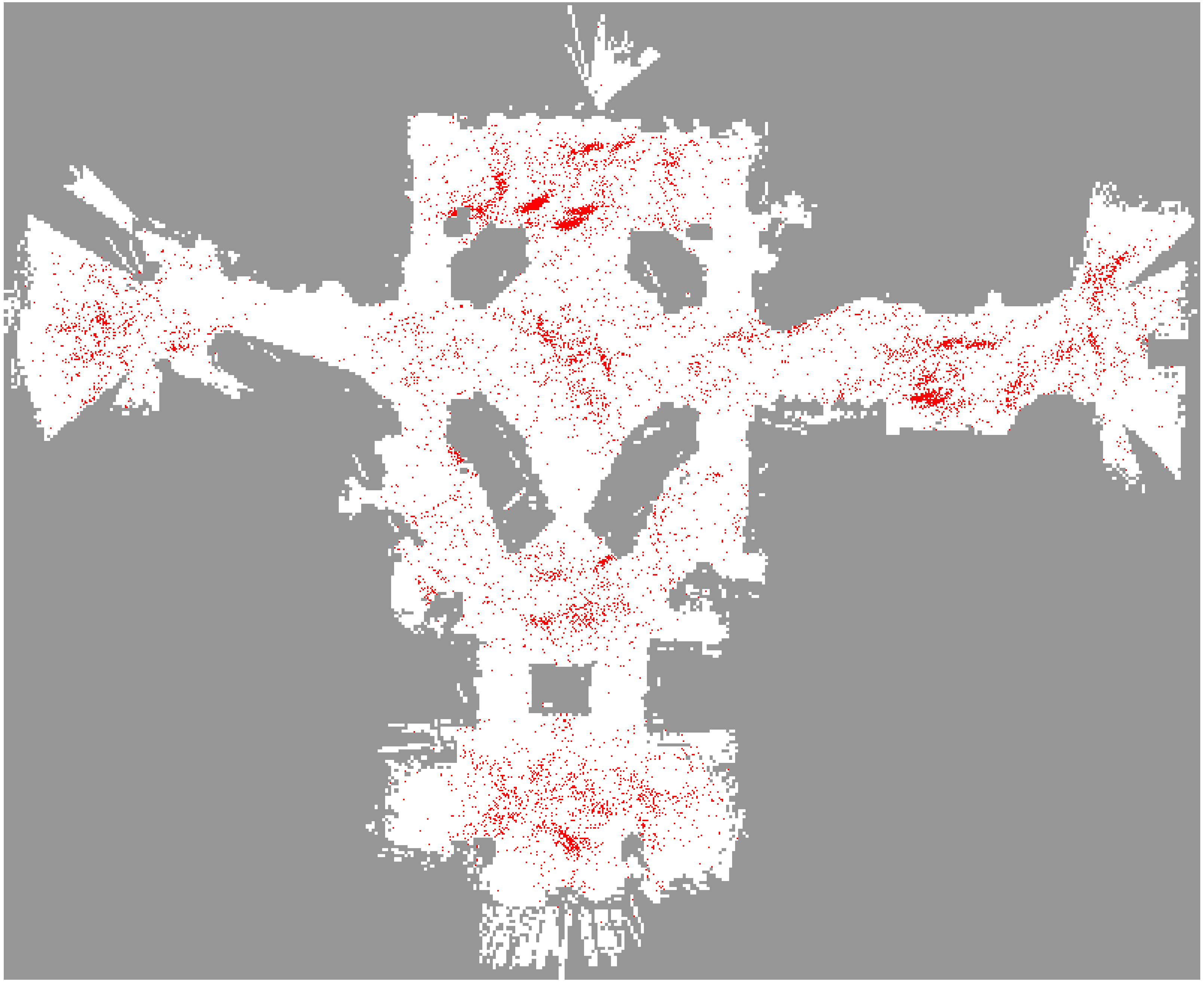
Laser sensor

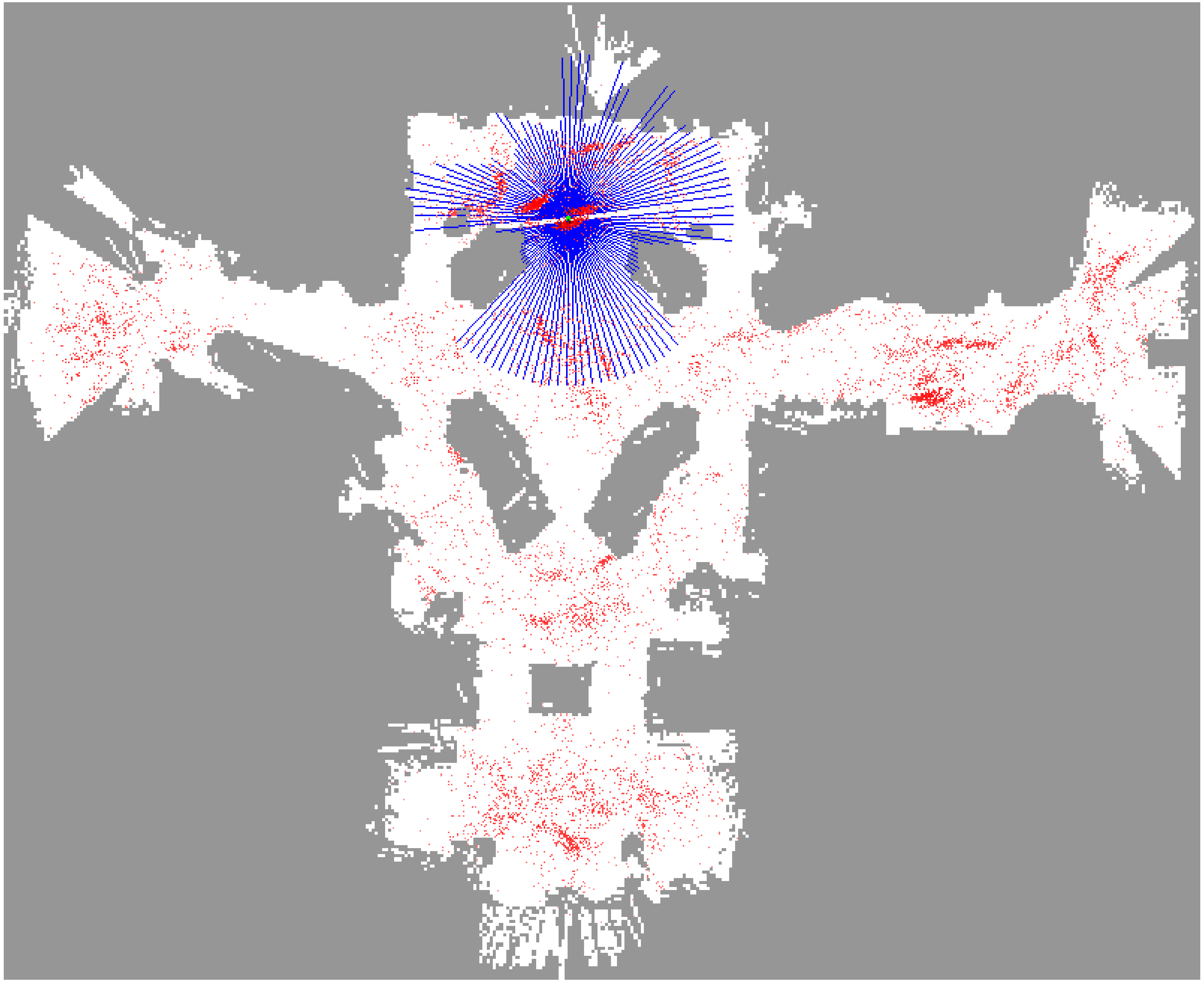


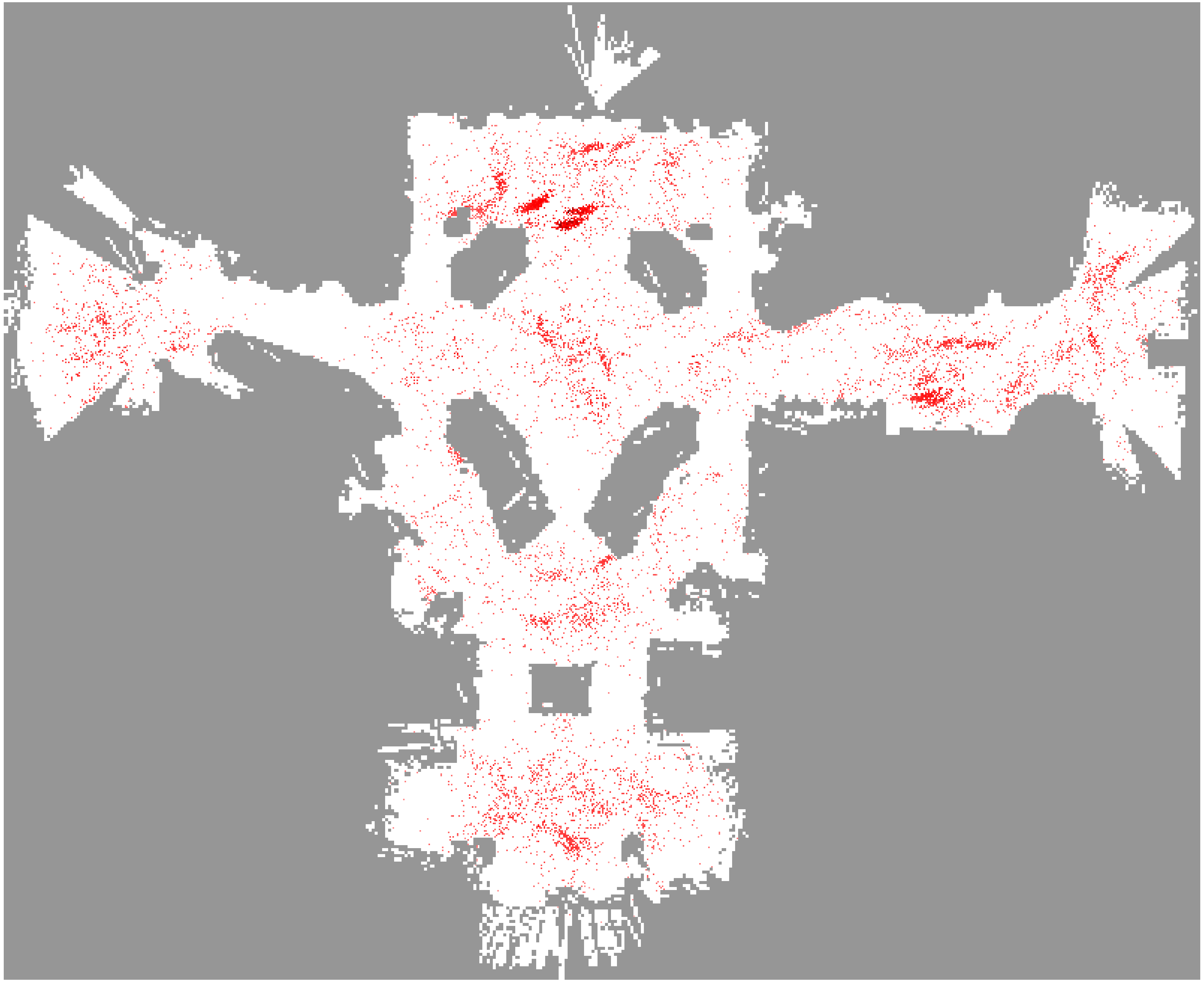
Sonar sensor

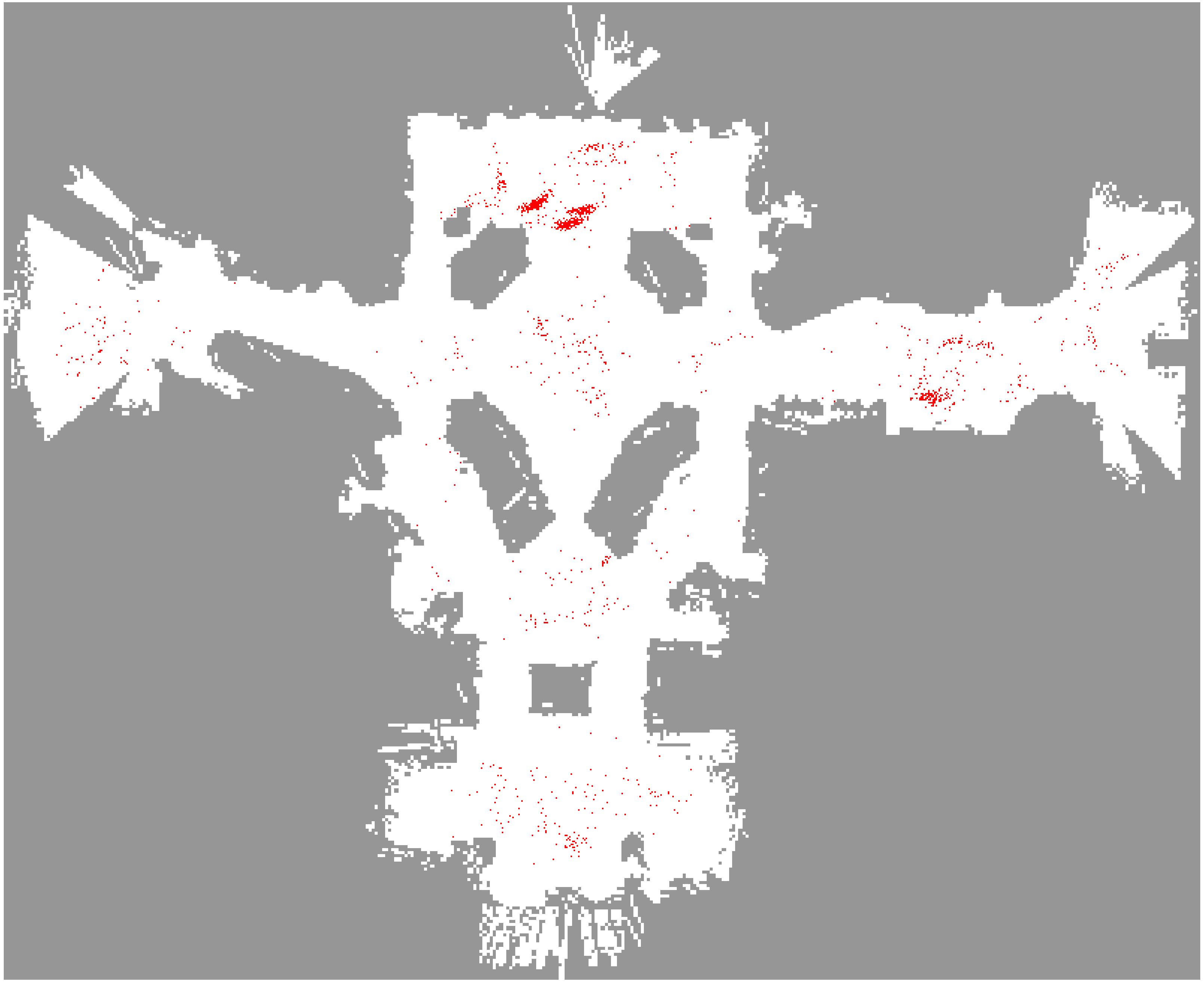


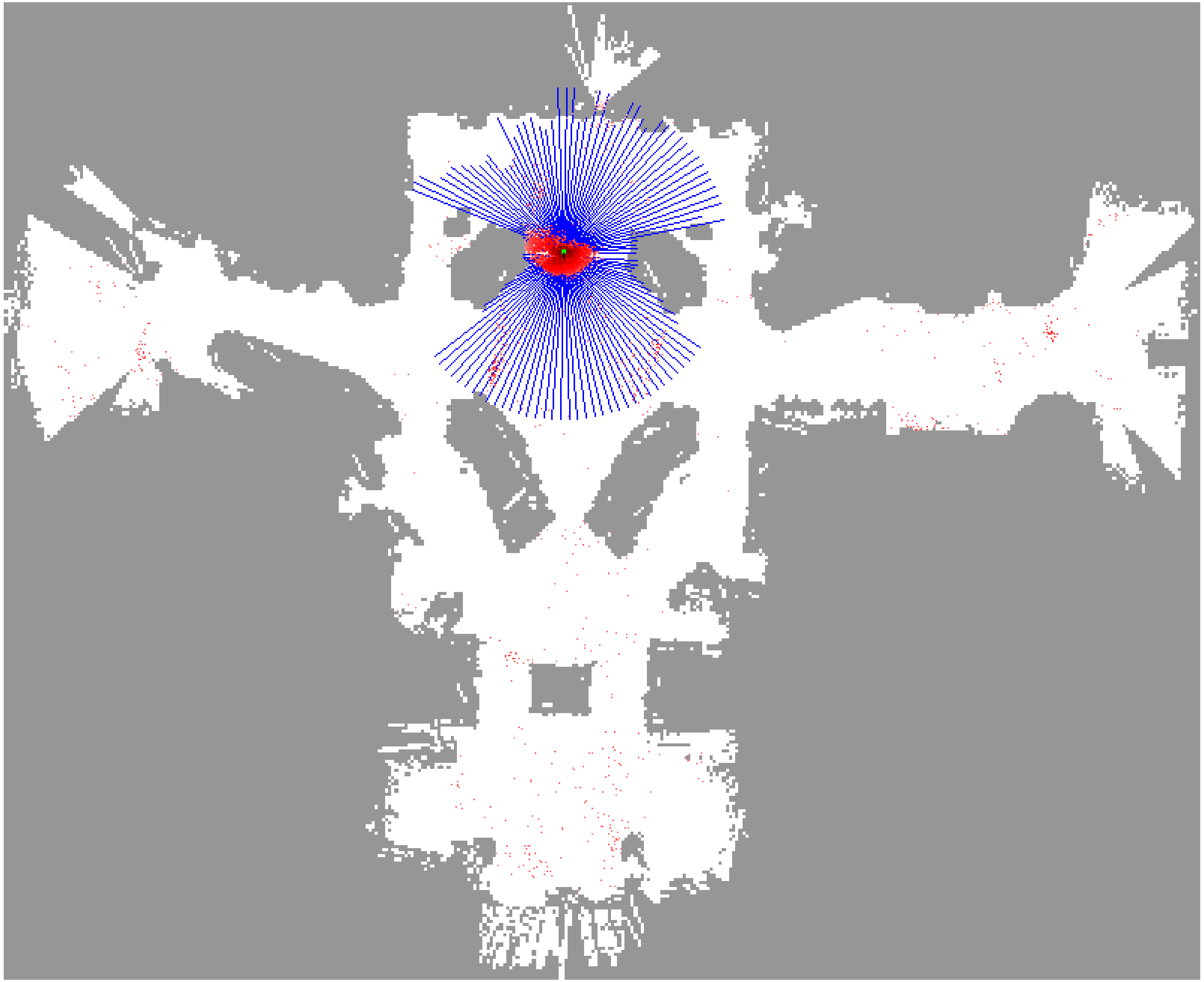


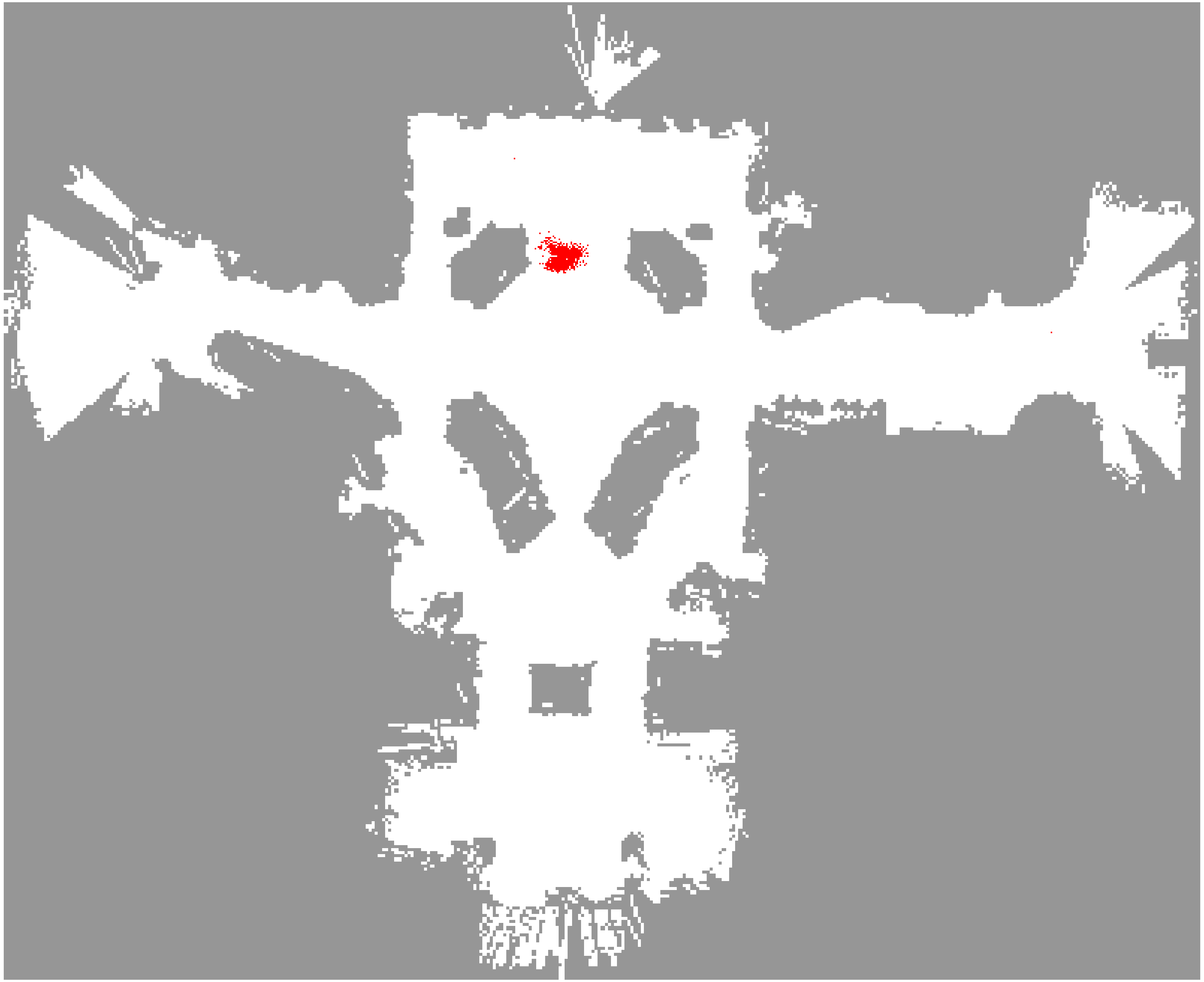


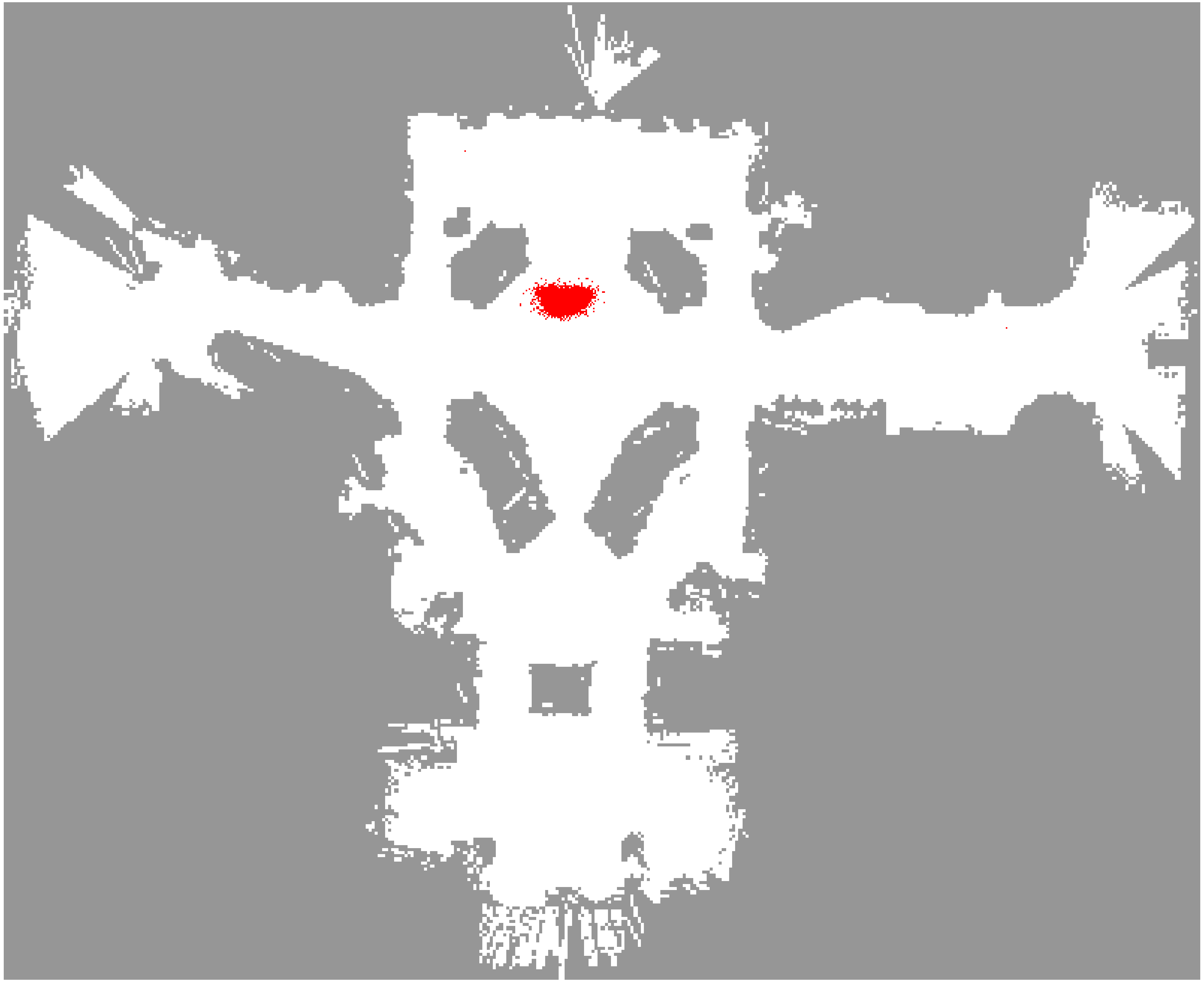


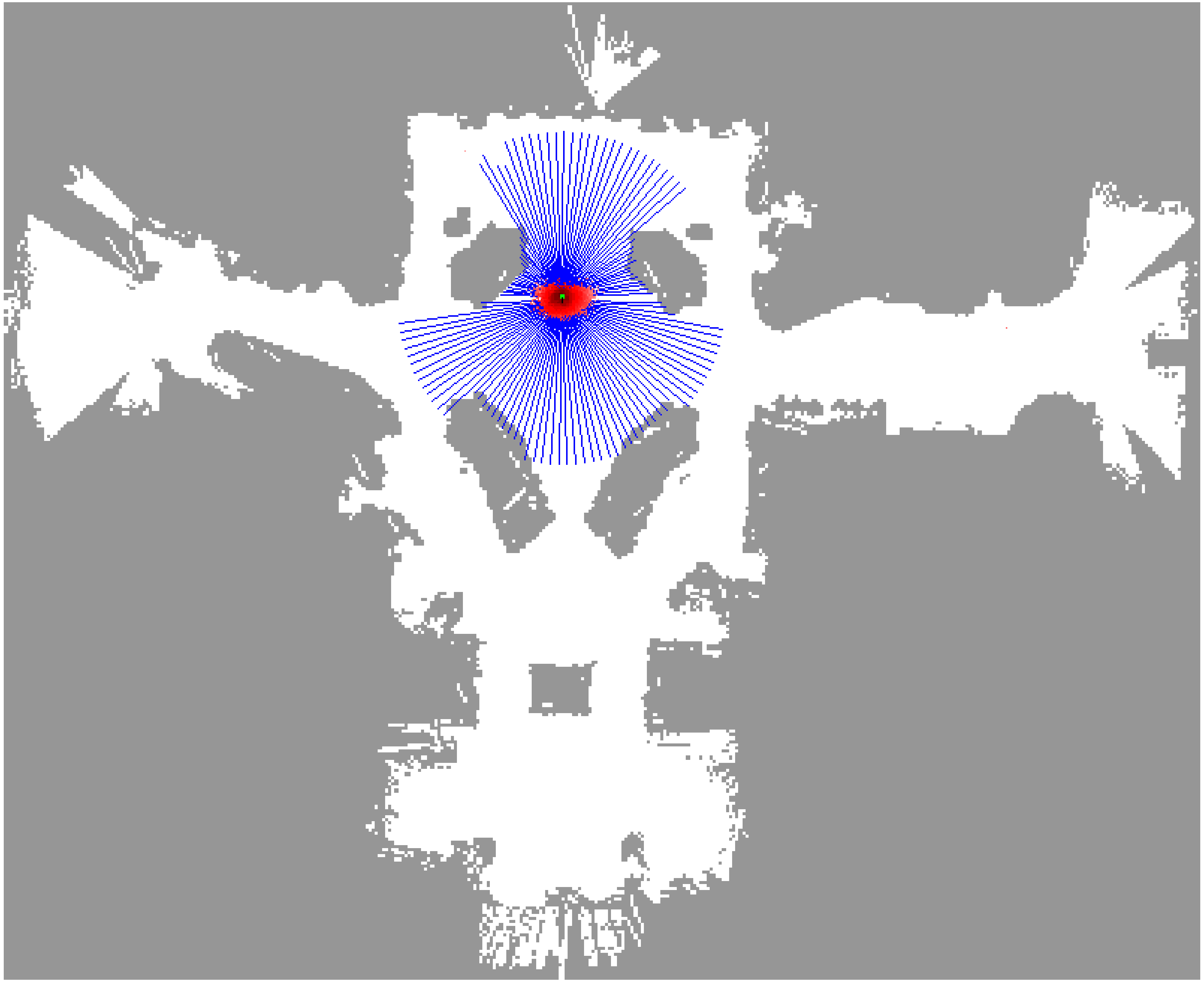


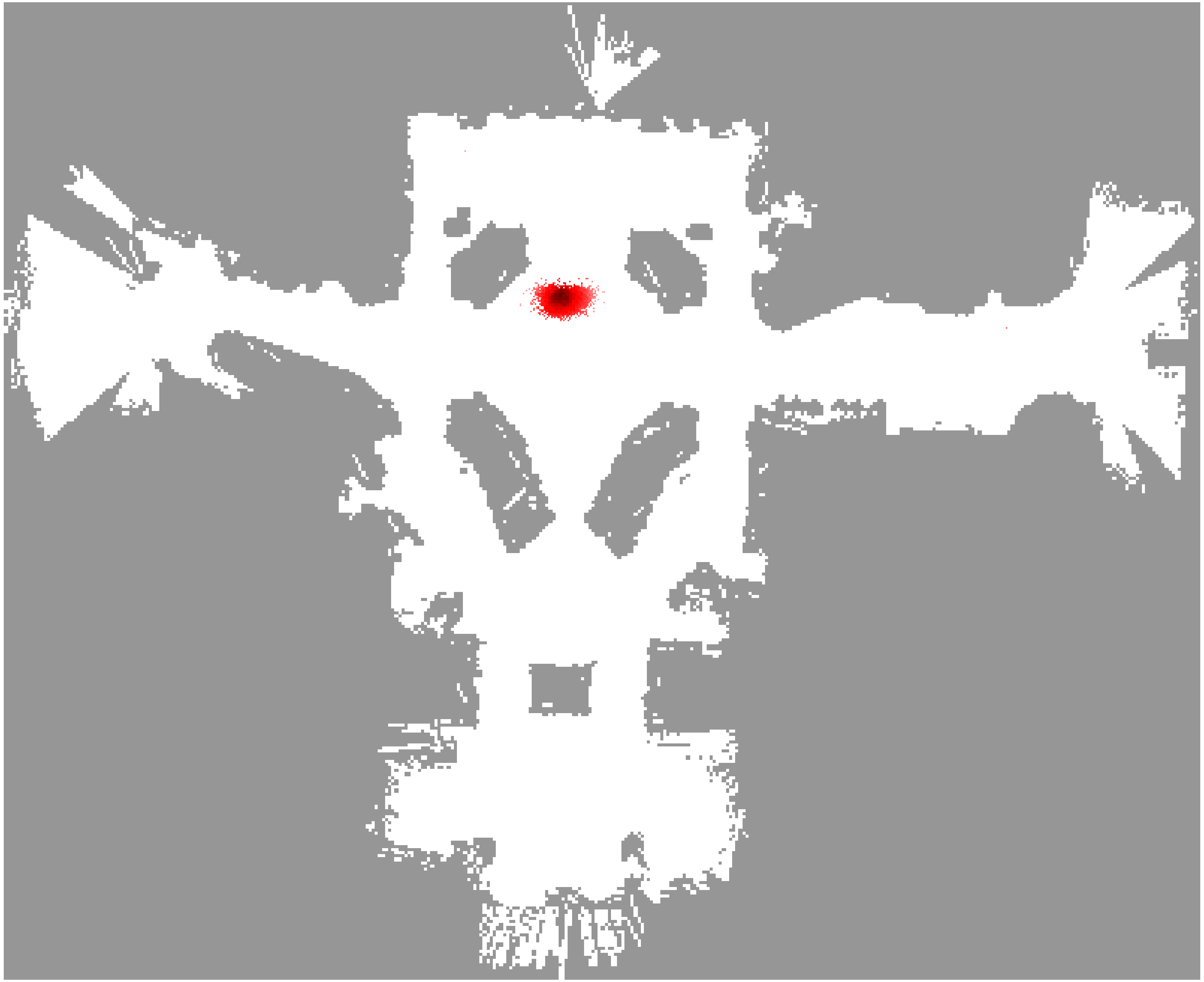


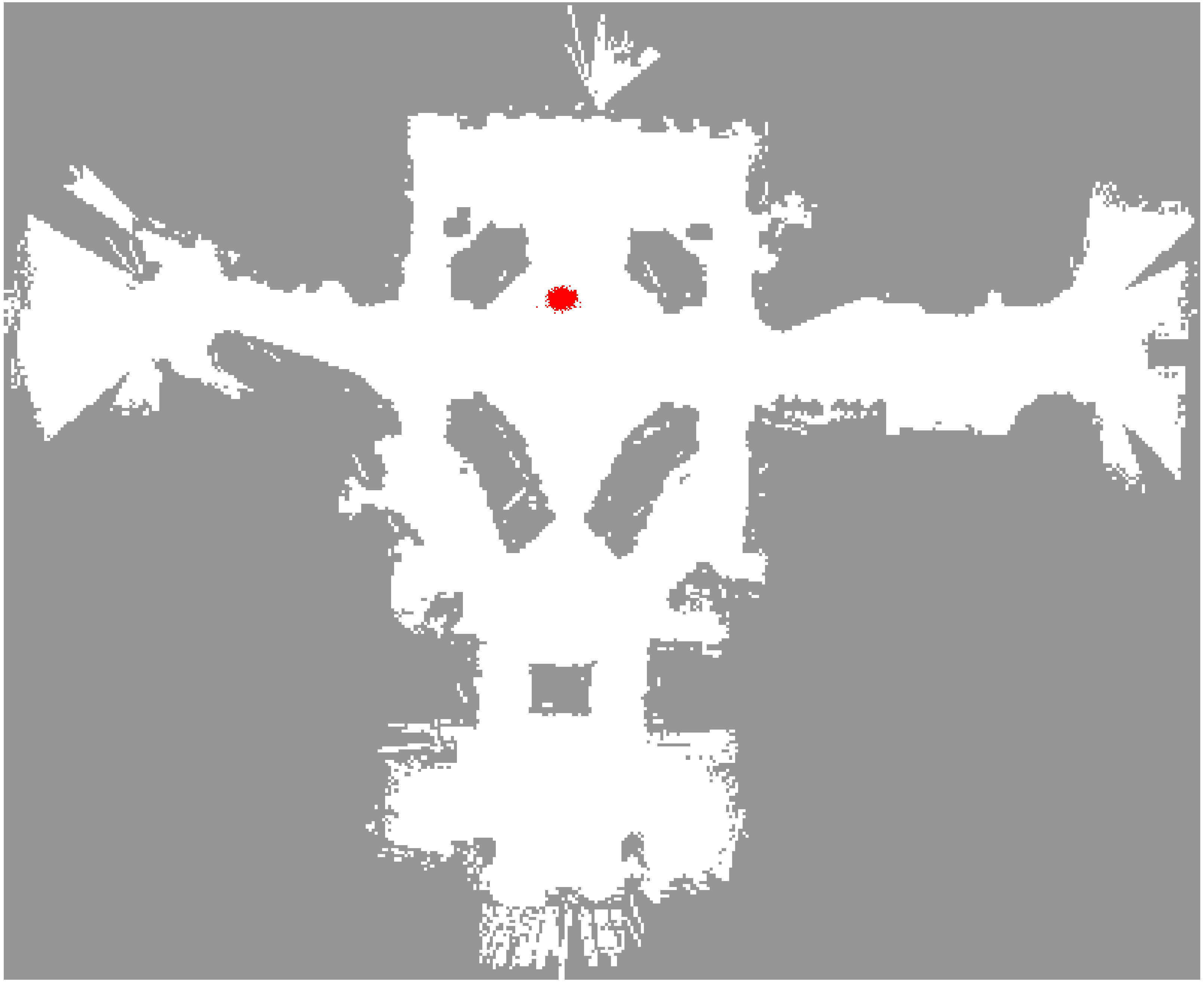


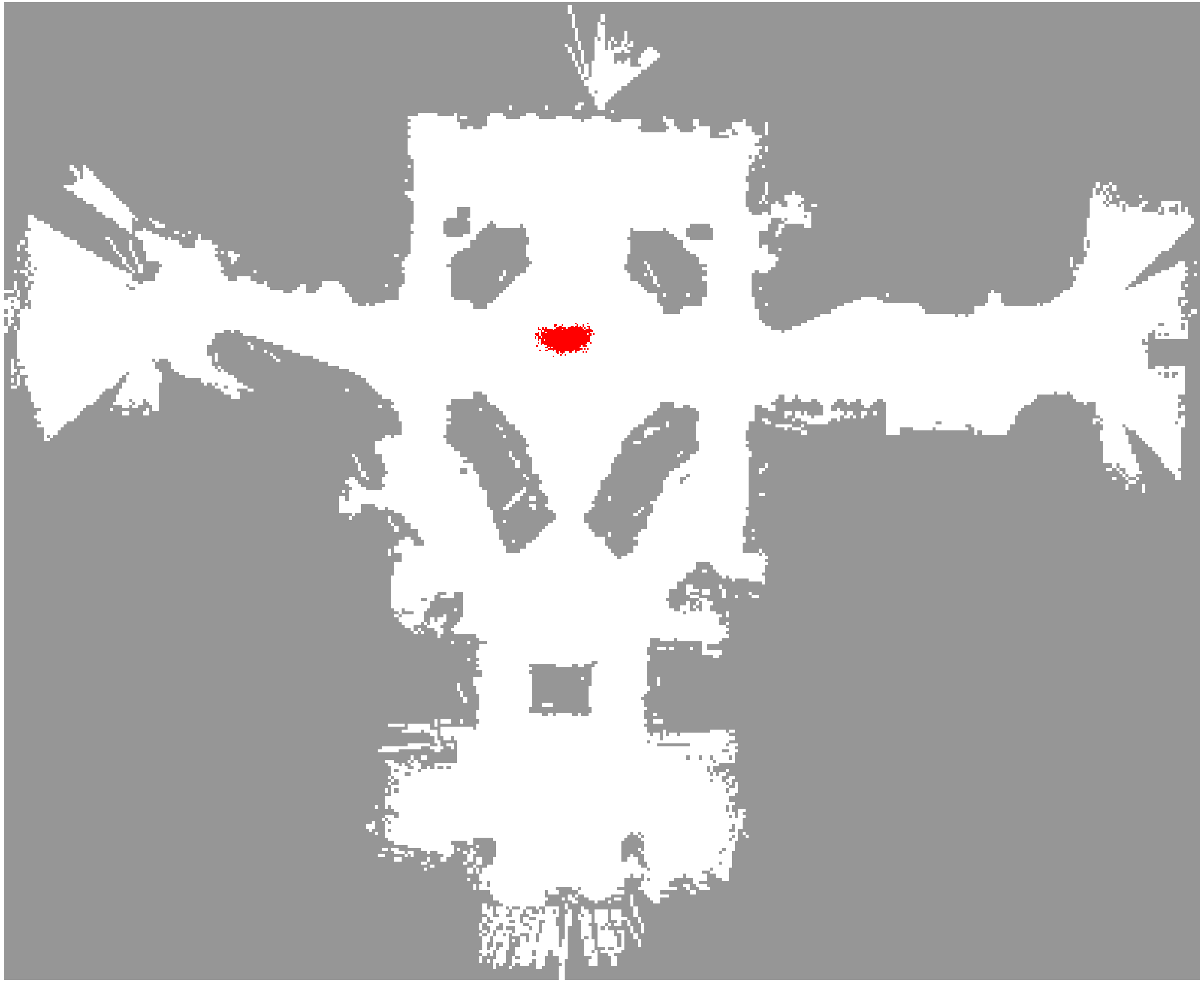


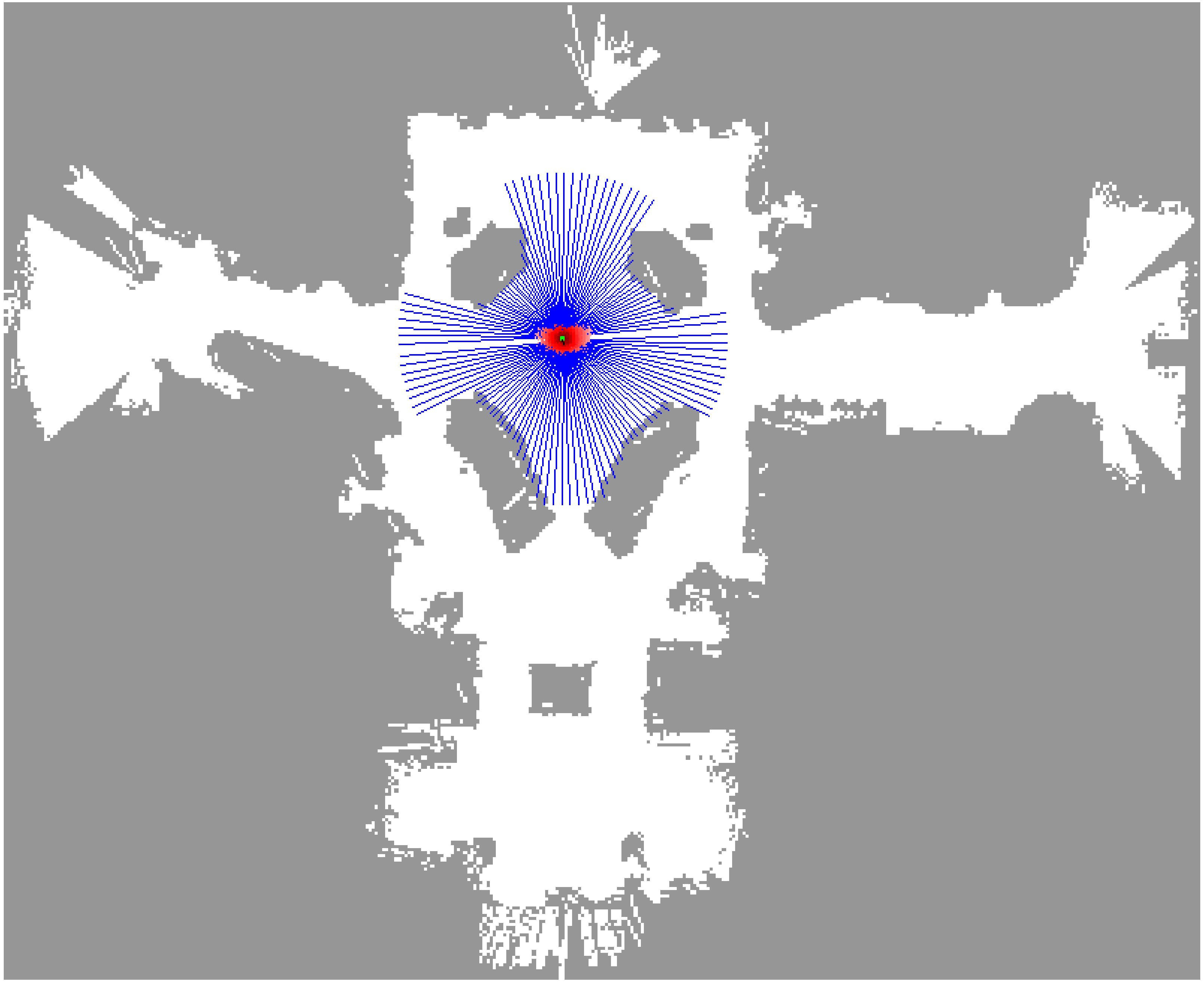










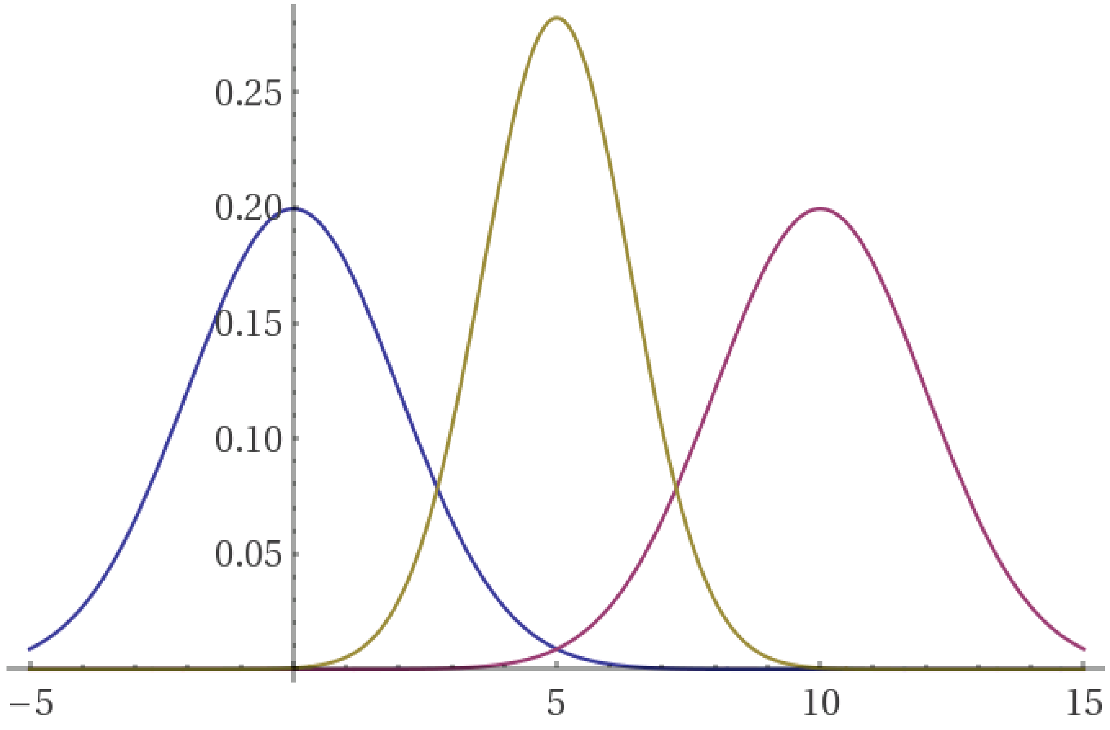


Particle Filter Summary

- Very flexible tool as we get to make our choice of proposal distributions (as long as we can properly compute importance weight)
- Performance is guaranteed *given infinite samples!*
- The particle cloud and its weights represent our distribution, but making decisions can still be complex:
  - Act based on the most likely particle
  - Act using a weighted summation over particles
  - Act conservatively, accounting for the worst particle
- In practice, the number of particles required to perform well scales with the problem complexity and this can be hard to measure

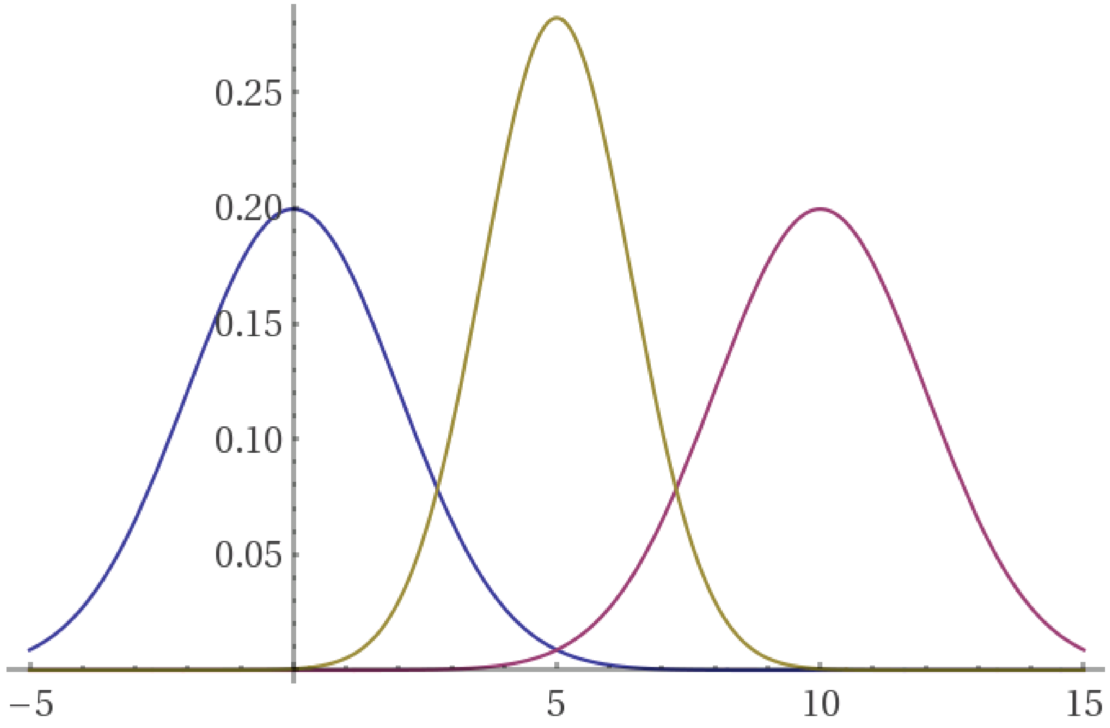
Part 2: Kalman Filters

- Intuition: track Bel(x) with a Gaussian distribution, simplifying assumptions to ensure updates are all possible
- Payoffs:
  - Continuous representation
  - Efficient computation
- Uses:
  - Rocketry
  - Mobile devices
  - Drones
  - GPS
  - (the list is very long…)

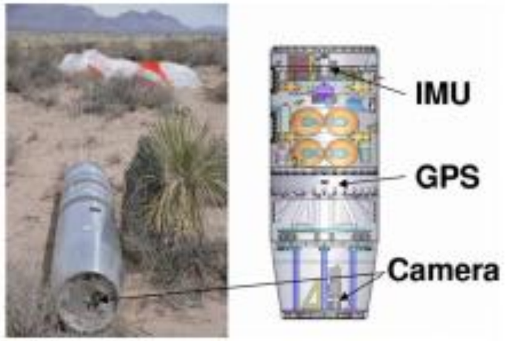


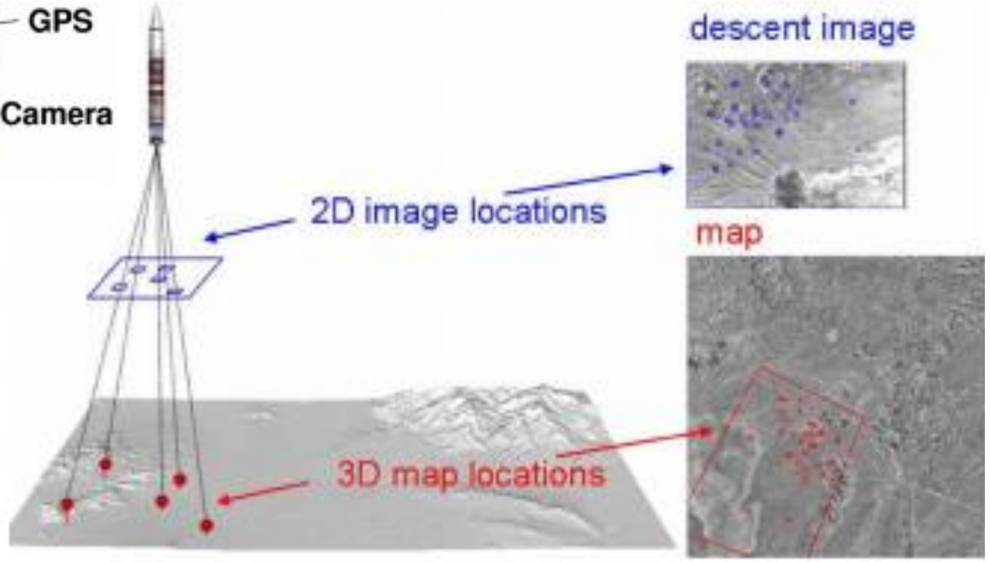
Part 2: Kalman Filters

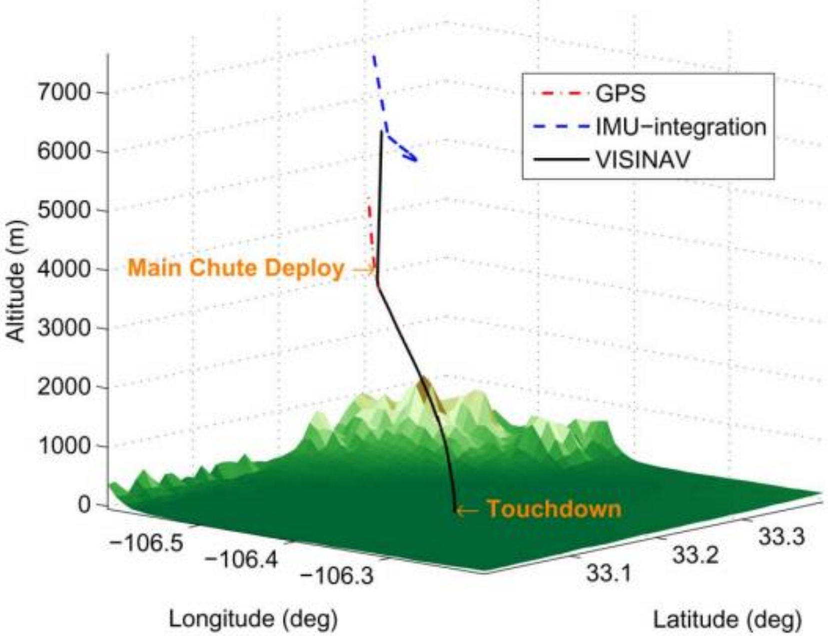
- Intuition: track Bel(x) with a Gaussian distribution, simplifying assumptions to ensure updates are all possible
- Payoffs:
  - Continuous representation
  - Efficient computation
- Uses:
  - Rocketry
  - Mobile devices
  - Drones
  - GPS
  - (the list is very long…)



Example: Landing on mars







264 IEEE TRANSACTIONS ON ROBOTICS, VOL. 25, NO. 2, APRIL 2009

Vision-Aided Inertial Navigation for Spacecraft Entry, Descent, and Landing

Anastasios I. Mourikis, *Member, IEEE*, Nikolas Trawny, *Student Member, IEEE*,
Stergios I. Roumeliotis, *Member, IEEE*, Andrew E. Johnson, Adnan Ansar, and Larry Matthies, *Senior Member, IEEE*

Kalman Filter: an instance of Bayes' Filter

Kalman Filter: Approach

$$
\begin{aligned}
bel(x_t) &= p(x_t | u_{0:t-1}, z_{0:t}) \\
&= \eta \, p(z_t | x_t) \int p(x_t | u_{t-1}, x_{t-1}) \, bel(x_{t-1}) \, dx_{t-1}
\end{aligned}
$$

Linear dynamics with Gaussian noise

$$
\begin{aligned}
x_t &= Ax_{t-1} + Bu_{t-1} + Gw_{t-1} \\
& \quad \text{with noise } w_{t-1} \sim \mathcal{N}(0, Q)
\end{aligned}
$$

Linear observations with Gaussian noise

$$
\begin{aligned}
z_t &= Hx_t + n_t \\
& \quad \text{with noise } n_t \sim \mathcal{N}(0, R)
\end{aligned}
$$

+ Initial belief is Gaussian

$$bel(x_0) \sim \mathcal{N}(\mu_0, \Sigma_0)$$

Kalman Filter: assumptions

- Two assumptions inherited from Bayes' Filter
- Linear dynamics and observation models
- Initial belief is Gaussian
- Noise variables and initial state

  $x_0, w_0, w_1, ..., n_0, n_1, ...$

  are jointly Gaussian and independent
- Noise variables w_t are independent and identically distributed $\mathcal{N}(0, Q)$
- Noise variables n_t are independent and identically distributed $\mathcal{N}(0, R)$

Kalman Filter: why so many assumptions?

- Two assumptions inherited from Bayes' Filter
- Linear dynamics and observation models
- Initial belief is Gaussian
- Noise variables and initial state

  $x_0, w_0, w_1, \ldots, n_0, n_1, \ldots$

  are jointly Gaussian and independent
- Noise variables w_t are independent and identically distributed $\mathcal{N}(0, Q)$
- Noise variables n_t are independent and identically distributed $\mathcal{N}(0, R)$

Without linearity there is no closed-form solution for the posterior belief in the Bayes' Filter. Recall that if X is Gaussian then Y=AX+b is also Gaussian. This is not true in general if Y=h(X).

Also, we will see later that applying Bayes' rule to a Gaussian prior and a Gaussian measurement likelihood results in a Gaussian posterior.

Kalman Filter: why so many assumptions?

- Two assumptions inherited from Bayes' Filter
- Linear dynamics and observation models
- Initial belief is Gaussian
- Noise variables and initial state

  $x_0, w_0, w_1, \ldots, n_0, n_1, \ldots$

  are jointly Gaussian and independent
- Noise variables w_t are independent and identically distributed $\mathcal{N}(0, Q)$
- Noise variables n_t are independent and identically distributed $\mathcal{N}(0, R)$

This results in the belief remaining Gaussian after each propagation and update step. This means that we only have to worry about how the mean and the covariance of the belief evolve recursively with each prediction step and update step → COOL!

Kalman Filter: why so many assumptions?

- Two assumptions inherited from Bayes' Filter
- Linear dynamics and observation models
- Initial belief is Gaussian
- Noise variables and initial state

  $x_0, w_0, w_1, ..., n_0, n_1, ...$

  are jointly Gaussian and independent
- Noise variables w_t are independent and identically distributed $\mathcal{N}(0, Q)$
- Noise variables n_t are independent and identically distributed $\mathcal{N}(0, R)$

This makes the recursive updates of the mean and covariance much simpler.

Kalman Filter: an instance of Bayes' Filter

Assumptions guarantee that if the prior belief before the prediction step is Gaussian

$$
\begin{aligned}
bel(x_t) &= p(x_t | u_{0:t-1}, z_{0:t}) \\
&= \eta \, p(z_t | x_t) \int p(x_t | u_{t-1}, x_{t-1}) \, bel(x_{t-1}) \, dx_{t-1}
\end{aligned}
$$

then the prior belief after the prediction step will be Gaussian

and the posterior belief (after the update step) will be Gaussian.

Kalman Filter: an instance of Bayes' Filter

$$
\begin{aligned}
bel(x_t) &= p(x_t | u_{0:t-1}, z_{0:t}) \\
&= \eta\, p(z_t|x_t)\, p(x_t|u_{0:t-1}, z_{0:t-1}) \\
&= \eta\, p(z_t|x_t) \int p(x_t|u_{t-1}, x_{t-1})\, bel(x_{t-1})\, dx_{t-1} \\
&= \eta\, p(z_t|x_t)\, \overline{bel}(x_t)
\end{aligned}
$$

Belief after prediction step (to simplify notation)

So, under the Kalman Filter assumptions we get

$$bel(x_{t-1}) \sim \mathcal{N}(\mu_{t-1|t-1}, \Sigma_{t-1|t-1})$$

$$\overline{bel}(x_t) \sim \mathcal{N}(\mu_{t|t-1}, \Sigma_{t|t-1})$$

$$bel(x_t) \sim \mathcal{N}(\mu_{t|t}, \Sigma_{t|t})$$

Notation: estimate at time t given history of observations and controls up to time t-1

Kalman Filter: an instance of Bayes' Filter

$$\begin{aligned} bel(x_t) &= p(x_t|u_{0:t-1}, z_{0:t}) \\ &= \eta\, p(z_t|x_t) \int p(x_t|u_{t-1}, x_{t-1})\, bel(x_{t-1})\, dx_{t-1} \\ &= \eta\, p(z_t|x_t)\, \overline{bel}(x_t) \end{aligned}$$

So, under the Kalman Filter assumptions we get

$$bel(x_{t-1}) \sim \mathcal{N}(\mu_{t-1|t-1}, \Sigma_{t-1|t-1})$$

$$\downarrow$$

$$\overline{bel}(x_t) \sim \mathcal{N}(\mu_{t|t-1}, \Sigma_{t|t-1})$$

$$\downarrow$$

$$bel(x_t) \sim \mathcal{N}(\mu_{t|t}, \Sigma_{t|t})$$

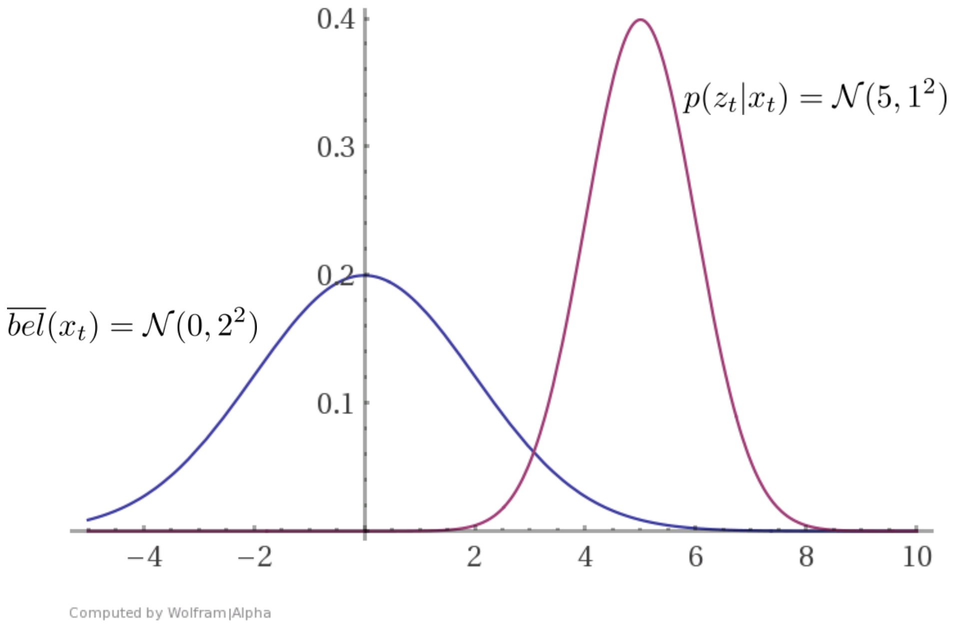
Two main questions:

1. How to get prediction mean and covariance from prior mean and covariance?
2. How to get posterior mean and covariance from prediction mean and covariance?

These questions were answered in the 1960s. The resulting algorithm was used in the Apollo missions to the moon, and in almost every system in which there is a noisy sensor involved → COOL!

Kalman Filter with 1D state

- Let's start with the update step recursion. Here's an example:



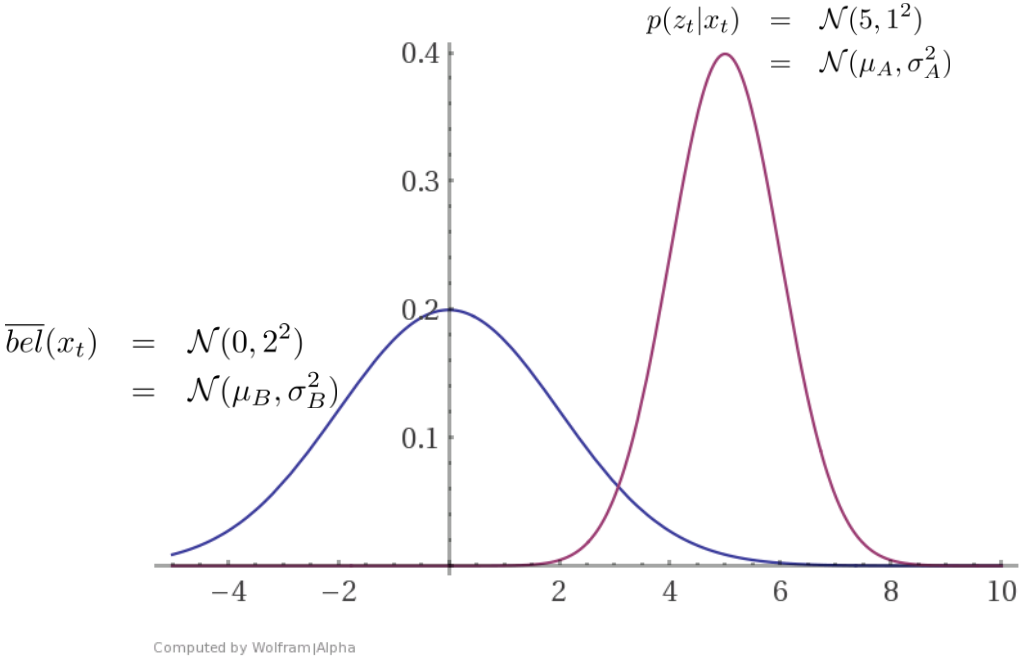
Suppose your measurement model is $z_t = x_t + n_t$ with $n_t \sim \mathcal{N}(0, 1^2)$

Suppose your belief after the prediction step is $\overline{bel}(x_t) = \mathcal{N}(0, 2^2)$

Suppose your first noisy measurement is $z_0 = 5$

Q: What is the mean and covariance of $bel(x_t)$?

Kalman Filter with 1D state: the update step



From Bayes' Filter we get $bel(x_t) = \eta \; p(z_t|x_t) \; \overline{bel}(x_t)$ so

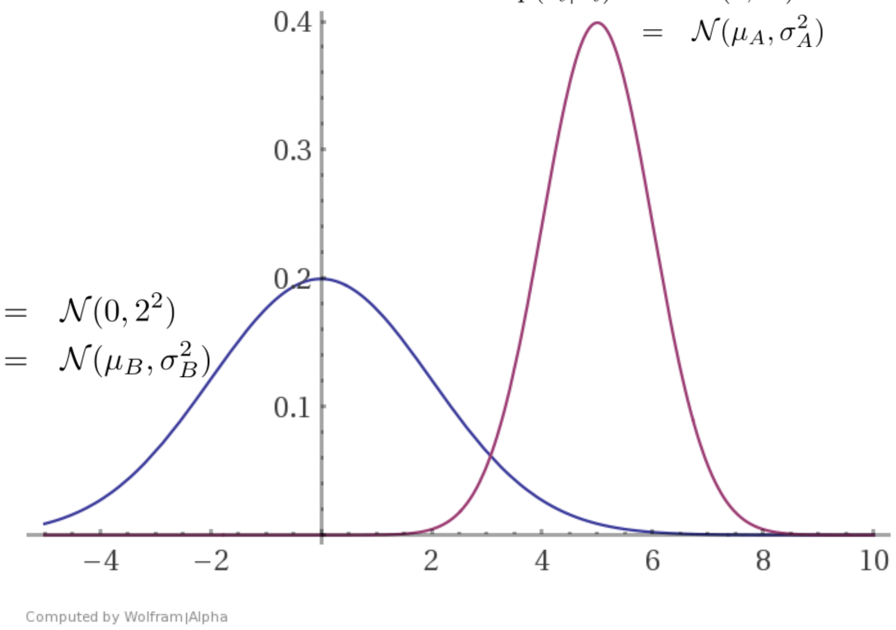
$$
\begin{aligned}
p(z_t|x_t) \; \overline{bel}(x_t) &= \mathcal{N}(\mu_A, \sigma_A^2)\mathcal{N}(\mu_B, \sigma_B^2) \\
&= \ldots \\
&= \text{see Appendix 1 for proof} \\
&= \ldots \\
&= \mathcal{N}(\mu, \sigma^2)/\eta
\end{aligned}
$$

$$\mu = \mu_B + \frac{\sigma_B^2}{\sigma_A^2 + \sigma_B^2}(\mu_A - \mu_B)$$

$$\sigma^2 = \sigma_B^2 - \frac{\sigma_B^2}{\sigma_A^2 + \sigma_B^2}\sigma_B^2$$

Kalman Filter with 1D state: the update step

From Bayes' Filter we get $bel(x_t) = \eta\, p(z_t|x_t)\, \overline{bel}(x_t)$ so



$$\begin{aligned}
p(z_t|x_t)\, \overline{bel}(x_t) &= \mathcal{N}(\mu_A, \sigma_A^2)\mathcal{N}(\mu_B, \sigma_B^2) \\
&= \ldots \\
&= \text{see Appendix 1 for proof} \\
&= \ldots \\
&= \mathcal{N}(\mu, \sigma^2)/\eta
\end{aligned}$$

$$\mu = \mu_B + \frac{\sigma_B^2}{\sigma_A^2 + \sigma_B^2}\boxed{(\mu_A - \mu_B)}$$

$$\sigma^2 = \sigma_B^2 - \frac{\sigma_B^2}{\sigma_A^2 + \sigma_B^2}\sigma_B^2$$

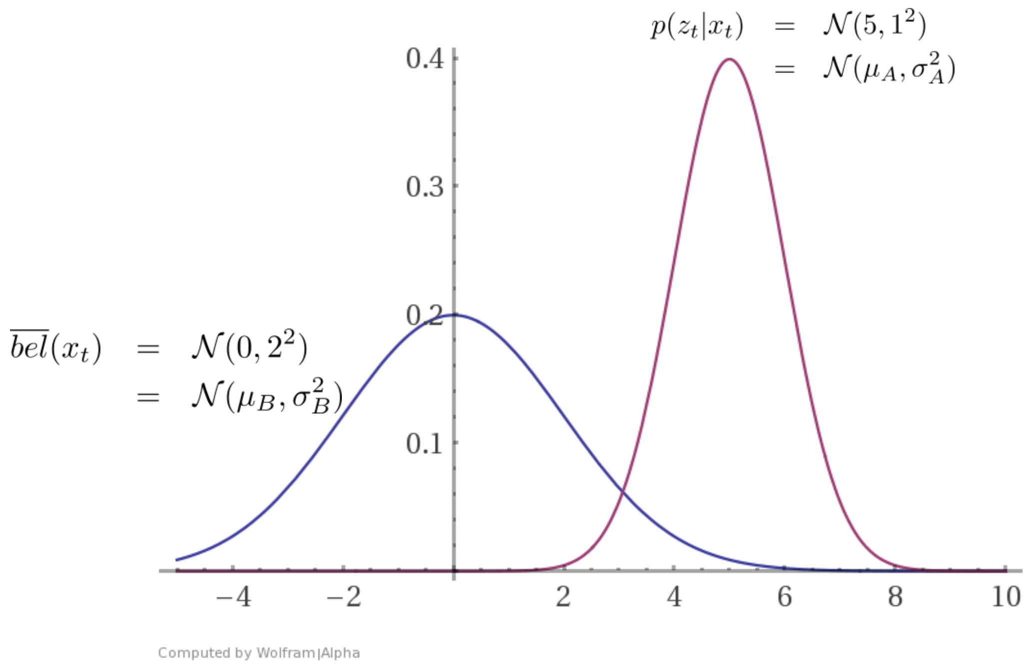
Prediction residual/error between actual observation and expected observation.

You expected the measured mean to be 0, according to your prediction prior, but you actually observed 5.

The smaller this prediction error is the better your estimate will be, or the better it will agree with the measurements.

Kalman Filter with 1D state: the update step

From Bayes' Filter we get $bel(x_t) = \eta \, p(z_t|x_t) \, \overline{bel}(x_t)$ so



$$
\begin{aligned}
p(z_t|x_t) \, \overline{bel}(x_t) &= \mathcal{N}(\mu_A, \sigma_A^2)\mathcal{N}(\mu_B, \sigma_B^2) \\
&= \ldots \\
&= \text{see Appendix 1 for proof} \\
&= \ldots \\
&= \mathcal{N}(\mu, \sigma^2)/\eta
\end{aligned}
$$

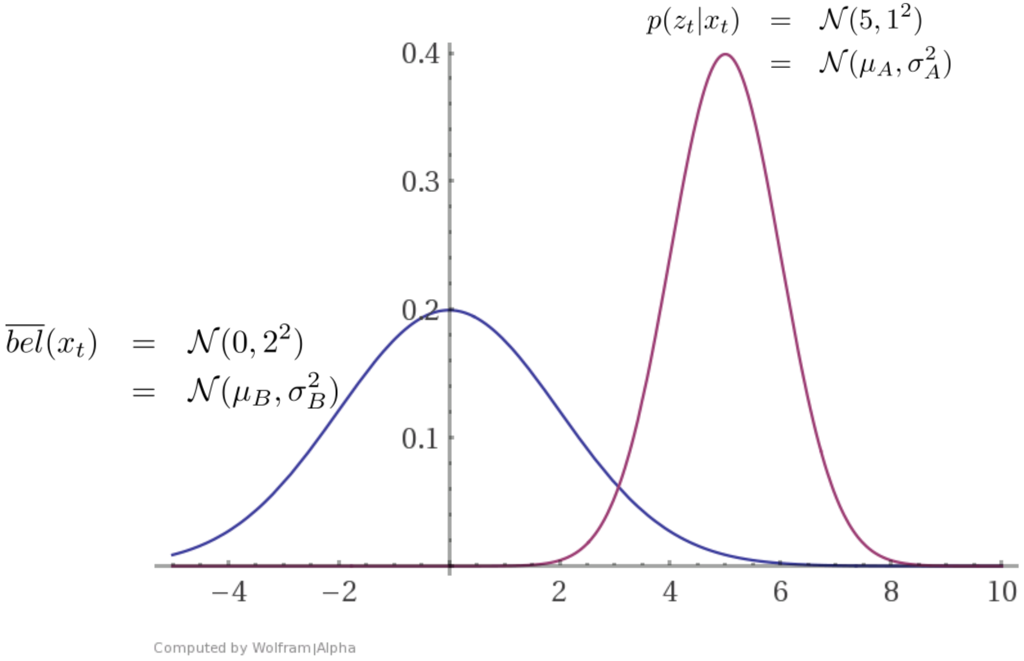
$$\mu = \mu_B + \boxed{\frac{\sigma_B^2}{\sigma_A^2 + \sigma_B^2}}(\mu_A - \mu_B)$$

$$\sigma^2 = \sigma_B^2 - \boxed{\frac{\sigma_B^2}{\sigma_A^2 + \sigma_B^2}}\sigma_B^2$$

Kalman Gain: specifies how much effect will the measurement have in the posterior, compared to the prediction prior. Which one do you trust more, your prior $\overline{bel}(x_t)$, or your measurement $p(z_t|x_t)$?

Kalman Filter with 1D state: the update step

From Bayes' Filter we get $bel(x_t) = \eta\, p(z_t|x_t)\, \overline{bel}(x_t)$ so



$$
\begin{aligned}
p(z_t|x_t)\, \overline{bel}(x_t) &= \mathcal{N}(\mu_A, \sigma_A^2)\mathcal{N}(\mu_B, \sigma_B^2) \\
&= \ldots \\
&= \text{see Appendix 1 for proof} \\
&= \ldots \\
&= \mathcal{N}(\mu, \sigma^2)/\eta
\end{aligned}
$$

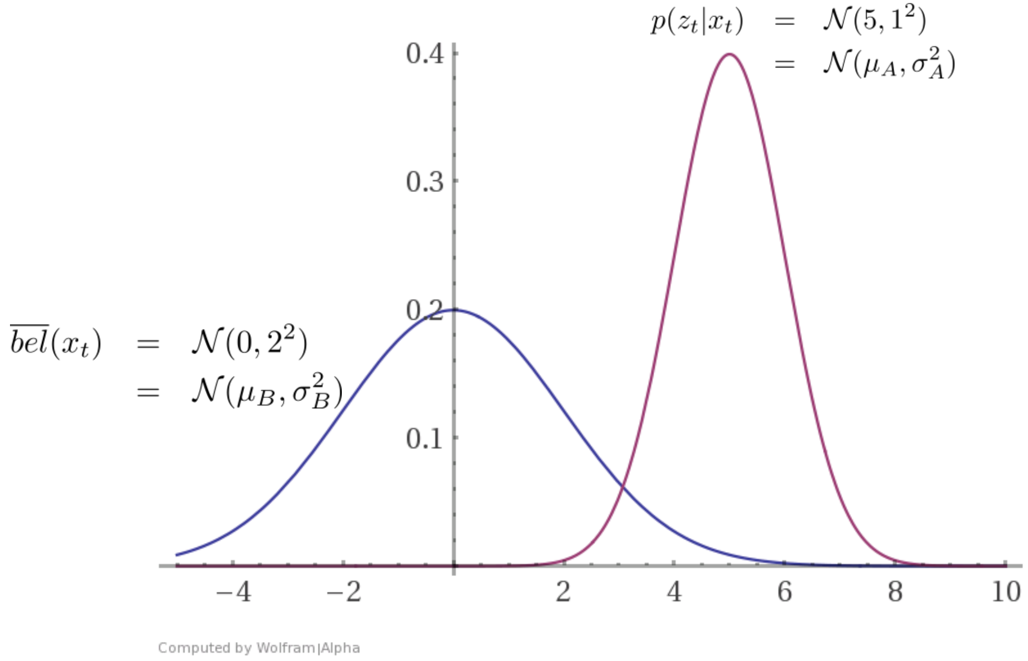
$$\mu = \mu_B + \boxed{\frac{\sigma_B^2}{\sigma_A^2 + \sigma_B^2}}(\mu_A - \mu_B)$$

The measurement is more confident (lower variance) than the prior, so the posterior mean is going to be closer to 5 than to 0.

$$\sigma^2 = \sigma_B^2 - \frac{\sigma_B^2}{\sigma_A^2 + \sigma_B^2}\sigma_B^2$$

Kalman Filter with 1D state: the update step

From Bayes' Filter we get $bel(x_t) = \eta\, p(z_t|x_t)\, \overline{bel}(x_t)$ so



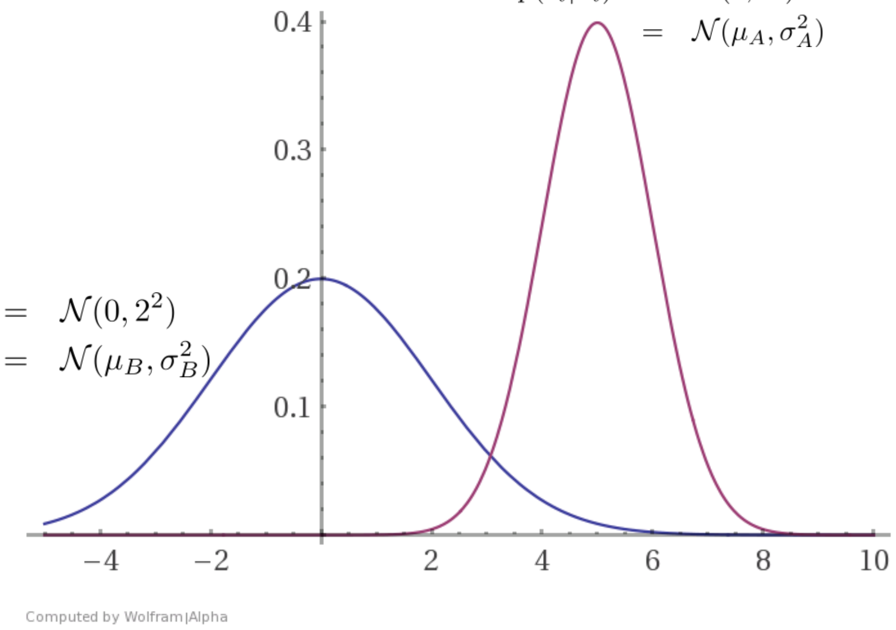
$$
\begin{aligned}
p(z_t|x_t)\, \overline{bel}(x_t) &= \mathcal{N}(\mu_A, \sigma_A^2)\mathcal{N}(\mu_B, \sigma_B^2) \\
&= \ldots \\
&= \text{see Appendix 1 for proof} \\
&= \ldots \\
&= \mathcal{N}(\mu, \sigma^2)/\eta
\end{aligned}
$$

$$\mu = \mu_B + \frac{\sigma_B^2}{\sigma_A^2 + \sigma_B^2}(\mu_A - \mu_B)$$

$$\sigma^2 = \sigma_B^2 \boxed{-} \frac{\sigma_B^2}{\sigma_A^2 + \sigma_B^2}\sigma_B^2$$

No matter what happens, the variance of the posterior is going to be reduced. I.e. new measurement increases confidence no matter how noisy it is.

Kalman Filter with 1D state: the update step



From Bayes' Filter we get $bel(x_t) = \eta \, p(z_t|x_t) \, \overline{bel}(x_t)$ so

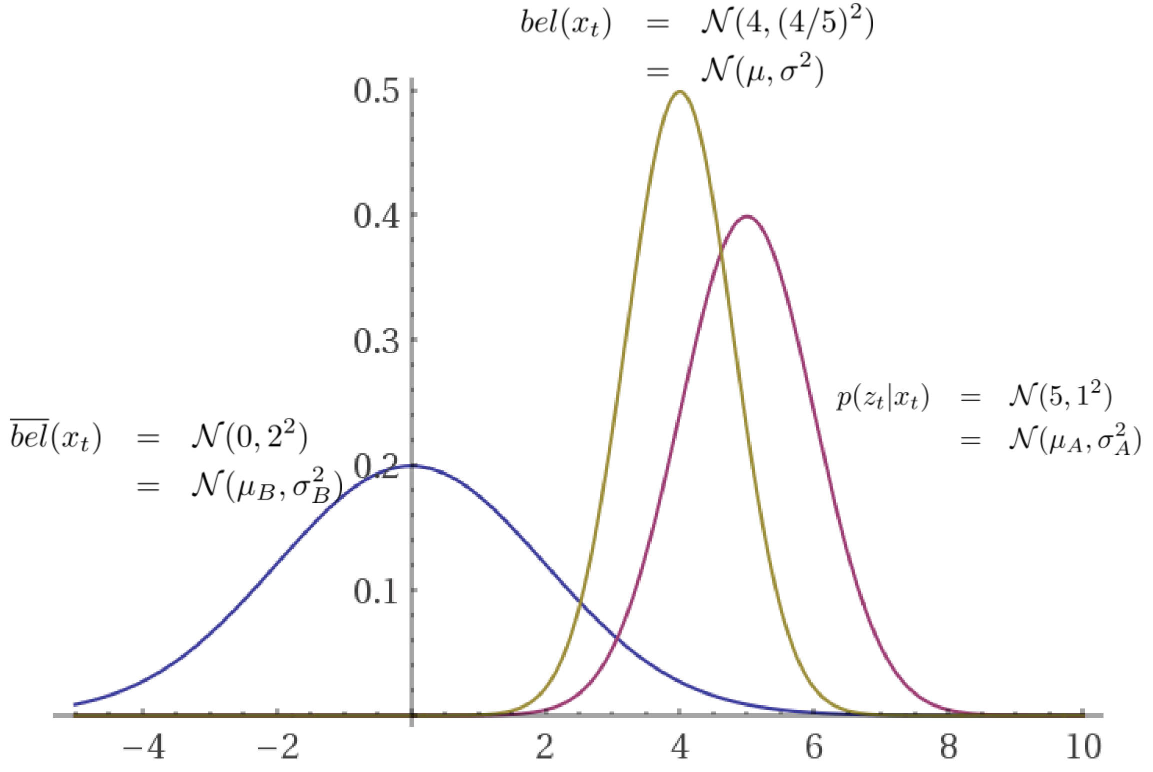
$$
\begin{aligned}
p(z_t|x_t) \, \overline{bel}(x_t) &= \mathcal{N}(\mu_A, \sigma_A^2)\mathcal{N}(\mu_B, \sigma_B^2) \\
&= \ldots \\
&= \text{see Appendix 1 for proof} \\
&= \ldots \\
&= \mathcal{N}(\mu, \sigma^2)/\eta
\end{aligned}
$$

$$\mu = \mu_B + \frac{\sigma_B^2}{\sigma_A^2 + \sigma_B^2}(\mu_A - \mu_B)$$

$$\boxed{\sigma^2 = \sigma_B^2 - \frac{\sigma_B^2}{\sigma_A^2 + \sigma_B^2}\sigma_B^2}$$

In fact you can write this as $\frac{1}{\sigma^2} = \frac{1}{\sigma_A^2} + \frac{1}{\sigma_B^2}$
so $\sigma < \sigma_A$ and $\sigma < \sigma_B$
I.e. the posterior is more confident than both the prior and the measurement.

Kalman Filter with 1D state: the update step



From Bayes' Filter we get $bel(x_t) = \eta\, p(z_t|x_t)\, \overline{bel}(x_t)$ so

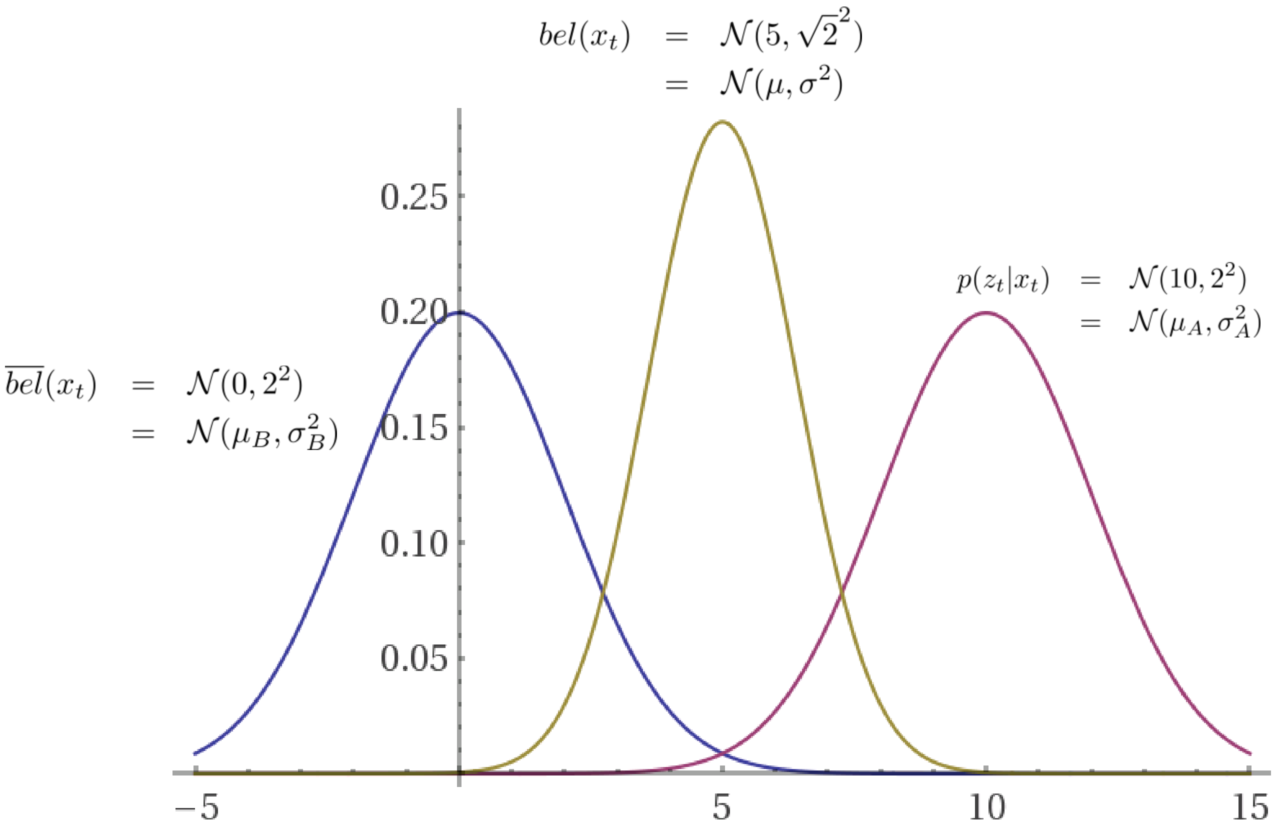
$$\begin{aligned} p(z_t|x_t)\,\overline{bel}(x_t) &= \mathcal{N}(\mu_A,\sigma_A^2)\mathcal{N}(\mu_B,\sigma_B^2) \\ &= \ldots \\ &= \text{see Appendix 1 for proof} \\ &= \ldots \\ &= \mathcal{N}(\mu,\sigma^2)/\eta \end{aligned}$$

In this example:

$$\mu = \mu_B + \frac{\sigma_B^2}{\sigma_A^2+\sigma_B^2}(\mu_A-\mu_B) = 4$$

$$\sigma^2 = \sigma_B^2 - \frac{\sigma_B^2}{\sigma_A^2+\sigma_B^2}\sigma_B^2 = 4/5$$

Kalman Filter with 1D state: the update step



Another example:

$$\mu = \mu_B + \frac{\sigma_B^2}{\sigma_A^2 + \sigma_B^2}(\mu_A - \mu_B) = 5$$

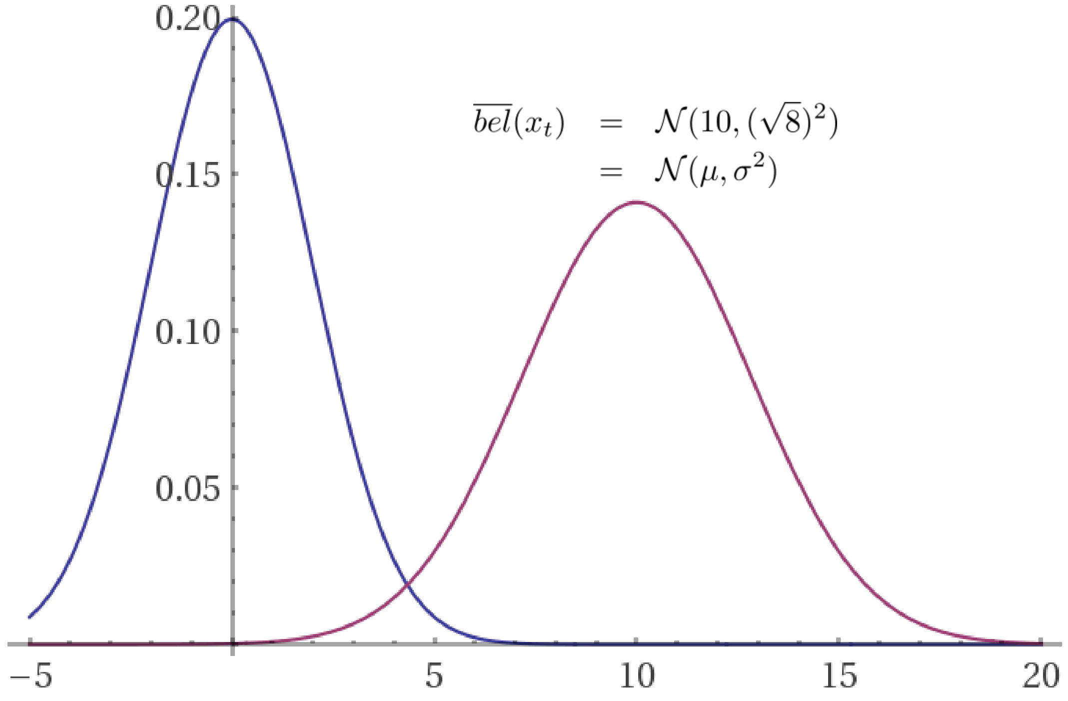
$$\sigma^2 = \sigma_B^2 - \frac{\sigma_B^2}{\sigma_A^2 + \sigma_B^2}\sigma_B^2 = \sigma_B^2/2 = 2$$

Kalman Filter with 1D state: the update step

Take-home message: new observations, no matter how noisy, always **reduce uncertainty** in the posterior. The mean of the posterior, on the other hand, only changes when there is a nonzero prediction residual.

Kalman Filter with 1D state: the propagation/prediction step

$$\begin{aligned} bel(x_{t-1}) &= \mathcal{N}(0, 2^2) \\ &= \mathcal{N}(\mu_C, \sigma_C^2) \end{aligned}$$



$$\begin{aligned} \overline{bel}(x_t) &= \mathcal{N}(10, (\sqrt{8})^2) \\ &= \mathcal{N}(\mu, \sigma^2) \end{aligned}$$

Suppose that the dynamics model is

$x_t = x_{t-1} + u_{t-1} + w_{t-1}$ with $w_{t-1} \sim \mathcal{N}(0, q^2)$

and you applied the command $u_{t-1} = 10$. Then

$$\begin{aligned} \mu &= \mathbb{E}[x_t | z_{0:t-1}, u_{0:t-1}] \\ &= \mathbb{E}[x_{t-1} + u_{t-1} + w_{t-1} | z_{0:t-1}, u_{0:t-1}] \\ &= \mathbb{E}[x_{t-1} + w_{t-1} | z_{0:t-1}, u_{0:t-1}] + u_{t-1} \\ &= \mathbb{E}[x_{t-1} | z_{0:t-1}, u_{0:t-1}] + u_{t-1} \\ &= \mathbb{E}[x_{t-1} | z_{0:t-1}, u_{0:t-2}] + u_{t-1} \\ &= \mu_C + u_{t-1} \end{aligned}$$

Recall: this notation means expected value with respect to conditional expectation, i.e

$$\int x_t \, p(x_t | z_{0:t-1}, u_{0:t-1}) \, dx_t$$

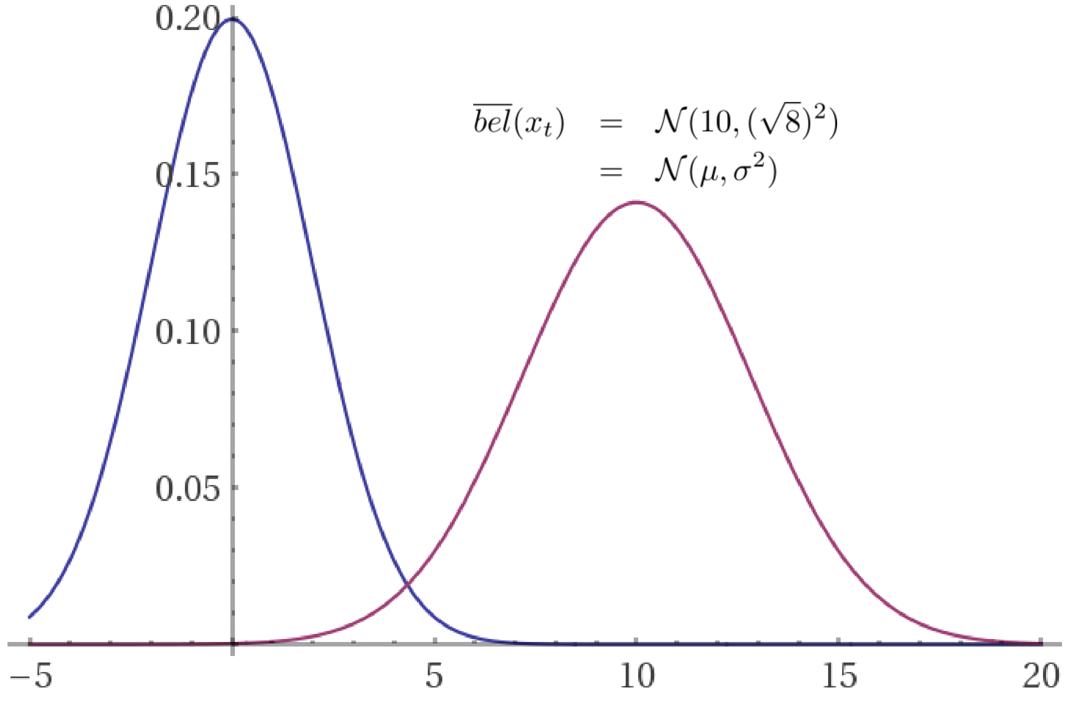
$$= \int x_t \, \overline{bel}(x_t) \, dx_t$$

Control is a constant with respect to the distribution $\overline{bel}(x_t)$

Dynamics noise is zero mean, and independent of observations and controls

Kalman Filter with 1D state: the propagation/prediction step

$$\begin{aligned} bel(x_{t-1}) &= \mathcal{N}(0, 2^2) \\ &= \mathcal{N}(\mu_C, \sigma_C^2) \end{aligned}$$



Suppose that the dynamics model is

$x_t = x_{t-1} + u_{t-1} + w_{t-1}$ with $w_{t-1} \sim \mathcal{N}(0, q^2)$

and you applied the command $u_{t-1} = 10$. Then

$$\begin{aligned} \mu &= \mathbb{E}[x_t | z_{0:t-1}, u_{0:t-1}] \\ &= \mathbb{E}[x_{t-1} + u_{t-1} + w_{t-1} | z_{0:t-1}, u_{0:t-1}] \\ &= \mathbb{E}[x_{t-1} + w_{t-1} | z_{0:t-1}, u_{0:t-1}] + u_{t-1} \\ &= \mathbb{E}[x_{t-1} | z_{0:t-1}, u_{0:t-1}] + u_{t-1} \\ &= \mathbb{E}[x_{t-1} | z_{0:t-1}, u_{0:t-2}] + u_{t-1} \\ &= \mu_C + u_{t-1} \end{aligned}$$

$$\begin{aligned} \sigma^2 &= \mathrm{Cov}[x_t | z_{0:t-1}, u_{0:t-1}] \\ &= \mathrm{Cov}[x_{t-1} + u_{t-1} + w_{t-1} | z_{0:t-1}, u_{0:t-1}] \\ &= \mathrm{Cov}[x_{t-1} + w_{t-1} | z_{0:t-1}, u_{0:t-1}] \end{aligned}$$

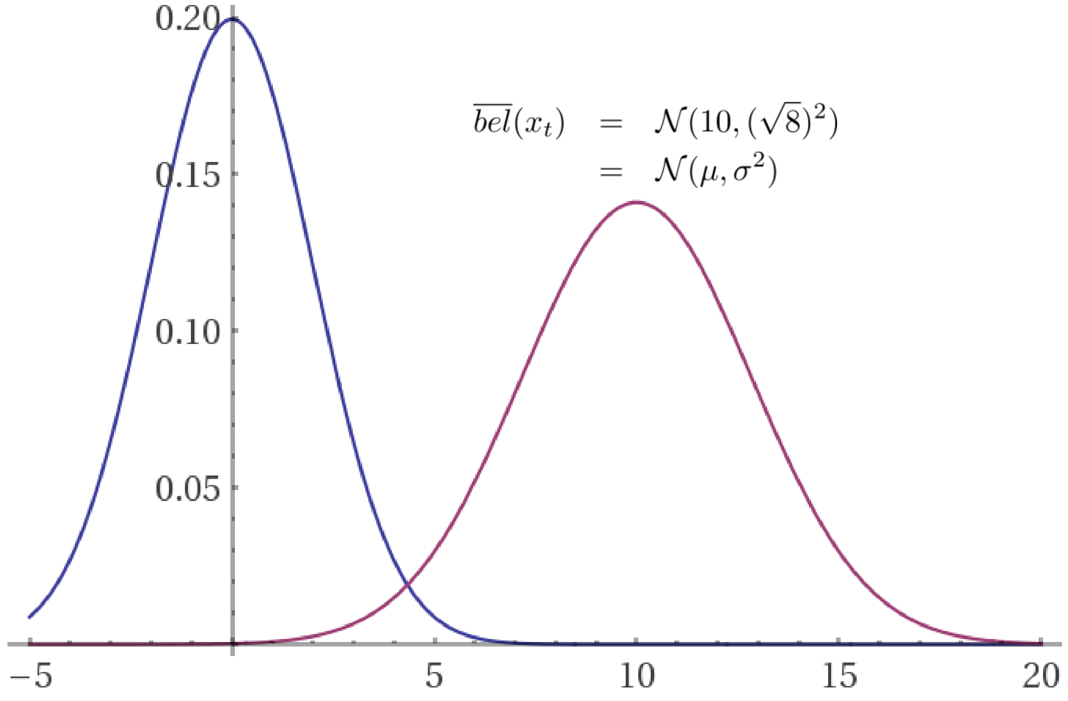
Recall: this notation means covariance with respect to conditional expectation, i.e

$$\mathrm{Cov}[x_t | z_{0:t-1}, u_{0:t-1}] = \mathbb{E}[x_t^2 | z_{0:t-1}, u_{0:t-1}] - (\mathbb{E}[x_t | z_{0:t-1}, u_{0:t-1}])^2$$

Kalman Filter with 1D state: the propagation/prediction step

$$
\begin{aligned}
bel(x_{t-1}) &= \mathcal{N}(0, 2^2) \\
&= \mathcal{N}(\mu_C, \sigma_C^2)
\end{aligned}
$$

$$
\begin{aligned}
\overline{bel}(x_t) &= \mathcal{N}(10, (\sqrt{8})^2) \\
&= \mathcal{N}(\mu, \sigma^2)
\end{aligned}
$$



Suppose that the dynamics model is

$x_t = x_{t-1} + u_{t-1} + w_{t-1}$ with $w_{t-1} \sim \mathcal{N}(0, q^2)$

and you applied the command $u_{t-1} = 10$. Then

$$
\begin{aligned}
\mu &= \mathbb{E}[x_t | z_{0:t-1}, u_{0:t-1}] \\
&= \mathbb{E}[x_{t-1} + u_{t-1} + w_{t-1} | z_{0:t-1}, u_{0:t-1}] \\
&= \mathbb{E}[x_{t-1} + w_{t-1} | z_{0:t-1}, u_{0:t-1}] + u_{t-1} \\
&= \mathbb{E}[x_{t-1} | z_{0:t-1}, u_{0:t-1}] + u_{t-1} \\
&= \mathbb{E}[x_{t-1} | z_{0:t-1}, u_{0:t-2}] + u_{t-1} \\
&= \mu_C + u_{t-1}
\end{aligned}
$$

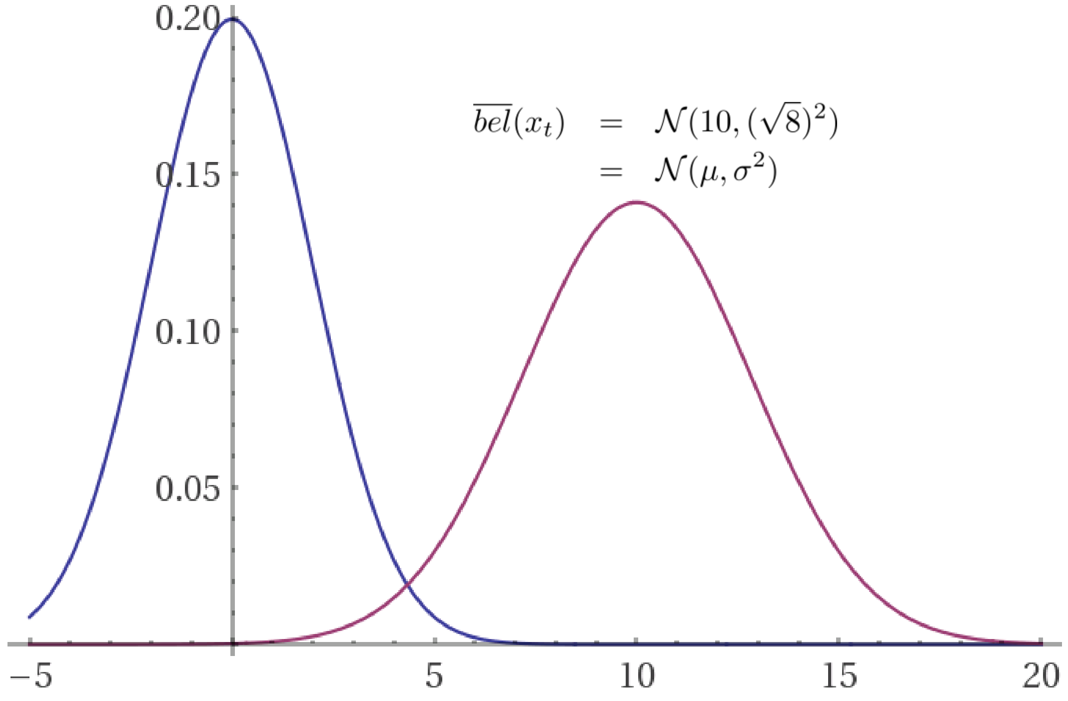
$$
\begin{aligned}
\sigma^2 &= \mathrm{Cov}[x_t | z_{0:t-1}, u_{0:t-1}] \\
&= \mathrm{Cov}[x_{t-1} + u_{t-1} + w_{t-1} | z_{0:t-1}, u_{0:t-1}] \\
&= \mathrm{Cov}[x_{t-1} + w_{t-1} | z_{0:t-1}, u_{0:t-1}]
\end{aligned}
$$

Recall: covariance neglects addition of constant terms, i.e. Cov(X+b) = Cov(X)

Kalman Filter with 1D state: the propagation/prediction step

$$\begin{aligned} bel(x_{t-1}) &= \mathcal{N}(0, 2^2) \\ &= \mathcal{N}(\mu_C, \sigma_C^2) \end{aligned}$$

$$\begin{aligned} \overline{bel}(x_t) &= \mathcal{N}(10, (\sqrt{8})^2) \\ &= \mathcal{N}(\mu, \sigma^2) \end{aligned}$$



Suppose that the dynamics model is

$x_t = x_{t-1} + u_{t-1} + w_{t-1}$ with $w_{t-1} \sim \mathcal{N}(0, q^2)$

and you applied the command $u_{t-1} = 10$. Then

$$\begin{aligned} \mu &= \mathbb{E}[x_t | z_{0:t-1}, u_{0:t-1}] \\ &= \mathbb{E}[x_{t-1} + u_{t-1} + w_{t-1} | z_{0:t-1}, u_{0:t-1}] \\ &= \mathbb{E}[x_{t-1} + w_{t-1} | z_{0:t-1}, u_{0:t-1}] + u_{t-1} \\ &= \mathbb{E}[x_{t-1} | z_{0:t-1}, u_{0:t-1}] + u_{t-1} \\ &= \mathbb{E}[x_{t-1} | z_{0:t-1}, u_{0:t-2}] + u_{t-1} \\ &= \mu_C + u_{t-1} \end{aligned}$$

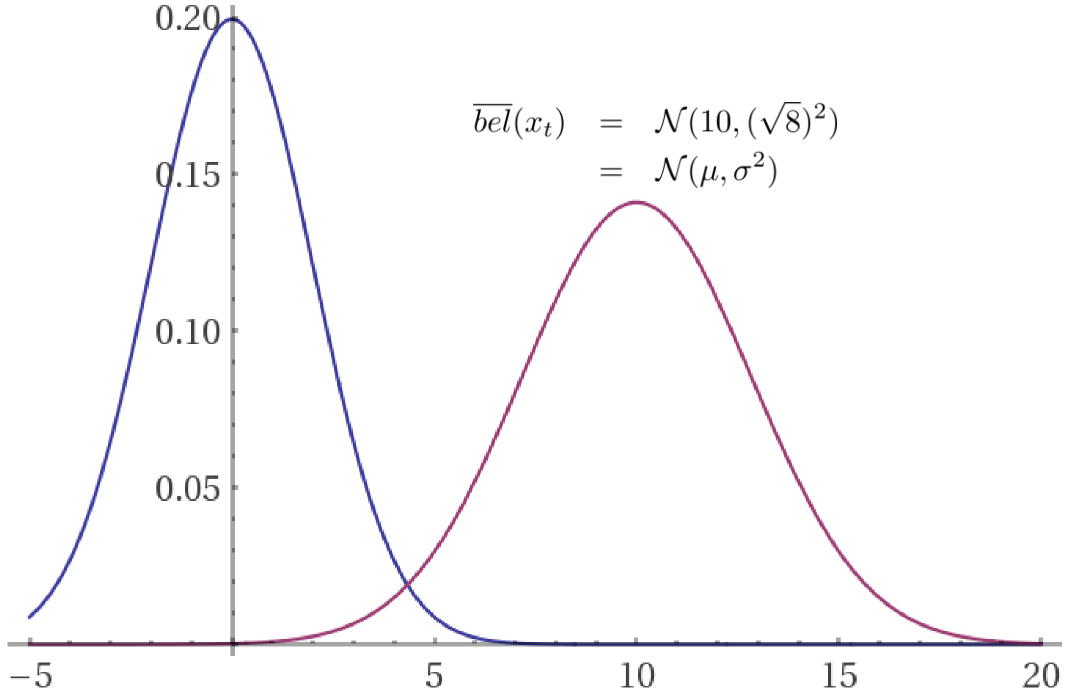
$$\begin{aligned} \sigma^2 &= \mathrm{Cov}[x_t | z_{0:t-1}, u_{0:t-1}] \\ &= \mathrm{Cov}[x_{t-1} + u_{t-1} + w_{t-1} | z_{0:t-1}, u_{0:t-1}] \\ &= \mathrm{Cov}[x_{t-1} + w_{t-1} | z_{0:t-1}, u_{0:t-1}] \\ &= \mathrm{Cov}[x_{t-1} | z_{0:t-1}, u_{0:t-1}] + \mathrm{Cov}[w_{t-1} | z_{0:t-1}, u_{0:t-1}] - 2\mathrm{Cov}[x_{t-1}, w_{t-1} | z_{0:t-1}, u_{0:t-1}] \end{aligned}$$

Recall: Cov(X+Y)=Cov(X)+Cov(Y)-2Cov(X,Y)

Recall: we denote Cov(X,X)=Cov(X) as a shorthand

Kalman Filter with 1D state: the propagation/prediction step

$$
\begin{aligned}
bel(x_{t-1}) &= \mathcal{N}(0, 2^2) \\
&= \mathcal{N}(\mu_C, \sigma_C^2)
\end{aligned}
$$



Suppose that the dynamics model is

$x_t = x_{t-1} + u_{t-1} + w_{t-1}$ with $w_{t-1} \sim \mathcal{N}(0, q^2)$

and you applied the command $u_{t-1} = 10$. Then

$$
\begin{aligned}
\mu &= \mathbb{E}[x_t | z_{0:t-1}, u_{0:t-1}] \\
&= \mathbb{E}[x_{t-1} + u_{t-1} + w_{t-1} | z_{0:t-1}, u_{0:t-1}] \\
&= \mathbb{E}[x_{t-1} + w_{t-1} | z_{0:t-1}, u_{0:t-1}] + u_{t-1} \\
&= \mathbb{E}[x_{t-1} | z_{0:t-1}, u_{0:t-1}] + u_{t-1} \\
&= \mathbb{E}[x_{t-1} | z_{0:t-1}, u_{0:t-2}] + u_{t-1} \\
&= \mu_C + u_{t-1}
\end{aligned}
$$

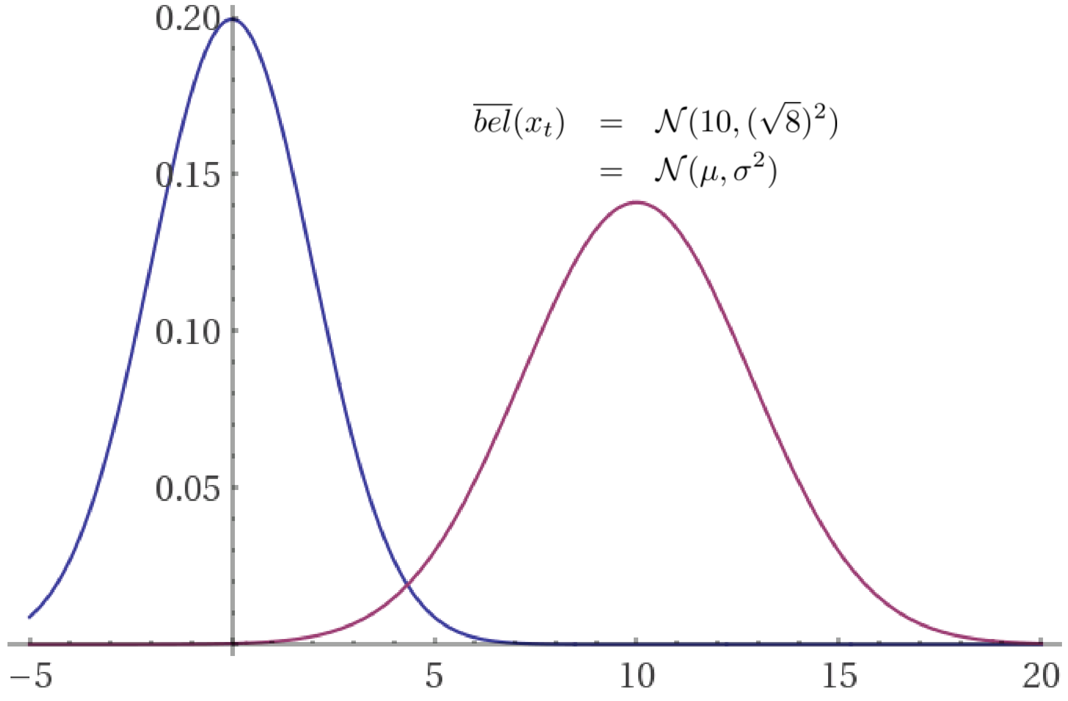
$$
\begin{aligned}
\sigma^2 &= \mathrm{Cov}[x_t | z_{0:t-1}, u_{0:t-1}] \\
&= \mathrm{Cov}[x_{t-1} + u_{t-1} + w_{t-1} | z_{0:t-1}, u_{0:t-1}] \\
&= \mathrm{Cov}[x_{t-1} + w_{t-1} | z_{0:t-1}, u_{0:t-1}] \\
&= \mathrm{Cov}[x_{t-1} | z_{0:t-1}, u_{0:t-1}] + \mathrm{Cov}[w_{t-1} | z_{0:t-1}, u_{0:t-1}] - 2\mathrm{Cov}[x_{t-1}, w_{t-1} | z_{0:t-1}, u_{0:t-1}]
\end{aligned}
$$

We assumed dynamics noise is independent of past measurement and controls

We assumed noise variables are independent of state. So this covariance is zero.

Kalman Filter with 1D state: the propagation/prediction step

$$
\begin{aligned}
bel(x_{t-1}) &= \mathcal{N}(0, 2^2) \\
&= \mathcal{N}(\mu_C, \sigma_C^2)
\end{aligned}
$$



Suppose that the dynamics model is

$x_t = x_{t-1} + u_{t-1} + w_{t-1}$ with $w_{t-1} \sim \mathcal{N}(0, q^2)$

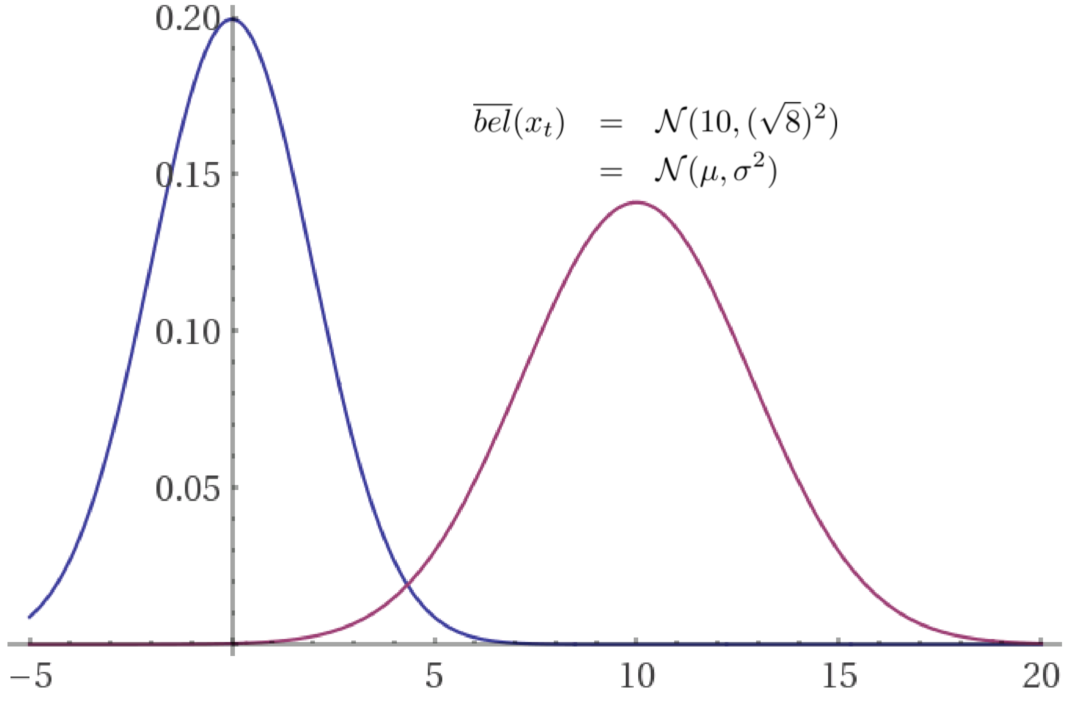
and you applied the command $u_{t-1} = 10$. Then

$$
\begin{aligned}
\mu &= \mathbb{E}[x_t | z_{0:t-1}, u_{0:t-1}] \\
&= \mathbb{E}[x_{t-1} + u_{t-1} + w_{t-1} | z_{0:t-1}, u_{0:t-1}] \\
&= \mathbb{E}[x_{t-1} + w_{t-1} | z_{0:t-1}, u_{0:t-1}] + u_{t-1} \\
&= \mathbb{E}[x_{t-1} | z_{0:t-1}, u_{0:t-1}] + u_{t-1} \\
&= \mathbb{E}[x_{t-1} | z_{0:t-1}, u_{0:t-2}] + u_{t-1} \\
&= \mu_C + u_{t-1}
\end{aligned}
$$

$$
\begin{aligned}
\sigma^2 &= \mathrm{Cov}[x_t | z_{0:t-1}, u_{0:t-1}] \\
&= \mathrm{Cov}[x_{t-1} + u_{t-1} + w_{t-1} | z_{0:t-1}, u_{0:t-1}] \\
&= \mathrm{Cov}[x_{t-1} + w_{t-1} | z_{0:t-1}, u_{0:t-1}] \\
&= \mathrm{Cov}[x_{t-1} | z_{0:t-1}, u_{0:t-1}] + \mathrm{Cov}[w_{t-1} | z_{0:t-1}, u_{0:t-1}] - 2\mathrm{Cov}[x_{t-1}, w_{t-1} | z_{0:t-1}, u_{0:t-1}] \\
&= \mathrm{Cov}[x_{t-1} | z_{0:t-1}, u_{0:t-1}] + \mathrm{Cov}[w_{t-1}]
\end{aligned}
$$

Kalman Filter with 1D state: the propagation/prediction step

$$\begin{aligned} bel(x_{t-1}) &= \mathcal{N}(0, 2^2) \\ &= \mathcal{N}(\mu_C, \sigma_C^2) \end{aligned}$$



$$\begin{aligned} \overline{bel}(x_t) &= \mathcal{N}(10, (\sqrt{8})^2) \\ &= \mathcal{N}(\mu, \sigma^2) \end{aligned}$$

Suppose that the dynamics model is

$x_t = x_{t-1} + u_{t-1} + w_{t-1}$ with $w_{t-1} \sim \mathcal{N}(0, q^2)$

and you applied the command $u_{t-1} = 10$. Then

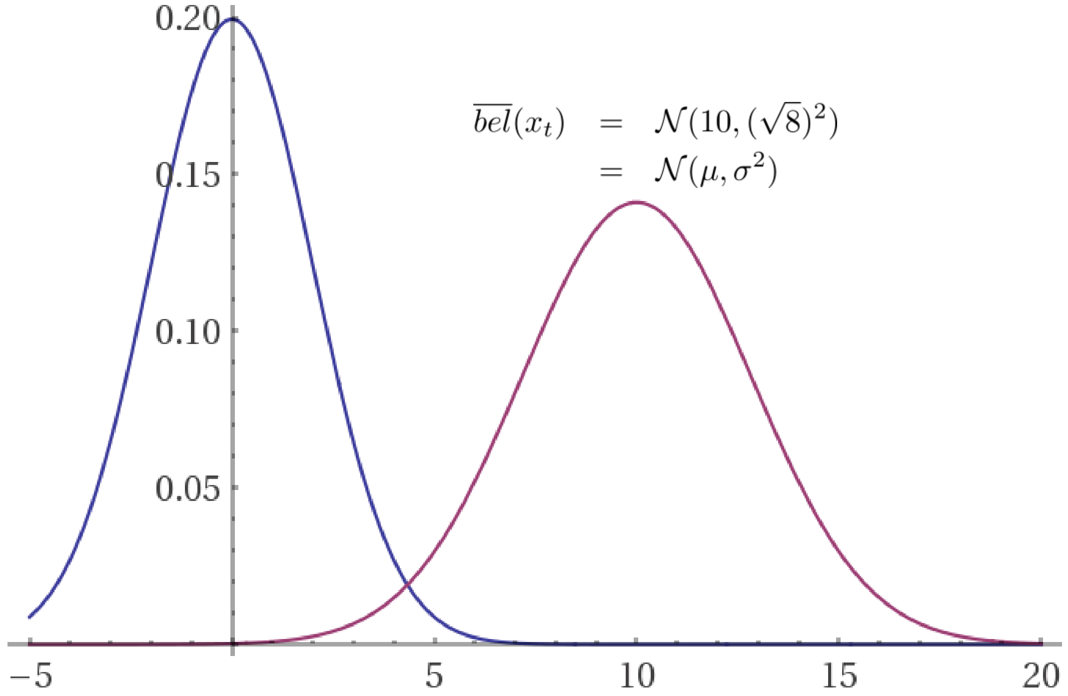
$$\begin{aligned} \mu &= \mathbb{E}[x_t | z_{0:t-1}, u_{0:t-1}] \\ &= \mathbb{E}[x_{t-1} + u_{t-1} + w_{t-1} | z_{0:t-1}, u_{0:t-1}] \\ &= \mathbb{E}[x_{t-1} + w_{t-1} | z_{0:t-1}, u_{0:t-1}] + u_{t-1} \\ &= \mathbb{E}[x_{t-1} | z_{0:t-1}, u_{0:t-1}] + u_{t-1} \\ &= \mathbb{E}[x_{t-1} | z_{0:t-1}, u_{0:t-2}] + u_{t-1} \\ &= \mu_C + u_{t-1} \end{aligned}$$

$$\begin{aligned} \sigma^2 &= \mathrm{Cov}[x_t | z_{0:t-1}, u_{0:t-1}] \\ &= \mathrm{Cov}[x_{t-1} + u_{t-1} + w_{t-1} | z_{0:t-1}, u_{0:t-1}] \\ &= \mathrm{Cov}[x_{t-1} + w_{t-1} | z_{0:t-1}, u_{0:t-1}] \\ &= \mathrm{Cov}[x_{t-1} | z_{0:t-1}, u_{0:t-1}] + \mathrm{Cov}[w_{t-1} | z_{0:t-1}, u_{0:t-1}] - 2\mathrm{Cov}[x_{t-1}, w_{t-1} | z_{0:t-1}, u_{0:t-1}] \\ &= \mathrm{Cov}[x_{t-1} | z_{0:t-1}, u_{0:t-1}] + \mathrm{Cov}[w_{t-1}] \\ &= \mathrm{Cov}[x_{t-1} | z_{0:t-2}, u_{0:t-2}] + \mathrm{Cov}[w_{t-1}] \end{aligned}$$

Kalman Filter with 1D state: the propagation/prediction step

$$
\begin{aligned}
bel(x_{t-1}) &= \mathcal{N}(0, 2^2) \\
&= \mathcal{N}(\mu_C, \sigma_C^2)
\end{aligned}
$$

$$
\begin{aligned}
\overline{bel}(x_t) &= \mathcal{N}(10, (\sqrt{8})^2) \\
&= \mathcal{N}(\mu, \sigma^2)
\end{aligned}
$$



Suppose that the dynamics model is

$$x_t = x_{t-1} + u_{t-1} + w_{t-1} \quad \text{with} \quad w_{t-1} \sim \mathcal{N}(0, q^2)$$

and you applied the command $u_{t-1} = 10$. Then

$$
\begin{aligned}
\mu &= \mathbb{E}[x_t | z_{0:t-1}, u_{0:t-1}] \\
&= \mathbb{E}[x_{t-1} + u_{t-1} + w_{t-1} | z_{0:t-1}, u_{0:t-1}] \\
&= \mathbb{E}[x_{t-1} + w_{t-1} | z_{0:t-1}, u_{0:t-1}] + u_{t-1} \\
&= \mathbb{E}[x_{t-1} | z_{0:t-1}, u_{0:t-1}] + u_{t-1} \\
&= \mathbb{E}[x_{t-1} | z_{0:t-1}, u_{0:t-2}] + u_{t-1} \\
&= \mu_C + u_{t-1}
\end{aligned}
$$

$$
\begin{aligned}
\sigma^2 &= \text{Cov}[x_t | z_{0:t-1}, u_{0:t-1}] \\
&= \text{Cov}[x_{t-1} + u_{t-1} + w_{t-1} | z_{0:t-1}, u_{0:t-1}] \\
&= \text{Cov}[x_{t-1} + w_{t-1} | z_{0:t-1}, u_{0:t-1}] \\
&= \text{Cov}[x_{t-1} | z_{0:t-1}, u_{0:t-1}] + \text{Cov}[w_{t-1} | z_{0:t-1}, u_{0:t-1}] - 2\text{Cov}[x_{t-1}, w_{t-1} | z_{0:t-1}, u_{0:t-1}] \\
&= \text{Cov}[x_{t-1} | z_{0:t-1}, u_{0:t-1}] + \text{Cov}[w_{t-1}] \\
&= \text{Cov}[x_{t-1} | z_{0:t-2}, u_{0:t-2}] + \text{Cov}[w_{t-1}] \\
&= \sigma_C^2 + q^2
\end{aligned}
$$

Kalman Filter with 1D state: the propagation/prediction step

Take home message: uncertainty **increases** after the prediction step, because we are speculating about the future.

Kalman Filter Algorithm

1. Algorithm **Kalman_filter**($\mu_{t-1}, \Sigma_{t-1}, u_t, z_t$):

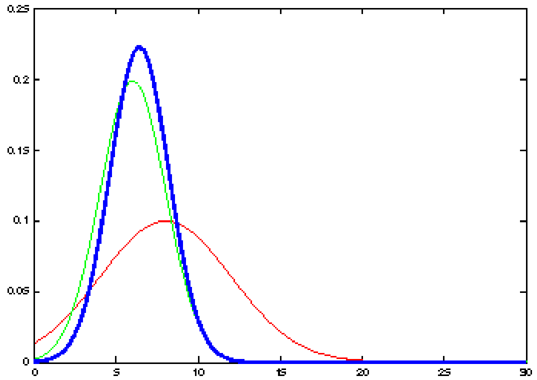
2. Prediction:
3. $\bar{\mu}_t = A_t \mu_{t-1} + B_t u_t$
4. $\bar{\Sigma}_t = A_t \Sigma_{t-1} A_t^T + R_t$

5. Correction:
6. $K_t = \bar{\Sigma}_t C_t^T (C_t \bar{\Sigma}_t C_t^T + Q_t)^{-1}$
7. $\mu_t = \bar{\mu}_t + K_t (z_t - C_t \bar{\mu}_t)$
8. $\Sigma_t = (I - K_t C_t) \bar{\Sigma}_t$

9. Return μ_t, Σ_t

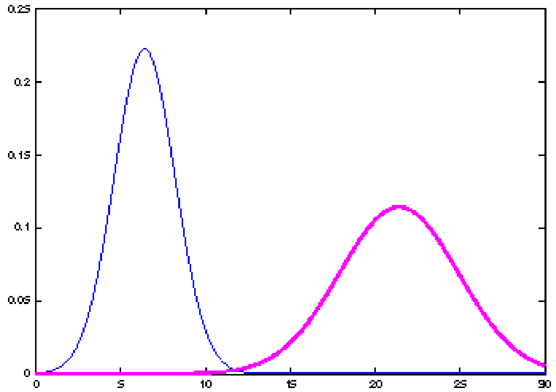
The Prediction-Correction-Cycle

Prediction

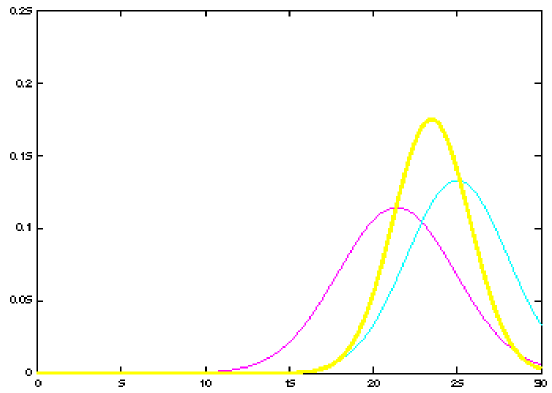


$$\overline{bel}(x_t) = \begin{cases} \bar{\mu}_t = a_t \mu_{t-1} + b_t u_t \\ \bar{\sigma}_t^2 = a_t^2 \sigma_t^2 + \sigma_{act,t}^2 \end{cases}$$

$$\overline{bel}(x_t) = \begin{cases} \bar{\mu}_t = A_t \mu_{t-1} + B_t u_t \\ \bar{\Sigma}_t = A_t \Sigma_{t-1} A_t^T + R_t \end{cases}$$

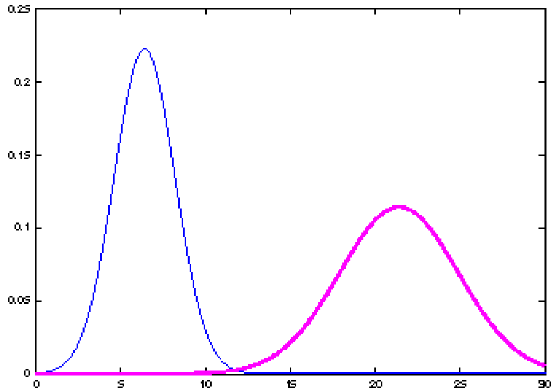


The Prediction-Correction-Cycle



$$bel(x_t) = \begin{cases} \mu_t = \bar{\mu}_t + K_t(z_t - \bar{\mu}_t) \\ \sigma_t^2 = (1-K_t)\bar{\sigma}_t^2 \end{cases}, K_t = \frac{\bar{\sigma}_t^2}{\bar{\sigma}_t^2 + \bar{\sigma}_{obs,t}^2}$$

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



Correction

The Prediction-Correction-Cycle

Prediction

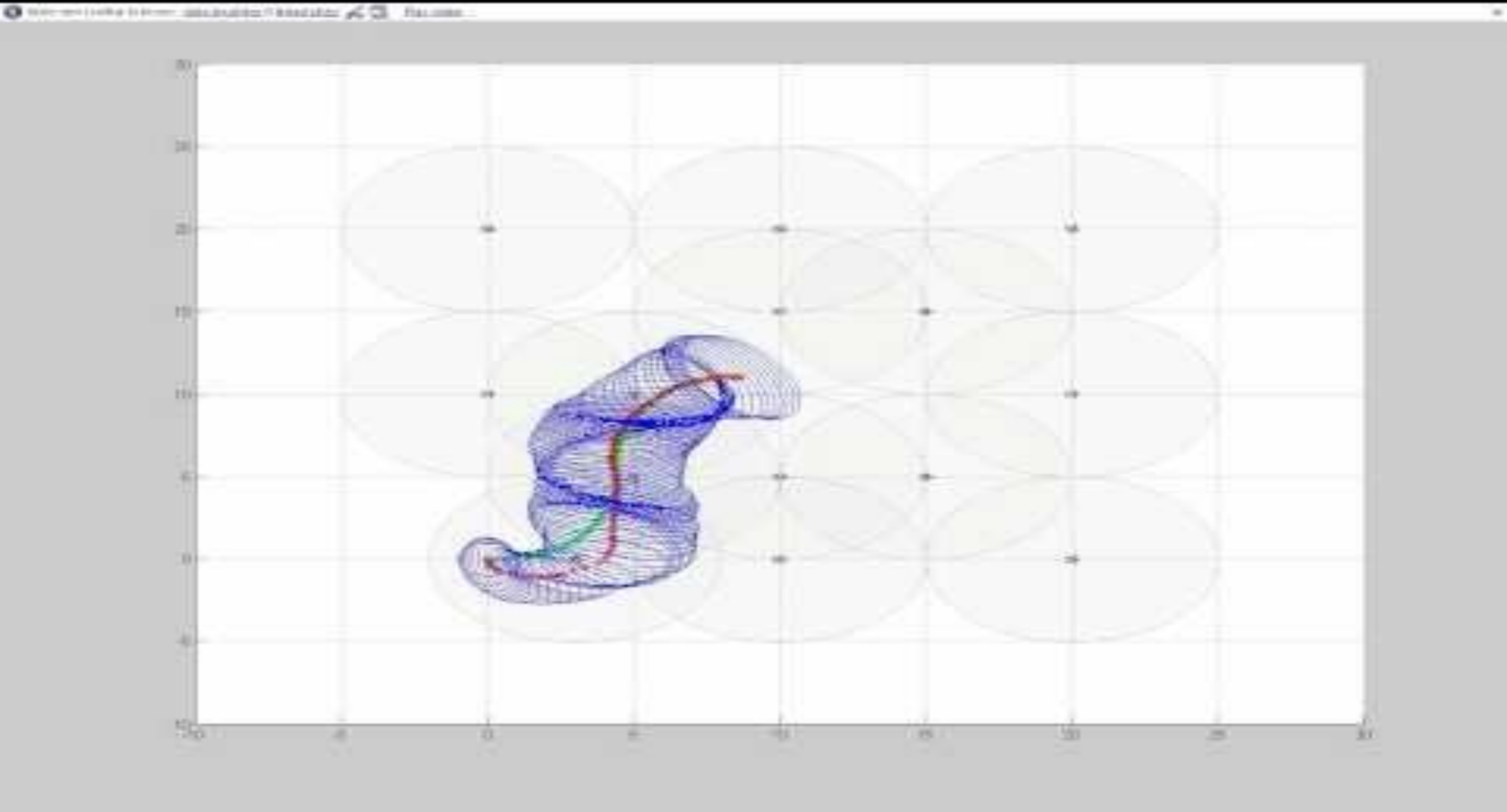
$$\overline{bel}(x_t) = \begin{cases} \bar{\mu}_t = a_t \mu_{t-1} + b_t u_t \\ \bar{\sigma}_t^2 = a_t^2 \sigma_t^2 + \sigma_{act,t}^2 \end{cases}$$

$$\overline{bel}(x_t) = \begin{cases} \bar{\mu}_t = A_t \mu_{t-1} + B_t u_t \\ \bar{\Sigma}_t = A_t \Sigma_{t-1} A_t^T + R_t \end{cases}$$

Correction

$$bel(x_t) = \begin{cases} \mu_t = \bar{\mu}_t + K_t (z_t - \bar{\mu}_t) \\ \sigma_t^2 = (1 - K_t) \bar{\sigma}_t^2 \end{cases}, K_t = \frac{\bar{\sigma}_t^2}{\bar{\sigma}_t^2 + \bar{\sigma}_{obs,t}^2}$$

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



Kalman Filter Summary

- Highly efficient: Polynomial in measurement dimensionality k and state dimensionality n:
$$O(k^{2.376} + n^2)$$

- Optimal for linear Gaussian systems!

- Most robotics systems are nonlinear!