Recursive Estimation

Lecture 11
The Particle Filter (PF) (continued)

ETH Zurich

Spring 2021

Date compiled: May 14, 2021

Learning Objectives

Topic: The Particle Filter (PF) (continued)

Objectives

- You can explain the *derivation of the PF*.
- You can *design and implement* a PF for a general nonlinear system and general noise distributions.
- You can explain the implementation issue of *sample impoverishment* and how to counteract it.
- You can assess for *what types of problems* a PF is suitable.

Recap: Auxiliary variables

Recall the definitions of the auxiliary random variables from last lecture:

$$
\left.
\begin{array}{ll}
\textbf{Init:} & x_{\mathrm{m}}(0) := x(0) \\[2ex]
\textbf{S1:} & x_{\mathrm{p}}(k) := q_{k-1}\big(x_{\mathrm{m}}(k{-}1), v(k{-}1)\big) \\
\textbf{S2:} & z_{\mathrm{m}}(k) := h_k\big(x_{\mathrm{p}}(k), w(k)\big) \\
 & x_{\mathrm{m}}(k) \text{ defined via its PDF} \\
 & \quad p_{x_{\mathrm{m}}(k)}(\xi) := p_{x_{\mathrm{p}}(k)|z_{\mathrm{m}}(k)}\big(\xi|\bar{z}(k)\big) \quad \forall \xi
\end{array}
\right\} \quad k = 1, 2, \ldots
$$

For all ξ and $k = 1, 2, \ldots,$

$$
\begin{aligned}
p_{x_{\mathrm{p}}(k)}(\xi) &= p_{x(k)|z(1:k-1)}\big(\xi|\bar{z}(1:k{-}1)\big) \\
p_{x_{\mathrm{m}}(k)}(\xi) &= p_{x(k)|z(1:k)}\big(\xi|\bar{z}(1:k)\big).
\end{aligned}
$$

We derived the prior update of the Particle Filter (PF) as an approximation of the Bayesian state estimator using Monte Carlo sampling. Next, we address the measurement update.

Outline

Measurement Update

Given the PDF $p_{x_\mathrm{p}(k)}$ of $x_\mathrm{p}(k) \in \mathcal{X}$ (from prior update) and after obtaining a measurement $\bar{z}(k)$, we can construct the PDF $p_{x_\mathrm{m}(k)}$ of $x_\mathrm{m}(k)$. Again, we *approximate* the PDF by Monte Carlo sampling.

Scaling of Particles by Measurement Likelihood (1/3)

- From Bayes' rule, we obtain

$$
\begin{aligned}
p_{x_\mathrm{m}(k)}(\xi) :=& p_{x_\mathrm{p}(k)|z_\mathrm{m}(k)}(\xi|\bar{z}(k)) \\
=& \frac{p_{z_\mathrm{m}(k)|x_\mathrm{p}(k)}(\bar{z}(k)|\xi)\ p_{x_\mathrm{p}(k)}(\xi)}{\underbrace{\sum_{\zeta\in\mathcal{X}} p_{z_\mathrm{m}(k)|x_\mathrm{p}(k)}(\bar{z}(k)|\zeta)\ p_{x_\mathrm{p}(k)}(\zeta)}_{\text{normalization constant, not a function of } \xi}}, \quad \forall\xi. \qquad (1)
\end{aligned}
$$

The sum is replaced by an integral in the case of $x_\mathrm{p}(k)$ being a CRV.

- We approximated $p_{x_\mathrm{p}(k)}$ by Monte Carlo samples (prior update); that is,

$$
p_{x_\mathrm{p}(k)}(\xi) \approx \frac{1}{N}\sum_{n=1}^{N}\delta(\xi - \bar{x}_\mathrm{p}^n(k)), \quad \forall\xi.
$$

Scaling of Particles by Measurement Likelihood (2/3)

- Substituting the expression for $p_{x_\mathrm{p}(k)}$ in (1),

$$p_{x_\mathrm{p}(k)|z_\mathrm{m}(k)}(\xi|\bar{z}(k)) \;\approx\; \alpha\, p_{z_\mathrm{m}(k)|x_\mathrm{p}(k)}(\bar{z}(k)|\xi)\; \sum_{n=1}^{N} \delta(\xi - \bar{x}_\mathrm{p}^n(k))$$

  for all ξ, where α is a normalization constant.
  This suggests the following form,

$$\alpha\, p_{z_\mathrm{m}(k)|x_\mathrm{p}(k)}(\bar{z}(k)|\xi)\; \sum_{n=1}^{N} \delta(\xi - \bar{x}_\mathrm{p}^n(k)) =: \sum_{n=1}^{N} \beta_n\, \delta(\xi - \bar{x}_\mathrm{p}^n(k)),$$

- We can obtain N equations by substituting $\xi = \bar{x}_\mathrm{p}^n(k)$ for $n = 1, 2, \ldots, N$ (recall that δ is zero except when its argument is zero):

$$\beta_n = \alpha\, p_{z_\mathrm{m}(k)|x_\mathrm{p}(k)}\big(\bar{z}(k)|\bar{x}_\mathrm{p}^n(k)\big) \qquad n = 1, 2, \ldots, N\,.$$

Scaling of Particles by Measurement Likelihood (3/3)

We have an additional equation, $\sum_{n=1}^{N} \beta_n = 1$, which is required for $p_{x_\mathrm{p}(k)|z_\mathrm{m}(k)}(\xi|\bar{z}(k))$ to be a valid PDF. Therefore,

$$\boxed{\alpha = \left(\sum_{n=1}^{N} p_{z_\mathrm{m}(k)|x_\mathrm{p}(k)}\big(\bar{z}(k)|\bar{x}_\mathrm{p}^n(k)\big) \right)^{-1}}.$$

- Note that we assumed that all particles $\{\bar{x}_\mathrm{p}^n(k)\}$ are distinct. However, the resulting expressions for β_n and α remain valid for the general case.
- Intuition: treat each particle separately. At points of high prior, we have many particles. The posterior has the same particles, but scaled by measurement likelihood.
- Summary: We obtained a representation for $p_{x_\mathrm{m}(k)}$,

$$p_{x_\mathrm{m}(k)}(\xi) = p_{x_\mathrm{p}(k)|z_\mathrm{m}(k)}(\xi|\bar{z}(k)) \approx \sum_{n=1}^{N} \beta_n \, \delta(\xi - \bar{x}_\mathrm{p}^n(k)), \quad \forall \xi.$$

We are not done yet – this is not in the format that we need: the particles must be of equal weight, $1/N$, as required in the next prior update.

Resampling

The objective of the resampling is to reselect N particles, where the probability of selecting particle n is β_n. Thus, we apply the basic "sampling a distribution" algorithm (see lecture #2) to the distribution $\sum_{n=1}^{N} \beta_n \, \delta(\xi - n)$.

Algorithm. Repeat N times:

- Select a random number r uniformly on $(0, 1)$.
- Pick particle $\overline{n}$ such that $\displaystyle\sum_{n=1}^{\overline{n}-1} \beta_n < r$ and $\displaystyle\sum_{n=1}^{\overline{n}} \beta_n \geq r$.

This gives N *new* particles $\bar{x}_{\mathrm{m}}^{n}(k)$, which are a subset of the old particles (because we only select from those). The new particles all have equal weight,

$$p_{x_{\mathrm{m}}(k)}(\xi) \approx \frac{1}{N} \sum_{n=1}^{N} \delta(\xi - \bar{x}_{\mathrm{m}}^{n}(k)), \quad \forall \xi$$

as required. This completes the measurement update step.

Outline

Summary

The *basic* PF algorithm is given by:

Initialization: Draw N samples $\{\bar{x}_{\mathrm{m}}^n(0)\}$ from $p_{x(0)}$.

Step 1 (S1): Prior update/Prediction step

The process equation is applied to the particles $\{\bar{x}_{\mathrm{m}}^n(k-1)\}$:

$$\bar{x}_{\mathrm{p}}^n(k) := q_{k-1}(\bar{x}_{\mathrm{m}}^n(k-1), \bar{v}^n(k-1)), \qquad \text{for } n = 1, 2, \ldots, N,$$

which requires N noise samples from $p_{v(k-1)}$.

Step 2 (S2): A posteriori update/Measurement update step

Scale each particle by the measurement likelihood:

$$\beta_n = \alpha \, p_{z(k)|x(k)}\left(\bar{z}(k)|\bar{x}_{\mathrm{p}}^n(k)\right), \qquad \text{for } n = 1, 2, \ldots, N,$$

where α is the normalization constant chosen such that $\sum_{n=1}^N \beta_n = 1$.

Resample (using the resampling algorithm in the previous slide) to get the N posterior particles $\{\bar{x}_{\mathrm{m}}^n(k)\}$, all with equal weights.

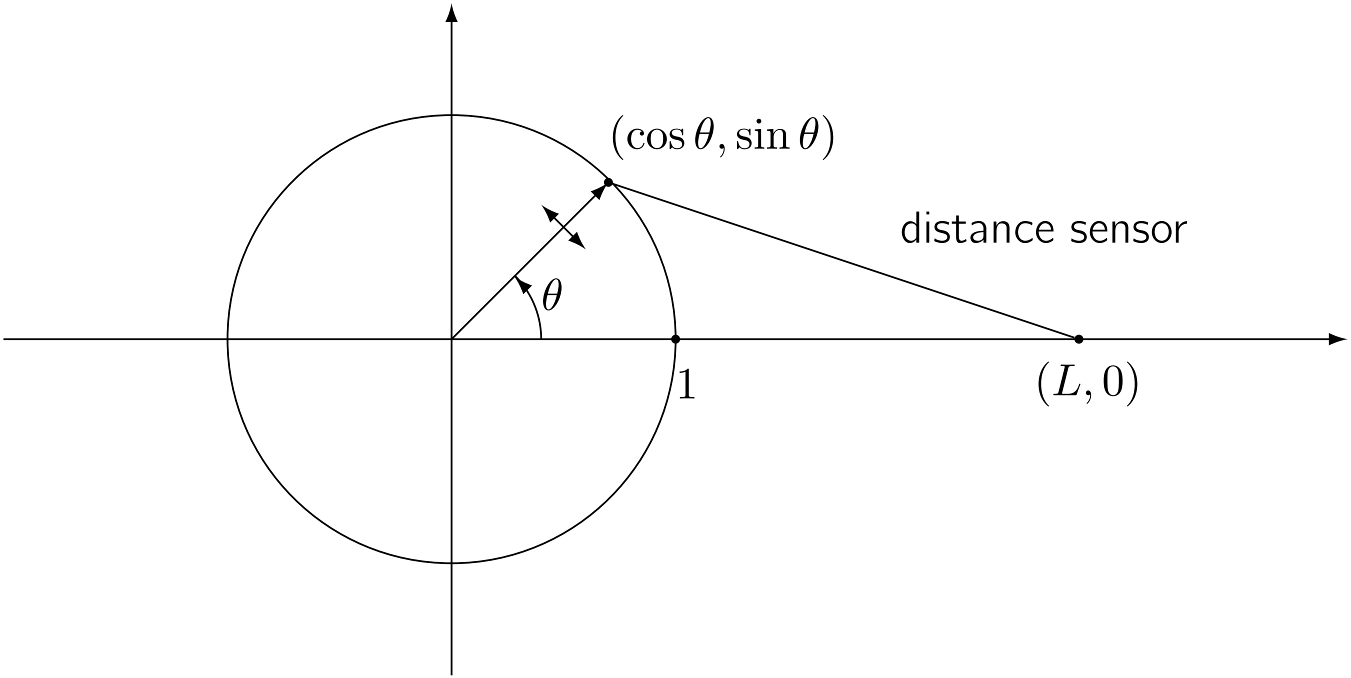
Outline

The Particle Filter (continued)

- Measurement Update
- Summary
- Example (continued)
- Sample Impoverishment
- Remarks

Example (continued) (1/3)

We continue last lecture's example, now with sensor measurements.

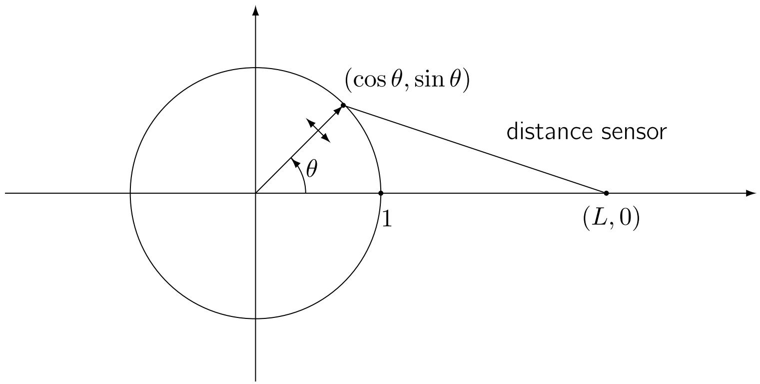


Example (continued) (2/3)

- Recall the dynamics (x_1 is the process noise bias b, and x_2 is the angle θ of the object):

$$
\begin{aligned}
x_1(k) &= x_1(k-1) \\
x_2(k) &= \text{mod}\big(x_2(k-1) + x_1(k-1) \\
&\qquad\qquad + v(k-1),\ 2\pi\big),
\end{aligned}
$$

  where $x_1(0)$ and $v(k-1)$ are both uniformly distributed on $[-\bar{s}, \bar{s}]$.



- A distance sensor at $(L, 0)$ measures the distance to the object corrupted by noise:

$$z_1(k) = \sqrt{(L - \cos x_2(k))^2 + (\sin x_2(k))^2} + w(k)$$

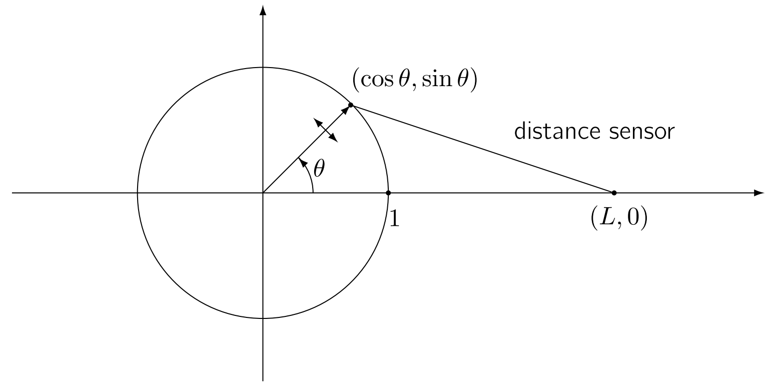
  with $w(k)$ uniformly distributed on $[-e, e]$. We receive distance measurements only occasionally.

Example (continued) (3/3)

- A half-plane measurement indicates whether the object is in the upper or the lower half plane:

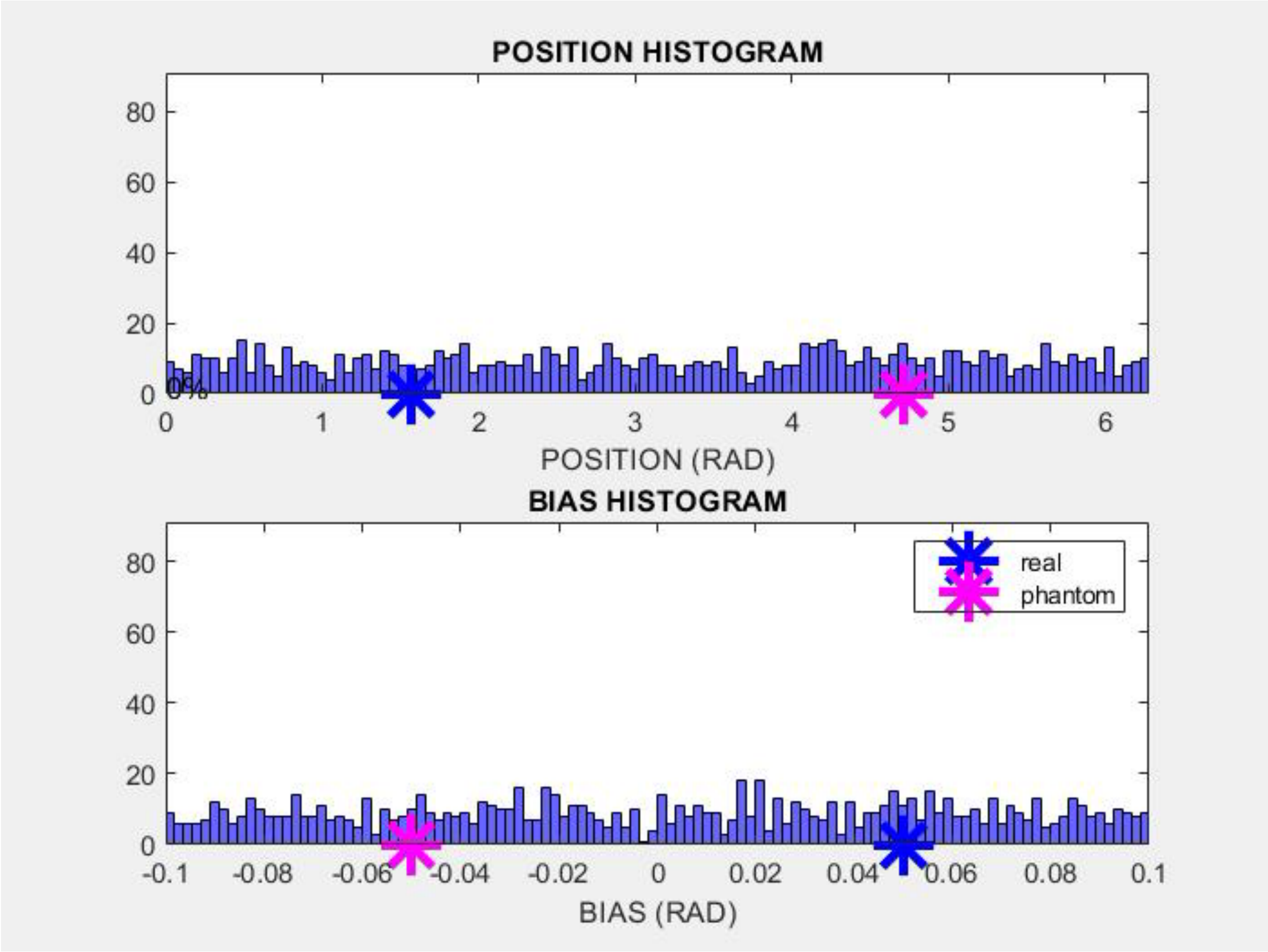
$$z_2(k) = \begin{cases} 1 & \text{if} \quad x_2(k) \in [0, \pi) \\ -1 & \text{if} \quad x_2(k) \in [\pi, 2\pi) \end{cases}$$

We only receive half-plane measurements occasionally.

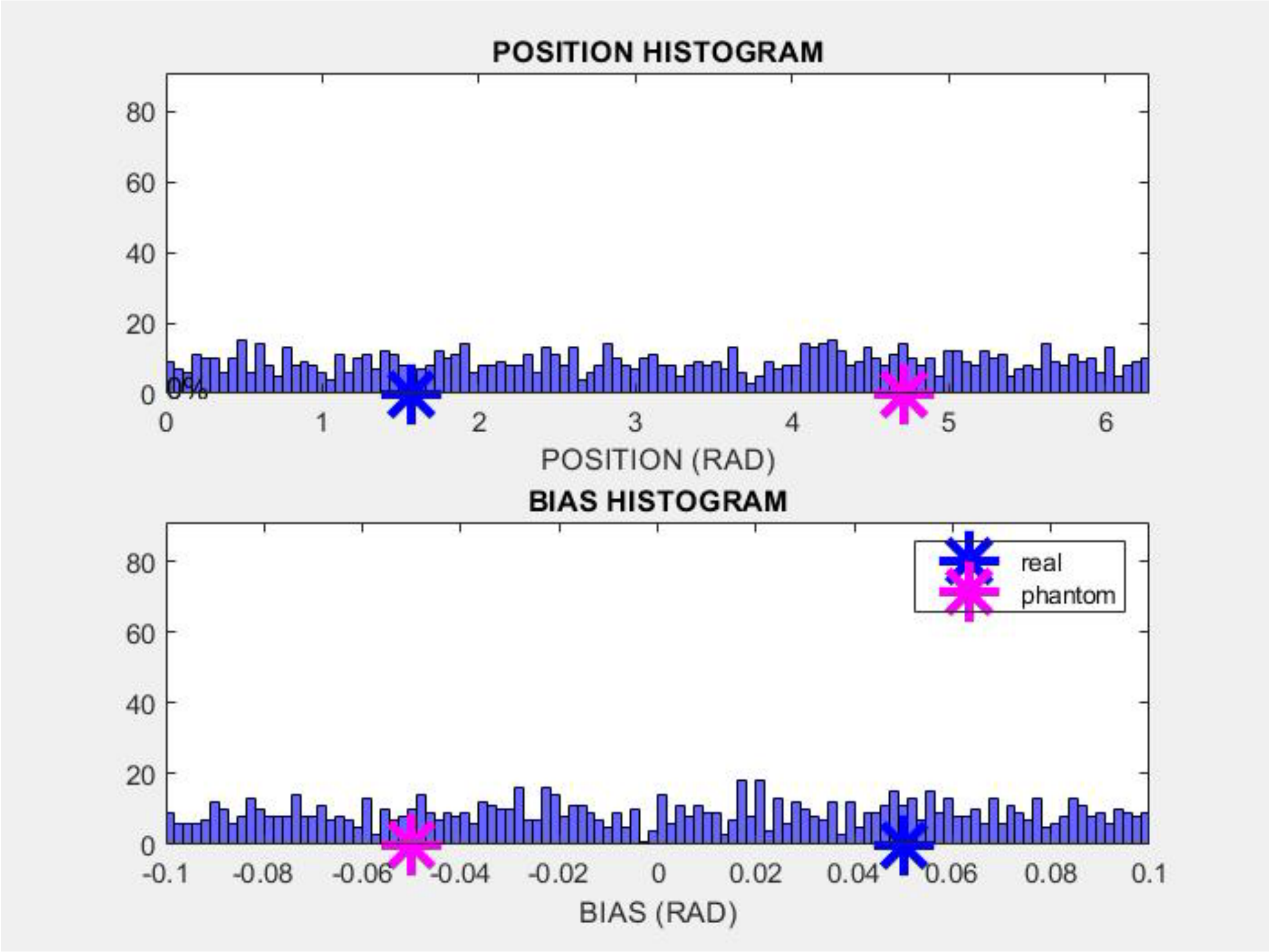


- Simulation results:
  - With distance measurements only, the approximate state PDF has two modes corresponding to the true location and the “phantom location” (mirror image). Eventually, one mode will become dominant (not always the right one).
  - The half-plane sensor allows the PF to track the right one.
  - Sample impoverishment: particles converge to only few particles or even the same one (especially for small process noise; here, in particular, the bias).

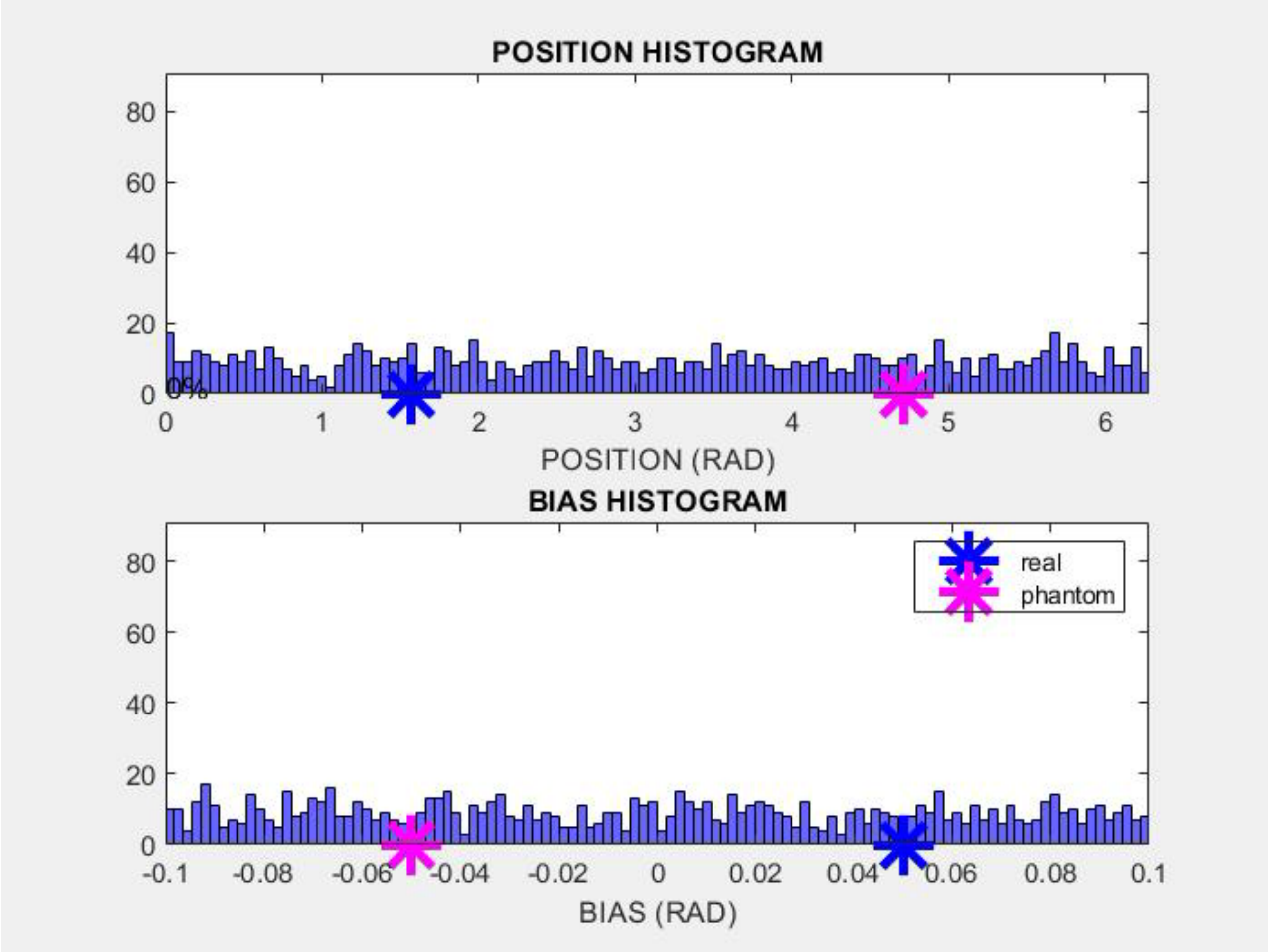
Video: Without Measurements



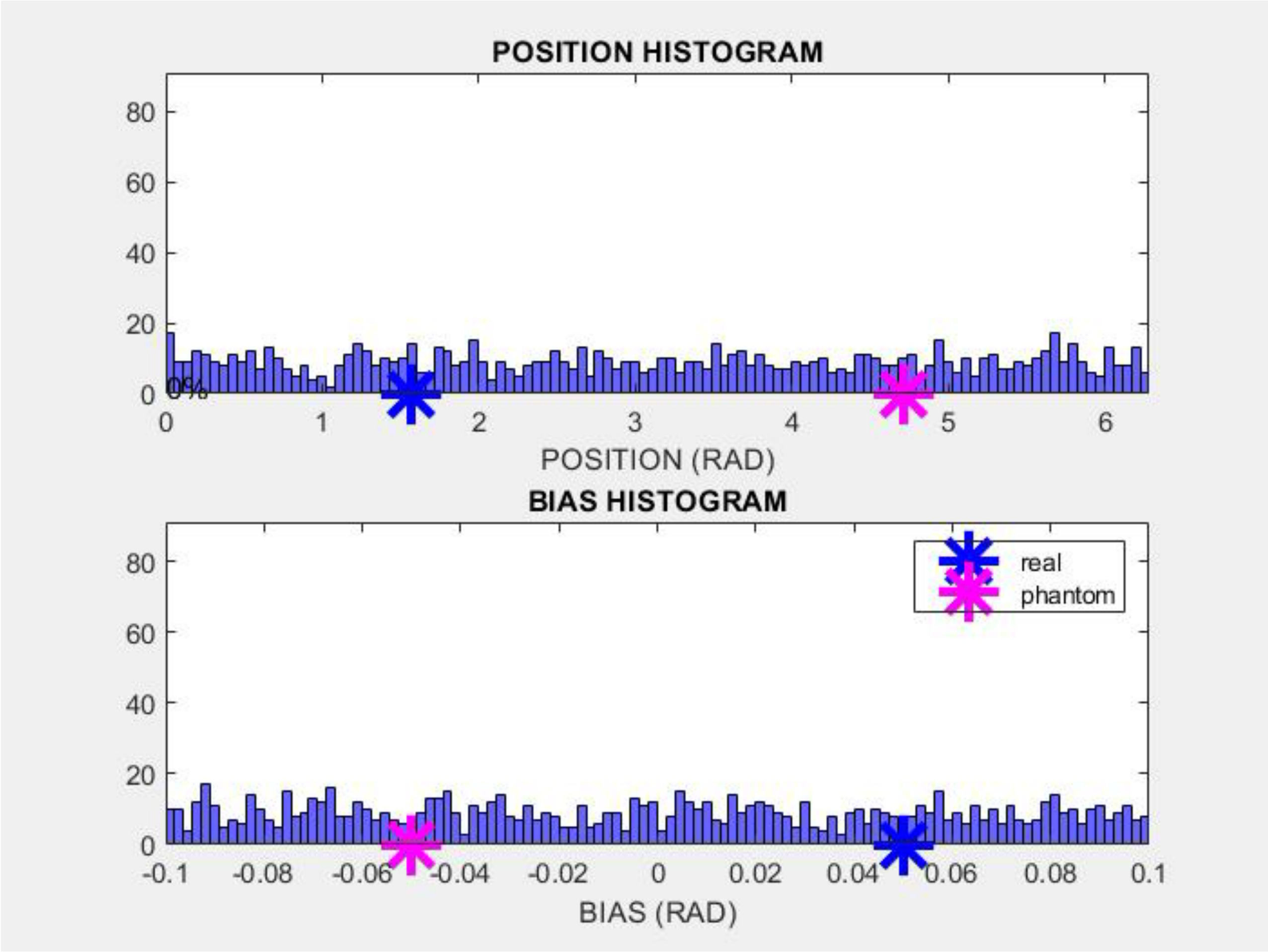
Video: With Distance Sensor - Seed 1



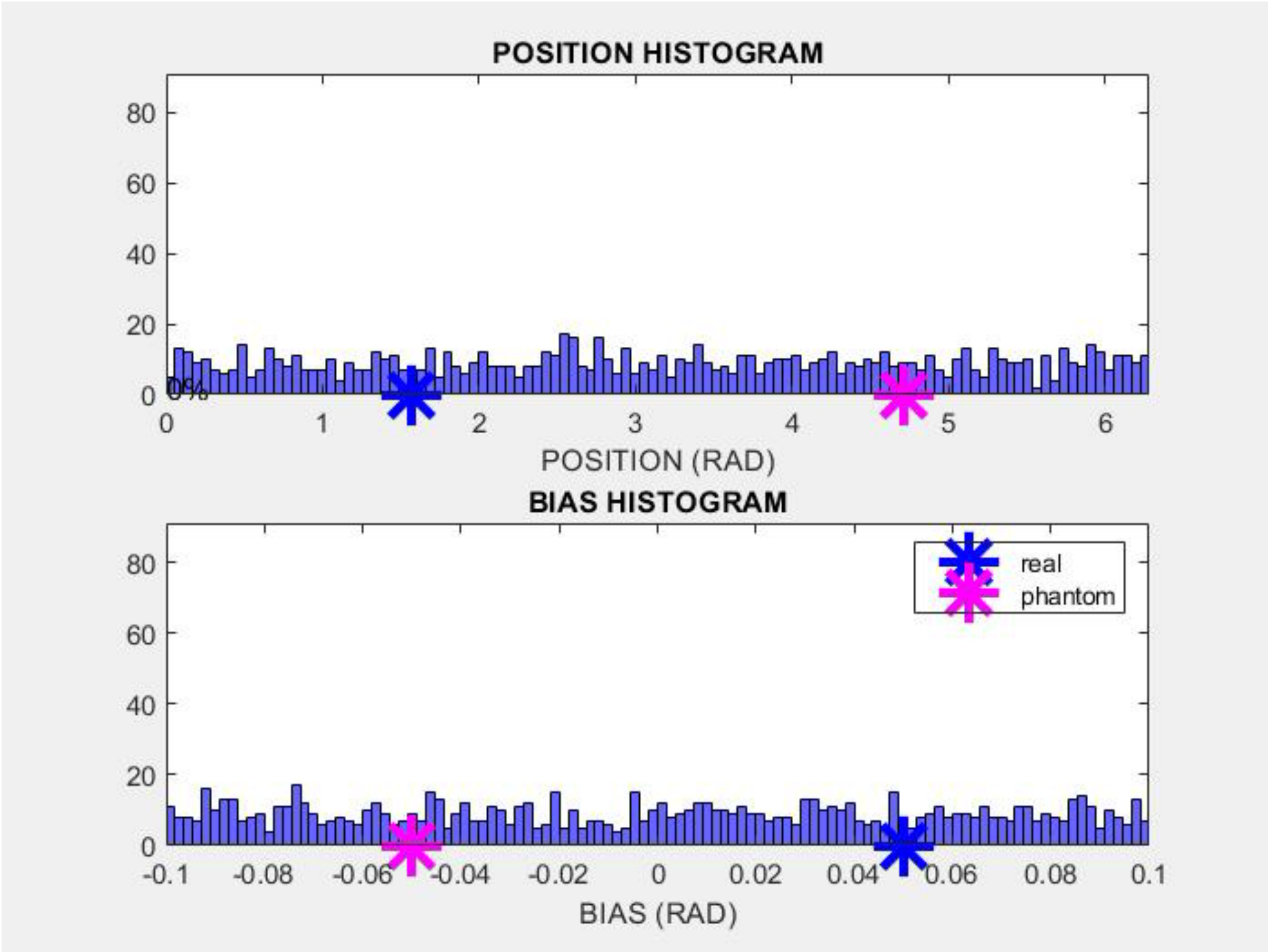
Video: With Distance Sensor - Seed 2



Video: With Both Sensors



Video: Sample Impoverishment



Outline

The Particle Filter (continued)

- Measurement Update
- Summary
- Example (continued)
- Sample Impoverishment
- Remarks

Sample Impoverishment

A possible problem of the particle filter is that all particles may converge to the same one, and these particles are therefore no longer a good representation of the state PDF. This has to do with the lumpiness of the approximation – when we resample, we only retain a subset of the particles. In the limit as $N \to \infty$, this takes an infinite amount of time to happen – however, for finite N this turns out to be an issue.

One approach (probably the simplest) to prevent sample impoverishment is roughening.

Roughening

Perturb the particles *after* resampling,

$$\bar{x}^n_{\mathrm{m}}(k) \leftarrow \bar{x}^n_{\mathrm{m}}(k) + \Delta x^n(k),$$

where $\Delta x^n(k)$ is drawn from a zero-mean, finite-variance distribution.

There are many ways to choose the variance (or, more generally, the distribution) for $\Delta x^n(k)$. We present one possible way:

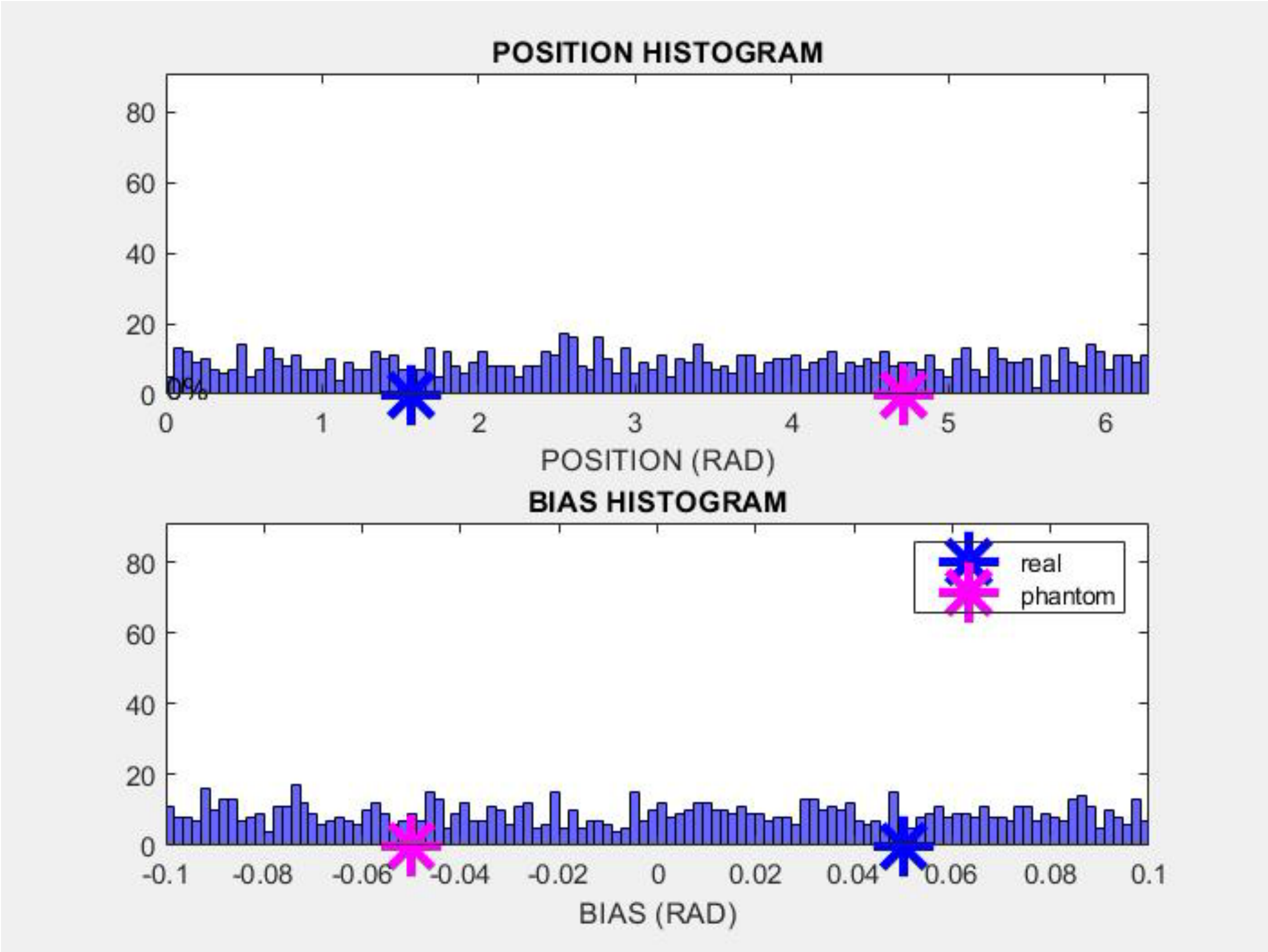
Let σ_i be the standard deviation of $\Delta x^n_i(k)$, where the index i represents the i-th element of a vector. Then, choose

$$\sigma_i = K\, E_i\, N^{-\frac{1}{d}},$$

with

K: tuning parameter, typically $K \ll 1$
d: dimension of the state space
E_i: $\max\limits_{n_1,n_2} |\bar{x}^{n_1}_{\mathsf{m},i}(k) - \bar{x}^{n_2}_{\mathsf{m},i}(k)|$, the maximum inter-sample variability
$N^{-\frac{1}{d}}$: related to the spacing between nodes of a uniform grid.

Video: With Roughening



Remarks

- The PF is an approximation of the Bayesian state estimator. The fact that the PF can, in principle, handle general nonlinear systems and general noise distributions comes at the expense of a possibly large computational effort. In particular, the large number of particles N that may be required to reliably capture the state PDF may be prohibitive for a practical implementation.
- The PF presented in this lecture is the simplest, most basic form of particle filtering. There are many practical (numerical) issues that often require tuning and exploiting the problem structure for a satisfactory filter performance.