Recap: Auxiliary variables

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

$$\begin{aligned}
&\textbf{Init:} && x_m(0) := x(0) \\
&\textbf{S1:} && x_p(k) := q_{k-1}(x_m(k{-}1), v(k{-}1)) \\
&\textbf{S2:} && z_m(k) := h_k(x_p(k), w(k)) \\
& && x_m(k) \text{ defined via its PDF} \\
& && \quad f_{x_m(k)}(\xi) := f_{x_p(k)|z_m(k)}(\xi|\mathbf{z}(k)) \quad \forall \xi
\end{aligned} \quad \Bigg\} \; k = 1, 2, \ldots$$

For all ξ and $k = 1, 2, \ldots$,

$$f_{x_p(k)}(\xi) = f_{x(k)|z(1:k-1)}(\xi|\mathbf{z}(1{:}k{-}1))$$
$$f_{x_m(k)}(\xi) = f_{x(k)|z(1:k)}(\xi|\mathbf{z}(1{:}k))\,.$$

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.

9.6 Measurement Update

Given the PDF $f_{x_p(k)}(\xi)$ of $x_p(k) \in \mathcal{X}$ (from prior update) and after obtaining a measurement $\mathbf{z}(k)$, we construct the PDF $f_{x_m(k)}(\xi)$ of $x_m(k)$. Again, we *approximate* the PDF by Monte Carlo sampling.

Scaling of particles by measurement likelihood

- From Bayes' rule, we get

$$f_{x_m(k)}(\xi) := f_{x_p(k)|z_m(k)}(\xi|\mathbf{z}(k)) = \frac{f_{z_m(k)|x_p(k)}(\mathbf{z}(k)|\xi)\;\; f_{x_p(k)}(\xi)}{\underbrace{\sum_{\xi\in\mathcal{X}} f_{z_m(k)|x_p(k)}(\mathbf{z}(k)|\xi)\;\; f_{x_p(k)}(\xi)}_{\text{normalization, constant in } \xi}}\,. \tag{9.1}$$

  The sum is replaced by an integral in the case of $x_p(k)$ being a CRV.

- We approximated $f_{x_p(k)}(\xi)$ by Monte Carlo samples (prior update); that is,

$$f_{x_p(k)}(\xi) \approx \frac{1}{N}\sum_{n=1}^{N} \delta\left(\xi - x_p^n(k)\right).$$

- Substituting the expression for $f_{x_p(k)}(\xi)$ in (**??**) suggests the following form for $f_{x_p(k)|z_m(k)}(\xi|\mathbf{z}(k))$:

$$f_{x_p(k)|z_m(k)}(\xi|\mathbf{z}(k)) \;\approx\; \sum_{n=1}^{N} \beta_n\, \delta(\xi - x_p^n(k)) \;=\; \alpha\, f_{z_m(k)|x_p(k)}(\mathbf{z}(k)|\xi) \sum_{n=1}^{N} \delta(\xi - x_p^n(k)),$$

  where α is a normalization constant.

- We get N equations by substituting $\xi = x_p^n(k)$ for $n = 1, 2, \ldots, N$ (recall that $\delta(\cdot)$ is zero except at zero):

$$\beta_n = \alpha\, f_{z_m(k)|x_p(k)}\big(\mathbf{z}(k)|x_p^n(k)\big) \qquad n = 1, 2, \ldots, N\,.$$

  We have an additional equation, $\sum_{n=1}^{N} \beta_n = 1$, which is required for $f_{x_p(k)|z_m(k)}(\xi|\mathbf{z}(k))$ to be a valid PDF. Therefore,

$$\alpha = \left(\sum_{n=1}^{N} f_{z_m(k)|x_p(k)}\big(\mathbf{z}(k)|x_p^n(k)\big)\right)^{-1}.$$

Date compiled: May 22, 2013

- Intuition: treat each particle separately. At points of high prior, we have many particles. Posterior has the same particles, but scaled by measurement likelihood.
- Note that $f_{z_m(k)|x_p(k)}(\mathbf{z}(k)|\xi) = f_{z(k)|x(k)}(\mathbf{z}(k)|\xi)$ for all ξ by the definition of $z_m(k)$.
- Summary: We obtained a representation for $f_{x_m(k)}(\xi)$,

$$f_{x_m(k)}(\xi) = f_{x_p(k)|z_m(k)}(\xi|\mathbf{z}(k)) \approx \sum_{n=1}^{N} \beta_n \, \delta(\xi - x_p^n(k)).$$

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

Resampling

Objective: Reselect N particles, where the probability of selecting particle n is β_n. This is essentially like drawing N samples from the distribution given by $\sum_{n=1}^{N} \beta_n \, \delta(\xi - x_p^n(k))$. We apply the basic "sampling a distribution" algorithms (see lecture # 2).

Algorithm. Repeat N times:

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

This gives N *new* particles $x_m^n(k)$, which are a subset of the old particles (because we only select from those). The new particles all have equal weight,

$$f_{x_m(k)}(\xi) \approx \frac{1}{N} \sum_{n=1}^{N} \delta(\xi - x_m^n(k)),$$

as required. This completes the measurement update step.

9.7 Summary

The *basic* PF algorithm is given by:

Initialization: Draw N samples $\{x_m^n(0)\}$ from $f(x(0))$. These are the initial particles.

Step 1 (S1): Prior update/Prediction step

To obtain the prior particles $\{x_p^n(k)\}$, simulate the particles $\{x_m^n(k\text{–}1)\}$ via the process equation:

$$x_p^n(k) := q_{k-1}(x_m^n(k\text{–}1), v^n(k\text{–}1)), \qquad \text{for } n = 1, 2, \ldots, N,$$

which will require N noise samples from $f(v(k\text{–}1))$.

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

Scale each particle by the measurement likelihood:

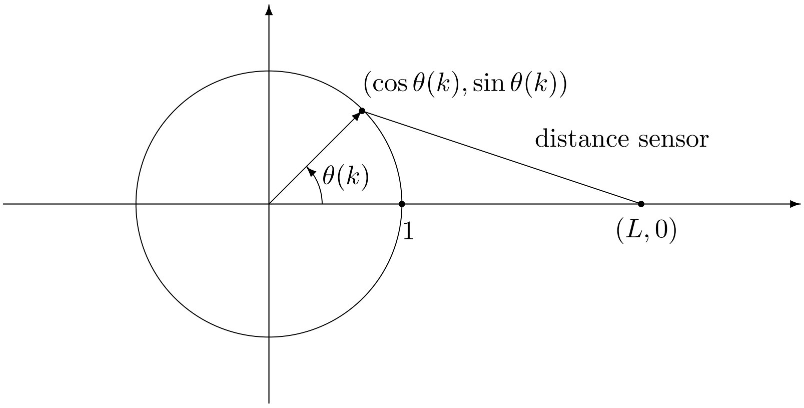
$$\beta_n = \alpha \, f_{z(k)|x(k)}\big(\mathbf{z}(k)|x_p^n(k)\big), \qquad \text{for } n = 1, 2, \ldots, N,$$

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

Resample (using the algorithm above) to get the N posterior particles $\{x_m^n(k)\}$, all with equal weights.

9.8 Example (continued)

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



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

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

  where $x_1(0)$ and $v(k\text{–}1)$ 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.

- 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, the PF will "select" one of them.
  - 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).

9.9 Sample Impoverishment

Problem: All particles converge to the same one and they are therefore no longer a good representation of the state PDF. This has to do with the lumpiness of the approximation. In the limit as $N \to \infty$, this takes an infinite amount of time to happen. It turns out to be an issue for a finite N.

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

Roughening

Perturb the particles *after* resampling,

$$x_m^n(k) \leftarrow x_m^n(k) + \Delta x^n(k),$$

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 in the following.
- Let σ_i be the standard deviation of $\Delta x_i^n(k)$, 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: $\max\limits_{n_1, n_2} |x_{m,i}^{n_1}(k) - x_{m,i}^{n_2}(k)|$, the maximum inter-sample variability

  $N^{-\frac{1}{d}}$: spacing between nodes of the corresponding uniform, rectangular grid.

9.10 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, a 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 utilizing the problem structure for a satisfactory filter performance.