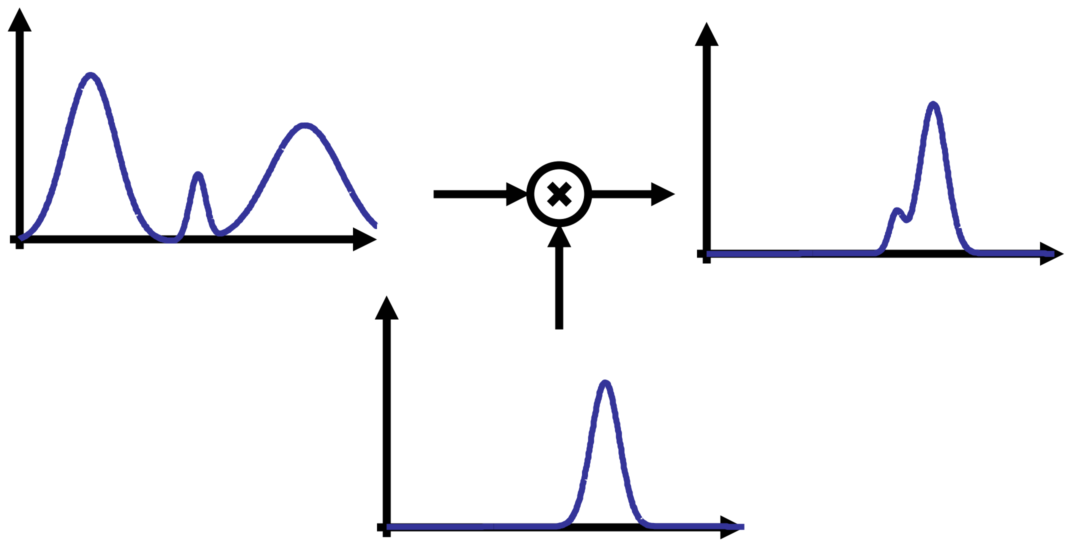
Stochastic Filtering

A brief tutorial

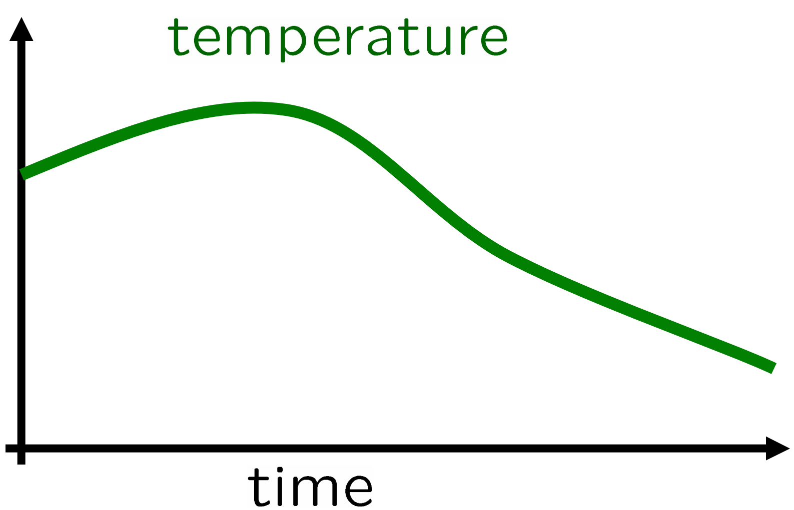


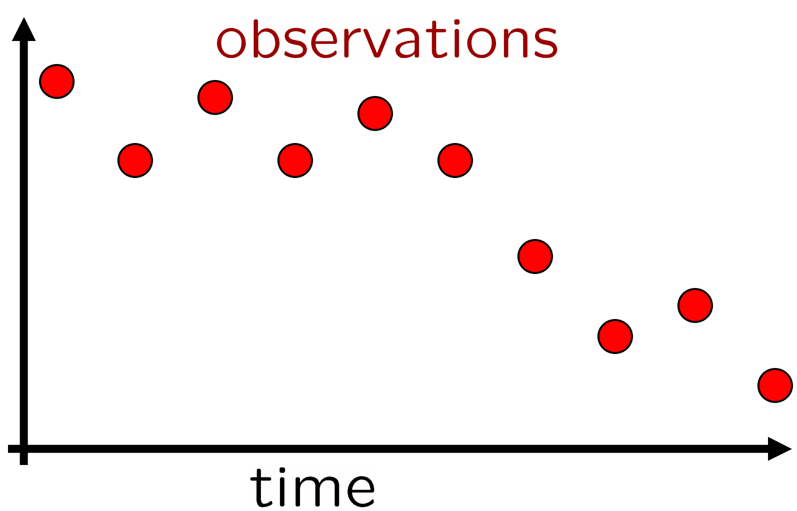
Rui Castro

Motivation

In many applications one wants to track time-varying processes

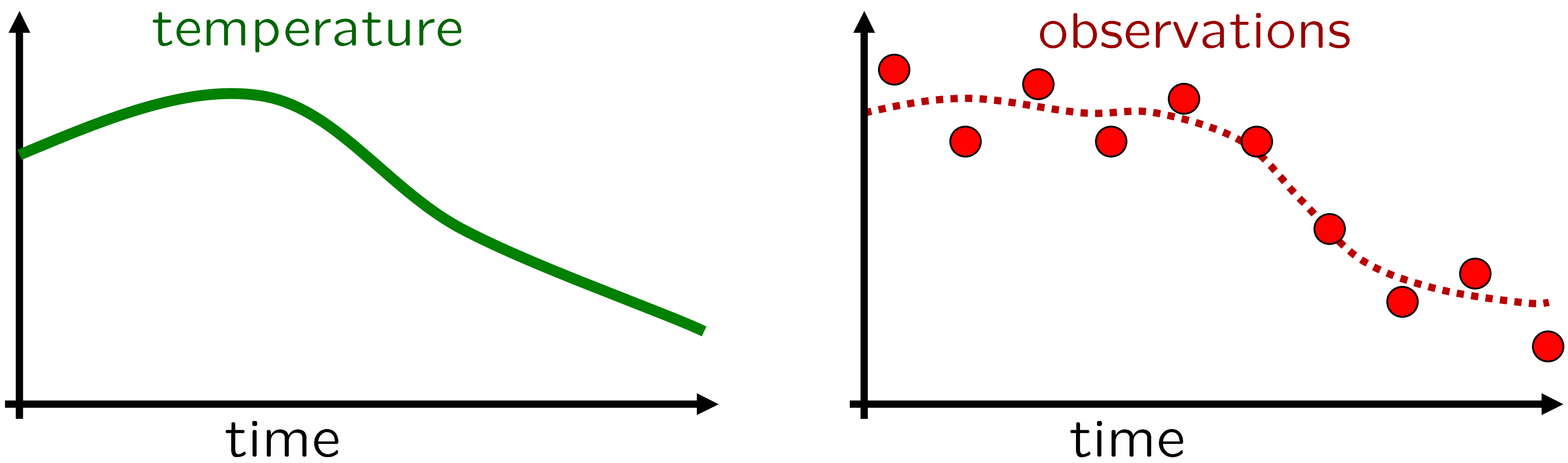
Example: tracking temperature in a museum room with a inaccurate thermometer with large error.





How can we accurately estimate the room temperature???

Motivation



Some kind of windowed estimate should do the trick... (low pass filtering)

Key fact: Knowledge of the process dynamics (*i.e.*, temperature changes sloooooowly...

Modeling - Dynamics

Let $\{\boldsymbol{X}_n\}_{n\geq 1}$ denote a random process characterized by the p.d.f.

$$\begin{aligned} f_{\boldsymbol{X}_n|\boldsymbol{X}_{n-1},\ldots,\boldsymbol{X}_1}(\boldsymbol{x}_n|\boldsymbol{x}_{n-1},\ldots,\boldsymbol{x}_1) &= f_{\boldsymbol{X}_n|\boldsymbol{X}_{n-1}}(\boldsymbol{x}_n|\boldsymbol{x}_{n-1}) \\ &\triangleq S_n(\boldsymbol{x}_n|\boldsymbol{x}_{n-1}) \end{aligned}$$

Dynamics Kernel

This random process is the quantity we are interested in estimating (*e.g.*, temperature of the room)

The distribution of $\boldsymbol{X}_n$ depends only on the value of $\boldsymbol{X}_{n-1}$. Together with the knowledge of $f_{\boldsymbol{X}_1}$ this completely defines the random process

Dynamics - Example

Position-Velocity (PV) Model:

$$\begin{aligned} \frac{\partial}{\partial t} x(t) &= v(t) \\ \frac{\partial}{\partial t} v(t) &= u(t) \ , \end{aligned} \qquad u(t) - \text{Gaussian White noise process}$$

→ Velocity is modeled as Brownian motion

Discrete counterpart:

$$\begin{bmatrix} \boldsymbol{x}_{n+1} \\ \boldsymbol{v}_{n+1} \end{bmatrix} = \begin{bmatrix} 1 & \Delta \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \boldsymbol{x}_{n} \\ \boldsymbol{v}_{n} \end{bmatrix} + \begin{bmatrix} 0 \\ \sigma^2 \end{bmatrix} u(t) \ ,$$

where $u(t) \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0,1)$, and Δ is small

Observation Model

In general we cannot observe the process $\{\boldsymbol{X}_n\}$ directly

$$\begin{aligned} f(\boldsymbol{z}_n|\boldsymbol{z}_{n-1},\ldots,\boldsymbol{z}_1,\boldsymbol{x}_n\ldots,\boldsymbol{x}_1) &= f(\boldsymbol{z}_n|\boldsymbol{x}_n) \\ &\triangleq H_n(\boldsymbol{z}_n|\boldsymbol{x}_n) \end{aligned}$$

Observation Likelihood

Model Assumptions:

- Observation at time *n* depends only on state at that time
- Observations do not affect the physical process

Main Goal

Given a set of observations up to time *n*,

$$\overline{\mathbf{Z}}_n = \{z_1, \ldots, z_n\}$$

we want to estimate $\boldsymbol{X}_n$.

We follow a Bayesian approach, therefore all the information about $\boldsymbol{X}_n$ is provided by the density

$$f(x_n | \overline{\mathbf{Z}}_n) \triangleq F_n(x_n)$$

Filtering Density

How do we do it???

Density Propagation – First Observation

Can we estimate X_1 ?

$$f(x_1) = f(x_1|\overline{Z}_0) \triangleq P_1(x_1)$$

Prior Density

After we observe z_1 we can use Bayes rule

$$\begin{aligned} F_1(x_1) = f(x_1|\overline{Z}_1) &= \frac{f(z_1|x_1)f(x_1)}{f(z_1)} \\ &= c_1 H(z_1|x_1) P_1(x_1), \end{aligned}$$

where c_1 is a normalization constant dependent on z_1.

Density Propagation – Prediction

Can we predict X_2 ?

$$
\begin{aligned}
f(x_1, x_2 | \overline{\mathbf{Z}}_1) &= \frac{f(x_1, x_2, z_1)}{f(z_1)} \\
&= \frac{f(x_2 | x_1, z_1) f(x_1, z_1)}{f(z_1)} \\
&= \frac{f(x_2 | x_1) f(x_1 | z_1) f(z_1)}{f(z_1)} \\
&= S(x_2 | x_1) F_1(x_1)
\end{aligned}
$$

Density Propagation – Prediction

Our prediction is then given by

$$
\begin{aligned}
f(\boldsymbol{x}_2|\overline{\boldsymbol{Z}}_1) &= \int f(\boldsymbol{x}_1, \boldsymbol{x}_2|\overline{\boldsymbol{Z}}_1)\mathrm{d}\boldsymbol{x}_1 \\
&= \int S_n(\boldsymbol{x}_2|\boldsymbol{x}_1)F_1(\boldsymbol{x}_1)\mathrm{d}\boldsymbol{x}_1 \\
&= (S_n * F_1)(\boldsymbol{x}_2) \\
&\triangleq P_2(\boldsymbol{x}_2)
\end{aligned}
$$

Prediction Density

Density Propagation – General

Start with a prediction based on previous observations

$$P_n(\boldsymbol{x}_n) \triangleq f(\boldsymbol{x}_n|\overline{\boldsymbol{Z}}_{n-1})$$

Prediction Density

After we observe z_n we can compute

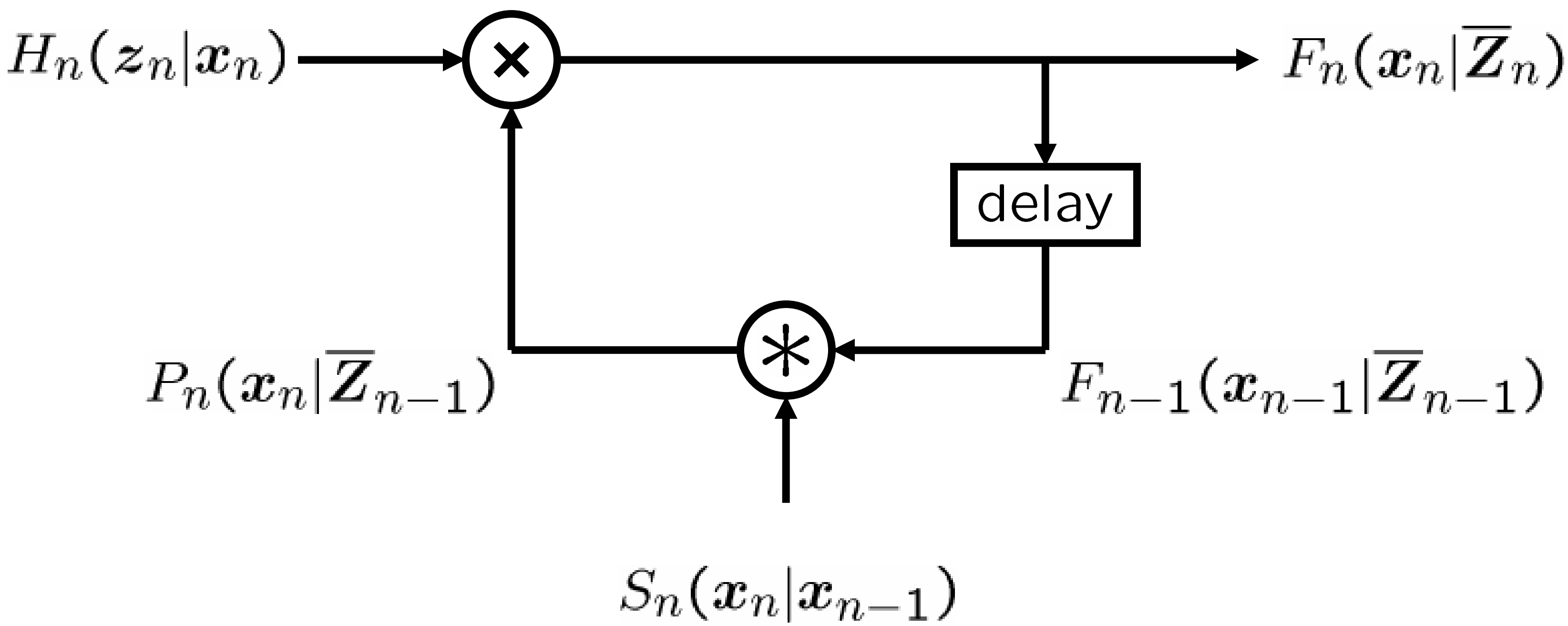
$$\begin{aligned} F_n(\boldsymbol{x}_n) = f(\boldsymbol{x}_n|\overline{\boldsymbol{Z}}_n) &= \frac{f(\boldsymbol{z}_n|\boldsymbol{x}_n) f(\boldsymbol{x}_n|\overline{\boldsymbol{Z}}_{n-1}) f(\overline{\boldsymbol{Z}}_{n-1})}{f(\overline{\boldsymbol{Z}}_n)} \\ &\propto H(\boldsymbol{z}_n|\boldsymbol{x}_n) P_n(\boldsymbol{x}_n), \end{aligned}$$

Density Propagation – General

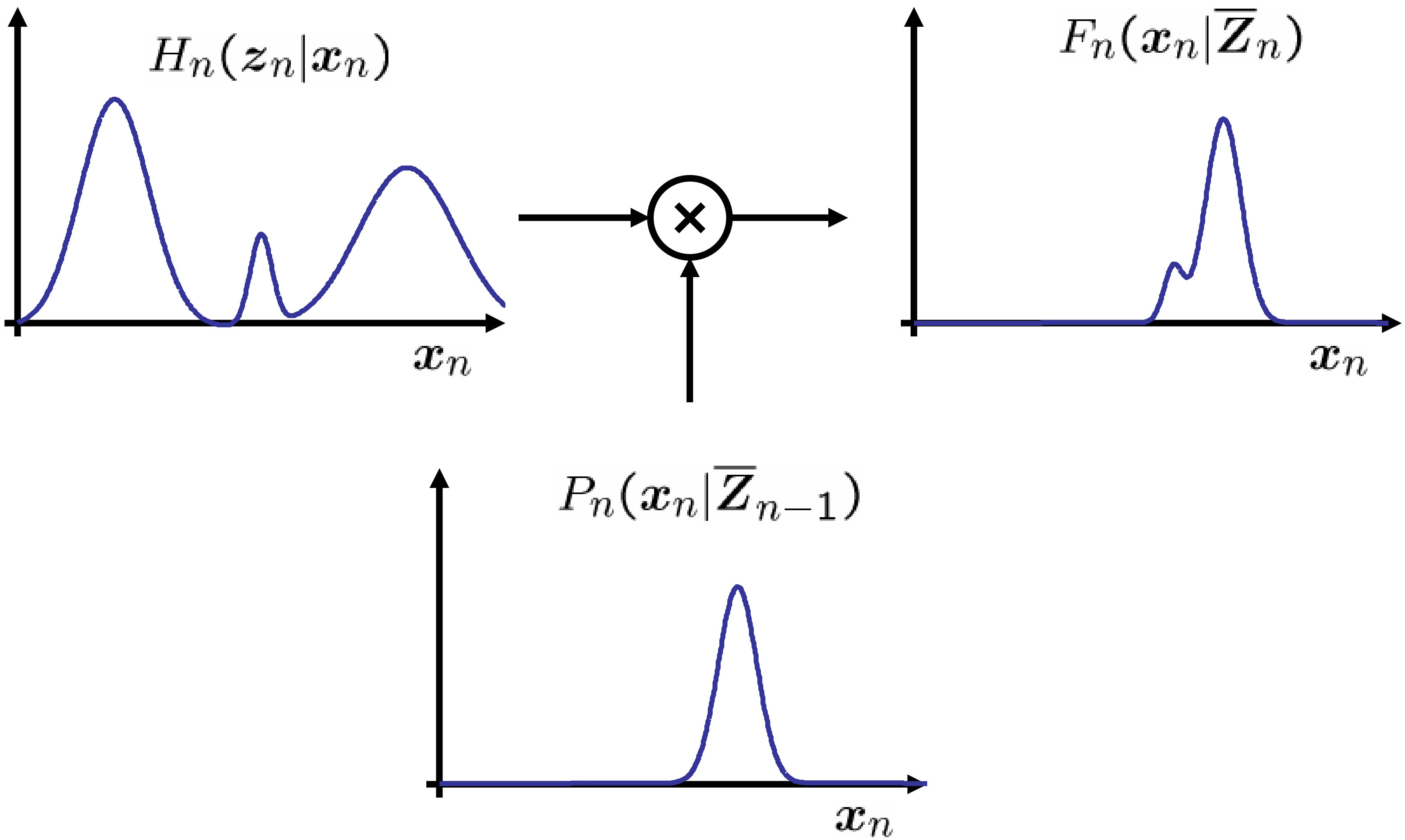
We can now predict $\boldsymbol{X}_{n+1}$

$$
\begin{aligned}
P_{n+1}(\boldsymbol{x}_{n+1}) &= f(\boldsymbol{x}_{n+1}|\overline{\boldsymbol{Z}}_n) \\
&= \int f(\boldsymbol{x}_{n+1}, \boldsymbol{x}_n|\overline{\boldsymbol{Z}}_n) \mathrm{d}\boldsymbol{x}_n \\
&= \int \frac{f(\boldsymbol{x}_{n+1}, \boldsymbol{x}_n, \overline{\boldsymbol{Z}}_n)}{f(\overline{\boldsymbol{Z}}_n)} \mathrm{d}\boldsymbol{x}_n \\
&= \int \frac{f(\boldsymbol{x}_{n+1}|\boldsymbol{x}_n) f(\boldsymbol{x}_n|\overline{\boldsymbol{Z}}_n) f(\overline{\boldsymbol{Z}}_n)}{f(\overline{\boldsymbol{Z}}_n)} \mathrm{d}\boldsymbol{x}_n \\
&= \int S_{n+1}(\boldsymbol{x}_{n+1}|\boldsymbol{x}_n) F_n(\boldsymbol{x}_n) \mathrm{d}\boldsymbol{x}_n
\end{aligned}
$$

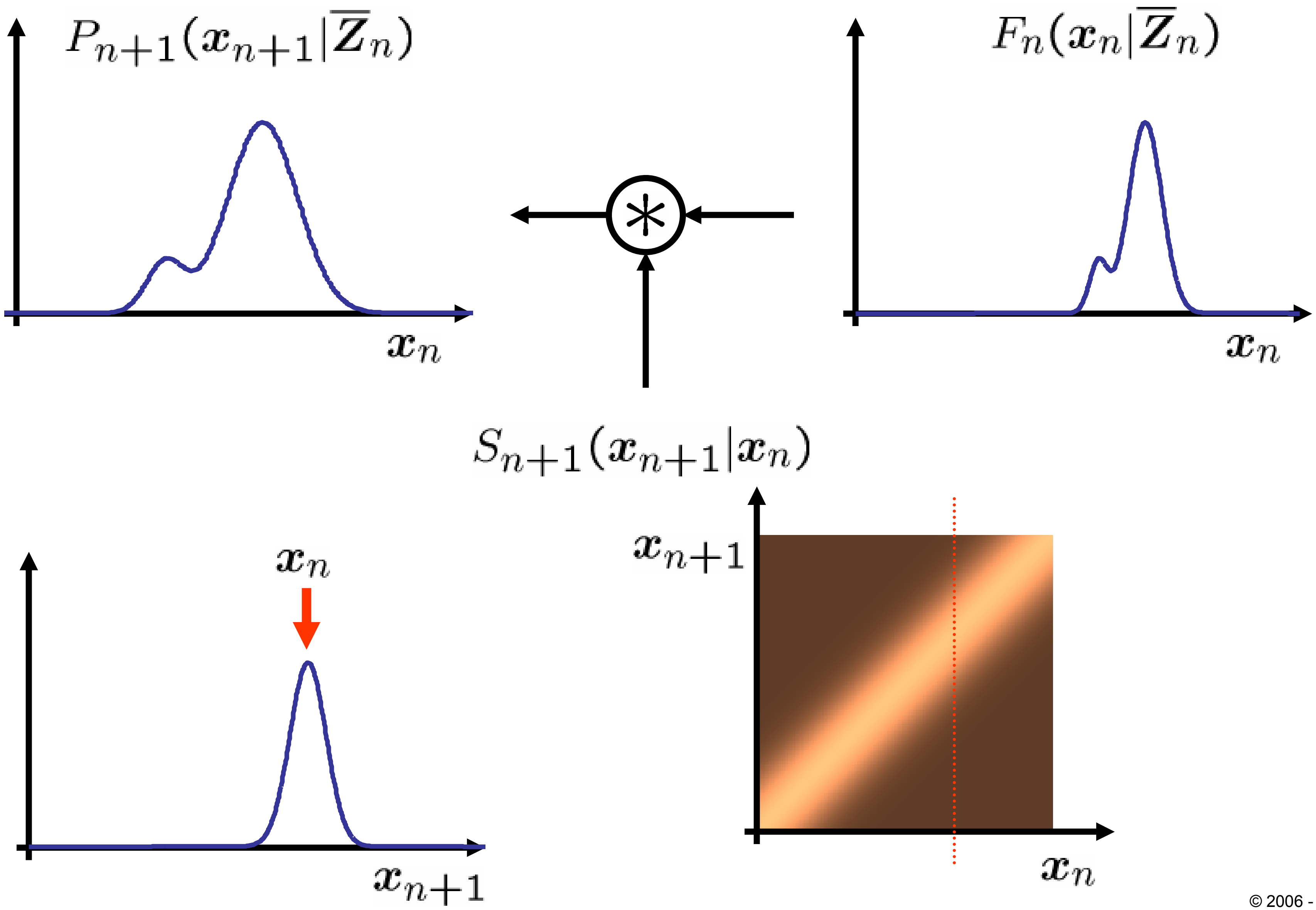
Density Propagation – Graphically



Density Propagation – Filtering



Density Propagation – Prediction



Main Goal

Now that we have

$$f(\boldsymbol{x}_n|\overline{\boldsymbol{Z}}_n) = F_n(\boldsymbol{x}_n)$$

what do we do with it? That is, how do we construct an estimate $\widehat{\boldsymbol{X}}_n$

Many possibilities, for example if we want $\widehat{\boldsymbol{X}}_n$ to minimize

$$\mathbb{E}\left[(\widehat{\boldsymbol{X}}_n - \boldsymbol{X}_n)^2|\overline{\boldsymbol{Z}}_n\right]$$

the optimal choice is

$$\widehat{\boldsymbol{X}}_n = \mathbb{E}\left[\boldsymbol{X}_n|\overline{\boldsymbol{Z}}_n\right] = \int \boldsymbol{x}_n f(\boldsymbol{x}_n|\overline{\boldsymbol{Z}}_n)\mathrm{d}\boldsymbol{x}$$

A Particular Case: **The Kalman Filter**

$$\boldsymbol{X}_i \in \mathbb{R}^d, \qquad \boldsymbol{X}_n = \underbrace{A}_{d \times d} \cdot \boldsymbol{X}_{n-1} + \underbrace{B}_{d \times k} \cdot \boldsymbol{U}_n \;, \qquad \boldsymbol{U}_n \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, I_k)$$

$$\boldsymbol{Z}_i \in \mathbb{R}^p, \qquad \boldsymbol{Z}_n = \underbrace{C}_{p \times d} \cdot \boldsymbol{X}_n + \underbrace{D}_{p \times l} \cdot \boldsymbol{V}_n \;, \qquad \boldsymbol{V}_n \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, I_l)$$

And, **very important**

$$\begin{aligned} f(\boldsymbol{X}_1) &= P_1(\boldsymbol{X}_1) \\ &\propto \exp\left(-\frac{1}{2}(\boldsymbol{x}_1 - \eta_{P_1})^T V_{P_1}^{-1}(\boldsymbol{x}_1 - \eta_{P_1})\right) \end{aligned}$$

In other words, we **assume** that

$$X_1 \sim \mathcal{N}(\eta_{P_1}, V_{P_1})$$

A Particular Case: **The Kalman Filter**

Since

$$P_1(\boldsymbol{x}_1), \quad S_n(\boldsymbol{x}_n|\boldsymbol{x}_{n-1}), \text{ and } \quad H_n(\boldsymbol{z}_n|\boldsymbol{x}_n)$$

are all Gaussian it follows that

$$F_n(\boldsymbol{x}_n) \text{ and } P_n(\boldsymbol{x}_n)$$

are also Gaussian (because the product of two Gaussians is a Gaussian and the convolution of two Gaussians is also a Gaussian).

Therefore we need only to propagate means and covariance matrices (a nice closed form solution)

A Particular Case: **The Kalman Filter**

$P_n(x_n)$ parameterized by η_{P_n}, V_{P_n}

$F_n(x_n)$ parameterized by η_{F_n}, V_{F_n}

- Start with the prediction density $P_n(x_n)$
- After we observe z_n compute $F_n(x_n)$

$$\eta_{F_n} = \eta_{P_n} + V_{P_n} C^T \left(C V_{P_n} C^T + D^T D \right)^{-1} \left(z_n - C \eta_{P_n} \right)$$
$$V_{F_n} = \left(I - V_{P_n} C^T \left(C V_{P_n} C^T + D^T D \right)^{-1} C \right) V_{P_n}$$

- Compute the updated prediction $P_{n+1}(x_{n+1})$

$$\eta_{P_{n+1}} = A \eta_{F_n}$$
$$V_{P_{n+1}} = A V_{F_n} A^T + B^T B$$

A Particular Case: **The Kalman Filter**

Taking $\widehat{\boldsymbol{X}}_n = \eta_{F_n}$

yields the Kalman Filter

$$\begin{aligned} \widehat{\boldsymbol{X}}_n &= \mathbb{E}\left[\boldsymbol{X}_n | \overline{\boldsymbol{Z}}_n\right] \\ &= \arg\min_{\boldsymbol{y}} \mathbb{E}\left[(\boldsymbol{y} - \boldsymbol{X}_n)^2 | \overline{\boldsymbol{Z}}_n\right] \end{aligned}$$

Numerous applications, but restricted to linear models

What if models are non-linear???

The Extended Kalman Filter

$$\boldsymbol{X}_i \in \mathbb{R}^d, \qquad \boldsymbol{X}_n = \Phi(\boldsymbol{X}_{n-1}) + B \cdot \boldsymbol{U}_n \;, \qquad \boldsymbol{U}_n \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, I_k)$$

$$\boldsymbol{Z}_i \in \mathbb{R}^p, \qquad \boldsymbol{Z}_n = \Psi(\boldsymbol{X}_n) + D \cdot \boldsymbol{V}_n \;, \qquad \boldsymbol{V}_n \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, I_l)$$

Suppose $P_n(\boldsymbol{x}_n)$ is normal $\mathcal{N}(\eta_{P_n}, V_{P_n})$

If our prediction density is "accurate" then

$$\boldsymbol{X}_n \approx \eta_{P_n}$$

Idea: Linearize around the prediction mean

$$\Psi(\boldsymbol{X}_n) \approx \Psi(\eta_{P_n}) + \mathbb{D}_\Psi(\eta_{P_n})(\boldsymbol{X}_n - \eta_{P_n})$$

$$\underbrace{\boldsymbol{Z}_n - \Psi(\eta_{P_n}) + \mathbb{D}_\Psi(\eta_{P_n})\eta_{P_n}}_{\boldsymbol{Z'}_n} \approx \underbrace{\mathbb{D}_\Psi(\eta_{P_n})}_{C_n}\boldsymbol{X}_n + D\boldsymbol{V}_n$$

The Extended Kalman Filter

$$\eta_{F_n} = \eta_{P_n} + V_{P_n} C_n^T \left(C_n V_{P_n} C_n^T + D^T D\right)^{-1} \left(\boldsymbol{z}_n - \Psi(\eta_{P_n})\right)$$

$$V_{F_n} = \left(I - V_{P_n} C_n^T \left(C_n V_{P_n} C_n^T + D^T D\right)^{-1} C_n\right) V_{P_n}$$

Similarly, if $\eta_{F_n} \approx \boldsymbol{x}_n$

$$\Phi(\boldsymbol{x}_n) \approx \Phi(\eta_{F_n}) + \mathbb{D}_\Phi(\eta_{F_n})(\boldsymbol{x}_n - \eta_{F_n})$$

$$\boldsymbol{X}_{n+1} - \Phi(\eta_{F_n}) + \mathbb{D}_\Phi(\eta_{F_n})\eta_{F_n} \approx \underbrace{\mathbb{D}_\Phi(\eta_{P_n})}_{A_n} \boldsymbol{X}_n + B\boldsymbol{V}_n$$

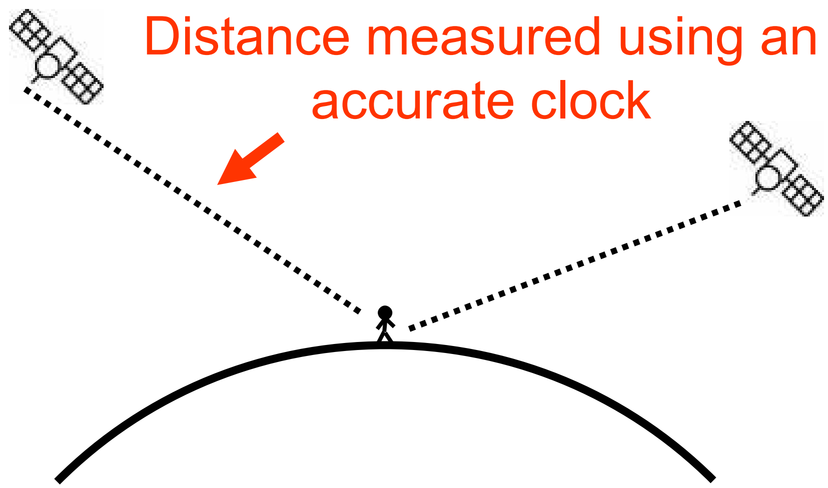
$$\eta_{P_{n+1}} = \Phi(\eta_{F_n})$$

$$V_{P_{n+1}} = A_n V_{F_n} A_n^T + B^T B$$

The Extended Kalman Filter

This procedure is widely used in practice, for example in the

Global Positioning System - GPS

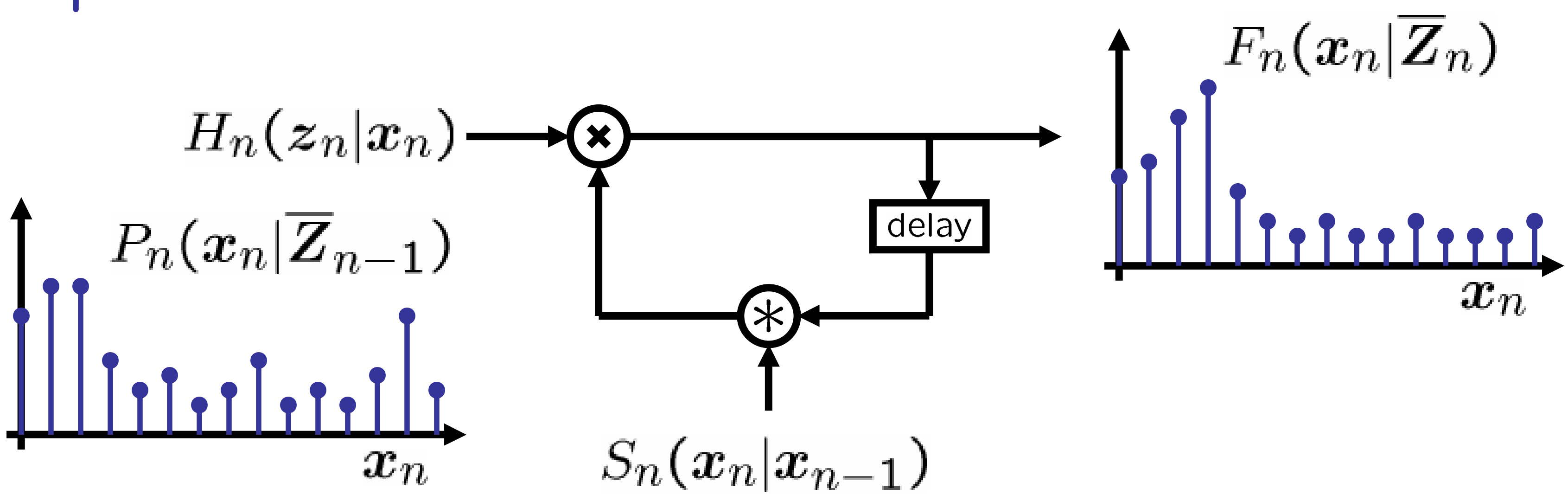


The Extended Kalman filter works is you have an accurate clock and know where you are... If not the linearization is poor and there's not much you can do

Other Non-Linear Model Procedures

- Approximation of non-linearities by Gaussian Mixture models

- Evaluate the densities only at a finite (fixed) set of points – the **Point-Mass Filter**

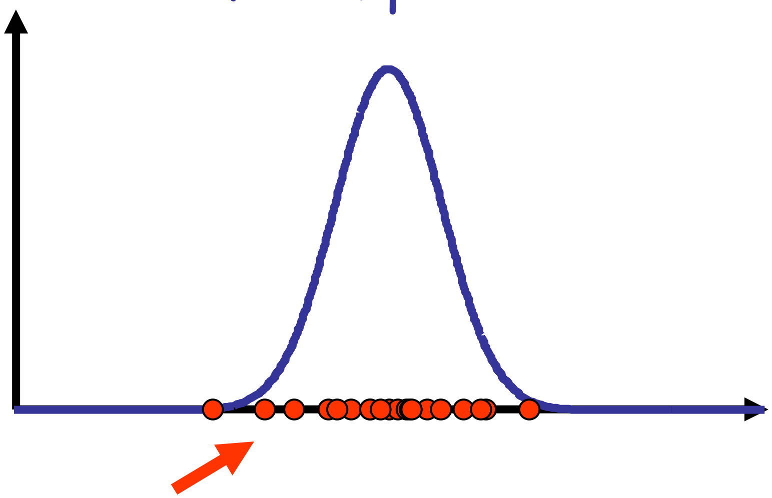


Effective only in small state spaces!!!

Particle Filters

Same idea as in the point-mass filter, but dynamically allocate the location of the point masses.

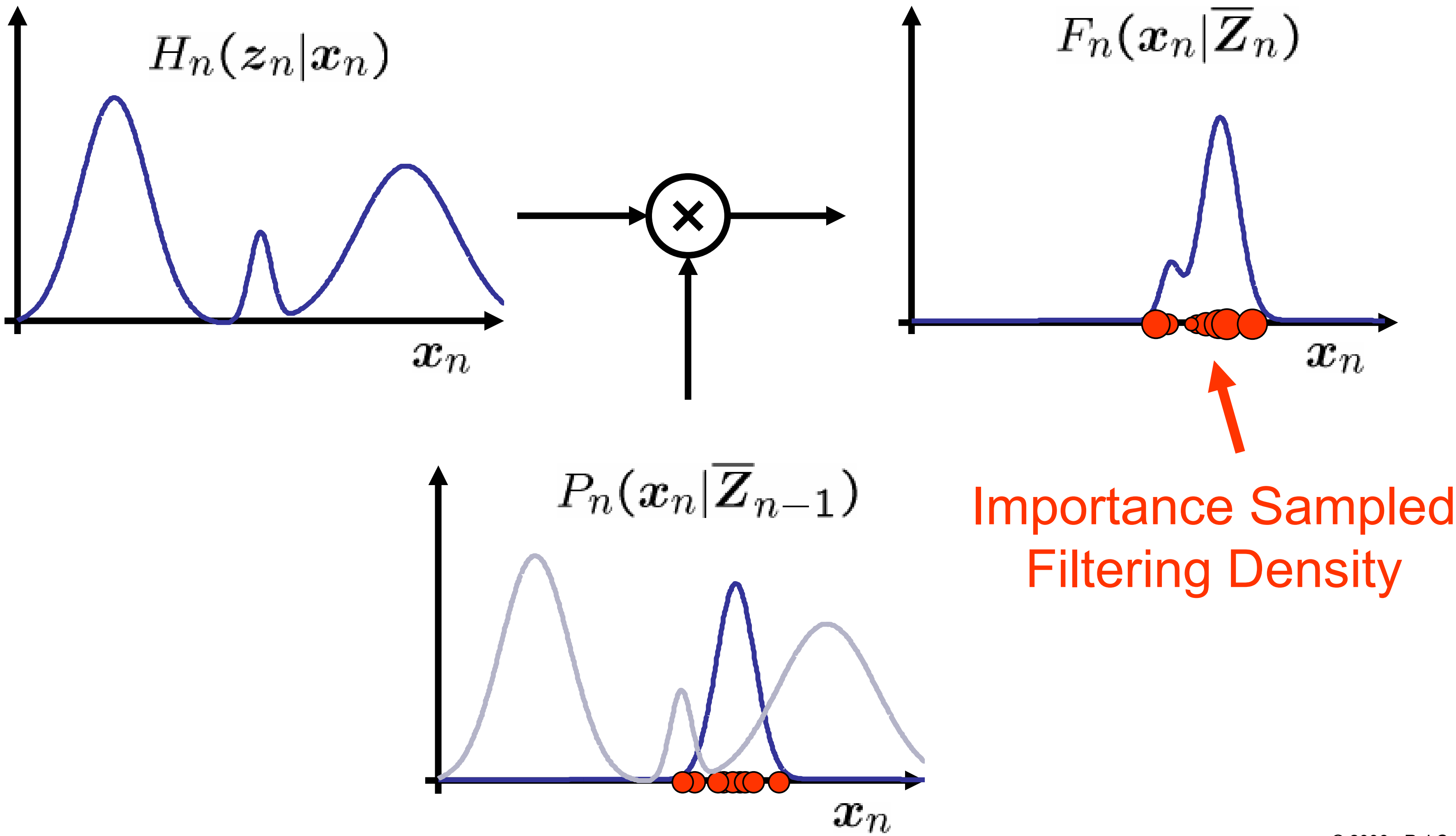
In other words, describe the densities in a empirical way, though a set of samples:



The samples can be used to compute almost anything interesting about the density

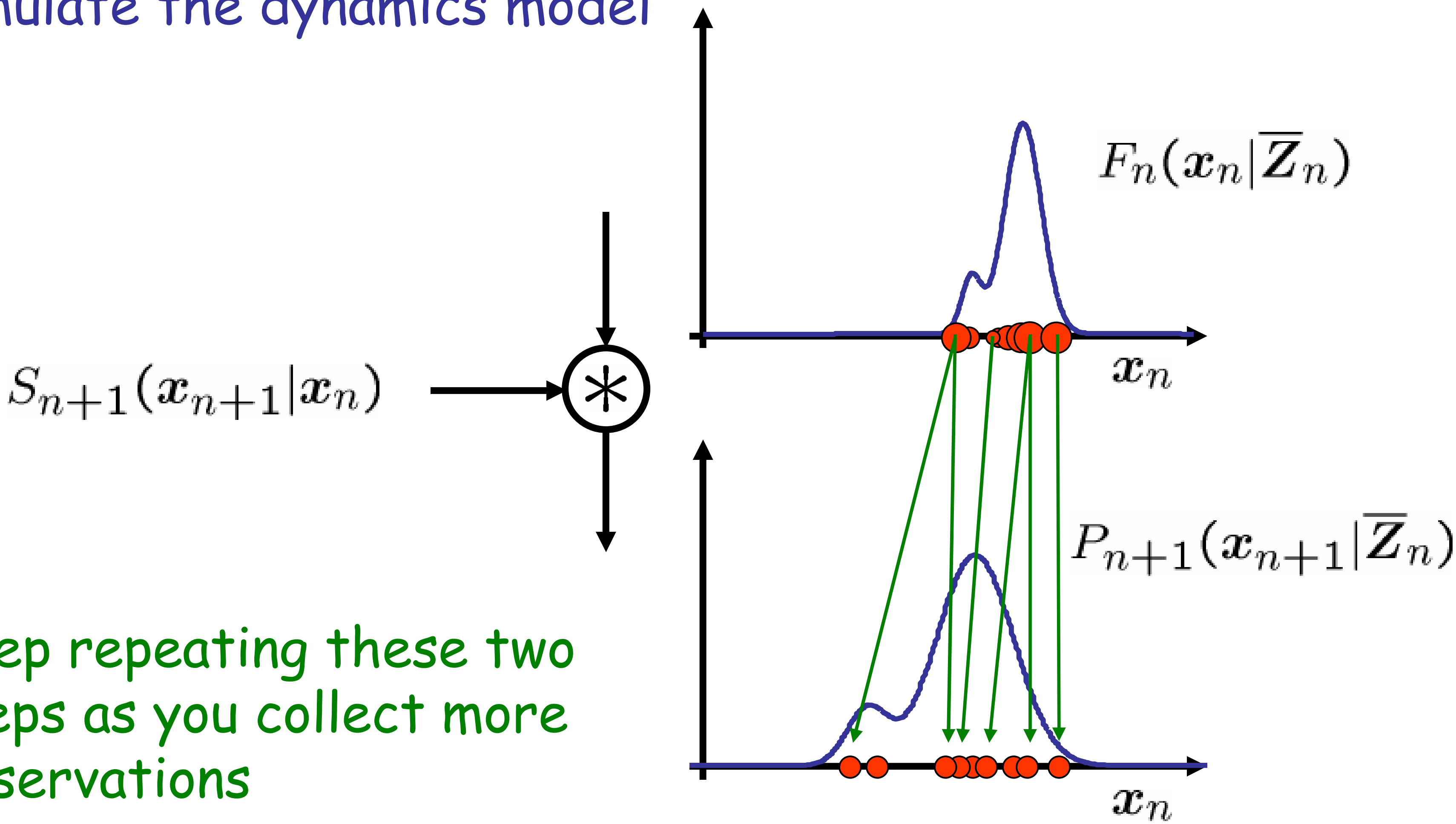
Condensation Algorithm (Blake *et al*)

Start with samples from $P_n(\boldsymbol{x}_n)$



Condensation Algorithm

Sample according to these importance samples and simulate the dynamics model



Keep repeating these two steps as you collect more observations

Condensation Algorithm

This procedure is asymptotically optimal as the number of particles grows

Many "tricks of the trade" can be used to improve performance and reduce computational cost, namely:

- Choose the number of particles dynamically
- Augment/reduce the size of the dynamical model to allow multi-target tracking

➡ **Key Fact:** For computationally feasible procedures the model of the dynamics must be easily simulated.

Recently, non-asymptotic guarantees for these procedures became available.

A Tracking Example

Tracking an orange (clementine), in clutter (including other oranges). Movie taken with a handheld camera (no camera motion registration).

A Tracking Example

A particle filter tracking the left-most orange, using a PV dynamics model and 1000 particles.

Final Remarks

- Stochastic Filtering is a very general (Bayesian) framework for sequential estimation in a model-based setting.

- For linear and Gaussian models the densities being propagated have a closed-form solution and the result is simply the well known Kalman filter.

- When using non-linear models closed-form solutions are generally not possible, and so approximate methods need to be used.