732A96/TDDE15 Advanced Machine Learning

State Space Models

Jose M. Peña
IDA, Linköping University, Sweden

Lecture 8: Extended, Unscented, and Particle Kalman Filters

Contents

Literature

- Main source
  - Thrun, S. et al. *Probabilistic Robotics*. MIT Press, 2005. Chapters 3.3, 3.4 and 4.3.
- Additional source
  - Bishop, C. M. *Pattern Recognition and Machine Learning*. Springer, 2006. Chapter 13.3.

Nonlinear Gaussian State Space Models

- Previous assumption:

$$
\begin{aligned}
x_t &= A_t x_{t-1} + B_t u_t + \epsilon_t \\
z_t &= C_t x_t + \delta_t \\
x_0 &= \mu_0 + \tau_0
\end{aligned}
$$

  where

$$
\begin{aligned}
\epsilon_t &\sim \mathcal{N}(\epsilon_t | 0, R_t) \\
\delta_t &\sim \mathcal{N}(\delta_t | 0, Q_t) \\
\tau_0 &\sim \mathcal{N}(\tau_0 | 0, \Sigma_0)
\end{aligned}
$$

- Current assumption:

$$
\begin{aligned}
x_t &= g(u_t, x_{t-1}) + \epsilon_t \\
z_t &= h(x_t) + \delta_t \\
x_0 &= \mu_0 + \tau_0
\end{aligned}
$$

- In other words, generalization by replacing linear with nonlinear functions.
- Consequence: No closed forms for prediction and correction steps in KF.
- Solution: Linearization of g and h via Taylor expansion.

Nonlinear Gaussian State Space Models

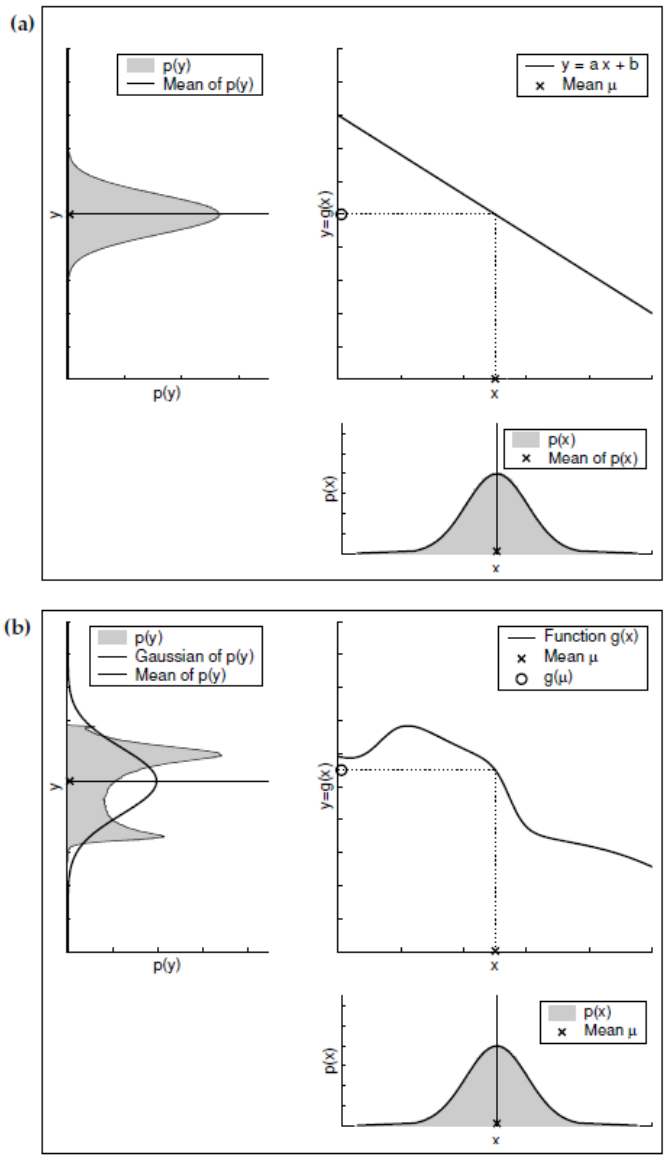


Figure 3.3 (a) Linear and (b) nonlinear transformation of a Gaussian random variable. The lower right plots show the density of the original random variable, X. This random variable is passed through the function displayed in the upper right graphs (the transformation of the mean is indicated by the dotted line). The density of the resulting random variable Y is plotted in the upper left graphs.

Note the similarities between this figure and the prediction step of KF (only the additive noise is missing).

Extended Kalman Filter

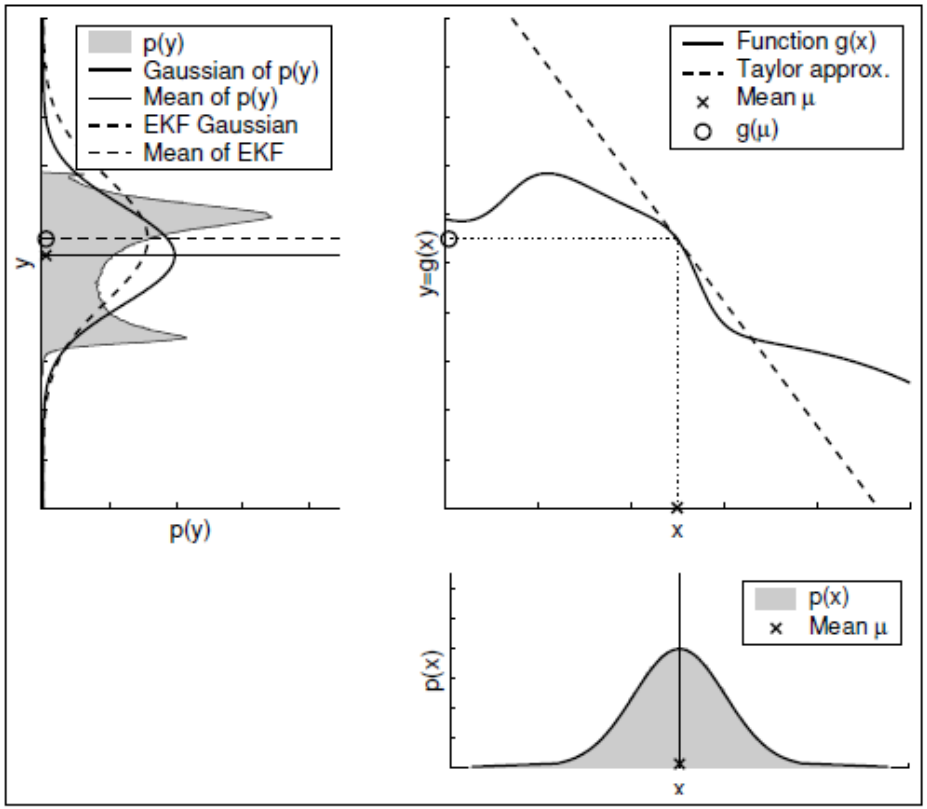


Figure 3.4 Illustration of linearization applied by the EKF. Instead of passing the Gaussian through the nonlinear function g, it is passed through a linear approximation of g. The linear function is tangent to g at the mean of the original Gaussian. The resulting Gaussian is shown as the dashed line in the upper left graph. The linearization incurs an approximation error, as indicated by the mismatch between the linearized Gaussian (dashed) and the Gaussian computed from the highly accurate Monte-Carlo estimate (solid).

Extended Kalman Filter

- As linearization method, we use the first order Taylor expansion, i.e. linear approximation of g from its value and slope at μ_{t-1}, which is the most likely state according to $bel(x_{t-1})$. That is

$$g(u_t, x_{t-1}) \approx g(u_t, \mu_{t-1}) + G_t \cdot (x_{t-1} - \mu_{t-1})$$

where $G_t = g'(u_t, \mu_{t-1})$ and $g'(u_t, x_{t-1}) = \frac{\partial g(u_t, x_{t-1})}{\partial x_{t-1}}$.

$$x_t = g(u_t, x_{t-1}) + \epsilon_t \approx g(u_t, \mu_{t-1}) + G_t \cdot (x_{t-1} - \mu_{t-1}) + \epsilon_t$$

or in other words

$$p(x_t|u_t, x_{t-1}) = \mathcal{N}(x_t|g(u_t, \mu_{t-1}) + G_t \cdot (x_{t-1} - \mu_{t-1}), R_t)$$

- We do the same for h but around $\overline{\mu}_t$, which is the most likely state according to $\overline{bel}(x_t)$. That is

$$h(x_t) \approx h(\overline{\mu}_t) + H_t \cdot (x_t - \overline{\mu}_t)$$

where $H_t = h'(\overline{\mu}_t)$ and $h'(x_t) = \frac{\partial h(x_t)}{\partial x_t}$.

$$z_t = h(x_t) + \delta_t \approx h(\overline{\mu}_t) + H_t \cdot (x_t - \overline{\mu}_t) + \delta_t$$

or in other words

$$p(z_t|x_t) = \mathcal{N}(z_t|h(\overline{\mu}_t) + H_t \cdot (x_t - \overline{\mu}_t), Q_t)$$

- Note that G_t and H_t are matrices (since x_{t-1} may be a vector) whose values depend on u_t and μ_{t-1} (i.e., they may differ for different time points).

Extended Kalman Filter

- $\overline{bel}(x_t) = \mathcal{N}(x_t|\overline{\mu}_t, \overline{\Sigma}_t)$
- $bel(x_t) = \mathcal{N}(x_t|\mu_t, \Sigma_t)$

Extended Kalman filter algorithm

---

1 Set μ_0 and Σ_0 from $p(x_0)$
2 For $t = 1, \ldots$
3 $\quad \overline{\mu}_t = g(u_t, \mu_{t-1})$ // Prediction
4 $\quad \overline{\Sigma}_t = G_t \Sigma_{t-1} G_t^T + R_t$ // Prediction
5 $\quad \overline{K}_t = \overline{\Sigma}_t H_t^T (H_t \overline{\Sigma}_t H_t^T + Q_t)^{-1}$ // Kalman gain
6 $\quad \mu_t = \overline{\mu}_t + K_t(z_t - h(\overline{\mu}_t))$ // Correction
7 $\quad \Sigma_t = (I - K_t H_t)\overline{\Sigma}_t$ // Correction

---

- Differences with KF:
  - $g(\cdot)$ and $h(\cdot)$ replace the linear models for state and measurement predictions.
  - G_t replaces A_t and B_t, and H_t replaces C_t.
- Similarities with KF:
  - Prediction, information gain, and correction steps.
  - $(z_t - h(\overline{\mu}_t))$ in line 6 is the deviation between the actual measurement and the predicted measurement.
  - The Kalman gain K_t specifies the impact of the deviation in the correction.

Extended Kalman Filter

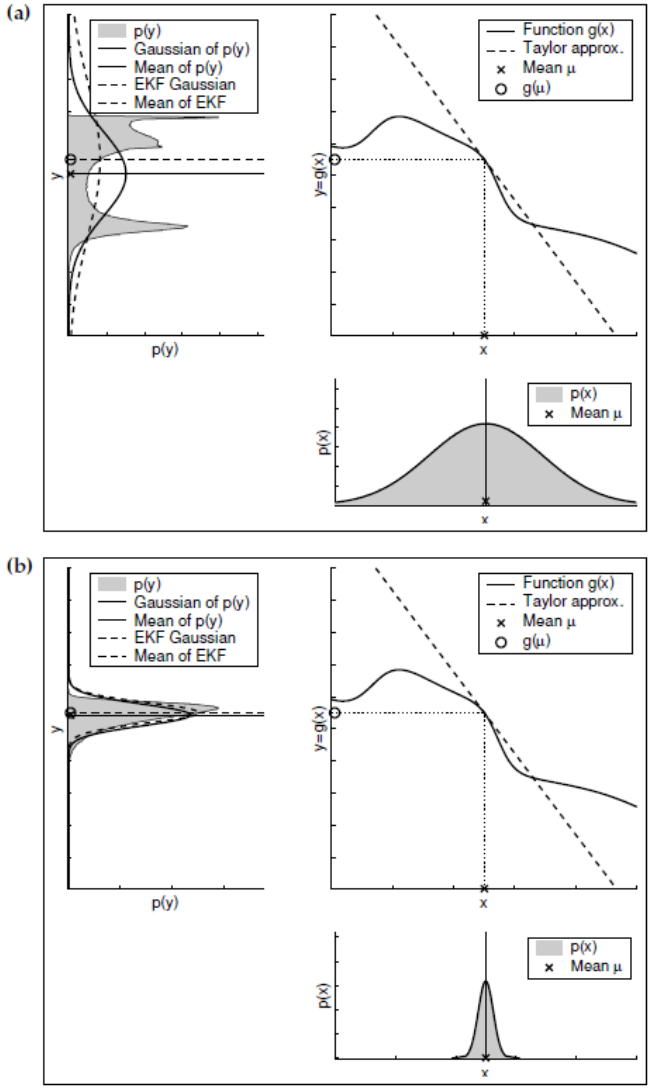


Figure 3.5 Dependency of approximation quality on uncertainty. Both Gaussians (lower right) have the same mean and are passed through the same nonlinear function (upper right). The higher uncertainty of the left Gaussian produces a more distorted density of the resulting random variable (gray area in upper left graph). The solid lines in the upper left graphs show the Gaussians extracted from these densities. The dashed lines represent the Gaussians generated by the linearization applied by the EKF.

The accuracy of the linearization depends on the degree of uncertainty.

Extended Kalman Filter

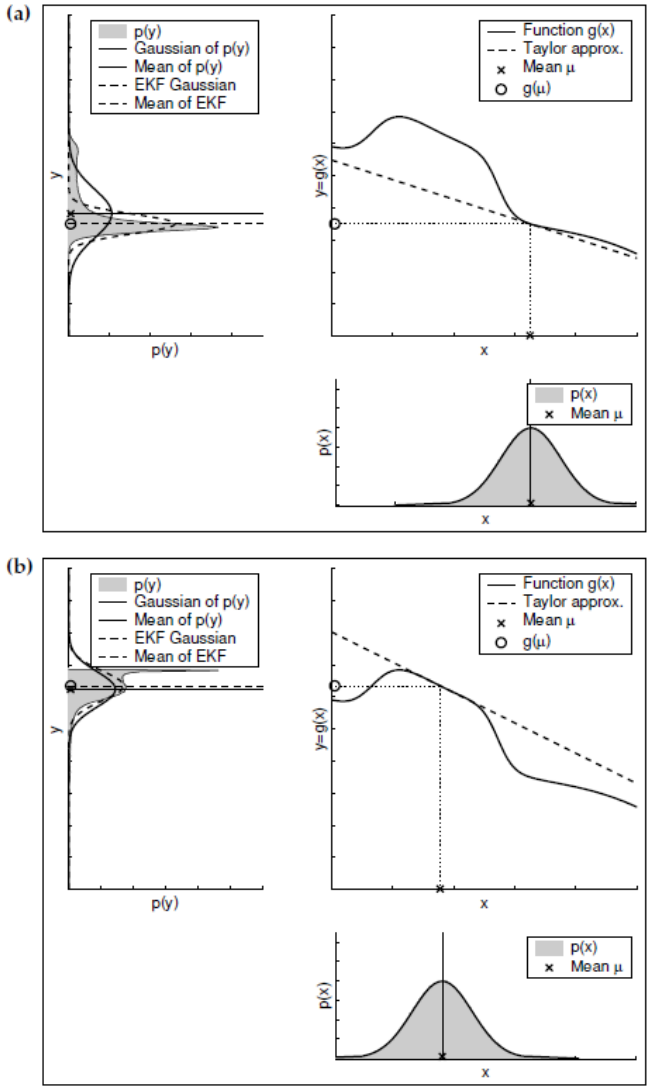


Figure 3.6 Dependence of the approximation quality on local nonlinearity of the function g. Both Gaussians (lower right in each of the two panels) have the same covariance and are passed through the same function (upper right). The linear approximation applied by the EKF is shown as the dashed lines in the upper right graphs. The solid lines in the upper left graphs show the Gaussians extracted from the highly accurate Monte-Carlo estimates. The dashed lines represent the Gaussians generated by the EKF linearization.

The accuracy of the linearization depends on the degree of local nonlinearity.

Unscented Kalman Filter

- Derivative-free linearization.
- Typically more accurate than EKF, as it may be seen as a second order Taylor expansion linearization:

$$g(u_t, x_{t-1}) \approx g(u_t, \mu_{t-1}) + g'(u_t, \mu_{t-1})(x_{t-1} - \mu_{t-1}) + g''(u_t, \mu_{t-1})(x_{t-1} - \mu_{t-1})^2/2$$

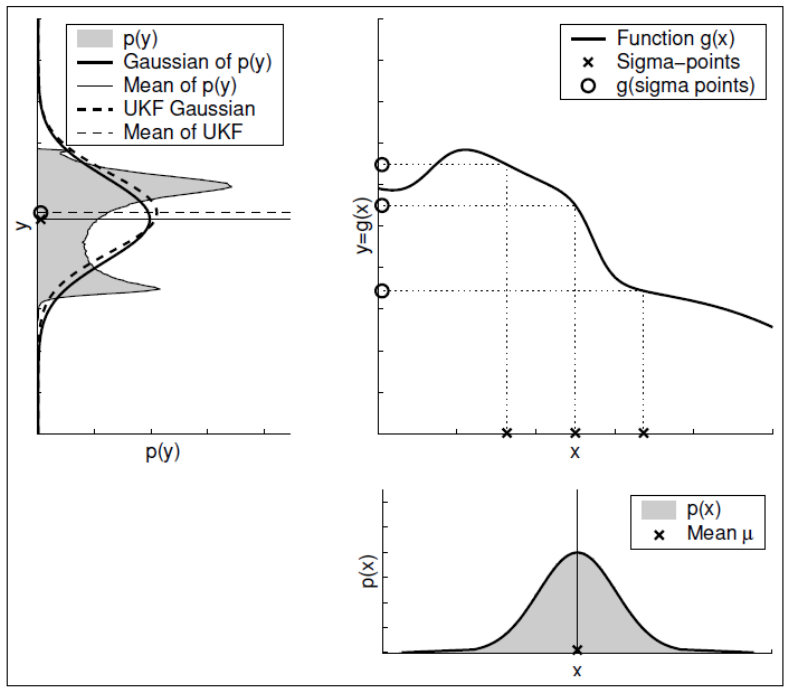


Figure 3.7 Illustration of linearization applied by the UKF. The filter first extracts $2n + 1$ weighted sigma points from the n-dimensional Gaussian ($n = 1$ in this example). These sigma points are passed through the nonlinear function g. The linearized Gaussian is then extracted from the mapped sigma points (small circles in the upper right plot). As for the EKF, the linearization incurs an approximation error, indicated by the mismatch between the linearized Gaussian (dashed) and the Gaussian computed from the highly accurate Monte-Carlo estimate (solid).

Unscented Kalman Filter

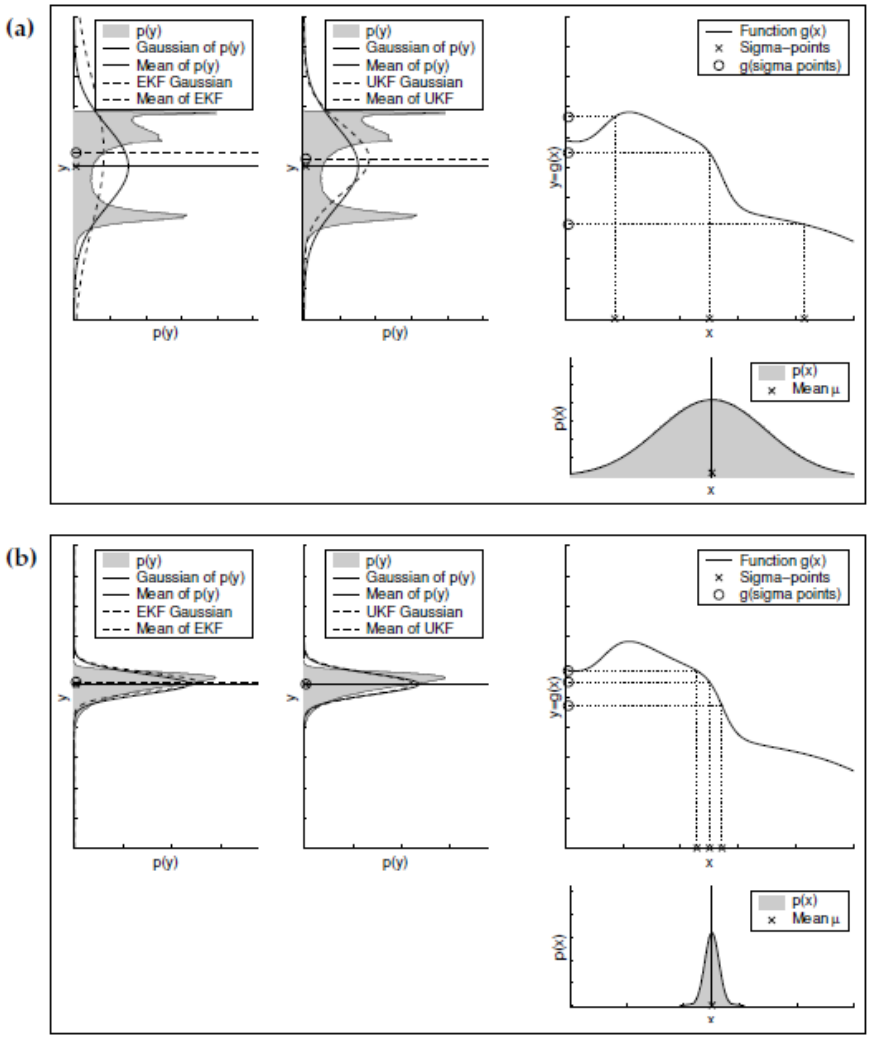


Figure 3.8 Linearization results for the UKF depending on the uncertainty of the original Gaussian. The results of the EKF linearization are also shown for comparison (c.f. Figure3.5). The unscented transform incurs smaller approximation errors, as can be seen by the stronger similarity between the dashed and the solid Gaussians.

EKF vs UKF dependence on the degree of uncertainty.

Unscented Kalman Filter

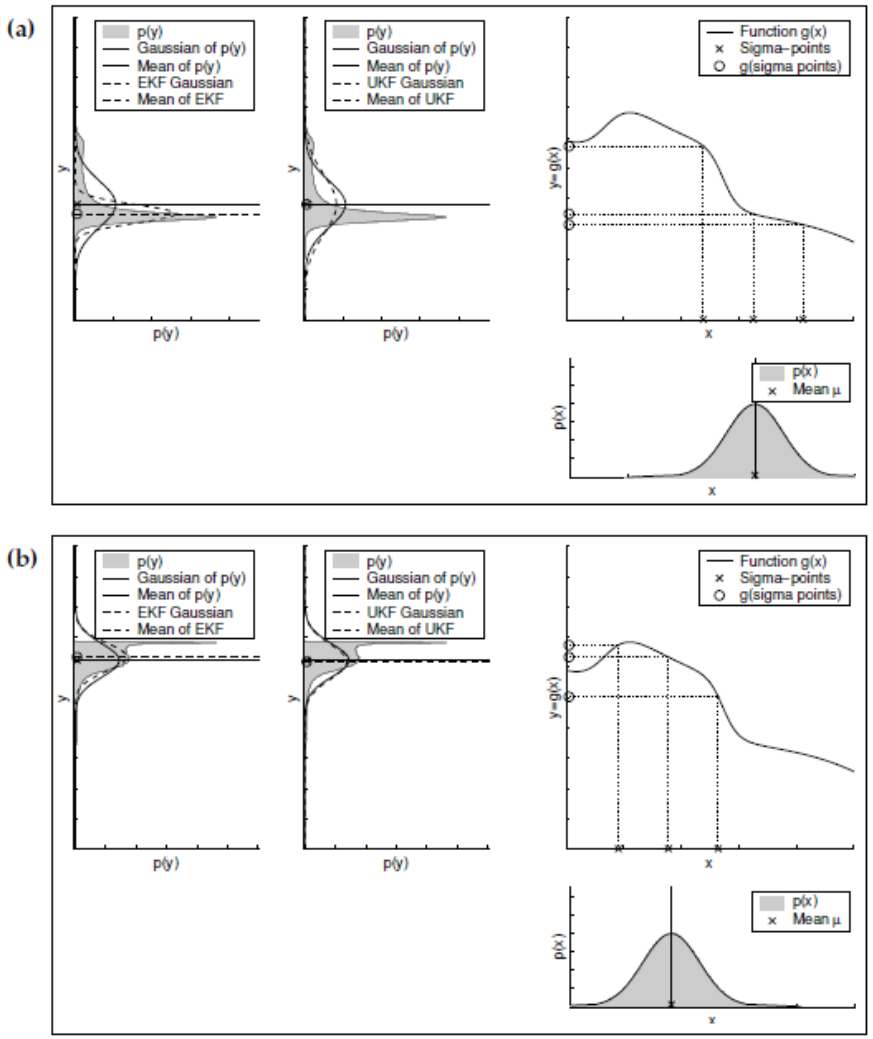


Figure 3.9 Linearization results for the UKF depending on the mean of the original Gaussian. The results of the EKF linearization are also shown for comparison (c.f. Figure3.6). The sigma point linearization incurs smaller approximation errors, as can be seen by the stronger similarity between the dashed and the solid Gaussians.

EKF vs UKF dependence on the degree of local nonlinearity.

Unscented Kalman Filter

- $\overline{bel}(x_t) = \mathcal{N}(x_t|\overline{\mu}_t, \overline{\Sigma}_t)$
- $bel(x_t) = \mathcal{N}(x_t|\mu_t, \Sigma_t)$

	Unscented Kalman filter algorithm
1	Set μ_0 and Σ_0 from $p(x_0)$
2	For $t = 1, \ldots$
3	Choose (deterministically) state sigma points from μ_{t-1} and Σ_{t-1}
4	Propagate the sigma points through $g(\cdot)$
5	Estimate $\overline{\mu}_t$ as the sample mean of the propagated sigma points
6	Estimate $\overline{\Sigma}_t$ as the sample covariance matrix plus R_t
7	Choose (deterministically) observation sigma points from $\overline{\mu}_t$ and $\overline{\Sigma}_t$
8	Propagate the sigma points through $h(\cdot)$
9	Estimate $\hat{z}_t$ as the sample mean of the propagated sigma points
10	Estimate S_t as the sample covariance matrix plus Q_t
11	Estimate $\overline{\Sigma}_t^{x,z}$ as the sample covariance between state and obs. sigma points
12	$\overline{K}_t = \overline{\Sigma}_t^{x,z} S_t^{-1}$
13	$\mu_t = \overline{\mu}_t + K_t(z_t - \hat{z}_t)$
14	$\Sigma_t = \overline{\Sigma}_t - K_t S_t K_t^T$

- Differences with KF and EKF: Use of sigma points instead of (approximated) closed forms for state and measurement predictions.
- Similarities with KF and EKF: Prediction (lines 3-11), information gain (line 12), and correction (lines 13-14).

Particle Filter

- In KF, EKF and UKF, $bel(x_t)$ and $\overline{bel}(x_t)$ are modeled as Gaussian distributions.
- Consequence: Unable to represent multimodal beliefs.
- Two main approaches to remedy this:
  - Discretize the state space: HMM a.k.a. histogram filter.
  - Approximate the state space by a random sample of $bel(x_t)$: Particle filter.

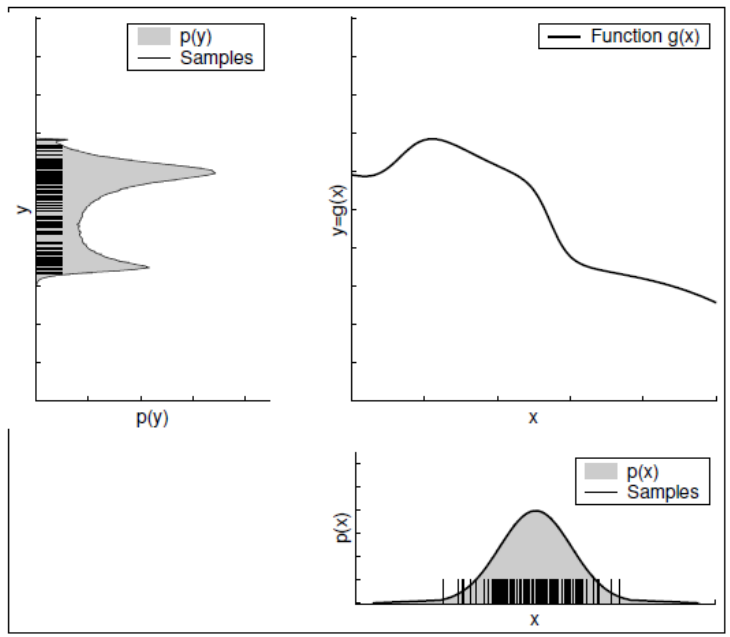


Figure 4.3 The "particle" representation used by particle filters. The lower right graph shows samples drawn from a Gaussian random variable, X. These samples are passed through the nonlinear function shown in the upper right graph. The resulting samples are distributed according to the random variable Y.

- Disadvantage: No closed forms and, thus, higher computational complexity.

Particle Filter

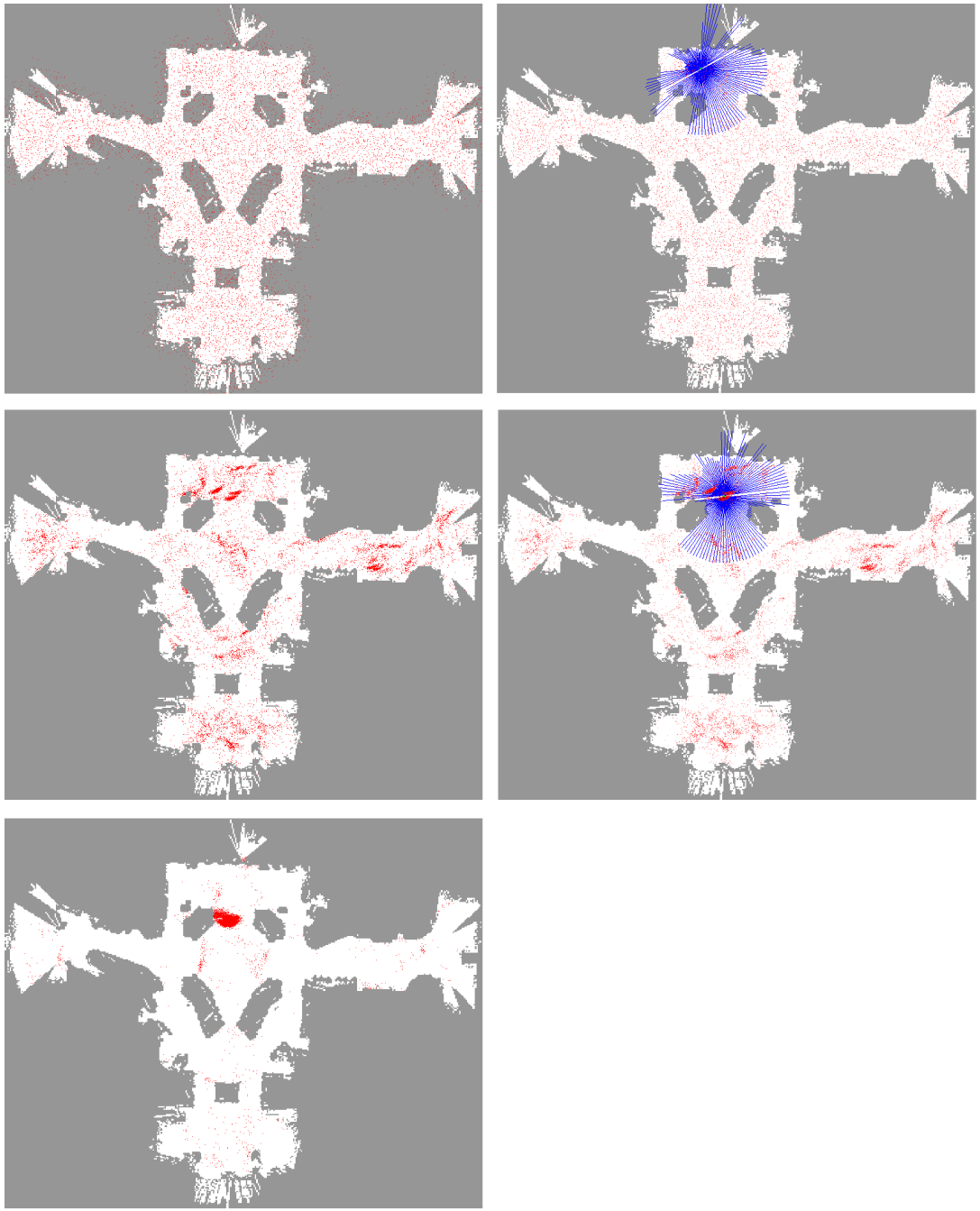
- $\overline{bel}(x_t)$ = set of samples (particles) from $p(x_t|z_{1:t-1}, u_{1:t}) = \overline{\mathcal{X}}_t$
- $bel(x_t)$ = set of samples (particles) from $p(x_t|z_{1:t}, u_{1:t}) = \mathcal{X}_t$
- Intuitively, a particle is a hypothesis about the world state at time t.

Particle filter algorithm

1	For $m = 1, \ldots, M$ do	// Initialization
2	Sample x_0^m from $p(x_0)$ and add it to $\mathcal{X}_0$	
3	For $t = 1, \ldots$	
4	For $m = 1, \ldots, M$ do	// Prediction
5	Sample x_t^m from $p(x_t\|u_t, x_{t-1}^m)$	
6	Add x_t^m to $\overline{\mathcal{X}}_t$	
7	$w_t^m = p(z_t\|x_t^m)$	// Importance weight
8	For $m = 1, \ldots, M$ do	// Correction
9	Draw with replacement x_t^i from $\overline{\mathcal{X}}_t$ with probability proportional to w_t^i	
10	Add x_t^i to $\mathcal{X}_t$	

- Note that the particles in line 5 are approximately distributed according to $\overline{bel}(x_t)$, whereas the particles in line 9 are approximately distributed according to $bel(x_t)$ because recall that $bel(x_t) \propto p(z_t|x_t)\overline{bel}(x_t)$.
- Note that we have to be able to sample $p(x_0)$ and $p(x_t|u_t, x_{t-1}^m)$, and evaluate $p(z_t|x_t^m)$.

Particle Filter



Particle Filter

- In some applications we may need to represent $bel(x_t)$ as a distribution.

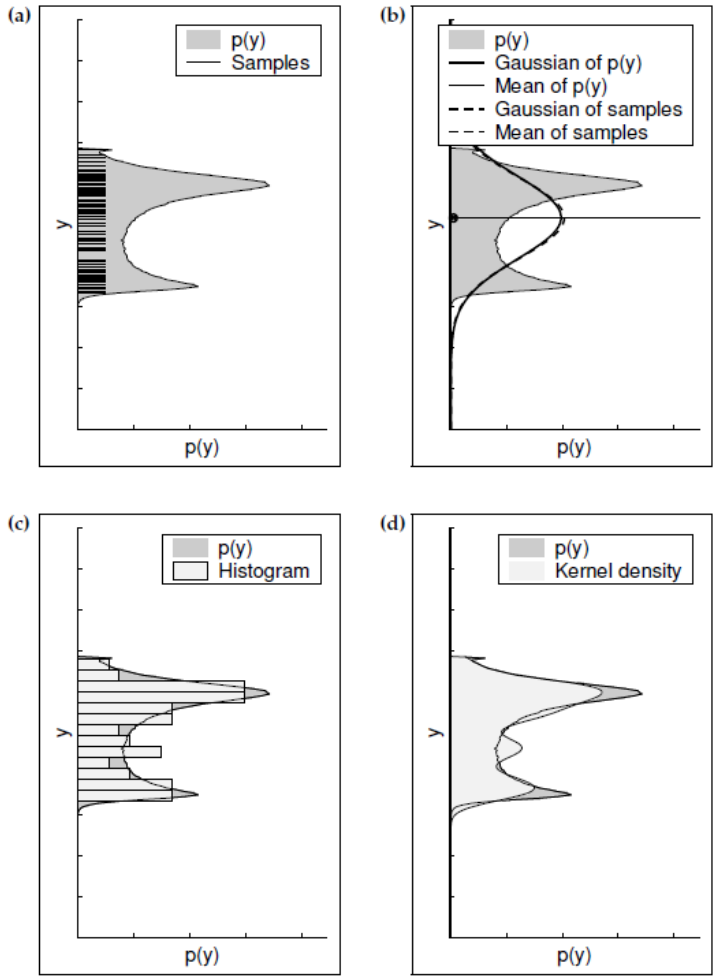


Figure 4.5 Different ways of extracting densities from particles. (a) Density and sample set approximation, (b) Gaussian approximation (mean and variance), (c) histogram approximation, (d) kernel density estimate. The choice of approximation strongly depends on the specific application and the computational resources.

Contents

- Extended Kalman Filter
- Unscented Kalman Filter
- Particle Filter

Thank you