Lecture 26, Mar 16, 2026

Gaussian Filters – Extended and Unscented Kalman Filtering

- Standard Kalman filter: assuming a linear model $\begin{cases} \boldsymbol{x}_k = \boldsymbol{A}_{k-1}\boldsymbol{x}_{k-1} + \boldsymbol{v}_k + \boldsymbol{w}_k \\ \boldsymbol{y}_k = \boldsymbol{C}_k\boldsymbol{x}_k + \boldsymbol{n}_k \end{cases}$
  - Prediction step: $\begin{cases} \check{\boldsymbol{P}}_k = \boldsymbol{A}_{k-1}\hat{\boldsymbol{P}}_{k-1}\boldsymbol{A}_{k-1}^T + \boldsymbol{Q}_k \\ \check{\boldsymbol{x}}_k = \boldsymbol{A}_{k-1}\hat{\boldsymbol{x}}_{k-1} + \boldsymbol{v}_k \end{cases}$
  - Kalman gain: $\boldsymbol{K}_k = \check{\boldsymbol{P}}_k\boldsymbol{C}_k^T(\boldsymbol{C}_k\check{\boldsymbol{P}}_k\boldsymbol{C}_k^T + \boldsymbol{R}_k)^{-1}$
  - Corrections step: $\begin{cases} \hat{\boldsymbol{P}}_k = (\boldsymbol{1} - \boldsymbol{K}_k\boldsymbol{C}_k)\check{\boldsymbol{P}}_k \\ \hat{\boldsymbol{x}}_k = \check{\boldsymbol{x}}_k + \boldsymbol{K}_k(\boldsymbol{y}_k - \boldsymbol{C}_k\check{\boldsymbol{x}}_k) \end{cases}$
  - Assuming modelling is correct, the Kalman filter is *consistent* (correct covariance) and *unbiased* (correct mean)

Extended Kalman Filtering (EKF)

- Now we assume nonlinear models: $\begin{cases} \boldsymbol{x}_k = \boldsymbol{f}(\boldsymbol{x}_{k-1}, \boldsymbol{v}_k, \boldsymbol{w}_k) \\ \boldsymbol{y}_k = \boldsymbol{g}(\boldsymbol{x}_k, \boldsymbol{n}_k) \end{cases}$
- Derive the EKF from the Bayes filter: $$\begin{aligned} & p(\boldsymbol{x}_k|\check{\boldsymbol{x}}_0, \boldsymbol{v}_{1:k}, \boldsymbol{y}_{0:k}) \\ &= \eta p(\boldsymbol{y}_k|\boldsymbol{x}_k)\int p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1}, \boldsymbol{v}_k)p(\boldsymbol{x}_{k-1}|\check{\boldsymbol{x}}_0, \boldsymbol{v}_{1:k-1}, \boldsymbol{y}_{0:k-1})\,\mathrm{d}\boldsymbol{x}_{k-1} \end{aligned}$$
  - First approximation: assume belief and noise are always Gaussian
    - $p(\boldsymbol{x}_k|\check{\boldsymbol{x}}_0, \boldsymbol{v}_{1:k}, \boldsymbol{y}_{0:k}) = \mathcal{N}(\hat{\boldsymbol{x}}_k, \hat{\boldsymbol{P}}_k)$
    - Assume $\boldsymbol{w}_k \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{Q}_k), \boldsymbol{n}_k \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{R}_k)$
  - Second approximation: linearize the motion and observation models
    - $\boldsymbol{f}(\boldsymbol{x}_{k-1}, \boldsymbol{v}_k, \boldsymbol{w}_k) \approx \check{\boldsymbol{x}}_k + \boldsymbol{F}_{k-1}(\boldsymbol{x}_{k-1} - \hat{\boldsymbol{x}}_{k-1}) + \boldsymbol{w}_k'$
      - $\boldsymbol{F}_{k-1} = \dfrac{\partial \boldsymbol{f}(\boldsymbol{x}_{k-1}, \boldsymbol{v}_k, \boldsymbol{w}_k)}{\partial \boldsymbol{x}_{k-1}}$
    - $\boldsymbol{g}(\boldsymbol{x}_k, \boldsymbol{n}_k) \approx \check{\boldsymbol{y}}_k + \boldsymbol{G}_k(\boldsymbol{x}_k - \check{\boldsymbol{x}}_k) + \boldsymbol{n}_k'$
      - $\boldsymbol{G}_k = \dfrac{\partial \boldsymbol{g}(\boldsymbol{x}_k, \boldsymbol{n}_k)}{\partial \boldsymbol{x}_k}$
    - The predictions $\check{\boldsymbol{x}}_k, \check{\boldsymbol{y}}_k$ are calculated using the nonlinear model
    - Note $\boldsymbol{w}_k', \boldsymbol{n}_k'$ are not the same as the original noise terms if the noise affects the model in a nonlinear way; usually we just have an additive noise model however, so these would be the same
      - For non-additive noise, we transform the noise and noise covariance using the Jacobians $\dfrac{\partial \boldsymbol{f}(\boldsymbol{x}_{k-1}, \boldsymbol{v}_k, \boldsymbol{w}_k)}{\partial \boldsymbol{w}_k}$ and $\dfrac{\partial \boldsymbol{g}(\boldsymbol{x}_k, \boldsymbol{n}_k)}{\partial \boldsymbol{n}_k}$
  - Taking expectations, $\begin{cases} p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1}, \boldsymbol{v}_k) = \mathcal{N}(\check{\boldsymbol{x}}_k + \boldsymbol{F}_{k-1}(\boldsymbol{x}_{k-1} - \hat{\boldsymbol{x}}_{k-1}), \boldsymbol{Q}_k') \\ p(\boldsymbol{y}_k|\boldsymbol{x}_k) \approx \mathcal{N}(\check{\boldsymbol{y}}_k + \boldsymbol{G}_k(\boldsymbol{x}_k - \check{\boldsymbol{x}}_k), \boldsymbol{R}_k') \end{cases}$
  - Now we substitute the approximated PDFs into the Bayes filter equation, and use the direct product and Gaussian inference identities
  - We have the new distribution $\mathcal{N}(\check{\boldsymbol{x}}_k + \boldsymbol{K}_k(\boldsymbol{y}_K - \check{\boldsymbol{y}}_k), (\boldsymbol{1} - \boldsymbol{K}_k\boldsymbol{G}_k)(\boldsymbol{F}_{k-1}\hat{\boldsymbol{P}}_{k-1}\boldsymbol{F}_{k-1}^T + \boldsymbol{Q}_k'))$ where the Kalman gain is $\boldsymbol{K}_k = \check{\boldsymbol{P}}_k\boldsymbol{G}_k^T(\boldsymbol{G}_k\check{\boldsymbol{P}}_k\boldsymbol{G}_k^T + \boldsymbol{R}_k')^{-1}$
  - Final update equations:
    - Prediction: $\begin{cases} \check{\boldsymbol{P}}_k = \boldsymbol{F}_{k-1}\hat{\boldsymbol{P}}_k\boldsymbol{F}_{k-1}^T + \boldsymbol{Q}_k' \\ \check{\boldsymbol{x}}_k = \boldsymbol{f}(\hat{\boldsymbol{x}}_{k-1}, \boldsymbol{v}_k, \boldsymbol{0}) \end{cases}$
    - Kalman gain: $\boldsymbol{K}_k = \check{\boldsymbol{P}}_k\boldsymbol{G}_k^T(\boldsymbol{G}_k\check{\boldsymbol{P}}_k\boldsymbol{G}_k^T + \boldsymbol{R}_k')^{-1}$
    - Correction: $\begin{cases} \hat{\boldsymbol{P}}_k = (\boldsymbol{1} - \boldsymbol{K}_k\boldsymbol{G}_k)\check{\boldsymbol{P}}_k \\ \hat{\boldsymbol{x}}_k = \check{\boldsymbol{x}}_k + \boldsymbol{K}_k(\boldsymbol{y}_k - \boldsymbol{g}(\check{\boldsymbol{x}}_k, \boldsymbol{0})) \end{cases}$
- EKFs only work well for mildly nonlinear non-Gaussian systems, as our approximation breaks down if

the model deviates too much
  - Our linearization point is about the estimate instead of the true state, so as the system becomes more uncertain the approximation gets worse
  - The distribution also becomes less and less Gaussian as we pass it through the nonlinearity
  - Therefore the EKF can be biased and inconsistent and lead to divergence
  - The Jacobian is a first-order approximation; we need a better way to transform PDFs through nonlinearities

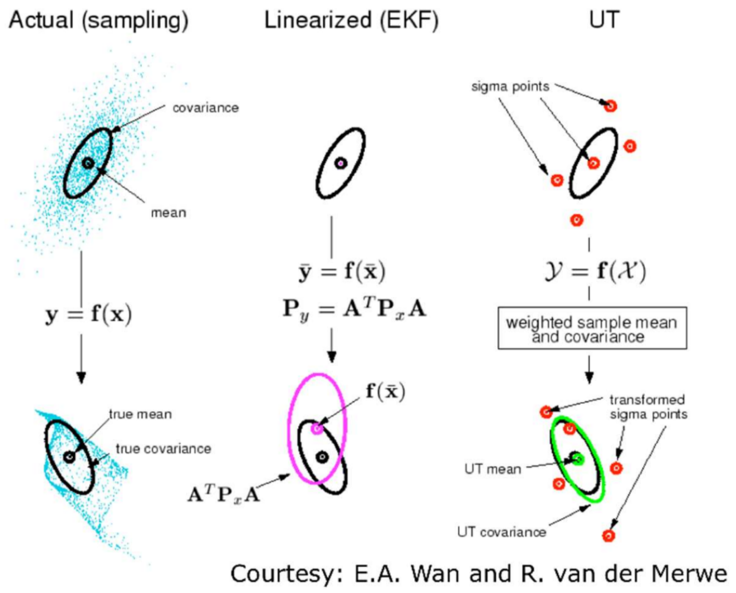
Unscented Kalman Filtering (UKF)

- The *unscented* or *sigma point* transformation transforms PDFs by drawing a small number of deterministic and representative samples, transforming them with the nonlinearity, and using the result to reconstruct the output PDF
  1. Compute a set of $2L+1$ *sigma points*, where L is the dimensionality of the state
     - Take the Cholesky decomposition: $\boldsymbol{L}\boldsymbol{L}^T = \boldsymbol{\Sigma}_x$
     - The first point is the mean $\boldsymbol{x}_0 = \boldsymbol{\mu}_x$
     - For other samples, $\begin{cases} \boldsymbol{x}_i = \boldsymbol{\mu}_x + \sqrt{L+\kappa}\operatorname{col}_i \boldsymbol{L} \\ \boldsymbol{x}_{i+L} = \boldsymbol{\mu}_x - \sqrt{L+\kappa}\operatorname{col}_i \boldsymbol{L} \end{cases}$
       - κ is a parameter we control; the larger it is, the more spread out the samples are
       - There is a particular κ which minimizes error for Gaussians
     - We essentially take samples in all directions of the mean, one for each dimension
  2. Pass each through the nonlinearity: $\boldsymbol{y}_i = \boldsymbol{g}(\boldsymbol{x}_i)$
  3. Compute the output PDF:
     - $\boldsymbol{\mu}_y = \sum_{i=0}^{2L} \alpha_i \boldsymbol{y}_i$ with weight $\alpha_i = \begin{cases} \dfrac{\kappa}{L+\kappa} & i = 0 \\ \dfrac{1}{2}\dfrac{1}{L+\kappa} & \text{otherwise} \end{cases}$
     - $\boldsymbol{\Sigma}_{yy} = \sum_{i=0}^{2L} \alpha_i (\boldsymbol{y}_i - \boldsymbol{\mu}_y)(\boldsymbol{y}_i - \boldsymbol{\mu}_y)^T$
     - Then we approximate the output density as a Gaussian with this mean and covariance
- The *unscented Kalman filter* (UKF) uses the transform:
  - Prediction step:
    1. Stack the prior belief and motion noise as $\boldsymbol{\mu}_z = \begin{bmatrix} \hat{\boldsymbol{x}}_{k-1} \\ \boldsymbol{0} \end{bmatrix}, \boldsymbol{\Sigma}_{zz} = \begin{bmatrix} \hat{\boldsymbol{P}}_{k-1} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{Q}_k \end{bmatrix}$
    2. Sample sigma points $\boldsymbol{z}_i$
    3. Extract state and motion noise from each sigma point as $\boldsymbol{z}_i = \begin{bmatrix} \hat{\boldsymbol{x}}_{k-1,i} \\ \boldsymbol{w}_{k,i} \end{bmatrix}$, and pass through nonlinear motion model $\check{\boldsymbol{x}}_{k,i} = \boldsymbol{f}(\hat{\boldsymbol{x}}_{k-1,i}, \boldsymbol{v}_k, \boldsymbol{w}_{k,i})$
    4. Compute new predicted mean and covariance from transformed sigma points:
       - $\check{\boldsymbol{x}}_k = \sum_{i=0}^{2L} \alpha_i \check{\boldsymbol{x}}_{k,i}$
       - $\check{\boldsymbol{P}}_k = \sum_{i=0}^{2L} \alpha_i (\check{\boldsymbol{x}}_{k,i} - \check{\boldsymbol{x}}_k)(\check{\boldsymbol{x}}_{k,i} - \check{\boldsymbol{x}}_k)^T$
  - Correction step:
    1. Stack prediction belief and observation noise as $\boldsymbol{\mu}_z = \begin{bmatrix} \check{\boldsymbol{x}}_k \\ \boldsymbol{0} \end{bmatrix}, \boldsymbol{\Sigma}_{zz} = \begin{bmatrix} \check{\boldsymbol{P}}_k & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{R}_k \end{bmatrix}$
    2. Sample sigma points
    3. Extract state and measurement noise for each sigma point as $\boldsymbol{z}_i = \begin{bmatrix} \check{\boldsymbol{x}}_{k,i} \\ \boldsymbol{n}_{k,i} \end{bmatrix}$ and pass through nonlinear measurement model
    4. Compute new predicted mean and covariance

    - $\boldsymbol{\mu}_{y,k} = \sum_{i=0}^{2L} \alpha_i \check{\boldsymbol{y}}_{k,i}$
    - $\boldsymbol{\Sigma}_{yy,k} = \sum_{i=0}^{2L} \alpha_i (\check{\boldsymbol{y}}_{k,i} - \boldsymbol{\mu}_{y,k})(\check{\boldsymbol{y}}_{k,i} - \boldsymbol{\mu}_{y,k})^T$
    - $\boldsymbol{\Sigma}_{xy,k} = \sum_{i=0}^{2L} \alpha_i (\check{\boldsymbol{x}}_{k,i} - \check{\boldsymbol{x}}_k)(\check{\boldsymbol{y}}_{k,i} - \boldsymbol{\mu}_{y,k})^T$

  5. Update: $\begin{cases} \boldsymbol{K}_k = \boldsymbol{\Sigma}_{xy,k}\boldsymbol{\Sigma}_{yy,k}^{-1} \\ \hat{\boldsymbol{P}}_k = \check{\boldsymbol{P}}_k - \boldsymbol{K}_k \boldsymbol{\Sigma}_{xy,k}^T \\ \hat{\boldsymbol{x}}_k = \check{\boldsymbol{x}}_k + \boldsymbol{K}_k(\boldsymbol{y}_k - \boldsymbol{\mu}_{y,k}) \end{cases}$

- The UKF still assumes Gaussian beliefs, but the linearization considers a much larger area and captures the resulting distribution much better
  - The unscented transform is a third-order approximation compared to the first-order approximation of EKF Jacobians
  - The UKF also does not need Jacobians to be explicitly computed
  - The parameter κ can be tweaked to work better with the specific nonlinearity



Courtesy: E.A. Wan and R. van der Merwe

Figure 1: Comparison of EKF and UFK approximations.