A Review of Probabilistic Data Assimilation Methods for Online State Estimation

Xin Li

Florida State University

xli2@math.fsu.edu

February 11, 2019
Outline

1. Introduction
2. Kalman Filter
3. Ensemble Kalman Filter
4. SIR Filter
5. Auxiliary Particle Filter
6. Implicit Particle Filter
7. Experiments
8. Conclusion and Future Work
Data Assimilation

Data assimilation is a mathematical discipline that seeks to optimally combine theory (usually in the form of a numerical model) with observations.

- **Objective:** determine information about the unknown quantity in numerical model given observations.
- **Used in:** state estimation, determining initial conditions, parameter estimation.
- **Two key problems:** filtering (on-line) and smoothing (off-line)
- **Objective of this presentation:** Review five filtering methods for state estimation.
Discrete Time Set-up for Filtering

- Initial state $x_0$ is given.
- Dynamic equation
  \[ x_{t+1} = f(x_t) + v_t \]
  where $t \in \mathbb{N}$, $f \in C(\mathbb{R}^{nx}, \mathbb{R}^{nx})$, $\{v_t\}_{t \in \mathbb{N}} \overset{i.i.d}{\sim} N(0, Q)$
- Measurement equation
  \[ y_{t+1} = h(x_{t+1}) + w_{t+1} \]
  where $t \in \mathbb{N}$, $h \in C(\mathbb{R}^{nx}, \mathbb{R}^{ny})$, $\{w_t\}_{t \in \mathbb{N}} \overset{i.i.d}{\sim} N(0, R)$
- Goal: Estimate $x_{t+1}$ based on noisy measurement $y_1 = Y_1, \ldots, y_{t+1} = Y_{t+1}$
Notation

- **Random Variables** \( x_0, x_t, y_t, v_t, w_t \) for \( t \in \mathbb{Z}^+ \)
- **Markov Chain** \( x_{0:t+1} = \{x_0, x_1, x_2, \ldots, x_{t+1}\} \)
- **Data** \( Y_{1:t+1} = \{y_1 = Y_1, \ldots, y_{t+1} = Y_{t+1}\} \)
- **Transition Density** \( p(x_{t+1}|x_t) \)
- **Prior** \( p(x_{t+1}|y_{1:t}) \)
- **Likelihood** \( p(y_{t+1}|x_{t+1}) \)
- **Posterior** \( p(x_{0:t+1}|y_{1:t+1}) \)
- **Filtering Density** \( p(x_{t+1}|y_{1:t+1}) \)
- **Mean** \( m_t = E[x_t|y_{1:t} = Y_{1:t}] \)
- **Predicted Mean** \( \hat{m}_{t+1} = E[x_{t+1}|y_{1:t} = Y_{1:t}] \)
- **Covariance** \( C_t = \text{Cov}[x_t|y_{1:t} = Y_{1:t}] \)
- **Predicted Covariance** \( \hat{C}_{t+1} = \text{Cov}[x_{t+1}|y_{1:t} = Y_{1:t}] \)
A Probabilistic View of Dynamic System

- Initial guess $x_0 \sim N(m_0, C_0)$ is given.

- Dynamic equation $x_{t+1} = f(x_t) + v_t, \{v_t\}_{t \in \mathbb{N}} \overset{i.i.d.}{\sim} N(0, Q)$ at fixed $x_t = X_t$ describes the transition density of $x_{t+1}$

  $$p(x_{t+1}|x_t = X_t) = N(x_{t+1} - f(X_t); 0, Q)$$

  $$\propto \exp(-\frac{1}{2}|Q^{-1/2}(x_{t+1} - f(X_t))|^2)$$

- Measurement equation $y_{t+1} = h(x_{t+1}) + w_{t+1}, \{v_t\}_{t \in \mathbb{N}} \overset{i.i.d.}{\sim} N(0, R)$ at fixed $x_{t+1} = X_{t+1}$ describes the likelihood of $y_{t+1}$

  $$p(y_{t+1}|x_{t+1} = X_{t+1}) = N(y_{t+1} - h(X_{t+1}); 0, R)$$

  $$\propto \exp(-\frac{1}{2}|R^{-1/2}(y_{t+1} - h(X_{t+1}))|^2)$$
Objective of filtering: determining $p(x_{t+1}|y_{1:t+1})$

Compute $p(x_{t+1}|Y_{1:t+1})$ sequentially in time in two steps:

1. Prediction: $p(x_t|Y_{1:t}) \rightarrow p(x_{t+1}|Y_{1:t})$

$$p(x_{t+1}|Y_{1:t}) = \int_{\mathbb{R}^{n_x}} p(x_{t+1}|Y_{1:t}, x_t) p(x_t|Y_{1:t}) \, dx_t$$

$$= \int_{\mathbb{R}^{n_x}} p(x_{t+1}|x_t) p(x_t|Y_{1:t}) \, dx_t$$

2. Update: $p(x_{t+1}|Y_{1:t}) \rightarrow p(x_{t+1}|Y_{1:t+1})$

$$p(x_{t+1}|Y_{1:t+1}) = \frac{p(Y_{t+1}|x_{t+1}, Y_{1:t}) p(x_{t+1}|Y_{1:t})}{p(Y_{t+1}|Y_{1:t})}$$

$$\propto p(Y_{t+1}|x_{t+1}) p(x_{t+1}|Y_{1:t})$$
Initial guess $x_0 \sim N(m_0, C_0)$ is given.

Dynamic equation:

$$x_{t+1} = Fx_t + v_t$$

$v_t \overset{i.i.d.}{\sim} N(0, Q), F \in \mathbb{R}^{n_x \times n_x}$

Measurement equation:

$$y_{t+1} = Hx_{t+1} + w_{t+1}$$

$w_{t+1} \overset{i.i.d.}{\sim} N(0, R), H \in \mathbb{R}^{n_y \times n_x}$
Kalman Filter

- Given filtering distribution at time $t$:
  \[ p(x_t|Y_{1:t}) = N(x_t; m_t, C_t) \]

- Both Prediction and Update steps preserve Gaussianity.
  - Prior:
    \[ p(x_{t+1}|Y_{1:t}) = N(x_{t+1}; \hat{m}_{t+1}, \hat{C}_{t+1}) \]
  - Filtering:
    \[ p(x_{t+1}|Y_{1:t+1}) = N(x_{t+1}; m_{t+1}, C_{t+1}) \]

- Derivation objective: Derive the map of mean and covariance in prediction and update steps.
  \[ m_t, C_t \xrightarrow{\text{prediction}} \hat{m}_{t+1}, \hat{C}_{t+1} \xrightarrow{\text{update}} m_{t+1}, C_{t+1} \]
Kalman Filter: Derivation of Prediction Mapping

Derive the map: \( m_t, C_t \xrightarrow{\text{prediction}} \hat{m}_{t+1}, \hat{C}_{t+1} \) from dynamic equation

\[
x_{t+1} = Fx_t + v_t, \quad v_t \overset{i.i.d.}{\sim} N(0, Q)
\]

- Compute the predicted mean,

\[
\hat{m}_{t+1} = \mathbb{E}[x_{t+1} | Y_{1:t}] = \mathbb{E}[Fx_t | Y_{1:t}] + \mathbb{E}[v_t | Y_{1:t}] = Fm_t
\]

- Compute the predicted covariance.

\[
\hat{C}_{t+1} = \mathbb{E}[(x_{t+1} - \hat{m}_{t+1})(x_{t+1} - \hat{m}_{t+1}) | Y_{1:t}] = \\
\ldots = F \mathbb{E}[(x_t - m_t)(x_t - m_t) | Y_{1:t}]F^T + Q \\
= FC_tF^T + Q
\]
Kalman Filter: Derivation of Update Mapping

Derive the map: \( \hat{m}_{t+1}, \hat{C}_{t+1} \xrightarrow{\text{update}} m_{t+1}, C_{t+1} \)

- By Bayes theorem,

\[
p(x_{t+1}|y_{1:t+1}) \propto p(y_{t+1}|x_{t+1})p(x_{t+1}|y_{1:t})
\]

- Equating the exponential term

\[
\exp\left(-\frac{1}{2}C_{t+1}^{-1/2}(x_{t+1} - m_{t+1})^2\right)
\]

\[
\propto \exp\left(-\frac{1}{2}R^{-1/2}(y_{t+1} - H(x_{t+1}))^2\right)\exp\left(-\frac{1}{2}\hat{C}_{t+1}^{-1/2}(x_{t+1} - \hat{m}_{t+1})^2\right)
\]

- Equating quadratic terms in \( x_{t+1} \) gives

\[
C_{t+1}^{-1} = \hat{C}_{t+1}^{-1} + H^T R^{-1} H
\]

- Equating linear terms in \( x_{t+1} \) gives

\[
C_{t+1}^{-1} m_{t+1} = \hat{C}_{t+1}^{-1} \hat{m}_{t+1} + H^T R^{-1} y_{t+1}
\]
Kalman Filter: Derivation of Update Mapping

- Compute $C_{t+1}$ by Sherman-Morrison-Woodbury Formula.

$$C_{t+1} = (C_t^{-1})^{-1} = (\hat{C}_{t+1}^{-1} + H^T R^{-1} H)^{-1}$$

$$\cdots = \hat{C}_{t+1} - (\hat{C}_{t+1} H^T (H \hat{C}_{t+1} H^T + R)^{-1}) H \hat{C}_{t+1}$$

$$= \hat{C}_{t+1} - K_{t+1} H \hat{C}_{t+1}$$

- Compute $m_{t+1}$

$$m_{t+1} = C_{t+1}(C_{t+1}^{-1} m_{t+1}) = \cdots = \hat{m}_{t+1} + K_{t+1}(y_{t+1} - H \hat{m}_{t+1})$$
Kalman Filter: Iterative Process

update step

\[ K := \hat{C} H^T \left[ H \hat{C} H^T + R \right]^{-1} \]
\[ m := \hat{m} + K (y - H \hat{m}) \]
\[ C := \hat{C} - KH \hat{C} \]

prediction step

\[ \hat{m} := Fm \]
\[ \hat{C} := FCF^T + Q \]

initial estimate
\[ m := m_0 \]
\[ C := C_0 \]

filter output
\[ m_t := m \]
\[ C_t := C \]
Kalman Filter: Properties

- Convergence theorem: The rate of adaptation to new data $r$ is defined by $R$ and $Q$. As $t \to \infty$, $C_t \to C = AR$ [Mike West and Jeff Harrison, p44]
  where $0 < A < I$
  \[
  A = \frac{r}{2}(\sqrt{1 + \frac{4}{r}} - 1)
  \]

- Complexity:
  - Calculating and storing the $n_x \times n_x$ matrices $\hat{C}_{t+1|t}$ and $\hat{C}_{t+1|t+1}$ are expensive.
  - Calculating the effective of the $n_y \times n_y$ matrix in Kalman Gain is expensive.
  - Computational complexity $O(n_x^3)$
Kalman Filter Approximation: Ensemble Kalman Filter

Generate N ensemble members from initial guess ($N \ll n_x$)

$$\{x_0^{(i)}\}_{i=1}^N \sim N(m_0, C_0)$$

Propagate in prediction and update steps

$$\{x_t^{(i)}\}_{i=1}^N \xrightarrow{\text{prediction}} \{\hat{x}_{t+1}^{(i)}\}_{i=1}^N \xrightarrow{\text{update}} \{x_{t+1}^{(i)}\}_{i=1}^N$$

The mean $m_{t+1}$ and covariance $C_{t+1}$ of Kalman filter are approximated by ensemble

$$\tilde{m}_{t+1} = \frac{1}{N} \sum_{n=1}^{N} x_{t+1}^{(i)}$$

$$\tilde{C}_{t+1} = \frac{1}{N-1} \sum_{n=1}^{N} (x_{t+1}^{(i)} - \tilde{m}_{t+1})(x_{t+1}^{(i)} - \tilde{m}_{t+1})^T$$
Ensemble Kalman Filter: Properties

Convergence:
- EnKF results converge to the ones in Kalman Filter as $N \to \infty$ under Monte Carlo approximation.
- No proof about convergence in time steps for fixed ensemble size $N$ and unclear about how large of of $N$ is needed for high dimensional approximation.

Complexity:
- Storing and update $N$ vectors with size $n_x \times 1$
- Computational complexity $O(n_xN^2)$
- Applicable in high dimensional state vector.
Derive the map \( \{ x_t^{(i)} \}_{i=1}^N \xrightarrow{\text{prediction}} \{ \hat{x}_{t+1}^{(i)} \}_{i=1}^N \)

- Given ensemble \( \{ x_t^{(i)} \}_{i=1}^N \)
- Compute predicted ensemble members,

\[
\hat{x}_{t+1}^{(i)} = Fx_t^{(i)} + v_t^{(i)}
\]

where \( v_t^{(i)} \) is a realization of \( v_t \) from \( N(0, Q) \)

As \( N \to \infty \), \( N(E[\hat{x}_{t+1}^{(i)}], \text{Cov}[\hat{x}_{t+1}^{(i)}]) \) converges to \( N(\hat{m}_{t+1}, \hat{C}_{t+1}) \). [Geir Evensen, 2003]
Derive the map \( \{ \hat{x}_{t+1}^{(i)} \}_{i=1}^{N} \xrightarrow{\text{update}} \{ x_{t+1}^{(i)} \}_{i=1}^{N} \)

- Generate measurement sample using predicted ensemble,

\[ \hat{y}_{t+1}^{(i)} = H\hat{x}_{t+1}^{(i)} + w_{t+1}^{(i)} \]

where \( w_{t+1}^{(i)} \) is a realization of \( w_{t+1} \) from \( N(0, R) \)

- Compute ensemble members at present time.

\[ x_{t+1}^{(i)} = \hat{x}_{t+1}^{(i)} + K_{t+1}(y_{t+1} - \hat{y}_{t+1}^{(i)}) \]

As \( N \to \infty \), \( N(E[x_{t+1}^{(i)}], Cov[x_{t+1}^{(i)}]) \) converges to \( N(m_{t+1}, C_{t+1}) \). [Geir Evensen, 2003]
Ensemble Kalman Filter Algorithm

Initialization: Generate N ensemble members, \( \{ x_0^{(i)} \}_{i=1}^N \sim \mathcal{N}(m_0, C_0) \)

For \( t = 1 : T \)

1. For \( i = 1 : N \)
   - \( \hat{x}_{t+1}^{(i)} = F(x_t^{(i)}) + v_t^{(i)} \)
   - \( \hat{m}_{t+1} = \frac{1}{N} \sum_{n=1}^{N} \hat{x}_{t+1}^{(i)} \)
   - \( \hat{C}_{t+1} = \frac{1}{N-1} \sum_{n=1}^{N} (\hat{x}_{t+1}^{(i)} - \hat{m}_{t+1}) (\hat{x}_{t+1}^{(i)} - \hat{m}_{t+1})^T \)

2. For \( i = 1 : N \)
   - \( K_{t+1} = \hat{C}_{t+1} H^T (H \hat{C}_{t+1} H^T + R)^{-1} \)
   - \( \hat{y}_{t+1}^{(i)} = H \hat{x}_{t+1}^{(i)} + w_{t+1}^{(i)} \)
   - \( x_{t+1}^{(i)} = \hat{x}_{t+1}^{(i)} + K_{t+1} (Y_{t+1} - \hat{y}_{t+1}^{(i)}) \)

The output at each time step are ensemble members \( \{ x_{t+1}^{(i)} \}_{i=1}^N \).
In estimation of the form (intractable integral)

\[ E_p[x_{t+1}|y_{1:t+1}] = \int x_{t+1}|y_{1:t+1} p(x_{t+1}|y_{1:t+1}) dx_{t+1}|y_{1:t+1} \]

\( p(x_{t+1}|y_{1:t+1}) \) could be approximated by

\[ \hat{p}(x_{t+1}|y_{1:t+1}) = \frac{1}{N} \sum_{i=1}^{N} \delta(x_{t+1} - x_{t+1}^{(i)}) \]

where \( \{x_{t+1}^{(i)}\}_{i=1}^{N} \) are i.i.d. from \( p(x_{t+1}|y_{1:t+1}) \), In EnKF, \( \{x_{t+1}^{(i)}\}_{i=1}^{N} \) are obtained from \( \{x_{t}^{(i)}\}_{i=1}^{N} \) in filtering process.

So the estimation could be approximated by tractable weighted sum:

\[ E_p[x_{t+1}|y_{1:t+1}] \approx \frac{1}{N} \sum_{i=1}^{N} x_{t+1}^{(i)} \]
Nonlinear Filtering Problem

Nonlinear dynamic with linear measurement and Gaussian noise:

\[ x_{t+1} = f(x_t) + v_t \]

\[ y_{t+1} = Hx_{t+1} + w_t \]

- True filtering distribution is non-Gaussian.
- Particle filter algorithms are used in solving nonlinear filtering problems.
  - Pro: Provable of convergence to true filtering distribution.
  - Con: Computationally expensive in high dimensional case.
Monte Carlo Approximation: Importance Sampling

If $p(x)$ is difficult to sample from but easy to evaluate. To have a Monte Carlo approximation of the form,

$$E_p[x] = \int xp(x)dx$$

Choose a proposal density $q(x)$ that is easy to sample from and write

$$E_p[x] = \int xp(x)dx = \int x\frac{p(x)}{q(x)}q(x)dx = E_q[x\omega]$$

Sample from proposal density $\{x^{(i)}\}_{i=1}^N \sim q(x)$ and weight the importance $\omega^{(i)} = p(x^{(i)})/q(x^{(i)})$ The estimation

$$E_p[x] = E_q[x\omega] \approx \sum_{i=1}^N x^{(i)}\omega^{(i)}$$

Where $p(x)$ is approximated by $\hat{p}(x) = \sum_{i=1}^N \delta(x - x^{(i)})\omega^{(i)}$
SIR (Sequential Importance Sampling with Resampling) filter is the simplest particle filter.

Initialization

\[ \{x_0^{(i)}\}_{i=1}^N \sim N(m_0, C_0) \]

The filtering process as follows,

\[ \{x_t^{(i)}, \frac{1}{N}\}_{i=1}^N \xrightarrow{\text{state prediction}} \{\hat{x}_{t+1}^{(i)}, \frac{1}{N}\}_{i=1}^N \xrightarrow{\text{weight update}} \{\hat{x}_{t+1}^{(i)}, \hat{\omega}_{t+1}^{(i)}\}_{i=1}^N \]

\[ \xrightarrow{\text{resampling}} \{x_{t+1}^{(i)}, \frac{1}{N}\}_{i=1}^N \]

Set proposal density to be \( p(x_{t+1}|x_t) \), so that

\[ \hat{x}_{t+1}^{(i)} = f(x_t^{(i)}) + \nu_t^{(i)} \]
From Bayes rule,

\[
p(x_{0:t+1}|y_{1:t+1}) = p(y_{t+1}|x_{t+1})p(x_{t+1}|x_t)p(x_{0:t}|y_{1:t})/p(y_{t+1}|y_{1:t})
\]

\[\propto p(y_{t+1}|x_{t+1})p(x_{t+1}|x_t)\]

Choose proposal to be transition density function

\[
q(x_{0:t+1}|y_{1:t+1}) = p(x_{t+1}|x_t)
\]

By importance sampling, the weight is

\[
\omega_{t+1} = \frac{p(x_{0:t+1}|y_{1:t+1})}{q(x_{0:t+1}|y_{1:t+1})} \propto p(y_{t+1}|x_{t+1})
\]

Given particles \(\{\hat{x}_{t+1}^{(i)}\}_{i=1}^{N}\) and data \(Y_{t+1}\),

\[
\omega_{t+1}^{(i)} \propto p(Y_{t+1}|\hat{x}_{t+1}^{(i)})
\]
Particle Degeneracy: the variance of weights increase over time [Kong and Liu, 1994]

Degeneracy can be measured by

\[ N_{ess} = \frac{N}{1 + \text{Var}(\omega_{t+1}^{(i)}, \text{true})} \]

Or \( \hat{N}_{ess} = \frac{1}{\sum_{i=1}^{N} (\omega_{t+1}^{(i)})^2} \)

- Resampling to reduce the effect of degeneracy.

\[ \{ \hat{x}_{t+1}^{(i)}, \hat{\omega}_{t+1}^{(i)} \}_{i=1}^{N} \xrightarrow{\text{resampling}} \{ \frac{1}{N} \}_{i=1}^{N} \]

- Monte Carlo methods require effective samples \( N_{ess} \to \infty \) to ensure the convergence to true distribution.

- Computational power is wasted on particles with zero weight.
1. **Goal of resampling**

\[
\{\hat{x}_{t+1}^{(i)}, \hat{\omega}_{t+1}^{(i)}\}_{i=1}^{N} \xrightarrow{\text{resampling}} \{x_{t+1}, \frac{1}{N}\}_{i=1}^{N}
\]

2. **Define**

\[
A = \sum_{i=1}^{N} \hat{\omega}_{t+1}^{(i)}
\]

3. **Generate** \(N\) random numbers \(\theta_k, k = 1, \ldots, N\) from uniform distribution \(U(0, 1)\)

4. **Choose**

\[
x_{t+1}^{(k)} = \hat{x}_{t+1}^{(i)}
\]

such that

\[
A^{-1} \sum_{j=1}^{i-1} \hat{\omega}_{t+1}^{(j)} < \theta_k \leq A^{-1} \sum_{j=1}^{i} \hat{\omega}_{t+1}^{(j)}
\]
Initialize particle \( \{x_0^{(i)}\}_{i=1}^N \sim N(m_0, C_0) \)

For \( t = 0 : T \)

1. For each particle \( i = 1 : N \)
   - State prediction:
     \[ \hat{x}_{t+1}^{(i)} = f(x_t^{(i)}) + v_t^{(i)} \]
     where \( v_t^{(i)} \sim N(0, Q) \)
   - Weight update:
     \[ \omega_{t+1}^{(i)} \propto \exp(-\frac{1}{2} |R^{-1/2}(Y_{t+1} - H(\hat{x}_{t+1}^{(i)}))|^2) \]

2. Resample.
Good choice of proposal density also reduce the effect of degeneracy.

Introduce an auxiliary variable $j$ which denote the $j$-th particle of time $t$.

By Bayes theorem,

$$p(x_{t+1}, j | y_{1:t+1}) \propto p(y_{t+1} | x_{t+1}) p(x_{t+1} | x_t^{(j)})$$

Define the proposal density close to the joint filtering density,

$$q(x_{t+1}, j | y_{1:t+1}) \propto p(y_{t+1} | \mu_{t+1}^{(j)}) p(x_{t+1} | x_t^{(j)})$$

where $\mu_{t+1}^{(j)} = E[x_{t+1} | x_t^{(j)}]$
Auxiliary Particle Filter

- Initialization \( \{x_0^{(i)}\}^N_{i=1} \sim N(m_0, C_0) \)

- Process to sample the auxiliary variable use present data \( Y_{t+1} \)

\[
\{x_t^{(i)}, \frac{1}{N}\}^N_{i=1} \xrightarrow{\text{look forward}} \{\mu_{t+1}^{(i)}, \frac{1}{N}\}^N_{i=1} \xrightarrow{\text{weight update}} \{\mu_{t+1}^{(i)}, \tilde{\omega}_{t}^{(i)}\}^N_{i=1}
\]

\[
\{x_t^{(i)}, \tilde{\omega}_{t}^{(i)}\}^N_{i=1} \xrightarrow{\text{resampling}} \{x_t^{(j)}, \frac{1}{N}\}^N_{j=1}
\]

- Filtering process

\[
\{x_t^{(j)}, \frac{1}{N}\}^N_{j=1} \xrightarrow{\text{prediction}} \{\hat{x}_{t+1}^{(j)}, \frac{1}{N}\}^N_{j=1}
\]

\[
\xrightarrow{\text{update}} \{\hat{x}_{t+1}^{(j)}, \omega_{t+1}^{(j)}\}^N_{j=1} \xrightarrow{\text{resampling}} \{x_{t+1}^{(j)}, \frac{1}{N}\}^N_{j=1}
\]
Look forward of state and use present data $Y_{t+1}$ to update the weight.

\[
\{x_t^{(i)}, \frac{1}{N}\}_{i=1}^N \xrightarrow{\text{look forward}} \{\mu_{t+1}^{(i)}, \frac{1}{N}\}_{i=1}^N \xrightarrow{\text{weight update}} \{\mu_{t+1}^{(i)}, \tilde{\omega}_t^{(i)}\}_{i=1}^N
\]

Use dynamic equation to make a prediction

\[
\mu_{t+1}^{(i)} = f(x_t^{(i)})
\]

The weight $\tilde{\omega}_t^{(i)}$ is obtained by combining the following two equations.

**Definition**:

\[
q(x_{t+1}, i | Y_{1:t+1}) \propto p(Y_{t+1} | \mu_{t+1}^{(i)})p(x_{t+1} | x_t^{(i)})
\]

**Conditional**:

\[
q(x_{t+1}, i | Y_{1:t+1}) = q(i | Y_{1:t+1})q(x_{t+1} | i, Y_{1:t+1}) = q(i | Y_{1:t+1})p(x_{t+1} | x_t^{(i)})
\]

Therefore,

\[
\tilde{\omega}_t^{(i)} = q(i | Y_{1:t+1}) \propto p(Y_{t+1} | \mu_{t+1}^{(i)})
\]
APF: filtering process

Filtering process:

\[
\{x_t^{(i)}, \tilde{\omega}_t^{(i)}\}_{i=1}^N \xrightarrow{\text{resampling}} \{x_t^{(j)}, \frac{1}{N}\}_{j=1}^N \xrightarrow{\text{prediction}} \{\hat{x}_{t+1}^{(j)}, \frac{1}{N}\}_{j=1}^N \\
\xrightarrow{\text{update}} \{\hat{x}_{t+1}^{(j)}, \omega_{t+1}^{(j)}\}_{j=1}^N \xrightarrow{\text{resampling}} \{x_{t+1}^{(j)}, \frac{1}{N}\}_{j=1}^N
\]

1. Resample to obtain particle \(\{x_t^{(j)}\}_{j=1}^N\)
2. Propagate in dynamic equation:
   \[
   \hat{x}_{t+1}^{(j)} = f(x_t^{(j)}) + v_t^{(j)}
   \]
3. Weight update
   \[
   \omega_{t+1}^{(j)} \propto \frac{p(Y_{t+1}|\hat{x}_{t+1}^{(j)})}{p(Y_{t+1}|\mu_{t+1}^{(j)})}
   \]
4. Resampling to eliminate the variance of weight.
Auxiliary Particle Filter (ASIR) Algorithm

Initialization \( \{x_0^{(i)}\}_{i=1}^N \sim N(m_0, C_0) \)

For \( t = 0 : T \)

1. Given \( \{x_t^{(i)}\}_{i=1}^N \), For \( i = 1 : N \)
   - State Prediction: \( \mu_t^{(i)} = f(x_t^{(i)}) \)
   - Weight Update: \( \tilde{\omega}_t^{(i)} = N(Y_{t+1} | H\mu_{t+1}^{(i)}, R) \)

2. Resampling for \( \{x_t^{(i)}, \tilde{\omega}_t^{(i)}\}_{i=1}^N \) to obtain \( \{x_t^{(j)}, 1/N\}_{j=1}^N \)

3. For each particle, given \( x_t^{(j)} \)
   - State prediction:
     \[
     \hat{x}_{t+1}^{(j)} = f(x_t^{(j)}) + v_t^{(j)}
     \]
     where \( v_t^{(j)} \sim N(0, Q) \)
   - Weight update: \( \omega_{t+1}^{(j)} = \frac{N(Y_{t+1} | H\hat{x}_{t+1}^{(j)}, R)}{N(Y_{t+1} | H\mu_{t+1}^{(j)}, R)} \)

4. Resample \( \{\hat{x}_{t+1}^{(j)}, \omega_{t+1}^{(j)}\}_{j=1}^N \) to obtain \( \{x_{t+1}^{(j)}, 1/N\}_{j=1}^N \)
Implicit Particle Filter (IPF)

- From the Bayes theorem,

\[ p(x_{0:t+1}|y_{1:t+1}) = p(y_{t+1}|x_{t+1})p(x_{t+1}|x_t)p(x_{0:t}|y_{1:t})/p(y_{t+1}|y_{1:t}) \]

- Given particle \( \{x^{(i)}_{0:t}, \frac{1}{N}\}_{i=1}^{N} \), and data \( Y_{1:t+1} \), we have a expression of \( x_{t+1} \)

\[ p(x^{(i)}_{0:t}, x_{t+1}|Y_{1:t+1}) \propto p(Y_{t+1}|x_{t+1})p(x_{t+1}|x^{(i)}_t) \]

- Objective: Sample \( x^{(i)}_{t+1} \) on high probability region of the posterior \( p(x^{(i)}_{0:t}, x_{t+1}|Y_{1:t+1}) \).
IPF: Obtain High Probability Particles

- General Idea: Rather than find particles from better proposal density and then estimate their probability, first pick a probability and then find a sample that carries it.

Given \( \{x_t^{(i)}\}_{i=1}^N \), sample \( \{x_{t+1}^{(i)}\}_{i=1}^N \) as follows:

1. Pick sample \( \xi_{t+1}^{(i)} \) from a known, fixed, pdf. e.g. a Gaussian \( N(0, I) \),
   \[
p(\xi) = \exp\left(-\frac{1}{2}\xi_{t+1}^T \xi_{t+1}\right)/(2\pi)^{n_x/2}
   \]
2. Write the posterior as
   \[
p(Y_{t+1}|x_{t+1}^{(i)})p(x_{t+1}^{(i)}|x_t^{(i)}) \text{ in the form } exp(-F_{t+1}^{(i)}(x_{t+1}^{(i)}))
   \]
3. To obtain high probability particles, solve
   \[
   F_{t+1}^{(i)}(x_{t+1}^{(i)}) - \min F_{t+1}^{(i)}(x_{t+1}^{(i)}) = \frac{1}{2}((\xi_{t+1}^{(i)})^T \xi_{t+1}^{(i)})
   \]

- The right pdf is sampled if map \( \xi_{t+1}^{(i)} \to x_{t+1}^{(i)} \) is one-to-one and onto.
IPF: Quadratic Approximation of $F_{t+1}^{(i)}$

With nonlinear dynamics and linear observation in the model

- Quadratic approximation of $F_{t+1}^{(i)}$ is a formula of $x_{t+1}^{(i)}$

  $$F_{t+1}^{(i)}(x_{t+1}^{(i)}) = \frac{1}{2}(x_{t+1}^{(i)} - f(x_t^{(i)}))^T Q^{-1}(x_{t+1}^{(i)} - f(x_t^{(i)}))$$

  $$+ \frac{1}{2}(y_{t+1} - Hx_{t+1}^{(i)})^T R^{-1}(y_{t+1} - Hx_{t+1}^{(i)})$$

- Completing the square,

  $$F_{t+1}^{(i)}(x_{t+1}^{(i)}) = \frac{1}{2}(x_{t+1}^{(i)} - \bar{m}_{t+1}^{(i)})^T \Sigma^{-1}(x_{t+1}^{(i)} - \bar{m}_{t+1}^{(i)}) + \phi_{t+1}^{(i)}$$

where

- $\Sigma^{-1} = (Q^T Q)^{-1} + H^T (R^T R) H$
- $\bar{m}_{t+1}^{(i)} = \Sigma((Q^T Q)^{-1} x_t^{(i)}) + H(Q^T Q)^{-1} Y_{t+1}$
- $\phi_{t+1}^{(i)} = \frac{1}{2}(Y_{t+1} - Hf(x_{t+1}^{(i)}))^T (HQ^T QH^T + R^T R)^{-1}(Y_{t+1} - Hf(x_{t+1}^{(i)}))$
From

\[ F_{t+1}^{(i)}(x_{t+1}^{(i)}) = \frac{1}{2}(x_{t+1}^{(i)} - \bar{m}_{t+1}^{(i)})^T \Sigma^{-1}(x_{t+1}^{(i)} - \bar{m}_{t+1}^{(i)}) + \phi_{t+1}^{(i)} \]

It follows that

\[ \min F_{t+1}^{(i)}(x_{t+1}^{(i)}) = \phi_{t+1}^{(i)} \]

Therefore

\[ \frac{1}{2}(x_{t+1}^{(i)} - \bar{m}_{t+1}^{(i)})^T \Sigma^{-1}(x_{t+1}^{(i)} - \bar{m}_{t+1}^{(i)}) = \frac{1}{2} \xi_{t+1}^{(i)} T \xi_{t+1}^{(i)} \]
IPF: Solving the Underdetermined Equation

- One equation $n_x$ unknowns
  \[
  \frac{1}{2}(x^{(i)}_{t+1} - \bar{m}^{(i)}_{t+1})^T \Sigma^{-1}(x^{(i)}_{t+1} - \bar{m}^{(i)}_{t+1}) = \frac{1}{2}\xi^{(i)}_{t+1} \xi^{(i)}_{t+1}
  \]

- A one-to-one mapping $\xi \to x$ is
  \[
  L^{-1}(x^{(i)}_{t+1} - \bar{m}^{(i)}_{t+1}) = \xi^{(i)}_{t+1} \text{ Or } x^{(i)}_{t+1} = \bar{m}^{(i)}_{t+1} + L\xi^{(i)}_{t+1}
  \]
  where $\Sigma = LL^T$ is Cholesky decomposition.
The probability density of reference variable $\xi$ is Gaussian by our choice.

The proposal density is,

$$q(x_{t+1}^{(i)}) = \frac{p(\xi_{t+1}^{(i)})}{J} \propto \exp\left(-\frac{1}{2} \xi_{t+1}^{(i) T} \xi_{t+1}^{(i)}\right)$$

$$= \frac{\exp(\phi_{t+1}^{(i)} - F(x_{t+1}^{(i)}))}{J} = \frac{\exp(\phi_{t+1}^{(i)})}{J} p(Y_{t+1}|x_{t+1}^{(i)}) p(x_{t+1}^{(i)}|x_{t}^{(i)})$$

where $J$ is the determinant of Jacobian matrix.

From the mapping

$$x_{t+1}^{(i)} = \bar{m}_{t+1}^{(i)} + L\xi_{t+1}^{(i)}$$

We have

$$J = | \det(\frac{\partial x_{t+1}^{(i)}}{\partial \xi^{(i)}})| = | \det L |$$
By importance sampling,

\[ \omega_{t+1}^{(i)} = \frac{p(x_{0:t+1}^{(i)}|Y_{1:t+1})}{q(x_{t+1}^{(i)})} \propto \exp(\phi_{t+1}^{(i)}) p(Y_{t+1}|x_{t+1}^{(i)}) p(x_{t+1}^{(i)}|x_t^{(i)}) \exp(-\phi_{t+1}^{(i)}) |\det L| \]
IPF: Algorithm

Initialization $\{x_0^{(i)}\}_{i=1}^N \sim N(m_0, C_0)$

Compute $L$ by Cholesky decomposition $\Sigma = LL^T$

For $t = 0 : T$

1. Given $\{x_t^{(i)}\}_{i=1}^N$, For $i = 1 : N$
   - Generate $\xi^{(i)} \sim N(0, I)$
   - Compute $\tilde{m}^{(i)}, \phi_{t+1}, |\det L|$
   - State Prediction: $\hat{x}_{t+1}^{(i)} = \tilde{m}^{(i)} + L\xi^{(i)}$
   - Weight Update: $\omega_{t+1}^{(i)} = \exp(-\phi_{t+1}^{(i)})|\det L|$

2. Resampling for $\{\hat{x}_{t+1}^{(i)}, \omega_{t+1}^{(i)}\}_{i=1}^N$ to obtain $\{x_{t+1}^{(i)}, 1/N\}_{i=1}^N$
Compare the performance of filtering algorithms with same sample size,

- The number of times steps for the estimation to track 'on target' from initial guess.
- The accuracy of estimation at certain time step, e.g. $t = 100$. 
One dimensional particle moving in the potential

\[ V(x) = \alpha(x^2 - 1)^2 \]

With the force

\[ -\nabla V(x) = 4\alpha(x - x^3) \]

The resulting SDE

\[ \frac{dx}{dt} = 4\alpha(x - x^3) + u, \ u \sim N(0, q) \]
Discretization in time by Euler scheme.

\[ x_{t+1} = x_t + 4\alpha(x_t - x_t^3)\delta_t + v_t, \quad v_t \sim N(0, q\delta_t) \]

Define the linear measurement equation

\[ y_{t+1} = x_{t+1} + w, \quad w \sim N(0, r) \]

Set initial state \( x_0 \sim N(0, \sigma_0) \)

\( \alpha = 2.5, \delta_t = 0.02, q = 0.3, r = 0.1, \sigma_0 = 10 \)
Experiment: Convergence in Time

**Ensemble Kalman Filter**

**Auxiliary Particle Filter**

**Particle Filter**

**Implicit Particle Filter**
Accuracy of Estimation

- Compare the RMSE at $t = 100$ by simulating the process 1000 times with particle size $N = 20$

$$RMSE = \sqrt{\frac{1}{1000} \sum_{j=1}^{1000} (x_{100}^{true,j} - m_{100}^j)^2}, \text{where } m_{100} = \sum_{i=1}^{20} x_{100}^{(i)}$$

- Average effective sample size is

$$\text{Average } N_{ess} = \frac{1}{1000} \sum_{j=1}^{1000} N_{ess}^j, \text{where } N_{ess} = \frac{1}{\sum_{i=1}^{20} (\omega_{t+1}^{(i)})^2}$$

<table>
<thead>
<tr>
<th></th>
<th>RMSE</th>
<th>average effective sample size</th>
</tr>
</thead>
<tbody>
<tr>
<td>EnKF</td>
<td>0.093</td>
<td>20</td>
</tr>
<tr>
<td>SIR</td>
<td>0.064</td>
<td>14.73</td>
</tr>
<tr>
<td>APF</td>
<td>0.051</td>
<td>19.19</td>
</tr>
<tr>
<td>IPF</td>
<td>0.048</td>
<td>18.33</td>
</tr>
</tbody>
</table>
Conclusion: reviewed five filtering algorithms for state estimation.

Future Work:
- Apply the idea of implicit sampling to more sophisticated problems (high dimensional, nonlinear observations, non-Gaussian noise)
- Derive filtering methods for both parameter and state estimation.
[7] Adam M. Johansen, Arnaud Doucet, A Note on Auxiliary Particle Filters