Here we have:

\[ P(X_{t+1}, Y_{t+1} | Y_1, \ldots, Y_t) \sim N \left( \begin{bmatrix} \hat{X}_{t+1} \\ \hat{C}_{t+1} \hat{X}_{t+1} \end{bmatrix}, \begin{bmatrix} P_{t+1|t} & P_{t+1|t} C^T \\ C P_{t+1|t} & C P_{t+1|t} C^T + R \end{bmatrix} \right) \]

\[
\begin{align*}
\hat{X}_{t+1} &= \hat{X}_{t+1|t} + P_{t+1|t} C^T (C P_{t+1|t} C^T + R)^{-1} (Y_t - C \hat{X}_{t+1|t}) \\
P_{t+1|t+1} &= P_{t+1|t} - P_{t+1|t} C^T (C P_{t+1|t} C^T + R)^{-1} C P_{t+1|t}
\end{align*}
\]

Kalman Gain

Measurement (or dynamics) may not be linear:

Observation of 3D particle via 2D pinhole camera:

\[
\begin{align*}
X_c &= \frac{a_1 X + b_2 Y + c_3 Z}{a_3 X + b_2 Y + c_3 Z} \\
Y_c &= \frac{a_2 X + b_2 Y + c_3 Z}{a_2 X + b_2 Y + c_3 Z}
\end{align*}
\]

"Perspective Transform"

\( (X_c, Y_c) \) = particle's location
\( (X_c, Y_c) \) = pixel coordinate
The noise is still assumed to be Gaussian. Use Extended Kalman Filter (EKF) to represent non-linearity.

Before: \[ y_t = Cx_t + r_t \] Jacobian evaluated at \( \hat{x}_{t|t-1} \).

Now: \[ y_t = h(x_t) + r_t \]
\[ \approx h(\hat{x}_{t|t-1}) + D_h(\hat{x}_{t|t-1})(x_t - \hat{x}_{t|t-1}) + r_t \]
\[ \mathbb{E}(x_t|y_{1:t-1}, y_t) \]
\[ \tilde{y}_t = y_t - h(\hat{x}_{t|t-1}) + D_h(\hat{x}_{t|t-1}) \hat{x}_{t|t-1} \]

New measurement equation is \( \tilde{y}_t = D_h(\hat{x}_{t|t-1})x_t + r_t \)

Apply KF with \( C \) replaced by \( D_h(\hat{x}_{t|t-1}) \).

Another popular scheme: Unscented Kalman Filter (UKF).

More general technique: Particle filter which is an example of a general class of techniques called sampling.

In general, two reasons why you need an approximation algorithm:
1. Integration is too hard.
2. Graph complexity (clique size)

Two types of approximations:
1. Variational approach – drop belief propagation
2. Sampling (Monte Carlo)

Generate \( x_1, x_2, \ldots, x_N \sim N(0, \sigma^2) \) from \( U_1, U_2, \ldots, U_N \sim U(1)

Algorithm:
\[ x = \frac{1}{\sigma^2} \left( \frac{-2 \ln U}{} \right)^{1/2} \sim N(0, \sigma^2) \]
Target problem: \( \mathbb{E}_p[f(x)] = \int p(x) f(x) \, dx \)

Most single-minded sampling technique \( \{ \)
\[ \hat{\mathbb{E}}_p[f(x)] = \frac{1}{N} \sum_{i=1}^{N} f(X_i) \xrightarrow{\text{CLT}} \mathbb{E}_p[f(x)] \]
\[ \text{Var}[\hat{\mathbb{E}}_p] = \frac{\text{Var}(f(x))}{M} \]

Rejection sampling:

Assumption 1: \( p(x) \) is hard to sample but computable.
Assumption 2: easy to sample \( q(x) \) "proposal distribution" \( q(x) M \geq p(x) \)

1. Sample \( x \) with \( q(x) \)
2. Sample \( u \sim \text{uniform}[0, Mq(x)] \)
3. Accept \( x \) as a sample if \( u < p(x) \)

\[ \Rightarrow \text{sucks (exponentially bad) when scaled up to high dimension for complicated } p(x) \]

Importance Sampling:

\[ \mathbb{E}_p(f(x)) = \int f(x) p(x) \, dx \]
\[ = \int f(x) \frac{p(x)}{q(x)} q(x) \, dx \]
\[ = \mathbb{E}_q(f(x) \cdot \frac{p(x)}{q(x)}) \]

Sample: \( x_1, x_2, \ldots, x_N \sim q(x) \)

Evaluate: \( f(x_1), f(x_2), \ldots, f(x_N) \)

Re-weight: \( w_i = \frac{p(x_i)}{q(x_i)} \)

\[ \hat{E}_p(f(x)) = \sum_{i=1}^{N} w_i f(x_i) \]
\[ = \frac{\sum_{i=1}^{N} w_i f(x_i)}{\sum_{i=1}^{N} w_i} \]

if \( p(x), q(x) \) are un-normalized.

This forms the basis of particle filtering:

In PF, dynamics/measurement can be non-linear (calculable)
noise can be non-Gaussian

\[ p(x_t | y_1, \ldots, y_t) \propto (x_i, w_i) \quad i = 1, \ldots, N \]

Time update: \[ p(x_{t+1} | y_1, \ldots, y_t) = \int p(x_{t+1} | x_t) p(x_t | y_1, \ldots, y_t) \, dx_t \]
Prediction: \[ p(y_{t+1} | x_1, \ldots, x_t) = \int p(y_{t+1} | x_{t+1}) p(x_{t+1} | y_1, \ldots, y_t) \, dx_{t+1} \]
Measurement: \[ p(x_{t+1} | y_1, \ldots, y_{t+1}) \propto p(y_{t+1} | x_{t+1}) p(x_{t+1} | y_1, \ldots, y_t) \]

Need the "samples & weights" representation of
\[ p(x_{t+1} | y_1, \ldots, y_t) \] "convolution" and \[ p(x_{t+1} | y_1, \ldots, y_{t+1}) \] "multiplication"
"Multiplication":

\[
\begin{align*}
\text{start } (x_i, w_i)_{i=1,\ldots,n} \text{ represent } p(x) \\
\text{Find samples } X_i \text{ and weights such that } p'(x) \propto p(x) p(y) \\
X_i \Rightarrow X'_i \\
w_i \Rightarrow w'_i = B(x_i) \cdot w_i \\
\frac{p(x)}{q(x)} \propto \frac{p'(x)}{q(x)} \propto \frac{p(x) p(y)}{q(x)}
\end{align*}
\]

"Convolution":

\[
\text{Sample from uniform } \ast \uparrow \uparrow = \text{ samples}
\]

Every particle to have the same weights more concentrated at high probability

1. Resampling \( (x_i, w_i)_{i=1,\ldots,n} \) to a new set of particles \( (x'_i, w'_i)_{i=1,\ldots,n} \) based on uniform sampling on the following line.

\[
\begin{align*}
U_i \Rightarrow x'_i = x_i \\
U_i \downarrow \quad w_i \downarrow \quad w'_i \downarrow \quad w_n
\end{align*}
\]

2. \( q(y) = \int f(y|x) p(x) \, dx \Rightarrow q(y) \sim (x_i, \frac{1}{M} \sum_{i=1}^{M} f(y, x'_i)) \)