Examples

Eugene Morozov

Contents

1 Kalman Filter

1.1 Least Mean-Square Error approach

Let's consider a linear, discrete-time dynamical system with its process equation:

$$\mathbf{x}_{k+1} = \mathbf{F}_{k+1,k}\mathbf{x}_k + \mathbf{w}_k, \tag{1}$$

where $\mathbf{F}_{k+1,k}$ is the transition matrix taking the (often hidden or latent) state $\mathbf{x}_k$ from time k to time $k+1$. The process noise $\mathbf{w}_k$ is assumed to be additive, white, and Gaussian, with zero mean and with covariance matrix defined by

$$E[\mathbf{w}_n\mathbf{w}_k^T] = \begin{cases} \mathbf{Q}_k & for\; n = k \\ \mathbf{0} & for\; n \neq k, \end{cases} \tag{2}$$

(therefore it's a Markov sequence.)

Measurement equation:

$$\mathbf{y}_k = \mathbf{H}_k\mathbf{x}_k + \mathbf{v}_k, \tag{3}$$

where $\mathbf{y}_k$ is the observable variable at time k and $\mathbf{H}_k$ is the measurement matrix. The measurement noise $\mathbf{v}_k$ is assumed to be additive, white, and Gaussian, with zero mean and with covariance matrix defined by

$$E[\mathbf{v}_n\mathbf{v}_k^T] = \begin{cases} \mathbf{R}_k & for\, n = k \\ \mathbf{0} & for\, n \neq k. \end{cases} \tag{4}$$

Moreover, the measurement noise $\mathbf{v}_k$ is uncorrelated with the process noise $\mathbf{w}_k$, i.e. $cov(\mathbf{w}_i, \mathbf{v}_j) = 0$. Also $cov(\mathbf{x}_0, \mathbf{x}_0) \equiv var(\mathbf{x}_0) = \mathbf{P}_0$, $cov(\mathbf{x}_0, \mathbf{w}_k) = cov(\mathbf{x}_0, \mathbf{v}_k) = 0$ *for all k*.

The goal is to find the minimum mean-square error estimate of $\mathbf{x}_k$ given $\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_k$. Let $\hat{\mathbf{x}}_k$ denote the a posteriori estimate of the signal $\mathbf{x}_k$, given the observations $\mathbf{y}_i$, $i = 1, \ldots, k$. The state-error vector is defined by

$$\tilde{\mathbf{x}}_k \equiv \mathbf{x}_k - \hat{\mathbf{x}}_k. \tag{5}$$

In addition to having zero mean, it has covariance $\mathbf{P}_k$.

Let's define a cost (loss) function for incorrect estimates:

- The cost function is non-negative.
- The cost function is a non-decreasing function of the estimation error $\tilde{\mathbf{x}}_k$.

These two requirements are satisfied by the mean-square error defined by

$$\mathbf{J}_k = E[(\mathbf{x}_k - \hat{\mathbf{x}}_k)^2] = E[\tilde{\mathbf{x}}_k^2] \tag{6}$$

To derive an optimal value for the estimate $\hat{\mathbf{x}}_k$, we may invoke two theorems taken from stochastic process theory:

Theorem 1.1 Conditional mean estimator. *If the stochastic processes $\mathbf{x}_k$ and $\mathbf{y}_k$ are jointly Gaussian, then the optimum estimate $\hat{\mathbf{x}}_k$ that minimizes the mean-square error $\mathbf{J}_k$ is the conditional mean estimator:*

$$\hat{\mathbf{x}}_k = E[\mathbf{x}_k | \mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_k].$$

Theorem 1.2 Principle of orthogonality. *Let the stochastic processes $\mathbf{x}_k$ and $\mathbf{y}_k$ be of zero means; that is, $E[\mathbf{x}_k] = E[\mathbf{y}_k] = 0$ for all k.*

Then:

(i) the stochastic processes $\mathbf{x}_k$ and $\mathbf{y}_k$ are jointly Gaussian; or

(ii) if the optimal estimate $\hat{\mathbf{x}}_k$ is restricted to be a linear function of the observables and the cost function is the mean-square error,

(iii) then the optimum estimate $\hat{\mathbf{x}}_k$, given the observables $\mathbf{y}_k$, is the orthogonal projection of $\mathbf{x}_k$ on the space spanned by these observables.

Suppose that a measurement on a linear dynamical system, described by Eqs. (1) and (3), has been made at time k. The requirement is to use the information contained in the new measurement $\mathbf{y}_k$ to update the estimate of the unknown state $\mathbf{x}_k$. Let $\hat{\mathbf{x}}_k^-$ denote a priori estimate of the state, which is already available at time k. With a linear estimator as the objective, we may express the a posteriori estimate $\hat{\mathbf{x}}_k$ as a linear combination of the a priori estimate and the new measurement, as shown by

$$\hat{\mathbf{x}}_k = \mathbf{G}_k^* \hat{\mathbf{x}}_k^- + \mathbf{G}_k \mathbf{y}_k, \tag{7}$$

where the multiplying matrix factors $\mathbf{G}_k^*$ and $\mathbf{G}_k$ are to be determined. To find these two matrices, we invoke the principle of orthogonality stated under Theorem 1.2:

$$E[\tilde{\mathbf{x}}_k \mathbf{y}_i^T] = 0 \; for \; i = 1, 2, \dots, k-1 \tag{8}$$

Using (3), (7), and (5), (8), we get

$$E[(\mathbf{x}_k - \mathbf{G}_k^* \hat{\mathbf{x}}_k^- - \mathbf{G}_k \mathbf{H}_k \mathbf{x}_k - \mathbf{G}_k \mathbf{v}_k) \mathbf{y}_i^T] = 0 \; for \; i = 1, 2, \dots, k-1 \tag{9}$$

The noise is uncorrelated: $E[\mathbf{v}_k \mathbf{y}_i^T] = 0 \; for \; i = 1, 2, \dots, k-1$
since $E[\mathbf{v}_k (\mathbf{H}_i \mathbf{x}_i + \mathbf{v}_i)^T] = \mathbf{H}_i^T E[\mathbf{v}_k \mathbf{x}_i^T] = \mathbf{H}_i^T E[\mathbf{v}_k (\mathbf{F}_{i-1} \mathbf{x}_{i-1} + \mathbf{w}_{i-1})^T] = 0$
Therefore

$$E[(\mathbf{I} - \mathbf{G}_k \mathbf{H}_k - \mathbf{G}_k^*) \mathbf{x}_k \mathbf{y}_i^T + \mathbf{G}_k^* (\mathbf{x}_k - \hat{\mathbf{x}}_k^-) \mathbf{y}_i^T] = 0 \tag{10}$$

From the principle of orthogonality (both $\hat{\mathbf{x}}_k^-$ and $\hat{\mathbf{x}}_k$ are unbiased estimations) we now note that

$$E[(\mathbf{x}_k - \hat{\mathbf{x}}_k^-) \mathbf{y}_i^T] = 0 \tag{11}$$

Accordingly, (10) simplifies to

$$(\mathbf{I} - \mathbf{G}_k \mathbf{H}_k - \mathbf{G}_k^*) E[\mathbf{x}_k \mathbf{y}_i^T] = 0 \; for \; i = 1, 2, \dots, k-1 \tag{12}$$

For arbitrary values of the state $\mathbf{x}_k$ and the observable $\mathbf{y}_i$,(12) can only be satisfied if the scaling factors $\mathbf{G}_k$ and $\mathbf{G}_k^*$ are related as follows:

$$\mathbf{G}_k^* = \mathbf{I} - \mathbf{G}_k \mathbf{H}_k \tag{13}$$

Substituting (13) into (7), we may express the a posteriori estimate of the state at time k as

$$\hat{\mathbf{x}}_k = (\mathbf{I} - \mathbf{G}_k \mathbf{H}_k) \hat{\mathbf{x}}_k^- + \mathbf{G}_k \mathbf{y}_k \tag{14}$$

$$= \hat{\mathbf{x}}_k^- + \mathbf{G}_k (\mathbf{y}_k - \mathbf{H}_k \hat{\mathbf{x}}_k^-), \tag{15}$$

in light of which, the matrix $\mathbf{G}_k$ is called the Kalman gain.

Now let's derive an explicit formula for $\mathbf{G}_k$. Again, from the principle of orthogonality, similar to (11), we have

$$E[(\mathbf{x}_k - \hat{\mathbf{x}}_k) \mathbf{y}_k^T] = 0 \tag{16}$$

Let the error (innovation, i.e. a measure of the“new” information contained in $\mathbf{y}_k$)

$$\tilde{\mathbf{y}}_k = \hat{\mathbf{y}}_k^- - \mathbf{y}_k \tag{17}$$

$$= \mathbf{H}_k \hat{\mathbf{x}}_k^- - \mathbf{y}_k \tag{18}$$

The parameter $\hat{\mathbf{x}}_k$ depends linearly on $\mathbf{x}_k$, which depends linearly on $\mathbf{y}_k$. Therefore, from (16)

$$E[(\mathbf{x}_k - \hat{\mathbf{x}}_k)\hat{\mathbf{y}}_k^{T-}] = 0 \tag{19}$$

and also (by subtracting (16) from (19))

$$E[(\mathbf{x}_k - \hat{\mathbf{x}}_k)\tilde{\mathbf{y}}_k^T] = 0 \tag{20}$$

Using (3) and (15), we may express the state-error vector $\mathbf{x}_k - \hat{\mathbf{x}}_k$ as

$$\mathbf{x}_k - \hat{\mathbf{x}}_k = \mathbf{x}_k - \hat{\mathbf{x}}_k^- - \mathbf{G}_k(\mathbf{y}_k - \mathbf{H}_k \hat{\mathbf{x}}_k^-)$$

$$= \mathbf{x}_k - \hat{\mathbf{x}}_k^- - \mathbf{G}_k(\mathbf{H}_k \mathbf{x}_k + \mathbf{v}_k - \mathbf{H}_k \hat{\mathbf{x}}_k^-)$$

$$= \mathbf{x}_k - \mathbf{G}_k \mathbf{H}_k \mathbf{x}_k - \hat{\mathbf{x}}_k^- + \mathbf{G}_k \mathbf{H}_k \hat{\mathbf{x}}_k^- - \mathbf{G}_k \mathbf{v}_k$$

$$= (\mathbf{I} - \mathbf{G}_k \mathbf{H}_k)\mathbf{x}_k - (\mathbf{I} - \mathbf{G}_k \mathbf{H}_k)\hat{\mathbf{x}}_k^- - \mathbf{G}_k \mathbf{v}_k$$

$$= (\mathbf{I} - \mathbf{G}_k \mathbf{H}_k)\tilde{\mathbf{x}}_k^- - \mathbf{G}_k \mathbf{v}_k \tag{21}$$

Hence, substituting (18) and (21) into (20), we get

$$E[((\mathbf{I} - \mathbf{G}_k \mathbf{H}_k)\tilde{\mathbf{x}}_k^- - \mathbf{G}_k \mathbf{v}_k)(\mathbf{H}_k \hat{\mathbf{x}}_k^- - \mathbf{y}_k)^T] = 0$$

$$= E[((\mathbf{I} - \mathbf{G}_k \mathbf{H}_k)\tilde{\mathbf{x}}_k^- - \mathbf{G}_k \mathbf{v}_k)(\mathbf{H}_k \hat{\mathbf{x}}_k^- - \mathbf{H}_k \mathbf{x}_k - \mathbf{v}_k)^T]$$

$$= E[((\mathbf{I} - \mathbf{G}_k \mathbf{H}_k)\tilde{\mathbf{x}}_k^- - \mathbf{G}_k \mathbf{v}_k)(\mathbf{H}_k \tilde{\mathbf{x}}_k^- + \mathbf{v}_k)^T] = 0 \tag{22}$$

Since the measurement noise $\mathbf{v}_k$ is independent of the state $\mathbf{x}_k$ and therefore the error $\tilde{\mathbf{x}}_k^-$, the expectation of (22) reduces to

$$(\mathbf{I} - \mathbf{G}_k \mathbf{H}_k)E[\tilde{\mathbf{x}}_k^- - \tilde{\mathbf{x}}_k^{T-}]\mathbf{H}_k^T - \mathbf{G}_k E[\mathbf{v}_k \mathbf{v}_k^T] = 0 \tag{23}$$

Define the a priori covariance matrix

$$\mathbf{P}_k^- = E[(\mathbf{x}_k - \hat{\mathbf{x}}_k^-)(\mathbf{x}_k - \hat{\mathbf{x}}_k^-)^T]$$

$$= E[\tilde{\mathbf{x}}_k^- \tilde{\mathbf{x}}_k^{T-}] \tag{24}$$

Then we rewrite (23) as

$$(\mathbf{I} - \mathbf{G}_k \mathbf{H}_k)\mathbf{P}_k^- \mathbf{H}_k^T - \mathbf{G}_k \mathbf{R}_k = 0 \tag{25}$$

Solving this equation for $\mathbf{G}_k$, we get the desired formula

$$\mathbf{P}_k^- \mathbf{H}_k^T - \mathbf{G}_k \mathbf{H}_k \mathbf{P}_k^- \mathbf{H}_k^T - \mathbf{G}_k \mathbf{R}_k = 0$$

$$\mathbf{G}_k(\mathbf{H}_k \mathbf{P}_k^- \mathbf{H}_k^T + \mathbf{R}_k) = \mathbf{P}_k^- \mathbf{H}_k^T$$

$$\mathbf{G}_k = \mathbf{P}_k^- \mathbf{H}_k^T (\mathbf{H}_k \mathbf{P}_k^- \mathbf{H}_k^T + \mathbf{R}_k)^{-1} \tag{26}$$

Similar for the a posteriori covariance

$$\mathbf{P}_k = E[\tilde{\mathbf{x}}_k \tilde{\mathbf{x}}_k^T] \tag{27}$$

By substituting (13) into (7), we obtain

$$\hat{\mathbf{x}}_k = (\mathbf{I} - \mathbf{G}_k \mathbf{H}_k)\hat{\mathbf{x}}_k^- + \mathbf{G}_k \mathbf{y}_k$$

$$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k^- + \mathbf{G}_k(\mathbf{y}_k - \mathbf{H}_k \hat{\mathbf{x}}_k^-) \tag{28}$$

Subtract $\mathbf{x}_k$ from both sides of the latter equation to obtain

$$\hat{\mathbf{x}}_k - \mathbf{x}_k = \hat{\mathbf{x}}_k^- + \mathbf{G}_k \mathbf{H}_k \mathbf{x}_k + \mathbf{G}_k \mathbf{v}_k - \mathbf{G}_k \mathbf{H}_k \hat{\mathbf{x}}_k^- - \mathbf{x}_k$$

$$\tilde{\mathbf{x}}_k = \tilde{\mathbf{x}}_k^- - \mathbf{G}_k \mathbf{H}_k \tilde{\mathbf{x}}_k^- - \mathbf{G}_k \mathbf{v}_k$$

$$\tilde{\mathbf{x}}_k = (\mathbf{I} - \mathbf{G}_k \mathbf{H}_k)\tilde{\mathbf{x}}_k^- - \mathbf{G}_k \mathbf{v}_k \tag{29}$$

By substituting (29) into (27) and noting that $E[\tilde{\mathbf{x}}_k^- \mathbf{v}_k^T] = 0$, we obtain

$$\mathbf{P}_k = E[((\mathbf{I} - \mathbf{G}_k \mathbf{H}_k)\tilde{\mathbf{x}}_k^- - \mathbf{G}_k \mathbf{v}_k)((\mathbf{I} - \mathbf{G}_k \mathbf{H}_k)\tilde{\mathbf{x}}_k^- - \mathbf{G}_k \mathbf{v}_k)^T]$$

$$= (\mathbf{I} - \mathbf{G}_k \mathbf{H}_k)E[\tilde{\mathbf{x}}_k^- \tilde{\mathbf{x}}_k^{T-}](\mathbf{I} - \mathbf{G}_k \mathbf{H}_k)^T + \mathbf{G}_k E[\mathbf{v}_k \mathbf{v}_k^T]\mathbf{G}_k^T$$

$$= (\mathbf{I} - \mathbf{G}_k \mathbf{H}_k)\mathbf{P}_k^- (\mathbf{I} - \mathbf{G}_k \mathbf{H}_k)^T + \mathbf{G}_k \mathbf{R}_k \mathbf{G}_k^T \tag{30}$$

This is so-called “Joseph form” of the covariance update equation. By substituting for $\mathbf{G}_k$ from (26), it can be put into the following form

$$\mathbf{P}_k = (\mathbf{P}_k^- - \mathbf{G}_k \mathbf{H}_k \mathbf{P}_k^-)(\mathbf{I} - \mathbf{G}_k \mathbf{H}_k)^T + \mathbf{G}_k \mathbf{R}_k \mathbf{G}_k^T$$

$$= \mathbf{P}_k^- - \mathbf{G}_k\mathbf{H}_k\mathbf{P}_k^- - \mathbf{P}_k^-\mathbf{H}_k^T\mathbf{G}_k^T + \mathbf{G}_k\mathbf{H}_k\mathbf{P}_k^-\mathbf{H}_k^T\mathbf{G}_k^T + \mathbf{G}_k\mathbf{R}_k\mathbf{G}_k^T$$

$$= (\mathbf{I} - \mathbf{G}_k\mathbf{H}_k)\mathbf{P}_k^- - \mathbf{P}_k^-\mathbf{H}_k^T\mathbf{G}_k^T + \mathbf{G}_k(\mathbf{H}_k\mathbf{P}_k^-\mathbf{H}_k^T + \mathbf{R}_k)\mathbf{G}_k^T \tag{31}$$

From (25)

$$\mathbf{G}_k\mathbf{H}_k\mathbf{P}_k^-\mathbf{H}_k^T = \mathbf{P}_k^-\mathbf{H}_k^T - \mathbf{G}_k\mathbf{R}_k$$

Substituting this into (31)

$$\mathbf{P}_k = (\mathbf{I} - \mathbf{G}_k\mathbf{H}_k)\mathbf{P}_k^- - \mathbf{P}_k^-\mathbf{H}_k^T\mathbf{G}_k^T + (\mathbf{P}_k^-\mathbf{H}_k^T - \mathbf{G}_k\mathbf{R}_k + \mathbf{G}_k\mathbf{R}_k)\mathbf{G}_k^T$$

$$= (\mathbf{I} - \mathbf{G}_k\mathbf{H}_k)\mathbf{P}_k^- - \mathbf{P}_k^-\mathbf{H}_k^T\mathbf{G}_k^T + \mathbf{P}_k^-\mathbf{H}_k^T\mathbf{G}_k^T$$

$$= (\mathbf{I} - \mathbf{G}_k\mathbf{H}_k)\mathbf{P}_k^- \tag{32}$$

This is the one most often used in computation. It implements the effect that *conditioning on the measurement* has on the covariance matrix of estimation uncertainty.

Note: for better numerical stability (to preserve the positive definiteness) use the representation via the square root matrix.

Let's see how the covariance changes in time.

$$\hat{\mathbf{x}}_k^- = \mathbf{F}_{k-1}\hat{\mathbf{x}}_{k-1}$$

For notational simplicity let $\hat{\mathbf{x}}_{k-1} \equiv \hat{\mathbf{x}}_{k-1}^+$ and $\mathbf{P}_{k-1} \equiv \mathbf{P}_{k-1}^+$.

Subtracting $\mathbf{x}_k$ from both sides to obtain

$$\hat{\mathbf{x}}_k^- - \mathbf{x}_k = \mathbf{F}_{k-1}\hat{\mathbf{x}}_{k-1} - \mathbf{x}_k$$

$$\tilde{\mathbf{x}}_k^- = \mathbf{F}_{k-1}(\hat{\mathbf{x}}_{k-1} - \mathbf{x}_{k-1}) - \mathbf{w}_{k-1}$$

$$= \mathbf{F}_{k-1}\tilde{\mathbf{x}}_{k-1} - \mathbf{w}_{k-1} \tag{33}$$

for the propagation of the estimation error $\tilde{\mathbf{x}}$. Post-multiply it by $\tilde{\mathbf{x}}_k^{T-}$ and take the expected values. Use the fact that $E[\tilde{\mathbf{x}}_{k-1}\mathbf{w}_{k-1}^T] = 0$ to obtain results

$$E[\tilde{\mathbf{x}}_k^-\tilde{\mathbf{x}}_k^{T-}] = \mathbf{P}_k^-$$

$$= \mathbf{F}_{k-1}E[\tilde{\mathbf{x}}_{k-1}\tilde{\mathbf{x}}_k^{T-}]$$

$$= \mathbf{F}_{k-1}E[\tilde{\mathbf{x}}_{k-1}\tilde{\mathbf{x}}_{k-1}^T]\mathbf{F}_{k-1}^T + E[\mathbf{w}_{k-1}\mathbf{w}_{k-1}^T]$$

$$= \mathbf{F}_{k-1}\mathbf{P}_{k-1}\mathbf{F}_{k-1}^T + \mathbf{Q}_{k-1} \tag{34}$$

which gives the a priori value of the covariance matrix of estimation uncertainty as a function of the previous a posteriori value.

These results obtained with the least-mean-squared estimation error do not depend on what probability distribution is used, as long as it has the required first and second moments.

1.2 Maximum Likelihood approach

Now let's look from the linear Gaussian maximum likelihood estimator prospective. We'll use the mean μ_x of X and the information matrix $Y_{xx} \equiv P_{xx}^{-1}$ as parameters for a Gaussian distribution.

$$p(x, \mu_x, P_{xx}) = \frac{1}{\sqrt{2\pi|P_{xx}|}} e^{-\frac{1}{2}(x-\mu_x)^T P_{xx}^{-1}(x-\mu_x)} \tag{35}$$

where P_{xx} is the covariance (second central moment).

The Gaussian likelihood function equivalent to (35) would be of the same form

$$\mathcal{L}(x, \mu_x, P_{xx}) = c\, e^{-\frac{1}{2}(x-\mu_x)^T P_{xx}^{-1}(x-\mu_x)} \tag{36}$$

and the log-likelihood

$$\ln \mathcal{L} = \ln\, c - \frac{1}{2}(x-\mu_x)^T Y_{xx}(x-\mu_x) \tag{37}$$

All covariance matrices in Kalman filtering are symmetric and positive definite, because the variances of estimated quantities are never absolutely zero. Consequently, all covariance matrices P_{xx} in Kalman filtering will have a matrix inverse $Y_{xx} = P_{xx}^{-1}$, the corresponding information matrix. Also Y_{xx} will also be symmetric and positive definite. In fact, they have the same eigenvectors, and the corresponding eigenvalues of Y_{xx} will be the reciprocals of those of P_{xx}.

In Maximum Likelihood estimation, however, the information matrices Y_{xx} are only symmetric and non-negative definite (i.e. with zero eigenvalues possible) and, therefore, not necessarily invertible.

Using the information matrix in place of the covariance matrix in Gaussian likelihood functions allows us to model what estimation theorists would call "flat priors", a condition under which prior assumptions have no influence on the ultimate estimate. This cannot be done using covariance matrices, because it would require that some eigenvalues be infinite. It can be done using information matrices by allowing them to have zero eigenvalues whose eigenvectors represent linear combinations of the state space in which there is zero information. For example, information matrices can be used to represent the information in a measurement, and the dimension of which may be less than the dimension of the state vector.

Using the singular value decomposition we can write

$$Y_{xx} = \sum_i \lambda_i \mathbf{v}_i \mathbf{v}_i^T = V \, diag(\lambda) \, V^T$$

The Moore-Penrose generalized inverse of Y_{xx} can be defined in terms of its *svd* as

$$Y_{xx}^{\dagger} = \sum_{\lambda_i \neq 0} \lambda_i^{-1} \mathbf{v}_i \mathbf{v}_i^T \tag{38}$$

which is always symmetric and non-negative definite and of the same rank as Y_{xx}.

Two probability distributions are called statistically independent if and only if their joint probability is the product of the individual probabilities. The same is true for likelihoods. Let's denote the joint likelihood function $\mathcal{L}_c(x, \mu_c, Y_c)$ of two independent Gaussian likelihoods $\mathcal{L}_a(x, \mu_a, Y_a)$ and $\mathcal{L}_b(x, \mu_b, Y_b)$.

$$\mathcal{L}_c = c_c e^{-\frac{1}{2}(x-\mu_c)^T Y_c (x-\mu_c)}$$

$$= \mathcal{L}_a \mathcal{L}_b = c_a e^{-\frac{1}{2}(x-\mu_a)^T Y_a (x-\mu_a)} c_b e^{-\frac{1}{2}(x-\mu_b)^T Y_b (x-\mu_b)}$$

$$= c_a c_b e^{-\frac{1}{2}(x-\mu_a)^T Y_a (x-\mu_a) - \frac{1}{2}(x-\mu_b)^T Y_b (x-\mu_b)}$$

Taking the logarithm of both sides and differentiating once and twice with respect to x will yield the following sequence of equations:

$$\ln\, c_c - \frac{1}{2}(x-\mu_c)^T Y_c (x-\mu_c) = \ln\, c_a + \ln\, c_b - \frac{1}{2}(x-\mu_a)^T Y_a (x-\mu_a) - \frac{1}{2}(x-\mu_b)^T Y_b (x-\mu_b) \tag{39}$$

$$Y_c(x - \mu_c) = Y_a(x - \mu_a) + Y_b(x - \mu_b) \tag{40}$$

$$Y_c = Y_a + Y_b \tag{41}$$

the last line of which says that information is additive. Setting $x = 0$ in the next-to-last line yields the equation

$$Y_c \mu_c = Y_a \mu_a + Y_b \mu_b \tag{42}$$

The following substitutions will be made in (41) and (42):

$$\left.\begin{aligned}\mu_a &= \hat{\mathbf{x}}_k^- a\,priori\,estimate\\ Y_a &= \mathbf{P}_k^{-1(-)} a\,priori\,information\\ \mu_b &= \mathbf{H}_k^{\dagger}\mathbf{y}_k measurement\,mean\\ Y_b &= \mathbf{H}_k^T\mathbf{R}_k^{-1}\mathbf{H}_k measurement\,information\\ \mu_c &= \hat{\mathbf{x}}_k a\,posteriori\,estimate\\ Y_c &= \mathbf{P}_k^{-1} a\,posteriori\,information\end{aligned}\right\} \tag{43}$$

$$\mathbf{P}_k^{-1} = \mathbf{P}_k^{-1(-)} + \mathbf{H}_k^T\mathbf{R}_k^{-1}\mathbf{H}_k \tag{44}$$

where $\mathbf{P}_k^{-1(-)}$ is the a priori state information and $\mathbf{H}_k^T\mathbf{R}_k^{-1}\mathbf{H}_k$ is the information in the kth measurement $\mathbf{y}_k$. The measurement estimations come from (3) by taking the expectation

$$\mathbf{H}^{-1}\mathbf{y} = \mathbf{x} + \mathbf{H}^{-1}\mathbf{v}$$

$$E[\mathbf{H}^{-1}\mathbf{y}] = E[\mathbf{x}] + E[\mathbf{H}^{-1}\mathbf{v}]$$

the last term expands to $\mathbf{H}^{-1}\mathbf{R}\mathbf{H}^{-1(T)}$; reverse it to obtain the information $\mathbf{H}^T\mathbf{R}^{-1}\mathbf{H}$.

Using an inverse matrix modification formula

$$(A^{-1} + BC^{-1}D)^{-1} = A - AB(C + DAB)^{-1}DA \tag{45}$$

and substituting here

$$\left.\begin{aligned}A^{-1} &= \mathbf{P}_k^{-1(-)} a\,priori\,information\,matrix\,for\,\hat{\mathbf{x}}_k\\ A &= \mathbf{P}_k^- a\,priori\,covariance\,matrix\,for\,\hat{\mathbf{x}}_k\\ B &= \mathbf{H}_k^T transpose\,of\,measurement\,sensitivity\,matrix\\ C &= \mathbf{R}_k covariance\,of\,measurement\,noise\,\mathbf{v}_k\\ D &= \mathbf{H}_k measurement\,sensitivity\,matrix\end{aligned}\right\} \tag{46}$$

The equation (45) becomes

$$(\mathbf{P}_k^{-1(-)} + \mathbf{H}_k^T\mathbf{R}_k^{-1}\mathbf{H}_k)^{-1} = \mathbf{P}_k^- - \mathbf{P}_k^-\mathbf{H}_k^T(\mathbf{H}_k\mathbf{P}_k^-\mathbf{H}_k^T + \mathbf{R}_k)^{-1}\mathbf{H}_k\mathbf{P}_k^- \tag{47}$$

using (44)

$$\mathbf{P}_k = \mathbf{P}_k^- - \mathbf{G}_k\mathbf{H}_k\mathbf{P}_k^- \tag{48}$$

where $\mathbf{G}_k = \mathbf{P}_k^-\mathbf{H}_k^T(\mathbf{H}_k\mathbf{P}_k^-\mathbf{H}_k^T + \mathbf{R}_k)^{-1}$ (cf. with (32) and (26)).
Solving (42)

$$\mathbf{P}_k^{-1}\hat{\mathbf{x}}_k = \mathbf{P}_k^{-1(-)}\hat{\mathbf{x}}_k^- + \mathbf{H}_k^T\mathbf{R}_k^{-1}\mathbf{H}_k\mathbf{H}_k^\dagger\mathbf{y}_k \tag{49}$$

$$\hat{\mathbf{x}}_k = \mathbf{P}_k(\mathbf{P}_k^{-1(-)}\hat{\mathbf{x}}_k^- + \mathbf{H}_k^T\mathbf{R}_k^{-1}\mathbf{H}_k\mathbf{H}_k^\dagger\mathbf{y}_k) \tag{50}$$

Substituting (48) and expanding $\mathbf{G}_k$

$$\hat{\mathbf{x}}_k = (\mathbf{P}_k^- - \mathbf{P}_k^-\mathbf{H}_k^T(\mathbf{H}_k\mathbf{P}_k^-\mathbf{H}_k^T + \mathbf{R}_k)^{-1}\mathbf{H}_k\mathbf{P}_k^-)(\mathbf{P}_k^{-1(-)}\hat{\mathbf{x}}_k^- + \mathbf{H}_k^T\mathbf{R}_k^{-1}\mathbf{H}_k\mathbf{H}_k^\dagger\mathbf{y}_k) \tag{51}$$

Opening the parenthesis and rearranging terms yields

$$\hat{\mathbf{x}}_k = [\mathbf{I} - \mathbf{P}_k^-\mathbf{H}_k^T(\mathbf{H}_k\mathbf{P}_k^-\mathbf{H}_k^T + \mathbf{R}_k)^{-1}\mathbf{H}_k](\hat{\mathbf{x}}_k^- + \mathbf{P}_k^-\mathbf{H}_k^T\mathbf{R}_k^{-1}\mathbf{H}_k\mathbf{H}_k^\dagger\mathbf{y}_k) \tag{52}$$

and again

$$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k^- + \mathbf{P}_k^-\mathbf{H}_k^T(\mathbf{H}_k\mathbf{P}_k^-\mathbf{H}_k^T + \mathbf{R}_k)^{-1}\{[(\mathbf{H}_k\mathbf{P}_k^-\mathbf{H}_k^T + \mathbf{R}_k)\mathbf{R}_k^{-1} - \mathbf{H}_k\mathbf{P}_k^-\mathbf{H}_k^T\mathbf{R}_k^{-1}]\mathbf{y}_k - \mathbf{H}_k\hat{\mathbf{x}}_k^-\} \tag{53}$$

$$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k^- + \mathbf{P}_k^-\mathbf{H}_k^T(\mathbf{H}_k\mathbf{P}_k^-\mathbf{H}_k^T + \mathbf{R}_k)^{-1}\{[\mathbf{H}_k\mathbf{P}_k^-\mathbf{H}_k^T\mathbf{R}_k^{-1} + \mathbf{I} - \mathbf{H}_k\mathbf{P}_k^-\mathbf{H}_k^T\mathbf{R}_k^{-1}]\mathbf{y}_k - \mathbf{H}_k\hat{\mathbf{x}}_k^-\} \tag{54}$$

and finally

$$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k^- + \mathbf{P}_k^-\mathbf{H}_k^T(\mathbf{H}_k\mathbf{P}_k^-\mathbf{H}_k^T + \mathbf{R}_k)^{-1}(\mathbf{y}_k - \mathbf{H}_k\hat{\mathbf{x}}_k^-) \tag{55}$$

Let's define here

$$\mathbf{G}_k \equiv \mathbf{P}_k^-\mathbf{H}_k^T(\mathbf{H}_k\mathbf{P}_k^-\mathbf{H}_k^T + \mathbf{R}_k)^{-1} \tag{56}$$

Compare with (26) and (15) (or (28)).

1.3 Maximum A Posteriori Probability approach

There is an alternative class of estimators called maximum a posteriori probability (MAP) estimators which use Bayes' rule to compute the argmax of the a posteriori probability density function to select the value of the variable to be estimated at which its probability density is greatest (maximum mode) (i.e. maximize $\hat{\mathbf{x}}$ in (57)). These estimators are applicable to a more general class of problems (including non-Gaussian and nonlinear) than the Kalman filter, but they tend to have computational complexities that would eliminate them from consideration for real-time practical implementations as filters. They are used for some nonlinear and non-real-time applications, however.

Bayesian estimation combines a priori information with the measurements through a conditional density function of $\mathbf{x}$ given the measurements $\mathbf{y}$. This conditional probability density function is known as the a posteriori distribution of $\mathbf{x}$. Therefore, Bayesian estimation requires the probability density functions of both the measurement noise and unknown parameters. The posterior density function $p(\mathbf{x}|\mathbf{y})$ for $\mathbf{x}$ (taking the measurement sample $\mathbf{y}$ into account) is given by Bayes' rule:

$$p(\mathbf{x}|\mathbf{y}) = \frac{p(\mathbf{y}|\mathbf{x})p(\mathbf{x})}{p(\mathbf{y})} \tag{57}$$

Note since $\mathbf{y}$ is treated as a set of known quantities, then $p(\mathbf{y})$ provides the proper normalization factor to ensure that $p(\mathbf{x}|\mathbf{y})$ is a probability density function. Alternatively,

$$p(\mathbf{y}) = \int\limits_{-\infty}^{\infty} p(\mathbf{y}|\mathbf{x})p(\mathbf{x})d\mathbf{x}$$

If this integral exists then the posterior function $p(\mathbf{x}|\mathbf{y})$ is said to be *proper*; if it does not exist then $p(\mathbf{x}|\mathbf{y})$ is *improper*, in which case we let $p(\mathbf{y}) = 1$.

Since $p(\mathbf{y})$ does not depend on $\mathbf{x}$ we seek to maximize $p(\mathbf{y}|\mathbf{x})p(\mathbf{x})$, or its natural logarithm

$$J_{MAP}(\hat{\mathbf{x}}) = \ln p(\mathbf{y}|\hat{\mathbf{x}}) + \ln p(\hat{\mathbf{x}}) \tag{58}$$

The first term in the sum is actually the log-likelihood function, and the second term gives the a priori information on the to-be-determined parameters.

Therefore, the MAP estimator maximizes

$$J_{MAP}(\hat{\mathbf{x}}) = \ln \mathcal{L}(\mathbf{y}|\hat{\mathbf{x}}) + \ln p(\hat{\mathbf{x}}) \tag{59}$$

Maximum a posteriori estimation has the following properties:

1. if the a priori distribution $p(\hat{\mathbf{x}})$ is uniform, then MAP estimation is equivalent to maximum likelihood estimation;
2. MAP estimation shares the asymptotic consistency and efficiency properties of maximum likelihood estimation;
3. the MAP estimator converges to the maximum likelihood estimator for large samples;
4. the MAP estimator also obeys the invariance principle.

Let's consider a process following a Gaussian distribution. The assumed probability density functions for this case are given by

$$\mathcal{L}(\mathbf{y}|\hat{\mathbf{x}}) = p(\mathbf{y}|\hat{\mathbf{x}}) = \frac{1}{(2\pi)^{m/2}|\mathbf{R}|^{1/2}} e^{-\frac{1}{2}(\mathbf{y}-\mathbf{H}\hat{\mathbf{x}})^T\mathbf{R}^{-1}(\mathbf{y}-\mathbf{H}\hat{\mathbf{x}})} \tag{60}$$

$$p(\hat{\mathbf{x}}) = \frac{1}{(2\pi)^{n/2}|\mathbf{Q}|^{1/2}} e^{-\frac{1}{2}(\hat{\mathbf{x}}^- - \hat{\mathbf{x}})^T\mathbf{Q}^{-1}(\hat{\mathbf{x}}^- - \hat{\mathbf{x}})} \tag{61}$$

where $\mathbf{H}$ has the dimensions of $m \times n$.

Maximizing (59) w.r.t. $\hat{\mathbf{x}}$ leads to the following estimator:

$$\frac{d}{d\hat{\mathbf{x}}}(\mathbf{y} - \mathbf{H}\hat{\mathbf{x}})^T \mathbf{R}^{-1}(\mathbf{y} - \mathbf{H}\hat{\mathbf{x}}) + \frac{d}{d\hat{\mathbf{x}}}(\hat{\mathbf{x}}^- - \hat{\mathbf{x}})^T \mathbf{Q}^{-1}(\hat{\mathbf{x}}^- - \hat{\mathbf{x}}) = 0$$

$$\frac{d}{d\hat{\mathbf{x}}}(\mathbf{y}^T\mathbf{R}^{-1}\mathbf{y} - \hat{\mathbf{x}}^T\mathbf{H}^T\mathbf{R}^{-1}\mathbf{y} - \mathbf{y}^T\mathbf{R}^{-1}\mathbf{H}\hat{\mathbf{x}} + \hat{\mathbf{x}}^T\mathbf{H}^T\mathbf{R}^{-1}\mathbf{H}\hat{\mathbf{x}}) +$$

$$+\frac{d}{d\hat{\mathbf{x}}}(\hat{\mathbf{x}}_-^T\mathbf{Q}^{-1}\hat{\mathbf{x}}_- - \hat{\mathbf{x}}^T\mathbf{Q}^{-1}\hat{\mathbf{x}}_- - \hat{\mathbf{x}}_-^T\mathbf{Q}^{-1}\hat{\mathbf{x}} + \hat{\mathbf{x}}^T\mathbf{Q}^{-1}\hat{\mathbf{x}}) = 0$$

$$-\mathbf{H}^T\mathbf{R}^{-1}\mathbf{y} - \mathbf{y}^T\mathbf{R}^{-1}\mathbf{H} + 2\mathbf{H}^T\mathbf{R}^{-1}\mathbf{H}\hat{\mathbf{x}} - \mathbf{Q}^{-1}\hat{\mathbf{x}}_- - \hat{\mathbf{x}}_-^T\mathbf{Q}^{-1} + 2\hat{\mathbf{x}}^T\mathbf{Q}^{-1} = 0$$

$$-2\mathbf{H}^T\mathbf{R}^{-1}\mathbf{y} + 2\mathbf{H}^T\mathbf{R}^{-1}\mathbf{H}\hat{\mathbf{x}} - 2\mathbf{Q}^{-1}\hat{\mathbf{x}}_- + 2\mathbf{Q}^{-1}\hat{\mathbf{x}} = 0$$

$$\hat{\mathbf{x}} = (\mathbf{H}^T\mathbf{R}^{-1}\mathbf{H} + \mathbf{Q}^{-1})^{-1}(\mathbf{H}^T\mathbf{R}^{-1}\mathbf{y} + \mathbf{Q}^{-1}\hat{\mathbf{x}}^-) \tag{62}$$

Let's compare this with our previous results. Substitute (26) into (15) to obtain

$$\hat{\mathbf{x}} = \hat{\mathbf{x}}^- + \mathbf{P}^-\mathbf{H}^T(\mathbf{H}\mathbf{P}^-\mathbf{H}^T + \mathbf{R})^{-1}(\mathbf{y} - \mathbf{H}\hat{\mathbf{x}}^-) \tag{63}$$

For our Gaussian (61) $\hat{\mathbf{x}}^- = \mathbf{x} + \mathbf{w}$

$$\mathbf{P}^- = E[(\mathbf{x}-\mathbf{x} - \mathbf{w})(\mathbf{x}-\mathbf{x} - \mathbf{w})^T] = \mathbf{Q}$$

Then (63) becomes

$$\hat{\mathbf{x}} = \hat{\mathbf{x}}^- + \mathbf{Q}\mathbf{H}^T(\mathbf{H}\mathbf{Q}\mathbf{H}^T + \mathbf{R})^{-1}(\mathbf{y} - \mathbf{H}\hat{\mathbf{x}}^-) \tag{64}$$

1. Let's consider the coefficient at $\mathbf{y}$:

$$(\mathbf{H}^T\mathbf{R}^{-1}\mathbf{H} + \mathbf{Q}^{-1})^{-1}\mathbf{H}^T\mathbf{R}^{-1} \overset{?}{=} \mathbf{Q}\mathbf{H}^T(\mathbf{H}\mathbf{Q}\mathbf{H}^T + \mathbf{R})^{-1}$$

$$\mathbf{H}^T\mathbf{R}^{-1}(\mathbf{H}\mathbf{Q}\mathbf{H}^T + \mathbf{R}) \overset{?}{=} (\mathbf{H}^T\mathbf{R}^{-1}\mathbf{H} + \mathbf{Q}^{-1})\mathbf{Q}\mathbf{H}^T$$

$$\mathbf{H}^T\mathbf{R}^{-1}\mathbf{H}\mathbf{Q}\mathbf{H}^T + \mathbf{H}^T = \mathbf{H}^T\mathbf{R}^{-1}\mathbf{H}\mathbf{Q}\mathbf{H}^T + \mathbf{H}^T$$

2. and for $\hat{\mathbf{x}}^-$:

$$(\mathbf{H}^T\mathbf{R}^{-1}\mathbf{H} + \mathbf{Q}^{-1})^{-1}\mathbf{Q}^{-1} \overset{?}{=} \mathbf{I} - \mathbf{Q}\mathbf{H}^T(\mathbf{H}\mathbf{Q}\mathbf{H}^T + \mathbf{R})^{-1}\mathbf{H}$$

$$\mathbf{Q}^{-1} \overset{?}{=} \mathbf{H}^T\mathbf{R}^{-1}\mathbf{H} + \mathbf{Q}^{-1} - (\mathbf{H}^T\mathbf{R}^{-1}\mathbf{H} + \mathbf{Q}^{-1})\mathbf{Q}\mathbf{H}^T(\mathbf{H}\mathbf{Q}\mathbf{H}^T + \mathbf{R})^{-1}\mathbf{H}$$

$$\mathbf{H}^T\mathbf{R}^{-1}\mathbf{H} \overset{?}{=} (\mathbf{H}^T\mathbf{R}^{-1}\mathbf{H} + \mathbf{Q}^{-1})\mathbf{Q}\mathbf{H}^T(\mathbf{H}\mathbf{Q}\mathbf{H}^T + \mathbf{R})^{-1}\mathbf{H}$$

$$\mathbf{H}^T\mathbf{R}^{-1}(\mathbf{H}\mathbf{Q}\mathbf{H}^T + \mathbf{R}) \overset{?}{=} (\mathbf{H}^T\mathbf{R}^{-1}\mathbf{H} + \mathbf{Q}^{-1})\mathbf{Q}\mathbf{H}^T$$

$$\mathbf{H}^T\mathbf{R}^{-1}\mathbf{H}\mathbf{Q}\mathbf{H}^T + \mathbf{H}^T = \mathbf{H}^T\mathbf{R}^{-1}\mathbf{H}\mathbf{Q}\mathbf{H}^T + \mathbf{H}^T$$

Q.E.D.

1.4 Example

In this 2-D example I'll simulate a hound chasing a hare (or an air-to-air missile chasing a MiG). The hare is running along an ellipse (dashed red line). The hound is old and his eyes and nose are no so sharp as they used to be. He detects the hare with an error which is normally distributed around the hare position at time k (solid red dots). But his mind is still sharp. He keeps track of his own position (x_1, x_2), speed (v_1, v_2) and acceleration (a_1, a_2), and calculates his projected position at time $k+1$ (green dots):

$$\mathbf{x}(t) = \mathbf{x}_0 + \mathbf{v}t + \frac{1}{2}\mathbf{a}t^2 + \mathbf{w}(t) = \mathbf{x}_0 + \dot{\mathbf{x}}t + \frac{1}{2}\ddot{\mathbf{x}}t^2 + \mathbf{w}(t)$$

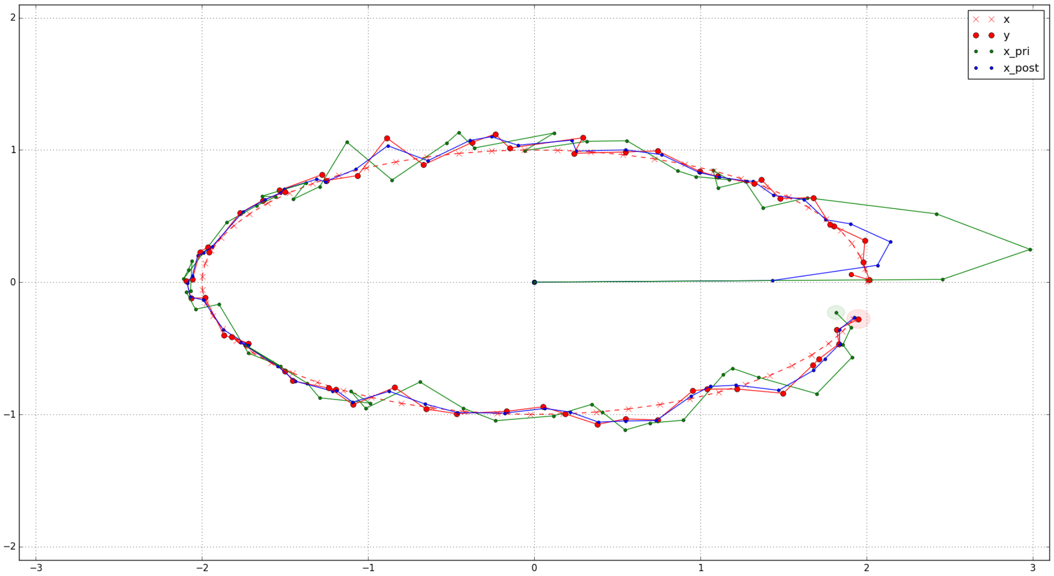
or in discrete form:

$$\mathbf{x}_{k+1} = \begin{pmatrix} x_1 \\ x_2 \\ v_1 \\ v_2 \\ a_1 \\ a_2 \end{pmatrix}_{k+1} = \begin{pmatrix} 1 & 0 & \triangle t & 0 & \frac{1}{2}\triangle t^2 & 0 \\ 0 & 1 & 0 & \triangle t & 0 & \frac{1}{2}\triangle t^2 \\ 0 & 0 & 1 & 0 & \triangle t & 0 \\ 0 & 0 & 0 & 1 & 0 & \triangle t \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ v_1 \\ v_2 \\ a_1 \\ a_2 \end{pmatrix}_{k} + \mathcal{N}(0, q)$$

The process noise $\mathbf{w}$ is due to the paws slippage, wind and other factors not included in our model, and assumed to be Gaussian. As an example, the hound observes only his position $\mathbf{x}$ and not speed or acceleration, i.e.

$$\mathbf{H} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

As mentioned his observation error is also assumed Gaussian: $\mathcal{N}(0, r)$ (solid red dots). The updated positions (the a posteriori approximations) are displayed as blue dots. The last dots also plot circles in light hue with the radius proportional to its respective variance. The initial approximation, arbitrary chosen as $(0, 0)$, is way off but it converges quite fast. As we can see, in most cases the a posteriori position is closer to the truth than both the prediction and observation.



2 Unscented Kalman filter

Let's consider a nonlinear, discrete-time dynamical system with its process equation:

$$\mathbf{x}_{k+1} = f(\mathbf{x}_k, \mathbf{u}_k, \mathbf{q}_k) \tag{65}$$

where $\mathbf{u}_k$ is the input vector. The process noise $\mathbf{q}_k$ caused by disturbances and modeling errors is assumed to be Gaussian (unlike in the particle filter), with zero mean and with covariance matrix defined by

$$E[\mathbf{q}_n \mathbf{q}_k^T] = \begin{cases} \mathbf{Q}_k & for\, n = k \\ \mathbf{0} & for\, n \neq k \end{cases} \tag{66}$$

In this case it is not additive as it transforms through f. The observation equation:

$$\mathbf{y}_k = h(\mathbf{x}_k, \mathbf{u}_k, \mathbf{v}_k) \tag{67}$$

where the measurement noise is Gaussian with zero mean and covariance:

$$E[\mathbf{v}_n \mathbf{v}_k^T] = \begin{cases} \mathbf{R}_k & for\, n = k \\ \mathbf{0} & for\, n \neq k \end{cases} \tag{68}$$

The noises are uncorrelated: $E[\mathbf{q}_n \mathbf{v}_k^T] = 0$.

We seek the minimum-mean-squared error (MMSE) estimate. The MMSE estimate of $\mathbf{x}(k)$ is the conditional mean. Let $\hat{\mathbf{x}}(i|j)$ be the mean of $\mathbf{x}(i)$ conditioned on all of the observations up to and including time j

$$\hat{\mathbf{x}}(i|j) = E[\mathbf{x}(i)|\mathbf{Y}^{(j)}] \tag{69}$$

where $\mathbf{Y}^{(j)} = [y(1), \ldots, y(j)]$. The covariance of this estimate is denoted $\mathbf{P}(i|j)$.

The Kalman filter propagates the first two moments of the distribution of $\mathbf{x}(k)$ recursively. Given an estimate $\hat{\mathbf{x}}(k|k)$, the filter first predicts what the future state of the system will be, using the process model.

$$\hat{\mathbf{x}}(k+1|k) = E[f(\mathbf{x}(k), \mathbf{u}(k), \mathbf{q}(k))|\mathbf{Y}^{(k)}]$$

$$\mathbf{P}(k+1|k) = E[\{\mathbf{x}(k+1) - \hat{\mathbf{x}}(k+1|k)\}\{\mathbf{x}(k+1) - \hat{\mathbf{x}}(k+1|k)\}^T|\mathbf{Y}^{(k)}]$$

The expectations can be calculated only if the distribution of $\mathbf{x}(k)$ conditioned on $\mathbf{Y}^{(k)}$ is known. In general, the distribution cannot be described by a finite number of parameters and most practical systems employ an approximation of some kind. Often the distribution of $\mathbf{x}(k)$ is assumed Gaussian at any time k. Two justifications are made. First, only the mean and covariance need to be maintained. Second, given just the first two moments the Gaussian distribution is the entropy maximizing or least informative distribution.

The estimate at time $k+1$ is given through updating the prediction by the linear update rule. See (15), (30) and (26). The EKF exploits the fact that the error in the prediction, $\tilde{\mathbf{x}}(i|j) = \mathbf{x}(i) - \hat{\mathbf{x}}(i|j)$, can be attained by expanding (65), (67) as a Taylor series about the estimate $\hat{\mathbf{x}}(k|k)$. Truncating this series at the first order yields the approximate linear expression for the propagation of state error as

$$\tilde{\mathbf{x}}(k+1|k) \approx \nabla f_x \tilde{\mathbf{x}}(k|k) + \nabla f_q \mathbf{q}(k)$$

Using this approximation, the state prediction equations are

$$\hat{\mathbf{x}}(k+1|k) = f(\hat{\mathbf{x}}(k|k), \mathbf{u}(k), 0) \tag{70}$$

$$\mathbf{P}(k+1|k) = \nabla f_x \mathbf{P}(k|k) \nabla^T f_x + \nabla f_q \mathbf{Q}(k+1) \nabla^T f_q \tag{71}$$

The problems with EKF are that in many practical applications this linearization introduces significant biases or errors. Also calculating Jacobians at every prediction step can be cumbersome and time consuming.

We use the intuition that it is easier to approximate a probability distribution than it is to approximate an arbitrary nonlinear function or transformation. Following this intuition, we generate a set of points whose sample mean and sample covariance are $\hat{\mathbf{x}}(k|k)$ and $\mathbf{P}(k|k)$, respectively. The nonlinear function is applied to each of these points in turn to yield a transformed sample, and the predicted mean and covariance are calculated from the transformed sample. Unlike in a Monte Carlo method, the sample are not drawn at random but rather carefully chosen so they capture specific information about the distribution. In general, this intuition can be applied to capture many kind of information about many types of distribution. Here we consider the special case of (i) capturing the mean and covariance of an (ii) assumed Gaussian distribution.

The n-dimensional random variable $\mathbf{x}(k)$ with mean $\hat{\mathbf{x}}(k|k)$ and covariance $\mathbf{P}(k|k)$ is approximated by $2n+1$ weighted samples or σ points selected by the algorithm

$$\begin{array}{c}
\mathcal{X}_0(k|k) = \hat{\mathbf{x}}(k|k) \\
W_0 = \frac{\varkappa}{n+\varkappa} \\
\mathcal{X}_i(k|k) = \hat{\mathbf{x}}(k|k) + \left(\sqrt{(n+\varkappa)\mathbf{P}(k|k)}\right)_i \\
W_i = \frac{1}{2(n+\varkappa)} \\
\mathcal{X}_{i+n}(k|k) = \hat{\mathbf{x}}(k|k) - \left(\sqrt{(n+\varkappa)\mathbf{P}(k|k)}\right)_i \\
W_{i+n} = \frac{1}{2(n+\varkappa)}
\end{array} \tag{72}$$

where $\varkappa \in \mathbb{R}$, $\left(\sqrt{(n+\varkappa)\mathbf{P}(k|k)}\right)_i$ is the ith row (for $\mathbf{P} = \mathbf{A}^T\mathbf{A}$) or column (for $\mathbf{P} = \mathbf{A}\mathbf{A}^T$) of the matrix square root of $(n+\varkappa)\mathbf{P}(k|k)$, and W_i is the weight that is associated with the$i$th point.

Valuable insight into the Unscented Transformation can be gained by relating it to a numerical technique called the Gauss-Hermite quadrature rule in the context of state estimation. A close similarity also exists between the UT and the central difference interpolation filtering (CDF) techniques.

Theorem 1: The set of samples chosen by (72) have the same sample mean, covariance, and all higher odd-ordered central moments as the distribution of $\mathbf{x}(k)$. The matrix square root and $\varkappa$ affect the 4th and higher order sample moments of the sigma points.

Proof: The matching of the mean, covariance, and all odd-ordered moments can be directly demonstrated. Because the points are symmetrically distributed and chosen with equal weights about $\bar{\mathbf{x}}$, the sample mean is obviously $\bar{\mathbf{x}}$ and all odd-ordered moments are zero. The sample covariance $\mathbf{P}$ is

$$\begin{aligned}\mathbf{P} &= \sum_{i=0}^{2n} W_i \left[\mathcal{X}_i(k|k) - \hat{\mathbf{x}}(k|k)\right]\left[\mathcal{X}_i(k|k) - \hat{\mathbf{x}}(k|k)\right]^T \\ &= \sum_{i=1}^{n} 2W_i(n+\varkappa)\left(\sqrt{\mathbf{P}(k|k)}\right)_i \left(\sqrt{\mathbf{P}(k|k)}\right)_i^T \\ &= \sum_{i=1}^{n} \left(\sqrt{\mathbf{P}(k|k)}\right)_i \left(\sqrt{\mathbf{P}(k|k)}\right)_i^T \\ &= \mathbf{P}(k|k)\end{aligned}$$

■

Remark 1: The above properties hold for any choice of the matrix square root. Efficient and stable methods, such as Cholesky decomposition, should be used.

Given the set of samples generated by (72), the prediction procedure is as follows.

1. The state is augmented with the noises:

$$\mathbf{x}_a(k) = \begin{pmatrix} \mathbf{x}(k) \\ \mathbf{q}(k) \\ \mathbf{v}(k) \end{pmatrix}$$

$$\hat{\mathbf{x}}_a(k) = \begin{pmatrix} \hat{\mathbf{x}}(k) \\ E[\mathbf{q}(k)] \\ E[\mathbf{v}(k)] \end{pmatrix}$$

or in our case

$$\hat{\mathbf{x}}_a(k) = \begin{pmatrix} \hat{\mathbf{x}}(k) \\ 0 \\ 0 \end{pmatrix}$$

n will now be the size of the vector $\mathbf{x}_a(k)$. The covariance matrix is augmented

$$\mathbf{P}_a(k) = \begin{pmatrix} \mathbf{P}(k) & \mathbf{P}_{xq}(k) & \mathbf{P}_{xv}(k) \\ \mathbf{P}_{xq}^T(k) & \mathbf{Q}(k) & \mathbf{P}_{qv}(k) \\ \mathbf{P}_{xv}^T(k) & \mathbf{P}_{qv}^T(k) & \mathbf{R}(k) \end{pmatrix}$$

or in our case

$$\mathbf{P}_a(k) = \begin{pmatrix} \mathbf{P}(k) & 0 & 0 \\ 0 & \mathbf{Q}(k) & 0 \\ 0 & 0 & \mathbf{R}(k) \end{pmatrix}$$

2. The initial values

$$\hat{\mathbf{x}}(0) = E[\mathbf{x}(0)]$$

$$\mathbf{P}(0) = E[(\mathbf{x}(0) - \hat{\mathbf{x}}(0))(\mathbf{x}(0) - \hat{\mathbf{x}}(0))^T]$$

$$\hat{\mathbf{x}}_a(0) = E[\mathbf{x}_a(0)] = \begin{pmatrix} \hat{\mathbf{x}}(0) \\ 0 \\ 0 \end{pmatrix}$$

$$\mathbf{P}_a(0) = E[(\mathbf{x}_a(0) - \hat{\mathbf{x}}_a(0))(\mathbf{x}_a(0) - \hat{\mathbf{x}}_a(0))^T] = \begin{pmatrix} \mathbf{P}(0) & 0 & 0 \\ 0 & Q & 0 \\ 0 & 0 & R \end{pmatrix}$$

3. Calculate the σ points

$$\mathcal{X}_i^a(k) = \begin{cases} \hat{\mathbf{x}}_a(k) & i = 0 \\ \hat{\mathbf{x}}_a(k) + \left(\sqrt{(n+\varkappa)\mathbf{P}_a(k)}\right)_i & i = 1, \ldots, n \\ \hat{\mathbf{x}}_a(k) - \left(\sqrt{(n+\varkappa)\mathbf{P}_a(k)}\right)_{i-n} & i = n+1, \ldots, 2n \end{cases}$$

for ith column.

4. Each σ point is instantiated through the process model to yield a set of transformed samples

$$\mathcal{X}_i^a(k+1|k) = f(\mathcal{X}_i^a(k|k), \mathbf{u}(k), k) \tag{73}$$

5. The predicted mean is computed as

$$\hat{\mathbf{x}}^{(-)}(k+1|k) = \sum_{i=0}^{2n} W_i \mathcal{X}_i^a(k+1|k) \tag{74}$$

6. The predicted covariance is computed as

$$\mathbf{P}^{(-)}(k+1|k) = \sum_{i=0}^{2n} W_i [\mathcal{X}_i^a(k+1|k) - \hat{\mathbf{x}}^{(-)}(k+1|k)][\mathcal{X}_i^a(k+1|k) - \hat{\mathbf{x}}^{(-)}(k+1|k)]^T \tag{75}$$

7. The predicted observation

$$\mathcal{Y}(k+1|k) = h(\mathcal{X}^a(k+1|k)) \tag{76}$$

8. The mean observation is given by

$$\hat{\mathbf{y}}^{(-)}(k+1|k) = \sum_{i=0}^{2n} W_i \mathcal{Y}_i(k+1|k) \tag{77}$$

9. The update step

$$\mathbf{P}_{yy}(k+1|k+1) = \sum_{i=0}^{2n} W_i \left[\mathcal{Y}_i(k+1|k) - \hat{\mathbf{y}}^{(-)}(k+1|k)\right]\left[\mathcal{Y}_i(k+1|k) - \hat{\mathbf{y}}^{(-)}(k+1|k)\right]^T \tag{78}$$

$$\mathbf{P}_{xy}(k+1|k+1) = \sum_{i=0}^{2n} W_i \left[\mathcal{X}_i^a(k+1|k) - \hat{\mathbf{x}}^{(-)}(k+1|k)\right]\left[\mathcal{Y}_i(k+1|k) - \hat{\mathbf{y}}^{(-)}(k+1|k)\right]^T \tag{79}$$

$$\mathbf{G}(k+1|k+1) = \mathbf{P}_{xy}\mathbf{P}_{yy}^{-1} \tag{80}$$

$$\hat{\mathbf{x}}(k+1|k+1) = \hat{\mathbf{x}}^{(-)}(k+1|k) + \mathbf{G}(k+1|k+1)\left[\mathbf{y}(k+1|k+1) - \hat{\mathbf{y}}^{(-)}(k+1|k)\right] \tag{81}$$

$$\mathbf{P}(k+1|k+1) = \mathbf{P}^{(-)}(k+1|k) - \mathbf{G}(k+1|k+1)\mathbf{P}_{yy}(k+1|k+1)\mathbf{G}^T(k+1|k+1) \tag{82}$$

The innovation vector (or measurement prediction error or residual) points from our predicted measurement to the actual measurement: $\mathbf{y}(k+1|k+1) - \hat{\mathbf{y}}^{(-)}(k+1|k)$, and $\mathbf{G}$ determines how a vector in measurement space maps to a correction in state space.

Theorem 2: The prediction algorithm introduces errors in estimating the mean and covariance at the 4th and higher orders in the Taylor series. These higher order terms are a function of $\varkappa$ and the matrix square root used.

Proof: Let's consider a Gaussian-distributed random variable $\mathbf{x}$ with mean $\bar{\mathbf{x}}$ and covariance $\mathbf{P}_x$. We wish to calculate the mean $\bar{\mathbf{y}}$ and covariance $\mathbf{P}_y$ of the random variable $\mathbf{y}$, which is related to $\mathbf{x}$ through the nonlinear analytic function $\mathbf{y} = f(\mathbf{x})$. Note that $\mathbf{y}$ here is not the observable variable in the KF but rather corresponds to the prior $\mathbf{x}^{(-)}$.

Noting that $\mathbf{x}$ can be written as $\mathbf{x} = \bar{\mathbf{x}} + \delta\mathbf{x}$, where $\delta\mathbf{x}$ is a zero-mean Gaussian random variable with covariance $\mathbf{P}_x$, the nonlinear transformation can be expanded as a Taylor series about $\bar{\mathbf{x}}$

$$\mathbf{y} = f(\bar{\mathbf{x}} + \delta\mathbf{x}) = \sum_{i=0}^{\infty} \left[\frac{(\delta\mathbf{x} \cdot \nabla_x)^i f(\mathbf{x})}{i!}\right]_{\mathbf{x}=\bar{\mathbf{x}}} \tag{83}$$

If we define the operator $\mathbf{D}_{\delta\mathbf{x}}^i f$ as

$$\mathbf{D}^i_{\delta\mathbf{x}} f \equiv \left[\left(\delta\mathbf{x} \cdot \nabla_x \right)^i f(\mathbf{x}) \right]_{\mathbf{x}=\bar{\mathbf{x}}}$$

then the Taylor series expansion of the nonlinear transformation $\mathbf{y} = f(\mathbf{x})$ can be written as

$$\mathbf{y} = f(\mathbf{x}) = f(\bar{\mathbf{x}}) + \mathbf{D}_{\delta\mathbf{x}} f + \frac{1}{2}\mathbf{D}^2_{\delta\mathbf{x}} f + \frac{1}{3!}\mathbf{D}^3_{\delta\mathbf{x}} f + \frac{1}{4!}\mathbf{D}^4_{\delta\mathbf{x}} f + \dots \tag{84}$$

The true mean of $\mathbf{y}$ is given by

$$\bar{\mathbf{y}} = E[\mathbf{y}] = E[f(\mathbf{x})]$$

$$= E\left[f(\bar{\mathbf{x}}) + \mathbf{D}_{\delta\mathbf{x}} f + \frac{1}{2}\mathbf{D}^2_{\delta\mathbf{x}} f + \frac{1}{3!}\mathbf{D}^3_{\delta\mathbf{x}} f + \frac{1}{4!}\mathbf{D}^4_{\delta\mathbf{x}} f + \dots \right] \tag{85}$$

If we assume that $\mathbf{x}$ is a symmetrically distributed random variable, then all odd moments will be zero (expectation which is an integral of $\delta\mathbf{x}$ which is $\mathbf{x} - \bar{\mathbf{x}}$ is 0). Also note that

$$E\left[\delta\mathbf{x}\delta\mathbf{x}^T \right] = \mathbf{P}_x \tag{86}$$

Given this, the mean can be reduced further to

$$\bar{\mathbf{y}} = f(\bar{\mathbf{x}}) + \frac{1}{2}\left[\left(\nabla^T \mathbf{P}_x \nabla \right) f(\mathbf{x}) \right]_{\mathbf{x}=\bar{\mathbf{x}}} + E\left[\frac{1}{4!}\mathbf{D}^4_{\delta\mathbf{x}} f + \frac{1}{6!}\mathbf{D}^6_{\delta\mathbf{x}} f + \dots \right] \tag{87}$$

The Unscented Transformation calculates the posterior mean from the propagated σ points using (74). The sigma points are given by (72), i.e.

$$\mathcal{X}_i = \bar{\mathbf{x}} \pm \left(\sqrt{n+\varkappa} \right) \sigma_i = \bar{\mathbf{x}} \pm \tilde{\sigma}_i$$

where σ_i denotes the ith column of the matrix square root of $\mathbf{P}_x$. This implies that

$$\sum_{i=1}^{n} (\sigma_i \sigma_i^T) = \mathbf{P}_x \tag{88}$$

Given this formulation of the sigma points, we can again write the propagation of each point through the nonlinear function as a Taylor series expansion about $\bar{\mathbf{x}}$:

$$f(\mathcal{X}_i) = f(\bar{\mathbf{x}}) + \mathbf{D}_{\tilde{\sigma}_i} f + \frac{1}{2}\mathbf{D}^2_{\tilde{\sigma}_i} f + \frac{1}{3!}\mathbf{D}^3_{\tilde{\sigma}_i} f + \frac{1}{4!}\mathbf{D}^4_{\tilde{\sigma}_i} f + \dots \tag{89}$$

Using (74) and (72), the UT predicted mean is

$$\bar{\mathbf{y}}_{UT} = \frac{\varkappa}{n+\varkappa} f(\bar{\mathbf{x}}) + \frac{1}{2(n+\varkappa)} \sum_{i=1}^{2n} \left(f(\bar{\mathbf{x}}) + \mathbf{D}_{\tilde{\sigma}_i} f + \frac{1}{2}\mathbf{D}^2_{\tilde{\sigma}_i} f + \frac{1}{3!}\mathbf{D}^3_{\tilde{\sigma}_i} f + \frac{1}{4!}\mathbf{D}^4_{\tilde{\sigma}_i} f + \dots \right) =$$

$$= f(\bar{\mathbf{x}}) + \frac{1}{2(n+\varkappa)} \sum_{i=1}^{2n} \left(\mathbf{D}_{\tilde{\sigma}_i} f + \frac{1}{2}\mathbf{D}_{\tilde{\sigma}_i}^2 f + \frac{1}{3!}\mathbf{D}_{\tilde{\sigma}_i}^3 f + \frac{1}{4!}\mathbf{D}_{\tilde{\sigma}_i}^4 f + \ldots \right) \tag{90}$$

Since the sigma points are symmetrically distributed around $\bar{\mathbf{x}}$, all the odd moments are zero. This results in the simplification

$$\bar{\mathbf{y}}_{UT} = f(\bar{\mathbf{x}}) + \frac{1}{2(n+\varkappa)} \sum_{i=1}^{2n} \left(\frac{1}{2}\mathbf{D}_{\tilde{\sigma}_i}^2 f + \frac{1}{4!}\mathbf{D}_{\tilde{\sigma}_i}^4 f + \ldots \right) \tag{91}$$

and since

$$\frac{1}{2(n+\varkappa)} \sum_{i=1}^{2n} \frac{1}{2}\mathbf{D}_{\tilde{\sigma}_i}^2 f = \frac{1}{2(n+\varkappa)} \left(\nabla f\right)^T \left(\frac{1}{2} \sum_{i=1}^{2n} \sqrt{n+\varkappa}\sigma_i\sigma_i^T \sqrt{n+\varkappa} \right) \nabla f =$$

$$= \frac{n+\varkappa}{2(n+\varkappa)} \left(\nabla f\right)^T \frac{1}{2} \left(\sum_{i=1}^{2n} \sigma_i\sigma_i^T \right) \nabla f =$$

$$= \frac{1}{2}\left[\left(\nabla^T \mathbf{P}_x \nabla\right) f(\mathbf{x}) \right]_{\mathbf{x}=\bar{\mathbf{x}}}$$

the UT predicted mean can be further simplified to

$$\bar{\mathbf{y}}_{UT} = f(\bar{\mathbf{x}}) + \frac{1}{2}\left[\left(\nabla^T \mathbf{P}_x \nabla\right) f(\mathbf{x}) \right]_{\mathbf{x}=\bar{\mathbf{x}}} + \frac{1}{2(n+\varkappa)} \sum_{i=1}^{2n} \left(\frac{1}{4!}\mathbf{D}_{\tilde{\sigma}_i}^4 f + \frac{1}{6!}\mathbf{D}_{\tilde{\sigma}_i}^6 f + \ldots \right) \tag{92}$$

When we compare (92) and (87), we can clearly see that the true posterior mean and the mean calculated by the UT agrees exactly to the third order and that errors are only introduced in the 4th and higher order terms. The magnitudes of these errors depends on the choice of the composite scaling parameter $\varkappa$ as well as the higher-order derivatives of f. The parameter $\varkappa$ provides an extra degree of freedom to "fine tune" the higher order moments of the approximation, and can be used to reduce the overall prediction errors. When $\mathbf{x}(k)$ is assumed Gaussian, a useful heuristic is to select $n + \varkappa = 3$. If a different distribution is assumed for $\mathbf{x}(k)$, then a different choice of $\varkappa$ might be more appropriate.

Now let's look at the *accuracy of the covariance*. The true posterior covariance is given by

$$\mathbf{P}_y = E\left[(\mathbf{y} - \bar{\mathbf{y}})(\mathbf{y} - \bar{\mathbf{y}})^T \right] \tag{93}$$

where the expectation is taken over the distribution of $\mathbf{y}$. Substituting (84) and (85) into (93) we get

$$\mathbf{y} - \bar{\mathbf{y}} = f(\bar{\mathbf{x}}) + \mathbf{D}_{\delta\mathbf{x}} f + \frac{1}{2!}\mathbf{D}_{\delta\mathbf{x}}^2 f + \frac{1}{3!}\mathbf{D}_{\delta\mathbf{x}}^3 f + \ldots - f(\bar{\mathbf{x}}) - E\left[\frac{1}{2!}\mathbf{D}_{\delta\mathbf{x}}^2 f + \frac{1}{4!}\mathbf{D}_{\delta\mathbf{x}}^4 f + \ldots \right]$$

post multiplying the state error by the transpose of itself, taking the expectation, and recalling that all odd moments of $\delta\mathbf{x}$ are zero owing to symmetry (e.g. $E[\frac{1}{2!}\mathbf{D}^2_{\delta\mathbf{x}}f \cdot (\mathbf{D}_{\delta\mathbf{x}}f)^T] = 0$)

$$\mathbf{P}_y = E\left[\mathbf{D}_{\delta\mathbf{x}}f \cdot (\mathbf{D}_{\delta\mathbf{x}}f)^T + \frac{1}{3!}\mathbf{D}^3_{\delta\mathbf{x}}f \cdot (\mathbf{D}_{\delta\mathbf{x}}f)^T + \frac{1}{2!}\mathbf{D}^2_{\delta\mathbf{x}}f \cdot (\frac{1}{2!}\mathbf{D}^2_{\delta\mathbf{x}}f)^T + \mathbf{D}_{\delta\mathbf{x}}f \cdot (\frac{1}{3!}\mathbf{D}^3_{\delta\mathbf{x}}f)^T + \ldots\right] -$$

$$-E\left[E\left[\frac{1}{2!}\mathbf{D}^2_{\delta\mathbf{x}}f\right] \cdot (\frac{1}{2!}\mathbf{D}^2_{\delta\mathbf{x}}f)^T + \frac{1}{2!}\mathbf{D}^2_{\delta\mathbf{x}}f \cdot \left(E\left[\frac{1}{2!}\mathbf{D}^2_{\delta\mathbf{x}}f\right]\right)^T - E\left[\frac{1}{2!}\mathbf{D}^2_{\delta\mathbf{x}}f\right] E\left[\frac{1}{2!}\mathbf{D}^2_{\delta\mathbf{x}}f\right]^T + \ldots\right]$$

Limiting to the 3rd order and using (86)

$$\mathbf{P}_y = \left[\left(\nabla\mathbf{P}_x\nabla^T\right) f(\mathbf{x})\right]_{\mathbf{x}=\bar{\mathbf{x}}} + O(\delta\mathbf{x}^4) \qquad (94)$$

The unscented KF predicts the covariance using (see (72))

$$\mathcal{X}_0 = \bar{\mathbf{x}}$$

$$\mathcal{X}_i = \bar{\mathbf{x}} \pm \sigma_i$$

which assures that

$$\mathbf{P}_x = \frac{1}{2(n+\varkappa)}\sum_{i=1}^{2n}(\mathcal{X}_i - \bar{\mathbf{x}})(\mathcal{X}_i - \bar{\mathbf{x}})^T$$

The transformed set of sigma points are evaluated by

$$\mathcal{Y}_i = f(\mathcal{X}_i)$$

The predicted mean is computed as

$$\bar{\mathbf{y}} = \frac{1}{n+\varkappa}\left(\varkappa\mathcal{Y}_0 + \frac{1}{2}\sum_{i=1}^{2n}\mathcal{X}_i\right)$$

And the predicted covariance is computed as

$$\mathbf{P}_y = \sum_{i=0}^{2n} W_i(\mathcal{X}_i - \bar{\mathbf{x}})(\mathcal{X}_i - \bar{\mathbf{x}})^T = \frac{1}{n+\varkappa}\left[\varkappa(\mathcal{Y}_0 - \bar{\mathbf{y}})(\mathcal{Y}_0 - \bar{\mathbf{y}})^T + \frac{1}{2}\sum_{i=1}^{2n}(\mathcal{Y}_i - \bar{\mathbf{y}})(\mathcal{Y}_i - \bar{\mathbf{y}})^T\right] \qquad (95)$$

Here

$$\mathcal{Y}_0 - \bar{\mathbf{y}} = f(\bar{\mathbf{x}}) - f(\bar{\mathbf{x}}) - \frac{1}{2(n+\varkappa)}\sum_{i=1}^{2n}\left(\frac{1}{2!}\mathbf{D}^2_{\bar{\sigma}_i}f + \frac{1}{4!}\mathbf{D}^4_{\bar{\sigma}_i}f + \ldots\right)$$

$$(\mathcal{Y}_0-\bar{\mathbf{y}})(\mathcal{Y}_0-\bar{\mathbf{y}})^T = \frac{1}{4(n+\varkappa)^2}\sum_{i=1}^{2n}\left(\frac{1}{2!}\mathbf{D}^2_{\tilde{\sigma}_i}f + \frac{1}{4!}\mathbf{D}^4_{\tilde{\sigma}_i}f + \ldots\right)\sum_{i=1}^{2n}\left(\frac{1}{2!}\mathbf{D}^2_{\tilde{\sigma}_i}f + \frac{1}{4!}\mathbf{D}^4_{\tilde{\sigma}_i}f + \ldots\right)^T = O(\tilde{\sigma}^4)$$

Using (89) and (91)

$$\mathcal{Y}_i-\bar{\mathbf{y}} = \mathbf{D}_{\tilde{\sigma}_i}f+\frac{1}{2!}\mathbf{D}^2_{\tilde{\sigma}_i}f+\frac{1}{3!}\mathbf{D}^3_{\tilde{\sigma}_i}f+\ldots-\frac{1}{2(n+\varkappa)}\sum_{i=1}^{2n}\left(\frac{1}{2!}\mathbf{D}^2_{\tilde{\sigma}_i}f + \frac{1}{4!}\mathbf{D}^4_{\tilde{\sigma}_i}f + \ldots\right)$$

$$(\mathcal{Y}_i-\bar{\mathbf{y}})(\mathcal{Y}_i-\bar{\mathbf{y}})^T = \mathbf{D}_{\tilde{\sigma}_i}f\cdot(\mathbf{D}_{\tilde{\sigma}_i}f)^T + O(\tilde{\sigma}^4)$$

Plugging these back to (95) yields

$$\mathbf{P}_y = \frac{1}{2(n+\varkappa)}\sum_{i=1}^{2n}\mathbf{D}_{\tilde{\sigma}_i}f\cdot(\mathbf{D}_{\tilde{\sigma}_i}f)^T + O(\tilde{\sigma}^4) =$$

$$= \frac{1}{2(n+\varkappa)}\nabla f\left(\sum_{i=1}^{2n}\sqrt{n+\varkappa}\sigma_i\sigma_i^T\sqrt{n+\varkappa}\right)(\nabla f)^T + O(\tilde{\sigma}^4)$$

and substituting (88)

$$\mathbf{P}_y = \nabla f\mathbf{P}_x(\nabla f)^T + O(\tilde{\sigma}^4)$$

Compare it with (94)

■

To reiterate, when a set of carefully chosen sample points (σ points) propagated through the nonlinear system, for non-Gaussian inputs, they capture the posterior mean and covariance accurately to at least the 2nd order (Taylor series expansion) for any nonlinearity, with the accuracy of 3rd and higher order moments being determined by the choice of $\varkappa$. For a symmetrical (e.g. Gaussian) pdf the accuracy is at least the 3rd order.

2.1 Mackey-Glass Example

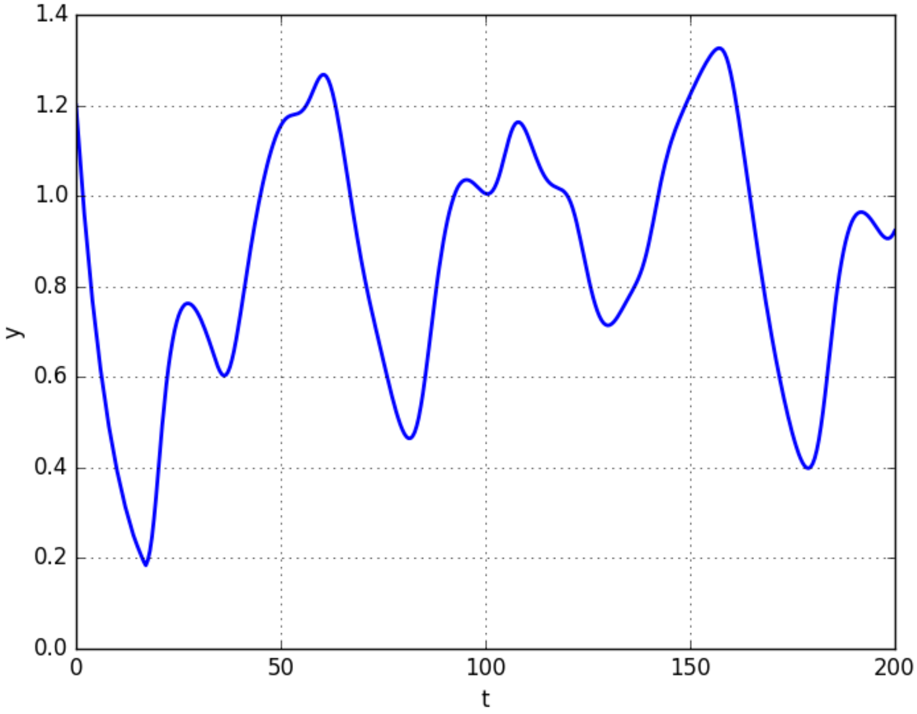
For this example I'll try to predict the Mackey-Glass chaotic series which are defined by

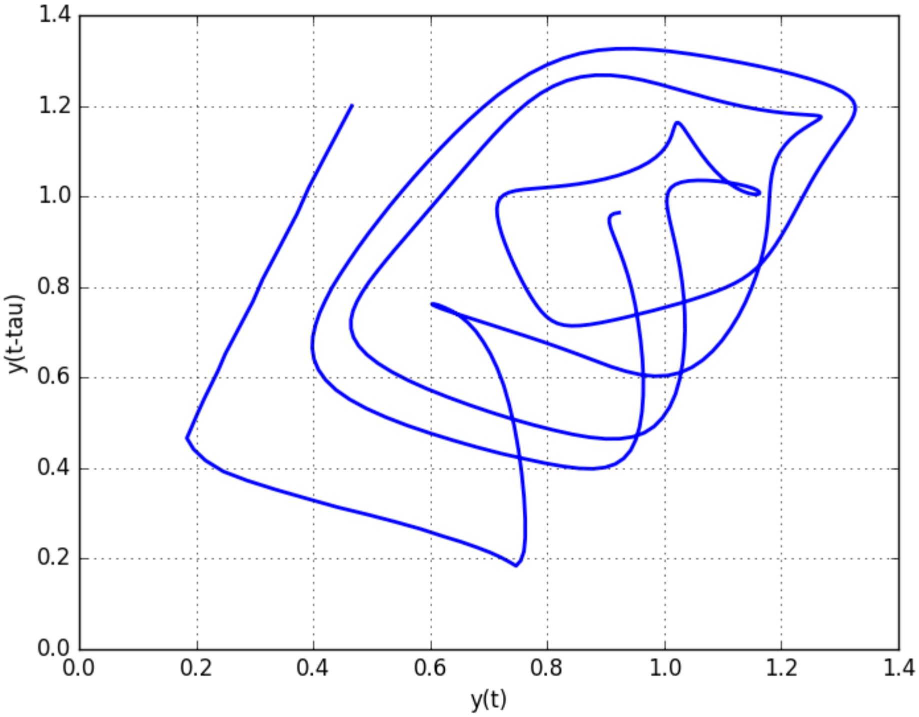
$$\dot{\mathbf{x}} = \beta\frac{\mathbf{x}(t-\tau)}{1+\mathbf{x}(t-\tau)^n} - \gamma\mathbf{x}(t) \tag{96}$$

or for a discrete time t

$$\mathbf{x}(t) = \mathbf{x}(t-1) + \left[\beta\frac{\mathbf{x}(t-\tau)}{1+\mathbf{x}(t-\tau)^n} - \gamma\mathbf{x}(t-1)\right]\triangle t$$

For the delay $\tau = 17$, $\beta = 0.2$, $\gamma = 0.1$, $n = 10$, $\triangle t = 0.1 and x = 1.2 for t \leq 0$ the series look like



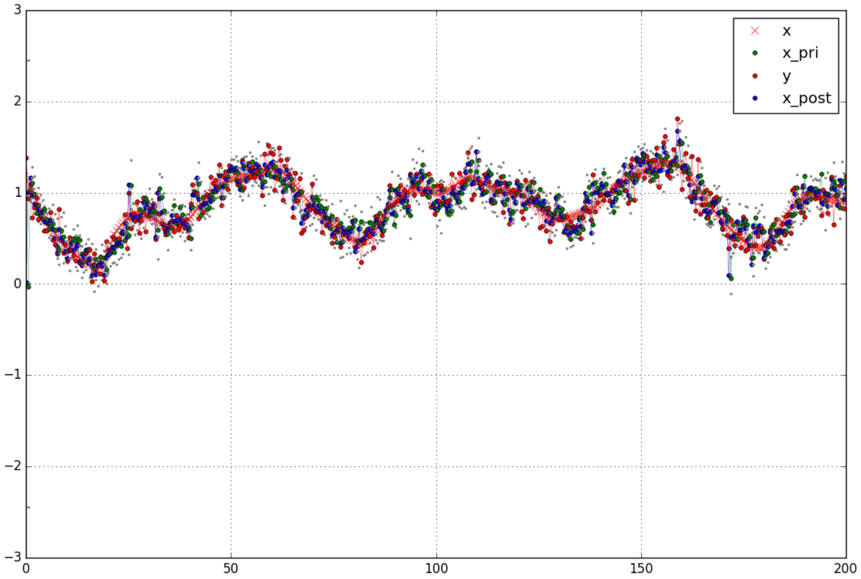


Our augmented state will consist of a scalar function value x and scalar noises:

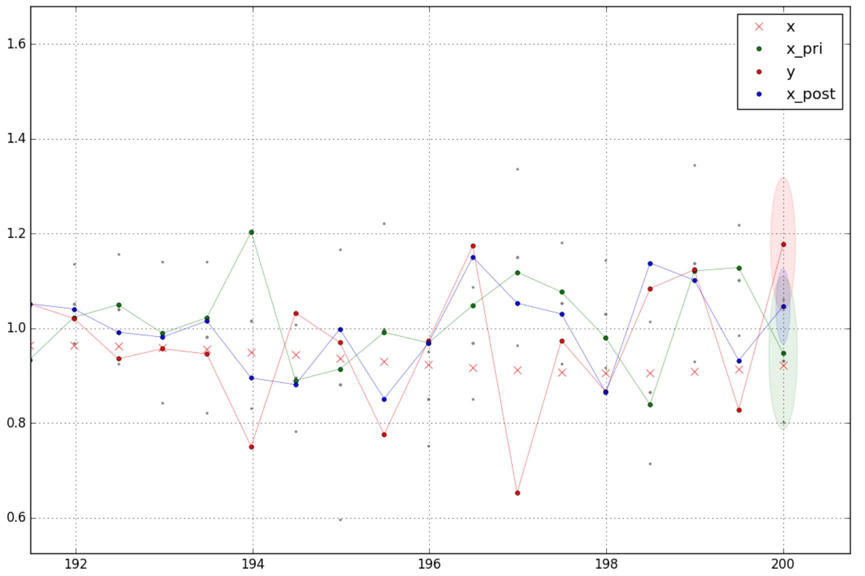
$$\mathbf{x}_a(k) = \begin{pmatrix} x(k) \\ q(k) \\ v(k) \end{pmatrix}_{3\times 1}$$

The initial position is arbitrary chosen at $x = 0$. The initial covariance matrix is equal to

$$\mathbf{P}_a(0) = \begin{pmatrix} 2 & 0 & 0 \\ 0 & 0.02 & 0 \\ 0 & 0 & 0.02 \end{pmatrix}$$



and zoom in at the end



The gray dots indicate the sigma points. The mean a priori values $\hat{\mathbf{x}}^{(-)}$

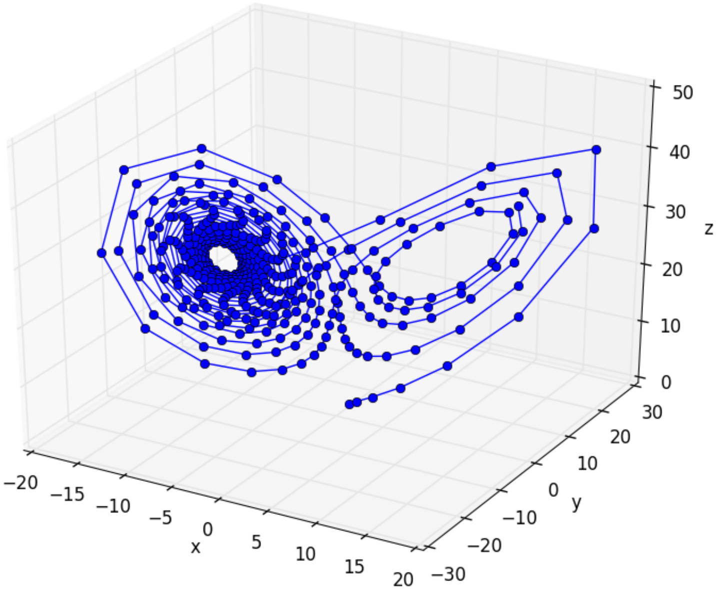
(predictions) are shown in green. The greenish circle of uncertainty has the radius of $\left(\sqrt{\mathbf{P}^{(-)}}\right)_{0,0}$. The observations $\mathbf{y}$ are depicted with red dots. We observe only the first component with some additive Gaussian noise with its standard deviation equal to the square root of r (shown as a reddish circle). The updated a posteriori values $\hat{\mathbf{x}}$ are blue dots with the circle of uncertainty shown in bluish. Its radius is $\left(\sqrt{\mathbf{P}}\right)_{0,0}$.

2.2 Lorenz Attractor Example

The Lorenz equations are

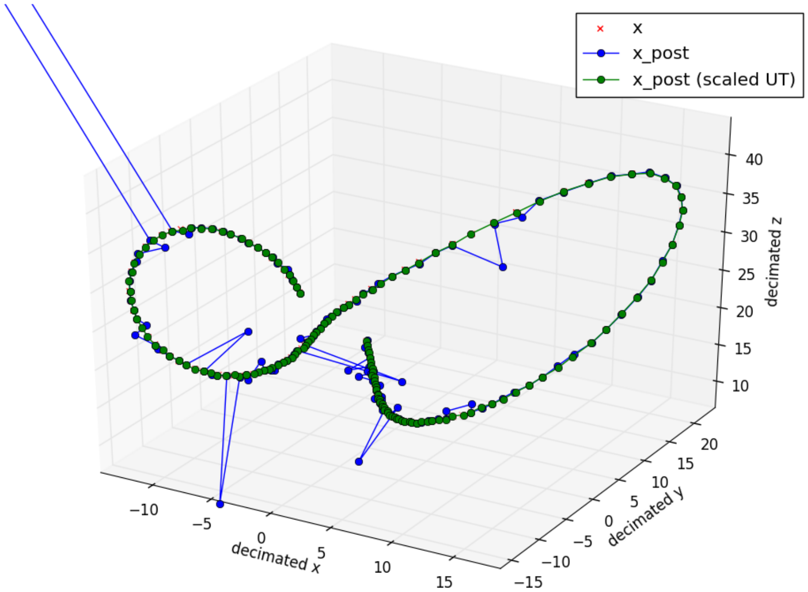
$$\begin{cases} \dot{\mathbf{x}} & = \sigma(\mathbf{y} - \mathbf{x}) \\ \dot{\mathbf{y}} & = \mathbf{x}(\rho - \mathbf{z}) - \mathbf{y} \\ \dot{\mathbf{z}} & = \mathbf{xy} - \beta \mathbf{z} \end{cases} \tag{97}$$

where $\rho = 28,\ \sigma = 10,\ \beta = 8/3$.



The sigma point selection scheme used in the unscented transformation has the property that as the dimension of the state-space increases, the radius of the sphere that bounds all the sigma points increases as well. Even though the mean and covariance of the prior distribution are still captured correctly, it does so at the cost of sampling non-local effects. If the nonlinearities in question are very severe, this can lead to significant difficulties. In order to address this problem, the sigma points can be scaled towards or away from the mean of the prior distribution by a proper choice of $\varkappa$. The distance of the ith sigma point from $\bar{x}$, $|\mathcal{X}_i - \bar{\mathbf{x}}|$, is proportional to $\sqrt{n+\varkappa}$. When $\varkappa = 0$, the distance is proportional to $\sqrt{n}$. When $\varkappa > 0$ the points are scaled further from $\bar{\mathbf{x}}$ and when $\varkappa < 0$ the points are scaled towards $\bar{\mathbf{x}}$. For the special case of $\varkappa = 3 - n$, the desired dimensional scaling invariance is achieved by canceling the effect of n. However, when $\varkappa = 3 - n < 0$ the weight $W_0 < 0$ and the calculated covariance can be non-positive semidefinite. The scaled unscented transformation was developed to address this problem.

Jumping ahead, here is the comparison of Unscented Kalman filter with the scaled one.



We can see that the scaled UKF (green) is much more stable than the unscaled UKF (blue). Red is the truth.

Our augmented state is:

$$\mathbf{x}_a(k) = \begin{pmatrix} x_1 & x_2 & x_3 & \rho & \sigma & \beta & q_1 & q_2 & q_3 & v_1 & v_2 & v_3 \end{pmatrix}^T$$

The initial position is arbitrary chosen as

$$\mathbf{x}_a(0) = \begin{pmatrix} 0 & 0 & 0 & 28 & 10 & \frac{8}{3} & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}^T$$

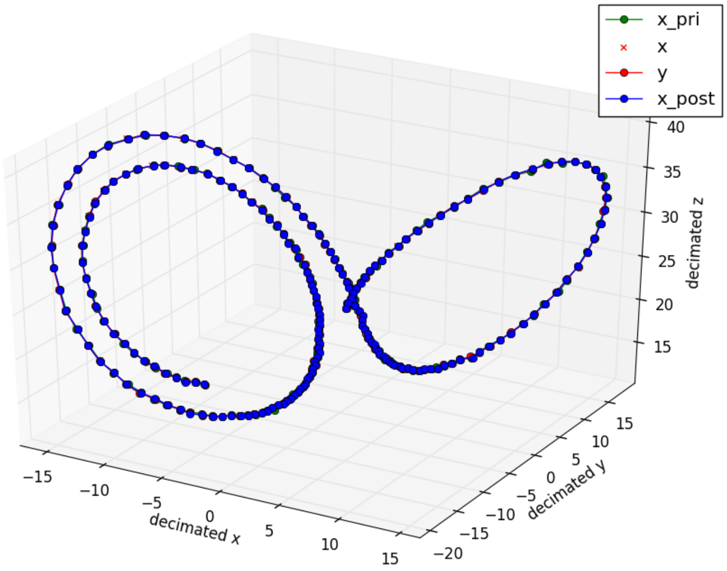
The initial covariance matrix is equal to

$$\mathbf{P}_a(0) = diag\left(\begin{array}{cccccccccccc} 0.2 & 0.2 & 0.2 & 0.01 & 0.01 & 0.01 & 0.01 & 0.01 & 0.01 & 0.005 & 0.005 & 0.005 \end{array}\right)$$

We add a Gaussian noise to the entire state and pass it through f (97) which makes it nonlinear.

We only observe the first 3 components.

For the scaled transformation $\alpha = 0.001,\ \beta = 2$.



As we can see it approximates the function pretty good.

3 Unscented Particle Filter

Particle filters or Sequential Monte Carlo methods are a set of genetic, Monte Carlo algorithms used to solve filtering problems arising in signal processing and Bayesian statistical inference. The particle filter methodology is used to solve Hidden Markov Chain (Model) (HMM) and nonlinear filtering problems.

Our state-space model is the same as in (65) and (67) with the exception that the noise does not have the Gaussian distribution and may even be unknown. The states follow a first order Markov process and the observations are assumed to be independent given the states. The posterior density $p(\mathbf{x}_{0:t}|\mathbf{y}_{1:t})$, where $\mathbf{x}_{0:t} = \{\mathbf{x}_0, \mathbf{x}_1, \ldots, \mathbf{x}_t\}$ and $\mathbf{y}_{1:t} = \{\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_t\}$, constitutes the complete solution to the sequential estimation problem. For such distribution the expectation is calculated as

$$E[g(\mathbf{X})] = I = \int\limits_{\Omega} p(x)g(x)dx \tag{98}$$

One can use the Law of Large Numbers, which states that for a collection of independent identically distributed random variables $\{X_i\}_{i=1}^{\infty}$:

$$E[g(\mathbf{X})] = \lim_{N\to\infty} \frac{1}{N}\sum_{i=1}^{N} g(\mathbf{X}_i)$$

Therefore (98) can be approximated by N random variates $\{X_i\}_{i=1}^{N}$ (particles) with distribution $p(x)$ on Ω. The Monte-Carlo method has an accuracy which can be estimated as:

$$\begin{aligned} error &= \left|\frac{1}{N}\sum_{i=1}^{N} g(\mathbf{X}_i) - I\right| \\ &= \left|\frac{\sigma_g}{\sqrt{N}}\left(\frac{\sum_{i=1}^{N} g(\mathbf{X}_i) - NI}{\sigma_g\sqrt{N}}\right)\right| \\ &\approx \left|\frac{\sigma_g}{\sqrt{N}}\eta(0,1)\right| \end{aligned} \tag{99}$$

where

$$\sigma_g^2 = \int\limits_{\Omega} p(x)(g(x) - I)^2 dx \tag{100}$$

and $\eta(0,1)$ denotes a Gaussian random variable with mean 0 and variance 1. The last approximation was obtained by using the Central Limit Theorem, which states that for a sum of i.i.d. random variables Y_i with mean μ and finite variance σ^2:

$$\frac{\sum_{i=1}^{N} Y_i - N\mu}{\sigma\sqrt{N}} \to \eta(0,1) \; as \; N \to \infty$$

This shows that asymptotically the error converges at a rate $\mathcal{O}(1/\sqrt{N})$, independent of the dimensionality of the problem considered. Furthermore, the convergence rate in the Monte-Carlo method is strongly influenced by the prefactor σ_g which depends on the function $g(x)$ and the sampling distribution with density $p(x)$ that is used. The prefactor σ_g presents the primary avenue by which the convergence rate can be improved.

Transformations of random variables

For a random variables X and $Y = g(X)$ we have $P\{g(X) \in A\} = P\{X \in g^{-1}(A)\}$. For increasing functions the cumulative distribution of X

$$F_Y(y) = P\{Y \leq y\} = P\{g(X) \leq y\} = P\{X \leq g^{-1}(y)\} = F_X(g^{-1}(y))$$

Now use the chain rule to compute the density of Y

$$f_Y(y) = F'_Y(y) = \frac{d}{dy} F_X(g^{-1}(y)) = f_X(g^{-1}(y)) \frac{d}{dy} g^{-1}(y)$$

For g decreasing on the range of X

$$F_Y(y) = P\{Y \leq y\} = P\{g(X) \leq y\} = P\{X \geq g^{-1}(y)\} = 1 - F_X(g^{-1}(y))$$

and the density

$$f_Y(y) = F'_Y(y) = -\frac{d}{dy} F_X(g^{-1}(y)) = -f_X(g^{-1}(y)) \frac{d}{dy} g^{-1}(y)$$

For g decreasing, we also have g^{-1} decreasing and consequently the density of Y is indeed positive. We can combine these 2 cases to obtain the transformation formula for a monotonic function

$$f_Y(y) = f_X(g^{-1}(y)) \left| \frac{d}{dy} g^{-1}(y) \right| \tag{101}$$

Let X be a continuous random variable whose distribution function F_X is strictly increasing on the possible values of X. Then F_X has an inverse function. Let $U = F_X(X)$, then for $u \in [0,1]$

$$P\{U \leq u\} = P\{F_X(X) \leq u\} = P\{X \leq F_X^{-1}(u)\} = F_X\left(F_X^{-1}(u)\right) = u$$

In other words, U is a uniform random variable on $[0,1]$. Most random number generators simulate independent copies of this random variable. Consequently, we can simulate independent random variables having distribution function F_X by simulating U, a uniform random variable on $[0,1]$, and then taking

$$X = F_X^{-1}(U)$$

And for the probability density function we've got

Theorem: Probability Density Transformation. Let $p_X(x)$ be the probability density function for a general n-dimensional random variable $X \in \mathbb{R}^n$. Then the random variable $Z = h(X)$ obtained from an invertible transformation $h : \mathbb{R}^n \to \mathbb{R}^n$ has the probability density

$$p_Z(z) = p_X\left(h^{-1}(z)\right)\left|\frac{dh^{-1}(z)}{dz}\right|$$

where the Jacobian of h^{-1} is defined as

$$\left|\frac{dh^{-1}(z)}{dz}\right| = det\begin{pmatrix} \frac{\partial h_1^{-1}}{\partial z_1} & \frac{\partial h_1^{-1}}{\partial z_2} & \cdots & \frac{\partial h_1^{-1}}{\partial z_n} \\ \vdots & \vdots & & \vdots \\ \frac{\partial h_n^{-1}}{\partial z_1} & \frac{\partial h_n^{-1}}{\partial z_2} & \cdots & \frac{\partial h_n^{-1}}{\partial z_n} \end{pmatrix}$$

Proof:

By definition of the random variable Z and invertibility of h we have:

$$P\{Z \in h(A)\} = P\{X \in A\}$$

From the definition of the probability density we have:

$$P\{X \in A\} = \int\limits_A p_X(x)dx$$

$$P\{Z \in h(A)\} = \int\limits_{h(A)} p_Z(z)dz$$

By the change of variable $z = h(x)$ we have:

$$P\{X \in A\} = \int\limits_{h(A)} p_X(h^{-1}(z))\left|\frac{dh^{-1}(z)}{dz}\right|dz$$

This implies that for any set A we have:

$$\int\limits_{h(A)} p_Z(z)dz = \int\limits_{h(A)} p_X(h^{-1}(z))\left|\frac{dh^{-1}(z)}{dz}\right|dz$$

This requires that

$$p_Z(z) = p_X(h^{-1}(z))\left|\frac{dh^{-1}(z)}{dz}\right|$$

almost everywhere.

■

Sampling

For some probability densities $g(x)$ it may be difficult to determine analytically an appropriate transformation. In such cases the desired random variates can be obtained by generating candidate samples which are either accepted or rejected to obtain the desired distribution.

Importance sampling (to "cover" the function under the integral) is concerned with the choosing $p(x)$ for the random variates X_i so that regions which contribute significantly to the expectation of $g(X)$ are sampled with greater frequency. Thus regions where $f(x)$ is large should be sampled more frequently than those regions where $f(x)$ is comparatively very small. This is done in order to reduce the error (see (100) and (99)).

The most common strategy is to sample from the probabilistic model of the states evolution (transition prior). This strategy can, however, fail if the new measurements appear in the tail of the prior or if the likelihood is too peaked in comparison to the prior. This situation does indeed arise in several areas of engineering and finance where one can encounter sensors that are very accurate (peaked likelihoods) or data that undergoes sudden changes (non-stationarities). To overcome this problem, several techniques based on linearization have been proposed in the literature. For example, the EKF Gaussian approximation is used as the proposal distribution for a particle filter. Here we'll use the scaled Unscented Kalman filter to generate the importance proposal distribution. The UKF allows the particle filter to incorporate the latest observations into a prior updating routine. In addition, the UKF generates proposal distributions that match the true posterior more closely and also has the capability of generating heavier tailed distributions than the well known extended Kalman filter.

Since it is often impossible to sample directly from the posterior density function we can circumvent this difficulty by sampling from a known, easy-to-sample, proposal distribution $q(x_{0:t}|y_{1:t})$ and making use of Bayes' theorem and the following substitution

$$\begin{aligned}
E[g_t(\mathbf{x}_{0:t})] &= \int g_t(\mathbf{x}_{0:t}) \frac{p(\mathbf{x}_{0:t}|\mathbf{y}_{1:t})}{q(\mathbf{x}_{0:t}|\mathbf{y}_{1:t})} q(\mathbf{x}_{0:t}|\mathbf{y}_{1:t}) d\mathbf{x}_{0:t} \\
&= \int g_t(\mathbf{x}_{0:t}) \frac{p(\mathbf{y}_{1:t}|\mathbf{x}_{0:t}) p(\mathbf{x}_{0:t})}{p(\mathbf{y}_{1:t}) q(\mathbf{x}_{0:t}|\mathbf{y}_{1:t})} q(\mathbf{x}_{0:t}|\mathbf{y}_{1:t}) d\mathbf{x}_{0:t} \\
&= \int g_t(\mathbf{x}_{0:t}) \frac{w_t(\mathbf{x}_{0:t})}{p(\mathbf{y}_{1:t})} q(\mathbf{x}_{0:t}|\mathbf{y}_{1:t}) d\mathbf{x}_{0:t}
\end{aligned}$$

where the variables $w_t(\mathbf{x}_{0:t})$ are known as the unnormalized importance weights (likelihood that the particle best represents the true state)

$$w_t(\mathbf{x}_{0:t}) = \frac{p(\mathbf{y}_{1:t}|\mathbf{x}_{0:t}) p(\mathbf{x}_{0:t})}{q(\mathbf{x}_{0:t}|\mathbf{y}_{1:t})} \tag{102}$$

We can get rid of the unknown normalizing density $p(\mathbf{y}_{1:t})$ as follows

$$\begin{aligned}
E[g_t(\mathbf{x}_{0:t})] &= \frac{1}{p(\mathbf{y}_{1:t})} \int g_t(\mathbf{x}_{0:t}) w_t(\mathbf{x}_{0:t}) q(\mathbf{x}_{0:t}|\mathbf{y}_{1:t}) d\mathbf{x}_{0:t} \\
&= \frac{\int g_t(\mathbf{x}_{0:t}) w_t(\mathbf{x}_{0:t}) q(\mathbf{x}_{0:t}|\mathbf{y}_{1:t}) d\mathbf{x}_{0:t}}{\int p(\mathbf{y}_{1:t}|\mathbf{x}_{0:t}) p(\mathbf{x}_{0:t}) \frac{q(\mathbf{x}_{0:t}|\mathbf{y}_{1:t})}{q(\mathbf{x}_{0:t}|\mathbf{y}_{1:t})} d\mathbf{x}_{0:t}} \\
&= \frac{\int g_t(\mathbf{x}_{0:t}) w_t(\mathbf{x}_{0:t}) q(\mathbf{x}_{0:t}|\mathbf{y}_{1:t}) d\mathbf{x}_{0:t}}{\int w_t(\mathbf{x}_{0:t}) q(\mathbf{x}_{0:t}|\mathbf{y}_{1:t}) d\mathbf{x}_{0:t}} \\
&= \frac{E_{q(\cdot|\mathbf{y}_{1:t})}[w_t(\mathbf{x}_{0:t}) g_t(\mathbf{x}_{0:t})]}{E_{q(\cdot|\mathbf{y}_{1:t})}[w_t(\mathbf{x}_{0:t})]}
\end{aligned}$$

where the notation $E_{q(\cdot|\mathbf{y}_{1:t})}$ has been used to emphasize that the expectations are taken over the proposal distribution $q(\cdot|\mathbf{y}_{1:t})$. Hence, by drawing samples from the proposal function $q(\cdot|\mathbf{y}_{1:t})$, we can approximate the expectations of interest by the following estimate

$$\begin{aligned}
\overline{E[g_t(\mathbf{x}_{0:t})]} &= \frac{1/N \sum_{i=1}^{N} g_t(\mathbf{x}_{0:t}^{(i)}) w_t(\mathbf{x}_{0:t}^{(i)})}{1/N \sum_{i=1}^{N} w_t(\mathbf{x}_{0:t}^{(i)})} \\
&= \sum_{i=1}^{N} g_t(\mathbf{x}_{0:t}^{(i)}) \tilde{w}_t(\mathbf{x}_{0:t}^{(i)}) \qquad (103)
\end{aligned}$$

where the normalized importance weights $\tilde{w}_t^{(i)}$ are given by

$$\tilde{w}_t^{(i)} = \frac{w_t^{(i)}}{\sum_{j=1}^{N} w_t^{(j)}}$$

The estimate of equation (103) is biased as it involves a ratio of estimates. However, it is possible to obtain asymptotic convergence and a central limit theorem for $\overline{E[g_t(\mathbf{x}_{0:t})]}$ under the following assumptions:

1. $\mathbf{x}_{0:t}^{(i)}$ corresponds to a set of i.i.d. samples drawn from the proposal distribution, the support of the proposal distribution includes the support of the posterior distribution and $E[g_t(\mathbf{x}_{0:t})]$ exists and is finite.

2. The expectations of w_t and $w_t g_t^2(\mathbf{x}_{0:t})$ over the posterior distribution exist and are finite.

A sufficient condition to verify the second assumption is to have bounds on the variance of $g_t(\mathbf{x}_{0:t})$ and on the importance weights. Thus, as N tends to infinity, the posterior density function can be approximated arbitrarily well by the point-mass estimate

$$\hat{p}(\mathbf{x}_{0:t}|\mathbf{y}_{1:t}) = \sum_{i=1}^{N} \tilde{w}_t^{(i)} \delta_{\mathbf{x}_{0:t}^{(i)}}(d\mathbf{x}_{0:t})$$

Sequential Importance Sampling

If we assume that the states correspond to a Markov process and that the observations are conditionally independent given the states then

$$q(\mathbf{x}_{0:t}|\mathbf{y}_{1:t}) = q(\mathbf{x}_{0:t-1}|\mathbf{y}_{1:t-1})q(\mathbf{x}_t|\mathbf{x}_{0:t-1}, \mathbf{y}_{1:t}) \tag{104}$$

$$p(\mathbf{x}_{0:t}) = p(\mathbf{x}_0)\prod_{j=1}^{t} p(\mathbf{x}_j|\mathbf{x}_{j-1}) \; and \; p(\mathbf{y}_{1:t}|\mathbf{x}_{0:t}) = \prod_{j=1}^{t} p(\mathbf{y}_j|\mathbf{x}_j) \tag{105}$$

By substituting (104) and (105) into (102), a recursive estimate for the importance weights can be derived as follows

$$\begin{aligned} w_t &= \frac{p(\mathbf{y}_{1:t}|\mathbf{x}_{0:t})p(\mathbf{x}_{0:t})}{q(\mathbf{x}_{0:t-1}|\mathbf{y}_{1:t-1})q(\mathbf{x}_t|\mathbf{x}_{0:t-1}, \mathbf{y}_{1:t})} \\ &= w_{t-1}\frac{p(\mathbf{y}_{1:t}|\mathbf{x}_{0:t})p(\mathbf{x}_{0:t})}{p(\mathbf{y}_{1:t-1}|\mathbf{x}_{0:t-1})p(\mathbf{x}_{0:t-1})}\frac{1}{q(\mathbf{x}_t|\mathbf{x}_{0:t-1}, \mathbf{y}_{1:t})} \\ &= w_{t-1}\frac{p(\mathbf{y}_t|\mathbf{x}_t)p(\mathbf{x}_t|\mathbf{x}_{t-1})}{q(\mathbf{x}_t|\mathbf{x}_{0:t-1}, \mathbf{y}_{1:t})} \end{aligned} \tag{106}$$

Equation (106) provides a mechanism to sequentially update the importance weights, given an appropriate choice of proposal distribution $q(\mathbf{x}_t|\mathbf{x}_{0:t-1}, \mathbf{y}_{1:t})$. The exact form of this distribution is a critical design issue and is usually approximated in order to facilitate easy sampling. Since we can sample from the proposal distribution and evaluate the likelihood and transition probabilities, all we need to do is generate a prior set of samples and iteratively compute the importance weights. This procedure, known as sequential importance sampling (SIS), allows us to obtain the type of estimates described by equation (103).

It was proven that the proposal distribution $q(\mathbf{x}_t|\mathbf{x}_{0:t-1}, \mathbf{y}_{1:t}) = p(\mathbf{x}_t|\mathbf{x}_{0:t-1}, \mathbf{y}_{1:t})$ minimizes the variance of the importance weights conditional on $\mathbf{x}_{0:t-1}$ and $\mathbf{y}_{1:t}$. Nonetheless, the distribution

$$q(\mathbf{x}_t|\mathbf{x}_{0:t-1}, \mathbf{y}_{1:t}) \cong p(\mathbf{x}_t|\mathbf{x}_{t-1})$$

(the transition prior) is the most popular choice of proposal function. Although it results in higher Monte Carlo variation than the optimal proposal $p(\mathbf{x}_t|\mathbf{x}_{0:t-1}, \mathbf{y}_{1:t})$, as a result of it not incorporating the most recent observations, it is usually easier to implement. The transition prior is defined in terms of the probabilistic model governing the states' evolution (65) and the process noise statistics. For example, if an additive Gaussian process noise model is used, the transition prior is simply,

$$p(\mathbf{x}_t|\mathbf{x}_{t-1}) = \mathcal{N}\left(f(\mathbf{x}_{t-1}, 0), Q_{t-1}\right)$$

But because this fails to use the latest available information to propose new values for the states, only a few particles will have significant importance weights when their likelihood are evaluated, and therefore we will not use it.

Resampling (Selection)

It was shown that the variance of importance weights (102) or (106) increases stochastically over time. This can be observed as one of the normalized importance weights tends to 1, while the remaining weights tend to 0. A large number of samples are thus effectively removed from the sample set because their importance weights become numerically insignificant. To avoid this degeneration or depletion of samples a selection (re-sampling) stage may be used to eliminate samples with low importance weights and multiply samples with high importance weights. It is possible to see an analogy to the steps in *genetic algorithms*.

A selection scheme associates to each particle $\mathbf{x}_{0:t}^{(i)}$ a number of "children", say $N_i \in \mathbb{N}$, such that $\sum_{i=1}^{N} N_i = N$. Several selection schemes have been proposed in the literature. These schemes satisfy $E[N_i] = N\tilde{w}_t^{(i)}$ but their performance varies in terms of the variance of the particles $var(N_i)$.

As we have already mentioned, resampling has the effect of removing particles with low weights and multiplying particles with high weights. However, this is at the cost of immediately introducing some additional variance. If particles have unnormalized weights with a small variance then the resampling step might be unnecessary. Consequently, in practice, it is more sensible to resample only when the variance of the unnormalized weights is superior to a pre-specified threshold. This is often assessed by looking at the variability of the weights using the so-called Effective Sample Size (ESS) criterion which is given by

$$ESS = \left(\sum_{i=1}^{N} \left(\tilde{w}_t^{(i)} \right)^2 \right)^{-1}$$

Its interpretation is that in a simple IS setting, inference based on the N weighted samples is approximately equivalent (in terms of estimator variance) to inference based on ESS perfect samples from the target distribution. The ESS takes values between 1 and N and we resample only when it is below a threshold N_T; typically $N_T = N/2$. Alternative criteria can be used such as the entropy of the weights $\tilde{w}_t^{(i)}$ which achieves its maximum value when $\tilde{w}_t^{(i)} = 1/N$. In this case, we resample when the entropy is below a given threshold.

Because of this sample degeneracy any Sequential MC algorithm which relies upon the distribution of full paths $x_{1:n}$ will fail for large enough n for any finite sample size N, in spite of the asymptotic justification. It is intuitive that one should endeavor to employ algorithms which do not depend upon the full path of the samples, but only upon the distribution of some finite component $x_{n-L:n}$ for some fixed L which is independent of n. Furthermore, ergodicity (a tendency for the future to be essentially independent of the distant past) of the underlying system will prevent the accumulation of errors over time.

Although sample degeneracy emerges as a consequence of resampling, it is really a manifestation of a deeper problem – one which resampling actually mitigates. It is inherently impossible to accurately represent a distribution on a space of arbitrary high dimension with a sample of fixed, finite size. Sample impoverishment is a term which is often used to describe the situation in which very few different particles have significant weight. This problem has much the same effect as sample degeneracy and occurs, in the absence of resampling, as the inevitable consequence of multiplying together incremental importance weights from a large number of time steps. It is, of course, not possible to circumvent either problem by increasing the number of samples at every iteration to maintain a constant effective sample size as this would lead to an exponential growth in the number of samples required. This sheds some light on the resampling mechanism: it"resets the system" in such a way that its representation of final time marginals remains well behaved at the expense of further diminishing the quality of the path-samples. By focusing on the fixed-dimensional final time marginals in this way, it allows us to circumvent the problem of increasing dimensionality.

We will now present a number of selection or resampling schemes, namely: *sampling-importance resampling* (SIR), *residual resampling* and *minimum variance sampling*. We found that the specific choice of resampling scheme does not significantly affect the performance of the particle filter, so we used residual resampling in all of the experiments.

Sampling-importance resampling (SIR) and multinomial sampling

Resampling involves mapping the Dirac random measure $\left\{\mathbf{x}_{0:t}^{(i)}, \tilde{w}_t^{(i)}\right\}$ into an equally weighted random measure $\left\{\mathbf{x}_{0:t}^{(i)}, N^{-1}\right\}$. This can be accomplished by sampling uniformly from the discrete set $\left\{\mathbf{x}_{0:t}^{(i)}; i = 1, \ldots, N\right\}$ with probabilities $\left\{\tilde{w}_t^{(i)}; i = 1, \ldots, N\right\}$. We do this by constructing the cumulative distribution of the discrete set. Then a uniformly drawn sampling index i is projected onto the distribution range and then onto the distribution domain. The intersection with the domain constitutes the new sample index j. That is, the vector $\mathbf{x}_{0:t}^{(j)}$ is accepted as the new sample. Clearly, the vectors with the larger sampling weights will end up with more copies after the resampling process.

Sampling N times from the cumulative discrete distribution $\sum_{i=1}^{N} \tilde{w}_t^{(i)} \delta_{\mathbf{x}_{0:t}^{(i)}}(d\mathbf{x}_{0:t})$ is equivalent to drawing $(N_i; i = 1, \ldots, N)$ from a multinomial distribution with parameters N and $\tilde{w}_t^{(i)}$. This procedure can be implemented in $\mathcal{O}(N)$ operations. As we are sampling from a multinomial distribution, the variance is $var(N_i) = N\tilde{w}_t^{(i)}(1 - \tilde{w}_t^{(i)})$.

Residual resampling

This procedure involves the following steps. Firstly, set $\tilde{N}_i = \left\lfloor N\tilde{w}_t^{(i)} \right\rfloor$. Secondly, perform an SIR procedure to select the remaining $\bar{N}_t = N - \sum_{i=1}^{N} \tilde{N}_i$ samples with new weights $w_t'^{(i)} = \bar{N}_t^{-1}(\tilde{w}_t^{(i)} N - \tilde{N}_i)$. Finally, add the results to the current $\tilde{N}_i$. For this scheme, the variance $var(N_i) = \bar{N}_t w_t'^{(i)}(1 - w_t'^{(i)})$ is smaller than the one given by the SIR scheme. Moreover, this procedure is computationally cheaper.

Minimum variance sampling

This strategy includes the stratified/systematic sampling procedures and the Tree Based Branching Algorithm. One samples a set of N points U in the interval $[0,1]$, each of the points a distance N^{-1} apart. The number of children N_i is taken to be the number of points that lie between $\sum_{j=1}^{i-1} \tilde{w}_t^{(j)}$ and $\sum_{j=1}^{i} \tilde{w}_t^{(j)}$. This strategy introduces a variance on N_i even smaller than the residual resampling scheme, namely $var(N_i) = \bar{N}_t w_t'^{(i)}(1 - \bar{N}_t w_t'^{(i)})$. Its computational complexity is $\mathcal{O}(N)$.

MCMC Move Step

In the resampling stage, any particular sample with a high importance weight will be duplicated many times. As a result, the cloud of samples may eventually collapse to a single sample. This degeneracy will limit the ability of the algorithm to search for lower minima in other regions of the error surface. In other words, the number of samples used to describe the posterior density function will become too small and inadequate. A brute force strategy to overcome this problem is to increase the number of particles. A more refined strategy is to implement a Markov chain Monte Carlo (MCMC) step after the selection step.

After the selection scheme at time t, we obtain N particles distributed marginally approximately according to $p(\mathbf{x}_{0:t}|\mathbf{y}_{1:t})$. Since the selection step favors the creation of multiple copies of the "fittest" particles, it enables us to track time varying filtering distributions. However, many particles might end up having no children ($N_i = 0$), whereas others might end up having a large number of children, the extreme case being $N_i = N$ for a particular value i. In this case, there is a severe depletion of samples. We, therefore, require a procedure to introduce sample variety after the selection step without affecting the validity of the approximation.

A transition kernel for a Markov chain when X is discrete is simply a transition matrix $\mathcal{K}$ with elements

$$\mathcal{K} = \begin{pmatrix} p_{1,1} & \cdots & p_{1,n} \\ \vdots & & \vdots \\ p_{m,1} & \cdots & p_{m,n} \end{pmatrix}$$

where $p_{i,j}$ is the probability of moving from the current state i to a new state j; $i, j \in X$. Now, rather than start with a specific state we consider a probability distribution over these states, $\mathbf{v}^{(0)} = \left(v_1^{(0)}, v_2^{(0)}, \ldots, v_n^{(0)}\right)^T$. If we randomly select the initial state from this distribution, then the probability distribution of the next state in the chain is given by

$$\mathbf{v}^{(1)} = \mathcal{K}\mathbf{v}^{(0)}$$

This idea can be extended, the probability distribution over the states after the second move is simply $\mathbf{v}^{(2)} = \mathcal{K}\mathbf{v}^{(1)} = \mathcal{K}^2\mathbf{v}^{(0)}$. This idea can be generalized; specifically, $\mathbf{v}^{(t)} = \mathcal{K}^t\mathbf{v}^{(0)}$. Of particular interest, is the distribution as the chain becomes long. As the chain's length increases then the distribution over the states becomes less and less determined by the starting distribution and more and more determined by the transition probabilities. Indeed, providing the chain satisfies certain regularity conditions, i.e. it does not get stuck in one state, there exists a unique invariant distribution associated with every transition matrix. Let π represent this invariant distribution. So for any starting distribution, $\pi^{(0)}$, as the chain becomes long then the $\pi^{(t)}$ tends to π ($\lim\limits_{t\to\infty} \pi^{(t)} = \pi$).

There are two ways to calculate this invariant distribution. The first is analytical. This method exploits the fact that $\pi = \mathcal{K}\pi$, and solves this system of equations. The second is to simulate π by actually running the Markov chain. This involves choosing a starting value and simply running the Markov chain. The initial values in the chain depend strongly upon the starting values, hence they are usually discarded. However, as the chain becomes longer then the elements of the chain represent random draws from the (invariant) probability distribution π. Another practical problem is the high auto-correlation between elements in the chain. This reduces the rate at which convergence is achieved. A practical solution is to sub-sample.

These ideas are readily extendable to continuous state space models, where the transition matrix is replaced by a transition kernel (a conditional probability density over the next state that depends only upon the current state) (notation: commonly written $P(x, y)$ instead of $p(y|x)$)

$$\mathcal{K}(x, x')P(X \in A|x) = \int\limits_A \mathcal{K}(x, x')dx' = \int\limits_A f(x'|x)dx'$$

where $f(x'|x)$ is a density function.

Most of Markov theory revolves around finding the invariant distribution of Markov chains. MCMC turns the problem around. Rather than finding the invariant distribution of a specific Markov chain, it starts with a specific invariant distributions and says, can I find a Markov chain that has this invariant distribution. (Each Markov process has a unique invariant distribution. Yet, many Markov chains could have the same invariant distribution. Thus, we are free to use any of these process to simulate the invariant distribution.) Typically, we already know the distribution of interest: the posterior distribution of the parameters. The key is to find a transition kernel that has this invariant distribution $p(\mathbf{x}_{0:t}|\mathbf{y}_{1:t})$.

A strategy for solving the sample depletion problem involves introducing MCMC steps of invariant distribution $p(\mathbf{x}_{0:t}|\mathbf{y}_{1:t})$ on each particle. The basic idea is that if the particles are distributed according to the posterior $p(\tilde{\mathbf{x}}_{0:t}|\mathbf{y}_{1:t})$, then applying a Markov chain transition kernel $\mathcal{K}(\mathbf{x}_{0:t}|\tilde{\mathbf{x}}_{0:t})$, with invariant distribution $p(\mathbf{x}_{0:t}|\mathbf{y}_{1:t})$ such that $\int \mathcal{K}(\mathbf{x}_{0:t}|\tilde{\mathbf{x}}_{0:t})p(\tilde{\mathbf{x}}_{0:t}|\mathbf{y}_{1:t}) = p(\mathbf{x}_{0:t}|\mathbf{y}_{1:t})$, still results in a set of particles distributed according to the posterior of interest. However, the new particles might have been moved to more interesting areas of the state-space. In fact, by applying a Markov transition kernel, the total variation of the current distribution with respect to the invariant distribution can only decrease. Note that we can incorporate any of the standard MCMC methods, such as the Gibbs sampler and Metropolis Hastings algorithms, into the filtering framework, but we no longer require the kernel to be ergodic. The MCMC move step can also be interpreted as sampling from the finite mixture distribution $N^{-1}\sum_{i=1}^{N}\mathcal{K}(\mathbf{x}_{0:t}|\tilde{\mathbf{x}}_{0:t}^{(i)})$.

One can generalize this idea by introducing MCMC steps on the product space with invariant distribution $\prod_{i=1}^{N} p(\mathbf{x}_{0:t}^{(i)}|\mathbf{y}_{1:t})$, that is to apply MCMC steps on the entire population of particles. It should be noted that independent MCMC steps spread out the particles in a particular mode more evenly, but do not explore modes devoid of particles, unless "clever" proposal distributions are available. By adopting MCMC steps on the whole population, we can draw upon many of the ideas developed in parallel MCMC computation. In this work, however, we limit ourselves to the simpler case of using independent MCMC transitions steps on each particle. In the case of standard particle filters, we propose to sample from the transition prior and accept according to a Metropolis-Hastings (MH) step as follows.

It is easy to construct a Markov kernel with a specified invariant distribution. For example, we could consider the following kernel, based upon the Gibbs sampler: set $x'_{1:n-L} = x_{1:n-L}$ then sample x'_{n-L+1} from $p(x_{n-L+1}|y_{1:n}, x'_{1:n-L}, x_{n-L+2:n})$, sample x'_{n-L+2} from $p(x_{n-L+2}|y_{1:n}, x'_{1:n-L+1}, x_{n-L+3:n})$ and so on until we sample x'_n from $p(x_n|y_{1:n}, x'_{1:n-1})$; that is

$$\mathcal{K}_n(x'_{1:n}|x_{1:n}) = \delta_{x_{1:n-L}}(x'_{1:n-L}) \prod_{k=n-L+1}^{n} p(x'_k|y_{1:n}, x'_{1:k-1}, x_{k+1:n})$$

and we write, with a slight abuse of notation, the non-degenerate component of the MCMC kernel $\mathcal{K}_n(x'_{n-L+1:n}|x_{1:n})$. It is straightforward to verify that this kernel is $p(x_{1:n}|y_{1:n})$-invariant.

If it is not possible to sample from $p(x'_k|y_{1:n}, x'_{1:k-1}, x_{k+1:n}) = p(x'_k|y_k, x'_{k-1}, x_{k+1})$, we can instead employ a Metropolis-Hastings (MH) strategy and sample a candidate according to some proposal $q(x'_k|y_k, x'_{k-1}, x_{k:k+1})$ and accept it with the usual MH acceptance probability

$$min\left(1, \frac{p(x'_{1:k}, x_{k+1:n}|y_{1:n})q(x_k|y_k, x'_{k-1}, x'_k, x_{k+1})}{p(x'_{1:k-1}, x_{k+1:n}|y_{1:n})q(x'_k|y_k, x'_{k-1}, x_{k:k+1})}\right)$$

$$= min\left(1, \frac{g(y_k|x'_k)f(x_{k+1}|x'_k)f(x'_k|x'_{k-1})q(x_k|y_k, x'_{k-1}, x'_k, x_{k+1})}{g(y_k|x_k)f(x_{k+1}|x_k)f(x_k|x'_{k-1})q(x'_k|y_k, x'_{k-1}, x_{k:k+1})}\right)$$

It is clear that these kernels can be ergodic only if $L = n$ and *all* of the components of $x_{1:n}$ are updated. However, in our context we will typically not use ergodic kernels as this would require sampling an increasing number of variables at each time step. In order to obtain truly online algorithms, we restrict ourselves to updating the variables $X_{n-L+1:n}$ for some fixed or bounded L.

To expand on MH update, it preserves any distribution π specified by an unnormalized density h with respect to a measure μ. There is no restriction on $h(x)$ other than that it actually be an unnormalized density (its normalizing constant is nonzero and finite) and that it can be evaluated, that is, for each x we can calculate $h(x)$. There is no requirement that we be able to do any integrals or know the value of the normalizing constant. In particular, unlike the Gibbs sampler, we do not need to know anything about any conditional distributions of π.

The Metropolis-Hastings update uses an auxiliary transition probability specified by a density $q(x, y)$ called the proposal distribution. For every point x in the state space, $q(x, \cdot)$ is a (normalized) probability density with respect to μ having 2 properties: for each x we can simulate a random variate y having the density $q(x, \cdot)$ and for each x and y we can evaluate the $q(x, y)$. To summarize, this is what we need

1. For each x we can evaluate $h(x)$.
2. For each x and y we can evaluate $q(x, y)$.
3. For each x we can simulate a random variate with density $q(x, \cdot)$ with respect to μ.

There is no necessary connection between the auxiliary density $q(x, y)$ and the density $h(x)$ of the stationary distribution. We can choose any density that we know how to simulate. For example, if the state space is d-dimensional Euclidean space $\mathbb{R}^d$ we could use a multivariate normal proposal density with mean x and variance a constant times the identity. If ϕ denotes a $Normal(0, \sigma^2 I)$ density, then we have $q(x, y) = \phi(y - x)$. We can easily simulate multivariate normal variates and evaluate the density.

The Metropolis-Hastings update then works as follows. The current position is x, and the update changes x to its value at the next iteration.

1. Simulate a random variate y having the density $q(x, \cdot)$.

2. Calculate the "Hastings ratio"

$$R = \frac{h(y)q(y,x)}{h(x)q(x,y)} \tag{107}$$

3. Do "Metropolis rejection": with probability $min(1, R)$ set $x = y$ (otherwise x remains the same).

Note also that the denominator of the Hastings ratio (107) can never be zero if the chain starts at a point where $h(x)$ is nonzero. A proposal y such that $q(x, y) = 0$ occurs with probability zero, and a proposal y such that $h(y) = 0$ is accepted with probability zero. Thus there is probability zero that denominator of the Hastings ratio is ever zero during an entire run of the Markov chain so long as $h(X_1) > 0$. If we do not start in the support of the stationary distribution we have the problem of defining how the chain should behave when $h(x) = h(y) = 0$, that is, how the chain should move when both the current position and the proposal are outside the support of the stationary distribution. The Metropolis-Hastings algorithm says nothing about this. It is a problem that is best avoided by starting at a point where $h(x)$ is positive.

Also note specifically that there is no problem if the proposal is outside the support of the stationary distribution. If $h(y) = 0$, then $R = 0$ and the proposal is always rejected, but this causes no difficulties.

The special case when we use a proposal density satisfying $q(x, y) = q(y, x)$ is called the Metropolis update. In this case the Hastings ratio (107) reduces to the odds ratio

$$R = \frac{h(y)}{h(x)}$$

and there is no need to be able to evaluate $q(x, y)$ only to be able to simulate it.

For our example that uses the Normal distribution as a proposed one, if we choose σ too small, the chain can't get anywhere in any reasonable number of iterations. If σ is chosen ridiculously large, all of the proposals will be so far out in the tail that none will be accepted in any reasonable number of iterations. A rule of thumb is to choose σ such that about 20% of proposals are accepted but this may fail sometimes.

When the state X is a vector $X = (X_1, \ldots, X_d)$, the Metropolis-Hastings update can be done one variable at a time, just like the Gibbs update.

Smoothing MH step

1. sample $v \sim U_{[0,1]}$

2. sample the proposal candidate $x_t^{*(i)} \sim p(x_t|x_{t-1}^{(i)})$

3. if $v \leq min\left\{1, \frac{p(\mathbf{y}_t|\mathbf{x}_t^{*(i)})}{p(\mathbf{y}_t|\tilde{\mathbf{x}}_t^{(i)})}\right\}$

$$\rightarrow \textit{then accept move :} \qquad \mathbf{x}_{0:t}^{(i)} = \left(\tilde{\mathbf{x}}_{0:t-1}^{(i)}, \mathbf{x}_t^{*(i)}\right)$$

$$\rightarrow \textit{else reject move :} \qquad \mathbf{x}_{0:t}^{(i)} = \tilde{\mathbf{x}}_{0:t}^{(i)}$$

Algorithm

1. initialization: $t = 0$

for $i = 1, \ldots, N$ draw the states (particles) $\mathbf{x}_0^{(i)}$ from the prior $p(\mathbf{x}_0)$ and set

$$\begin{aligned}
\bar{\mathbf{x}}_0^{(i)} &= E[\mathbf{x}_0^{(i)}] \\
\mathbf{P}_0^{(i)} &= E\left[\left(\mathbf{x}_0^{(i)} - \bar{\mathbf{x}}_0^{(i)}\right)\left(\mathbf{x}_0^{(i)} - \bar{\mathbf{x}}_0^{(i)}\right)^T\right] \\
\bar{\mathbf{x}}_0^{(i)a} &= E[\mathbf{x}_0^{(i)a}] = \left[(\bar{\mathbf{x}}_0^{(i)})^T \; \mathbf{0} \; \mathbf{0}\right]^T
\end{aligned}$$

$$\mathbf{P}_0^{(i)a} = E\left[\left(\mathbf{x}_0^{(i)a} - \bar{\mathbf{x}}_0^{(i)a}\right)\left(\mathbf{x}_0^{(i)a} - \bar{\mathbf{x}}_0^{(i)a}\right)^T\right] = \begin{pmatrix} \mathbf{P}_0^{(i)} & 0 & 0 \\ 0 & \mathbf{Q} & 0 \\ 0 & 0 & \mathbf{R} \end{pmatrix}$$

2. for $t = 1, 2, \ldots$

2.1. importance sampling step

(a) for $i = 1, \ldots, N$:

— update the particles with the UKF:

* calculate sigma points:

$$\mathcal{X}_{t-1}^{(i)a} = \left(\bar{\mathbf{x}}_{t-1}^{(i)a},\ \bar{\mathbf{x}}_{t-1}^{(i)a} \pm \sqrt{(n_a + \lambda)\mathbf{P}_{t-1}^{(i)a}}\right)$$

* propagate particle into future (time update):

$$\mathcal{X}_{t|t-1}^{(i)x} = f\left(\mathcal{X}_{t-1}^{(i)x},\ \mathcal{X}_{t-1}^{(i)v}\right) \qquad \bar{\mathbf{x}}_{t|t-1}^{(i)} = \sum_{j=0}^{2n_a} W_j^{(m)} \mathcal{X}_{j,t|t-1}^{(i)x}$$

$$\mathbf{P}_{t|t-1}^{(i)} = \sum_{j=0}^{2n_a} W_j^{(c)} \left(\mathcal{X}_{j,t|t-1}^{(i)x} - \bar{\mathbf{x}}_{t|t-1}^{(i)}\right)\left(\mathcal{X}_{j,t|t-1}^{(i)x} - \bar{\mathbf{x}}_{t|t-1}^{(i)}\right)^T$$

$$\mathcal{Y}_{t|t-1}^{(i)} = h\left(\mathcal{X}_{t|t-1}^{(i)x},\ \mathcal{X}_{t-1}^{(i)n}\right) \qquad \bar{\mathbf{y}}_{t|t-1}^{(i)} = \sum_{j=0}^{2n_a} W_j^{(m)} \mathcal{Y}_{j,t|t-1}^{(i)}$$

* incorporate new observation (measurement update):

$$\mathbf{P}_{\tilde{\mathbf{y}}_t\tilde{\mathbf{y}}_t} = \sum_{j=0}^{2n_a} W_j^{(c)} \left(\mathcal{Y}_{j,t|t-1}^{(i)} - \bar{\mathbf{y}}_{t|t-1}^{(i)}\right)\left(\mathcal{Y}_{j,t|t-1}^{(i)} - \bar{\mathbf{y}}_{t|t-1}^{(i)}\right)^T$$

$$\mathbf{P}_{\mathbf{x}_t\mathbf{y}_t} = \sum_{j=0}^{2n_a} W_j^{(c)} \left(\mathcal{X}_{j,t|t-1}^{(i)x} - \bar{\mathbf{x}}_{t|t-1}^{(i)}\right)\left(\mathcal{Y}_{j,t|t-1}^{(i)} - \bar{\mathbf{y}}_{t|t-1}^{(i)}\right)^T$$

$$\mathbf{K}_t = \mathbf{P}_{\mathbf{x}_t\mathbf{y}_t}\mathbf{P}_{\tilde{\mathbf{y}}_t\tilde{\mathbf{y}}_t}^{-1}$$

$$\bar{\mathbf{x}}_t^{(i)} = \bar{\mathbf{x}}_{t|t-1}^{(i)} + \mathbf{K}_t\left(\mathbf{y}_t - \bar{\mathbf{y}}_{t|t-1}^{(i)}\right) \qquad \hat{\mathbf{P}}_t^{(i)} = \mathbf{P}_{t|t-1}^{(i)} - \mathbf{K}_t\mathbf{P}_{\tilde{\mathbf{y}}_t\tilde{\mathbf{y}}_t}\mathbf{K}_t^T$$

— sample $\hat{\mathbf{x}}_t^{(i)} \sim q(\mathbf{x}_t^{(i)}|\mathbf{x}_{0:t-1}^{(i)}, \mathbf{y}_{1:t}) = \mathcal{N}(\bar{\mathbf{x}}_t^{(i)}, \hat{\mathbf{P}}_t^{(i)})$
— set $\hat{\mathbf{x}}_{0:t}^{(i)} \equiv (\mathbf{x}_{0:t-1}^{(i)}, \hat{\mathbf{x}}_t^{(i)})$ and$\hat{\mathbf{P}}_{0:t}^{(i)} \equiv (\mathbf{P}_{0:t-1}^{(i)}, \hat{\mathbf{P}}_t^{(i)})$

(b) for $i = 1, \ldots, N$: evaluate the importance weights up to a normalizing constant

$$w_t^{(i)} \propto \frac{p(\mathbf{y}_t|\hat{\mathbf{x}}_t^{(i)})p(\hat{\mathbf{x}}_t^{(i)}|\mathbf{x}_{t-1}^{(i)})}{q(\hat{\mathbf{x}}_t^{(i)}|\mathbf{x}_{0:t-1}^{(i)}, \mathbf{y}_{1:t})}$$

(c) for $i = 1, \ldots, N$: normalize the importance weights

$$\tilde{w}_t^{(i)} = w_t^{(i)}\left[\sum_{j=1}^{N} w_t^{(j)}\right]^{-1}$$

2.2. selection step (resampling)

Multiply/suppress particles/samples $\left(\hat{\mathbf{x}}_{0:t}^{(i)}, \hat{\mathbf{P}}_{0:t}^{(i)}\right)$ with high/low importance weights $\tilde{w}_t^{(i)}$, respectively, to obtain N random particles $\left(\tilde{\mathbf{x}}_{0:t}^{(i)}, \tilde{\mathbf{P}}_{0:t}^{(i)}\right)$ (approximately distributed according to $p(\mathbf{x}_{0:t}^{(i)}|\mathbf{y}_{1:t})$).

2.3. MCMC step (optional)

Apply a Markov transition kernel with invariant distribution $p(\mathbf{x}_{0:t}^{(i)}|\mathbf{y}_{1:t})$ to obtain $\left(\mathbf{x}_{0:t}^{(i)}, \mathbf{P}_{0:t}^{(i)}\right)$.

2.4. output (expectation)

The output of the algorithm is a set of samples that can be used to approximate the posterior distribution as follows

$$p(\mathbf{x}_{0:t}|\mathbf{y}_{1:t}) \approx \hat{p}(\mathbf{x}_{0:t}|\mathbf{y}_{1:t}) = \frac{1}{N}\sum_{i=1}^{N} \delta_{\mathbf{x}_{0:t}^{(i)}}(d\mathbf{x}_{0:t})$$

One obtains straightforwardly the following estimate of $E[g_t(\mathbf{x}_{0:t})]$

$$E[g_t(\mathbf{x}_{0:t})] = \int g_t(\mathbf{x}_{0:t}) p(\mathbf{x}_{0:t}|\mathbf{y}_{1:t}) d\mathbf{x}_{0:t} \approx \frac{1}{N} \sum_{i=1}^{N} g_t(\mathbf{x}_{0:t}^{(i)})$$

for some function of interest $g_t : (\mathbb{R}^{n_x})^{(t+1)} \to \mathbb{R}^{n_{g_t}}$ integrable with respect to $p(\mathbf{x}_{0:t}|\mathbf{y}_{1:t})$.

Examples of appropriate functions include the marginal conditional mean of $\mathbf{x}_{0:t}$, in which case $g_t(\mathbf{x}_{0:t}) = \mathbf{x}_t$,

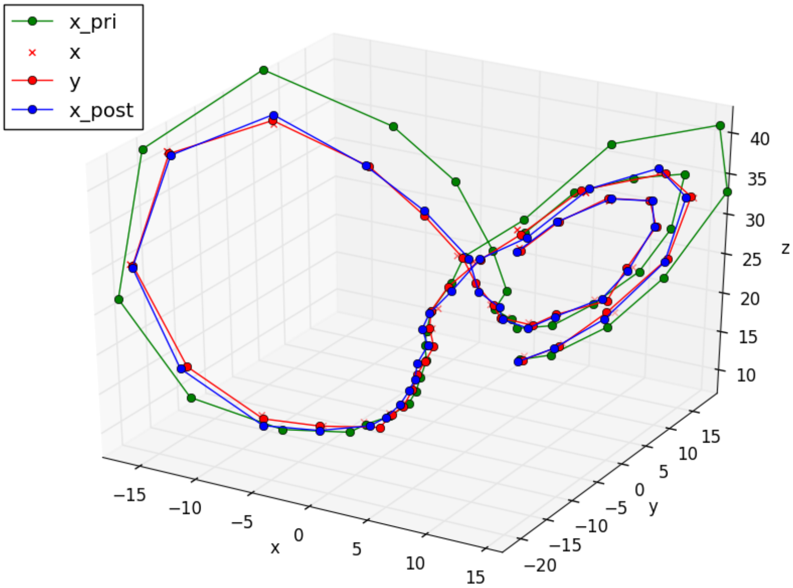
or mode,

or the marginal conditional covariance of $\mathbf{x}_{0:t}$ with $g_t(\mathbf{x}_{0:t}) = \mathbf{x}_t\mathbf{x}_t' - E_{p(\mathbf{x}_t|\mathbf{y}_{1:t})}[\mathbf{x}_t] E'_{p(\mathbf{x}_t|\mathbf{y}_{1:t})}[\mathbf{x}_t]$.

The marginal conditional mean is often the quantity of interest, because it is the optimal MMSE estimate of the current state of the system.

3.1 Lorenz Attractor Example

I use the same ρ, σ *and* β as in the section 2.2. The number of particles used is 200. With the time step 50 times bigger than in UKF, and noises are 10 times bigger a piece of the function looks like



3.2 Bond Mid-Price Estimation

The Kalman filter assumes a linear system with a Gaussian noise. When the system is inherently nonlinear, the first solution is to linearize it through the Taylor expansion as in the Extended Kalman filter. The next improvement is the Unscented Kalman filter (a.k.a. sigma points filter) which uses a clever approach of estimating a probability distribution rather than the function. This achieves a better accuracy. The particle filter removes the assumption of a Gaussian noise; it may even be of unknown distribution. It comes at the computational expense, though.

For bond pricing it presents additional advantages:

- It is well known that prices do not exhibit the normal distribution (fat tails).
- It is not easy to incorporate information about "Traded Away" transactions reported in RFQs (Request For Quote) in Kalman-like filters.

In this model the distance to mid (DTM) is assumed symmetrical and does not depend on the side for simplicity. Although this skewness, e.g. due to a holding position, can be added later.

Our state $\mathbf{x}$ will include d bonds. Instead of a price we'll consider the yield to benchmark (YtB) which is customary for investment grade (IG) bonds. We assume that it follows the Weiner process without a drift

$$d\mathbf{x}_t = \sigma d\mathbf{W}_t \tag{108}$$

where $\mathbf{W}_t$ is a d-dimensional Brownian motion with

$$d\left\langle W_t^i, W_t^j \right\rangle = \rho^{i,j} dt$$

We denote by Σ the covariance matrix for process noise:

$$\Sigma^{ij} = \rho^{i,j} \sigma^i \sigma^j$$

Another improvement to this model would be to consider an Ornstein–Uhlenbeck process $d\mathbf{x}_t = \theta(\mu - \mathbf{x}_t)dt + \sigma d\mathbf{W}_t$, where $\theta > 0$, $\sigma > 0$ (this is also mentioned as the Vasicek model).

Let's introduce another process $\mathbf{z}_t$ following a d-dimensional Ornstein–Uhlenbeck process:

$$d\mathbf{z}_t = -A\mathbf{z}_t dt + V d\mathbf{B}_t \tag{109}$$

where $\mathbf{z}_0$ is given, A and V are $d \times d$ matrices, and $\mathbf{B}_t$ a d-dimensional standard Brownian motion assumed to be independent from the process $\mathbf{W}_t$.

Meucci showed in [33] that

$$\mathbb{E}[\mathbf{z}_{t+\tau} | \mathbf{z}_t] = e^{-A\tau} \mathbf{z}_t$$

and

$$\mathbb{V}[\mathbf{z}_{t+\tau}|\mathbf{z}_t] = \Gamma(\tau)$$

$(\tau > 0)$

$$vec\left(\Gamma(\tau)\right) = \left(A \otimes I_d + I_d \otimes A\right)^{-1}\left(I_{d^2} - exp\left[-\left(A \otimes I_d + I_d \otimes A\right)\tau\right]\right)vec(VV^T)$$

where $vec(\cdot)$ refers to the vectorization operator, i.e.

$$vec\left(\left(M_{i,j}\right)_{1 \le i,j \le d}\right) = \left(M_{1,1}, \ldots, M_{d,1}, \ldots, M_{1,d}, \ldots, M_{d,d}\right)^T$$

In practice, we consider A to be a diagonal matrix. Then we have

$$\Gamma_{ij}(\tau) = \frac{1}{a^i + a^j}\left(1 - e^{-(a^i+a^j)\tau}\right)(VV^T)_{ij}$$

Let's denote the DTM (half bid-ask spread) of asset i as ψ_t^i and define the stochastic process as

$$\psi_t^i = \psi_0^i exp(z_t^i), \qquad \psi_0^i \, given$$

In other words, $x_t^i + \psi_t^i$ and $x_t^i - \psi_t^i$ are the bid-YtB and the ask-YtB, respectively (the bid-YtB has to be higher than the ask-YtB for the bid price to be lower than the ask price). That is, our state for each particle at time t is $\{\mathbf{x}, \psi\}$, vectors $\mathbf{x}$ and ψ of dimension d.

From the viewpoint of a given dealer D the information available to her corresponds to 5 different situations:

		D's observation at time t
1- D2C	A client buys bond i from D at time t.	$y_t^i = x_t^i - \psi_t^i + \epsilon_t^i$
2- D2C	A client sells bond i to D at time t.	$y_t^i = x_t^i + \psi_t^i + \epsilon_t^i$
3- Traded Away	A client buys bond i from another dealer at time t.We assume D has proposed $\check{y}$ but was not chosen because a better price was found elsewhere.	$y_t^i = x_t^i - \psi_t^i + \epsilon_t^i$ for D observation is $1_{y \ge \check{y}}$
4- Traded Away	A client sells bond i to another dealer at time t.We assume D has proposed $\check{y}$ but was not chosen because a better price was found elsewhere.	$y_t^i = x_t^i + \psi_t^i + \epsilon_t^i$ for D observation is $1_{y \le \check{y}}$
5-D2D	Another dealer transacted bond i with D on the inter-dealer broker (IDB) market.	$y_t^i \in [x_t^i - a_t^i + \epsilon_t^i,\, x_t^i + a_t^i + \epsilon_t^i]$ where a_t^i can for instance be chosen $\sim$to ψ_t^i or $\sim$to a typical size for the bid-ask spread, e.g. of CBBT

where $\epsilon_t^i \sim \mathcal{N}\left(0, \sigma_\epsilon^{i^2}\right)$ assumed to be independent of all other random variables (observation noise).

It should be noted that for scenarios 3-5 when calculating the normal probability for likelihood the cumulative distribution function Φ should be used, namely

3	$\Phi\left(-(\breve{y}-x+\psi)/R\right)$
4	$\Phi\left((\breve{y}-x-\psi)/R\right)$
5	$\Phi\left((y-x+a)/R\right)-\Phi\left((y-x-a)/R\right)$

,

and when sampling for $\hat{\mathbf{x}}$ in the importance sampling step and for $\mathbf{x}'$ in the MCMC step, the truncated Gaussian distribution should be used (right-sided, left-sided and two-sided, respectively).

Because we receive an observation for 1 bond at a time, e.g. j^{th}, we apply the filter treatment to this component of $\mathbf{x}$. The rest are updated in accordance with their correlations. For each particle

$$x_{t+1}^{i\neq j}=\begin{pmatrix} x_{t+1}^{1} \\ \vdots \\ x_{t+1}^{j-1} \\ x_{t+1}^{j+1} \\ \vdots \\ x_{t+1}^{d} \end{pmatrix}+\left(x_{t+1}^{j}-x_{t}^{j}\right)\begin{pmatrix} \rho^{j,1}\frac{\sigma^{1}}{\sigma^{j}} \\ \vdots \\ \rho^{j,j-1}\frac{\sigma^{j-1}}{\sigma^{j}} \\ \rho^{j,j+1}\frac{\sigma^{j+1}}{\sigma^{j}} \\ \vdots \\ \rho^{j,d}\frac{\sigma^{d}}{\sigma^{j}} \end{pmatrix}$$

and

$$\varSigma^{i\neq j}=\begin{pmatrix} {\sigma^{1}}^{2} & \cdots & \rho^{1,j-1}\sigma^{1}\sigma^{j-1} & \rho^{1,j+1}\sigma^{1}\sigma^{j+1} & \cdots & \rho^{1,d}\sigma^{1}\sigma^{d} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ \rho^{j-1,1}\sigma^{j-1}\sigma^{1} & \cdots & \rho^{j-1,j-1}\sigma^{j-1}\sigma^{j-1} & \rho^{j-1,j+1}\sigma^{j-1}\sigma^{j+1} & \cdots & \rho^{j-1,d}\sigma^{j-1}\sigma^{d} \\ \rho^{j+1,1}\sigma^{j+1}\sigma^{1} & \cdots & \rho^{j+1,j-1}\sigma^{j+1}\sigma^{j-1} & \rho^{j+1,j+1}\sigma^{j+1}\sigma^{j+1} & \cdots & \rho^{j+1,d}\sigma^{j+1}\sigma^{d} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ \rho^{d,1}\sigma^{d}\sigma^{1} & \cdots & \rho^{d,j-1}\sigma^{d}\sigma^{j-1} & \rho^{d,j+1}\sigma^{d}\sigma^{j+1} & \cdots & \rho^{d,d}\sigma^{d}\sigma^{d} \end{pmatrix}-$$

$$-\begin{pmatrix} \rho^{j,1}\sigma^{1} \\ \vdots \\ \rho^{j,j-1}\sigma^{j-1} \\ \rho^{j,j+1}\sigma^{j+1} \\ \vdots \\ \rho^{j,d}\sigma^{d} \end{pmatrix}\left(\rho^{j,1}\sigma^{1},\ldots,\rho^{j,j-1}\sigma^{j-1},\rho^{j,j+1}\sigma^{j+1},\ldots,\rho^{j,d}\sigma^{d}\right).$$

It is noteworthy that between 2 observations times, one could continue to diffuse particles by using the dynamics given by (108) and (109) to obtain an empirical estimation of the distribution of mid-YtBs and half bid-ask spreads.

The parameters $\varSigma$, A, $V\,and\,\psi_0$ can be either estimated using a fully Bayesian approach (like in the Lorenz attractor example) or calibrated off-line on historical data.

For the covariance matrix $\varSigma$ one can assume that the correlation structure and the volatility levels of the YtBs associated with the CBBT mid-prices are the same as those of our mid-YtBs. Therefore, it is reasonable to estimate Σ on CBBT mid-price data.

Unfortunately, I cannot present results because of proprietorship of market data.

References

[1] Christopher Bishop “Pattern recognition and machine learning”

[2] Mussa-Ivaldi Shadmehr “Biological learning and control”

[3] R. E. Kalman “A New Approach to Linear Filtering and Prediction Problems”, 1960

[4] Simon Haykin “Kalman filtering and neural networks”

[5] Simon Haykin “Neural networks and learning machines”

[6] Charles Geyer “Markov chain Monte Carlo lecture notes”

[7] Paul Atzberger “The Monte-Carlo Method”

[8] Simon Julier and Jeffrey Uhlmann “A general method for approximating nonlinear transformations of probability distributions”

[9] Simon Julier, Jeffrey Uhlmann, Hugh Durrant–Whyte “A new method for the nonlinear transformation of means and covariances in filters and estimators”

[10] Julier, Simon J.; Uhlmann, Jeffrey K. "A new extension of the Kalman filter to nonlinear systems"; 1997

[11] Eric Wan and Rudolph van der Merwe “The Unscented Kalman Filter for Nonlinear Estimation”

[12] van der Merwe, de Freitas, Doucet, Wan ”The unscented particle filter”; 2001

[13] Arnaud Doucet, Adam Johansen “A Tutorial on Particle Filtering and Smoothing: Fifteen years later”

[14] Simon Julier “The scaled unscented transformation”

[15] Grewal, Andrews “Kalman filtering. Theory and practice using Matlab”

[16] Anderson, Moore “Optimal filtering”

[17] Crassidis, Junkins “Optimal estimation of dynamic systems”

[18] Bain, Crisan “Fundamentals of stochastic filtering”

[19] Stephen Marsland “Machine learning. An algorithmic perspective”

[20] Snoek, Larochelle, Adams “Practical Bayesian optimization of machine learning algorithms”

[21] Rasmussen, Wiliams “Gaussian processes for machine learning”

[22] Chow, Ferrer, Nesselroade “An unscented Kalman filter approach to the estimation of nonlinear dynamical systems models”

[23] Klaas, de Freitas, Doucet “Toward Practical N^2 Monte Carlo: the Marginal Particle Filter”

[24] Zhe Chen “Bayesian Filtering: From Kalman Filters to Particle Filters, and Beyond”

[25] anuncommonlab.com, github.com/tuckermcclure/

[26] dynare.org

[27] github.com/zlongshen/ReBEL

[28] cs.ubc.ca/~nando/software.html

[29] stats.ox.ac.uk/~doucet/smc_resources.html

[30] John Hull “Options, futures, and other derivatives”

[31] O. Gueant, J. Pu “Mid-price estimation for European corporate bonds: a particle filtering approach”

[32] Daniel Liberzon “Calculus of variations and optimal control”

[33] A. Meucci “Review of Statistical Arbitrage, Cointegration, and Multivariate Ornstein–Uhlenbeck”; 2009

[34] Peter Forsyth “A Hamilton Jacobi Bellman approach to optimal trade execution”

[35] J. Wang, P. Forsyth “Numerical Solution of the Hamilton-Jacobi-Bellman Formulation for Continuous Time Mean Variance Asset Allocation”

[36] Ulrich Rieder, Nicole Bäuerle “Portfolio optimization with unobservable Markov-modulated drift process”

[37] Dang, Forsyth “Better than pre-commitment mean-variance portfolio allocation strategies: a semi-self-financing Hamilton-Jacobi-Bellman equation approach”

—