Forecasting "High" and "Low" of financial time series by Particle systems and Kalman filters

S. DABLEMONT, S. VAN BELLEGEM, M. VERLEYSEN

Université catholique de Louvain, Machine Learning Group, DICE
3, Place du Levant, B-1348 Louvain-la-Neuve - BELGIUM
Tel : +32 10 47 25 51 - Fax : +32 10 47 25 98
E-mail : {dablemont, verleysen}@dice.ucl.ac.be
vanbellegem@stat.ucl.ac.be

Subjects

Finances, Neural networks, Particle systems, Kalman filters, Forecasting, Tick Data

Abstract

The analysis of financial time series is very useful in the economic world. This paper deals with a data-driven empirical analysis of financial time series.

In this paper we present a forecasting method of the first stopping times, when the prices cross for the first time a "high" or "low" threshold defined by the trader, based on an empirical functional analysis of the past "tick data" of the series, without resampling.

Seuil

An originality of this method is that it does not use a theoretical financial model but a non-parametric space state representation with non-linear RBF neural networks. Modelling and forecasting are made by Particles systems and Kalman filters.

This method can be applied to any forecasting problem of stopping time, but is particularly suited for data showing nonlinear dependencies and observed at irregularly and randomly spaced times like financial time series of «tick data» do.

The method is applied to the forecasting of stopping times of "high" and "low" of financial time series in order to be useful for speculators

1 Introduction

The topic of this paper is the forecasting of the first stopping time, when prices cross for the first time a "high" or "low" threshold defined by the trader, estimated from a trading model, and based on an empirical functional analysis, using a very high frequency data set, as tick by tick asset prices, without resampling.

There is a distinction between a price forecasting and a trading recommendation. A trading model includes a price forecasting, but it must also account for the specific constraints of the dealer of the trading model because it is constrained by the trading history and the positions to which it is committed. A trading model thus goes beyond a price forecasting so that it must decide if and at what time a certain action has to be taken.

This trading model should give, at time t, a signal of buying or selling an asset, and depending on the trader's decision, when the price of this asset crosses for the first time a threshold defined by the trader, in order to close his position with a yielding profit, also it should be able to send a stop-loss order.

The paper is organized as follows, the models we use are presented in section 2. Bayesian Estimation is introduced in section 3. Gaussian Bayesian Estimation is introduced in section 4 in the framework of the Kalman filters, and Non-Gaussian Bayesian Estimation in section 5 in the framework of the Particle filters. State, Parameters and Dual estimations are introduced in section 6. Tests and results are shown in section 7, and the Conclusions in section 8.

2 Methodology

2.1 Basic Models

The theory of finance is mainly treated in term of stochastic differential equations such as, the value of a stock price S is supposed to follow a diffusion geometrical Wiener process

$$dS_t = S_t \left(\mu_t dt + \sqrt{V_t} dB_t\right) \tag{1}$$

where μ_t is the drift function, and V_t is the volatility of the stock price. For a stochastic volatility model, the variance function is also modeled as Brownian motion,

$$dV_t = \alpha(S,t)dt + \beta(S,t)dZ_t \tag{2}$$

where B_t and Z_t are correlated Brownian motions, and the form of V_t depends on the particular stochastic volatility model under study, and where $\alpha(S,t)$ and $\beta(S,t)$ are some functions of V_t (Andersen et al., 2002).

2.2 Dynamic State-Space stochastic volatility Model

If we take the logarithms of stock price, $y_t = \log(S_t)$ and of the volatility, $h_t = \log(V_t)$, and using the Itô's formula we derive the process in a continuous dynamic state-space formulation

$$\begin{aligned} dy_t &= \left[\mu_t - \frac{1}{2}V_t\right] dt + \sqrt{V_t}\, dB_t && (3) \\ dh_t &= \kappa\left[\theta - V_t\right]dt + \xi V_t^p dZ_t && (4) \end{aligned}$$

where κ, θ, and ξ are fixed constants, and $p = \frac{1}{2}$ for a Heston model, $p = 1$ for a Garch model, and $p = \frac{3}{2}$ for a $\frac{3}{2}$ model.

2.3 Parameters Estimation

Maximum Likelihood estimation (MLE) is an approach to parameter estimation, and its statistical efficiency is well known. It involves constructing a log-likelihood function that relates the unknown parameters to the observations, and maximizing it to obtain the parameters. But the likelihood function in diffusion-type volatility models is extremely difficult to evaluate. Markov property is lost due to stochastic volatility, therefore it is only possible to simulate the likelihood function (Ait-Sahalia, 2002).

Bayesian methods are easier, because inference can be jointly made for the augmented parameter vector including the latent variable as an element.

2.4 Discrete Dynamic State Space Model

In practice, observations can be only made at random discrete time intervals and we must approximate the continuous time stochastic volatility models by a system of stochastic difference equations

$$\begin{aligned}\log V_{t+1} &= \log V_t + \frac{1}{V_t}\big[\kappa(\theta - V_t) - \frac{1}{2}\xi^2 V_t^{2p-1} - \rho\xi V_t^{p-\frac{1}{2}}(\mu - \frac{1}{2}V_t)\big]\Delta t + \\ &\quad \rho\xi V_t^{p-\frac{3}{2}}\big(\ln S_t - \ln S_{t-1}\big) + \xi V_t^{p-1}\sqrt{\Delta t}\sqrt{1-\rho}Z_t \qquad (5) \\ \ln S_t &= \ln S_{t-1} + \big(\mu - \frac{1}{2}V_t\big)\Delta t + \sqrt{\Delta t}\sqrt{V_t}B_t, \qquad (6)\end{aligned}$$

we have got a *Dynamic State Space Model* (DSSM), where Equation 5 is the nonlinear state equation, and Equation 6 is the nonlinear observation equation.

2.5 Non-parametric Dynamic State-Space Model

In our case, we would like forecast the future curve for a couple of hours and not only the next value of the time series. Thus, we can not use the parametric Dynamic State Space Model, but based on its development, we derive a non-parametric Dynamic State Space Model where random variables may be scalars, vectors or curves.

$$\begin{aligned}\mathbf{x}_{k+1} &= \mathbf{f}\big(\mathbf{x}_k, \mathbf{u}_k, \mathbf{v}_k\big) \qquad (7) \\ \mathbf{y}_k &= \mathbf{h}\big(\mathbf{x}_k, \mathbf{u}_k, \mathbf{n}_k\big)\ , \qquad (8)\end{aligned}$$

where $\mathbf{x}_k$ is the state, $\mathbf{y}_k$ is the observation, $\mathbf{u}_k$ is an exogenous input, $\mathbf{v}_k$ is the process noise, and $\mathbf{n}_k$ is the measurement noise. These noises may be non-Gaussian. The functions $\mathbf{f}$ and $\mathbf{h}$ will be nonlinear and we will use a non-parametric representation by *Radial Basis Functions* (RBF) because they possess the universal approximation property (Haykin, 1999). But, in this case, the random states variables have not got economic signification any longer.

2.6 How to realize the forecasting

The basic idea underlying this paper is to use a *dynamic state-space model* (DSSM) with nonlinear, non-parametric equations, as *Radial Basis Function* (RBF), and to use the framework of the *Particle filters* (PF) combined with the *Unscented Kalman filters* (UKF) to estimate the parameters of the model as well as the hidden variables.

2.7 Which tools do we use for the forecasting

The *Kalman filter* (KF) can not be used for this analysis since the functions are nonlinear and the transition density of the state space is non-Gaussian. But with the advent of new estimation methods such as *Markov Chain Monte Carlo* (MCMC) and *Particle filters* (PF), exact estimation tools for nonlinear state-space and non-Gaussian random variables became available.

Particle filters are now widely employed in the estimation of models for financial markets, in particular for stochastic volatility models (Pitt and Shephard, 1999) and (Lopes and Marigno, 2001), applications to macroeconometrics, for the analysis of general equilibrium models (Villaverde and Ramirez, 2004a) and (Villaverde and Ramirez, 2004b), the extraction of latent factors in business cycle analysis (Billio, and al., 2004). The main advantage of these techniques lies in their great flexibility when treating nonlinear dynamic models with non-Gaussian noises, which can not be handled through the traditional Kalman filter.

3 Bayesian Estimation

3.1 Introduction

Suppose we have the data $x = \big(x_1, \cdots, x_n\big)$ with distribution $p(x|\theta)$ where θ is the unknown parameter we want to estimate. The basic idea of the Bayesian approach is to treat the parameter θ as a random variable and to use an *a priori* knowledge of the distribution $\pi(\theta)$ of θ and then to estimate θ by calculating the *a posteriori* distribution $\pi(\theta|x)$ of θ.

The one-dimension case.
In the one-dimensional case the *a posteriori* distribution $\pi(\theta|x)$ of θ is calculated by the so called *Bayes's formula* using the *a priori* distribution $\pi(\theta)$ as follows

$$\pi(\theta|x) = \frac{p(x|\theta)\ \pi(\theta)}{\int\ p(x|\theta)\ \pi(\theta)\ d\theta} \tag{9}$$

where the denominator is a proportionality constant making the total a posteriori probability equal to one. Now by using the a posteriori distribution $\pi(\theta|x)$ the parameter θ can be estimated by the mean $\hat{\theta} = E\Big[\pi(\theta|x)\Big]$.

The multi-dimension case.
In the multi-dimensional case $\theta = (\theta_1, \cdots, \theta_k)$, the a posteriori distribution of θ can be calculated by the Bayes formula as follow

$$\pi(\theta|x) = \frac{p(x|\theta)\ \pi(\theta)}{\int \cdots \int\ p(\theta|x)\ \pi(\theta)\ d\theta_1 \cdots \theta_k}\ . \tag{10}$$

By using the marginal distribution $\pi(\theta_i|x)$ of the joint a posteriori distribution $\pi(\theta|x)$

$$\pi(\theta_i|x) = \int \cdots \int\ \pi(\theta|x)\ d\theta_1 \cdots d\theta_{i-1}\ d\theta_{i+1} \cdots d\theta_k \tag{11}$$

we are able to estimate θ by the ways described in the one-dimensional case. Usually problems arise in calculating the integrals in Equation 11 which require approximation techniques as *Markov Chain Monte Carlo* methods (MCMC).

3.2 Markov Chain Monte Carlo methods (MCMC)

Suppose we want to generate a sample from an a posteriori distribution $\pi(\theta|x)$ for $\theta \in \Theta \subseteq R_k$ but we can not directly do this. However, suppose we are able to construct a Markov chain with state space Θ and with distribution $\pi(\theta|x)$. Then under suitable regularity conditions asymptotic results exist, showing in which case the sample output from such a chain with distribution $\pi(\theta|x)$ can be used to mimic a random sample from $\pi(\theta|x)$ or to estimate the expected value of a function $f(\theta)$ with respect to $\pi(\theta|x)$. If $\theta^1, \cdots, \theta^k, \cdots$ is a realization from a suitable chain then $\theta^k \rightarrow \theta$ in distribution as k tends to infinity, $\theta \approx \pi(\theta|x)$ and $\frac{1}{k}\sum_{i=1}^{k} f\left(\theta^i\right) \rightarrow E_{\theta|x}\left[f\left(\theta\right)\right]$a.s. as k tends to infinity.

3.3 Why do we use a Bayesian representation :

Many real-world applications require estimation unknown quantities from some given observations at each time step. In most of the applications, even though the dynamics of the system are not known exactly, prior knowledge about the phenomenon being modelling is generally available to construct a suitable model. In the Bayesian modelling, prior distributions for the states and likelihood functions relating these states to the observations are derived from the model. Within this context, estimates of the states is based on the posterior distribution obtained from *Bayes's theorem.* In order to avoid storing the complete data one is interested in performing inference on-line with recursive filters suitable for this task. These filters consist essentially of a *prediction step*, where the state is predicted for the next time step according to the dynamical model, and an *update step*, where the prediction is updated according to the latest observation.

3.4 Limitations of Kalman filters :

If the data are modeled by a linear Gaussian state-space model, then the *Kalman filter* (Kalman, 1960) is the optimal filter in order to minimise the mean square error between the true state and its estimate. For partially observed linear systems, the *Hidden Markov Model* (HMM), gives the solution. However, for many practical applications, linear models or the assumption of Gaussian noise, are not plausible. Various filters, such as Extended Kalman Filter(EKF) (Anderson, and More, 1979) , Unscented Kalman Filter (UKF) (Julier, 1997) , Gaussian Sum approximations (Alspach et al., 1972) have been developed to deal with this problem. But they are only suboptimal solutions since they only approximate the nonlinearity and the non-Gaussianity of the model.

3.5 Why do we use Particle filters :

Sequential Monte Carlo (SMC), or *Particle filters* (Doucet, de Freitas, and Gordon, 2001) methods are recursive Bayesian filters which provide a convenient and attractive approach to approximate the posterior distributions when the model is nonlinear and when the noises are not Gaussian. These techniques provide general solutions to many problems, where linearisation and Gaussian approximations are intractable or would yield too low performances. Non-Gaussian noise assumptions and incorporation of constraints on the state variables can also be performed in a natural way. Moreover, SMC methods are very flexible, easy to implement, parallelizable and applicable in very general settings (Del Moral, 2004).

4 Gaussian Bayesian Estimation

4.1 Introduction

We will be addressing the *sequential recursive probabilistic inference* problem within discrete-time non-linear dynamic systems that can be described by a *dynamic state-space model.* The hidden system state $\mathbf{x}_k$, with initial probability density $p(\mathbf{x}_0)$, evolves over time as a partially observed *first order Markov process* according to the conditional probability density $p(\mathbf{x}_k|\mathbf{x}_{k-1})$. The observations $\mathbf{y}_k$ are conditionally independent given the state and are generated according to the conditional probability density $p(\mathbf{y}_k|\mathbf{x}_k)$. The evolution of the state sequence is given by the *Transition* equation :

$$\mathbf{x}_k \quad = \quad \mathbf{f}_k(\mathbf{x}_{k-1}, \mathbf{u}_{k-1}, \mathbf{v}_k; \mathbf{w}) \ , \tag{12}$$

and the *Measurement* equation is given by :

$$\mathbf{y}_k \quad = \quad \mathbf{h}_k(\mathbf{x}_k, \mathbf{n}_k; \mathbf{w}) \ , \tag{13}$$

where :

- $\mathbf{x}_k$: the state vector at the discrete time index k,
- $\mathbf{y}_k$: the measurement vector,
- $\mathbf{u}_k$: an exogenous input of the system, assumed known,
- $\mathbf{v}_k$: the process noise that drives the dynamic system,
- $\mathbf{n}_k$: the measurement noise corrupting the observation of the state,
- $\mathbf{f}_k$: a time-variant, linear or non-linear function,
- $\mathbf{h}_k$: a time-variant, linear or non-linear function.
- $\mathbf{w}$: the parameters vector.

The state transition density $p(\mathbf{x}_k|\mathbf{x}_{k-1})$ is fully specified by $\mathbf{f}_k$ and the process noise distribution $p(\mathbf{v}_k)$, whereas $\mathbf{h}_k$ and the observation noise distribution $p(\mathbf{n}_k)$ fully specify the observation likelihood $p(\mathbf{y}_k|\mathbf{x}_k)$. The problem of sequential probabilistic inference can be framed as follow : How do we estimate the hidden variables in a recursive fashion as noisy observations becomes available online?

The process of recursive filtering is defined as calculating an *a priori* estimate of the state $\widehat{\mathbf{x}}_{k|k-1}$ given the observation information $\{\mathbf{y}_{1:k-1}\}$

$$\widehat{\mathbf{x}}_{k|k-1} \quad = \quad E[\mathbf{x}_k|\mathbf{y}_{1:k-1}] = \int \mathbf{x}_k \ p(\mathbf{x}_k|\mathbf{y}_{1:k-1}) \ d\mathbf{x}_k \ . \tag{14}$$

After the observation $\mathbf{y}_k$ has become available, the *a posteriori* estimate $\widehat{\mathbf{x}}_{k|k}$ is made by

$$\widehat{\mathbf{x}}_{k|k} \quad = \quad E[\mathbf{x}_k|\mathbf{y}_{1:k}] = E[\mathbf{x}_k|\mathbf{y}_k] \ , \tag{15}$$

where the latter equals sign comes from the fact that the process equation is a first order Markov process.

The posterior at time $(k-1)$, $p(\mathbf{x}_{k-1}|\mathbf{y}_{1:k-1})$, is first projected forward in time in order to calculate *the prior* at time k. In terms of probability distributions, this means that the prior distribution is obtained by

$$\begin{aligned} p(\mathbf{x}_{k|k-1}|\mathbf{y}_{1:k-1}) &= \int p(\mathbf{x}_k|\mathbf{x}_{k-1}, \mathbf{y}_{1:k-1})\, p(\mathbf{x}_{k-1}|\mathbf{y}_{1:k-1})\, d\mathbf{x}_{k-1} && \text{(16a)} \\ &= \int p(\mathbf{x}_k|\mathbf{x}_{k-1})\, p(\mathbf{x}_{k-1}|\mathbf{y}_{1:k-1})\, d\mathbf{x}_{k-1}\ , && \text{(16b)} \end{aligned}$$

whereas it is assumed that the initial prior of the state vector $p(\mathbf{x}_0|\mathbf{y}_0) \equiv p(\mathbf{x}_0)$ is available.

Next, the latest noisy measurement is incorporated using the observation *likelihood* to generate the updated *posterior*. When the observation $\mathbf{y}_k$ becomes available via Bayes' rule we have got :

$$\begin{aligned} p(\mathbf{x}_k|\mathbf{y}_{1:k}) &= \frac{p(\mathbf{y}_k|\mathbf{x}_k)\, p(\mathbf{x}_k|\mathbf{y}_{1:k-1})}{p(\mathbf{y}_k|\mathbf{y}_{1:k-1})} && \text{(17)} \\ p(\mathbf{y}_k|\mathbf{y}_{1:k-1}) &= \int p(\mathbf{y}_k|\mathbf{x}_k)\, p(\mathbf{x}_k|\mathbf{y}_{1:k-1}) d\mathbf{x}_k\ , && \text{(18)} \end{aligned}$$

and the state transition *prior* and the observation *likelihood* densities are given by

$$\begin{aligned} p(\mathbf{x}_k|\mathbf{x}_{k-1}) &= \int \delta(\mathbf{x}_k - \mathbf{f}(\mathbf{x}_{k-1}, \mathbf{u}_k, \mathbf{v}_k; \mathbf{w}))\, p(\mathbf{v}_k)\, d\mathbf{v}_k && \text{(19)} \\ p(\mathbf{y}_k|\mathbf{x}_k) &= \int \delta(\mathbf{y}_k - \mathbf{h}(\mathbf{x}_k, \mathbf{u}_k, \mathbf{n}_k; \mathbf{w}))\, p(n_k)\, d\mathbf{n}_k\ , && \text{(20)} \end{aligned}$$

where $\delta(.)$ is the Dirac function. This way, the *posterior* density is computed from the *prior* density.

The above equations describe the optimal Bayesian solution which can, in general case, not be calculated analytically. Solutions only exist under certain restrictions, as is the case for the Kalman filter. In addition, Extended Kalman filter, Unscented Kalman filter and Particle filters approximate the optimal Bayesian solution when there is no analytical solution.

4.2 Kalman Filter (KF)

A misconception about the Kalman framework is that it requires the state space to be linear as well as all probability densities to be Gaussian. This is in fact incorrect. In the original derivation of the Kalman filter we only have the following assumptions (Kalman, 1960):

- consistent minimum variance estimates of the random variables, thus the posterior state distribution can be calculated by maintaining only their first and second order moments;
- the estimator (measurement update) itself is a linear function of the prior knowledge of the system, summarized by $p(\mathbf{x}_k|\mathbf{y}_{1:k-1})$
- accurate predictions of the state and of the system observations can be calculated. These predictions are needed to approximate the first and second order moments of $p(\mathbf{x}_k|\mathbf{y}_{1:k-1})$ and $p(\mathbf{y}_k|\mathbf{x}_k)$.

Based on these assumptions, Kalman derived the following recursive form of the optimal Gaussian approximate linear Bayesian update of the conditional mean of the state $\widehat{\mathbf{x}}_k = E[\mathbf{x}_{k|k}|\mathbf{y}_{1:k})$ and its covariance $\mathbf{P}\widehat{\mathbf{x}}_{k|k}$:

$$\begin{aligned} \widehat{\mathbf{x}}_{k|k-1} &= (\text{prediction of } \mathbf{x}_k) + \mathbf{K}_k(\mathbf{y}_k - (\text{prediction of } \mathbf{y}_k)) \\ &= \widehat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k(\mathbf{y}_k - \widehat{\mathbf{y}}_{k|k-1}) \qquad (21) \\ \mathbf{P}_{\widehat{\mathbf{x}}_{k|k}} &= \mathbf{P}_{\widehat{\mathbf{x}}_{k|k-1}} - \mathbf{K}_k \mathbf{P}_{\widetilde{\mathbf{y}}_k} \mathbf{K}_k^T \; . \qquad (22) \end{aligned}$$

The optimal terms in this recursion are given by :

$$\begin{aligned} \widehat{\mathbf{x}}_{k|k-1} &= E[\mathbf{f}(\mathbf{x}_{k-1}, \mathbf{v}_{k-1}, \mathbf{u}_k)] \qquad (23) \\ \widehat{\mathbf{y}}_{k|k-1} &= E[\mathbf{h}(\widehat{\mathbf{x}}_{k|k-1}, \mathbf{n}_k)] \qquad (24) \\ \mathbf{K}_k &= E[(\mathbf{x}_k - \widehat{\mathbf{x}}_{k|k-1})(\mathbf{x}_k - \widehat{\mathbf{x}}_{k|k-1})^T] E[(\mathbf{y}_k - \widehat{\mathbf{y}}_{k|k-1})(\mathbf{y}_k - \widehat{\mathbf{y}}_{k|k-1})^T]^{-1} \\ &= \mathbf{P}_{\widehat{\mathbf{x}}_{k|k}\widetilde{\mathbf{y}}_{k|k-1}} \mathbf{P}^{-1}_{\widetilde{\mathbf{y}}_{k|k-1}} \; , \qquad (25) \end{aligned}$$

where the optimal prediction $\widehat{\mathbf{x}}_{k|k-1}$ corresponds to the expectation of a nonlinear function $\mathbf{f}$ of the random function variables $\mathbf{x}_{k-1}$ and $\mathbf{v}_{k-1}$, and the optimal prediction $\widehat{\mathbf{y}}_{k|k-1}$ corresponds to the expectation of a nonlinear function $\mathbf{h}$ of the random function variables $\mathbf{x}_k$ and $\mathbf{n}_k$ taken over the prior distribution of the state at time k. The gain K_k is a function of the expected covariance matrix of the state prediction error and the observation prediction error, and the expected auto-correlation matrix of the innovations. As a consequence, even for nonlinear, non-Gaussian systems, the Kalman filter framework is still the minimum variance optimal Gaussian approximate linear estimator.

4.3 Linear Kalman Filter (LKF)

The assumption of the Kalman filter is that the posterior density is Gaussian at every time step. If $p(\mathbf{x}_k|\mathbf{y}_{1:k-1})$ is Gaussian, also $p(\mathbf{x}_k|\mathbf{y}_{1:k-1})$ will be Gaussian if :

- $\mathbf{v}_k$ and $\mathbf{n}_k$ are Gaussian,
- $\mathbf{x}_k = \mathbf{f}(\mathbf{x}_{k-1}, \mathbf{u}_{k-1}, \mathbf{v}_k; \mathbf{w})$ is a linear function of $\mathbf{x}_{k-1}$, $\mathbf{u}_{k-1}$ and $\mathbf{v}_k$,
- $\mathbf{y}_k = \mathbf{h}(x_k, \mathbf{n}_k; \mathbf{w})$ is a linear function of $\mathbf{x}_k$ and $\mathbf{n}_k$.

4.3.1 State space representation

We have the system :

$$\begin{aligned} \mathbf{x}_{k+1} &= \mathbf{A}_k\mathbf{x}_k + \mathbf{B}_k\mathbf{u}_k + \mathbf{G}_k\mathbf{v}_k \qquad (26) \\ \mathbf{y}_k &= \mathbf{C}_k\mathbf{x}_k + \mathbf{n}_k \; , \qquad (27) \end{aligned}$$

where :

- $\mathbf{x}_k \in R^n$, $\mathbf{y}_k \in R^q$, $\mathbf{u}_k \in R^m$, $\mathbf{v}_k \in R^p$, $\mathbf{n}_k \in R^q$,
- $\mathbf{A}_k$ $[n,n]$, $\mathbf{B}_k$ $[n,m]$, $\mathbf{G}_k$ $[n,p]$, $\mathbf{C}_k$ $[q,n]$, these matrix may be time-variant but are known,
- $\{\mathbf{u}_k\}$ is a deterministic sequence,
- $\{\mathbf{v}_k\}$, $\{\mathbf{n}_k\}$ process and measurement noise sequences; with $\mathbf{v}_k \sim N(0, \mathbf{Q}_k)$, $\mathbf{n}_k \sim N(0, \mathbf{R}_k)$,
- $\mathbf{x}_0 \sim N(\overline{\mathbf{x}_0}, \mathbf{P}_0)$.

4.3.2 Algorithm

Initialization

$$
\begin{aligned}
\widehat{\mathbf{x}}_0 &= E\{\mathbf{x}_0\} = \overline{\mathbf{x}_0} \\
\mathbf{P}_0 &= E[(\mathbf{x}_0 - \widehat{\mathbf{x}}_0)(\mathbf{x}_0 - \widehat{\mathbf{x}}_0)^T] \\
\mathbf{R} &= E[(\mathbf{v} - \overline{\mathbf{v}})(\mathbf{v} - \overline{\mathbf{v}})^T] \\
\mathbf{Q} &= E[(\mathbf{n} - \overline{\mathbf{n}})(\mathbf{n} - \overline{\mathbf{n}})^T]
\end{aligned}
$$

for $k = 1, 2, \cdots$ N
Prediction step
Compute the predicted state mean and covariance (time update)

$$
\begin{aligned}
\widehat{\mathbf{x}}_{k|k-1} &= \mathbf{A}_{k-1}\widehat{\mathbf{x}}_{k-1|k-1} + \mathbf{B}_{k-1}\mathbf{u}_{k-1} && (28) \\
\mathbf{P}_{k|k-1} &= \mathbf{A}_{k-1}\mathbf{P}_{k-1|k-1}\mathbf{A}_{k-1}^T + \mathbf{G}_{k-1}\mathbf{Q}_{k-1}\mathbf{G}_{k-1}^T && (29)
\end{aligned}
$$

Correction step
Update estimates with latest observation (measurement update)

$$
\begin{aligned}
\mathbf{\Sigma}_{k|k-1} &= \mathbf{C}_k\mathbf{P}_{k|k-1}\mathbf{C}_k^T + \mathbf{R}_k && (30) \\
\mathbf{K}_k &= \mathbf{P}_{k|k-1}\mathbf{C}_k^T\mathbf{\Sigma}_{k|k-1}^{-1} && (31) \\
\mathbf{e}_k &= \mathbf{y}_k - \mathbf{C}_k\widehat{\mathbf{x}}_{k|k-1} && (32) \\
\widehat{\mathbf{x}}_{k|k} &= \widehat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k\mathbf{e}_k && (33) \\
\mathbf{P}_{k|k} &= \mathbf{P}_{k|k-1} - \mathbf{K}_k\mathbf{C}_k\mathbf{P}_{k|k-1} && (34)
\end{aligned}
$$

When the above assumptions hold, the Kalman filter is the optimal solution to the problem. For this reason, the Kalman filter is the minimum variance estimator. (Anderson, and More, 1979)

4.4 Extended Kalman Filter (EKF)

When the strict assumptions of the Kalman filter do not hold, approximate filters must be used. The Extended Kalman filter (EKF) assumes that the posterior density $p(\mathbf{x}_k|\mathbf{y}_{1:k})$ is approximated by a Gaussian distribution. However, the system and/or the measurement equation are no longer linear and must be linearized by computing the *Jacobian* matrix. After linearisation, the equations for the Linear Kalman filter can be used.

4.4.1 State space representation

We have the system :

$$
\begin{aligned}
\mathbf{x}_{k+1} &= \mathbf{f}_k(\mathbf{x}_k, \mathbf{u}_k, \mathbf{v}_k; \mathbf{w}) && (35) \\
\mathbf{y}_k &= \mathbf{h}_k(\mathbf{x}_k, \mathbf{n}_k; \mathbf{w}) \ . && (36)
\end{aligned}
$$

4.4.2 Algorithm

Initialization

$$
\begin{aligned}
\widehat{\mathbf{x}}_0 &= E\{\mathbf{x}_0\} = \overline{\mathbf{x}_0} \\
\mathbf{P}_0 &= E[(\mathbf{x}_0 - \widehat{\mathbf{x}}_0)(\mathbf{x}_0 - \widehat{\mathbf{x}}_0)^T] \\
\mathbf{R} &= E[(\mathbf{v} - \overline{\mathbf{v}})(\mathbf{v} - \overline{\mathbf{v}})^T] \\
\mathbf{Q} &= E[(\mathbf{n} - \overline{\mathbf{n}})(\mathbf{n} - \overline{\mathbf{n}})^T]
\end{aligned}
$$

for $k = 1, 2, \cdots$ N

Prediction step

Compute the process model Jacobians :

$$\begin{aligned} \mathbf{F}_k &= \nabla_{\mathbf{x}} \mathbf{f}(\mathbf{x}, \overline{\mathbf{v}}, \mathbf{u}_k)|_{\mathbf{x}=\widehat{\mathbf{x}}_{k-1}} && (37) \\ \mathbf{G}_k &= \nabla_{\mathbf{v}} \mathbf{f}(\widehat{\mathbf{x}}_{\mathbf{k-1}}, \mathbf{v}, \mathbf{u}_k)|_{\mathbf{v}=\widehat{\mathbf{v}}} && (38) \end{aligned}$$

Compute the predicted state mean and covariance (time update)

$$\begin{aligned} \widehat{\mathbf{x}}_{k|k-1} &= \mathbf{f}(\widehat{\mathbf{x}}_{k|k-1}, \overline{\mathbf{v}}, \mathbf{u}_k) && (39) \\ \mathbf{P}_{k|k-1} &= \mathbf{F}_k \mathbf{P}_{k-1|k-1} \mathbf{F}_k^T + \mathbf{G}_k \mathbf{R} \mathbf{G}_k^T && (40) \end{aligned}$$

Correction step

Compute the observation model Jacobians :

$$\begin{aligned} \mathbf{H}_k &= \nabla \mathbf{h}(\mathbf{x}, \mathbf{n})|_{\mathbf{x}=\widehat{\mathbf{x}}_{k|k-1}} && (41) \\ \mathbf{D}_k &= \nabla \mathbf{h}(\mathbf{x}_{k|k-1}, \mathbf{n})|_{\mathbf{n}=\overline{\mathbf{n}}} && (42) \end{aligned}$$

Update estimates with latest observation (measurement update)

$$\begin{aligned} \mathbf{\Sigma}_{k|k-1} &= \mathbf{H}_k \mathbf{P}_{k|k-1} \mathbf{H}_k^T + \mathbf{D}_k \mathbf{R} \mathbf{D}_k^T && (43) \\ \mathbf{K}_k &= \mathbf{P}_{k|k-1} \mathbf{H}_k^T \mathbf{\Sigma}_{k|k-1}^{-1} && (44) \\ \mathbf{e}_k &= \mathbf{y}_k - \mathbf{h}(\widehat{\mathbf{x}}_{k|k-1}, \overline{\mathbf{n}}) && (45) \\ \widehat{\mathbf{x}}_{k|k} &= \widehat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k \mathbf{e}_k && (46) \\ \mathbf{P}_{k|k} &= \mathbf{P}_{k|k-1} - \mathbf{K}_k \mathbf{H}_k \mathbf{P}_{k|k-1} && (47) \end{aligned}$$

Clearly these approximations will only valid if all the higher order derivatives of the nonlinear functions are effectively zero over the uncertainly region of $\mathbf{x}$, as summarized by the support of its prior distribution. In many cases, the EKF calculated mean will be biased and the posterior covariance will be under estimated.

4.5 Unscented Kalman Filter (UKF)

The Unscented Kalman filter is an approach first introduced by (Julier, 1997) for Kalman filtering in the case of nonlinear equations, based onto the intuition : " *With a fixed number of parameters it should be easier to approximate a Gaussian distribution than it is to approximate an arbitrary function* " .

They also approximate the posterior density $p(\mathbf{x}_k|\mathbf{y}_{1:k})$ with a Gaussian, but compared to EKF, which use Jacobians, UKF approximates the distribution of the state variable by using an unscented transformation (Julier, 1997).

Let the propagation of a L dimensional random variable $\mathbf{x}$ through an arbitrary function $\mathbf{y} = \mathbf{g}(\mathbf{x})$. Assume $\mathbf{x}$ has mean $\overline{\mathbf{x}}$ and covariance $\mathbf{P}_{\mathbf{x}}$. To calculate the first two moments of $\mathbf{y}$ we form a set of $2L + 1$ *sigma-points*, $S_i = \{w_i, \mathcal{X}_i\}$ deterministically calculated using the mean and square-root decomposition of the covariance matrix of the prior random variable $\mathbf{x}$, such as :

$$\begin{aligned} \boldsymbol{\mathcal{X}}_{\mathbf{0}} &= \overline{\mathbf{x}}, && (48) \\ \boldsymbol{\mathcal{X}}_{\boldsymbol{i}} &= \overline{\mathbf{x}} + \zeta(\sqrt{\mathbf{P}_{\mathbf{x}}})_i, \quad i = 1, \cdots, L \, , && (49) \\ \boldsymbol{\mathcal{X}}_{\boldsymbol{i}} &= \overline{\mathbf{x}} - \zeta(\sqrt{\mathbf{P}_{\mathbf{x})}})_i, \quad i = L+1, \cdots, 2L \, , && (50) \end{aligned}$$

where ζ is a scaling factor that determines the spread of the sigma-points around $\overline{\mathbf{x}}$ and $(\mathbf{P_x})_i$ indicates the column i of the matrix square-root of the covariance matrix $\mathbf{P_x}$. Each sigma-point is than propagated through the nonlinear function,

$$\boldsymbol{\mathcal{Y}}_{\boldsymbol{i}} = \mathbf{g}(\boldsymbol{\mathcal{X}}_i), \quad i = 0, \cdots, 2L \tag{51}$$

to give the mean $\overline{\mathbf{y}}$, covariance $\mathbf{P_y}$, and cross-covariance $\mathbf{P_{xy}}$, using a weighted sample mean and covariance of the posterior sigma-points,

$$\begin{aligned}
\overline{\mathbf{y}} &\approx \sum_{i=0}^{2L} w_i^m \, \boldsymbol{\mathcal{Y}}_i && (52)\\
\mathbf{P_y} &\approx \sum_{i=0}^{2L}\sum_{j=0}^{2L} w_{ij}^c \, \boldsymbol{\mathcal{Y}}_i \, \boldsymbol{\mathcal{Y}}_j^T && (53)\\
\mathbf{P_{xy}} &\approx \sum_{i=0}^{2L}\sum_{j=0}^{2L} w_{ij}^c \, \boldsymbol{\mathcal{X}}_i \, \boldsymbol{\mathcal{Y}}_j^T && (54)
\end{aligned}$$

where w_i^m and w_{ij}^c are scalar weights.

The method presented in this section is based on : (Wan and van der Merwe, 2000), and (Wan and van der Merwe, 2001)

4.5.1 State space representation

We have the system :

$$\begin{aligned}
\mathbf{x}_{k+1} &= \mathbf{f}_k(\mathbf{x}_k, \mathbf{v}_k) && (55)\\
\mathbf{y}_k &= \mathbf{h}_k(\mathbf{x}_k, \mathbf{n}_k) \; . && (56)
\end{aligned}$$

4.5.2 Implementing the Unscented Kalman filter

The state random variable is redefined as the concatenation of the original state and the process and observation noise random variables :

$$\mathbf{x}_k^a = \begin{pmatrix} \mathbf{x}_k^x \\ \mathbf{x}_k^v \\ \mathbf{x}_k^n \end{pmatrix} = \begin{pmatrix} \mathbf{x}_k \\ \mathbf{v}_k \\ \mathbf{n}_k \end{pmatrix} \tag{57}$$

The effective dimension of this augmented state RV is now $L = L_x + L_v + L_n$. In a similar manner the augmented state covariance matrix is built up from the individual covariances matrices of $\mathbf{x}$, $\mathbf{v}$ and $\mathbf{n}$:

$$\mathbf{P}^a = \begin{pmatrix} \mathbf{P_x} & 0 & 0 \\ 0 & \mathbf{R_v} & 0 \\ 0 & 0 & \mathbf{R_n} \end{pmatrix} \tag{58}$$

4.5.3 Algorithm

Initialization

The algorithm is initialized with the initial weights for the sigma-points, and with an initial state and state covariance.

$$w_0^m = \frac{\lambda}{L+\lambda} \tag{59a}$$

$$w_0^c = \frac{\lambda}{L+\lambda} + (1-\alpha^2+\beta) \tag{59b}$$

$$w_i^m = w_i^c = \frac{1}{2(L+\lambda)} \text{ for } i = 1, \cdots, 2L, \tag{59c}$$

$$\begin{aligned}
\mathbf{x}_0 &= E[\mathbf{x}_0] \\
P_{x_0} &= E[(\mathbf{x}_0 - \widehat{\mathbf{x}}_0)(\mathbf{x}_0 - \widehat{\mathbf{x}}_0)^T] \\
\mathbf{x}_0^a &= E[\begin{pmatrix} \mathbf{x}_0 & \mathbf{0} & \mathbf{0} \end{pmatrix}]^T \\
\mathbf{P}_0^a &= E[(\mathbf{x}_0^a - \widehat{\mathbf{x}}_0^a)(\mathbf{x}_0^a - \widehat{\mathbf{x}}_0^a)^T]
\end{aligned}$$

Calculate sigma-points

$$\mathcal{X}_{k-1}^a = \begin{pmatrix} \widehat{\mathbf{x}}_{k-1}^a & \widehat{\mathbf{x}}_{k-1}^a + \gamma\sqrt{\mathbf{P}_{k-1}^a} & \widehat{\mathbf{x}}_{k-1}^a - \gamma\sqrt{\mathbf{P}_{k-1}^a} \end{pmatrix} \tag{60}$$

Prediction step

The equations for the prediction of the state value and covariance are :

$$\mathcal{X}_{k|k-1}^x = \mathbf{f}(\mathcal{X}_{k-1}^x, \mathcal{X}_{k-1}^v, \mathbf{u}_{k-1}) \tag{61}$$

$$\mathbf{x}_{k|k-1} = \sum_{i=0}^{2L} w_i^m \mathcal{X}_{i,k|k-1}^x \tag{62}$$

$$\mathbf{P}_{\mathbf{x}_{k|k-1}} = \sum_{i=0}^{2L} w_i^c (\mathcal{X}_{i,k|k-1}^x - \widehat{\mathbf{x}}_{k|k-1})(\mathcal{X}_{i,k|k-1}^x - \widehat{\mathbf{x}}_{k|k-1})^T \tag{63}$$

Innovation

By using the state prediction, the innovation and the prediction error $\mathbf{e}_k$ are :

$$\mathcal{Y}_{k|k-1} = \mathbf{h}(\mathcal{X}_{k|k-1}^x, \mathcal{X}_{k-1}^n) \tag{64}$$

$$\mathbf{y}_{k|k-1} = \sum_{i=0}^{2L} w_i^m \mathcal{Y}_{i,k|k-1} \tag{65}$$

$$\mathbf{e}_k = \mathbf{y}_k - \widehat{\mathbf{y}}_{k|k-1} \tag{66}$$

Measurement Update step

Finally, by computing the predicted covariance, we get the Kalman gain :

$$\mathbf{P}_{\widetilde{\mathbf{y}}_k} = \sum_{i=0}^{2L} w_i^c (\mathcal{Y}_{i,k|k-1} - \widehat{\mathbf{y}}_{k|k-1})(\mathcal{Y}_{i,k|k-1} - \widehat{\mathbf{y}}_{k|k-1})^T \tag{67}$$

$$\mathbf{P}_{\mathbf{x}_k\mathbf{y}_k} = \sum_{i=0}^{2L} w_i^c (\mathcal{X}_{i,k|k-1}^x - \widehat{\mathbf{x}}_{k|k-1})(\mathcal{Y}_{i,k|k-1} - \widehat{\mathbf{y}}_{k|k-1})^T \tag{68}$$

$$\mathbf{K}_k = \mathbf{P}_{\mathbf{x}_k\mathbf{y}_k} \mathbf{P}_{\widetilde{\mathbf{y}}_k}^{-1} \tag{69}$$

As before, we can now update the system state and covariance :

$$\begin{aligned} \widehat{\mathbf{x}}_{k|k} &= \widehat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k \mathbf{e}_k & (70) \\ \mathbf{P}_{\mathbf{x}_{k|k}} &= \mathbf{P}_{\mathbf{x}_{k|k-1}} - \mathbf{K}_k \mathbf{P}_{\widetilde{\mathbf{y}}_k} \mathbf{K}_k^T & (71) \end{aligned}$$

Parameters
$\mathbf{x}^a = \begin{pmatrix} \mathbf{x}^T & \mathbf{v}^T & \mathbf{n}^T \end{pmatrix}^T$, $\quad \mathcal{X}^a = \begin{pmatrix} (\mathcal{X}^x)^T & (\mathcal{X}^v)^T & (\mathcal{X}^n)^T \end{pmatrix}^T$,
$\gamma = \sqrt{L+\lambda}$, $\quad \lambda = \alpha^2(L+\kappa) - L$, $\quad 0 \le \alpha \le 1$, $\quad \beta \ge 0$, $\quad \kappa \ge 0$.

5 Non-Gaussian Bayesian Estimation

The Kalman filter (KF), Extended Kalman filter (EKF) and Unscented Kalman filter (UKF), still assume a Gaussian posterior which can fail in certain nonlinear non-Gaussian problems with multi-modal and/or heavy tailed posterior distributions. *Particle filters* (PF) are used to recursively update the posterior distribution using *Sequential Importance Sampling* (SSI) and *Resampling*. These methods approximate the posterior by a set of weighted samples without making any explicit assumptions about its form and can thus be used in general nonlinear, non-Gaussian systems.

Particle filtering is a *Monte Carlo* (MC) simulation method for recursive estimation. It makes no assumption on the noise processes nor the functional form of the system, but it requires as input specification of the prior distribution of the state, transition distribution and the likelihood. Essentially, this means particle filtering is Bayesian in nature, hence does not involve a likelihood function.

5.1 Particles Filters

We have the system :

$$\begin{aligned} \mathbf{x}_{k+1} &= \mathbf{f}_k\big(\mathbf{x}_k, \mathbf{u}_k, \mathbf{v}_k\big) & (72) \\ \mathbf{y}_k &= \mathbf{h}_k\big(\mathbf{x}_k, \mathbf{u}_k, \mathbf{n}_k\big)\ . & (73) \end{aligned}$$

The Particle filter algorithm consists of the four steps "*Initialisation*", "*Prediction*", *Updating*" and "*Resampling*".

During the *initialisation*, we sample N times from the initial distribution η_0. By saying that we sample $x^{(i)}$ from a distribution μ, for $i = 1, \cdots, N$ we mean that we simulate N independent random samples, named particles, according to μ. Hence, the N random variables $\{x^{(i)}\}$ for $i = 1, \cdots, N$ are independent and identical distributed (i.i.d.) according to η_0. Afterwards, the values of the particles are *predicted* for the next time step according to the dynamics of the state Markov process. During the "*Updating*" step, each predicted particle is weighted by the likelihood function $g_k(y_k - h_k(.))$, which is determined by the observation process. The "*Resampling*" step can be view as a special case of a "Selection" step. The particles are selected in accordance with the weighting function g_k. This step gives birth to some particles at the expense of light particles which die.

5.2 Sampling Importance Resampling

The particle filter theory presented in this section is inspired by (Doucet, de Freitas, and Gordon, 2001; van der Merwe, 2004; Del Moral, 2004).

The *Sequential Importance Sampling* (SIS) algorithm is a *Monte Carlo* (MC) method that forms the basis for most sequential Monte Carlo filters developed over the past decades. It is a technique for implementing a recursive Bayesian filter by Monte Carlo simulations. The key idea is to represent the required posterior density function by set of random samples with associated weights and to compute estimates based on these samples and weights. As the number of samples become very large, this Monte Carlo characterization becomes an equivalent representation to the usual functional description of the posterior pdf, and the SIS filter approaches the optimal Bayesian estimate.

The working mechanism of particle filters is following : The state space is partitioned as many parts, in which the particles are filled according to some probability measure. The higher probability, the denser the particles are concentrated. The particle system evolves along the time according to the state equation. Since the pdf can be approximated by the point-mass histogram, by random sampling of the state space, we get a number of particles representing the evolving pdf. However, since the posterior density model is unknown or hard to sample, we would rather choose another distribution for the sake of efficient sampling.

To avoid intractable integration in the Bayesian statistics, the posterior distribution or density is empirically represented by a weighted sum of N_s samples drawn from the posterior distribution. Let $\{\mathbf{x}_{0:k}^i, w_k^i\}$ for $i = 1, \cdots, N_s$ a Random Measure that characterises the posterior pdf $p(\mathbf{x}_{0:k}|\mathbf{y}_{1:k})$, where $\{\mathbf{x}_{0:k}^i\}$ for $i = 1, \cdots, N_s$ is a set of support points with associated weights $\{w_k^i\}$ for $i = 1, \cdots, N_s\}$ and $\mathbf{x}_{0:k} = \{\mathbf{x}_j, j = 0, \cdots, k\}$ is the set of all states up to time k. The weights are normalised such that $\sum_i w_k^i = 1$. Then, the posterior density at k can be approximated as

$$p(\mathbf{x}_{0:k}|\mathbf{y}_{1:k}) \quad \approx \quad \sum_{i=1}^{N_s} w^i \, \delta(\mathbf{x}_{0:k} - \mathbf{x}_{0:k}^i) \equiv \widehat{p}(\mathbf{x}_{0:k}|\mathbf{y}_{1:k}), \tag{74}$$

where $\{\mathbf{x}_{0:k}^i\}$ are assumed to be i.i.d. drawn from $p(\mathbf{x}_{0:k}|\mathbf{y}_{1:k})$. When N_s is sufficiently large, $\widehat{p}(\mathbf{x}_{0:k}|\mathbf{y}_{1:k})$ approximates the true posterior $p(\mathbf{x}_{0:k}|\mathbf{y}_{1:k})$. By this approximation, we can estimate the mean of a nonlinear function

$$\begin{aligned} E[\mathbf{f}(\mathbf{x}_{0:k})] &= \int \mathbf{f}(\mathbf{x}_{0:k}) \, \widehat{p}(\mathbf{x}_{0:k}|\mathbf{y}_{1:k}) \, d\mathbf{x}_{0:k} && \text{(75a)} \\ &= \frac{1}{N_s} \sum_{i=1}^{N_s} \int \mathbf{f}(\mathbf{x}_{0:k}) \, \delta(\mathbf{x}_{0:k} - \mathbf{x}_{0:k}^{(i)}) \, d\mathbf{x}_{0:k} && \text{(75b)} \\ &= \frac{1}{N_s} \sum_{i=1}^{N_s} \mathbf{f}(\mathbf{x}_{0:k}^{(i)}) \equiv \widehat{\mathbf{f}}_{N_s}(\mathbf{x}) \; . && \text{(75c)} \end{aligned}$$

Since it is usually impossible to sample from the true posterior, it is common to sample from an easy-to-implement distribution, the so called proposal distribution denoted by

$\pi(\mathbf{x}_{0:k}|\mathbf{y}_{1:k})$, hence

$$\begin{aligned} E[\mathbf{f}(\mathbf{x}_{0:k})] &= \int \mathbf{f}(\mathbf{x}_{0:k}) \frac{p(\mathbf{x}_{0:k}|\mathbf{y}_{1:k})}{\pi(\mathbf{x}_{0:k}|\mathbf{y}_{1:k})} \pi(\mathbf{x}_{0:k}|\mathbf{y}_{1:k})\, d\mathbf{x}_{0:k} && (76a) \\ &= \int \mathbf{f}(\mathbf{x}_{0:k}) \frac{p(\mathbf{y}_{1:k}|\mathbf{x}_{0:k})\, p(\mathbf{x}_{0:k})}{p(\mathbf{y}_{1:k})\, \pi(\mathbf{x}_{0:k}|\mathbf{y}_{1:k}} \pi(\mathbf{x}_{0:k}|\mathbf{y}_{1:k})\, d\mathbf{x}_{0:k} && (76b) \\ &= \int \mathbf{f}(\mathbf{x}_{0:k}) \frac{w_k(\mathbf{x}_{0:k})}{p(\mathbf{y}_{1:k})} \pi(\mathbf{x}_{0:k}|\mathbf{y}_{1:k})\, d\mathbf{x}_{0:k} \ , && (76c) \end{aligned}$$

where the variables $w_k(\mathbf{x}_{0:k})$ are known as the unnormalized importance weights, and are given by

$$w_k(\mathbf{x}_{0:k}) = \frac{p(\mathbf{y}_{1:k}|\mathbf{x}_{0:k})\, p(\mathbf{x}_{0:k})}{\pi(\mathbf{x}_{0:k}|\mathbf{y}_{1:k})} \ . \qquad (77)$$

We can get rid of the generally unknown or hard to calculate normalizing density $p(\mathbf{y}_{1:k})$ in Equation 76 as follow:

$$\begin{aligned} E[\mathbf{f}(\mathbf{x}_{0:k})] &= \frac{1}{p(\mathbf{y}_{1:k})} \int \mathbf{f}(\mathbf{x}_{0:k})\, w_k(\mathbf{x}_{0:k})\, \pi(\mathbf{x}_{0:k}|\mathbf{y}_{1:k})\, d\mathbf{x}_{0:k} && (78a) \\ &= \frac{\int \mathbf{f}(\mathbf{x}_{0:k})\, w_k(\mathbf{x}_{0:k})\, \pi(\mathbf{x}_{0:k}|\mathbf{y}_{1:k})\, d\mathbf{x}_{0:k}}{\int w_k(\mathbf{x}_{0:k})\, \pi(\mathbf{x}_{0:k}|\mathbf{y}_{1:k})\, d\mathbf{x}_{0:k}} && (78b) \\ &= \frac{E_\pi \left[\mathbf{f}(\mathbf{x}_{0:k})\, w_k(\mathbf{x}_{0:k})\right]}{E_\pi[w_k(\mathbf{x}_{0:k})]} \ , && (78c) \end{aligned}$$

where the notation $E_\pi[.]$ emphases that the expectations are taken over the proposal distribution $\pi(\mathbf{x}_{0:k}|\mathbf{y}_{1:k})$. By drawing the i.i.d. samples $\{\mathbf{x}_{0:k}^{(i)}\}$ from the proposal distribution $\pi(\mathbf{x}_{0:k}|\mathbf{y}_{1:k})$, we can approximate the expectations of interest by the following estimate:

$$\begin{aligned} E[\mathbf{f}(\mathbf{x}_{0:k})] \approx \widetilde{E}[\mathbf{f}(\mathbf{x}_{0:k})] &= \frac{\frac{1}{N_s}\sum_{i=1}^{N_s} \mathbf{f}\left(\mathbf{x}_{0:k}^{(i)}\right)\, w_k\left(\mathbf{x}_{0:k}^{(i)}\right)}{\frac{1}{N_s}\sum_{i=1}^{N_s} w_k\left(\mathbf{x}_{0:k}^{(i)}\right)} && (79a) \\ &= \sum_{i=1}^{N_s} \widetilde{w}_k^{(i)}\, \mathbf{f}\left(\mathbf{x}_{0:k}^{(i)}\right) \ , && (79b) \end{aligned}$$

where the normalized importance weights $\widetilde{w}_k^{(i)}$ are given by:

$$\widetilde{w}_k^{(i)} = \frac{w_k\left(\mathbf{x}_{0:k}^{(i)}\right)}{\sum_{i=1}^{N_s} w_k\left(\mathbf{x}_{0:k}^{(i)}\right)} \ . \qquad (80)$$

Suppose the proposal distribution has the following form :

$$\begin{aligned} \pi(\mathbf{x}_{0:k}|\mathbf{y}_{1:k}) &= \pi(\mathbf{x}_{0:k-1}|\mathbf{y}_{1:k-1})\, \pi(\mathbf{x}_k|\mathbf{x}_{0:k-1}, \mathbf{y}_{1:k}) && (81a) \\ &= \pi(\mathbf{x}_0) \prod_{j=1}^{k} \pi(\mathbf{x}_j|\mathbf{x}_{0:j-1}, \mathbf{y}_{1:j}) \ . && (81b) \end{aligned}$$

With this representation of the proposal distribution, we can realize an estimate of the posterior distribution at time k without modifying the previously simulated $\mathbf{x}_{0:k-1}^{(i)}$, then one can obtain samples $\mathbf{x}_{0:k}^{(i)} \sim \pi(\mathbf{x}_{0:k}|\mathbf{y}_{1:k})$ by augmenting each of the existing samples

$\mathbf{x}_{0:k-1}^{(i)} \sim \pi(\mathbf{x}_{0:k-1}|\mathbf{y}_{1:k-1})$ with the new state $\mathbf{x}_k^{(i)} \sim \pi(\mathbf{x}_k|\mathbf{x}_{0:k-1}, \mathbf{y}_{1:k})$. To derive the weight update equation, $\pi(\mathbf{x}_{0:k}|\mathbf{y}_{1:k})$ is first expressed in terms of $\pi(\mathbf{x}_{0:k-1}|\mathbf{y}_{1:k-1})$, $\pi(\mathbf{x}_k|\mathbf{y}_k)$, and $\pi(\mathbf{x}_k|\mathbf{x}_{k-1})$.

Under the assumptions that the states correspond to a first order Markov process and that the observations are conditionally independent given the states, we get :

$$\begin{aligned} p(\mathbf{x}_{0:k}) &= p(\mathbf{x}_0) \prod_{j=1}^{k} p(\mathbf{x}_j|\mathbf{x}_{j-1}) && (82) \\ p(\mathbf{y}_{1:k}|\mathbf{x}_{0:k}) &= \prod_{j=1}^{k} p(\mathbf{y}_j|\mathbf{x}_j) \ , && (83) \end{aligned}$$

and the posterior distribution can be factorized as :

$$\begin{aligned} p(\mathbf{x}_{0:k}|\mathbf{y}_{1:k}) &= \frac{p(\mathbf{y}_k|\mathbf{x}_{0:k}, \mathbf{y}_{1:k-1}) \ p(\mathbf{x}_{0:k}|\mathbf{y}_{1:k-1})}{p(\mathbf{y}_k|\mathbf{y}_{1:k-1})} && \text{(84a)} \\ &= \frac{p(\mathbf{y}_k|\mathbf{x}_{0:k}, \mathbf{y}_{1:k-1}) \ p(\mathbf{x}_k|\mathbf{x}_{0:k-1}, \mathbf{y}_{1:k-1})}{p(\mathbf{y}_k|\mathbf{y}_{1:k-1})} \ p(\mathbf{x}_{0:k-1}|\mathbf{y}_{1:k-1}) && \text{(84b)} \\ &= \frac{p(\mathbf{y}_k|\mathbf{x}_k) \ p(\mathbf{x}_k|\mathbf{x}_{k-1})}{p(\mathbf{y}_k|\mathbf{y}_{1:k-1})} \ p(\mathbf{x}_{0:k-1}|\mathbf{y}_{1:k-1}) && \text{(84c)} \\ &\propto \ p(\mathbf{y}_k|\mathbf{x}_k) \ p(\mathbf{x}_k|\mathbf{x}_{k-1}) \ p(\mathbf{x}_{0:k-1}|\mathbf{y}_{1:k-1}) \ . && \text{(84d)} \end{aligned}$$

Thus a recursive estimate for the importance weights can be factorized as follow :

$$\begin{aligned} w_k^{(i)} &= \frac{p(\mathbf{x}_{0:k}^{(i)}|\mathbf{y}_{1:k})}{\pi(\mathbf{x}_{0:k}^{(i)}|\mathbf{y}_{1:k})} && \text{(85a)} \\ &= \frac{p(\mathbf{y}_{1:k}|\mathbf{x}_{0:k}^{(i)}) \ p(\mathbf{x}_{0:k}^{(i)})}{\pi(\mathbf{x}_{0:k-1}^{(i)}|\mathbf{y}_{1:k-1}) \ \pi(\mathbf{x}_k^{(i)}|\mathbf{x}_{0:k-1}^{(i)}, \mathbf{y}_{1:k})} && \text{(85b)} \\ &= w_{k-1}^{(i)} \ \frac{p(\mathbf{y}_{1:k}|\mathbf{x}_{0:k}^{(i)}) \ p(\mathbf{x}_{0:k}^{(i)})}{p(\mathbf{y}_{1:k-1}|\mathbf{x}_{0:k-1}^{(i)}) \ p(\mathbf{x}_{0:k-1}^{(i)}) \ \pi(\mathbf{x}_k^{(i)}|\mathbf{x}_{0:k-1}^{(i)}, \mathbf{y}_{1:k})} && \text{(85c)} \\ &= w_{k-1}^{(i)} \ \frac{\prod_{j=1}^{k} \ p(\mathbf{y}_j|\mathbf{x}_j^{(i)}) \ p(\mathbf{x}_0^{(i)}) \ \prod_{j=1}^{k} \ p(\mathbf{x}_j^{(i)}|\mathbf{x}_{j-1}^{(i)})}{\prod_{j=1}^{k-1} \ p(\mathbf{y}_j|\mathbf{x}_j^{(i)}) \ p(\mathbf{x}_0^{(i)}) \ \prod_{j=1}^{k-1} \ p(\mathbf{x}_j^{(i)}|\mathbf{x}_{j-1}^{(i)}) \ \pi(\mathbf{x}_k^{(i)} \mathbf{x}_{0:k-1}^{(i)}, \mathbf{y}_{1:k})} && \text{(85d)} \\ &= w_{k-1}^{(i)} \ \frac{p(\mathbf{y}_k|\mathbf{x}_k^{(i)}) \ p(\mathbf{x}_k^{(i)}|\mathbf{x}_{k-1}^{(i)})}{\pi(\mathbf{x}_k^{(i)}|\mathbf{x}_{0:k-1}^{(i)}, \mathbf{y}_{1:k})} \ . && \text{(85e)} \end{aligned}$$

Furthermore, if $\pi(\mathbf{x}_k|\mathbf{x}_{0:k-1}, \mathbf{y}_{1:k}) = \pi(\mathbf{x}_k|\mathbf{x}_{0:k-1}, \mathbf{y}_k)$, then the importance density becomes only dependent on $\mathbf{x}_{k-1}$ and $\mathbf{y}_k$. This is particularly useful in the common case when only a filtered estimate of $p(\mathbf{x}_k|\mathbf{y}_{1:k})$ is required at each time step. In such case, only $\mathbf{x}_k^{(i)}$ need to be stored, and so one can discard the path, $\{\mathbf{x}_{0:k-1}^{(i)}\}$, and the history of the observations, $\{\mathbf{y}_{1:k-1}\}$. The modified weight is then :

$$w_k^{(i)} \ \propto \ w_{k-1}^{(i)} \ \frac{p(\mathbf{y}_k|\mathbf{x}_k^{(i)}) \ p(\mathbf{x}_k^{(i)}|\mathbf{x}_{k-1}^{(i)})}{\pi(\mathbf{x}_k^{(i)}|\mathbf{x}_{0:k-1}^{(i)}, \mathbf{y}_k)} \ , \tag{86}$$

and the posterior filtered density $p(\mathbf{x}_k|\mathbf{y}_{1:k})$ can be approximated as :

$$p(\mathbf{x}_k|\mathbf{y}_{1:k}) \ \approx \ \sum_{i=1}^{N_s} w_k^{(i)} \ \delta(\mathbf{x}_k - \mathbf{x}_k^{(i)}) \ , \tag{87}$$

where the weights are defined in Equation 86. It can be shown that as $N_s \to \infty$ the approximation Equation 87 approach the true posterior density $p(\mathbf{x}_k|\mathbf{y}_{1:k})$ (Doucet, 1997). These point-mass estimates can approximate any general distribution arbitrarily well, limited only by the number of particles used and how well the importance sampling conditions are met. In contrast, the posterior distribution calculated by the EKF is a minimum-variance Gaussian approximation of the true posterior distribution, which cannot capture complex structure such as multimodalities, skewness, or other higher-order moments.

5.3 Degeneracy Problem

A common problem with the SIS particle filters is the degeneracy phenomenon, where after a few iterations, all but one particle will have negligible weight. It has been shown (Doucet, 1998) that the variance of the importance weights can only increase over time, and thus it is impossible to avoid the degeneracy phenomenon. This degeneracy implies that a large computational effort is devoted to updating particles whose contribution to the approximation to $p(\mathbf{x}_k|\mathbf{y}_{1:k})$ is almost zero. We could use a very large number of particles N_s to reduce these effects, but it is often impractical, and so we rely on two other methods : a good choice of *Importance Density* and the use of *Resampling.*

5.4 Choice of Proposal Distribution

It has been shown (Doucet, 1997) that the optimal proposal distribution

$$\pi(\mathbf{x}_k^{(i)}|\mathbf{x}_{0:k-1}^{(i)}, \mathbf{y}_{1:k}) \quad \doteq \quad p(\mathbf{x}_k^{(i)}|\mathbf{x}_{k-1}^{(i)}, \mathbf{y}_k) \ , \tag{88}$$

minimizes the variance of the proposal weights conditional on $\mathbf{x}_{0:k-1}^{(i)}$ and $\mathbf{y}_{1:k}$. Nonetheless, the distribution

$$\pi(\mathbf{x}_k^{(i)}|\mathbf{x}_{0:k-1}^{(i)}, \mathbf{y}_{1:k}) \quad \doteq \quad p(\mathbf{x}_k^{(i)}|\mathbf{x}_{k-1}^{(i)}) \ , \tag{89}$$

(the transition prior) is the most popular choice of proposal distribution. This proposal distribution is usually easier to implement, but it is not incorporating the most recent observations. Substitution of Equations 81, 82, and 83 into Equation 86 yields

$$\begin{aligned} w_k^{(i)} &= w_{k-1}^{(i)} \frac{p(\mathbf{y}_k|\mathbf{x}_k^{(i)})\, p(\mathbf{x}_k^{(i)}|\mathbf{x}_{k-1}^{(i)})}{\pi(\mathbf{x}_k^{(i)}|\mathbf{x}_{0:k-1}^{(i)}, \mathbf{y}_k)} && \text{(90a)} \\ &= w_{k-1}^{(i)} \frac{p(\mathbf{y}_k|\mathbf{x}_k^{(i)})\, p(\mathbf{x}_k^{(i)}|\mathbf{x}_{k-1}^{(i)})}{p(\mathbf{x}_k^{(i)}|\mathbf{x}_{k-1}^{(i)})} && \text{(90b)} \\ &= w_{k-1}^{(i)}\, p(\mathbf{y}_k|\mathbf{x}_k^{(i)})\, p(\mathbf{x}_k^{(i)}|\mathbf{x}_{k-1}^{(i)}) \ . && \text{(90c)} \end{aligned}$$

Thus, if we chose the transition prior as our proposal distribution to sample from, the importance weights are easily updated by simply evaluating the observation likelihood density $p(\mathbf{y}_k|\mathbf{x}_k^{(i)})$ for the sampled particle set and multiply with the previous weights.

5.5 Resampling

The basic idea of resampling is to eliminate particles which have small weights and to concentrate on particles with very large weights. The resampling step involves generating a new set $\{\mathbf{x}_k^{(i)*}\}$ for $i = 1, \cdots, N_s$ by resampling with replacement N_s times from an approximate discrete representation of $p(\mathbf{x}_k|\mathbf{y}_{1:k})$ given by

$$p(\mathbf{x}_k|\mathbf{y}_{1:k}) \approx \sum_{i=1}^{N_s} w_k^i \, \delta(\mathbf{x}_k - \mathbf{x}_k^i) , \tag{91}$$

so that $Pr(\mathbf{x}_k^{(i)*} = \mathbf{x}_k^j) = w_k^j$. The resulting sample is in fact an i.i.d. sample from the discrete density Equation 91, and so the weights are now reset to $w_k^i = 1/N_s$. After the selection/resampling step at time k, we obtain N_s particles distributed marginally approximately according to the posterior distribution. Since the selection step favors the creation of multiple copies of the "fittest" particles, many particles may end up without children, whereas other might end up having a large number of children. Therefore, an additional procedure is required to introduce sample variety after the selection step without affecting the validity of the approximation they infer. This is achieved by performing a single *Markov Chain Monte Carlo* (MCMC) step on each particle.

5.6 Better Proposal Distributions

The success of the particle filter algorithm depends on the validity of the assumptions :

Monte Carlo (MC) assumption : The Dirac point-mass approximation provides an adequate representation of the posterior distribution.

Importance sampling assumption : It is possible to obtain samples from the posterior by sampling from a suitable proposal distribution and applying importance sampling corrections.

If any of these conditions are not met, the PF algorithm can perform poorly. In the resampling stage, any particular sample with a high importance weight will be duplicated many times, and the cloud of samples may collapse to a single sample. Thus, the number of samples used to describe the posterior density function will become too small and inadequate. We can get around this difficulty by implementing a Markov Chain Monte Carlo step after the selection step. But, this method is only successful if the point-mass posterior approximation is already a close approximation of the true posterior. One of the main causes of sample depletion is the failure to move particles to areas of high observation likelihood. This failure stems directly from the most common choice of importance distribution, the transition prior which do not incorporate the latest observation. To improve the performance of particle filters, we could design better proposal distribution that not only allow for easy sampling and evaluation of the importance weights, but also address the sample depletion problem. This can be done by choosing a proposal distribution that is conditioned on $\mathbf{y}_k$.

We accomplishes this by approximating this density by a tractable single Gaussian distribution as generated by a Gaussian approximate recursive Bayesian estimation framework such as the Kalman filter:

$$\begin{aligned} \pi(\mathbf{x}_k|\mathbf{x}_{0:k-1}, \mathbf{y}_{1:k}) &\doteq p(\mathbf{x}_k|\mathbf{x}_{k-1}, \mathbf{y}_k) && \text{(92a)} \\ &= q_{\mathcal{N}}(\mathbf{x}_k|\mathbf{y}_{1:k}) , && \text{(92b)} \end{aligned}$$

where $q_{\mathcal{N}}(.)$ denotes a Gaussian proposal distribution.

A tractable way of generating Gaussian approximate proposal distribution within the particle filter framework, is to use an adaptive bank of parallel running Unscented Kalman Filters (UKF) to generate and propagate a Gaussian proposal distribution for each particle,

$$q_{\mathcal{N}}(\mathbf{x}_k|\mathbf{y}_{1:k}) \;=\; \mathcal{N}\big(\, x_k;\; x_k^{(i)}\;,\; P_{x_k}^{(i)}\,\big) \quad i = 1, \cdots, N_s \tag{93}$$

each contributing its own estimate as a component in a very large adaptive mixture approximation of the posterior distribution.

5.7 The Sigma-Point Particle Filter (SPPF)

The sigma-point particle filter algorithm was published first in (van der Merwe, de Freitas, Doucet and Wan , 2001).

Initialization : $k = 0$

For $i = 1, \cdots, N$, draw particle $\mathbf{x}_0^{(i)}$ from the prior $p(\mathbf{x}_0)$.

For $\mathbf{k = 1, 2, \cdots}$

1. Importance sampling step

For $i = 1, \cdots, N$:

(a.) Update the Gaussian prior distribution for each particle with the SPKF :

Calculate sigma-points for particle

$$\boldsymbol{x}_{k-1}^{a,(i)} \;=\; \Big(\; \boldsymbol{x}_{k-1}^{(i)} \quad \overline{\mathbf{v}}_{k-1} \quad \overline{\mathbf{n}}_{k-1} \;\Big)^T$$

$$\boldsymbol{\mathcal{X}}_{k-1,(0,\cdots,2L)}^{a,(i)} \;=\; \Big(\; \boldsymbol{x}_{k-1}^{a,(i)} \quad \boldsymbol{x}_{k-1}^{a,(i)} + \gamma\sqrt{\mathbf{P}_{k-1}^{a,(i)}} \quad \boldsymbol{x}_{k-1}^{a,(i)} - \gamma\sqrt{\mathbf{P}_{k-1}^{a,(i)}} \;\Big)$$

Prediction step
Propagate sigma-points into future (time update) :

$$\begin{aligned}
\boldsymbol{\mathcal{X}}_{k|k-1,(0,\cdots,2L)}^{x,(i)} &= \mathbf{f}\big(\boldsymbol{\mathcal{X}}_{k-1,(0,\cdots,2L)}^{x,(i)},\; \boldsymbol{\mathcal{X}}_{k-1,(0,\cdots,2L)}^{v,(i)},\; \mathbf{u}_k\big) \\
\overline{\boldsymbol{x}}_{k|k-1}^{(i)} &= \sum_{j=0}^{2L} w_j^{(m)}\, \mathcal{X}_{k|k-1,j}^{x,(i)} \\
\mathbf{P}_{k|k-1}^{(i)} &= \sum_{j=0}^{2L} w_j^{(c)}\, \big(\mathcal{X}_{k|k-1,j}^{x,(i)} - \overline{\boldsymbol{x}}_{k|k-1}^{(i)}\big)\big(\mathcal{X}_{k|k-1,j}^{x,(i)} - \overline{\boldsymbol{x}}_{k|k-1}^{(i)}\big)^T
\end{aligned}$$

Innovation

By using the state prediction, the innovation and the prediction error $\mathbf{e}_k$ are :

$$\begin{aligned}
\boldsymbol{\mathcal{Y}}^{(i)}_{k|k-1,(0,\cdots,2L)} &= \mathbf{h}\big(\boldsymbol{\mathcal{X}}^{x,(i)}_{k|k-1,(0,\cdots,2L)}\ ,\ \boldsymbol{\mathcal{X}}^{n,(i)}_{k-1,(0,\cdots,2L)}\big) \\
\overline{\boldsymbol{y}}^{(i)}_{k|k-1} &= \sum_{j=0}^{2L} w_j^{(m)}\ \mathcal{Y}^{(i)}_{k|k-1,j} \\
\boldsymbol{e}^{(i)}_k &= \boldsymbol{y}_k - \overline{\boldsymbol{y}}^{(i)}_{k|k-1}
\end{aligned}$$

Measurement Update step

Incorporate new observation (measurement update)

$$\begin{aligned}
\mathbf{P}_{\boldsymbol{y}_k \boldsymbol{y}_k} &= \sum_{i=0}^{2L} w_j^{(c)} \big(\mathcal{Y}^{(i)}_{k|k-1,j} - \overline{\boldsymbol{y}}^{(i)}_{k|k-1}\big)\big(\mathcal{Y}^{(i)}_{k|k-1,j} - \overline{\boldsymbol{y}}^{(i)}_{k|k-1}\big)^T \\
\mathbf{P}_{\boldsymbol{x}_k \boldsymbol{y}_k} &= \sum_{i=0}^{2L} w_i^{(c)} \big(\mathcal{X}^{(i)}_{k|k-1,j} - \overline{\boldsymbol{x}}^{(i)}_{k|k-1}\big)\big(\mathcal{Y}^{(i)}_{k|k-1,j} - \overline{\boldsymbol{y}}^{(i)}_{k|k-1}\big)^T \\
\mathbf{K}_k &= \mathbf{P}_{\boldsymbol{x}_k \boldsymbol{y}_k} \mathbf{P}^{-1}_{\boldsymbol{y}_k \boldsymbol{y}_k}
\end{aligned}$$

As before, we can now update the system state and covariance :

$$\begin{aligned}
\overline{\boldsymbol{x}}^{(i)}_{k|k} &= \overline{\boldsymbol{x}}^{(i)}_{k|k-1} + \mathbf{K}_k \boldsymbol{e}^{(i)}_k \\
\mathbf{P}^{(i)}_{k|k} &= \mathbf{P}^{(i)}_{k|k-1} - \mathbf{K}_k \mathbf{P}_{\boldsymbol{y}_k \boldsymbol{y}_k} \mathbf{K}^T_k
\end{aligned}$$

(b.)Sample

$$\boldsymbol{x}^{(i)}_{k|k} \ \sim\ q_{\mathcal{N}}\big(\mathbf{x}_{k|k}, \boldsymbol{y}_{1:k}\big) = \mathcal{N}\big(\mathbf{x}_{k|k}\ ;\ \overline{\boldsymbol{x}}^{(i)}_{k|k}\ ,\ \mathbf{P}^{(i)}_{k|k}\big)$$

For $i = 1, \cdots, N$, evaluate the importance weights up to a normalizing constant :

$$w^{(i)}_k = w^{(i)}_{k-1} \frac{p\big(\boldsymbol{y}_k | \boldsymbol{x}^{(i)}_{k|k}\big)\ p\big(\boldsymbol{x}^{(i)}_{k|k} | \boldsymbol{x}^{(i)}_{k-1|k-1}\big)}{q_{\mathcal{N}}\big(\boldsymbol{x}^{(i)}_{k|k} | \boldsymbol{y}_{1:k}\big)}$$

For $i = 1, \cdots, N$ normalize the importance weights :

$$\widetilde{w}^{(i)}_k = \frac{w^{(i)}_k}{\sum_{j=1}^N w^{(j)}_k}$$

2. Resampling

Multiply/suppress samples $\boldsymbol{x}^{(i)}_k$ with high/low importance weights $\widetilde{w}^{(i)}_k$, to obtain N random samples approximately distributed according to $p(\mathbf{x}_k | \boldsymbol{y}_{1:k})$

For $i = 1, \cdots, N$, set $w_k^{(i)} = \widetilde{w}_k^{(i)} = N^{-1}$

3. Markov Chain Monte Carlo

Do a single MCMC move step to add further variety to the particle set without changing their distribution. Apply a Markov transition kernel with invariant distribution $p(\mathbf{x}_{0:k}^{(i)} | \boldsymbol{y}_{1:k})$ to obtain $\left(\mathbf{x}_{0:k}^{(i)}, \mathbf{P}_{0:k}^{(i)}\right)$.

4. Output

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

$$p(\mathbf{x}_k | \boldsymbol{y}_{1:k}) \quad = \quad \frac{1}{N} \sum_{i=1}^{N} \delta(\mathbf{x}_k - \boldsymbol{x}_k^{(i)})$$

from these samples, any estimated of the system state can be estimated, such as the MMSE estimate,

$$\widehat{\mathbf{x}}_k \quad = \quad E[\mathbf{x}_k | \boldsymbol{y}_{1:k}] \approx \frac{1}{N} \sum_{i=1}^{N} \boldsymbol{x}_k^{(i)}$$

6 State, Parameters and Dual Estimations

Kalman filters and Particle filters can be used to realize some estimations : *state estimation*, *parameter estimation* (machine learning), and *dual estimation* (expectation-maximization (EM) algorithm).

6.1 State estimation

The basis framework involves estimation of the state of a non-linear discrete state space model,

$$\mathbf{x}_{k+1} \quad = \quad \mathbf{f}_k(\mathbf{x}_k, \mathbf{u}_k, \mathbf{v}_k; \mathbf{w}) \tag{94}$$

$$\mathbf{y}_k \quad = \quad \mathbf{h}_k(\mathbf{x}_k, \mathbf{n}_k; \mathbf{w}) \ , \tag{95}$$

where $\mathbf{x}_k$ represents the unobserved state of the system, $\mathbf{n}_k$ is a known exogenous input, and $\mathbf{y}_k$ is the observed measurement signal. The *process noise* $\mathbf{v}_k$ drives the system, and the *observation noise* is given by $\mathbf{n}_k$. The system dynamic model $\mathbf{f}$ and $\mathbf{h}$ are assumed known and parameterized by a set of known parameters $\mathbf{w}$. In state estimation, the filter is an approximate method of choice to achieve a recursive maximum-likelihood estimation of the state $\mathbf{x}_k$.

6.2 Parameter estimation

Parameter estimation refers to a system identification or machine learning, which involves determining a nonlinear mapping

$$\mathbf{y}_k \quad = \quad \mathbf{g}(\mathbf{x}_k; \mathbf{w}) \ , \tag{96}$$

where $\mathbf{x}_k$ is the input, $\mathbf{y}_k$ the output, and the nonlinear map $\mathbf{g}(.)$ is parameterized by the vector of hyper-parameters $\mathbf{w}$ which are to be estimated in some optimal way. A training set is provided with sample pairs $\{\mathbf{x}_k, \mathbf{d}_k\}$. The error is defined as :

$$\mathbf{e}_k = \mathbf{d}_k - \mathbf{g}(\mathbf{x}_k; \mathbf{w}) \; , \tag{97}$$

and we have to estimate $\mathbf{w}$ in order to minimize the expectation of some error function. A filter may be used to estimate the parameters with a new state-space model :

$$\mathbf{w}_{k+1} = \mathbf{w}_k + \mathbf{r}_k \tag{98}$$

$$\mathbf{d}_k = \mathbf{g}(\mathbf{x}_k, \mathbf{w}_k) + \mathbf{e}_k \; , \tag{99}$$

where $\mathbf{w}_k$ corresponds to a stationary process, and $\mathbf{r}_k$ an "artificial" process noise. The output $\mathbf{d}_k$ corresponds to a nonlinear observation on $\mathbf{w}_k$.

6.3 Dual estimation

A special case arises when the input $\mathbf{x}_k$ is unobserved, and requires coupling both state estimation and parameter estimation :

$$\mathbf{x}_{k+1} = \mathbf{f}_k(\mathbf{x}_k, \mathbf{u}_k, \mathbf{v}_k; \mathbf{w}_k) \tag{100}$$

$$\mathbf{y}_k = \mathbf{h}_k(\mathbf{x}_k, \mathbf{n}_k; \mathbf{w}_k) \; , \tag{101}$$

where both, the system state $\mathbf{x}_k$ and the set of model parameters $\mathbf{w}_k$ for the dynamic system must be simultaneously estimated only from the observed noisy signal $\mathbf{y}_k$.

In the *dual filtering approach*, two separate state-space representations are used for the state and the parameters. Both filters are run simultaneously, in an iterative fashion, for state estimation and parameter estimation. At every time step, the current estimate of the parameters $\widehat{\mathbf{w}}_k$ is used in the state filter as fixed, and likewise the current estimate of the state $\widehat{\mathbf{x}}_k$ is used in the parameter filter.

In the *joint filtering approach*, the unknown system state and the parameters are concatenated into a single higher-dimensional joint state vector, $\widetilde{\mathbf{x}}_k$:

$$\widetilde{\mathbf{x}}_k = \begin{pmatrix} \mathbf{x}_k^T & \mathbf{w}_k^T \end{pmatrix}^T \; , \tag{102}$$

and the state space model is

$$\widetilde{\mathbf{x}}_{k+1} = \widetilde{\mathbf{f}}_k(\widetilde{\mathbf{x}}_k, \mathbf{u}_k, \widetilde{\mathbf{v}}_k) \tag{103}$$

$$\mathbf{y}_k = \widetilde{\mathbf{h}}_k(\widetilde{\mathbf{x}}_k, \mathbf{n}_k) \; , \tag{104}$$

which can be expanded to

$$\begin{pmatrix} \mathbf{x}_{k+1} \\ \mathbf{w}_{k+1} \end{pmatrix} = \begin{pmatrix} \mathbf{f}_k(\mathbf{x}_k, \mathbf{u}_k, \mathbf{v}_k; \mathbf{w}_k) \\ \mathbf{w}_k \end{pmatrix} + \begin{pmatrix} \mathbf{0} \\ \mathbf{r}_k \end{pmatrix} \tag{105}$$

$$\mathbf{y}_k = \mathbf{h}_k(\mathbf{x}_k, \mathbf{n}_k; \mathbf{w}_k) \; , \tag{106}$$

where $\widetilde{\mathbf{v}}_k = \begin{pmatrix} \mathbf{v}_k^T & \mathbf{r}_k^T \end{pmatrix}^T$. A single filter is now run on the joint state space to produce simultaneous estimates of the states $\mathbf{x}_k$ and the parameters $\mathbf{w}_k$.

7 Experimental Results

7.1 Data

We conduct our analysis using very high-frequency time series on IBM stock, for the period : January, 03, 1995 to May, 26, 1999. We use the period : January, 04, 1995 to December, 31, 1998 for training en validation of the models, and the period : January, 04, 1999 to May, 26, 1999 for testing.

7.2 Methodology

We build two *Dynamical State-Space Models* (DSSM), with nonlinear *Radial Basis Function* (RBF) (Benoudjit and Verleysen, 2003) $\mathbf{f}_k$ in the state and $\mathbf{h}_k$ in the measurement equations.

$$\begin{aligned} \mathbf{x}_{k+1} &= \mathbf{f}_k(\mathbf{x}_k, \mathbf{u}_k \mathbf{v}_k; \mathbf{w}) && (107) \\ \mathbf{y}_k &= \mathbf{h}_k(\mathbf{x}_k, \mathbf{n}_k; \mathbf{w}) \ , && (108) \end{aligned}$$

where $\mathbf{x}_k$ represents the hidden random state space variable, $\mathbf{u}_k$ a deterministic input, $\mathbf{y}_k$ the measurement, $\mathbf{v}_k$ the process noise, $\mathbf{n}_k$ the measurement noise, and $\mathbf{w}$ the parameters of the system.

The first model (Pricing model) gives the forecasting of the trend prices for the next hours. Based on this forecasting, the trader can decide to buy or to sell, or nothing. The second model (Trading model) estimates the first stopping time to close the position, depending of a threshold defined by the trader.

The pricing and trading models use very high-frequency time series of tick by tick data, without resampling, but as these observations are sparse, irregularly spaced, and occur at different time points for each subject, we smooth the rough data by projecting them onto a *functional basis*, and we use the coefficients of this projection for modelling purposes (Dablemont et al., 2007).

We realize the training phase, to estimate the parameters of the models, by *dual estimation*, but in order to reduce the CPU time, afterwards we realize the validation phase by *state estimation* with parameters fixed. In the simulation phase, we use a *joint estimation* for the parameters and the states, in order to take into account changes of the system dynamic. In a real-time application, it should be possible to re-estimate, every hour, the trend and the stopping time, so as to optimize the profits or to stop-lose immediately.

7.3 Simulation

Every day, at 11.00 AM, we realize the trading. We decide to buy or to sell **ONE** share or nothing, depending of the trend forecasting, then we close the position according to the predicted stopping time. We have chosen a threshold of 1 US $.

The resulting behavior of the trading process is presented as a function of time in Figure 1. The return figure includes the losses due to transaction costs (the bid-ask spreads).

We assume an investor with credit limit but no capital.

We also could make the simulation at more than one time-point of the day. With many buyings or sellings of the asset we could optimize the opportunities of profit but due to the CPU time, this was not possible on a standard PC.

The algorithms are written in Matlab, but in order to use this model in a real-time application, we should increase the performance and curb the CPU time, using a C++ executing file on a multi-processors unit.

At first sight, the stable profitability of trading models and the quality of the pricing models could be a good opportunity for trading actions on a very short horizon.

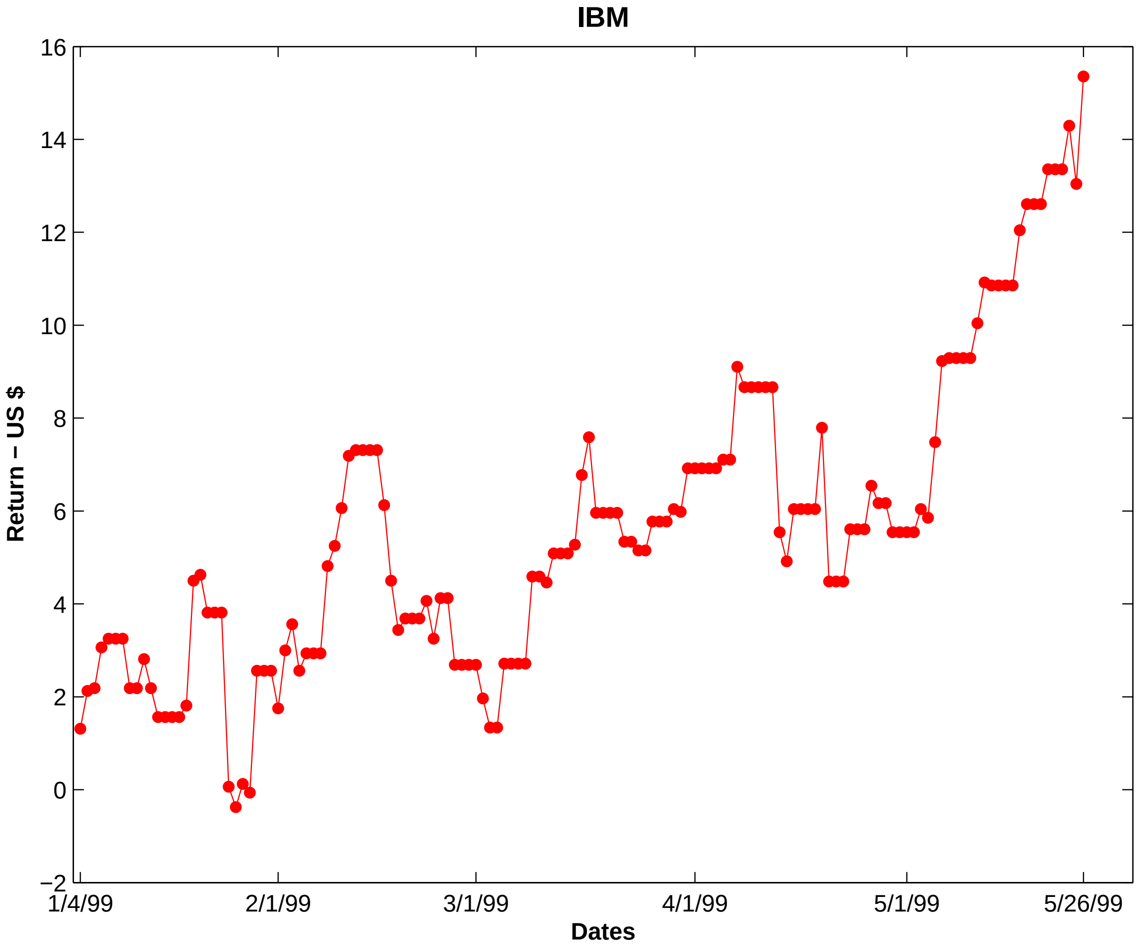

Figure 1: Returns of trading models as function of time.

8 Conclusion

We have presented a functional method for the trading of very hight-frequency time series by functional analysis, neural networks, and Bayesian estimation, with Particle and Kalman filters.

This trading model gives, at time t, a signal of buying or selling an asset, and depending on the trader's decision, when the price of this asset crosses for the first time a threshold defined by the trader, in order to close his position with a yielding profit, and also it can send a stop-loss order.

It can be applied when the observations are sparse, irregularly spaced and arrive at different time points for each day, as with tick by tick asset prices.

We have realized a simulation of trading on five months to test the algorithms.

These algorithms are written in Matlab, but they last a very long CPU times for the forecasting, requires for the recursive up-dating of the Particles. In case of a real-time application it should be useful to put these algorithms in a form suitable for parallelization in C++.

References

Ait-Sahalia, Y., and P.A. Myland,(March, 2003),"The effects of random and discrete sampling when estimating continuous-time diffusions",*Econometrica*, Vol. 71, pp. 483-549.

Ait-Sahalia, Y.,(january, 2002),"Maximum likelihood estimation of Discretely sampled Diffusions: a Closed-orm approximation Approach",*Econometrica*, Vol. 70, pp. 223-262.

Alspach, D.L., and H.W. Sorenson,"Nonlinear Bayesian estimation usin Gaussian Sum approximation.",*IEEE transaction on Automatic Control*, Vol. 17, 4 (1972) pp. 439-448.

Andersen, T.G., T. Bollerslev, and X. diebold, "Parametric and Nonparametric Volatility Measurement", in *Hanbook of Financial Econometrics*, Ed; Yacine Aït-Sahalia and Lars Petre Hansen. Elsevier Science July 2005.

Anderson, B.D., and J.B. More, "*Optimal Filtering*", Prentice-Hall, 1979.

Benoudjit, N. and M. Verleysen, "On the kernel widths in Radial-Basis Function Networks", Neural Processing Letters, Kluwer academic pub., vol. 18, no. 2, pp. 139-154, October 2003.

Billio, M., R. Casarin, and D. sartore ,(2004), "Bayesian Inference on Dynamic Modelss with Latent Factors", Official Monography, Edited by EUROSTAT.

Chopin, N., and F. Pelgrin,(2004),"Bayesian inference and state number determination for hidden Markov models : an introduction to the information content of the yield curve about inflation",*journal of Econometrics*, Vol. 123(2), pp. 327-344.

Dablemont, S., S. Van Belleghem and M. Verleysen,(2007),"Modelling and Forecasting financial time series of "tick data" by functional analysis and neural networks",*Forecasting Financial markets* (FFM2007)

de Freitas, J.F.G. "*Bayesian Methods for Neural Networks*", PhD thesis, University of Cambridge, England, 1999.

Del Moral, P. "*Feynman-Kac formulae. Genealogical and interacting particle approximations*", Springer New York, 2004,

Doucet, A., "*On Sequential Monte Carlo Methods for Bayesian Filtering*", 1998, Technical Report, University of Cambridge, UK, Departement of Ebgineering.

Doucet, A., N. de Freitas, and N. Gordon, "*Sequential MonteCarlo Methods in Practice*", Springer-Verlag, 2001.

Doucet, A. "*Monte Carlo Methods for Bayesian Estimation of Hidden Markov Models: Application to radiation Signals*", PhD thesis, University Paris-Sud, Orsay, France, 1997

Haykin, S.,(1999), " Neural Networks - A comprehensive foundation ", *Prentice Hall*,

Haykin, S. "*Kalman filtering and Neural Networks*", Wiley, 2001.

Julier, S.J. "*Comprefensive Process Models for High-Speed Navigation*", PhD thesis, University of Oxford, England, 1997.

Kalman, R.E. "A new approach to linear filtering and prediction problems", *Transactions of the ASME, Journal of basic Engineering*, Vol. 82, pp. 34-45, March 1960

Lopez, H.F., and C. Marigno,(2001), "A particle filter algorithm for tha Markov switching srtochastic volatility model", Working paper, University of Rio de Janeiro.

Pitt, M., and N. Shephard ,(1999), "Filtering via Simulation : Auxiliary particle Filters",*Journal of the American Statistical Association*, Vol. 94(446), pp. 590-599.

van der Merwe, R., N. de Freitas, A. Doucet and E. Wan, "The Unscented Particle Filter", in *Advances in Neural Information Processing System*, Vol. 13, pp. 584-590, Nov 2001.

van der Merwe, R. "*Sigma-Point Kalman Filters for Probabilistic Inference in Dynamic State-Space Models*", PhD thesis, OGI School of Science and Engineering, Oregon Health and Science University, 2004.

Villaverde, J.F., and J.F. Ramirez,(2004a), "Estimating nonlinear Dynamic Equilibrium Economies: A likelihood Approach", Atlanta FED, Working paper N. 03.

Villaverde, J.F., and J.F. Ramirez,(2004b), "Estimating nonlinear Dynamic Equilibrium Economies: Linear versus Nonlkinear Likelihood", Atlanta FED, Working paper N. 01.

Wan, E.A., and R. van der Merwe, "The Unscented Kalman Filter for nonlinear estimation", in *Proceedings of IEEE Symposium 2000 (AS-SPCC)*, IEEE, Lake louise, Alberta, Canada. 2000.

Wan, E.A., and R. van der Merwe, "Chapter 7 : The Unscented Kalman Filter", in *Kalman Filtering and Neural Networks*, S. Haykin, Ed., Wiley Publishing, 2001