UNIVERSITÉ CATHOLIQUE DE LOUVAIN
ÉCOLE POLYTECHNIQUE DE LOUVAIN
MACHINE LEARNING GROUP

Forecasting of High Frequency Financial Time Series

SIMON DABLEMONT

Thèse soutenue
en vue de l'obtention du grade de
Docteur en Sciences de l'Ingénieur

Membres du Jury:

Prof. **L. Vanderdorpe**, Président (Université catholique de Louvain, Belgique)
Dr. **C. Archambeau** (University College London, United Kingdom)
Prof. **G. Hübner** (HEC Management school - Université de Liege, Belgique)
Prof. **A. Ruttiens**, Membre du comité d'accompagnement (Head of Laxmi Hedge Fund, Prof. ESCP-EAP et Université Paris 1 - Sorbonne, France)
Prof. **S. Van Bellegem**, Promoteur (Université catholique de Louvain, Belgique)
Prof. **M. Verleysen**, Promoteur (Université catholique de Louvain, Belgique)

Novembre 2008

Remerciements

Mes remerciements iront à l'Université catholique de Louvain qui m'a permis d'atteindre cet objectif, parfois par des chemins très détournés.

Je remercierai mon promoteur, le Professeur Michel Verleysen, de m'avoir accepté dans son équipe, ainsi que les Professeurs Alain Ruttiens et Sébastien Van Belleghem, pour leur aide et conseils avisés. Je tiens aussi à remercier les lecteurs d'avoir accepté de participer à mon jury.

Ma plus profonde gratitude ira à mon épouse, Claire, pour sa patience et les encouragements qu'elle m'a toujours prodigués lors de ces investigations.

pour mon fils,

Abstract

Our objective is to forecast financial market, using high-frequency "tick-by-tick" time series without resampling, on an interval of three hours, for speculation.

To carry out this goal, we test two models:

A **Price Forecasting Model** realizes prices forecasting using a *Functional Clustering* combined with data smoothing by splines and local *Neural Networks* forecasting models.

A **Model of Trading** forecasts the first stopping-time, when an asset crosses for the first time a threshold selected by the trader.
This model combines a *Price Forecasting Model* for the prediction of the market trend, and a *Trading Recommendation* for the prediction of the first stopping time. We use a *Dynamic State Space Model*, combined with *Particle* and *Kalman Filters* algorithms for parameter estimation.

At first sight, these models could be a good opportunity for speculators, to trade on a very short horizon.

1

Notations

We define the general notations used in the text. Specific notations will be defined in the chapters were they are used

General Notations

a	Scalar.
$\mathbf{x}$	Vector.
$\mathbf{M}$	Matrix.
$\mathcal{S}$	Set.
$\mathcal{F}$	Algebra.

Symbols

$\mathbf{A}_{i,j}$	Entry of the matrix $\mathbf{A}$ in the i-th row and j-th column.
$\mathbf{I}_n$	Identity matrix of dimension $n \times n$.
$\mathbf{z}_{1:k}$	Stacked vector $\mathbf{z}_{1:k} = (z_1, z_2, \cdots, z_k)^T$.
$\mathbf{Z}_{1:k}$	Stacked matrix $\mathbf{Z}_{1:k} = \{\mathbf{z}_1, \mathbf{z}_2, \cdots, \mathbf{z}_k\}$.
$\mathbb{R}_n$	Euclidean n-dimensional space.
$\mathbb{N}$	The set of natural numbers (positive integers).
$p(\mathbf{z})$	Distribution of continuous random $\mathbf{z}$.
$p(\mathbf{z}\|\mathbf{y})$	Conditional distribution of $\mathbf{z}$ given $\mathbf{y}$.
$p(\mathbf{z}, \mathbf{y})$	Joint distribution of $\mathbf{z}$ and $\mathbf{y}$.
$\mathbf{z} \sim p(\mathbf{z})$	$\mathbf{z}$ is distributed according to $p(\mathbf{z})$.
$\mathbf{z}\|\mathbf{y} \sim p(\mathbf{z})$	The conditional distribution of $\mathbf{z}$ given $\mathbf{y}$ is $p(\mathbf{z})$.
$P(\mathbf{z})$	Probability of discrete random variable $\mathbf{z}$.
$P(\mathbf{z}\|\mathbf{y})$	Conditional probability of discrete variable $\mathbf{z}$.
$p(\mathbf{z}) \sim q(\mathbf{z})$	$p(\mathbf{z})$ is proportional to $q(\mathbf{z})$.
t	Continuous time.
k	Time point in discrete time.

Standard probability distributions

Gaussian	$\mathcal{N}(\mathbf{z}; \boldsymbol{\mu}, \boldsymbol{\Sigma})$	$\det(2\pi\boldsymbol{\Sigma})^{-\frac{1}{2}} \exp\big(-\frac{1}{2}(\mathbf{z}-\boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1}(\mathbf{z}-\boldsymbol{\mu})\big)$.
Uniform	$U_{\mathcal{A}}$	$[\int_{\mathcal{A}} d\mathbf{z}]^{-1} 1_{\mathcal{A}}(\mathbf{z})$.

Operators and functions

$\mathbf{A}^T$	Transpose of matrix $\mathbf{A}$.
$\mathbf{A}^{-1}$	Inverse of matrix $\mathbf{A}$.
$tr\mathbf{A}$	Trace of matrix $\mathbf{A}$.
$\det \mathbf{A}$	Determinant of matrix $\mathbf{A}$.
$1_{\mathcal{E}}(z)$	Indicator function of the set $\mathcal{E}$ (1 if $\mathbf{z} \in \mathcal{E}$, 0 otherwise).
$\delta_{\mathbf{z}_i}(d\mathbf{z})$	Dirac delta function (impulse function).
$\mathbb{E}(\mathbf{z})$	Expectation of the random variable $\mathbf{z}$.
$\mathrm{var}(\mathbf{z})$	Variance of the random variable $\mathbf{z}$.
$\log(.)$	logarithmic function.
$\min$, $\max$	Extrema with respect to an integer value.
$\inf$, $\sup$	Extrema with respect to a real value.
$\arg\min_{\mathbf{z}}$	The argument $\mathbf{z}$ that minimizes the operand.
$\arg\max_{\mathbf{z}}$	The argument $\mathbf{z}$ that maximizes the operand.

1.0.1 State Space Modelization in Discrete Time

$\mathbf{x}_k \in \mathbb{R}^n$	state at time step k,
$\mathbf{x}_{k\|k}$	filtered state at time step k, given observations until time step k,
$\mathbf{x}_{k\|k-1}$	predicted state at time step k, given observations until time step $k-1$,
$\mathbf{y}_k \in \mathbb{R}^m$	measurement at time step k,
$\mathbf{u}_k \in \mathbb{R}^q$	deterministic input,
$\mathbf{v}_k \in \mathbb{R}^n$	process noise, $\mathbf{v}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}_k)$,
$\mathbf{n}_k \in \mathbb{R}^m$	measurement noise, $\mathbf{n}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{R}_k)$,
$\mathbf{f}_k(\mathbf{x}_k) \in \mathbb{R}^n$	non stationary vector-value state function,
$\mathbf{h}_k(\mathbf{x}_k) \in \mathbb{R}^m$	non stationary vector-value measurement function,

1.0.2 State Space Modelization in Continuous Time

$\mathbf{x}_t \in \mathbb{R}^n$	state at time t,
$\mathbf{y}_t \in \mathbb{R}^m$	measurement at time t,
$\mathbf{f}_t(\mathbf{x}_t, t) : \mathbb{R}^n \times \mathbb{R}_+ \mapsto \mathbb{R}^n$	non stationary drift function,
$\mathbf{L}(\mathbf{x}_t, t) \in \mathbb{R}^{n \times s}$	diffusion coefficient for state equation,
$\mathbf{h}_t(\mathbf{x}_t, t) : \mathbb{R}^n \times \mathbb{R}_+ \mapsto \mathbb{R}^m$	non stationary vector-value measurement function,
$\mathbf{V}(\mathbf{x}_t, t) \in \mathbb{R}^{m \times v}$	diffusion coefficient for measurement equation,
$\boldsymbol{\beta}_t \in \mathbb{R}^s$	Wiener system process with diffusion matrix, $\mathbf{Q}_c(t) \in \mathbb{R}^{s \times s}$
$\boldsymbol{\eta}_t \in \mathbb{R}^v$	Wiener measurement process with diffusion matrix, $\mathbf{R}_c(t) \in \mathbb{R}^{v \times v}$,

2

Programs

We precise the Toolboxes used for the development of applications.

Programs

The application are developed in MATLAB.

Toolboxes

The following Toolboxes are used:

- Matlab toolboxes,
- Spatial Statistics Software and Spatial Data,
  from: http://www.spatial-statistics.com/software-index.htm,
- Bayes Net Toolbox for Matlab [Murphy , 2002]
  from: http://www.cs.ubc.ca/ murphyk/Software/BNT/bnt.html,
- ReBEL [van der Merwe , 2004]
  from: http://choosh.bme.ogi.edu/rebel,
- NETLAB [Bishop , 2000], and [Nabney , 2001]
  from: http://www.ncrg.aston.ac.uk/netlab/,
- NNSYSID [Norgaard , 2000]
  from: http://www.iau.dtu.dk/research/control/nnsysid.html,
- Functional Data Analysis in R, SPLUS and Matlab [Ramsay, and Silverman , 1997]
  from: ftp://ego.psych.mcgill.ca/pub/ramsay/FDAfuns/Matlab/,

- SDELab [Gilsing and Shardlow , 2005]
  from: http://www.ma.man.ac.uk/ sdelab
  http://wws.mathematik.hu-berlin.de/ gilsing/sdelab,

- MAPLE for Stochastic Differential [Cyganowski et al. , 2005]
  from: http://www.math.uni-frankfurt.de/ numerik/maplestoch/,

- SDE Tollbox [Picchini , 2007]
  from: http://sourceforge.net/projects/sdetoolbox/,

Algorithms

- algorithms coming from Toolboxes are simply sketched,

- algorithms adapted from Toolboxes or developed are detailed in Appendix.

Contents

3

Introduction

3.1 Introduction to Financial Models

The analysis of financial time series is of primary importance in the economic world. This thesis deals with a data-driven empirical analysis of financial time series. The goal is to obtain insights into the dynamics of series and out-of-sample forecasting.

Forecasting future returns on assets is of obvious interest in empirical finance. If one was able to forecast tomorrow's returns on an asset with some degree of precision, one could use this information in an investment today. Unfortunately, we are seldom able to generate a very accurate prediction for asset returns.

Financial time series display typical nonlinear characteristics, it exists clusters within which returns and volatility display specific dynamic behavior. For this reason, we consider here nonlinear forecasting models, based on local analysis into clusters. Although financial theory does not provide many motivations for nonlinear models, analyzing data by nonlinear tools seems to be appropriate, and they are at least as much informative as an analysis by more restrictive linear methods.

Time series of asset returns can be characterized as serially dependent. This is revealed by the presence of positive autocorrelation in squared returns, and sometimes in the returns too. The increased importance played by risk and uncertainty considerations in modern economic theory, has necessitated the development of new econometric time series techniques that allow the modeling of time varying means, variances and covariances.

Given the apparent lack of any structural dynamic economic theory explaining the variation in the second moment, econometricians have thus extended traditional time series tools such as AutoRegressive Moving Average (ARMA) models [Box and Jenkins, 1970] for the conditional means and equivalent models for the conditional variance. Indeed, the dynamics observed in the dispersion is clearly the dominating feature in the data. The most widespread modeling approach to capture these properties is to specify a dynamic model for the conditional means and the conditional variance, such as an ARMA-GARCH model or one of its various extensions [Engle, 1982], [Hamilton, 1994].

The Gaussian random walk paradigm - under the form of the diffusion geometric Wiener process - is the core of modeling of financial time series. Its robustness mostly suffices to keep it as the best foundation for any development in financial modeling, in addition to the fact that, in the long run, and with enough spaced out data, it is almost verified by the facts. Failures in its application are however well admitted on the (very) short term (market microstructure) [Fama, 1991], [Olsen et al., 1992], [Franke et al., 2002]. We think that, to some extent, such failures are actually caused by the uniqueness of the modeling process.

The first breach in such a unique process has appeared with two-regime or switching processes [Diebold et al., 1994], which recognizes that a return process could be originated by two different stochastic differential equations. But in such a case, the switch is governed by an exogenous cause (for example in the case of exchange rates, the occurrence of a central bank decision to modify its leading interest rate or to organize a huge buying or selling of its currency through major banks).

Market practitioners [Engle, 1982], however, have always observed that financial markets can follow different behaviors over time, such as overreaction, mean reversion, etc..., which look like succeeding each other with the passage of time. Such observations justify a rather fundamental divergence from the classical modeling foundations. More precisely, financial markets should not be modeled through a single process, but rather through a succession of different processes, even in absence of exogenous causes. Such a multiple switching process should imply, first, the determination of a limited number of competitive sub-processes, and secondly, the identification of the factor(s) causing the switch from one sub-process to another. The resulting model should not be Markovian, and, without doubt, would be hard to determine.

3.2 Purpose of the Thesis

This thesis describes the design of numerical models and algorithms, for the forecasting of financial time series, for speculation on a short time interval.

To this aim, we will use two models:

- a *Price Forecasting Model* predicts asset prices on an interval of three hours,
- a *Model of Trading* predicts "high" and "low" of an asset, on an interval of three hours,

using high-frequency "tick-by-tick" financial time series, without resampling (following Olsen's publications for these terms [Lynch and Zumbach, 2001], [Olsen et al., 1992], [Zumbach et al., 2002]), and without other economic information.

If we could forecast the future of an asset, we could use this information to realize an optimum trading, and make profits, but is it possible?

Fig.(3.1) shows the evolution of the Prices (top) and Volumes (bottom) of the IBM asset on one day.

Fig.(3.2) shows the distribution of the transactions for the same day. Each point is a transaction, with more transactions at the opening and closing of the NYSE. These transactions are sampled discretely in time and like it is often the case with financial data the time separating successive observations is itself random.

In this thesis, we present modelization and forecasting tools, and under which conditions we can predict the future of an asset.

3.3 Contribution of the Thesis

The main contributions of this thesis can be summarized as follow:

- The development of procedures and algorithms to forecast financial time series, using the *Functional Clustering Analysis*, the *Smoothing* of raw observations by splines, and the prediction by "*Artificial Neuronal Models*".

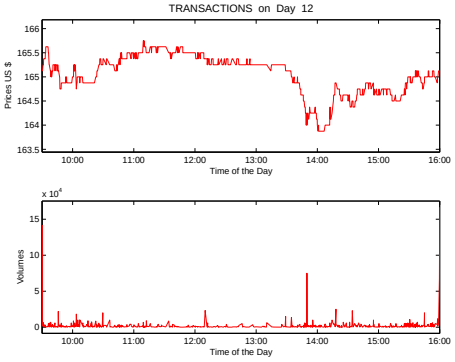

Figure 3.1: Prices (Top) et Volumes (Bottom)

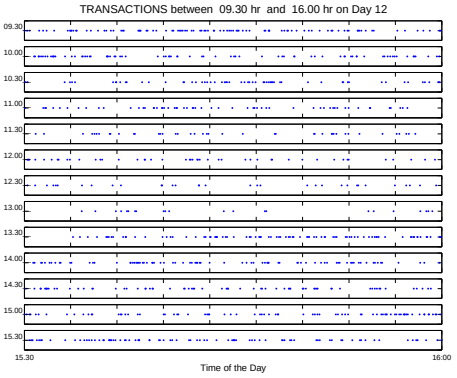

Figure 3.2: Distribution of transactions

- The development of procedures and algorithms of prediction of "high" and "low" values of an asset, using the "*Functional Analysis*", the "*Smoothing*" of observations by splines, "*Dynamic State Space Models*" representations, with "*Stochastic Differential Equations*" in continuous and discrete time, and "*Parameter Estimations*" by "*Stochastic Bayes's Filters*" such as "*Particle Filters*" and "*Kalman Filters*" in continuous and discrete time.

3.4 Structure of the Thesis

3.4.1 Part I - Forecasting Models

Price Forecasting Model

In Chapter [4], we present a *Price Forecasting Model* based on *Functional Clustering* and *Neural Networks*.

The aim of this model is, as a first step, to at least empirically verify that a multiple switching process leads to better short-term forecasting.

Model of Trading

In Chapter [5], we present a *Model of Trading* which forecasts the first stopping-time, when an asset crosses for the first time a threshold selected by the trader.

This model combines a *Price Forecasting Model* for the prediction of the market trend, and a *Trading Recommendation* for the prediction of the first stopping

time.

For the prediction of the market trend, we use a Functional Clustering and local Neural Networks forecasting models.

For the prediction of the first stopping time, we use an auto-adaptative *Dynamic State Space Model*, with *Particle* and *Kalman Filters* for parameter estimation.

3.4.2 Part II - Financial introduction

In Chapter [6] we introduce the specificities of financial markets and financial time series.

3.4.3 Part III - Mathematical Developments

Functional Analysis for Classification and Clustering

Chapter [7] addresses the problem of smoothing of observations sparsely and irregularly spaced, or occur at different time points for each individual, as with high-frequency financial time series, using Functional Data Analysis.

Bayesian Filtering

In Chapter [8] we introduce the Stochastic Bayes' Filters, such as Kalman Filters, and Particle Filters in Discrete Time.

Stochastic Differential Equations and Filtering

In Chapter [9] we introduce the Stochastic Filters in Continuous-Discrete Time, using Stochastic Differential Equations in Dynamic State Space Representations.

3.4.4 Part IV - Experiment Results

Experiments

In Chapter [10] we present the results of predictions of assets with "Price Forecasting Model", and "Model of Trading".

Conclusions

Chapter [11] presents some conclusions, and highlights areas of possible future research in this field.

3.4.5 Part V - Appendix

Chapter [Appendix] presents the mathematical developments, and algorithms.

Part I

Forecasting Models

Part One describes two Forecasting Models: a "Price Forecasting Model" for asset price predictions on an interval of three hours, and a "Trading Model" for "high" and "low" prediction of an asset.

4

Price Forecasting Model

A "Price Forecasting Model" forecasts the behavior of an asset for an interval of three hours.

The purpose of this chapter is to provide a description of a "*Price Forecasting Model*" based on a *Functional Clustering* and a smoothing by *cubic-splines* in the training phase, and a *Functional Classification* for generalization.

A "*Price Forecasting Model*" forecasts the behavior of an asset for an interval of three hours from high-frequency "tick-by-tick" financial time series without resampling.

For the data, we only use prices and implied volatility of this asset but no other economical information.

4.1 Introduction

The aim of this model is, as a first step, to at least empirically verify, with the help of functional clustering and neural networks, that a multiple switching process leads to better short-term forecasting based on an empirical functional analysis of the past of series (see Chapter [7]).

We build a "Price Forecasting Model" to predict prices of an asset for an interval of three hours. (see: Fig. [4.1]) by mendelizing characteristic patterns into time series, in a methodology similar at the "Technical Analysis" (see: [Lo et al., 2000]).

An originality of this method is that it does not make the assumption that a single model is able to capture the dynamics of the whole series. On the contrary, it splits the past of the series into clusters, and generates a specific local neural model for each of them. The local models are then combined in a prob-

abilistic way, according to the distribution of the series in the past.

This forecasting method can be applied to any time series forecasting problem, but is particularly suited for data showing nonlinear dependencies, cluster effects and observed at irregularly and randomly spaced times like high-frequency financial time series do.

One way to overcome the irregular and random sampling of "tick-data" is to resample them at low frequency, as it is done with "Intraday". However, even with optimal resampling using say five minute returns when transactions are recorded every second, a vast amount of data is discarded, in contradiction to basic statistical principles. Therefore modeling the noise and using all the data is a better solution, even if one misspecifies the noise distribution [Ait-Sahalia and Myland, 2003]. One way to reach this goal is to use *Functional Analysis* as done in this tart. This Pricing Model should give a prediction of the future evolution of prices for an interval of tree hours.

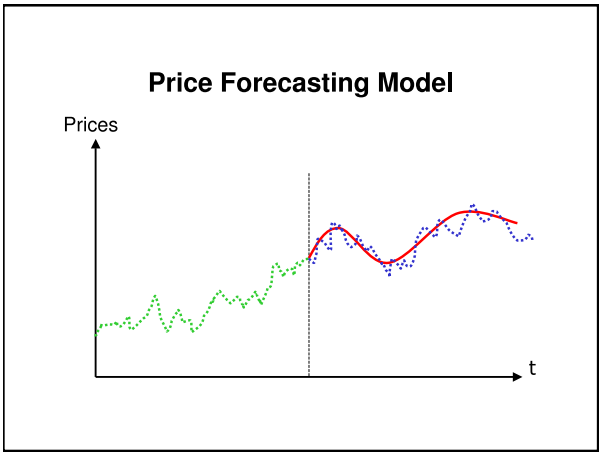

Figure 4.1: With this Price Forecasting Model we would like to predict prices of an asset for an interval of tree hours.

4.2 Methodology

In this section we describe the data we will use and the modelization which consists in finding characteristic patterns in time series, as for Technical Analysis used by practitioners for extracting useful information from markets prices.

4.2.1 Data

We use high-frequency financial "tick-by-tick" time series, without resampling, and we divide the data set into three parts: the *Training set* to build the models, the *Validation set* to choose the "best" model according to a criterion, and the *Test set* to measure its performances, as following:

- we fit models on the Training set,
- we select the "best" model using the Validation set,
- we assess prediction error using Test set.

All three data sets have to be representative samples of the data that the model will be applied to.

4.2.2 Patterns

Similarly to [Lo et al., 2000] in their description of Technical Analysis, we consider than prices evolve in a nonlinear fashion over time and that nonlinearities contain certain regularities or patterns. The presence of clearly identified support and resistance levels with a one-third retracement parameter suggest the presence of strong buying and selling opportunities in the near-term. Using such patterns a trader can decide to trade in order to yield profit. Technical analysis employs the tools of geometry and pattern recognition in a "visual" procedure.

To capture such patterns we consider that prices $S(t)$ can be described by an expression such as:

$$S(t) = m(t) + \epsilon(t) \tag{4.1}$$

where $m(t)$ is an arbitrary nonlinear function, and $\epsilon(t)$ is a white noise.

In Fig. [4.2] we have a representation of one of the numerous patterns used by the traders.

More specifically, the approach presented in this Section consists in several steps. First, the observed series is split into fixed time-length intervals (for example one day); the resulting data contain a variable number of observations, which are in addition irregularly sampled. Therefore these data cannot

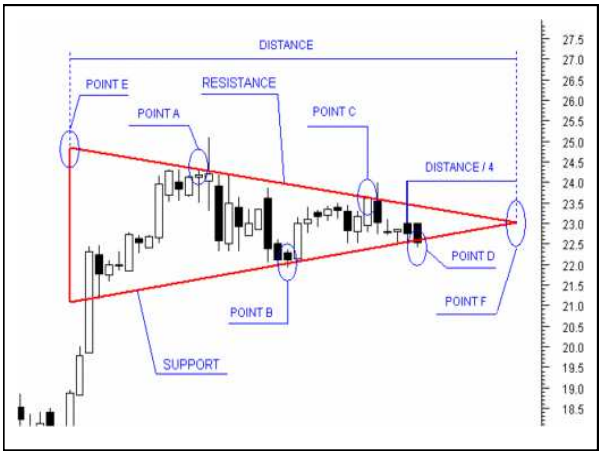

Figure 4.2: Pattern.

be compared by means of standard distance measure, which is a fundamental problem for most data analysis methods.

To overcome this problem, the rough tick data are projected onto a functional basis (for example cubic splines). Whatever is the number of observations in each data interval, the projection results in a fixed number of features. More conventional data analysis tools may thus be used on these features.

In particular, as we intend to build local models on subsets of data showing similar properties, the idea is to use clustering tools to group similar data together. The clustering is applied onto the coefficients of the projection on a functional basis.

Let us come back to the objective of this work. The final goal is to forecast time series of assets; a forecasting model should thus be designed.

4.2.3 Training phase

In the training phase, input-output pairs of data must be collected, and used to learn a supervised prediction model. In our context, both inputs and outputs consist in functional features extracted, as detailed below, from (respectively past and future) intervals of the series at hand. Implied volatility time series are also used as inputs, but not for clustering.

When input-output pairs are identified, they are split into subsets according to the clusters defined in the functional projection-clustering step. A local model is then trained on each subset. Training separate models allows following more closely the complex dynamics of a series. In this work, Radial-Basis Function Networks are used for each local model. A contingency table is also build, to evaluate the matching between clusters of the input and output series.

It should be noted that performing the projection and the clustering into two subsequent, separate steps has some limitations as detailed in Section [7.3]. We will thus perform the projection and clustering tasks in a single step. The result of this step is twofold:

- first, each data will be modeled by a fixed-size set of features,
- secondly, each data will be associated to one and only one cluster.

The clusters are themselves identified simultaneously in this step (only their number has to be fixed in advance).

4.2.4 Forecasting phase

Finally, performing the forecasting itself consists in getting the features from a new sample, and using the local models to generate predictions.

This sample consists in observations at random time-points, and very often we have interval of time with only a few observations. To get an optimal smoothing of these data, we must realize a functional classification and the smoothing in the same step.

The spline coefficients of this sample are used as input for each local model to get the local predictive cure. The local models used are those that correspond to the cluster associated to the input data at hand. The local predictions are then weighted according to the entries of the contingency table, to generate a single, global prediction.

Chapter [7] details functional classification and smoothing, and algorithms are given in Appendix [H.1.1] and [H.2.1].

4.3 Procedure

In this section we present a detailed model-based approach for clustering functional data and a time series forecasting method. This method will first be sketched to give an intuition of how the forecasting is performed. Then each step of the method will be detailed.

4.3.1 Method Description

The forecasting method is based on the "looking in the past" principle.

Let the observations belongs to the time interval $[t_0, T]$. To perform a functional prediction of the curve for the time interval $[t_n, t_n + \triangle t_{out}]$, we create two functional spaces.

A first functional space **IN** is built with past observations in time intervals $[t - \triangle t_{in}, t]$, the "regressors"; a similar second functional space **OUT** is built with observations in time intervals $[t - \triangle t_{in}, t + \triangle t_{out}]$. These two spaces are built with all data corresponding to times $t \in [t_0 + \triangle t_{in}, T - \triangle t_{out}]$, by intervals of $\triangle t$, such as $t \in \{t_1, t_2, \cdots t_N\}$, where

$$\begin{pmatrix} t_1 \\ t_2 \\ \vdots \\ t_N \end{pmatrix} = \begin{pmatrix} t_0 + \triangle t_{in} \\ t_1 + \triangle t \\ \vdots \\ t_{N-1} + \triangle t \end{pmatrix}. \tag{4.2}$$

In this equation N is chosen such that $t_N \leq T - \triangle t_{out}$, and $\triangle t_{in}$ and $\triangle t_{out}$ are the interval of time for the "regressors" and for the "prediction", respectively. $\triangle t$ represents the interval between times t_j and t_{j+1}, and there is no relationships between $\triangle t$ and $\triangle t_{in}$ or $\triangle t_{out}$.

These functional spaces are combined into a probabilistic way to build the functional prediction for the time interval $[t, t + \triangle t_{out}]$; they are quantized using the functional clustering algorithm. The relationship between the first and the second functional spaces issued from the clustering algorithms is encoded into a transition probability matrix constructed empirically on the datasets. For example, with 4 clusters *IN* and 3 clusters *OUT*, the transition probability matrix is

$$\begin{pmatrix} p(1,1)\ p(1,2)\ p(1,3) \\ p(2,1)\ p(2,2)\ p(2,3) \\ p(3,1)\ p(3,2)\ p(3,3) \\ p(4,1)\ p(4,2)\ p(4,3) \end{pmatrix} \tag{4.3}$$

where $p_{i,j}$ is the transition probability from cluster *IN* number i to cluster *OUT* number j.

In each of the clusters determined by the second clustering OUT, a local RBFN model is built to approximate the relationship between the functional output (the local prediction) and the functional input (the regressor).

Finally, the global functional prediction at time t_n for the interval $[t_n, t_n + \triangle t_{out}]$ is performed by combining the local model results associated to clusters OUT, according to their frequencies with respect to the class considered in the cluster IN, and using the interval $[t_n - \triangle t_{in}, t_n]$ as regressor for the local model.

4.3.2 Training phase

In the Training Phase, we smooth the high frequency "tick-by-tick" data observed at random time points, and we work with the spline coefficient vectors (see Fig. (4.3)).

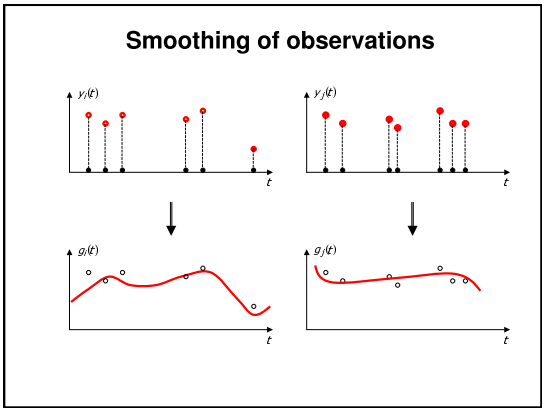

Figure 4.3: We start with observed data at random time-point. We smooth these data by splines, and we will work with the vectors of spline coefficients.

Quantizing the « inputs »

Consider a time series of scalars $\{x(.)\}$, where $x(t)$ is the value at time t, $(t \in [t_0, T])$. This original series is transformed into an array of N vectors of past observations $\{\mathbf{X}_{in}\}$:

$$\mathbf{X}_{in} = \begin{pmatrix} \mathbf{x}(1) \\ \mathbf{x}(2) \\ \mathbf{x}(3) \\ \vdots \\ \mathbf{x}(N) \end{pmatrix} = \begin{pmatrix} \mathbf{x}\big([t_1 - \triangle t_{in}, t_1]\big) \\ \mathbf{x}\big([t_2 - \triangle t_{in}, t_2]\big) \\ \mathbf{x}\big([t_3 - \triangle t_{in}, t_3]\big) \\ \vdots \\ \mathbf{x}\big([t_N - \triangle t_{in}, t_N]\big) \end{pmatrix} \tag{4.4}$$

with $t_i = t_{i-1} + \triangle t$ and N such as $t_N \leq T - \triangle t_{out}$.

Then the functional clustering algorithm [H.1.1] is applied to this *input* array $\mathbf{X}_{in}$; after convergence it gives K_{in} clusters represented by their centroids $\{\mathbf{C}_{in}\}$ and the spline coefficients for each curve in this IN map.

Quantizing the « outputs »

At each *input* vector of observations of the array $\{\mathbf{X}_{in}\}$ we aggregate the *future* observations to get a new array $\{\mathbf{Y}_{out}\}$ of vectors of observations of the "past" and "future" values,

$$\mathbf{Y}_{out} = \begin{pmatrix} \mathbf{y}(1) \\ \mathbf{y}(2) \\ \mathbf{y}(3) \\ \vdots \\ \mathbf{y}(N) \end{pmatrix} = \begin{pmatrix} \mathbf{x}\big([t_1 - \triangle t_{in}, t_1 + \triangle t_{out}]\big) \\ \mathbf{x}\big([t_2 - \triangle t_{in}, t_2 + \triangle t_{out}]\big) \\ \mathbf{x}\big([t_3 - \triangle t_{in}, t_3 + \triangle t_{out}]\big) \\ \vdots \\ \mathbf{x}\big([t_N - \triangle t_{in}, t_N + \triangle t_{out}]\big) \end{pmatrix} \tag{4.5}$$

Then the functional clustering algorithm [H.1.1] is applied to this $output$ array $\mathbf{Y}_{out}$; after convergence it gives K_{out} clusters represented by their centroids $\{\mathbf{C}_{out}\}$ and the spline coefficients for each curve in this OUT map.

By construction there is a one-to-one relationship between each $input$ and each $output$ vector of spline coefficients (see Fig. (4.4), and Fig. (4.5)).

Probability transition table

Both sets of codewords from maps *IN* and *OUT* only contain a static information. This information does not reflect completely the evolution of the time series.

The idea is thus to create a data structure that represents the dynamics of the time series, i.e. how each class of $output$ vectors of spline coefficients (including the values for the time interval $[t, t + \triangle t_{out}]$) is associated to each class of $input$ vectors of spline coefficients (for the time interval $[t - \triangle t_{in}, t]$). This structure is the probability transition table $\{p(i, j)\}$, with $1 \leq i \leq K_{in}$, $1 \leq j \leq K_{out}$.

$$\mathbf{P} = \begin{pmatrix} p(1,1) & p(1,2) & \cdots & p(1,K_{out}) \\ p(2,1) & p(2,2) & \cdots & p(2,K_{out}) \\ \vdots & \vdots & \ddots & \vdots \\ p(K_{in},1) & p(K_{in},2) & \cdots & p(K_{in},K_{out}) \end{pmatrix}. \tag{4.6}$$

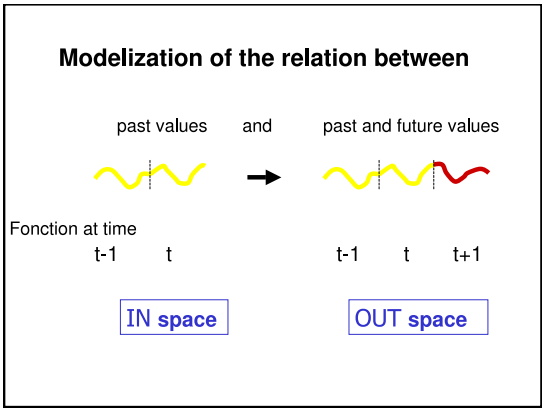

Figure 4.4: We build two functional IN and OUT spaces from spline coefficients.

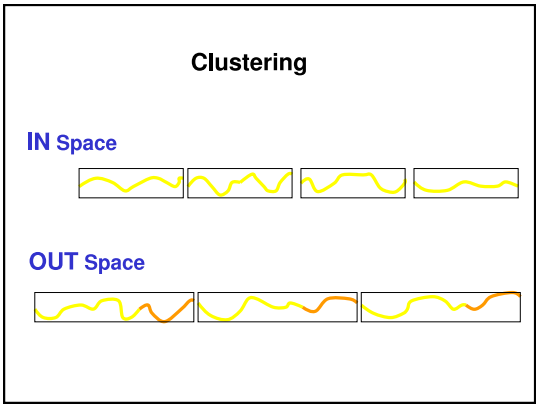

Figure 4.5: We realize the clustering for IN and OUT spaces.

Each element $p(i, j)$ in this table represents the proportion of *output* vectors that belongs to class j of the *OUT* map while their corresponding *input* vectors belong to class i of the *IN* map. Those proportions are computed empirically for the given dataset and sum to one in each line of the table (see Fig. (4.6)).

Intuitively the probability transition table represents all the possible evolutions at a given time t together with the probability that they effectively happen.

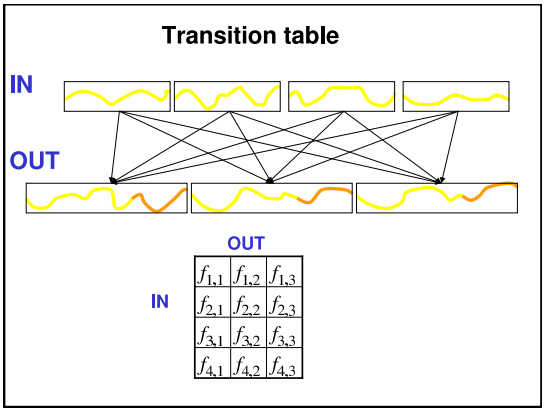

Figure 4.6: We create a Transition table between IN and OUT spaces.

Local RBFN models

When applied to the « outputs », the functional clustering algorithm provides K_{out} classes and the spline coefficients of the curves for the intervals $[t, t + \triangle t_{out}]$. In each of these classes, a RBFN model is learned. Each RBFN model has p inputs, the spline coefficients of the regressors, and r outputs, the spline coefficients of the prediction curve.

For each cluster k of the *OUT* map, $(k = 1, \ldots, K_{out})$, we use the spline coefficients of the past value observations $\mathbf{x}\big([t_j - \triangle t_{in}, \ldots, t_j]\big)$ in the time interval $[t_j - \triangle t_{in}, \ldots, t_j]$ as input, and the spline coefficients of future values observations $\mathbf{x}\big([t_j, \ldots, t_j + \triangle t_{out}]\big)$ in the time interval $[t_j, \ldots, t_j + \triangle t_{out}]$ as output to train the RBFN, with $\{j = 1, \ldots, N_k\}$ where N_k is the number of curves in the cluster k. Remind that the data are observed at irregularly and random time-points, thus the number of observations in $\mathbf{x}\big([t_j - \triangle t_{in}, \ldots, t_j]\big)$ and in $\mathbf{x}[(t_j), \ldots, t_j + \triangle t_{out}]$ are random and depend on the time t_j.

These models represent the local evolution of the time series, restricted to a specific class of regressors. The local information provided by these models

will be used when predicting the future evolution of the time series (see Fig. (4.7)).

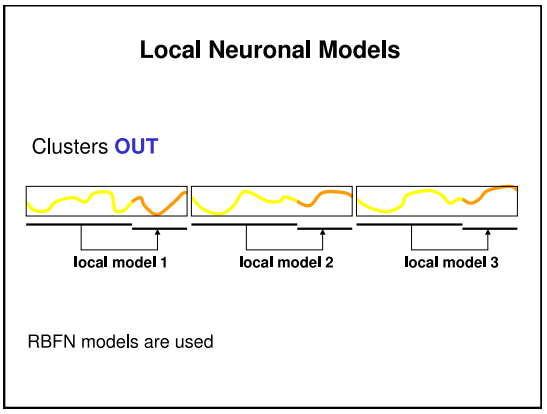

Figure 4.7: We build a Neural Local Model for each cluster OUT .

4.3.3 Forecasting

Classification and Smoothing

The relevant information has been extracted from the time series through both maps, the probability transition table and the local RBFN models detailed in the previous sections. Having this information, it is now possible to perform the forecasting itself.

At each time t, the purpose is to forecast the future functional curve for the time interval $[t, t + \triangle t_{out}]$, denoted $\hat{\mathbf{y}}\big([t, t + \triangle t_{out}]\big)$, from the functional curve of regressor for the interval $[t - \triangle t_{out}, t]$, denoted $\mathbf{x}\big([t - \triangle t_{in}, t]\big)$.

First the *input* vector of observations $\mathbf{x}\big([t - \triangle t_{in}, t]\big)$ is built, and by smoothing we get the spline coefficients input vector $\mathbf{v}(t)$ for the spline $\mathbf{s}(t)$. The vector of coefficients $\mathbf{v}(t)$ is presented to the *IN* map, and by classification, the nearest centroid $\mathbf{c}_k$ from the set of all centroids $\mathbf{C}_{in}$ of the *IN* map is identified ($1 \leq k \leq K_{in}$).

The functional classification algorithm [H.2.1] is applied onto the rough randomly observed sample to realize the classification and the smoothing of these

observations.

In line k of the frequency table, we have columns corresponding to classes of the *OUT* map for which the transition probabilities are non zero. This means that those columns represent possible evolutions for the considered input spline $\mathbf{s}(t)$, since $\mathbf{s}(t)$ has the same shape than splines in this class k (see Fig. (4.8)).

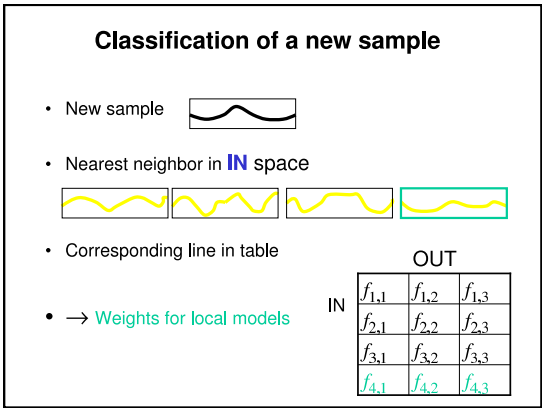

Figure 4.8: We realize the classification of a new sample in the IN space.

Curve prediction

For each of those potential evolutions, the local RBFN models are considered (one RBFN model has been built for each class in the *OUT* map). For each of them, we obtain a local spline forecasting $\widehat{\mathbf{y}}_j$, with ($1 \leq j \leq K_{out}$).

The final forecasting is a weighted sum of the local forecasting, the weights being the frequencies stored in the probability transition table (see Fig. (4.9)).

The global forecasting is thus :

$$\widehat{\mathbf{y}} = \sum_{j=1}^{K_{out}} p(k,j)\, \widehat{\mathbf{y}}_j. \tag{4.7}$$

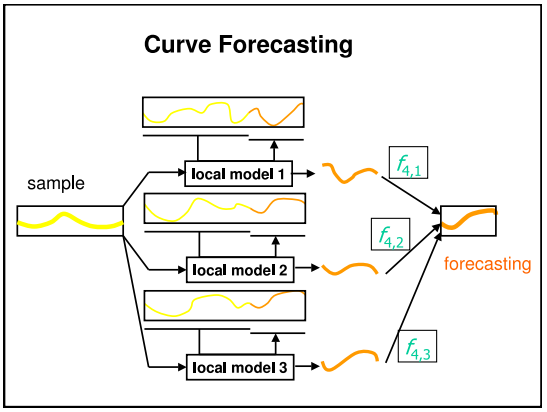

Figure 4.9: We realize the curve forecasting from the sample, given the weights of the transition table.

5

Model of Trading

A "Model of Trading" forecasts the first Stopping-time, the "high" and "low" of an asset, when this asset crosses for the first time a threshold selected by the trader.

The purpose of this chapter is to provide a description of "*Model of Trading*" in *Discrete Time* and *Continuous-Discrete Time* based on a *Dynamic State Space Representation*, using *Stochastic Bayes' Filters* for generalization.

A "*Model of Trading*" forecasts the first Stopping-time, the "high" and "low" of an asset on an interval of three hours, when this asset crosses for the first time a threshold selected by the trader. To this end, we use high-frequency "tick-by-tick" financial time series without resampling.

For the data, we only use prices of this asset but no other economical information.

5.1 Introduction

Based on martingales and stochastic integrals in the theory of continuous trading, and on the market efficiency principle, it is often assumed that the best expectation of the *to-morrow* price should be the *to-day* price, and looking at past data for an estimation of the future is not useful [Harrison and Pliska, 1981]. But if this assumption is wrong, the conclusion that forecasting is impossible is also questionable. Researchers questioned this assumption [Olsen et al., 1992] and [Franke et al., 2002]; moreover, practitioners use the technical analysis on past prices and technicians are usually fond of detecting geometric patterns in price history to generate trading decisions. Economists desire more systematic approaches so that such subjectivity has been criticized. However some studies tried to overcome this inadequacy [Lo et al., 2000]; applying algorithms to the data might help predicting future price moves.

The topic of this part 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.

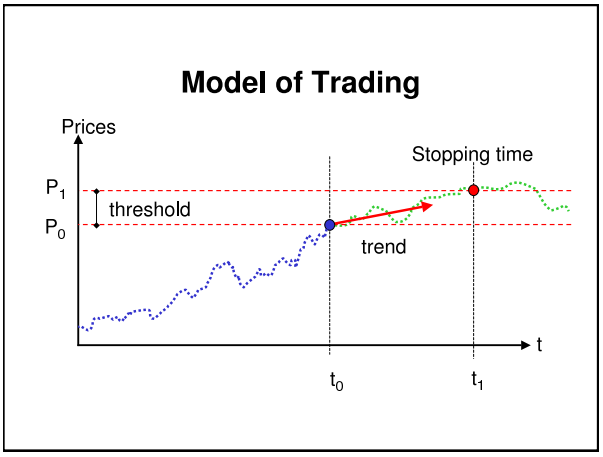

Figure 5.1: In the Trading Process we would like to predict the first stopping time, when prices cross for the first time a threshold defined by the trader.

At time t_0, see Fig. (5.1), the *Price Forecasting Model* forecasts the trend for the next hours. Based on this trend forecasting, the trader can decide to buy or to sell short a share, or to do nothing. The *Trading model* estimates the first stopping time t_1 in order to close the position, depending of a threshold defined by the trader, with a yielding profit. It should also be able to send a stop-loss order

There is a distinction between a price forecasting and a trading recommendation. A trading model includes a price forecasting, but must also account for the specific constraints of the trader 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.

5.1.1 Data

We use high frequency "tick-by-tick" time series observed at random time points. In order to prevent outliers, due to the rush of traders at the closing time of the market, we remove tick data between 15:30 and 16:00. But due to the randomness time of these observations, we cannot use such raw data as input in our models, we must smooth them and use their spline coefficient vectors as variables.

To realize the modelization, we divide the data set into three parts: the *Training set* to fit models, the *Validation set* to select the "best" model according to a criterion, and we assess prediction error using the *Test set*.

All three data sets have to be representative samples of the data that the model will be applied to.

5.1.2 Models of Trading

We will consider two models of trading:

1. a Trading Model, without "Stop Loss", in which:
   - at time t_0, we estimate the futur trend,
   - according to this trend, we realize a trading, buying or selling short an asset,
   - we estimate the stopping time t_1,
   - we close our position at this stopping time.

2. a Trading Model, with "Stop Loss",
   - at time t_0, we estimate the futur trend,
   - we realize a trading,
   - we estimate the stopping time t_1,
   - periodically, after time t_0, we estimate a new trend,
     - if the new trend is similar to the former, we continue,
     - if we have a reversal in trend, immediately, we close our position,
     - we realize an opposite trading,
     - we rerun the trading model to get a new stopping time.

Two models will be used to realize this objective:

1. for "Trend" prediction, we use a "Price Forecasting Model" (see Chapter [4]), based on Functional Clustering for the Training phase, (see Section. [7.3]), and on Functional Classification for the Forecasting phase (see Section. [7.4]),

2. for "Stopping time" prediction, we use a "Dynamic State Space Model" and Particle Filters (see Chap. [8]).

For forecasting with models in Dynamic State Space representation, we must realize parameter and state estimations. In the literature, the problem of estimating states is known as "*Inference*", while the problem of estimating parameters is referred to as "*Learning*".

5.1.3 Trend Model

The observed time series is split into fixed time-length intervals, for example one day for regressor, and three hours for prediction. We cannot directly work with these raw data, we have to realize a functional clustering and a smoothing of these data with splines. We will work with spline coefficient vectors as variables in our models.

Let be $g_i(t)$ the hidden true value for the curve i at time t (asset prices) and $\mathbf{g}_i$, $\mathbf{y}_i$ and $\boldsymbol{\epsilon}_i$, the random vectors of, respectively, hidden true values, measurements and errors. We have:

$$\mathbf{y}_i = \mathbf{g}_i + \boldsymbol{\epsilon}_i, \qquad i = 1, \cdots, N \ , \tag{5.1}$$

where N is the number of curves. The random errors $\boldsymbol{\epsilon}_i$ are assumed i.i.d., uncorrelated with each other and with $\mathbf{g}_i$.

For the trend forecasting, we use a functional clustering model on a q-dimensional space as:

$$\mathbf{y}_i = \mathbf{S}_i \left(\boldsymbol{\lambda}_0 \ + \ \boldsymbol{\Lambda} \, \boldsymbol{\alpha}_{ki} \ + \ \boldsymbol{\gamma}_i \right) + \boldsymbol{\epsilon}_i, \tag{5.2}$$

for $k = 1, \cdots K$ clusters.

with:

- $\boldsymbol{\epsilon}_i \sim N(\mathbf{0}, \mathbf{R})$,
- $\boldsymbol{\gamma}_i \sim N(\mathbf{0}, \boldsymbol{\Gamma})$,
- $\boldsymbol{\Lambda}$, a projection matrix onto a h-dimensional subspace, with $h \leq q$.

$\mathbf{S}_i = \left[\mathbf{s}(t_{i1}), \cdots, \mathbf{s}(t_{in_i}) \right]^T$ is the spline basis matrix for curve i :

$$\mathbf{S}_i = \begin{pmatrix} s_1(t_1) & \cdots & s_q(t_1) \\ \vdots & \ddots & \vdots \\ s_1(t_{n_i}) & \cdots & s_q(t_{n_i}) \end{pmatrix}. \tag{5.3}$$

The term $\boldsymbol{\alpha}_{ki}$ is defined as: $\boldsymbol{\alpha}_{ki} = \boldsymbol{\alpha}_k$ if curve i belongs to cluster k, where $\boldsymbol{\alpha}_k$ is a representation of the centroid of cluster k in a reduced h-dimensional subspace:

$$\boldsymbol{\alpha}_k = \left(\alpha_{k1}, \alpha_{k2} \ldots \alpha_{kh} \right)^T . \tag{5.4}$$

Then we have :

- $\mathbf{s}(t)^T \boldsymbol{\lambda}_0$: the representation of the global mean curve,
- $\mathbf{s}(t)^T (\boldsymbol{\lambda}_0 + \boldsymbol{\Lambda}\boldsymbol{\alpha}_k)$: the global representation of the centroid of cluster k,
- $\mathbf{s}(t)^T \boldsymbol{\Lambda}\boldsymbol{\alpha}_k$: the local representation of the centroid of cluster k in connection with the global mean curve,
- $\mathbf{s}(t)^T \boldsymbol{\gamma}_i$: the local representation of the curve i in connection with the centroid of its cluster k.

See Chap. [7] for a detailed description.

5.1.4 Stopping Time Model

The observed time series is split into fixed time-length intervals of three hours and we realize a functional clustering and a spline smoothing of this raw data set. We use the spline coefficient vectors as measurement in a Dynamic State Space model.

In the test phase every day at 11:00 we realize a trading, thus we must run the model every three hours interval, that means each day at 11:00 and 14:00.

For trading without stop loss, we have regular time intervals of three hours, thus we use discrete time equations for state and measurements (see Fig.(5.2)).

For trading with stop loss, after the stopping time prediction at 11:00, every ten minutes we rerun the trend prediction model. According to the result, we could have to rerun the stopping time model. In this case, we do not have regular time intervals of three hours anymore, and we must use a continuous time equation for state and a discrete time equation for measurements (see Fig.(5.3)).

Trading Model - Without Stop Loss

We would like forecasting the future curve of asset prices for a couple of hours and not only the next value of the time series. Thus, we will derive a parametric Dynamic State Space Model in discrete time, where random variables may be scalars, vectors or curves. This model is formalized by:

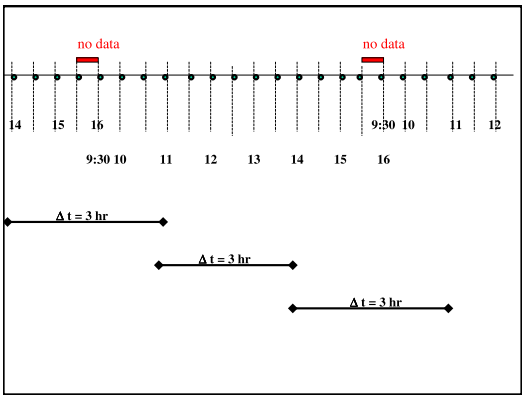

Figure 5.2: Trading without stop loss.

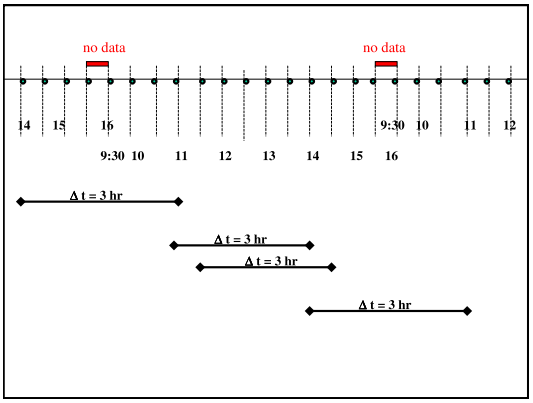

Figure 5.3: Trading with stop loss.

$$\mathbf{x}_k = \mathbf{f}_k(\mathbf{x}_{k-1}) + \mathbf{v}_k \tag{5.5}$$
$$\mathbf{y}_k = \mathbf{h}_k(\mathbf{x}_k) + \mathbf{n}_k, \tag{5.6}$$

where:

- $\mathbf{x}_k$ is the state, without economic signification,
- $\mathbf{y}_k$ is the observation, the spline coefficient vector of the smoothing of past raw three hours data,
- Eq. (5.5) is the state equation,
- Eq. (5.6) is the measurement equation,

- $\mathbf{v}_k$ is the process noise, with $\mathbf{v}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{Q})$,
- $\mathbf{n}_k$ is the measurement noise, with $\mathbf{n}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{R})$.

We consider:

- noises may be non-Gaussian;
- distributions of probability of state can be non-symmetric and non-unimodal,
- functions $\mathbf{f}_k$ and $\mathbf{h}_k$ are nonlinear.

For $\mathbf{f}_k$ and $\mathbf{h}_k$ functions, we will use a parametric representation by *Radial Basis Functions Network* (RBFN) because they possess the universal approximation property [Haykin, 1999].

However it should be noted that the random states variables do not carry economic signification any longer.

Trading Model - With Stop Loss

We will derive a parametric Dynamic State Space Model in continuous time for state equation, and in discrete time for measurement equation, where random variables may be scalars, vectors or curves. This model is formalized by:

$$d\mathbf{x}(t) = \mathbf{f}(\mathbf{x}, t)dt + \mathbf{L}(t)\ d\boldsymbol{\beta}_t \tag{5.7}$$

$$\mathbf{y}_k = \mathbf{h}_k(\mathbf{x}(t_k)) + \mathbf{n}_k, \tag{5.8}$$

where:

- $\mathbf{x}(t)$ is a stochastic state vector, without economic signification,
- $\mathbf{y}_k$ is the observation at time t_k, the spline coefficient vector of the smoothing of past raw three hours data,
- Eq. (5.7) is the differential stochastic state equation,
- Eq. (5.8) is the measurement equation,
- $\mathbf{x}_{t_0}$ is a stochastic initial condition satisfying $\mathbb{E}[||\mathbf{x}_{t_0}||^2] < \infty$,
- it is assumed that the drift term $\mathbf{f}(\mathbf{x}, t)$ satisfies sufficient regular conditions to ensure the existence of strong solution to Eq. (5.7), see [Jazwinski, 1970] and [Kloeden and Platen, 1992],
- it is assumed that the function $\mathbf{h}_k(\mathbf{x}(t_k))$ is continuously differentiable with respect to $\mathbf{x}(t)$.
- $\mathbf{n}_k$ is the measurement noise at time t_k, with $\mathbf{n}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{R})$
- $\boldsymbol{\beta}_t$ is a Wiener process, with diffusion matrix $\mathbf{Q}_c(t)$,
- $\mathbf{L}(t)$ is the dispersion matrix, see [Jazwinski, 1970].

5.1.5 How to realize the forecasting

The basic idea underlying these models is to use a *Dynamic State-Space Model* (DSSM) with nonlinear, parametric equations, as *Radial Basis Function Network* (RBFN), and to use the framework of the *Particle filters* (PF) combined with the *Unscented Kalman filters* (UKF) for stopping time forecasting.

The stopping time forecasting derives from a state forecasting in an *Inference procedure*. But before running these models, we must realize a parameter estimation in a *Learning procedure*.

5.1.6 Which tools do we use for stopping time forecasting

The *Kalman filter* (KF) cannot be used for this analysis since the functions are nonlinear and the transition density of the state space is non-symmetric or/and non-unimodal. 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 become 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, et al., 2004], etc. The main advantage of these techniques lies in their great flexibility when treating nonlinear dynamic models with non-Gaussian noises, which cannot be handled through the traditional Kalman filter.

See Chapter [8] for the description of Particle Filters in Discrete time, and Chapter [9] for the description of filters in Continuous-Discrete time.

5.2 Methodology in Discrete Time

The forecasting procedure in discrete time consists in three phases: "*Training*" step to fit models, "*Validation*" step to select the "best" model according to a criterion and we assess prediction error using the "*Test*" step.

We use a "*Bootstrap Particle*" filter (see algorithm in [H.6.1]), combined with a "*Unscented Kalman*" filters (see algorithm in [H.5.1]) for prediction of stopping

time, in a "*Sigma-Point Particle filter*" (see algorithm in [H.7.1]).

In an *inference* procedure, these filters realize a *state estimation* and a *one-step ahead measurement prediction* knowing the parameters of equations. In case of non stationary systems, we start with a good first initial parameter estimation, and with a *joint filtering* ([Nelson , 2000]) we can follow the parameter evolution and have a state estimation in the same step. But when we undergo an important modification of the process dynamic, the adaptive parameter estimation gives poor estimations and results are questionable. To get around this problem, periodically we must realize a new parameter initialization in a batch procedure in two steps: a "*Learning*" procedure for parameter estimation and an "*Inference*" procedure for state estimation, as described in next sections.

5.2.1 Model

Let consider a non stationary, nonlinear Discrete Time Model in a Dynamic State Space representation:

$$\mathbf{x}_k = \mathbf{f}_k(\mathbf{x}_{k-1}) + \mathbf{v}_k \tag{5.9}$$

$$\mathbf{y}_k = \mathbf{h}_k(\mathbf{x}_k) + \mathbf{n}_k, , \tag{5.10}$$

with non stationary, nonlinear RBF functions in State and Measurement equations:

$$\mathbf{f}_k(\mathbf{x}_k) = \sum_{i=1}^{I} \lambda_{ik}^{(f)} \exp\big[-\frac{1}{\sigma_{ik}^{(f)}} (\mathbf{x}_k - \mathbf{c}_{ik}^{(f)})^T (\mathbf{x}_k - \mathbf{c}_{ik}^{(f)}) \big] \tag{5.11}$$

$$\mathbf{h}_k(\mathbf{x}_k) = \sum_{j=1}^{J} \lambda_{jk}^{(h)} \exp\big[-\frac{1}{\sigma_{jk}^{(h)}} (\mathbf{x}_k - \mathbf{c}_{jk}^{(h)})^T (\mathbf{x}_k - \mathbf{c}_{jk}^{(h)}) \big]. \tag{5.12}$$

where

- $\mathbf{x}_k$ is the state, without economic signification,
- $\mathbf{y}_k$ is the observation, the spline coefficient vector of the smoothing of past raw three hours data,
- Eq. (5.9) is the state equation,
- Eq. (5.10) is the measurement equation,
- $\mathbf{v}_k$ is the process noise, with $\mathbf{v}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{Q})$,
- $\mathbf{n}_k$ is the measurement noise, with $\mathbf{n}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{R})$,
- I is the number of neurons in the hidden layer for function $\mathbf{f}_k$,
- $\mathbf{c}_{ik}^{(f)}$ is the centroid vector for neuron i,
- $\lambda_{ik}^{(f)}$ are the weights of the linear output neuron,
- $\sigma_{ik}^{(f)}$ are the variances of clusters,

- J is the number of neurons in the hidden layer for function $\mathbf{h}_k$,
- $\mathbf{c}_{jk}^{(h)}$ is the centroid vector for neuron j,
- $\lambda_{jk}^{(h)}$ are the weights of the linear output neuron,
- $\sigma_{jk}^{(h)}$ are the variances of clusters.

5.2.2 Parameter Estimation

In this section, we introduce the general concepts involved in parameter estimation for Stopping Time forecasting in discrete time. We only give a general background to problems detailed in Appendix [G].

In each Training, Validation, and Test phase, to realize the stopping time prediction, we need a state forecasting carries out an inference step. This inference step comes after a parameter estimation in a learning step.

We must estimate two sets of parameters for non stationary processes, such as:

- estimation of : $\{\widehat{\lambda}_{ik}^{(f)},\ \widehat{\sigma}_{ik}^{(f)},\ \mathbf{c}_{ik}^{(f)}\}$ for function $\mathbf{f}_k(\mathbf{x}_k)$,
- estimation of : $\{\widehat{\lambda}_{jk}^{(h)},\ \widehat{\sigma}_{jk}^{(h)},\ \mathbf{c}_{jk}^{(h)}\}$ for function $\mathbf{h}_k(\mathbf{x}_k)$.

But, for parameter estimation we need two sets of input and output data:

- for parameter estimation of $\mathbf{f}_k(\mathbf{x}_k)$, we need :
  - input data $\{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_{k-1}, \cdots, \mathbf{x}_{N-1}\}$
  - output data $\{\mathbf{x}_2, \mathbf{x}_3, \cdots, \mathbf{x}_k, \cdots, \mathbf{x}_N\}$

- for parameter estimation of $\mathbf{h}_k(\mathbf{x}_k)$, we need:
  - input data $\{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_{k-1}, \cdots, \mathbf{x}_N\}$
  - output data $\{\mathbf{y}_1, \mathbf{y}_2, \cdots, \mathbf{y}_{k-1}, \cdots, \mathbf{y}_N\}$

We know measurement $\{\mathbf{y}_1, \mathbf{y}_2, \cdots, \mathbf{y}_{k-1}, \cdots, \mathbf{y}_N\}$, but we do not know states $\{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_{k-1}, \cdots, \mathbf{x}_N\}$. Thus, we cannot directly realize a parameter estimation.

We could use a dual filtering procedure ([Nelson , 2000]) to estimate, in a recursive procedure, parameters and states. But when the number of parameters and state dimension are high, this procedure does not stabilize, and we must start the procedure with an initial parameter estimation.

5.2.3 Initial Parameter Estimation

To cope with this problem, we realize an initial parameter estimation with EM-algorithm in a "*learning*" procedure, and a state estimation by Kalman filters in an "*inference*" procedure, into separate steps as follow:

- Learning procedure for stationary, linearized model,
  - we consider a stationary, nonlinear system,
  - we realize a linearization of equations,
  - we use the EM-algorithm to estimate the parameters of the linearized model,
- Inference procedure for stationary, linearized model,
  - we use a Kalman smoother for state estimation, using the parameters of the linearized model,
- Learning procedure for stationary, nonlinear model,
  - if we have a small Δt, the state behaviors of nonlinear and linearized models are similar, and we can use the state values coming from the linearized model as input for the nonlinear model, in order to realize its parameter estimation,

Afterwards, we use these initial parameters for the non stationary, nonlinear model, with $\mathbf{f}_k$, and $\mathbf{h}_k$ functions, in a joint filter procedure, that gives a state prediction and an adaptive parameter estimation in the Training, Validation and Test phases.

In next sections, we sketch the learning and inference procedures for initial parameter estimation, these procedures are detailed in Appendix [F], and [G].

Learning phase for stationary, linearized model

From the non stationary, nonlinear system, Eq. (5.9), and (5.10), we derive the stationary, nonlinear system:

$$\mathbf{x}_k = \mathbf{f}(\mathbf{x}_{k-1}) + \mathbf{v}_k \tag{5.13}$$

$$\mathbf{y}_k = \mathbf{h}(\mathbf{x}_k) + \mathbf{n}_k, \tag{5.14}$$

with stationary, nonlinear RBF functions in State and Measurement equations:

$$\mathbf{f}(\mathbf{x}_k) = \sum_{i=1}^{I} \lambda_i^{(f)} \exp\big[-\frac{1}{\sigma_i^{(f)}} (\mathbf{x}_k - \mathbf{c}_i^{(f)})^T (\mathbf{x}_k - \mathbf{c}_i^{(f)}) \big] \tag{5.15}$$

$$\mathbf{h}(\mathbf{x}_k) = \sum_{j=1}^{J} \lambda_j^{(h)} \exp\big[-\frac{1}{\sigma_j^{(h)}} (\mathbf{x}_k - \mathbf{c}_j^{(h)})^T (\mathbf{x}_k - \mathbf{c}_j^{(h)}) \big]. \tag{5.16}$$

We consider a stationary, linearized version of the system:

$$\mathbf{x}_k = \mathbf{A}\mathbf{x}_{k-1} + \mathbf{v}_k \tag{5.17}$$

$$\mathbf{y}_k = \mathbf{C}\mathbf{x}_k + \mathbf{n}_k, \tag{5.18}$$

where

- $\mathbf{x}_k$ is the state,
- $\mathbf{y}_k$ is the observation,
- Eq. (5.17) is the state equation,
- Eq. (5.18) is the measurement equation,
- $\mathbf{v}_k$ is the process noise, with $\mathbf{v}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{Q})$,
- $\mathbf{n}_k$ is the measurement noise, with $\mathbf{n}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{R})$.

We estimate the parameters $\{\mathbf{A}, \mathbf{C}, \mathbf{R}, \mathbf{Q}, \mathbf{x}_0, \mathbf{P}_0\}$ by EM-algorithm in a learning phase (see description in Appendix [F]).

Inference phase for stationary, linearized model

From parameters estimated at the precedent step, we estimate the states:

$$\{\mathbf{x}_{0|N}, \mathbf{x}_{1|N}, \cdots, \mathbf{x}_{N-1|N}, \mathbf{x}_{N|N}\}$$

by the Kalman smoother in an inference phase applied to system given by equations (5.17), and (5.18) (see description in Appendix [G]).

Learning phase for stationary, nonlinear model

We use states given by the precedent step, and observations for a parameter estimation of RBF functions (Eq. (5.15), and (5.16)) in a learning phase, that gives the parameters:

$$\{\widehat{\lambda}_i^{(f)}\ ,\ \widehat{\sigma}_i^{(f)}\ ,\ \mathbf{c}_i^{(f)}\ ,\ \widehat{\lambda}_j^{(h)}\ ,\ \widehat{\sigma}_j^{(h)}\ ,\ \mathbf{c}_j^{(h)}\}$$

5.2.4 Training, Validation and Test Phases

After parameter initialization, we can start Training, Validation, and Test phases using Particle and Kalman filters.

We consider a non stationary, nonlinear process and we use the former parameters to realize the inference procedure for state estimation and curve prediction, such as:

$$\mathbf{x}_k = \mathbf{f}_k(\mathbf{x}_{k-1}) + \mathbf{v}_k \tag{5.19}$$

$$\mathbf{y}_k = \mathbf{h}_k(\mathbf{x}_k) + \mathbf{n}_k, \tag{5.20}$$

with non stationary, nonlinear RBF functions:

$$\mathbf{f}_k(\mathbf{x}_k) = \sum_{i=1}^{I} \lambda_{ik}^{(f)} \exp\big[-\frac{1}{\sigma_{ik}^{(f)}}(\mathbf{x}_k - \mathbf{c}_{ik}^{(f)})^T(\mathbf{x}_k - \mathbf{c}_{ik}^{(f)})\big] \tag{5.21}$$

$$\mathbf{h}_k(\mathbf{x}_k) = \sum_{j=1}^{J} \lambda_{jk}^{(h)} \exp\big[-\frac{1}{\sigma_{jk}^{(h)}}(\mathbf{x}_k - \mathbf{c}_{jk}^{(h)})^T(\mathbf{x}_k - \mathbf{c}_{jk}^{(h)})\big]. \tag{5.22}$$

Every day we want to realize a curve price forecasting for an interval of three hours in an inference procedure using "Sigma-Point Particle filter" (see algorithm in [H.7.1]). But the system is non stationary, and parameters depend on time and must be evaluate again in a learning procedure associated to an inference procedure.

Starting with a first good initial parameter estimation, we realize both jobs in a joint procedure, but when we have important dynamic modifications this joint procedure does not manage this parameter evolution and we have to realize a new parameter estimation in a batch procedure into two separate steps, as for initial parameter estimation.

5.2.5 Documentation

These procedures are detailed in:
Appendix: [F], State and Parameter Estimations of a Linear Model by EM algorithm,
Appendix: [G], State and Parameter Estimations of a Non Linear Model by EM algorithm.

Algorithms are given in:
Appendix: [H.3.1], Kalman Filter (KF),
Appendix: [H.4.1], Extended Kalman Filter (EKF),
Appendix: [H.5.1], Unscented Kalman Filter (UKF),
Appendix: [H.7.1], Sigma-Point Particle filter (SPPF),
Appendix: [H.10.1], State and Parameter Estimations for Linear Model by EM algorithm,
Appendix: [H.11.1], State and Parameter Estimations for Non Linear Model by EM algorithm.

The Kalman filter algorithms are adapted from [Grewal and Andrews , 2001]. The Dual and Joint filters are detailed in [Nelson , 2000].

5.3 Methodology in Continuous-Discrete Time

The forecasting procedure in continuous-discrete time consists in three phases: "*Training*" step to fit models, "*Validation*" step to select the "best" model according to a criterion and we assess prediction error using the "*Test*" step. We use a "*Continuous-Discrete Sequential Importance Resampling*" filter (see Section. 9.3.7), combined with a "*Continuous-Discrete Unscented Kalman*" filter (see Section 9.3) for prediction of stopping time.

In an *inference* procedure, these filters realize a *state estimation* and a *one-step ahead measurement prediction* knowing the parameters of equations. In case of non stationary systems, we start with a good first parameter estimation, and with a *joint filtering* we can follow the parameter evolution and have a state estimation in the same step. But when we undergo an important modification of the process dynamic, the adaptive parameter estimation gives poor estimations and results are questionable. To get around this problem, periodically we must realize a new parameter initialization in a batch procedure in two steps: a "*Learning*" procedure for parameter estimation and an "*Inference*" procedure for state estimation, as described in next section.

5.3.1 Model

Let consider a non stationary, nonlinear Continuous-Discrete Time Model in a Dynamic State space representation:

$$d\mathbf{x}_t = \mathbf{f}(\mathbf{x}_t, t)dt + \mathbf{L}_t \, d\boldsymbol{\beta}_t \tag{5.23}$$

$$\mathbf{y}_k = \mathbf{h}_k(\mathbf{x}_{t_k}) + \mathbf{n}_k, \tag{5.24}$$

for $t \in \{0, T\}$, and $k \in [1, 2, \cdots, N]$ such as $t_N \leq T$, with non stationary, nonlinear RBF functions in State and Measurement equations:

$$\mathbf{f}(\mathbf{x}_t, t) = \sum_{i=1}^{I} \lambda_{it}^{(f)} \exp\big[-\frac{1}{\sigma_{it}^{(f)}} (\mathbf{x}_t - \mathbf{c}_{it}^{(f)})^T (\mathbf{x}_t - \mathbf{c}_{it}^{(f)}) \big] \tag{5.25}$$

$$\mathbf{h}_k(\mathbf{x}_{t_k}) = \sum_{j=1}^{J} \lambda_{jk}^{(h)} \exp\big[-\frac{1}{\sigma_{jk}^{(h)}} (\mathbf{x}_{t_k} - \mathbf{c}_{jk}^{(h)})^T (\mathbf{x}_{t_k} - \mathbf{c}_{jk}^{(h)}) \big]. \tag{5.26}$$

where:

- $\mathbf{x}(t)$ is a stochastic state vector, without economic signification,
- $\mathbf{y}_k$ is the observation at time t_k, the spline coefficient vector of the smoothing of past raw three hours data,
- Eq. (5.23) is the differential stochastic state equation,
- Eq. (5.24) is the measurement equation,
- $\mathbf{x}_{t_0}$ is a stochastic initial condition satisfying $\mathbb{E}[||\mathbf{x}_{t_0}||^2] < \infty$,
- it is assumed that the drift term $\mathbf{f}(\mathbf{x}, t)$ satisfies sufficient regular conditions to ensure the existence of strong solution to Eq. (5.7), see [Jazwinski, 1970] and [Kloeden and Platen, 1992],
- it is assumed that the function $\mathbf{h}_k(\mathbf{x}(t_k))$ is continuously differentiable with respect to $\mathbf{x}(t)$.
- $\mathbf{n}_k$ is the measurement noise at time t_k, with $\mathbf{n}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{R})$
- $\boldsymbol{\beta}_t$ is a Wiener process, with diffusion matrix $\mathbf{Q}_c(t)$,

- $\mathbf{L}(t)$ is the dispersion matrix, see [Jazwinski, 1970],
- I is the number of neurons in the hidden layer for function $\mathbf{f}_k$,
- $\mathbf{c}_{ik}^{(f)}$ is the centroid vector for neuron i,
- $\lambda_{ik}^{(f)}$ are the weights of the linear output neuron,
- $\sigma_{ik}^{(f)}$ are the variances of clusters,
- J is the number of neurons in the hidden layer for function $\mathbf{h}_k$,
- $\mathbf{c}_{jk}^{(h)}$ is the centroid vector for neuron j,
- $\lambda_{jk}^{(h)}$ are the weights of the linear output neuron,
- $\sigma_{jk}^{(h)}$ are the variances of clusters.

5.3.2 Parameter Estimation

In this section, we only give a general background to the general concepts involved in parameter estimation for Stopping Time forecasting in continuous-discrete time.

In each Training, Validation, and Test phase, to realize the stopping time prediction, we need a state forecasting carries out an inference step. This inference step comes after a parameter estimation in a learning step.

The procedure is similar to the procedure in Discrete Time, but the complexity is higher due to a Stochastic Differential equation for state,

- we must use a quasi-likelihood instead of a real likelihood [Ait-Sahalia, 2002],
- we cannot use EM-algorithm with quasi-likelihood, we must use a direct optimization with Gauss-Newton procedure [Ait-Sahalia, 2002], and [Nielsen et al. , 2000],
- we must realize a discretization of Continuous Stochastic Differential Equation
  [Ait-Sahalia and Myland, 2003],
- we use Kalman-Bucy filters for initial parameter estimation [Kalman and Bucy, 1961], and [Grewal and Andrews , 2001],
- we use strong solutions for the model [Kloeden and Platen, 1992].

We must estimate two sets of parameters for non stationary processes, such as:

- estimation of : $\{\widehat{\lambda}_{it}^{(f)},\ \widehat{\sigma}_{it}^{(f)},\ \mathbf{c}_{it}^{(f)}\}$ for function $\mathbf{f}(\mathbf{x}_t, t)$,
- estimation of : $\{\widehat{\lambda}_{jk}^{(h)},\ \widehat{\sigma}_{jk}^{(h)},\ \mathbf{c}_{jk}^{(h)}\}$ for function $\mathbf{h}_k(\mathbf{x}_{t_k})$.

But, for parameter estimation we need two sets of input and output data:

- for parameter estimation of $\mathbf{f}(\mathbf{x}_t, t)$, we need : $\{\mathbf{x}_t\}$ for $t \in [0, T]$

- for parameter estimation of $\mathbf{h}_k(\mathbf{x}_{t_k})$, we need:
  - input data $\{\mathbf{x}_{t_1}, \mathbf{x}_{t_2}, \cdots, \mathbf{x}_{t_k}, \cdots, \mathbf{x}_{t_N}\}$
  - output data $\{\mathbf{y}_{t_1}, \mathbf{y}_{t_2}, \cdots, \mathbf{y}_{t_k}, \cdots, \mathbf{y}_{t_N}\}$,

We know measurement $\{\mathbf{y}_{t_1}, \mathbf{y}_{t_2}, \cdots, \mathbf{y}_{t_k}, \cdots, \mathbf{y}_{t_N}\}$, but we do not know states $\{\mathbf{x}_{t_1}, \mathbf{x}_{t_2}, \cdots, \mathbf{x}_{t_k}, \cdots, \mathbf{x}_{t_N}\}$. Thus, we cannot directly realize a parameter estimation.

We could use a dual filtering procedure ([Nelson , 2000]) to estimate, in a recursive procedure, parameters and states. But when the number of parameters and the state dimension are high, this procedure does not stabilize, and we must start the procedure with an initial parameter estimation.

5.3.3 Initial Parameter Estimation

In case of stochastic differential equations, we cannot use the EM algorithm (see [Ait-Sahalia, 2002], and [Nielsen et al. , 2000]). To cope with this problem, we realize an initial parameter estimation with a Gauss-Newton optimization in a "*learning*" procedure (see [Kim and Nelson , 1999]), and a state estimation by Kalman-Bucy filters in an "*inference*" procedure, into separate steps.

We sketch the methodology for initial parameter estimation.

- We transform the stochastic differential equation with Wiener process into a stochastic differential equation with white noise [Jazwinski, 1970],

- Learning procedure for stationary, linearized model,
  - we consider a stationary, nonlinear system,
  - we realize a linearization of equations,
  - we use the Gauss-Newton algorithm to estimate the parameters of the linearized model [Bjorck , 1996], and [Fletcher , 1987],

- Inference procedure for stationary, linearized model,
  - we use a Kalman-Bucy smoother for state estimation, using the parameters of the linearized model,

- Learning procedure for stationary, nonlinear model,
  - if we have a small Δt, the state behaviors of nonlinear and linearized models are similar, and we can use the state values coming from the linearized model as input for the nonlinear model, in order to realize its parameter estimation,

5.3.4 Training, Validation and Test Phases

After initial parameter initialization, we can start Training, Validation, and Test phases using Stochastic Particle filters and Kalman-Bucy filters in continuous time.

We consider a non stationary, nonlinear process and we use the former parameters to realize the inference procedure for state estimation and curve prediction.

$$d\mathbf{x}_t = \mathbf{f}(\mathbf{x}_t, t)dt + \mathbf{L}_t \, d\boldsymbol{\beta}_t \tag{5.27}$$

$$\mathbf{y}_k = \mathbf{h}_k(\mathbf{x}_{t_k}) + \mathbf{n}_k, \tag{5.28}$$

for $t \in \{0, T\}$, and $k \in [1, 2, \cdots, N]$ such as $t_N \leq T$, with non stationary,nonlinear RBF functions in State and Measurement equations:

$$\mathbf{f}(\mathbf{x}_t, t) = \sum_{i=1}^{I} \lambda_{it}^{(f)} \exp\big[- \frac{1}{\sigma_{it}^{(f)}} (\mathbf{x}_t - \mathbf{c}_{it}^{(f)})^T (\mathbf{x}_t - \mathbf{c}_{it}^{(f)}) \big] \tag{5.29}$$

$$\mathbf{h}_k(\mathbf{x}_{t_k}) = \sum_{j=1}^{J} \lambda_{jk}^{(h)} \exp\big[- \frac{1}{\sigma_{jk}^{(h)}} (\mathbf{x}_{t_k} - \mathbf{c}_{jk}^{(h)})^T (\mathbf{x}_{t_k} - \mathbf{c}_{jk}^{(h)}) \big]. \tag{5.30}$$

Every day we want to realize a curve price forecasting for an interval of three hours in an inference procedure using a "Continuous-Discrete Unscented Kalman Filter" (see algorithm in [9.3.3]), and a "Continuous-Discrete Sampling Importance Resampling Filter" (see algorithm in [9.3.6]). But parameters depend on time and must be evaluate again in a learning procedure associated to an inference procedure.

Starting with a first good initial parameter estimation, we realize both jobs in a joint procedure, but when we have important dynamic modifications this joint procedure does not manage this parameter evolution and we have to realize a new parameter estimation in a batch procedure into two separate steps, as for initial parameter estimation.

5.3.5 Documentation

The procedures are adapted from : [Deck and Theting., 2004], [Jimenez and Ozaki , 2005], [Nielsen et al. , 2000], [Maslowski and Pospisil , 2007], [Bishwal , 2007], [Taylan and Weber , 2007], [Ramsay et al. , 2007], and [Sparreman , 2006].

The procedures are detailed in:
Appendix: [D], Parameter Estimation of a Linear State Space Model by Gauss-Newton procedure,
Appendix: [E], Parameter Estimation of a non Linear State Space Model by Gauss-Newton procedure.

Algorithms are given in:
Appendix: [H.8.1], Parameter Estimation for a Linear State Space Model by Gauss-Newton
Appendix: [H.9.1], Parameter Estimation for a non Linear State Space Model by Gauss-Newton

Algorithms are sketched in Section [9.3] for:
- Continuous-discrete Kalman filter,
- Continuous-discrete Unscented Kalman filter,
- Continuous-discrete Sequential Importance Resampling filter,

The numerical methods for stochastic differential equations are sketched in Appendix: [C.1], they come from [Kloeden and Platen, 1992], and [Kloeden et al., 1992].

The Kalman filter algorithms are adapted from [Grewal and Andrews , 2001].
The Dual and Joint filters are detailed in [Nelson , 2000].
The Learning and Inference procedures for State-Space models in stochastic representation are adapted from [Murphy , 2002].

Part II

Financial Introduction

Part Two gives an introduction to the Financial Market specificities, and their behaviors.

6

Analysis of High Frequency Financial Time Series

This chapter gives an introduction to Financial Markets, "tick-by-tick" high-frequency financial time series, and theoretical financial models.

This chapter gives an introduction to Financial Markets, "tick-by-tick" high-frequency financial time series, and theoretical financial models.

Ours models will use high-frequency time series of raw data and we must take account of their specificities.

For example, before working with such time series, we must filter them to eliminate bad quotes, remove the seasonality and transform the universal time into a business time to eliminate market inactive periods such as nights, weekends and holidays.

6.1 Introduction

In former times it was usual to receive the latest data from stock exchanges with a delay of a few minutes. But now, in modern times it is possible to get the data immediately. Every person from every country can have every information about stock prices after a second. We live in a world with *High Frequency Finance*, thus, we will work with high frequency data.

Securities may be treated on a twenty-hours Market, such as the Foreign Exchange (FX) Market, seven days a week, or on a non-continuous Market that opens at 9:30 AM and closes at 4:00 PM, five days a week such as the New York Stock Exchange and Nasdaq.
Some securities are very liquid when they have a sufficient number of buyers and venders but others may be considered as non-liquid.

6.2 Data Characteristics

High Frequency financial time series contain tens of thousands of transactions in a single day time stamped to the nearest second. The analysis of these data are complicated by irregular and random sampling, diurnal patterns, price discreetness, and complex dependence. Some transactions occur only seconds apart while others may be five or ten minutes apart or more depending of the liquidity of the security. Thus, a choice must be made regarding the time intervals over which to analyze the data. If fixed intervals are chosen then we have to define an interpolation rule when no transaction occurs exactly at the end of the interval. If stochastic intervals are used then the spacing of the data will likely need to be taken into account.

Financial data is discrete and institutional rules can restrict prices to fall on a restricted set of multiples of ticks. These small price changes are imposed by discouraging price changes from one transaction to the next. The result is that price changes often fall on a very small number of possible outcomes.

Moreover, *Intraday* financial data contain also some periodic patterns. *Volatility* is systematically higher near the open and generally just prior to the close. The time between trades tend to be shorter near the open and just prior to the close [Engle and Russel, 1998].

We can also note that, High Frequency Returns data display strong dependence due to prices discretization and spreads between "*Bid*" and "*Ask*". Dependencees in price changes are also due to traders breaking orders into a sequence of smaller orders in hopes of transacting at an overall better price. High frequency data exhibit also volatility clustering, large price changes tend to follow large price changes and vice-versa.

Assets are traded in centralized markets. These markets might be order driven where a computer uses algorithms to match market participants or they might be open outcry where there is a centralized trading floor that market participants must trade through [Engle and Russel, 1998].

The data sets used by researchers contain detailed information about the transactions and quotes for each asset traded on the market. But the time of a transaction down to the nearest second, and "Bid" and "Ask" quotes are also available along with a time stamp for when the quotes became active. Quotes from the NYSE are valid for a fixed (typically small) quantity or depth, and quotes for EURONEXT are derived from the limit order book and represent the best "Ask" and "Bid" prices in the book.

6.3 Traders

In theory all new information would be immediately disseminated and interpreted by all market participants and prices would immediately adjust to a new equilibrium value determined by the agent preferences and the content of the information. But this is not the reality, and in practice not all relevant information is known by all participants at the same time, and this information is not processed at the same speed by all participants.

Moreover, some agents are referred to as privately informed, and agents without superior information are referred to as noise or liquidity traders and are assumed to be indistinguishable from the informed agents. These informed traders will make profitable transactions at the expense of the uninformed. Market makers can learn about private information by observing the actions of traders. Informed traders only transact when they have private information and would like to trade larger quantities to capitalize on their information before it becomes public. That means that characteristics of transactions carry information [Engle and Russel, 1998].

Prices adjust more quickly to reflect private information when the proportion of uninformed traders is higher. Volume is higher, transaction rates are higher when the proportion of uninformed traders is higher.

6.4 Efficient Market Hypothesis (EMH)

The *Efficient Market Hypothesis* is a backbreaker for forecasters. In its crudest form it says that financial series are unforeseeable.

If returns were foreseeable, many investors would use them to generate unlimited profits. The behavior of market participants induces returns that obey the EMH, otherwise there would exist a 'money-machine' producing unlimited wealth, which cannot occur in a stable economy, and that might appear to be the end of the story.

However, a market that is quite efficient ([Fama., 1970]) over a daily horizon might not have completely unpredictable returns from trade to trade or from minute to minute, ([Hillmer and Yu, 1979], ([Epps, 1979]), and ([Patell and Wolson , 1984]). Investors need time to absorb and act on new information, and short-horizon returns are predictable from past order flows, ([Chordia et al., 2000]).

6.5 Market heterogeneities

Financial markets are complex systems resulting from the action of millions of agents from many different countries. Although it is logical to assume that all market participants are either attempting to make money through investing or trying to reduce risk by hedging, the time frame over which they operate differs immensely, ranging from seconds to months. These agents can be grouped into a few investment styles. Each group has different motivations to buy and sell financial assets, resulting in different trading time frames [Lynch and Zumbach, 2001].

A classification of the market participants, in order of increasing characteristic time horizons, could be:

1. *Intraday traders* and *market makers* are looking to buy and sell an asset to realize a quick profit (or minimize a loss) over very short time horizons ranging from seconds to hours.
2. *Hedge funds* often trade over a few days, or on a "close to close" basis.
3. *Portfolio managers* follow trading strategies, such as index tracking. They adjust their portfolio to follow the fundamental information released by the companies, or the fluctuating prices of each asset and their corresponding weights in the benchmark index on a weekly to monthly basis, with little attention to intra-day prices.
4. *Central banks* take long term macro-economic views on foreign exchange and money market rates.
5. *Pension funds* have investments to hold for decades. This very long time horizon, as well as legal constraints, allows pension funds to invest part of their portfolio in real estate.

In the past, the actions of the intraday traders and hedge funds were insignificant to the pension funds, but nowadays, pension funds trade also hedge funds or even funds of edge funds in order to maximize their profits.

6.6 Forecasting

There are three "*schools of thought*" in terms of the ability to profit from the Financial market.

1. No investor can achieve above average trading advantages based on the historical and present information. The major theory includes the Random Walk Hypothesis and Efficient Market Hypothesis [Peters , 1991].

2. Fundamental analysis looks in depth at the financial condition of each country and studies the effects of supply and demand on each currency.

3. Technical analysis assumes that the assets move in trends and these trends can be captured and used for forecasting. It uses tools such as charting patterns, technical indicators and specialized techniques like Gann lines, Elliot waves and Fibonacci series [Plummer , 1991].

We can find in the literature many publications about the analysis of Financial data and the documentation of their behavior (see for Ex. [Andersen et al., 2005], [Engle, 1982], [Engle and Russel, 1998], [O'hara , 1995]).

Financial time series are affected by many highly correlated economic, political and even psychological factors. The interaction of these factors is in a very complex fashion. Therefore, to forecast the changes of financial assets is generally very difficult. Researchers and practitioners have been striving for an explanation of the movement of financial assets, and various kinds of forecasting methods have been developed by many researchers and experts. We can note than technical and fundamental analyses are the basics of major forecasting methodologies which are in popular use in financial forecasting.

Every economic time series has its own trend, cycle, season, and irregularity. Thus to identify, model, extrapolate and recombine these patterns and to forecast is the major challenge. Traditionally, statistical models such as Box-Jenkins models [Box and Jenkins, 1970] dominate the time series forecasting. But traditional statistical techniques for forecasting have reached their limitation in applications with nonlinearities in the data set such as stock indices [Refenes et al., 1994], [White , 1989].

To maximize profits, different forecasting techniques are used by traders. They use a variety of techniques to obtain multiple signals. Classical time series analysis, based on the theory of stationary stochastic processes ([Harvey, 1989]) does not perform satisfactorily on economic time series. Economic data are not simple autoregressive-integrated-moving-average(ARIMA) processes; they are not described by simple linear structural models, nor they are simple white noise or even random walks.

Hence the major challenge ahead is the development of new methods, or the modification or integration of existing ones, that are capable of accurately forecasting series whose patterns or relationships change over time.

To build a forecasting model, historical data are divided into three sets for training, validation and testing. A model is considered good if the error of

out-of-sample testing is the lowest compared with the other models. If the trained model is the best one for validation and also the best one for testing, one can assume that it is a good model for future forecasting.

The application of forecasting method includes two basic steps: analyzing data series and selecting the forecasting method that best fits the data series.

6.7 Errors in Financial Data

High frequency financial time series are inhomogeneous because the ticks happen at random time-points. Moreover, before using such raw data for modeling and forecasting we must check them and remove some errors. If we used such raw data with errors the results of modeling and forecasting would be very questionable.

We consider a data error is present if a piece of quoted data does not conform to the real situation of the market, in other words if it is neither a correctly reported transaction price nor a possible transaction price at the reported time.

There are many causes for data errors.

- human errors: errors directly caused by human data contributors:
  - unintentional errors, e. g. typing errors;
  - intentional errors, e. g. dummy quotes produced just for technical testing;
- system errors: errors caused by computer systems, their interactions and their failures.

System errors are also human errors because human operators have the ultimate responsibility for the correct operation of computer systems. However, the distance between the data error and the responsible person is much larger for system errors.

In many cases, it is impossible to find the exact reason for the error even if the quote is very aberrant, but we have to identify such outliers, whatever the reason. Sometimes the cause of the error can be guessed from the particular behavior of the bad quotes.

We have to deal with special errors of different types [Muller , 2001]:

1. *Decimal errors*:
   Failure to change a "big" decimal digit of the quote.
   Example: a bid price of 4.3498 is followed by a true quote 4.3505, but the published, bad quote is 4.3405. This error is most damaging if the quoting

software is using a cache memory somewhere. The wrong decimal digit may stay in the cache and cause a long series of bad quotes.

2. *"Test" quotes*:
   Some data contributors sometimes send test quotes to the system, usually at times when the market is not liquid. These test quotes can cause a lot of damage because they may look plausible.

   - "Early morning test":
     A contributor sends a bad quote very early in the morning, in order to test whether the connection to the data distributor (e. g. Reuters) is operational. If the market is inactive overnight, no trader would take this test quote seriously.

   - Some contributors test the performance and the time delay of their data connection by sending a long series of linearly increasing quotes at inactive times such as overnight or during a weekend. This is hard to detect because quote-to-quote changes look plausible. Only the monotonic behavior in the long run can be used to identify the fake nature of this data.

3. *Repeated quotes*:
   Some contributors let their computers repeat the last quote in more or less regular time intervals. This is harmless if it happens in a moderate way.

4. *Quote copying*:
   Some contributors employ computers to copy and re-send the quotes of other contributors, just to show a strong presence on the data feed. Thus they decrease the data quality, but there is no reason for a filter to remove copied quotes that are on a correct level.

5. *Scaling problem*:
   Quoting conventions may differ or be officially redefined in some markets.

6.7.1 Examples of "errors"

These examples are not exceptional; very often we have many transactions at the same time-point, even transactions with different prices.
Figure [6.1] shows trades recorded at 16:04:44hr with very different prices. The reality of such prices can be suspicious. It could be possible transaction prices but not at the good reported time. But in this example, it was at the close time of the market, we could estimate a rush of traders to close their position.

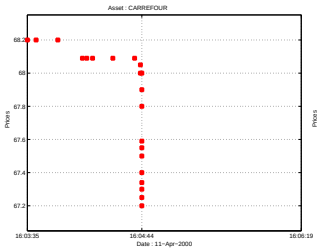

Figure 6.1:

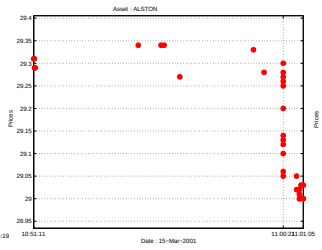

Figure 6.2:

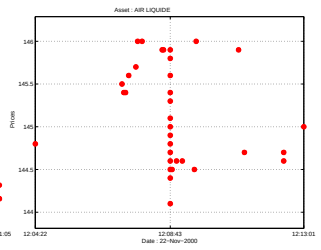

Figure 6.3:

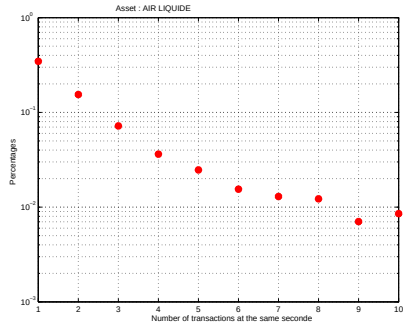

Figure 6.4: Number of transactions recorded at the same time-point

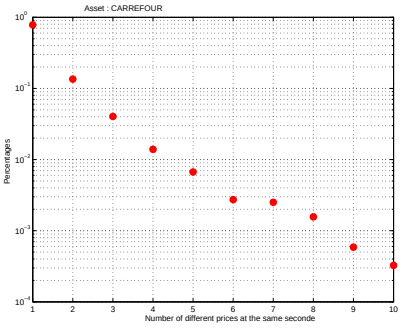

Figure 6.5: Number of different prices at the same time-point

Figure 6.2 shows trades recorded at 11:00:21hr with a drop of prices. When this happens at the training phase we can use past, present and future prices to improve filtering. But in a real time application, we only use past prices to check present prices. In this case, such trades could be questionable.

Figure 6.3 shows trades recorded at 12:08:43hr with strange transaction prices. There are some difficulties to check these transactions without having a look at news. But this could be impossible in a real-time phase.

Figure 6.4 shows that for more than 65 % of time-points we have multi-transactions recorded at the same second.

Figure 6.5 shows that for more than 25 % of transactions we have several prices at the same time-point.

6.8 Data Filtering

Filtering of financial quotes must detect and eliminate bad quotes in order to reduce their damaging consequences in applications.

Filtering of high-frequency time series data is a demanding, often underestimated task. It is complicated because of:

- the variety of possible errors and their causes,
- the variety of statistical properties of the filtered variables,
- the variety of data sources and contributors of different reliability,
- the irregularity of time interval,
- the complexity and variety of the quoted information,
  - transaction prices,
  - indicative prices,
  - FX forward premia,
  - interest rates,
  - derivative markets,
  - transaction volumes,
  - bid/ask quotes,
- the necessity of real-time filtering.

There are different possible approaches to filtering

- plausibility: we do not know the real cause of data errors. We judge the validity of a quote according to its plausibility, given the statistical properties of the series,
- we need a whole neighborhood of quotes for judging the credibility of a quote, by filtering window,
- the filter must be adaptive to estimate non stationary statistical properties,
- the filter needs a high execution speed. The algorithm must be recursive, only a minimal number of computations concerning the new quote is added,
- the filter has two modes: real-time in forecasting and historical in training and validation phases.

6.8.1 Filter structure

The filtering algorithms designed by Olsen Group [Muller , 2001], and [Dacorogna et al., 1993] use a hierarchical scheme of sub-algorithms. Their filter is univariate, it treats only one financial instrument at a time:

1. univariate filter: The complete filtering of one series,

2. full-quote filtering window : A sequence of recent full quotes, some of them possibly corrected according to a general filtering hypothesis,
3. scalar filtering window : A sequence of recent scalar quotes whose credibilities are still in the process of being modified.

6.8.2 Filtering of single scalar quotes

The level filter computes a first credibility of the value of the filtered variable.

The level filter first put the filtered variable value x into the perspective of its own statistical mean and standard deviation, by

$$\widehat{x} = \frac{x - \overline{x}}{MSD[\Delta\theta_r, 2; x]} \tag{6.1}$$

$$= \frac{x - \overline{x}}{\sqrt{EMA[\Delta\theta_r; (x - \overline{x})^2]}}, \tag{6.2}$$

where the mean value of $\overline{x}$ is also a moving average:

$$\overline{x} = EMA[\Delta\theta_r; x] \tag{6.3}$$

The variable $\Delta\theta_r$ denotes the configuration range of the kernel of the moving averages and should cover the time frame of the mean reversion of the filtered variable. For all the moving averages, a simple exponentially weighted moving average (EMA) is used. The business times θ are described in Section [6.11], and the operators EMA in Section [6.12].

A small $|\widehat{x}|$ value deserves high trust; an extreme $|\widehat{x}|$ value indicates an outlier with low credibility and negative trust capital.

6.8.3 Pair filtering

The time difference between two quotes plays a role, so the time scale on which it is measured has to be specified. The criterion is adaptive to the statistical expected volatility estimate.

The comparaison of the origins of two quotes provides information about contributors. The comparaison of quote origins has to be seen in the perpective of the observed diversity of quote origins.

The expected volatility is given by

$$d = EMA[\Delta\theta_r; \frac{c_d}{\Delta\theta}]. \tag{6.4}$$

An annualized squared micro-volatility is defined as a variance:

$$\nu = EMA\big[\Delta\theta_r; \frac{(\Delta x)^2}{\Delta\theta + \Delta\theta_0}\big]. \tag{6.5}$$

Pair filtering results add some credibility to the two quotes only if these are independent. Two identical quotes from the same contributor do not add a lot of confidence to the quote level. We estimate the diversity D by:

$$D = EMA[\text{tick-time}, r; I_{i,i-1}], \tag{6.6}$$

where $I_{i,i-1}$ is an independence measure with values 0 for identical origins and 1 for clearly different origins. The range r is the center of gravity of the kernel.

Time plays a role in the adaptive element of the level filter. Variable changes are tolerated more easily when separed by a large time interval between the time stamps.

The chose of the time scale is important. Therefore we cannot working using only physical time, but the following solutions are possibles:

- a very simple business time with two states: active (working days), and inactive (weekends and holidays),
- an adaptively weighted mean of three simple, generic business time scales θ, based on typical volatility patterns of three main markets: Asia, Europe, and North America.

In the second solution, these θ times are simply :

$$\frac{d\theta_k}{dt} = \begin{cases} 3.4 & \text{if } t_{\text{start},k} \leq t_d < t_{\text{end},k} \text{ on a working day,} \\ 0.01, & \text{otherwise (inactive times, weekends, holidays)} \end{cases} \tag{6.7}$$

where t_d is the daytime in Greenwich Mean time (GMT) and the generic start and end times of the working-daily activity periods are:

Market	number k	time t_{start}	time t_{end}
East Asia	1	21:00	7:00
Europe	2	6:00	16:00
North America	3	11:00	21:00

Once the three time scales θ_k are defined by integration, their adaptively weighed mean is constructed and used as the time scale θ for filtering.

$$\theta = \sum_{\text{all } k} w_k \, \theta_k \tag{6.8}$$

where

$$w_k = \frac{1}{F_k \ \sum_{\text{all } k' \neq k} \frac{1}{F_{k'}}} \tag{6.9}$$

$$F_k = EMA\big[\delta\theta_r; (\sigma_k - EMA[\delta\theta_r; \sigma_k])^2\big] \tag{6.10}$$

$$= MVar[\Delta\theta_r, 2; \sigma_k] \tag{6.11}$$

$$\sigma_k = \sqrt{EMA\big[\Delta\theta_{\text{smooth}}; \frac{(\delta x)^2}{\delta\theta_k + \delta\theta_0}\big]} \tag{6.12}$$

6.9 Deseasonalization

Financial high frequency data contain very strong intra-day and intra-week seasonalities due to the daily and weekly pattern of human activities. Olsen Group has demonstrated [Dacorogna et al., 1993] that physical time is no longer suitable with high frequency financial data. Instead they employ a business time scale related to business activity, which would only count business hours omitting inactive periods of time, such as holidays, and week-ends. In this approach one expands periods of high volatility (e.g; overlapping of markets) whereas periods of low activity (e.g. lunch time) are contracted. This is the main idea of the so-called θ-time which has been designed to model daily and weekly fluctuations of activity.

The seasonalities are filtered out by doing the computations in the proper business time scale. The aim of deseasonalization is to get a precise understanding of the data generating process and to constitute an appropriate mathematical/econometric motivated model to map and hopefully forecast the price, return and/or volatility process.

When working with daily data we simply omit the weekends and major holidays from computations, which has been done by many researchers in their econometric and financial analysis.

With intraday data, we also have to expand periods of high volatility (overlapping of markets) whereas periods of low activity (lunch time) are contracted.

This scale is a continuous-time generalization of the familiar daily business time scale which contains five days per week. A continuous business time scale θ allows us to map a physical time interval dt to a business time interval $d\theta$, where $d\theta/dt$ is proportional to the expected market activity.

The seasonality activity pattern of high frequency data is measured on a moving sample, and the dynamic time scale is constructed by integration the activity. The time scale is normalized such that, on average, a time interval of dt in physical time scale is equal to $d\theta$ in business time.

To obtain a really good forecasting model, one has to deseasonalize the obtained time series in the best way possible under the aspect that the deseasonalization has to be invertible, so after the forecasting in its deseasonalization form we are able to retransfer the seasonality into the time series [Breymann et al., 2000]

6.10 Scaling Law

In many empirical studies scaling law are found in different time series [Muller , 2001], and [Dacorogna et al., 1993] assumed the following scaling law model for the volatility process:

$$\left(\mathbb{E}[|r|^p]\right)^{\frac{1}{p}} = c(p)\Delta t^{D(p)}, \tag{6.13}$$

where

- $r = \Delta x$, is the return,
- x denotes the process of logarithmic middle bid and ask prices,
- $c(p)$ and $D(p)$ are functions relying on p, with $p > 0$. It is usual to set $p = 1$ (absolute returns), or $p = 2$ (quadratic returns); by assuming a zero mean return this directly corresponds to the volatility,
- D comes from a scaling law model for the volatility process. It is a drift operator similar to the Hurst exponent in the fractional Brownian motion context.

6.11 Business Time

The simplest form of new time scale is to omit weekends and holidays from our physical time. We will call it the standard business time. This time scale is widely used in the analysis of time series with daily observations. Thereby the idea is to revise for days with no trading activity. A consequence of this time

transformation is that a year now has only between 250 and 252 trading days.

The introduction of the business time is not as easy as it seems to be, because we need the exact start and end time point of weekends and holidays, if we consider a whole market segment. This leads to problems because we have to consider the temporal deferrals between Europe, East Asia and North America. This problem will be removed with the introduction of a new time scale based on a market activity model.

The problem of forecasting the business time scale is the smaller one at least when we consider a special country, because normally we know when weekends and holidays occur.

6.11.1 Business Time Scale

A very simple business time scale only count business hours while omitting inactive periods such as weekends and holidays. Modifying this approach one expands periods of high volatility (e.g. overlapping of markets) whereas periods of low activity (e.g. lunch time) are contracted.

We will consider a time based on a market activity model.

Definition 6.11.1 (Market Activity Variable) .
The market activity variable mesures the market activity between the physical time points t_1 and t_2 as follow

$$a_{1,2} = \frac{\Theta(t_2) - \Theta(t_1)}{t_2 - t_1}, \tag{6.14}$$

where Θ is a directing process which accounts for all seasonality effects, and which relates the theoretical market activity variable to the volatility of the returns [Dacorogna et al., 1993].

The idea is to approximate the observed intra-weekly activity pattern with a smooth function to get an easy model for the prediction of the activity variable and to define a forward θ time based on this smooth function. This smooth function is based on fundamental geographical considerations, due to the fact there are three main markets centers with different trading hours around the world.

Definition 6.11.2 (θ Time) .
The θ-time model defines an activity function $a(t)$ being the sum of three activity functions that can be associated with three markets: East Asia, Europe, and North America:

$$a(t) = a_0 + \sum_{k=1}^{3} a'_k(t), \tag{6.15}$$

where $a'_k(t)$ is a function of physical time t for market k. It is a measure of the market activity at time t [Muller et al., 1995].

The θ-time is then obtained via integration:

$$\theta(t) = \int_{t_0}^{t} \big(a_0 + \sum_{k=1}^{3} a'_k(t)\big) dt. \tag{6.16}$$

In practice, we want to relate this theoretical market activity to the volatility of the returns, such as:

$$\mathbb{E}[|r|] = c\Delta t^D. \tag{6.17}$$

This scaling law can be applied to subsamples in an intra-week analysis, in order to analyse the daily and weekly seasonalities. We divide a week into 168 hourly samples, and index the samples with $h = 1, 2, \cdots, 168$. All variables are also denoted with subscript h. With the help of this intra-week analysis and the knowledge that different volatilities for different hours are observed, we define a θ time as follow:

$$\Delta\Theta_h = \Big(\frac{E[|r_h|]}{c^*}\Big)^{\frac{1}{D}} \tag{6.18}$$

$$= k\,\Big(\frac{E[|r_h|]}{c}\Big)^{\frac{1}{D}}, \tag{6.19}$$

with $c^* = \frac{c}{k^D}$.

The idea is to solve the scaling law in every h^{th} subsample to Δt and to replace Δt with $\Delta\theta_h$.

6.11.2 Intrinsic Time Scale

Now, we base the definition of a new intrinsic τ time not on the raw time series in physical time, but on the time serties in θ time.

Definition 6.11.3 (Intrinsic τ Time) .

$$\tau(t_c) = \tau(t_{c-1}) + k\,\frac{\theta(t_c) - \theta(t_{c-1})}{t_c - t_{c-1}}\,\frac{|\Delta z|^{\frac{1}{D}}}{c}, \tag{6.20}$$

where

- *z is the time series,*
- *t_c is the current time,*
- *the price difference Δz is taken on the same interval as $\Delta\theta$,*
- *$\Delta z = z(t_c) - z(t_{c-1})$,*
- *the factor k is a calibration depending on the particular time series. It keeps τ in line with the physical time in the long run. That means τ time flows neither slowly nor faster than physical time or θ time.*

We can also write

$$\Delta\tau_{t_h} = \Delta\tau_h = k\,\frac{\Delta\theta_h}{\Delta\theta}\Big(\frac{\big|r_h^\theta(\Delta\theta)\big|}{c}\Big)^{\frac{1}{D}} \tag{6.21}$$

The idea for the definition of the intrinsic τ time is the same as the one of the definition of the business θ time.

Comparing both definitions we see that

- c^* is equivalent to c/k^D,
- $\Delta\theta_h/\Delta\theta$ is only the scaling factor to be able to compute τ time for other interval sizes than $\Delta\theta$.

The expectation operator is not necessary because there is only one θ time to obtain. However, this times, we start with θ time and end up with τ time, whereas last time we started with physical time t and ended up in θ time.

We see that the τ time is obtained only by applying the time deformation procedure twice. The aim of both time transformations is to expand periods of high volatility and contract periods of low volatility measured in the corresponding time scale.

A comprehensive development of this technique and its application to foreign exchange markets can be found in [Bruckner and Nolte, 2002].

6.12 Operators on Inhomogeneous Time Series

6.12.1 Linear operators

We represent a linear operator by a convolution with a kernel $\omega(t)$:

$$\Omega[z](t) = \int_{-\infty}^{t} \omega(t-s)\; z(s)\; ds \tag{6.22}$$

$$= \int_{0}^{\infty} \omega(s)\; z(t-s)\; ds. \tag{6.23}$$

The kernel $\omega(t)$ is defined on the semi-axis $t \geq 0$, and should decay for $t$large enough. The value of $\omega(t-s)$ is the weight of events in the past, at a time interval $(t-s)$ from t. In this convolution, $z(t)$ is a continuous time function.

We consider two families of operators:

- an average operator has a kernel which is non-negative, $\omega(t) \geq 0$, and normalized to unity, $\int \omega(t)dt = 1$. This gives $\Omega[parameter; Const] = Const$.
- derivative and difference operators have kernels and that measures the difference between a value now and a value in the past (with a typical lag τ). Their kernels have a zero average, $\int \omega(t)dt = 0$, such that $\Omega[parameters; Const] = 0$.

6.12.2 Range and width

The n-th moment of a causal kernel ω is:

$$\langle t^n \rangle_\omega = \int_0^\infty \omega(t) t^n dt. \tag{6.24}$$

The range r and the width w of an operator ω are defined by:

$$r[\omega] = \langle t \rangle_\omega = \int_0^\infty \omega(t)\ t\ dt \tag{6.25}$$

$$w^2[\omega] = \langle (t-r)^2 \rangle_\omega = \int_0^\infty \omega(t)\ (t-r)^2\ dt. \tag{6.26}$$

For most operators $\Omega[\tau]$ depending on a time range τ, the formula is set up so that:

$$\Big| r\big[\omega[\tau]\big] \Big| = \tau. \tag{6.27}$$

6.12.3 Convolution of kernels

A common computation is to successively apply two linear operators:

$$\Omega_c[z] = \Omega_2 \circ \Omega_1[z] = \Omega_2\big[\Omega_1[z]\big], \tag{6.28}$$

Ω_c is given by the convolution of the kernels of Ω_1 and Ω_2:

$$\omega_c = \omega_1 \star \omega_2 \tag{6.29}$$

$$\omega_c(t-s) = \int_{-\infty}^\infty \omega_1(t-v)\ \omega_2(v-s)\ dv. \tag{6.30}$$

6.12.4 Build-up time interval

Most kernels have an exponential tail for large t. This implies that, when starting the evaluation of an operator at time T, a build-up time interval must be elapsed before the result of the evaluation is meaningful; that the initial conditions at T are sufficiently forgotten.

We assume that the process $z(t)$ is known since T and is modeled before as an unknown random walk with no drift. In this case we have:

$$\Omega[T; z](t) = \int_T^t \omega(t-s)\ z(s)\ ds. \tag{6.31}$$

This infinite build-up corresponds to $\Omega[-\infty; z](t)$.

6.12.5 Robustness

Data errors (outliers) should be filtered prior to any computation. Outlier filtering is difficult and sometimes arbitrary for high-frequency data in finance. This data is stochastic with a fat-tailed distribution of price changes. Thus it is very desirable to build robust estimators to reduce the impact of outliers and the choice of the filtering algorithm.

6.12.6 Exponential moving average operator $\boldsymbol{EMA}[\boldsymbol{\tau}]$

Let consider z a generic inhomogeneous time series with values z_i at random time-point t_i such as $z_i = z(t_i)$. The sequence of sampling times is required to be growing, $t_i > t_{i-1}$.

Let $EMA[\theta, z](t)$ be an exponential moving average operator, from the space of time series z into itself, depending on parameter θ, with value at time t.

This Exponential Moving Average Operator computes a moving average with an exponentially losing power of the past, and it has an exponentially decaying kernel. Where τ determines the range of the operator and t indexes the time.

$$ema(t; \tau) = \frac{1}{\tau} \exp(-t), \tag{6.32}$$

which leads to a simple iterative formula for the convolution operator $EMA[\tau]$:

$$EMA[\tau; z](t_n) = \mu\ EMA[\tau; z](t_{n-1}) + (\nu - \mu) z_{n-1} + (1 - \nu) z_n, \tag{6.33}$$

with

- $\alpha = t_n - t_{n-1}/\tau$
- $\mu = \exp(-\alpha)$
- $\nu = (1-\mu)/\alpha$

The basic operator can be iterated

$$EMA[\tau, n; z](.) = EMA[\tau;\ EMA[\tau, n-1; z]](.), \tag{6.34}$$

where $EMA[\tau, 1; z](.) = EMA[\tau; z](.)$.

With this operator, we can get some useful tools:

1. Moving Average MA

$$MA[\tau, n] = \frac{1}{n}\sum_{k=1}^{n} EMA[\tau^{'}, k], \tag{6.35}$$

   with $\tau^{'} = 2\ \tau/(n+1)$
2. Moving Norm $MNorm$

$$MNorm[\tau, p; z] = MA[\tau; |z|^{1/p}], \tag{6.36}$$

3. Moving Variance MV

$$MVar[\tau, p; z] = MA[\tau; |z - MA[\tau; |z|]|^{p}], \tag{6.37}$$

4. Moving Standard Deviation $[MSD]$

$$MSD[\tau, p; z] = MA[\tau; |z - MA[\tau; z]|^{p}]^{1/p}, \tag{6.38}$$

5. Differential $\Delta[\tau]$

$$\Delta[\tau] = \gamma\ EMA[\alpha\ \tau, 1] + EMA[\alpha\ \tau, 2] - 2\ EMA[\alpha\ \beta\ \tau, 4]), \tag{6.39}$$

   with $\alpha = 1.22208$, $\beta = 0.65$, and $\alpha^{-1} = \gamma(8\ \beta - 3)$.
6. Derivative $D[\tau]$

$$D[\tau] = \frac{\Delta[\tau]}{\tau}, \tag{6.40}$$

7. γ-Deivative $D[\tau, \gamma]$

$$D[\tau, \gamma] = \frac{\Delta[\tau]}{\left(\frac{\tau}{1\text{year}}\right)^{\gamma}}, \tag{6.41}$$

   with $\gamma = 0$ for differential, $\gamma = 0.5$ for stochastic diffusion process, and $\gamma = 1$ for usual derivative.

From a time series z, we can derive a moving standardized time series

$$\widehat{z}[\tau] = \frac{z - MA[\tau; z]}{MSD[\tau; z]}, \tag{6.42}$$

from which, we can get the moving skewness and the moving kurtosis

$$MSkewness[\tau_1, \tau_2; z] = MA\big[\tau_1; \widehat{z}[\tau_2]^3\big], \tag{6.43}$$

$$MKurtosis[\tau_1, \tau_2; z] = MA\big[\tau_1; \widehat{z}[\tau_2]^4\big], \tag{6.44}$$

and also some moving correlations for inhomogeneous time series

$$MCorrelation[\tau; y; z] = \frac{MA\big[(y - MA[y])(z - MA[z])\big]}{\big(MSD[y]MSD[z]\big)}. \tag{6.45}$$

These operators have been used for filtering of financial quotes by Olsen and Associates for many years, and we can find in [Muller , 2001] some filtering algorithms to clean-up bad quotes.

6.13 Stochastic Volatility Models

In this section we give an introduction to theoretical financial models, with an overview on Stochastic Differential Volatility Models in continuous time and their approximation by Stochastic Difference Models in discrete time.

6.13.1 Continuous Time Models

The theory of finance is mainly treated in terms of stochastic differential equations such as, the value of a stock price S_t is supposed to follow a diffusion geometrical Brownian motion

$$dS_t = S_t \left(\mu_t dt + \sqrt{V_t} dB_t\right), \tag{6.46}$$

where

- μ_t is the expected drift rate function (i.e., average drift per unit of time),
- $\sqrt{V_t}$ is the volatility rate function of the stock price,
- B_t is a Wiener process.

For a stochastic volatility model, the volatility function is also modeled as Brownian motion,

$$dV_t = \alpha(S_t, t)dt + \beta(S_t, t)dZ_t, \tag{6.47}$$

where

- Z_t is a Wiener process,
- B_t and Z_t are correlated, such as

$$dB_t \, dZ_t = \rho dt, \tag{6.48}$$

where ρ is the correlation coefficient between these two Brownian motions. We can also rewrite Z_t as:

$$Z_t = \rho B_t + \sqrt{1-\rho^2}\widetilde{Z}_t, \tag{6.49}$$

where $\widetilde{Z}_t$ is a standard Brownian motion independent of B_t. There are some economic arguments for a negative correlation between stock price and volatility stocks. Empirical studies also show $\rho < 0$ from stock data.

The form of V_t depends on the particular stochastic volatility model under study, and where $\alpha(S_t, t)$ and $\beta(S_t, t)$ are some functions of V_t [Andersen et al., 2005].

1. **Hull-White**

   Hull and White [Hull, 2000] assume a geometric Brownian motion for the volatility

   $$dV_t = \alpha V_t dt + \beta V_t dZ_t \tag{6.50}$$

   where α and β are constants.

2. **Stein-Stein**

   Stein and stein [Stein and Stein , 1991] assume the driving process V_t is an Ornstein-Uhlenbeck (O-U) process (see Appendix: A.4.2)

   $$dV_t = \alpha(\omega - V_t)dt + \beta dZ_t. \tag{6.51}$$

   It is a mean-reversing process. From econometric studies, people believe that volatility is mean-reversing. So the O-U process is employed by many researchers to model the volatility. But it is not appropriate to simply assume the volatility is an O-U process, because V_t can be negative in O-U process.

3. **Heston**

   Heston [Heston , 1993] assumes that V_t follows a Cox-Ingersoll-Ross (CIR) process

   $$dV_t = \kappa(\theta - V_t)dt + \xi\sqrt{V_t}dZ_t, \tag{6.52}$$

   V_t is strictly positive when $2\kappa\theta \geq \xi^2$ and non-negative when $0 \leq 2\kappa\theta < \xi^2$.

6.13.2 Dynamic State-Space Stochastic Volatility Model

Let's consider the stochastic volatility model in continuous time:

$$dS_t = S_t \left(\mu_t dt + \sqrt{V_t} dB_t\right) \tag{6.53}$$
$$dV_t = \kappa(\theta - V_t)dt + \xi V_t^p dZ_t, \tag{6.54}$$

where

- S_t is the asset price at time t,
- $\sqrt{V_t}$ is the implied volatility,
- κ, θ, and ξ are fixed constants, and have to be estimated,
- $p = \frac{1}{2}$ for a Heston model, $p = 1$ for a Garch model, and $p = \frac{3}{2}$ for a $\frac{3}{2}$ model.

If we consider Brownian motion processes, such as Wiener representations with Gaussian distributions, for noises in price and volatility equations, these variables can take positive and negative values. But it is impossible to get negative values for prices or volatilities. To get around this problem, we have to consider log-normal distributions for these variables, by realizing a logarithmic transformation of prices and volatilities.

We take the logarithms of stock price, $y_t = \log(S_t)$ and of the volatility, $h_t = \log(V_t)$. We use the Itô's formula [A.3.3] to derive the process in a continuous dynamic state-space formulation:

$$dy_t = \left[\mu_t - \frac{1}{2}V_t\right] dt + \sqrt{V_t}\, dB_t \tag{6.55}$$
$$dh_t = \kappa\left[\theta - V_t\right]dt + \xi V_t^p dZ_t\ , \tag{6.56}$$

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.

6.13.3 Parameter 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. However the likelihood function in diffusion-type volatility models is extremely difficult to evaluate. Markov's property is lost due to stochastic volatility, therefore it is only possible to simulate the likelihood function [Ait-Sahalia, 2002].

Bayes methods, such as *Markov Chain Monte Carlo*(MCMC), do not encounter this issue, because inference can be jointly made for the augmented parameter vector including the latent variables as an element. This method essentially calls for construction of various conditional densities from which samples of parameters and the instantaneous volatility vector are drawn.

6.13.4 Discrete Dynamic State Space Model

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

$$\log V_{k+1} = \log V_k + \frac{1}{V_k}\big[\kappa(\theta - V_k) - \frac{1}{2}\xi^2 V_k^{2p-1} - \rho\xi V_k^{p-\frac{1}{2}}(\mu - \frac{1}{2}V_k)\big]\Delta t + \rho\xi V_k^{p-\frac{3}{2}}\big(\log S_k - \log S_{k-1}\big) + \xi V_k^{p-1}\sqrt{\Delta t}\sqrt{1-\rho^2}\widetilde{Z}_k \quad (6.57)$$

$$\log S_k = \log S_{k-1} + \big(\mu - \frac{1}{2}V_k\big)\Delta t + \sqrt{\Delta t}\sqrt{V_k}B_k, \quad (6.58)$$

where Wiener processes $\widetilde{Z}_k$ and B_t are uncorrelated, using

$$\widetilde{Z}_k = \frac{1}{\sqrt{1-\rho^2}}(z_t - \rho B_t). \quad (6.59)$$

These equations define a *Dynamic State Space Model* (DSSM) in discrete-time, where Eq. (6.57) is the nonlinear state equation, and Eq. (6.58) is the nonlinear observation equation. But in these equations, B_k and $\widetilde{Z}_k$ are now white Gaussian discrete noises as a derivative of the Wiener processes $B(t)$ and $\widetilde{Z}(t)$.

We could realize the estimation of parameters κ, θ, μ, and ξ of this system of stochastic difference Eq. (6.57) and Eq. (6.58) by Bayes filtering in discrete-time as in [Do, 2005], [Alspach et al., 1972], [Javaheri et al., 2003], [Bolland, and Connor, 1997], [Xu , 2005].

But we cannot pass these estimations onto Eq. (6.55) and Eq. (6.56) because in continuous-time, a white noise cannot exist in the mathematic sense. Thus, we have to realize a parameters estimation by filtering of the continuous dynamic state-space formulation with a Wiener process for the noises.

6.13.5 Space representation of prices

In financial theory, we represent asset prices as a continuous function in continuous time.

In practice, we have inhomogeneous high-frequency time series of "tick-by-tick" data in discrete time, because the transactions happen at random time points $\{t_1, t_2, \cdots, t_n\}$, with non equal intervals:

$$\Delta t_k \doteq (t_k - t_{k-1}) \neq \Delta t_l \doteq (t_l - t_{l-1}) \qquad \text{for } k \neq l \tag{6.60}$$

The "price" function is defined in a continuous $\mathbb{R}$ space, because the transactions can happen at any time during the opening of the market.

But we do not have a "value" function, in a continuous $\mathbb{R}$ space, due to market restrictions. Prices change by "ticks". A "tick" refers to a minimum change in value up or down (e.g. 1/16 \$), and the number of "ticks" between two transactions can also be limited by the market. Because Institution rules can restrict prices to fall onto a restricted set of multiples of ticks, in order to discouraging prices changes from one transaction to the next one. That means, price "values" are in a discrete $\mathbb{N}$ space, given by $\{p_0, p_1, p_2, \cdots\}$, where $p_k = k \times 1/16\$$.

We will have to take note of these specificities in the algorithms.

Part III

Mathematical Developments

Part Three develops the Mathematical tools which we use in the construction of Forecasting Models.

7

Functional Analysis for Classification and Clustering

This chapter gives a description of Functional Clustering and Classification applied to curves.

This chapter gives a description of Functional Clustering and Classification applied to curves, used for the "*Price Forecasting model*" describes in Chapter [4], and for "*Model of Trading*" describes in Chapter [5].

7.1 Introduction

Modern financial data sets may contain tens of thousands of transactions per day stamped to the nearest second. The analysis of these data are complicated due to stochastic temporal spacing, diurnal patterns, price discreteness, and complex temporal dependence.

Let consider data coming from an interval $[t_0, t_N]$. If we use very liquid series, like stocks, with observations spread over the whole interval of time, we could realize the smoothing by splines and afterwards the clustering from the spline coefficients, in two separate steps. If we do not have enough data near the limits t_0 or t_N of the interval, the smoothing by splines will not be accurate near these limits, we will have many outliers and the clustering will be very poor. If we use poorly liquid data with large fragments of curves without observations, like options, the smoothing could be very chaotic. In those cases we have to jointly smooth and classify the observations through an iterative procedure.

In summary, when the observations are sparse, irregularly spaced, or occur at different time points for each subject and moreover when only fragments of the functions are available, with the Linear Discriminant Analysis (LDA), many of the basis coefficients would have infinite variance, making it impossible to produce reasonable estimates [James et al., 2000]. Similar problems arise with clustering methods.

In this case, we will use a random effects model for the coefficients and we will realize, in the same step, the estimation of the spline coefficients for the classification or the clustering [James and Sugar, 2003].

This chapter is organized as follow:

- in section [7.2] we introduce the Functional data,
- in section [7.3] we describe the Functional Clustering.

7.2 Functional Data

7.2.1 Introduction

Financial time series are observed and recorded at randomly discrete time points, such as n pairs $\{y_i, t_i\}$, where y_i is the observation of a hidden real value continuous function $g(t_i)$ at time t_i. We consider techniques for converting these raw data y_i into a functional form by smoothing.

We consider the homoscedastic discrete noise model:

$$y_i = g(t_i) + \epsilon_i \quad i = 1, \cdots, n \tag{7.1}$$

where the disturbances ϵ_i are i.i.d., with $var(\epsilon_i) = \sigma^2$, and contribue to the roughness to the raw data. But in some financial applications we must take explicit account of nonhomogeneous variance.

The data y_i are noisy observations of the real hidden value $g(t_i)$. We consider an approximation $\widetilde{g}(t_i)$ of $g(t_i)$, and evaluating $\widehat{g}(t)$ involves some form of smoothing of the discrete observations y_i.

We will consider that the range of t is a bounded intervalle $[0, T]$, and that $\widetilde{g}$ satisfies reasonable continuity conditions on $[0, T]$, such as a certain number of derivatives exit at t.

The sampling rate of raw data is an essentially local property to the data, and the quality of the smoothing depends on the curvature in the data rather than the number n of observations, measured by $d^2\widetilde{g}(t)/dt$. If the curvature is high, we need enough points to estimate the function $\widetilde{g}(t)$.

7.2.2 Splines

We consider a spline to approximate an unknown regression function $\widetilde{g}(t)$, from discrete observations y_i.

Let consider a fixed sequence of J knots, such as:

$$-\infty < t_1 < t_2 < \cdots < t_{J-1} < t_J < \infty,$$

where t_1, and t_J define the interval $[0, T]$ over which the estimation is to take place.

A cubic-spline $z(t)$ is such as:

- $z(t)$ is a piecewise cubic polynomial onto the intervals:

$$]-\infty, t_1], [t_1, t_2], \cdots, [t_{J-1}, t_J], [t_J, \infty[$$

- $z(t)$ has continuous first two derivatives.

This collection of cubic-spline functions forms a q-dimensional linear space.

Let consider a cubic-spline basis $\{s_1, \cdots, s_q\}$. Then we can represent a function $z(t)$ by a linear combination of q known basis functions $s_i(t)$:

$$z(t) = \sum_{j=1}^{q} \eta_j \; s_j(t). \tag{7.2}$$

For the given knots, the spline method consists of finding the spline coefficients η_i by minimizing the least squares criterion $SMSSE(\mathbf{y}|\boldsymbol{\eta})$:

$$\min_{\boldsymbol{\eta}} \sum_{i=1}^{n} \left[y_i - \sum_{j=1}^{q} \eta_j \; s_j(t_i) \right]^2. \tag{7.3}$$

Let $\widehat{\boldsymbol{\eta}}$ be the best least squares estimates. Then the spline approach estimates $\widetilde{g}(t)$ by the spline function is:

$$\widetilde{g}(t) \approx \sum_{j=1}^{q} \widehat{\eta}_i \; s_j(t) \tag{7.4}$$

The above described spline method is sensitive to the choice of the knots: their number, and localization, and it will be depending on the problem in interest.

B-spline Basis

In the follow, we will use a B-spline basis, q-dimensional, where order of a B-spline basis plus the number of interior knots equals the number of basis functions. Order 4 (degree of the piecewise polynomial $+1$) is a frequent choice, implying piecewise cubic polynomials.

7.3 Functional Clustering

7.3.1 Motivation

Our purpose is to cluster the observations of time series observed in random time points into classes having homogenous properties.

When the observations are sparse, irregularly spaced, or occur at different time points for each subject, standard statistical tools cannot be used for two reasons:

1. the number of observations differs for each series,
2. the observations are not at the same time-point for each series.

Therefore it is necessary to represent these data by a fixed number of features.

One way to reach this goal is to smooth the rough data by projecting them onto a *functional basis*, for example cubic splines. Then, the coefficients of this projection may be used in a more standard way for clustering purposes.

7.3.2 Introduction

We will use basis functions in order to convert the original infinite-dimensional problem into a finite-dimensional one, but instead of treating the basis coefficients as parameters and fitting a separate spline for each individual, we will use *random-effects models* for the coefficients [Laird and Ware, 1982], and [Guo, 2002].

We use all curves of the cluster to smooth a single curve by using information in the collection to estimate its coefficients. This procedure borrows 'information' across curves, and it is well defined even when the observations on a particular subject are too sparse to support an ordinary least square fit, provided that the total number of observations is large enough. Moreover, it automatically weights the estimated spline coefficients according to their variances, which is highly efficient because it requires fitting few parameters [Rice and Wu, 2001].

The main idea for the Model-based Clustering in Panel Data comes from [Fruhwirth et al. , 2004]. The Functional Clustering method presented in this section is based on [James et al., 2000], [James and Sugar, 2003], [Dempster et al., 1984], [Harville , 1977], [Harville and Carriquiry , 1992], [Harville , 1976], [Henderson , 1982], and [Cunningham and Henderson , 1968].

7.3.3 Classical Clustering

Cluster analysis consists in identifying groups in data; it is the dual form of discriminant analysis, but in cluster analysis the group labels are not known a priori. It is an unsupervised process.

We assume that the observations $\{\mathbf{y}_1, \cdots, \mathbf{y}_N\}$ are generated according to a mixture distribution with G clusters. Let $f_k(\mathbf{y}|\theta_k)$ be the density distribution function corresponding to cluster k, with parameters θ_k, and let $\big(1_{\{k\}}(i)\big)$ be the cluster membership (indicator function of cluster k) for the observation i where $\big(1_{\{k\}}(i) = 1\big)$ if y_i is a member of cluster k and 0 otherwise.

The indicators $\big(1_{\{k\}}(i)\big)$ are unknown and they are treated in two ways:

- in the "classification likelihood" procedure, the indicators $\big(1_{\{k\}}(i)\big)$ are considered as parameters and the model is fit by maximization the likelihood:

$$L_c\big(\theta_1, \cdots, \theta_G \,;\, 1_{\{k\}}(1), \cdots, 1_{\{k\}}(N) \mid \mathbf{y}_1, \cdots, \mathbf{y}_N\big) = \prod_{i=1}^{N} f_{1_{\{k\}}(i)}\big(\mathbf{y}_i \mid \theta_{1_{\{k\}}(i)}\big) \,, \tag{7.5}$$

  when $f_k\big(\mathbf{y} \mid \theta_k\big)$ is multivariate normal with identity covariance matrix; this approach produces the k-means solution.

- in the "mixture likelihood", the cluster memberships $\big(1_{\{k\}}(i)\big)$ may be treated as missing data where $\big(1_{\{k\}}(i)\big)$ is multinomial with parameters $[\pi_1, \cdots, \pi_G]$, where π_k is the probability that an observation belongs to cluster k. Then the parameters are estimated by maximizing the likelihood:

$$L_M(\theta_1, \cdots, \theta_G \,;\, \pi_1, \cdots, \pi_G \mid \mathbf{y}_1, \cdots, \mathbf{y}_N) = \prod_{i=1}^{N} \sum_{k=1}^{G} \pi_k \, f_k(\mathbf{y}_i|\theta_k) \,. \tag{7.6}$$

The maximum likelihood corresponds to the most probable model, given the observations
$\{\mathbf{y}_1, \mathbf{y}_2 \cdots, \mathbf{y}_N\}$. But in both approaches it is generally necessary to use an iterative EM procedure to estimate the various parameters.

For the clustering of functional data, we have two methods:

- the regularization method works by discretization of the time interval, but generally the resulting data vectors are highly correlated and high-dimensional [DiPillo, 1976]; [Hastie et al., 1995], and by resampling at low frequency we loose much information [Ait-Sahalia and Myland, 2003],

- the filtering method project each curve (infinite-dimensional data) onto a finite-dimensional *basis* $\phi(x)$, and find the best projection of each curve onto this basis. The resulting basis coefficients can than be used as a finite-dimensional representation making it possible to use classical clustering methods on the basis coefficients [Ramsay, and Silverman , 1997].

These approaches can work well when every curve has been observed over the same fine grid of points, but they break down if the individual curves are sparsely sampled:

- the regularization method cannot be used for inhomogeneous time series, because the curves are sampled at different times,

- the filtering method also has some problems. Indeed in this case, the variance of the estimated basis coefficients is different for each individual because the curves are measured at different time-points. Moreover, for very sparse data sets many of the basis coefficients would have infinite variance, making it impossible to produce reasonable estimates. They are so few observations that it is impossible to fit a separate curve for each individual using a reasonable common basis.

A solution is to convert the original infinite-dimensional problem into a finite-dimensional one using basis functions and using a *random effects model* for the coefficients [Harville , 1977], [Harville and Carriquiry , 1992], [Harville , 1976], and [Henderson , 1982], and realizing, by iterations, in the same step, the smoothing and the clustering of curves.

7.3.4 Functional Clustering Model

Let $g_i(t)$ the hidden true value for the curve i at time t and $\mathbf{g}_i$, $\mathbf{y}_i$ and $\boldsymbol{\epsilon}_i$, the random vectors of, respectively, hidden true values, measurements and errors. We have :

$$\mathbf{y}_i = \mathbf{g}_i + \boldsymbol{\epsilon}_i, \qquad i = 1, \cdots, N \ , \tag{7.7}$$

where N is the number of curves. The random errors $\boldsymbol{\epsilon}_i$ are assumed i.i.d., having a Gaussian distribution with zero mean and covariance σ^2, uncorrelated

with each other and with $\mathbf{g}_i$.

Observing curve i on interval $[t_1, \ldots, t_l, \cdots, t_{n_i}]$ we define the vectors :

$$\mathbf{g}_i = \Big(g_i(t_1), \cdots, g_i(t_l), \cdots, g_i(t_{n_i})\Big)^T, \tag{7.8}$$

$$\mathbf{y}_i = \Big(y_i(t_1), \cdots, y_i(t_l), \cdots, y_i(t_{n_i})\Big)^T, \tag{7.9}$$

where t_l is the time-point for observation l of curve i, and n_i is the number of observations for curve i.

Strictly speaking, the curves are observed at irregularly and randomly spaced times. This means, for example, that the first time observation t_1 on curves i and j may differ, and should therefore be defined as t_{i1} and t_{j1}. However, we will use the notation t_1 for any curve in the remaining of this thesis, in order to simplify the notations.

The unknown true functions $\mathbf{g}_i$, are approximated by functions $\widetilde{\mathbf{g}}_i$. We project $\widetilde{\mathbf{g}}_i$ onto a functional basis by smoothing such as:

$$\widehat{g}_i(t) \approx \mathbf{s}^T(t)\, \boldsymbol{\eta}_i \,, \tag{7.10}$$

where $\mathbf{s}(t)$ is the spline basis vector of dimension q, and $\boldsymbol{\eta}_i$ is a Gaussian random vector of spline coefficients:

$$\mathbf{s}(t) = \Big(s_1(t), \cdots, s_q(t)\Big)^T, \tag{7.11}$$

$$\boldsymbol{\eta}_i = \Big(\eta_{i1}, \cdots, \eta_{iq}\Big)^T. \tag{7.12}$$

The Gaussian coefficients $\boldsymbol{\eta}_i$ are split into two terms :

$$\boldsymbol{\eta}_i = \boldsymbol{\mu}_{ki} + \boldsymbol{\gamma}_i, \tag{7.13}$$

with $\boldsymbol{\gamma} \sim N(\mathbf{0}, \boldsymbol{\Gamma})$.

The term $\boldsymbol{\mu}_{ki}$ is defined by $\boldsymbol{\mu}_{ki} = \boldsymbol{\mu}_k$ if curve i belongs to cluster k, where $\boldsymbol{\mu}_k$ represents the centroid of cluster k, and $\boldsymbol{\gamma}_i$ represents the deviation between curve i and the centroid of its cluster k.

$$\boldsymbol{\mu}_k = \big(\mu_{k1}, \mu_{k2} \ldots \mu_{kq}\big)^T, \tag{7.14}$$

$$\boldsymbol{\gamma}_i = \big(\gamma_{i1}, \gamma_{i2} \ldots \gamma_{iq}\big)^T. \tag{7.15}$$

In this representation $\boldsymbol{\eta}_i$'s are treated as random effects and need not be estimated directly. With this procedure, we borrow strength across curves, that

gives superior results for data containing a large number of sparsely sampled curves.

In the same way, we can represent the deviation between the centroid of cluster k and the global mean of the population $\boldsymbol{\lambda}_0$ by :

$$\boldsymbol{\mu}_k = \boldsymbol{\lambda}_0 + \boldsymbol{\Lambda}\boldsymbol{\alpha}_k \ , \tag{7.16}$$

where $\boldsymbol{\lambda}_0$ is a q-dimensional vector, $\boldsymbol{\alpha}_k$ a h-dimensional one, and $\boldsymbol{\Lambda}$ a (q, h) matrix, with $h \leq \min(q, G-1)$:

$$\boldsymbol{\lambda}_0 = \Big(\lambda_{01}, \cdots, \lambda_{0q}\Big)^T, \tag{7.17}$$

$$\boldsymbol{\alpha}_k = \Big(\alpha_{k1}, \cdots, \alpha_{kh}\Big)^T, \tag{7.18}$$

$$\boldsymbol{\Lambda} = \begin{pmatrix} \lambda_{11} & \cdots & \lambda_{1h} \\ \vdots & \ddots & \vdots \\ \lambda_{p1} & \cdots & \lambda_{qh} \end{pmatrix}. \tag{7.19}$$

With this formulation, the functional clustering model can be written as :

$$\mathbf{y}_i = \mathbf{S}_i(\boldsymbol{\lambda}_0 + \boldsymbol{\Lambda}\boldsymbol{\alpha}_{ki} + \boldsymbol{\gamma}_i) + \boldsymbol{\epsilon}_i \ , \qquad i = 1, \cdots, N, \tag{7.20}$$

with: $\boldsymbol{\epsilon}_i \sim N(\mathbf{0}, \mathbf{R})$, and $\boldsymbol{\gamma}_i \sim N(\mathbf{0}, \boldsymbol{\Gamma})$.
$\mathbf{S}_i = \Big[\mathbf{s}(t_{i1}), \cdots, \mathbf{s}(t_{in_i})\Big]^T$ is the spline basis matrix for curve i :

$$\mathbf{S}_i = \begin{pmatrix} s_1(t_1) & \cdots & s_q(t_1) \\ \vdots & \ddots & \vdots \\ s_1(t_{n_i}) & \cdots & s_q(t_{n_i}) \end{pmatrix}. \tag{7.21}$$

The term $\boldsymbol{\alpha}_{ki}$ is defined similarly to $\boldsymbol{\mu}_{ki}$ by $\boldsymbol{\alpha}_{ki} = \boldsymbol{\alpha}_k$ if curve i belongs to cluster k, where $\boldsymbol{\alpha}_k$ is a representation of the centroid of cluster k in a reduced h-dimensional subspace, and

$$\boldsymbol{\alpha}_k = \big(\, \alpha_{k1},\, \alpha_{k2} \,\ldots\, \alpha_{kh} \,\big)^T. \tag{7.22}$$

Then we have :

- $\mathbf{s}(t)^T\boldsymbol{\lambda}_0$: the representation of the global mean curve,
- $\mathbf{s}(t)^T(\boldsymbol{\lambda}_0 + \boldsymbol{\Lambda}\boldsymbol{\alpha}_k)$: the global representation of the centroid of cluster k,
- $\mathbf{s}(t)^T\boldsymbol{\Lambda}\boldsymbol{\alpha}_k$: the local representation of the centroid of cluster k in connection with the global mean curve,
- $\mathbf{s}(t)^T\boldsymbol{\gamma}_i$: the local representation of the curve i in connection with the centroid of its cluster k.

7.3.5 Hypotheses

Because we work with sparse data sets and we would like to use a small number of parameters, we choose the representations for $\mathbf{R}$ and $\boldsymbol{\Gamma}$ as follow:

- as measurements errors are i.i.d., the noise covariance matrix $\mathbf{R}$ is diagonal and the variance terms will be supposed identical for all noises, thus we will set: $\mathbf{R} = \sigma^2\mathbf{I}$
- we also suppose the same distribution for observations in each cluster, thus the within-class covariance matrix $\boldsymbol{\Gamma}$ will be the same for every cluster.

7.3.6 Constraints

However, $\boldsymbol{\lambda}_0$, $\boldsymbol{\alpha}_k$, and $\boldsymbol{\Lambda}$ could be confounded if no constraints were imposed. Therefore we impose the following restrictions on $\boldsymbol{\Lambda}$ and the $\boldsymbol{\alpha}_k$'s,

$$\sum_k \boldsymbol{\alpha}_k = \mathbf{0} \,, \tag{7.23}$$

which means that $\mathbf{s}(t)^T\boldsymbol{\lambda_0}$ may be interpreted as the overall mean curve.

We also require that

$$\boldsymbol{\Lambda}^T\mathbf{S}^T\boldsymbol{\Sigma}^{-1}\mathbf{S}\boldsymbol{\Lambda} = \mathbf{I} \,, \tag{7.24}$$

with :

$$\boldsymbol{\Sigma} = \sigma^2\mathbf{I} + \mathbf{S}\boldsymbol{\Gamma}\mathbf{S}^T, \tag{7.25}$$

where $\mathbf{S}$ is the spline basis matrix on a fine grid of time points over the full range of the data, in the interval $[0, \ldots, T]$. In practice this grid should include all time points in the data set. The unit of time for the observations will be chosen for the increments of this grid.

The reason for the particular form used in (7.24) will be given in Section [7.3.8].

Remark 7.3.1 ($\boldsymbol{\eta}_i$'s as random effects) .

In the filtering method, the $\boldsymbol{\eta}_i$ are treated as parameters or fixed effects and are estimated directly using only the values corresponding to that individual. This method does not give a robust estimation of spline coefficients, especially in case of very sparse time series.

In the functional clustering method, the $\boldsymbol{\eta}_i$ are treated as random effects and need not be estimated directly,

- *this allows properties to be shared across curves, providing superior results for inhomogeneous time series containing a large number of sparsely sampled curves,*
- *using* $\big(h < G-1\big)$ *reduces the number of parameters to be estimated which can result in a superior fit for sparse data,*
- *this parameterizations leads to a low-dimensional representation of the individual curves.*

7.3.7 Parametric Identification

We have to estimate the parameters $\boldsymbol{\lambda}_0$, $\boldsymbol{\Lambda}$, $\boldsymbol{\alpha}_k$, $\boldsymbol{\Gamma}$, σ^2 et π_k by maximization of a likelihood function.

For $\mathbf{y}_i$ we have a conditional distribution :

$$\mathbf{y}_i \sim N\Big(\mathbf{S}_i(\boldsymbol{\lambda}_0 + \boldsymbol{\Lambda}\boldsymbol{\alpha}_{ki}), \boldsymbol{\Sigma}_i\Big) \ , \tag{7.26}$$

where

$$\boldsymbol{\Sigma}_i = \sigma^2\mathbf{I} + \mathbf{S}_i\boldsymbol{\Gamma}\mathbf{S}_i^T \ . \tag{7.27}$$

We define $\mathbf{z}_i$ as the unknown cluster membership vector of curve i, which will be treated as missing data,

$$\mathbf{z}_i = \big(z_{1i} \ z_{2i} \ \cdots \ z_{ki} \ \cdots \ z_{Gi} \big) \tag{7.28}$$

with :

$$z_{ki} = \begin{cases} 1 & \text{if curve } i \text{ belongs to cluster } k, \\ 0 & \text{otherwise.} \end{cases}$$

The probability that curve i belongs to cluster k is then :

$$\pi_{ki} = P\big(z_{ki} = 1\big). \tag{7.29}$$

When the observations of the different curves are independent, the joint distribution of $\mathbf{y}$ and $\mathbf{z}$ is given by :

$$f(\mathbf{y}, \mathbf{z}) = \sum_{k=1}^{G} \pi_k \frac{1}{(2\pi)^{\frac{n}{2}}|\boldsymbol{\Sigma}|^{\frac{1}{2}}} \exp\Big[-\frac{1}{2}\big(\mathbf{y} - \mathbf{S}(\boldsymbol{\lambda}_0 + \boldsymbol{\Lambda}\boldsymbol{\alpha}_k)\big)^T\boldsymbol{\Sigma}^{-1}\big(\mathbf{y} - \mathbf{S}(\boldsymbol{\lambda}_0 + \boldsymbol{\Lambda}\boldsymbol{\alpha}_k)\big)\Big] \ , \tag{7.30}$$

and the likelihood for the parameters $\pi_k, \boldsymbol{\lambda}_0, \boldsymbol{\Lambda}, \boldsymbol{\alpha}_k, \boldsymbol{\Gamma}, \sigma^2$, given observations $\mathbf{y}_{1:N}$, and $\mathbf{z}_{1:N}$ is

$$L(\pi_k, \boldsymbol{\lambda}_0, \boldsymbol{\Lambda}, \boldsymbol{\alpha}_k, \boldsymbol{\Gamma}, \sigma^2 | \mathbf{y}_{1:N}, \mathbf{z}_{1:N}) = \prod_{i=1}^{N} \sum_{k=1}^{G} \pi_k \frac{1}{(2\pi)^{\frac{n_i}{2}} |\boldsymbol{\Sigma}_i|^{\frac{1}{2}}} \times \exp\Big[-\frac{1}{2} \big(\mathbf{y}_i - \mathbf{S}_i(\boldsymbol{\lambda}_0 + \boldsymbol{\Lambda}\boldsymbol{\alpha}_{ki})\big)^T \boldsymbol{\Sigma}_i^{-1} \big(\mathbf{y}_i - \mathbf{S}_i(\boldsymbol{\lambda}_0 + \boldsymbol{\Lambda}\boldsymbol{\alpha}_{ki})\big)\Big] . \tag{7.31}$$

Maximizing this likelihood would give us the parameters $\boldsymbol{\lambda}_0$, $\boldsymbol{\Lambda}$, $\boldsymbol{\alpha}_k$, π_k , $\boldsymbol{\Gamma}$ and σ^2.

Unfortunately, a direct maximization of this likelihood is a difficult non-convex optimization problem. If the $\boldsymbol{\gamma}_i$ were observed, the joint likelihood of $\mathbf{y}_i$, $\mathbf{z}_i$ and $\boldsymbol{\gamma}_i$ would simplify. As $\mathbf{z}_i$ and $\boldsymbol{\gamma}_i$ are independent, the joint distribution can now be written as :

$$f(\mathbf{y}, \mathbf{z}, \boldsymbol{\gamma}) = f(\mathbf{y}|\mathbf{z}, \boldsymbol{\gamma}) f(\mathbf{z}) f(\boldsymbol{\gamma}) , \tag{7.32}$$

where for each curve i,

- $\mathbf{z}_i$ are multinomial (π_k),
- $\boldsymbol{\gamma}_i$ are $N(0, \boldsymbol{\Gamma})$,
- $\mathbf{y}_i$ are conditional $\mathcal{N}\Big(\mathbf{S}_i(\boldsymbol{\lambda}_0 + \boldsymbol{\Lambda}\boldsymbol{\alpha}_k + \boldsymbol{\gamma}_i); \sigma^2\mathbf{I}\Big)$.

The joint distribution is now written as :

$$f(\mathbf{y}, \mathbf{z}, \boldsymbol{\gamma}) = \frac{1}{(2\pi)^{\frac{n+q}{2}} |\boldsymbol{\Gamma}|^{\frac{1}{2}}} \exp\big[-\frac{1}{2} \boldsymbol{\gamma}^T \boldsymbol{\Gamma}^{-1} \boldsymbol{\gamma}\big] \prod_{k=1}^{G} \Big\{ \pi_k \exp\big[-\frac{1}{2} n \log(\sigma^2)\big] \exp\big[-\frac{1}{2\sigma^2} \big(\mathbf{y} - \mathbf{S}(\boldsymbol{\lambda}_0 + \boldsymbol{\Lambda}\boldsymbol{\alpha}_k + \boldsymbol{\gamma})\big)^T \big(\mathbf{y} - \mathbf{S}(\boldsymbol{\lambda}_0 + \boldsymbol{\Lambda}\boldsymbol{\alpha}_k + \boldsymbol{\gamma})\big)\big] \Big\}^{z_k}, \tag{7.33}$$

and the likelihood of the parameters is given by :

$$L(\pi_k, \boldsymbol{\lambda}_0, \boldsymbol{\Lambda}, \boldsymbol{\alpha}_k, \boldsymbol{\Gamma}, \sigma^2 | \mathbf{y}_{1:N}, \mathbf{z}_{1:N}, \boldsymbol{\gamma}_{1:N}) = \prod_{i=1}^{N} \frac{1}{(2\pi)^{\frac{n_i+q}{2}} |\boldsymbol{\Gamma}|^{\frac{1}{2}}} \exp\big[-\frac{1}{2} \boldsymbol{\gamma}_i^T \boldsymbol{\Gamma}^{-1} \boldsymbol{\gamma}_i\big] \prod_{k=1}^{G} \Big\{ \pi_k \exp\big[-\frac{1}{2} n_i \log(\sigma^2)\big] \exp\big[-\frac{1}{2\sigma^2} \big(\mathbf{y}_i - \mathbf{S}_i(\boldsymbol{\lambda}_0 + \boldsymbol{\Lambda}\boldsymbol{\alpha}_k + \boldsymbol{\gamma}_i)\big)^T \big(\mathbf{y}_i - \mathbf{S}_i(\boldsymbol{\lambda}_0 + \boldsymbol{\Lambda}\boldsymbol{\alpha}_k + \boldsymbol{\gamma}_i)\big)\big] \Big\}^{z_{ki}}. \tag{7.34}$$

For parameter estimation by EM algorithm, we will use the log likelihood:

$$
\begin{aligned}
l(\pi_k, \boldsymbol{\lambda}_0, \boldsymbol{\Lambda}, \boldsymbol{\alpha}_k, \boldsymbol{\Gamma}, \sigma^2 | \mathbf{y}_{1:N}, \mathbf{z}_{1:N}, \boldsymbol{\gamma}_{1:N}) \quad &= \\
- \quad & \frac{1}{2} \sum_{i=1}^{N} (n_i + q) \log(2\pi) \\
+ \quad & \sum_{i=1}^{N} \sum_{k=1}^{G} z_{ki} \log(\pi_k) \qquad (7.35) \\
- \quad & \frac{1}{2} \sum_{i=1}^{N} \left[\log(|\boldsymbol{\Gamma}|) + \boldsymbol{\gamma}_i^T \boldsymbol{\Gamma}^{-1} \boldsymbol{\gamma}_i \right] \qquad (7.36) \\
- \quad & \frac{1}{2} \sum_{i=1}^{N} \sum_{k=1}^{G} z_{ki} \Big[n_i \log(\sigma^2) + \frac{1}{\sigma^2} \left\| \mathbf{y}_i - \mathbf{S}_i(\boldsymbol{\lambda}_0 + \boldsymbol{\Lambda}\boldsymbol{\alpha}_k + \boldsymbol{\gamma}_i) \right\|^2 \Big] \qquad (7.37)
\end{aligned}
$$

Algorithm 7.3.1 (Learning - Functional Clustering EM algorithm) .

The EM algorithm consists in iteratively maximizing the expected values of (7.35)*,* (7.36) *and* (7.37) *given* $\mathbf{y}_i$ *and the current parameters estimates. As these three parts involve separate parameters, we can optimize them separately.*
Initialization

First, we must initialize all parameters:

$$\{\boldsymbol{\lambda}_0, \boldsymbol{\Lambda}, \boldsymbol{\alpha}, \boldsymbol{\Gamma}, \sigma^2, z_{ik}, \pi_k, \boldsymbol{\gamma}_i\}.$$

E-step

The E-step consists in :

$$\widehat{\boldsymbol{\gamma}}_i = E\Big\{ \boldsymbol{\gamma}_i | \boldsymbol{y}_i, \boldsymbol{\lambda}_0, \boldsymbol{\Lambda}, \boldsymbol{\alpha}, \boldsymbol{\Gamma}, \sigma^2, z_{ik} \Big\} . \qquad (7.38)$$

with:

$$\widehat{\boldsymbol{\gamma}}_i = \left(\mathbf{S}_i^T \mathbf{S}_i + \sigma^2 \boldsymbol{\Gamma}^{-1} \right)^{-1} \mathbf{S}_i^T \left(\mathbf{y}_i - \mathbf{S}_i \boldsymbol{\lambda}_0 - \mathbf{S}_i \boldsymbol{\Lambda} \boldsymbol{\alpha}_k \right) . \qquad (7.39)$$

M-step

The ***M-step*** *involves maximizing :*

$$Q = E\Big\{ l(\pi_k, \boldsymbol{\lambda}_0, \boldsymbol{\Lambda}, \boldsymbol{\alpha}_k, \boldsymbol{\Gamma}, \sigma^2 | \boldsymbol{y}_{1:N}, \boldsymbol{z}_{1:N}, \boldsymbol{\gamma}_{1:N}) \Big\} , \qquad (7.40)$$

holding $\boldsymbol{\gamma}_{1:N}$ *fixed, and given by the E-step.*

The steps of this algorithm are detailed in Appendix: [H.1.1].

7.3.8 Low dimensional representation of curves

As the q-dimensional spline basis vector $\mathbf{s}(t)$ and covariance matrix $\boldsymbol{\Lambda}$ are the same for all clusters, centroids are represented in a reduced space by h-dimensional spline coefficients $\boldsymbol{\alpha}_k$. Thus, we can visualize these centroids in that low-dimensional space. With $h = 1$, centroids are represented by a point on a line, and with $h = 2$, they are represented by a point in a plane.

We can also represent curves by a point into a reduced h-dimensional subspace.

First, we project each curve $\mathbf{y}_i$ onto the q-dimensional spline space to give $\boldsymbol{\eta}_i$, afterward we project $\boldsymbol{\eta}_i$ into a reduced h-dimensional subspace to get $\boldsymbol{\alpha}_i$.

We have the functional model for a curve $\mathbf{y}_i$:

$$\mathbf{y}_i = \mathbf{S}_i(\boldsymbol{\eta}_i) + \boldsymbol{\epsilon}_i \ , \tag{7.41}$$

$$\mathbf{y}_i = \mathbf{S}_i(\boldsymbol{\lambda}_0 + \boldsymbol{\Lambda}\boldsymbol{\alpha}_{ki} + \boldsymbol{\gamma}_i) + \boldsymbol{\epsilon}_i \ , \tag{7.42}$$

and the covariance matrix :

$$\boldsymbol{\Sigma}_\mathbf{i} = \sigma^2\mathbf{I} + \mathbf{S_i}\boldsymbol{\Gamma}{\mathbf{S_i}}^T \ . \tag{7.43}$$

The curve $\mathbf{y}_i$ is projected onto the q-dimensional spline space to get $\widehat{\boldsymbol{\eta}}_i$; by Generalized Least Squares (GLS) we obtain :

$$\widehat{\boldsymbol{\eta}}_i = (\mathbf{S}_i^T\boldsymbol{\Sigma}_i^{-1}\mathbf{S}_i)^{-1}\mathbf{S}_i^T\boldsymbol{\Sigma}_i^{-1}\mathbf{y}_i \ . \tag{7.44}$$

Then, we have :

$$\mathbf{S}_i\big(\widehat{\boldsymbol{\eta}}_i - \boldsymbol{\lambda}_0\big) = \mathbf{S}_i\boldsymbol{\Lambda}\boldsymbol{\alpha}_i + \mathbf{S}_i\boldsymbol{\gamma}_i \ . \tag{7.45}$$

Next, $\widehat{\boldsymbol{\eta}}_i$ is projected onto the h-dimensional reduced space spanned by the means $\boldsymbol{\mu_k}$ to get $\boldsymbol{\lambda_0} - \boldsymbol{\Lambda}\widehat{\boldsymbol{\alpha}}_i$, by Generalized Least Squares (GLS) and we obtain :

$$\widehat{\boldsymbol{\alpha}}_i = \big(\boldsymbol{\Lambda}^T\mathbf{S}_i^T\boldsymbol{\Sigma}_i^{-1}\mathbf{S}_i\boldsymbol{\Lambda}\big)^{-1}\boldsymbol{\Lambda}^T\mathbf{S}_i^T\boldsymbol{\Sigma}_i^{-1}\mathbf{S}_i(\widehat{\boldsymbol{\eta}}_i - \boldsymbol{\lambda}_0) \ . \tag{7.46}$$

Thus, $\widehat{\boldsymbol{\alpha}}_i$ is a h-dimensional projection curve $\mathbf{y}_i$ onto the centroid space after centering.

If we chose

$$cov(\widehat{\boldsymbol{\alpha}}_i) = \left(\boldsymbol{\Lambda}^T \mathbf{S}_i^T \boldsymbol{\Sigma}_i^{-1} \mathbf{S}_i \boldsymbol{\Lambda}\right)^{-1} = \mathbf{I} \tag{7.47}$$

the transformed variables all have identity covariance, so that the distance between curves is Euclidean and can be easily estimated by visual inspection.

Moreover, we use the Eq.(7.47) as a constraint to get a non-confounded estimation of $\boldsymbol{\lambda}_0$, $\boldsymbol{\alpha}_k$, and $\boldsymbol{\Lambda}$ in Eq. (7.20).

7.4 Generalization - Classification of a new sample

We also have to realize the classification of a new sample into the best class given by the functional clustering.

The curve $\left(\mathbf{y}_i^{(k)}\right)$ will be classified to class k that minimizes:

$$k = \arg\min_{(k)} \left(\|\mathbf{y}_i^{(k)} - \mathbf{S}_i^{(k)} \boldsymbol{\lambda}_0 - \mathbf{S}_i^{(k)} \boldsymbol{\Lambda} \boldsymbol{\alpha}^{(k)}\|^2_{\boldsymbol{\Sigma}_i^{-1}} - 2 \log \pi_k \right) . \tag{7.48}$$

The steps of this algorithm are detailed in Appendix: [H.2.1].

8

Bayesian Filtering

This chapter gives a description of Bayes' Filters for Functional Data in Discrete Time.

This chapter gives a description of Bayes' Filters for Functional Data in Discrete Time, used in training and generalization of *Model of Trading* in discrete time describes in Chapter [5.2].

8.1 Introduction

In chapter [5] we built a "Model of Trading" in Discrete time for Stopping Time prediction of an asset, without Stop-Loss. In section [5.2] we described the procedure of forecasting with Particle and Kalman filters.

The goal of this chapter is to introduce the general concepts involved in Bayesian estimation for learning and generalization of non stationary, nonlinear systems in a Dynamic State Space representation with Particle and Kalman filters for parameter estimation and forecasting in an auto-adaptive procedure.

This chapter is organized as follow:

- in section [8.2] we introduce the Bayesian estimation,
- in section [8.3] we describe the Gaussian Bayesian estimation with Kalman filters,
- in section [8.4] we describe the Non-Gaussian Bayesian estimation with Particle filters,
- in section [8.5] we describe the Adaptive Particle filters,
- in section [8.6] we describe the Distributed Processing for filters on parallel computers,
- in section [8.7] we introduce the Rao-Blackwell Particle filters, when we can evaluate some filtering equations analytically,
- in section [8.8] we introduce the Stochastic Differential Equations,

- in section [8.9] we introduce the concepts of convergence, asymptotic results and robustness of filters.

8.1.1 Mathematical Preliminaries

Definition 8.1.1 (Probability space) .
A probability space is defined by the elements of $\{\Omega, \mathcal{F}, P\}$ where $\mathcal{F}$ is a σ-algebra of Ω and P is a complete, σ-additive probability measure on all $\mathcal{F}$. In other words, P is a set function whose arguments are random events (element of $\mathcal{F}$) such that axioms of probability hold.

Definition 8.1.2 (σ-algebra) .
Let $\mathcal{S}$ be a set and $\mathcal{F}$ be a family of subsets of $\mathcal{S}$. $\mathcal{F}$ is a σ-algebra if

- $\emptyset \in \mathcal{F}$
- *$A \in \mathcal{F}$ implies $A^c \in \mathcal{F}$*
- *$A_1, A_2, \cdots \in \mathcal{F}$ implies $\bigcup_{i=1}^{\inf} A_i \in \mathcal{F}$*

Definition 8.1.3 .
Let $p(\mathbf{x}) = \frac{dP(\mathbf{x})}{d\mu}$ denote Radon-Nikodym density of probability distribution $P(\mathbf{x})$ w.r.t. a measure $d\mu$.

- *when $\mathbf{x} \in X$ is discrete and μ is a counting measure, $p(\mathbf{x})$ is a probability mass function (pmf);*
- *when $\mathbf{x}$ is continuous and μ is a Lebesgue measure, $p(\mathbf{x})$ is a probability density function (pdf).*

Intuitively, the true distribution $P(\mathbf{x})$ can be replaced by the empirical distribution given the simulated samples $\mathbf{x}_{(i)}$,

$$\widehat{P}(\mathbf{x}) = \frac{1}{N} \sum_{i=1}^{N} \delta_0(\mathbf{x} - \mathbf{x}^{(i)}) \tag{8.1}$$

where $\delta_0(.)$ is a Radon-Nikodym density w.r.t. μ of the point-mass distribution concentrated at the point $\mathbf{x}$.

- when $\mathbf{x} \in X$ is discrete, $\delta_0(\mathbf{x} - \mathbf{x}^{(i)}$ is 1 for $\mathbf{x} = \mathbf{x}_{(i)}$ and 0 elsewhere,
- when $\mathbf{x} \in X$ is continuous, $\delta_0(\mathbf{x} - \mathbf{x}^{(i)})$ is a Dirac-delta function, and $\delta_0(\mathbf{x} - \mathbf{x}^{(i)}) = 0$ for all $\mathbf{x}_{(i)} \neq \mathbf{x}$, and $\int_X d\widehat{P}(\mathbf{x}) = \int_X \widehat{p}(\mathbf{x})\mathbf{dx} = 1$

Remark 8.1.1 .
For simplicity, we use $\mathbf{x}_t$ to denote both the random variable and its realization at time t. Consequently, we express continuous probability distribution using $p(d\mathbf{x}_t)$ instead of $Pr(\mathbf{X}_t \in d\mathbf{x}_t)$ and discrete distribution using $p(\mathbf{x}_t)$ instead of $Pr(\mathbf{X}_t = \mathbf{x}_t)$.
If these distributions admit densities with respect to an underlying mesure μ (counting or Lebesgue), we denote these densities by $p(\mathbf{x}_t)$.
With a slight abuse of terminology sometimes we refer to $p(\mathbf{x}_t)$ as a distribution.

8.1.2 Stochastic Filtering Problem

Before running with mathematical formulations, we have to clarify some basic concepts:

- *Filtering* : involves the extraction of information about a quantity of interest at time t by using data measured up to and including time t,
- *Prediction* : is an *a priori* form of estimation. We derive information about what a quantity of interest will be at some time $t+\tau$ in the future $(\tau > 0)$ by using data measured up to and including time t,
- *Smoothing* : is an *a posteriori* form of estimation in that the data measured after the time of interest are used for the estimation. Such as the estimation $\widehat{\mathbf{x}}_{t'}$ using data measured over the interval $[0, t]$, where $t^{'} < t$.

We will consider the following generic stochastic filtering problem in the more general Dynamic State Space form:

$$\frac{d\mathbf{x}_t}{dt} = \mathbf{f}(t, \mathbf{x}_t, \mathbf{u}_t, \mathbf{v}_t) \tag{8.2}$$

$$\frac{d\mathbf{y}_t}{dt} = \mathbf{h}(t, \mathbf{x}_t, \mathbf{n}_t) \tag{8.3}$$

where:

- Eq. (8.2) is the state equation,
- Eq. (8.3) is the measurement equation,
- $\mathbf{x}_t$ is the state vector,
- $\mathbf{y}_t$ is the measurement vector,
- $\mathbf{u}_t$ is the determinist input vector,
- $\mathbf{v}_t$ is the dynamic process noise vector,
- $\mathbf{n}_t$ is the measurement noise vector,
- $\mathbf{f} : \mathbb{R}^{N_x} \mapsto \mathbb{R}^{N_x}$ is a valued state function,
- $\mathbf{h} : \mathbb{R}^{N_x} \mapsto \mathbb{R}^{N_y}$ is a valued measurement function.

This formulation is in continuous-time domain, however in practice we are more concerned about discrete-time filtering, and we define the Discrete Dynamic State Space form:

$$\mathbf{x}_{k+1} = \mathbf{f}_k(\mathbf{x}_k, \mathbf{u}_k, \mathbf{v}_k) \tag{8.4}$$

$$\mathbf{y}_k = \mathbf{h}_k(\mathbf{x}_k, \mathbf{n}_k) \tag{8.5}$$

where:

- Eq. (8.4) characterizes the state transition probability $p(\mathbf{x}_{k+1}|\mathbf{x}_k)$,
- Eq. (8.5) characterizes the likelihood $p(\mathbf{y}_k|\mathbf{x}_k)$.

Given an initial density $p(\mathbf{x}_0)$, a transition density $p(\mathbf{x}_k|\mathbf{x}_{k-1})$, and a likelihood $p(\mathbf{y}_k|\mathbf{x}_k)$, the objective of the filtering is to estimate the optimal current state $\mathbf{x}_k$ at time k, given the observations up to time k, $\{\mathbf{y}_{1:k} = \mathbf{y}_1, \mathbf{y}_2, \cdots, \mathbf{y}_k\}$.

The posterior density $p(\mathbf{x}_k|\mathbf{y}_{1:k})$ would provide a complete solution to the stochastic filtering problem, but this problem still remains intractable since the density is a function rather than a finite-dimensional point estimate. Most of physical systems are not finite dimensional, thus the infinite-dimensional system can only be modeled approximately by a suboptimal finite-dimensional filter.

8.1.3 Nonlinear Stochastic Filtering

We use a set of measurements $\mathbf{y}_k$ at discrete time steps (hence $\mathbf{y}_{1:k}$), and provided the functions $\mathbf{f}_k$ and $\mathbf{h}_k$ are known, we would like to find the optimal or suboptimal $\widehat{\mathbf{x}}_k$.

This problem is an inverse mapping learning problem. We would like to find the input sequentially with a composite mapping function which yields the output data. This inversion learning problem is one-to-many, in a sense that the mapping from output to input space is generally non-unique.

A problem is said to be well-posed if it satisfied three conditions: existence, uniqueness and stability, otherwise it is said ill-posed [Chen and Haykin, 2002]. This stochastic filtering problem is ill-posed:

- The presence of the unknown noise corrupts the state and measurement equations; given limited noisy observations, the solution is non-unique,
- Supposing the state equation is differentiable and regular, the measurement function is possibly a many-to-one mapping function, which violates the uniqueness condition,
- The filtering problem is intrinsically a conditional posterior distribution density estimation problem, which is stochastically ill-posed especially in high-dimensional space [Vapnik, 1998].

8.2 Bayesian Estimation

8.2.1 Introduction

Suppose we have data $\mathbf{x} = \big(x_1, x_2, \cdots, x_n\big)$ with distribution $p(\mathbf{x}|\boldsymbol{\theta})$ where $\boldsymbol{\theta}$ is the unknown parameter we want to estimate. The basic idea of the Bayesian approach is to treat the parameter $\boldsymbol{\theta}$ as a random variable and to use an *a priori* knowledge of the distribution $\pi(\boldsymbol{\theta})$ of $\boldsymbol{\theta}$ is estimated by calculating the *a posteriori* distribution $\pi(\boldsymbol{\theta}|\mathbf{x})$ of $\boldsymbol{\theta}$.

The one-dimensional 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

$$\pi(\theta|x) = \frac{p(x|\theta)\ \pi(\theta)}{\int\ p(x|\theta)\ \pi(\theta)\ d\theta}\ , \tag{8.6}$$

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 $\hat{\theta} = \mathbb{E}_{\pi(\theta|x)}\big[\theta\big]$.

The multi-dimensional case

In the multi-dimensional case $\boldsymbol{\theta} = (\theta_1, \theta_2, \cdots, \theta_k)$; the a posteriori distribution of $\boldsymbol{\theta}$ can be calculated by the Bayes' formula as follows :

$$\pi(\boldsymbol{\theta}|\mathbf{x}) = \frac{p(\mathbf{x}|\boldsymbol{\theta})\ \pi(\boldsymbol{\theta})}{\int\cdots\int\ p(\boldsymbol{\theta}|\mathbf{x})\ \pi(\boldsymbol{\theta})\ d\theta_1\cdots\theta_k}\ . \tag{8.7}$$

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

$$\pi(\theta_i|\mathbf{x}) = \int\cdots\int\ \pi(\boldsymbol{\theta}|\mathbf{x})\ d\theta_1\cdots d\theta_{i-1}\ d\theta_{i+1}\cdots d\theta_k\ , \tag{8.8}$$

we are able to estimate $\boldsymbol{\theta}$ by the ways described in the one-dimensional case. Usually problems arise in calculating the integrals in Equation [8.8] which require approximation techniques as *Markov Chain Monte Carlo* methods (MCMC).

8.2.2 Markov Chain Monte Carlo methods (MCMC)

Suppose we want to generate a sample from an a posteriori distribution $\pi(\boldsymbol{\theta}|\mathbf{x})$ for $\boldsymbol{\theta} \in \boldsymbol{\Theta} \subseteq R^k$ but we cannot directly do this. However, suppose we are able to construct a Markov chain with state space $\boldsymbol{\Theta}$ and with distribution $\pi(\boldsymbol{\theta}|\mathbf{x})$. Then under suitable regularity conditions asymptotic results exist, showing in which case the sample output from such a chain with distribution $\pi(\boldsymbol{\theta}|\mathbf{x})$ can be used to mimic a random sample from $\pi(\boldsymbol{\theta}|\mathbf{x})$ or to estimate the expected value of a function $f(\boldsymbol{\theta})$ with respect to $\pi(\boldsymbol{\theta}|\mathbf{x})$.

If $\theta^1, \cdots, \theta^k, \cdots$ is a realization from a suitable chain then $\boldsymbol{\theta}^k \rightarrow \boldsymbol{\theta}$ in distribution as k tends to infinity, $\boldsymbol{\theta} \approx \pi(\boldsymbol{\theta}|\mathbf{x})$ and $\frac{1}{k}\sum_{i=1}^{k} f(\boldsymbol{\theta}^i) \rightarrow E_{\boldsymbol{\theta}|\mathbf{x}}\Big[f(\boldsymbol{\theta})\Big]$ a.s. as k tends to infinity.

8.2.3 Why do we use a Bayesian representation :

Many real-world applications require estimating unknown quantities from some given observations at each time step. Mostly applications, even though the dynamics of the system are not known exactly, a prior knowledge about the phenomenon being modeled is generally available to construct a suitable model.

In the Bayesian modeling, prior distributions for the states and likelihood functions relating these states to the observations are derived from the model.

Within this context, estimates of states are based on the posterior distribution obtained from *Bayes' 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 in 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.

8.2.4 Limitations of Kalman Filters :

If the data are modeled by a linear state-space model, and if initial state and noises are Gaussian, then the *Kalman Filter* [Kalman, 1960] is the optimal filter in order to minimize 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], and the 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.

8.2.5 Why do we use Particle filters :

Sequential Monte Carlo (SMC), or *Particle filters* [Doucet, et al., 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 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].

8.3 Gaussian Bayesian Estimation

8.3.1 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 state transition. 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 likelihood.

The evolution of the state sequence is given by the *Transition* equation :

$$\mathbf{x}_k = \mathbf{f}_k(\mathbf{x}_{k-1}, \mathbf{u}_k, \mathbf{v}_k; \mathbf{w}) \ , \tag{8.9}$$

and the *Measurement* equation is given by :

$$\mathbf{y}_k = \mathbf{h}_k(\mathbf{x}_k, \mathbf{n}_k; \mathbf{w}) \ , \tag{8.10}$$

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 driving 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 parameter vector.

Bayes' filters estimate the state $\mathbf{x}$ of a dynamic system from measurements $\mathbf{y}$ by a recursively estimate of the posterior probability density over the state space conditioned on the data collected.

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 $\mathbf{y}_{1:k-1} = \{\mathbf{y}_1, \mathbf{y}_2, \cdots, \mathbf{y}_{k-1}\}$

$$\widehat{\mathbf{x}}_{k|k-1} = \mathbb{E}\big[\mathbf{x}_k|\mathbf{y}_{1:k-1}\big] = \int \mathbf{x}_k \; p(\mathbf{x}_k|\mathbf{y}_{1:k-1}) \; d\mathbf{x}_k \; . \tag{8.11}$$

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

$$\widehat{\mathbf{x}}_{k|k} = \mathbb{E}\big[\mathbf{x}_k|\mathbf{y}_{1:k}\big] = \mathbb{E}\big[\mathbf{x}_k|\mathbf{y}_k\big] \; , \tag{8.12}$$

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, by the predictive step, in order to calculate *the prior* at time k. In terms of probability distributions, this means that the prior distribution is obtained by

$$p(\mathbf{x}_k|\mathbf{y}_{1:k-1}) = \int p(\mathbf{x}_k|\mathbf{x}_{k-1}, \mathbf{y}_{1:k-1}) \; p(\mathbf{x}_{k-1}|\mathbf{y}_{1:k-1}) \; d\mathbf{x}_{k-1} \tag{8.13a}$$

$$= \int p(\mathbf{x}_k|\mathbf{x}_{k-1}) \; p(\mathbf{x}_{k-1}|\mathbf{y}_{1:k-1}) \; d\mathbf{x}_{k-1} \; , \tag{8.13b}$$

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, by the updating step, 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 :

$$p(\mathbf{x}_k|\mathbf{y}_{1:k}) = \frac{p(\mathbf{y}_k|\mathbf{x}_k)\ p(\mathbf{x}_k|\mathbf{y}_{1:k-1})}{p(\mathbf{y}_k|\mathbf{y}_{1:k-1})} \tag{8.14}$$

with the evidence

$$p(\mathbf{y}_k|\mathbf{y}_{1:k-1}) = \int p(\mathbf{y}_k|\mathbf{x}_k)\ p(\mathbf{x}_k|\mathbf{y}_{1:k-1}) d\mathbf{x}_k\ , \tag{8.15}$$

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

$$p(\mathbf{x}_k|\mathbf{x}_{k-1}) = \int \delta_0\big(\mathbf{x}_k - \mathbf{f}(\mathbf{x}_{k-1}, \mathbf{u}_k, \mathbf{v}_k; \mathbf{w})\big)\ p(\mathbf{v}_k)\ d\mathbf{v}_k \tag{8.16}$$

$$p(\mathbf{y}_k|\mathbf{x}_k) = \int \delta_0\big(\mathbf{y}_k - \mathbf{h}(\mathbf{x}_k, \mathbf{n}_k; \mathbf{w})\big)\ p(\mathbf{n}_k)\ d\mathbf{n}_k\ , \tag{8.17}$$

When $\mathbf{x} \in X$ is discrete, $\delta_0(\mathbf{x} - \widehat{\mathbf{x}}) = 1$ for $\mathbf{x} = \widehat{\mathbf{x}}$ and 0 elsewhere. When $\mathbf{x} \in X$ is continuous, $\delta(.)$ is the Dirac function with $\delta_0(\mathbf{x} - \widehat{\mathbf{x}}) = 0$ for all $\mathbf{x} \neq \widehat{\mathbf{x}}$, and $\int_X dP(\mathbf{x}) = \int_X p(\mathbf{x}) d\mathbf{x} = 1$.

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.

Bayes' filters are an abstract concept in that they only provide a probabilistic framework for recursive state estimation. To implement Bayes' filters, we only have to specify the likelihood $p(\mathbf{y}_k|\mathbf{x}_k)$, the dynamics $p(\mathbf{x}_k|\mathbf{x}_{k-1})$, and the representation for the posterior $p(\mathbf{x}_k|\mathbf{y}_{1:k})$.

8.3.2 Filtering

The filtering is a real time procedure, and it consists in two steps.

- At time $k-1$ we know the posterior distribution $p(\widehat{\mathbf{x}}_{k-1|k-1}|\mathbf{y}_{1:k-1})$ of the estimated state $\widehat{\mathbf{x}}_{k-1|k-1}$, given all past observations $\mathbf{y}_{1:k-1}$,

- by the *prediction step* we can estimate the prior distribution $p(\widehat{\mathbf{x}}_{k|k-1}|\mathbf{y}_{1:k-1})$ of the predicted state at time k, $\widehat{\mathbf{x}}_{k|k-1}$, given all past observations $\mathbf{y}_{1:k-1}$,
- at time k we observe $\mathbf{y}_k$,
- by the *updating step* we can obtain the new posterior distribution $p(\widehat{\mathbf{x}}_{k|k}|\mathbf{y}_{1:k})$ of the estimated state $\widehat{\mathbf{x}}_{k|k}$, given all observations $\mathbf{y}_{1:k}$.

This procedure is iterated at each time with the new observations (see Fig [8.1]).

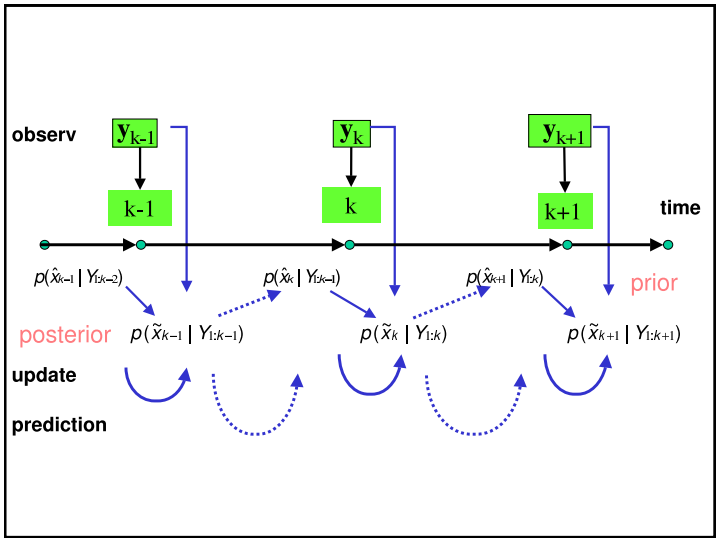

Figure 8.1: The Filtering consists in two steps, a prediction step and an update step. At time $k-1$ we know the posterior distribution $p(\widehat{\mathbf{x}}_{k-1|k-1}|\mathbf{y}_{1:k-1})$, and by the prediction step we estimate the prior distribution $p(\widehat{\mathbf{x}}_{k|k-1}|\mathbf{y}_{1:k-1})$. At time k we observe $\mathbf{y}_k$ and by the update step we can obtain the new posterior distribution $p(\widehat{\mathbf{x}}_{k|k}|\mathbf{y}_{1:k})$.

8.3.3 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 (distributions symmetric and unimodal);
- 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
$\widehat{\mathbf{x}}_{k|k} = E[\mathbf{x}_{k|k}|\mathbf{y}_{1:k}]$ and its covariance $\mathbf{P}_{\widehat{\mathbf{x}}_{k|k}}$:

$$\begin{aligned}\widehat{\mathbf{x}}_{k|k} &= (\text{prediction of } \mathbf{x}_k) + \mathbf{K}_k(\mathbf{y}_k - (\text{prediction of } \mathbf{y}_k)) \\ &= \widehat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k(\mathbf{y}_k - \widehat{\mathbf{y}}_{k|k-1}) \qquad (8.18)\\ \mathbf{P}_{\widehat{\mathbf{x}}_{k|k}} &= \mathbf{P}_{\widehat{\mathbf{x}}_{k|k-1}} - \mathbf{K}_k\mathbf{P}_{\widetilde{\mathbf{y}}_k}\mathbf{K}_k^T \ . \qquad (8.19)\end{aligned}$$

The optimal terms in this recursion are given by :

$$\begin{aligned}\widehat{\mathbf{x}}_{k|k-1} &= \mathbb{E}\big[\mathbf{f}_k(\mathbf{x}_{k-1}, \mathbf{v}_{k-1}, \mathbf{u}_k)\big] \qquad (8.20)\\ \widehat{\mathbf{y}}_{k|k-1} &= \mathbb{E}\big[\mathbf{h}_k(\widehat{\mathbf{x}}_{k|k-1}, \mathbf{n}_k)\big] \qquad (8.21)\\ \mathbf{K}_k &= \mathbb{E}\big[(\mathbf{x}_k - \widehat{\mathbf{x}}_{k|k-1})(\mathbf{x}_k - \widehat{\mathbf{x}}_{k|k-1})^T\big]\mathbb{E}\big[(\mathbf{y}_k - \widehat{\mathbf{y}}_{k|k-1})(\mathbf{y}_k - \widehat{\mathbf{y}}_{k|k-1})^T\big]^{-1} \\ &= \mathbf{P}_{\widehat{\mathbf{x}}_{k|k}\widetilde{\mathbf{y}}_{k|k-1}}\mathbf{P}^{-1}_{\widetilde{\mathbf{y}}_{k|k-1}} \ , \qquad (8.22)\end{aligned}$$

where the optimal prediction $\widehat{\mathbf{x}}_{k|k-1}$ corresponds to the expectation of a nonlinear function $\mathbf{f}_k$ of the random function variables $\mathbf{x}_{k-1}$ and $\mathbf{v}_k$, and the optimal prediction $\widehat{\mathbf{y}}_{k|k-1}$ corresponds to the expectation of a nonlinear function $\mathbf{h}_k$ 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 $\mathbf{K}_k$ 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.

8.3.4 Kalman Filter (KF)

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, $p(\mathbf{x}_k|\mathbf{y}_{1:k})$ will be Gaussian if :

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

State space representation

We have the system :

$$\mathbf{x}_{k+1} = \mathbf{A}_k\mathbf{x}_k + \mathbf{B}_k\mathbf{u}_k + \mathbf{G}_k\mathbf{v}_k \tag{8.23}$$
$$\mathbf{y}_k = \mathbf{C}_k\mathbf{x}_k + \mathbf{n}_k \ , \tag{8.24}$$

where :

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

Algorithm 8.3.1 (Linear Kalman Filter (KF)) .

Initialization

$$\widehat{\mathbf{x}}_0 = E\{\mathbf{x}_0\} = \overline{\mathbf{x}}_0 \tag{8.25a}$$
$$\mathbf{P}_0 = E[(\mathbf{x}_0 - \widehat{\mathbf{x}}_0)(\mathbf{x}_0 - \widehat{\mathbf{x}}_0)^T] \tag{8.25b}$$
$$\mathbf{R} = E[(\mathbf{v} - \overline{\mathbf{v}})(\mathbf{v} - \overline{\mathbf{v}})^T] \tag{8.25c}$$
$$\mathbf{Q} = E[(\mathbf{n} - \overline{\mathbf{n}})(\mathbf{n} - \overline{\mathbf{n}})^T] \tag{8.25d}$$

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

Prediction step
Compute the predicted state mean and covariance (time update)

$$\widehat{\mathbf{x}}_{k|k-1} = \mathbf{A}_{k-1}\widehat{\mathbf{x}}_{k-1|k-1} + \mathbf{B}_{k-1}\mathbf{u}_k + \mathbf{G}_k\overline{\mathbf{v}} \tag{8.26a}$$
$$\mathbf{P}_{k|k-1} = \mathbf{A}_{k-1}\mathbf{P}_{k-1|k-1}\mathbf{A}_{k-1}^T + \mathbf{G}_{k-1}\mathbf{Q}_{k-1}\mathbf{G}_{k-1}^T \tag{8.26b}$$

Correction step
Update estimates with latest observation (measurement update)

$$\boldsymbol{\Sigma}_{k|k-1} = \mathbf{C}_k \mathbf{P}_{k|k-1} \mathbf{C}_k^T + \mathbf{R}_k \tag{8.27a}$$
$$\mathbf{K}_k = \mathbf{P}_{k|k-1} \mathbf{C}_k^T \boldsymbol{\Sigma}_{k|k-1}^{-1} \tag{8.27b}$$
$$\mathbf{e}_k = \mathbf{y}_k - \mathbf{C}_k \widehat{\mathbf{x}}_{k|k-1} - \mathbf{D}_k \overline{\mathbf{n}} \tag{8.27c}$$
$$\widehat{\mathbf{x}}_{k|k} = \widehat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k \mathbf{e}_k \tag{8.27d}$$
$$\mathbf{P}_{k|k} = \mathbf{P}_{k|k-1} - \mathbf{K}_k \mathbf{C}_k \mathbf{P}_{k|k-1} \tag{8.27e}$$

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].

8.3.5 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 linearization, the equations for the Linear Kalman filter can be used.

State space representation

We have the system :

$$\mathbf{x}_k = \mathbf{f}_k(\mathbf{x}_{k-1}, \mathbf{u}_k, \mathbf{v}_k) \tag{8.28}$$
$$\mathbf{y}_k = \mathbf{h}_k(\mathbf{x}_k, \mathbf{n}_k) \tag{8.29}$$

where :

- $\mathbf{x}_k \in R^n$, $\mathbf{y}_k \in R^q$, $\mathbf{u}_k \in R^m$, $\mathbf{v}_k \in R^p$, $\mathbf{n}_k \in R^q$,
- $\mathbf{f}_k$ is the state equation, $\mathbf{h}_k$ is the measurement equation, these parametric functions may be time-variant but are known,
- $\{\mathbf{u}_k\}$ is a deterministic sequence,
- $\{\mathbf{v}_k\}$, $\{\mathbf{n}_k\}$ are process and measurement noise sequences; with $\mathbf{v}_k \sim \mathcal{N}(0, \mathbf{Q})$, $\mathbf{n}_k \sim \mathcal{N}(0, \mathbf{R})$,
- $\mathbf{x}_0 \sim \mathcal{N}(\overline{\mathbf{x}_0}, \mathbf{P}_0)$.

For nonlinear state space models, the Extended Kalman Filter linearizes the state equation around the current state estimate $\widehat{\mathbf{x}}_{k-1|k-1}$ and the measurement equation around the estimate $\widehat{\mathbf{x}}_{k|k-1}$ using a first-order Taylor series expansion.

In the linearized state equation, we use the Jacobian matrix of the state transition equation $\mathbf{f}_k$ evaluated around $\widehat{\mathbf{x}}_{k-1|k-1}$ and in the linearized measurement equation, we use the Jacobian matrix of the measurement equation $\mathbf{h}_k$ evaluated around $\widehat{\mathbf{x}}_{k|k-1}$.

Clearly these approximations will only valid if all the higher order derivatives of the nonlinear functions are effectively zero over the uncertainty 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 underestimated.

We can also use a second order Taylor series expansion, but in this case we need Jacobian and Hessian matrices, and the algorithms are more complicated, without really giving better results [Anderson, and More, 1979]; [Chui and Chen, 1990].

Algorithm 8.3.2 (Extended Kalman Filter (EKF)) .

Initialization

$$\widehat{\mathbf{x}}_0 = E\{\mathbf{x}_0\} = \overline{\mathbf{x}}_0 \tag{8.30a}$$
$$\mathbf{P}_0 = E[(\mathbf{x}_0 - \widehat{\mathbf{x}}_0)(\mathbf{x}_0 - \widehat{\mathbf{x}}_0)^T] \tag{8.30b}$$
$$\mathbf{R} = E[(\mathbf{v} - \overline{\mathbf{v}})(\mathbf{v} - \overline{\mathbf{v}})^T] \tag{8.30c}$$
$$\mathbf{Q} = E[(\mathbf{n} - \overline{\mathbf{n}})(\mathbf{n} - \overline{\mathbf{n}})^T] \tag{8.30d}$$

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

Prediction step
Compute the process model Jacobians :

$$\mathbf{F}_k = \nabla_{\mathbf{x}} \mathbf{f}_k(\mathbf{x}, \overline{\mathbf{v}}, \mathbf{u}_k)|_{\mathbf{x}=\widehat{\mathbf{x}}_{k-1|k-1}} \tag{8.31a}$$
$$\mathbf{G}_k = \nabla_{\mathbf{v}} \mathbf{f}_k(\widehat{\mathbf{x}}_{\mathbf{k-1}}, \mathbf{v}, \mathbf{u}_k)|_{\mathbf{v}=\widehat{\mathbf{v}}} \tag{8.31b}$$

Compute the predicted state mean and covariance (time update)

$$\widehat{\mathbf{x}}_{k|k-1} = \mathbf{f}_k(\widehat{\mathbf{x}}_{k|k-1}, \overline{\mathbf{v}}, \mathbf{u}_k) \tag{8.32a}$$
$$\mathbf{P}_{k|k-1} = \mathbf{F}_k \mathbf{P}_{k-1|k-1} \mathbf{F}_k^T + \mathbf{G}_k \mathbf{R} \mathbf{G}_k^T \tag{8.32b}$$

Correction step
Compute the observation model Jacobians :

$$\mathbf{H}_k = \nabla \mathbf{h}_k(\mathbf{x}, \mathbf{n})|_{\mathbf{x}=\widehat{\mathbf{x}}_{k|k-1}} \tag{8.33a}$$
$$\mathbf{D}_k = \nabla \mathbf{h}_k(\mathbf{x}_{k|k-1}, \mathbf{n})|_{\mathbf{n}=\overline{\mathbf{n}}} \tag{8.33b}$$

Update estimates with latest observation (measurement update)

$$\mathbf{\Sigma}_{k|k-1} = \mathbf{H}_k \mathbf{P}_{k|k-1} \mathbf{H}_k^T + \mathbf{D}_k \mathbf{R} \mathbf{D}_k^T \tag{8.34a}$$
$$\mathbf{K}_k = \mathbf{P}_{k|k-1} \mathbf{H}_k^T \mathbf{\Sigma}_{k|k-1}^{-1} \tag{8.34b}$$
$$\mathbf{e}_k = \mathbf{y}_k - \mathbf{h}_k(\widehat{\mathbf{x}}_{k|k-1}, \overline{\mathbf{n}}) \tag{8.34c}$$
$$\widehat{\mathbf{x}}_{k|k} = \widehat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k \mathbf{e}_k \tag{8.34d}$$
$$\mathbf{P}_{k|k} = \mathbf{P}_{k|k-1} - \mathbf{K}_k \mathbf{H}_k \mathbf{P}_{k|k-1} \tag{8.34e}$$

8.3.6 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 nonlinear function*".

It also approximates the posterior density $p(\mathbf{x}_k|\mathbf{y}_{1:k})$ with a Gaussian, but compared to EKF, which linearizes the equations using a first-order Taylor series expansion, and Jacobians matrices, UKF approximates the distribution of the state variable by using an unscented transformation (UT) [Julier, 1997]. UKF will provide a more accurate mean than EKF while giving the same accuracy for covariance.

In general, the state space must be augmented by concatenating the state and the system and measurement noises and will have a new dimension $L = L_x + L_v + L_n$ with L_x, L_v, and L_n are the dimensions of the state $\mathbf{x}$, the process noise $\mathbf{v}$ and the measurement noise $\mathbf{n}$ respectively.

However this is not necessary when the equations are linear in noise, such as:

$$\mathbf{x}_k = \mathbf{f}(\mathbf{x}_{k-1}) + \mathbf{v}_k \tag{8.35}$$
$$\mathbf{y}_k = \mathbf{h}(\mathbf{x}_k) + \mathbf{n}_k. \tag{8.36}$$

An explicit description of UKF can be found in [Doucet, 1998]; [Doucet, et al., 2001]; [van der Merwe , 2004]; [Wan and van der Merwe , 2000]; and [Wan and van der Merwe , 2001].

For convenience, we describe the main steps of the algorithm below.

State space representation

We have the system :

$$\mathbf{x}_k = \mathbf{f}_k(\mathbf{x}_{k-1}, \mathbf{u}_k, \mathbf{v}_k) \tag{8.37}$$
$$\mathbf{y}_k = \mathbf{h}_k(\mathbf{x}_k, \mathbf{n}_k) \tag{8.38}$$

where :

- $\mathbf{x}_k \in R^n$, $\mathbf{y}_k \in R^q$, $\mathbf{u}_k \in R^m$, $\mathbf{v}_k \in R^p$, $\mathbf{n}_k \in R^q$,
- $\mathbf{f}_k$ is the state equation, $\mathbf{h}_k$ is the measurement equation, these parametric functions may be time-variant but are known,
- $\{\mathbf{u}_k\}$ is a deterministic sequence,
- $\{\mathbf{v}_k\}$, $\{\mathbf{n}_k\}$ are process and measurement noise sequences; with $\mathbf{v}_k \sim \mathcal{N}(0, \mathbf{Q})$, $\mathbf{n}_k \sim \mathcal{N}(0, \mathbf{R})$,
- $\mathbf{x}_0 \sim \mathcal{N}(\overline{\mathbf{x}_0}, \mathbf{P}_0)$.

implementing the Unscented Kalman filter (UKF)

The method presented in this section is based on [Wan and van der Merwe , 2000], and [Wan and van der Merwe , 2001].

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

$$\mathbf{x}_k^a = \begin{pmatrix} \mathbf{x}_k^x \\ \mathbf{x}_k^v \\ \mathbf{x}_k^n \end{pmatrix} = \begin{pmatrix} \mathbf{x}_k \\ \mathbf{v}_k \\ \mathbf{n}_k \end{pmatrix} \tag{8.39}$$

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

$$\mathbf{P}^a = \begin{pmatrix} \mathbf{P_x} & 0 & 0 \\ 0 & \mathbf{R_v} & 0 \\ 0 & 0 & \mathbf{R_n} \end{pmatrix} \tag{8.40}$$

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 :

$$\boldsymbol{\mathcal{X}_0} = \overline{\mathbf{x}}, \tag{8.41a}$$

$$\boldsymbol{\mathcal{X}_i} = \overline{\mathbf{x}} + \zeta(\sqrt{\mathbf{P}_{\mathbf{x}}})_i, \quad i = 1, \cdots, L\ , \tag{8.41b}$$

$$\boldsymbol{\mathcal{X}_i} = \overline{\mathbf{x}} - \zeta(\sqrt{\mathbf{P}_{\mathbf{x}}})_{i-L}, \quad i = L+1, \cdots, 2L\ , \tag{8.41c}$$

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

$$\boldsymbol{\mathcal{Y}_i} = \mathbf{g}(\boldsymbol{\mathcal{X}}_i), \quad i = 0, \cdots, 2L \tag{8.42}$$

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

$$\overline{\mathbf{y}} \approx \sum_{i=0}^{2L} w_i^{(m)}\ \boldsymbol{\mathcal{Y}}_i \tag{8.43}$$

$$\mathbf{P}_{\mathbf{y}} \approx \sum_{i=0}^{2L} w_i^{(c)}\ (\boldsymbol{\mathcal{Y}}_i - \overline{\mathbf{y}})\ (\boldsymbol{\mathcal{Y}}_i - \overline{\mathbf{y}})^T \tag{8.44}$$

$$\mathbf{P}_{\mathbf{xy}} \approx \sum_{i=0}^{2L} w_i^{(c)}\ (\boldsymbol{\mathcal{X}}_i^x - \overline{\mathbf{x}})\ (\boldsymbol{\mathcal{Y}}_i - \overline{\mathbf{y}})^T \tag{8.45}$$

where $w_i^{(m)}$ and $w_i^{(c)}$ are scalar weights given by :

$$w_o^{(m)} = \frac{\lambda}{L+\lambda} \tag{8.46}$$

$$w_0^{(c)} = \frac{\lambda}{L+\lambda} + (1 - \alpha^2 + \beta) \tag{8.47}$$

$$w_i^{(m)} = w_i^{(c)} = \frac{1}{2(L+\lambda)} \quad \text{for } i = 1, \cdots, 2L \tag{8.48}$$

where :

- α controls the size of the sigma-point distribution, $0 \leq \alpha \leq 1$,
- β is a non-negative weighting term, $\beta \geq 0$,
- $\lambda \geq 0$ to guarantee positive semi-definiteness of the covariance matrix.

This algorithm is described in Appendix: [H.5.1].

8.4 Non-Gaussian Bayesian Estimation

8.4.1 Introduction

The Kalman filter (KF), Extended Kalman filter (EKF) and Unscented Kalman filter (UKF), still assume a Gaussian posterior which can fail in certain non-linear 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 a specification of the prior distribution of the state, the transition distribution and the likelihood. Essentially, this means particle filtering is Bayesian in nature.

8.4.2 Particles Filters (PF)

We have the system :

$$\mathbf{x}_k = \mathbf{f}_{k-1}\big(\mathbf{x}_k, \mathbf{u}_k, \mathbf{v}_k\big) \tag{8.49}$$

$$\mathbf{y}_k = \mathbf{h}_k\big(\mathbf{x}_k, \mathbf{n}_k\big)\ . \tag{8.50}$$

The Particle filter algorithm consists in four steps "*Initialization*", "*Prediction*", *Updating*" and "*Resampling*".

During the *initialization* step, we sample N_s times from the initial distribution $p(\mathbf{x}_0)$. By saying that we sample $\{\mathbf{x}^{(i)}\}$ from a distribution π, for $i = 1, \cdots, N_s$ we mean that we simulate N_s independent random samples, named particles, according to π. Hence, the N_s random variables $\{\mathbf{x}^{(i)}\}$ for $i = 1, \cdots, N_s$ are independent and identical distributed (i.i.d.) according to π.

In the "*Predictive*" step 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 $g_k(\mathbf{y}_k - \mathbf{h}_k(.))$, 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.

8.4.3 Sampling Importance Resampling (SIR)

The particle filter theory presented in this section is inspired by [Doucet, et al., 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 a 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 density, and the SIS filter approaches the optimal Bayesian estimate.

The working mechanism of particle filters is the following. The state space is partitioned in 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 time according to the state equation. Since the density can be approximated by the point-mass histogram, by random sampling of the state space, we get a number of particles representing the evolving density. However, since the posterior density model is unknown or hard to sample, we would rather choose another distribution, with the same support, 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 characterizes the posterior density $p(\mathbf{x}_{0:k}|\mathbf{y}_{1:k})$, 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 normalized such that $\sum_i w_k^{(i)} = 1$. Then, the posterior density at k can be approximated as

$$p(\mathbf{x}_{0:k}|\mathbf{y}_{1:k}) \approx \widehat{p}(\mathbf{x}_{0:k}|\mathbf{y}_{1:k}) = \sum_{i=1}^{N_s} w_k^{(i)}\ \delta_0(\mathbf{x}_{0:k} - \mathbf{x}_{0:k}^{(i)}) \equiv \widehat{p}(\mathbf{x}_{0:k}|\mathbf{y}_{1:k}), \quad (8.51)$$

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

$$\mathbb{E}\big[\mathbf{f}(\mathbf{x}_{0:k})\big] \approx \frac{1}{N_s} \sum_{i=1}^{N_s} \mathbf{f}(\mathbf{x}_{0:k}^{(i)}) \equiv \widehat{\mathbf{f}}_{N_s}(\mathbf{x})\ . \quad (8.52a)$$

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

$$\mathbb{E}\big[\mathbf{f}(\mathbf{x}_{0:k})\big] = \int \mathbf{f}(\mathbf{x}_{0:k}) \, \frac{w_k(\mathbf{x}_{0:k})}{p(\mathbf{y}_{1:k})} \, \pi(\mathbf{x}_{0:k}|\mathbf{y}_{1:k}) \, d\mathbf{x}_{0:k} \ , \tag{8.53a}$$

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

$$w_k(\mathbf{x}_{0:k}) = \frac{p(\mathbf{y}_{1:k}|\mathbf{x}_{0:k}) \, p(\mathbf{x}_{0:k})}{\pi(\mathbf{x}_{0:k}|\mathbf{y}_{1:k})} \ . \tag{8.54}$$

We can get rid of the generally unknown or hard to calculate normalizing density $p(\mathbf{y}_{1:k})$ in Eq. [8.53] as follow:

$$\mathbb{E}\big[\mathbf{f}(\mathbf{x}_{0:k})\big] = \frac{\mathbb{E}_\pi \, \big[\mathbf{f}(\mathbf{x}_{0:k}) \, w_k(\mathbf{x}_{0:k})\big]}{\mathbb{E}_\pi \big[w_k(\mathbf{x}_{0:k})\big]} \ , \tag{8.55}$$

where the notation $\mathbb{E}_\pi\big[.\big]$ 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:

$$\mathbb{E}\big[\mathbf{f}(\mathbf{x}_{0:k})\big] \approx \widetilde{\mathbb{E}}\big[\mathbf{f}(\mathbf{x}_{0:k})\big] = \sum_{i=1}^{N_s} \widetilde{w}_k^{(i)} \, \mathbf{f}\big(\mathbf{x}_{0:k}^{(i)}\big) \ , \tag{8.56}$$

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

$$\widetilde{w}_k^{(i)} = \frac{w_k\big(\mathbf{x}_{0:k}^{(i)}\big)}{\sum_{i=1}^{N_s} w_k\big(\mathbf{x}_{0:k}^{(i)}\big)} \ . \tag{8.57}$$

Suppose the proposal distribution has the following form :

$$\pi(\mathbf{x}_{0:k}|\mathbf{y}_{1:k}) = \pi(\mathbf{x}_0) \, \prod_{j=1}^{k} \pi(\mathbf{x}_j|\mathbf{x}_{0:j-1}, \mathbf{y}_{1:j}) \ . \tag{8.58}$$

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

first expressed in terms of $p(\mathbf{x}_{0:k-1}|\mathbf{y}_{1:k-1})$, $p(\mathbf{x}_k|\mathbf{y}_k)$, and $p(\mathbf{x}_k|\mathbf{x}_{k-1})$.

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 :

$$p(\mathbf{x}_{0:k}) = p(\mathbf{x}_0) \prod_{j=1}^{k} p(\mathbf{x}_j|\mathbf{x}_{j-1}) \tag{8.59}$$

$$p(\mathbf{y}_{1:k}|\mathbf{x}_{0:k}) = \prod_{j=1}^{k} p(\mathbf{y}_j|\mathbf{x}_j) \ , \tag{8.60}$$

and the posterior distribution can be factorized as :

$$p(\mathbf{x}_{0:k}|\mathbf{y}_{1:k}) = \frac{p(\mathbf{y}_k|\mathbf{x}_k)\ p(\mathbf{x}_k|\mathbf{x}_{k-1})}{p(\mathbf{y}_k|\mathbf{y}_{1:k-1})}\ p(\mathbf{x}_{0:k-1}|\mathbf{y}_{1:k-1}) \tag{8.61}$$

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

$$w_k^{(i)} = w_{k-1}^{(i)}\ \frac{p(\mathbf{y}_k|\mathbf{x}_k^{(i)})\ p(\mathbf{x}_k^{(i)}|\mathbf{x}_{k-1}^{(i)})}{\pi(\mathbf{x}_k^{(i)}|\mathbf{x}_{0:k-1}^{(i)}, \mathbf{y}_{1:k})} \ . \tag{8.62}$$

Furthermore, if $\pi(\mathbf{x}_k|\mathbf{x}_{0:k-1}, \mathbf{y}_{1:k}) = \pi(\mathbf{x}_k|\mathbf{x}_{k-1}, \mathbf{y}_k)$, 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 :

$$w_k^{(i)} \propto w_{k-1}^{(i)}\ \frac{p(\mathbf{y}_k|\mathbf{x}_k^{(i)})\ p(\mathbf{x}_k^{(i)}|\mathbf{x}_{k-1}^{(i)})}{\pi(\mathbf{x}_k^{(i)}|\mathbf{x}_{k-1}^{(i)}, \mathbf{y}_k)} \ , \tag{8.63}$$

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

$$p(\mathbf{x}_k|\mathbf{y}_{1:k}) \approx \sum_{i=1}^{N_s} w_k^{(i)}\ \delta_0(\mathbf{x}_k - \mathbf{x}_k^{(i)}) \ , \tag{8.64}$$

where the weights are defined in Eq. (8.63). It can be shown that as $N_s \to \infty$ the approximation Eq. (8.64) approaches 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.

8.4.4 Choice of Proposal Distribution

It has been shown [Doucet, 1997] that the optimal proposal distribution

$$\pi(\mathbf{x}_k^{(i)}|\mathbf{x}_{0:k-1}^{(i)}, \mathbf{y}_{1:k}) \doteq p(\mathbf{x}_k^{(i)}|\mathbf{x}_{k-1}^{(i)}, \mathbf{y}_k) \ , \tag{8.65}$$

minimizes the variance of the proposal weights conditional on $\mathbf{x}_{0:k-1}^{(i)}$ and $\mathbf{y}_{1:k}$. Nonetheless, the transition prior $p(\mathbf{x}_k^{(i)}|\mathbf{x}_{k-1}^{(i)})$ is the most popular choice of proposal distribution

$$\pi(\mathbf{x}_k^{(i)}|\mathbf{x}_{0:k-1}^{(i)}, \mathbf{y}_{1:k}) \doteq p(\mathbf{x}_k^{(i)}|\mathbf{x}_{k-1}^{(i)}) \ . \tag{8.66}$$

This proposal distribution is usually easier to implement, but it is not incorporating the most recent observations. Substitution of Eq. (8.58), (8.59), and (8.60) into Eq. (8.63) yields:

$$w_k^{(i)} = w_{k-1}^{(i)} \ p(\mathbf{y}_k|\mathbf{x}_k^{(i)}) \ . \tag{8.67}$$

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.

Algorithm 8.4.1 (Particle Filter (PF)) .

From the proposal distribution $\pi(\mathbf{x}_k|\mathbf{y}_{1:k})$ we draw N_s particles $\mathbf{x}_k^{(i)}$ that approximate the posterior distribution $p(\mathbf{x}_k|\mathbf{y}_{1:k})$ such that :

$$\mathbb{E}\big[\mathbf{g}(\mathbf{x}_k)\big] = \int \mathbf{g}(\mathbf{x}_k) \ p(\mathbf{x}_k|\mathbf{y}_{1:k}) \ d\mathbf{x}_k, \tag{8.68}$$

is approximated by :

$$\widetilde{\mathbb{E}}\big[\mathbf{g}(\mathbf{x}_k)\big] \ = \ \sum_{i=1}^{N_s} \ \widetilde{w}_k^{(i)} \ \mathbf{g}(\mathbf{x}_k^{(i)}) \tag{8.69}$$

with

$$\widetilde{w}_k^{(i)} \;=\; w_k^{(i)} \;\frac{1}{\sum_{j=1}^{N_s} w_k^{(j)}} \tag{8.70}$$

and $w_k^{(i)}$ *is given by*

$$w_k^{(i)} \;=\; \frac{p(\mathbf{y}_{1:k}|\mathbf{x}_k)\; p(\mathbf{x}_k)}{\pi(\mathbf{x}_k|\mathbf{y}_{1:k})}. \tag{8.71}$$

We can also derive a recursive estimate for the $w_k^{(i)}$ *such as*

$$w_k^{(i)} \;=\; w_{k-1}^{(i)}\; \frac{p(\mathbf{y}_k|\mathbf{x}_k)\; p(\mathbf{x}_k|\mathbf{x}_{k-1})}{\pi(\mathbf{x}_k|\mathbf{x}_{0:k-1}, \mathbf{y}_{1:k})} \tag{8.72}$$

Thus we can sample from the proposal distribution

$$\mathbf{x}_k^{(i)} \;\sim\; \pi(\mathbf{x}_k|\mathbf{x}_{0:k-1}, \mathbf{y}_{1:k}) \tag{8.73}$$

and evaluate the likelihood $p(\mathbf{y}_k|\mathbf{x}_k^{(i)})$ *and the transition prior* $p(\mathbf{x}_k^{(i)}|\mathbf{x}_{k-1})$*. We can choose*

$$\pi(\mathbf{x}_k|\mathbf{x}_{0:k-1}, \mathbf{y}_{1:k}) \doteq p(\mathbf{x}_k|\mathbf{x}_{k-1}, \mathbf{y}_k) \tag{8.74}$$

but to facilitate the implementation we choose

$$\pi(\mathbf{x}_k|\mathbf{x}_{0:k-1}, \mathbf{y}_{1:k}) \doteq p(\mathbf{x}_k|\mathbf{x}_{k-1}) \tag{8.75}$$

that gives

$$w_k \;=\; w_{k-1}\; p(\mathbf{y}_k|\mathbf{x}_k) \tag{8.76}$$

such as

$$w_k^{(i)} \;=\; w_{k-1}^{(i)} \;*\; llh_k^{(i)}. \tag{8.77}$$

where

$$llh_k^{(i)} \;=\; p(\mathbf{y}_k|\mathbf{x}_k^{(i)}) \tag{8.78}$$

In Fig. (8.2) we can see a description of the sequences of this Particle Filters algorithm.

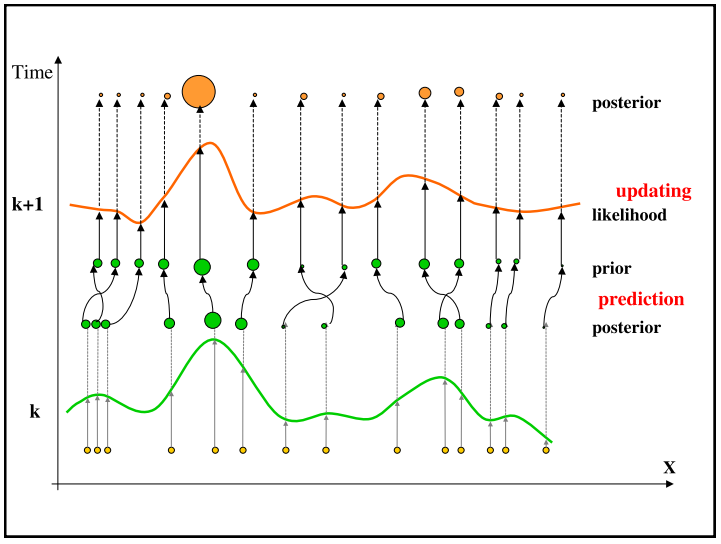

Figure 8.2: At time k we get the posterior $p(\mathbf{x}_k|\mathbf{y}_{1:k})$ with a weighted measure $\{\widetilde{\mathbf{x}}_k^{(i)}, \widetilde{w}_k^{(i)}\}$. At the *Prediction* step, using the state equation, we have an approximation of the proposal distribution and the prior $p(\mathbf{x}_{k+1}^{(i)}|\mathbf{y}_{1:k})$. At time $k+1$, we observe $\mathbf{y}_{k+1}$ and we get the likelihood function $p(\mathbf{y}_{k+1}|\mathbf{x}_{k+1}^{(i)})$. At the *Updating* step, for each particle we compute the importance weights $\{\widetilde{w}_{k+1}^{(i)}\}$ using this likelihood function, and we get the new posterior $p(\mathbf{x}_{k+1}|\mathbf{y}_{1:k+1})$ with a weighted measure $\{\widetilde{\mathbf{x}}_{k+1}^{(i)}, \widetilde{w}_{k+1}^{(i)}\}$.

8.4.5 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.

An intuitive solution could be to 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*,
- to multiply the particles with high normalized importance weights, and discard the particles with low normalized importance weights in a *Resampling* step.

To monitor how bad the weight degeneration is, we need a measure such as the effective sample size, N_{eff}:

$$N_{eff} = \frac{N_s}{1 + \text{Var}_{\pi(.|\mathbf{y}_{1:k})}[\widetilde{w}_k]} \tag{8.79}$$

$$= \frac{N_s}{\mathbb{E}_{\pi(.|\mathbf{y}_{1:k})}[(w_k)^2]} \leq N_s. \tag{8.80}$$

In practice, the true N_{eff} is not available, thus an estimate, $\widehat{N}_{eff}$ is given by:

$$\widehat{N}_{eff} = \frac{1}{\sum_{i=1}^{N_s} \left(w_k^{(i)}\right)^2}. \tag{8.81}$$

When $\widehat{N}_{eff}$ is below a predefined threshold N_T, the resampling procedure is performed.

8.4.6 Resampling

The basic idea of resampling is to eliminate particles which have small weights and to concentrate on particles with very large weights. From the Random Measure $\{\mathbf{x}_k^{(i)}, w_k^{(i)}\}$ that gives us an approximation of the posterior, we will generate a new Random Measure $\{\mathbf{x}_k^{(i)*}, N_s^{-1}\}$ that gives a new approximation of the same posterior.

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

$$p(\mathbf{x}_k|\mathbf{y}_{1:k}) \approx \sum_{i=1}^{N_s} w_k^{(i)}\ \delta_0(\mathbf{x}_k - \mathbf{x}_k^{(i)})\ , \tag{8.82}$$

so that $P\big(\mathbf{x}_k^{(i)*} = \mathbf{x}_k^{(j)}\big) = w_k^{(j)}$. The resulting sample is in fact an i.i.d. sample from the discrete density Eq. (8.82), and so the weights are now reset to $w_k^{(i)} = 1/N_s$.

8.4.7 Sequential Monte Carlo Algorithm

The *Sequential Monte Carlo Algorithm* consists in the four steps "*Initialisation*", "*Prediction*", *Updating*" and "*Resampling*".

During the *initialisation* step, we sample N_s independent random particles $\{\mathbf{x}^{(i)}\}$ from the initial distribution $p(\mathbf{x}_0)$.

The iterative procedures consist in the next steps.

At time $k-1$, we have an approximation of the posterior distribution $p(\mathbf{x}_{k-1}|\mathbf{y}_{1:k-1})$ by an unweighted Random measure $\{\widetilde{\mathbf{x}}_{k-1}^{(i)}, N_s^{-1}\}$.

Afterwards, the values of the particles are *predicted* for the next time step according to the dynamics of the state Markov process, and we obtain an approximation of the transition prior distribution $p(\mathbf{x}_k^{(i)}|\mathbf{x}_{k-1}^{(i)})$ by a new unweighted Random measure $\{\widehat{\mathbf{x}}_{k-1}^{(i)}, N_s^{-1}\}$.

Using the distribution of probability of the state noises $p(\mathbf{v}_k)$, we obtain an approximation of the prior distribution $p(\mathbf{x}_k^{(i)}|\mathbf{y}_{1:k-1})$ by a new unweighted Random measure $\{\widehat{\mathbf{x}}_{k-1}^{(i)}, N_s^{-1}\}$.

At time k, we observe $\mathbf{y}_k$ and we get the likelihood function $p(\mathbf{y}_k|\mathbf{x}_k^{(i)})$.

During the "*Updating*" step, each predicted particle is weighted by the likelihood function. For each particle we compute the importance weights $\{\widetilde{w}_k^{(i)}\}$ using this likelihood function, to get a new weighted Random measure $\{\widetilde{\mathbf{x}}_{k-1}^{(i)}, \widetilde{w}_k^{(i)}\}$.

The "*Resampling*" step can be view as a special case of a "Selection" step. The resampling step selects only the "fittest" particles to approximate the new posterior distribution $p(\mathbf{x}_k|\mathbf{y}_{1:k})$ by the final unweighted measure $\{\widetilde{\mathbf{x}}_k^{(i)}, N_s^{-1}\}$. This step gives birth to some particles at the expense of light particles which die.

Algorithm 8.4.2 (Bootstrap Particle Filter (BPF)) .

Let's the system

$$\mathbf{x}_k = \mathbf{f}_k(\mathbf{x}_{k-1}, \mathbf{u}_k, \mathbf{v}_k) \tag{8.83}$$

$$\mathbf{y}_k = \mathbf{h}_k(\mathbf{x}_k, \mathbf{n}_k) \tag{8.84}$$

where :

- $\mathbf{x}_k \in R^n$, $\mathbf{y}_k \in R^q$, $\mathbf{u}_k \in R^m$, $\mathbf{v}_k \in R^p$, $\mathbf{n}_k \in R^q$,
- $\mathbf{f}_k$ *is the state equation,* $\mathbf{h}_k$ *is the measurement equation, these parametric functions may be time-variant but are known,*
- $\{\mathbf{u}_k\}$ *is a deterministic sequence,*
- $\{\mathbf{v}_k\}$, $\{\mathbf{n}_k\}$ *are process and measurement noise sequences; with* $\mathbf{v}_k \sim \mathcal{N}(0, \mathbf{Q})$, $\mathbf{n}_k \sim \mathcal{N}(0, \mathbf{R})$,
- $\mathbf{x}_0 \sim \mathcal{N}(\overline{\mathbf{x}_0}, \mathbf{P}_0)$.

Initialization

We draw N *particles* $\{\mathbf{x}_0^{(i)}\}$ *from the estimated distribution* $p(\mathbf{x}_0)$ *of the initial state* $\mathbf{x}_0$.

at time $\mathbf{k-1}$, *we have particles* $\{\mathbf{x}_{k-1}^{(i)}\}$ *and weights* $\{w_k^{(i)}\}$ *that give us an approximation of the posterior distribution* $p(\mathbf{x}_{k-1}^{(i)}|\mathbf{y}_{1:k-1})$.

Prediction step

We generate process noise $\mathbf{v}_k^{(i)}$ *according to its distribution and we estimate* N *new particles, using the state equation*

$$\mathbf{x}_k^{(i)} \;=\; \mathbf{f}_k(\mathbf{x}_{k-1}^{(i)}, \mathbf{v}_k^{(i)}) \tag{8.85}$$

at time $\mathbf{k}$, *we observe* $\mathbf{y}_k$

Updating step

we estimate

$$\widehat{\mathbf{y}}_k^{(i)} \;=\; \mathbf{h}_k(\mathbf{x}_k^{(i)}) \tag{8.86}$$

and we have

$$\mathbf{e}_k^{(i)} \;=\; \mathbf{y}_k \;-\; \widehat{\mathbf{y}}_k^{(i)} \tag{8.87}$$

we estimate the likelihood function

$$Llh_k^{(i)} \;=\; \frac{1}{(2\pi)^{\frac{q}{2}}(detR)^{\frac{1}{2}}} \exp\Big[-\frac{1}{2}\big(\mathbf{e}_k^{(i)} - \boldsymbol{\mu}_n\big)^T \mathbf{R}^{-1}\big(\mathbf{e}_k^{(i)} - \boldsymbol{\mu}_n\big)\Big] \tag{8.88}$$

and we find the weights

$$w_k^{(i)} \;=\; \widetilde{w}_{k-1}^{(i)} \;*\; Llh_k^{(i)} \tag{8.89}$$

that we normalize

$$\widetilde{w}_k^{(i)} \;=\; w_k^{(i)} \;\frac{1}{\sum_{j=1}^{N} w_k^{(j)}} \tag{8.90}$$

and we get $\{(\mathbf{x}_k^{(i)}, \widetilde{w}_k^{(i)})\}$ *to approximate the new posterior distribution* $p(\mathbf{x}_k|\mathbf{y}_k)$.

Resampling step

From $\{\mathbf{x}_k^{(i)}\}$ *and* $\{\widetilde{w}_k^{(i)}\}$ *we eliminate particles with lower weights and duplicate particles with higher weights to give a new set of* N *particles* $\{\widetilde{\mathbf{x}}_k^{(i)}\}$ *with the same weights* $\{\widetilde{w}_k^{(i)} = \frac{1}{N}\}$.

Expectation step

We have got

$$E[\mathbf{x}_k] \;=\; \sum_{i=1}^{N} \widetilde{w}_k^{(i)} \widetilde{\mathbf{x}}_k^{(i)} \;. \tag{8.91}$$

In Fig. (8.3) we can see a description of the sequences of this Bootstrap algorithm.
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.

8.4.8 Markov Chain Monte Carlo (MCMC)

Consider a state vector $\mathbf{x} \in \mathbb{R}^{N_x}$ in a probability space of $\{\Omega, \mathcal{F}, P\}$. Let $K(.,.)$ a transition kernel in the state space, which represents the probability of moving from $\mathbf{x}$ to a point in a set $S \in \mathcal{B}$ (where $\mathcal{B}$ is a Borel σ-field on $\mathbb{R}^{N_x}$). A Markov chain is a sequence of random variable $\{\mathbf{x}_n\}_{n\geq 0}$ such that:

$$Pr(\mathbf{x}_n \in \mathcal{B}|\mathbf{x}_0, \cdots, \mathbf{x}_{n-1}) = Pr(\mathbf{x}_n \in \mathcal{B}|\mathbf{x}_{n-1}), \tag{8.92}$$

and:

$$K(\mathbf{x}_{n-1}, \mathbf{x}_n) = p(\mathbf{x}_n|\mathbf{x}_{n-1}). \tag{8.93}$$

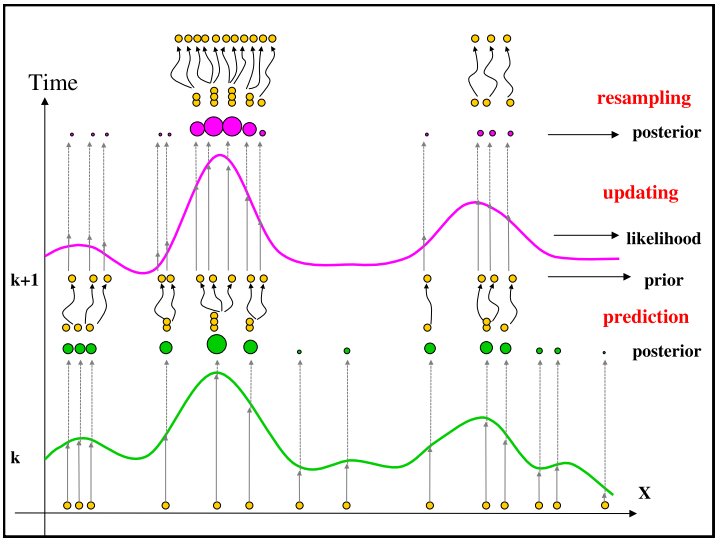

Figure 8.3: At time k, after resampling, we get an approximation of the posterior $p(\widetilde{\mathbf{x}}_k|\mathbf{y}_{1:k})$ with an unweighted measure $\{\widetilde{\mathbf{x}}_k^{(i)}, N_s^{-1}\}$. At the *Prediction* step, using the state equation, we have an approximation of the prior distribution $p(\widehat{\mathbf{x}}_{k+1}^{(i)}|\mathbf{y}_{1:k})$ by a new unweighted measure $\{\widehat{\mathbf{x}}_{k+1}^{(i)}, N_s^{-1}\}$, At time $k+1$, we observe $\mathbf{y}_{k+1}$ and we get the likelihood function $p(\mathbf{y}_{k+1}|\widehat{\mathbf{x}}_{k+1}^{(i)})$. At the *Updating* step, for each particle we compute the importance weights $\{\widetilde{w}_{k+1}^{(i)}\}$ using this likelihood function. Afterwards, the *Resampling* step selects only the "fittest" particles to approximate the new posterior distribution $p(\widetilde{\mathbf{x}}_{k+1}|\mathbf{y}_{1:k+1})$ by the final unweighted measure $\{\widetilde{\mathbf{x}}_{k+1}^{(i)}, N_s^{-1}\}$.

A Markov chain is characterized by the properties of its states, e.g. transiency, periodicity, irreducibility, and ergodicity. The foundation of Markov chain theory is the *Ergodicity Theorem*, which establishes under which a Markov chain can be analyzed to determine its steady state behavior.

Theorem 8.4.1 (Ergodic Theorem) . *If a Markov chain is ergodic, then there exists a unique steady state distribution π independent of the initial state.*

Markov chain theory is mainly concerned about finding the conditions under which there exists an invariant distribution Q and conditions under which it-

erations of transition kernel converge to the invariant distribution.

The invariant distribution satisfies

$$Q(d\mathbf{x}') = \int_x K(\mathbf{x}, d\mathbf{x}')\pi(\mathbf{x})d\mathbf{x}, \tag{8.94}$$

$$\pi(\mathbf{x}') = \int_x K(\mathbf{x}, \mathbf{x}')\pi(\mathbf{x})d\mathbf{x}. \tag{8.95}$$

where $\mathbf{x}' \in S \subset \mathbb{R}^{N_x}$, and π is the density w.r.t. Lebesgue measure of Q such that $Q(d\mathbf{x}') = \pi(\mathbf{x}')d\mathbf{x}'$.

The n-th iteration is thus given by $\int_x K^{(n-1)}(\mathbf{x}, d\mathbf{x}')K(\mathbf{x}', S)$. When $n \to \infty$, the initial state $\mathbf{x}$ will converge to the invariant distribution Q.

Markov chain Monte Carlo (MCMC) algorithms turn around the Markov chain theory. The invariant distribution or density is assumed to be known which corresponds to the target density $\pi(\mathbf{x})$, but the transition kernel is unknown. In order to generate samples from $\pi(.)$, the MCMC methods attempt to find a $K(\mathbf{x}, d\mathbf{x}')$ whose n-th iteration (for large n) converges to $\pi(.)$ given an arbitrary starting point.

Metropolis-Hastings Algorithm

Metropolis-Hastings algorithm is a kind of MCMC algorithm whose transition is associated with the acceptance probability.

Assume $q(\mathbf{x}, \mathbf{x}')$ as the proposal distribution (candidate target) that does not satisfy the reversibility condition, without loss of generality, suppose $\pi(\mathbf{x})q(\mathbf{x}, \mathbf{x}') > \pi(\mathbf{x}')q(\mathbf{x}', \mathbf{x})$, which means the probability moving from $\mathbf{x}$ to $\mathbf{x}'$ is bigger (more frequent) than the probability moving from $\mathbf{x}'$ to $\mathbf{x}$.

Intuitively, we want to change this situation to reduce the number of moves from $\mathbf{x}$ to $\mathbf{x}'$. By doing this, we introduce a probability of move, $0 < \alpha(\mathbf{x}, \mathbf{x}') < 1$; if the move is not performed, the process returns $\mathbf{x}$ as a value from the target distribution.

Hence the the transition from $\mathbf{x}$ to $\mathbf{x}'$ now becomes:

$$p_{MH}(\mathbf{x}, \mathbf{x}') = q(\mathbf{x}, \mathbf{x}')\alpha(\mathbf{x}, \mathbf{x}'), \tag{8.96}$$

where $\mathbf{x} \neq \mathbf{x}'$.

In order to make Eq. (8.96) satisfy the reversibility condition, $\alpha(\mathbf{x}, \mathbf{x}')$ needs to be set to [**?**]:

$$\alpha(\mathbf{x}, \mathbf{x}') = \begin{cases} \min\left[\frac{\pi(\mathbf{x}')\ q(\mathbf{x}', \mathbf{x})}{\pi(\mathbf{x})\ q(\mathbf{x}, \mathbf{x}')}\ ,\ 1\right], & \text{if } \pi(\mathbf{x})\ q(\mathbf{x}, \mathbf{x}') > 0, \\ 1, & \text{otherwise} \end{cases} \tag{8.97}$$

Hence the probability that the Markov process stays at $\mathbf{x}$ is given by

$$1 - \int_x q(\mathbf{x}, \mathbf{x}')\alpha(\mathbf{x}, \mathbf{x}')d\mathbf{x}', \tag{8.98}$$

and the transition kernel is given by

$$\begin{aligned} K_{MH}(\mathbf{x}, d\mathbf{x}') &= q(\mathbf{x}, \mathbf{x}')\alpha(\mathbf{x}, \mathbf{x}')d\mathbf{x}' \\ &\quad + \left[1 - \int_x q(\mathbf{x}, \mathbf{x}')\alpha(\mathbf{x}, \mathbf{x}')d\mathbf{x}'\right]\delta_{\mathbf{x}}(d\mathbf{x}')\ . \end{aligned} \tag{8.99}$$

The generic procedure is as follows [Chien and Fu, 1992]

Algorithm 8.4.3 (Metropolis-Hasting) .

- *for* $i = 1, \cdots, N_s$,
  1. *at iteration* $n = 0$, *draw a starting point* $\mathbf{x}_0$ *from a prior density;*
  2. *generate a uniform random variable* $u \sim U(0, 1)$, *and* $\mathbf{x}' \sim q(\mathbf{x}_n, .)$;
  3. – *if* $u < \alpha(\mathbf{x}_n, \mathbf{x}')$, *set* $\mathbf{x}_{n+1} = \mathbf{x}'$,
     – *else* $\mathbf{x}_{n+1} = \mathbf{x}_n$;
  4. $n = n + 1$, *repeat steps 2 and 3, until certain (say* k*) steps (i.e. burn-in time), store* $\mathbf{x}^{(i)} = \mathbf{x}_k$
- $i = i + 1$, *repeat the procedure until* N_s *samples are drawn,*
- *return the samples* $\{\mathbf{x}^{(1)}, \cdots, \mathbf{x}^{(N_s)}\}$.

Gibbs Sampling

The Gibbs sampler uses the concept of alternating (marginal) conditional sampling. Given an N_x-dimensional state vector $\mathbf{x} = [\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_{N_x}]^T$, we are interested in drawing the samples from the marginal density in the case where joint density is inaccessible or hard to sample.

The generic procedure is as follows [Casella and George, 1992]:

Algorithm 8.4.4 (Gibbs) .

1. *at iteration* $n = 0$, *draw* $\mathbf{x}_0$ *from the prior density* $p(\mathbf{x}_0)$,
2. *at iterations* $n = 1, 2, \cdots$,
   - *draw a sample* $\mathbf{x}_{1,n}$ *from* $p(\mathbf{x}_1|\mathbf{x}_{2,n-1}, \mathbf{x}_{3,n-1}, \cdots, \mathbf{x}_{N_x,n-1})$,
   - *draw a sample* $\mathbf{x}_{2,n}$ *from* $p(\mathbf{x}_2|\mathbf{x}_{1,n}, \mathbf{x}_{3,n-1}, \cdots, \mathbf{x}_{N_x,n-1})$,
   - $\cdots$,
   - *draw a sample* $\mathbf{x}_{N_x,n}$ *from* $p(\mathbf{x}_{N_x}|\mathbf{x}_{1,n}, \mathbf{x}_{2,n}, \cdots, \mathbf{x}_{N_x-1,n})$;

Remark 8.4.1 .

- *Gibbs sampling is an alternating sampling scheme, since the conditional density to be sampled is low-dimensional, the Gibbs sampler is a nice solution to estimation of hierarchical or structured probabilistic model,*
- *Gibbs sampling can be viewed as a Metropolis method in which the proposal distribution is defined in terms of the conditional distributions of the joint distribution and every proposal is always accepted [Carter and Kohn, 1994],*
- *Gibbs sampling has been extensively used for dynamic state space model within the Bayesian framework [Carter and Kohn, 1994].*

8.4.9 Better Proposal Distributions

The success of the particle filter algorithms depends on the validity of these 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 is 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 to 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 does 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:

$$\pi(\mathbf{x}_k|\mathbf{x}_{0:k-1}, \mathbf{y}_{1:k}) \doteq p(\mathbf{x}_k|\mathbf{x}_{k-1}, \mathbf{y}_k) \tag{8.100a}$$

$$= q_{\mathcal{N}}(\mathbf{x}_k|\mathbf{y}_{1:k}) \ , \tag{8.100b}$$

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,

$$q_{\mathcal{N}}(\mathbf{x}_k|\mathbf{y}_{1:k}) = \mathcal{N}\big(\ \mathbf{x}_k;\ \mathbf{x}_k^{(i)}\ ,\ \mathbf{P}_{\mathbf{x}_k}^{(i)}\ \big) \quad i = 1, \cdots, N_s \tag{8.101}$$

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

We can find a complete description of the "Sigma-Point Particle Filter (SPPF)" in Appendix: [H.7.1].

This Sigma-Point Particle Filter algorithm was published first in
[van der Merwe, et al. , 2001].

8.5 Adaptive Particle Filters

8.5.1 Introduction

The computational load of the particle filter algorithm is linear in the number of samples N_s used for the estimation. We could increase the efficiency of particle filters by adapting the number of samples during the time steps.

In the beginning of the procedure, the estimation of the state is highly uncertain and a large number of samples is needed to accurately represent the posterior $p(\mathbf{x}_k|\mathbf{y}_{1:k})$. But, when the iterations evolve, we get a better approximation of the true posterior and only a small number of samples should be enough to accurately track the state. Thus, we could adapt the number of samples during the process, choosing large sample sets at the beginning of the procedure, and small sample sets afterward.

Based on the likelihood of observations, we would like to determine the number of samples to approximate the true posterior. If the sample set is in tune with the measurements then, each individual importance weight is large and the sample set can be small. On the contrary, when the likelihood happens to be lie in one of the tails of the distribution, or if it is too narrow (highly peaked due to low measurement noise) then, the individual sample weights are small and the sample set must become larger.

The idea is to determine the optimal number of samples N_s to approximate the true posterior, $p(\mathbf{x}_k|\mathbf{y}_{1:k})$, such as the error between the true posterior and its sample-based approximation, $p(\mathbf{x}_k^{(i)}|\mathbf{y}_{1:k})$ is less than ϵ with probability $1-\delta$, using the Kullback-Leibler Divergence (KLD) as the distance between the sample-based maximum likelihood estimate (MLE) and the true posterior .

8.5.2 Adaptation

We draw N_s samples from this distribution with b different bins to get an approximate of the true posterior distribution.

We choose the number of samples N_s such as

$$N_s = \frac{1}{2\epsilon}\chi^2_{b-1,1-\delta} \tag{8.102a}$$

$$\doteq \frac{b-1}{2\epsilon}\Big(z_{1-\delta}\sqrt{\frac{2}{9(b-1)}} + \frac{2}{9(b-1)} + 1\Big)^3, \tag{8.102b}$$

where $z_{1-\delta}$ is the upper $1-\delta$ quantile of the normal distribution (see developments in Appendix [B.1]).

We can see that the required number of particles to approximate the true posterior distribution is proportional to the inverse of the error bound ϵ, and to the first order linear in the number of bins b with support. We assume that a bin of the multinomial distribution has support if its probability is above a threshold, that means it contains at least one particle.

8.5.3 Using adaptation in particle filters

To incorporate this result into particle filter algorithm we need the true posterior distribution $p(\mathbf{x}_k|\mathbf{y}_{1:k})$ which we do not know. But the solution is to rely on the sample-based representation of the transition prior $p(\mathbf{x}_k^{(i)}|\mathbf{x}_{k-1}^{(i)})$ as an estimate for the posterior.

Furthermore, Eq. (B.25), in Appendix [B.1], shows that it is not necessary to determine the complete discrete distribution, but it suffices to determine the number k of bins with support (for given ϵ and δ). Nevertheless, we do not know k before we actually generate all samples from the predictive distribution, we can estimate k by counting the number of bins with support during the sampling.

8.5.4 Problem with Kullback-Leibler Divergence Sampling

The problem with KLD-Sampling is the derivation of the bound using the empirical distribution, which has the implicit assumption that the samples come from the true distribution. This is not the case for particle filters where the samples come from an importance function.

Furthermore, the quality of the match between this function and the true distribution is one of the main elements that determines the accuracy of the filter, hence the suitable number of particles. The bound given by KLD-Sampling only uses information about the complexity of the true posterior, but it ignores any mismatch between the true and the proposal distribution.

To fix the problem of KLD-Sampling we need a way to quantify the degradation in the estimation using samples from the importance function. The goal is to find the equivalent number of samples from the importance and the true densities that captures the same amount of information about the latter.

In the context of Monte Carlo (MC) integration, [Geweke, 1989] introduced the concept of Relative Numerical Efficiency (RNE), which provides an index

to quantify the influence of sampling from an importance function. The idea behind RNE is to compare the relative accuracy of solving an integral using samples coming from both the true and the proposal density. Accuracy is measured according to the variance of the estimator of the integral.

If we use MC integration to estimate the mean value of the state $(\mathbb{E}_{MC}(x))$, the variance of the estimator is given by [Doucet, et al., 2001]

$$Var\left[\mathbb{E}^N_{MC}(x)\right] = \frac{Var_p(x)}{N} \tag{8.103}$$

where N is the number of samples coming from the true distribution $p(x)$, and the subscript p expresses that the variance involved is computed using the target distribution.

When the samples come from an importance function $q(x)$, the variance of the estimator corresponds to the variance of Importance Sampling (IS), which is given by [Geweke, 1989]:

$$Var\left[\mathbb{E}^N_{IS}(x)\right] = \frac{\mathbb{E}_q\Big[\big(x - \mathbb{E}_p(x)\big)^2\big(w(x)\big)^2\Big]}{N_{IS}} \tag{8.104a}$$

$$= \frac{\sigma^2_{IS}}{N_{IS}} \tag{8.104b}$$

where $w(x)$ corresponds to $p(x)/q(x)$ the weights of IS and N_{IS} is the number of samples coming from the importance function.

To achieve similar levels of accuracy, the variance of both estimators should be equal. This allow us to find a relation that quantifies the equivalence between samples from the true and the proposal density,

$$N = \frac{N_{IS} Var_p(x)}{\sigma^2_{IS}}, \tag{8.105}$$

Replacing Eq. (8.105) in Eq. (8.102) allows us to correct the bound given by KLD-Sampling when the samples do not come from the true distribution but from an importance function:

$$N_{IS} > \frac{\sigma^2_{IS}}{Var_p(x)} \frac{1}{2\epsilon} \chi^2_{k-1,1-\delta}. \tag{8.106}$$

Using MC integration, $Var_p(x)$ and σ^2_{IS} can be estimated by:

$$\begin{aligned} Var_p(x) &= \mathbb{E}_p(x^2) - \big[\mathbb{E}p(x)\big]^2 \\ &\approx \frac{\sum_{i=1}^N (x^{(i)})^2 w^{(i)}}{\sum_{i=1}^N w^{(i)}} - \big[\mathbb{E}_p(x)\big]^2 \end{aligned} \tag{8.107}$$

and

$$\sigma_{IS}^2 \approx \frac{\sum_{i=1}^{N}(x^{(i)})^2(w^{(i)})^2}{\sum_{i=1}^{N} w^{(i)}} - \frac{2\sum_{i=1}^{N} x^{(i)}(w^{(i)})^2\mathbb{E}_p(x)}{\sum_{i=1}^{N} w^{(i)}} + \frac{\sum_{i=1}^{N}(w^{(i)})^2\left[\mathbb{E}_p(x)^2\right]}{\sum_{i=1}^{N} w^{(i)}}, \tag{8.108}$$

with

$$\mathbb{E}_p(x) = \frac{\sum_{i=1}^{N} x^{(i)}w^{(i)}}{\sum_{i=1}^{N} w^{(i)}}. \tag{8.109}$$

Eq. (8.109) shows that using appropriate accumulators, it is possible to calculate the bound incrementally keeping the $O(N)$ complexity of the filter.

8.5.5 Asymptotic Normal distribution

Usually, the particle filter keeps track of the posterior density with the goal of estimating the mean or higher order moments of the state. This suggests an alternative error metric to determine the number of particles. Instead of checking the accuracy of the estimation of the posterior, it is possible to check the accuracy of a particle filter in the estimation of a moment of this density.

Under weak assumptions and using the strong law of large numbers, it is possible to show that at each iteration the estimation of the mean given by the particle filter is asymptotically unbiased [DeGroot, 1989]. Furthermore, if the variance of this estimator is finite, the central limit theorem justifies an asymptotic normal approximation for it [DeGroot, 1989], which is given by:

$$\mathbb{E}_{PF}(x) \sim \mathcal{N}(\mathbb{E}_p(x); \frac{\sigma_{IS}^2}{N_{IS}}), \tag{8.110}$$

where $\mathcal{N}(\mu; \sigma^2)$ denotes the normal distribution with mean μ and standard deviation σ^2.

Using this approximation, it is possible to build a one sided confidence interval for the number of particles that limits the error in the estimation of the mean:

$$P\Big(\Big|\frac{\mathbb{E}_{PF}(x) - \mathbb{E}_p(x)}{\mathbb{E}_p(x)}\Big| \leq \epsilon\Big) \geq (1-\alpha) \tag{8.111}$$

where

- $|.|$ denotes absolute value,

- $\mathbb{E}_p(x)$ is the true mean value of the state,
- ϵ corresponds to the desired error
- $(1-\alpha)$ corresponds to the confidence level.

Following the usual derivation of confidence intervals, Eq. (8.111) produces the following bound for the number of particles:

$$N_{IS} \geq \frac{Z^2_{1-\alpha/2}\sigma^2_{IS}}{\epsilon^2 \mathbb{E}_p(x_k)^2} \ . \tag{8.112}$$

8.6 Distributed Processing

8.6.1 Introduction

In order to curb the CPU time of Particle Filters algorithms, we would like to split the load on a set of parallel computers, but the main issue in distributing this computation involved in particle filtering comes from the need to use all particles to obtain the estimate and resample for the next iteration, [Bashiet et al., 2005].

Assuming a double floating point representation, the size of N_s particles is $8 \times N_s \times N_x$. For a system of 500 particles at each node and a state dimension of $N_x = 5$, this comes to approximately 20KB/node, obviously large enough to be debilitating to trivial distribution.

We will call the processing centers charged with carrying out the computation "*nodes*", and the fusion center charged with directing the computation the "*director*".
In general, the speed gain from parallelization is governed by Amdahl's Law: The effective speedup of any parallel algorithm is limited by those parts of the algorithm that must be performed sequentially. If the ratio of the time taken for the part of the algorithm that is parallelizable is $t_p < 1$, and the time for the part of the algorithm that has to be performed serially is $t_s = 1 - t_p$, then the speedup is given by

$$\text{Speedup} = \frac{1}{t_s + \frac{1-t_s}{S}} \tag{8.113}$$

where S is the number of processing nodes that the computation is distributed across. The maximum speedup is given by $1/t_s$ as $S \to \infty$.

8.6.2 Local Distributed Particle Filter

Samples are drawn and importance weights are calculated and normalized locally. Furthermore, resampling is also performed locally, thus removing the need for the director to send the particles to other nodes. Each node sends only sufficient data for the estimation of $\mathbf{x}_k$ back to the director, which is then responsible for providing the filter estimate.

In order to implement a distributed particle filter such that particles are resampled locally, we must ensure that the director can correctly reconstruct the estimate and the covariance at the end of each iteration.

Let $\{\mathbf{x}_k^{s,i_s}, w_k^{s,i_s}\}_{i_s=1}^{N_s}$ denotes the set of importance samples and their corresponding weights obtained at node s, where N_s represents the number of particles treated at node s, such as $\sum_{s=1}^{S} N_s = N$.

The local (at node s) estimate and error covariance are given by the sample mean and sample covariance, respectively

$$\widehat{\mathbf{x}}_k^s = \sum_{i_s=1}^{N_s} w_k^{s,i_s} \mathbf{x}_k^{s,i_s} \tag{8.114}$$

$$\mathbf{P}_k^s = \sum_{i_s=1}^{N_s} w_k^{s,i_s} (\mathbf{x}_k^{s,i_s} - \widehat{\mathbf{x}}_k^{s,i_s})(\mathbf{x}_k^{s,i_s} - \widehat{\mathbf{x}}_k^{s,i_s})^T \,. \tag{8.115}$$

Consider now the union of all local sets of particles $\bigcup_{s=1}^{S} \{\mathbf{x}_k^{s,i_s}, w_k^{s,i_s}\}_{i_s=1}^{N_s}$.

Globally (at the director) the estimate and the covariance are given by the sample mean and sample covariance over the entire collective set, respectively

$$\widehat{\mathbf{x}}_k = \sum_{s=1}^{S} \sum_{i_s=1}^{N_s} w_k^{s,i_s} \mathbf{x}_k^{s,i_s} \tag{8.116}$$

$$\mathbf{P}_k = \sum_{s=1}^{S} \sum_{i_s=1}^{N_s} w_k^{s,i_s} (\mathbf{x}_k^{s,i_s} - \widehat{\mathbf{x}}_k^{s,i_s})(\mathbf{x}_k^{s,i_s} - \widehat{\mathbf{x}}_k^{s,i_s})^T \tag{8.117}$$

We have reduced the amount of data that needs to be sent to a vector, a matrix and a scalar for each node.

There is, however a possibility that the number of particles at each node is not sufficient to avoid particle depletion, where there is not enough diversity in the particle pool to properly track the state. This will lead to divergence if the issue is left unaddressed (see: [Choy and Edelman, 2003].

8.7 Rao-Blackwell Particle Filters (RBPF)

The idea of the Rao-Blackwellized Particle Filter (RBPF) is that sometimes it is possible to evaluate some of the filtering equations analytically and the others with Monte Carlo sampling instead of computing everything with pure sampling. According to the Rao-Blackwell theorem this leads to estimators with less variance than what could be obtained with pure Monte Carlo sampling.

An intuitive way of understanding this is that the marginalization replaces the finite Monte Carlo particle set representation with an infinite closed form particle set, which is always more accurate than any finite set. The sampling and resampling approach that we describe is not necessarily the most efficient in all conditions, but it turns out to work well in some applications.

By tuning the resampling algorithm and possibly changing the order of weight computation and sampling, accuracy and computational efficiency of the algorithm could possibly be improved. An important issue is that sampling could be more efficient without replacement, such that duplicate samples are not stored.

For dynamic state space model, the basic principle of Rao-Blackwellization is to exploit the model structure in order to improve the inference efficiency and consequently to reduce the variance. For example, we can attempt to decompose the dynamic state space into two parts, one part being calculated exactly using Kalman filter, the other part being inferred approximately using particle filter. Since the first part is inferred exactly and quickly, the computing power is saved and the variance is reduced.

Let the states vector be partitioned into two parts $\mathbf{x}_k = [\mathbf{x}_k^1, \mathbf{x}_k^2]$, where the marginal density $p(\mathbf{x}_k^2|\mathbf{x}_k^1)$ is assumed to be tractable analytically. The expectation of $f(\mathbf{x}_k)$ w.r.t. the posterior can be rewritten by:

$$\mathbb{E}\big[f(\mathbf{x}_k)\big] = \int f(\mathbf{x}_k^1, \mathbf{x}_k^2) p(\mathbf{x}_k^1, \mathbf{x}_k^2)|\mathbf{y}_{1:k}) d\mathbf{x}_k \tag{8.118}$$

$$= \frac{\int \lambda(\mathbf{x}_{0:k}^1) p(\mathbf{x}_{0:k}^1) d\mathbf{x}_{0:k}^1}{\int\int p(\mathbf{y}_{1:k}|\mathbf{x}_{0:k}^1, \mathbf{x}_{0:k}^2) p(\mathbf{x}_{0:k}^2|\mathbf{x}_{0:k}^1) d\mathbf{x}_{0:k}^2 p(\mathbf{x}_{0:k}^1) d\mathbf{x}_{0:k}^1} \tag{8.119}$$

$$= \frac{\int \lambda(\mathbf{x}_{0:k}^1) p(\mathbf{x}_{0:k}^1) d\mathbf{x}_{0:k}^1}{p(\mathbf{y}_{1:k}|\mathbf{x}_{0:k}^1) p(\mathbf{x}_{0:k}^1) d\mathbf{x}_{0:k}^1} \tag{8.120}$$

where

$$\lambda(\mathbf{x}_{0:k}^1) = \int f(\mathbf{x}_k^1, \mathbf{x}_k^2) p(\mathbf{y}_{1:k}|\mathbf{x}_k^1, \mathbf{x}_k^2) p(\mathbf{x}_{0:k}^2|\mathbf{x}_{0:k}^1) d\mathbf{x}_{0:k}^2. \tag{8.121}$$

And the weighted Monte Carlo estimate is given by

$$\widehat{f}_{RB} = \frac{\sum_{i=1}^{N_p} \lambda(\mathbf{x}_{0:k}^{1,(i)}) w(\mathbf{x}_{0:k}^{1,(i)})}{\sum_{i=1}^{N_p} w(\mathbf{x}_{0:k}^{1,(i)})} \ . \tag{8.122}$$

The lower variance of marginalized estimate is achieved because of the Rao-Blackwellization theorem

$$Var\big[f(\mathbf{x})\big] = Var\Big[\mathbb{E}[f(\mathbf{x}^1, \mathbf{x}^2)|\mathbf{x}^1]\Big] + \mathbb{E}\Big[Var[f(\mathbf{x}^1, \mathbf{x}^2)|\mathbf{x}^1]\Big]. \tag{8.123}$$

It has been proved that [Doucet et al., 2000], the variance of ratio of two joint densities is not less than that of two marginal densities

$$\begin{aligned} Var_q\Big[\frac{p(\mathbf{x}^1, \mathbf{x}^2)}{q(\mathbf{x}^1, \mathbf{x}^2)}\Big] &= Var_q\Big[\frac{\int p(\mathbf{x}^1, \mathbf{x}^2) d\mathbf{x}^2}{q(\mathbf{x}^1, \mathbf{x}^2) d\mathbf{x}^2}\Big] \\ &\quad + \mathbb{E}_q\Big[Var_q\big[\frac{p(\mathbf{x}^1, \mathbf{x}^2)}{q(\mathbf{x}^1, \mathbf{x}^2)}\Big|\mathbf{x}^1\big]\Big] \qquad (8.124) \\ &\geq Var_q\Big[\frac{\int p(\mathbf{x}^1, \mathbf{x}^2) d\mathbf{x}^2}{q(\mathbf{x}^1, \mathbf{x}^2) d\mathbf{x}^2}\Big], \qquad (8.125) \end{aligned}$$

where

$$\frac{p(\mathbf{x}^1, \mathbf{x}^2) d\mathbf{x}^2}{q(\mathbf{x}^1, \mathbf{x}^2) d\mathbf{x}^2} = \mathbb{E}_q\Big[\frac{p(\mathbf{x}^1, \mathbf{x}^2)}{q(\mathbf{x}^1, \mathbf{x}^2)}\Big|\mathbf{x}^1\Big]. \tag{8.126}$$

Hence by decomposing the variance, it is easy to see that the variance of the importance weights via Rao-Blackwellization is smaller than the one obtained using direct Monte Carlo method.

8.8 Stochastic Differential Equations (SDE)

In the first parts of this chapter, we have modeled a process as discretely observed stochastic differential equations, such as pure *discrete*-time model (discrete-time in dynamics, and discrete-time measurements) with non-linear state and measurement equations and non-Gaussian components.

But, because in the Nature, time is continuous, and not discrete, often a physical more realistic approach than discrete-time filtering is needed.

In *continuous-discrete* filtering the state dynamics are modeled as *continuous*-time stochastic processes, that is , as *stochastic differential equations*, and the measurements are observed at *discrete* instances of time.

Sometimes we have also to realize the modelization of a process in pure *Continuous*-time with *Stochastic Differential Equations* for dynamics and measurements.

In this part, we consider the following generic stochastic filtering problem in a continuous dynamic state-space form:

$$\dot{\mathbf{x}}_t = \mathbf{f}(\mathbf{x}_t, \mathbf{u}_t, \mathbf{v}_t, t) \tag{8.127}$$
$$\mathbf{y}_t = \mathbf{g}(\mathbf{x}_t, \mathbf{u}_t, \mathbf{n}_t, t), \tag{8.128}$$

We can also formulate the continuous-time stochastic filtering problem by the *Stochastic Differential Equations* (SDE) theory [Karlin and Taylor, 1975].

Suppose $\{\mathbf{x}_t\}$ is a Markov process with an infinitesimal generator; we can rewriting state-space equations in the following form of Itô SDE [Oksendal , 1998]:

$$d\mathbf{x}_t = \mathbf{f}(\mathbf{x}_t, t)dt + \boldsymbol{\sigma}(\mathbf{x}_t, t)d\boldsymbol{\beta}_t, \tag{8.129a}$$
$$d\mathbf{y}_t = \mathbf{h}(\mathbf{x}_t, t)dt + d\boldsymbol{\eta}_t \tag{8.129b}$$

where:

- $\mathbf{f}(\mathbf{x}_t, t)$ is the non-linear drift,
- $\boldsymbol{\sigma}(\mathbf{x}_t, t)$ is the volatility or diffusion coefficient.
- $\{\boldsymbol{\beta}_t, \boldsymbol{\eta}_t, t \geq 0\}$ are two Wiener processes,
- $\mathbf{x}_t \in \mathbb{R}^{N_x}$,
- $\mathbf{y}_t \in \mathbb{R}^{N_y}$.

For all $t \geq 0$ we define a partial differential *backward diffusion operator* $\mathbf{L}_t$ as [Gardiner, 1990]:

$$\mathbf{L}_t = \sum_{i=1}^{N_x} \mathbf{f}_t^i \frac{\partial}{\partial \mathbf{x}_i} + \frac{1}{2} \sum_{i,j=1}^{N_x} a_t^{ij} \frac{\partial}{\partial \mathbf{x}_i \partial \mathbf{x}_j}. \tag{8.130}$$

where:

$$a_t^{ij} = \boldsymbol{\sigma}_i(\mathbf{x}_t, t)\boldsymbol{\sigma}_j(\mathbf{x}_t, t). \tag{8.131}$$

Operator $\mathbf{L}$ corresponds to an infinitesimal generator of the diffusion process $\{\mathbf{x}_t,\ t \geq 0\}$.

Now, we have to deduce conditions under which we can find a recursive and finite-dimensional scheme to compute the conditional probability distribution

$p(\mathbf{x}_t|\mathcal{Y}_t)$, given the filtration $\mathcal{Y}_t$ produced by the observation process.

Let's define an innovation process

$$\mathbf{e}_t = \mathbf{y}_t - \int_0^t \mathbb{E}\big[\mathbf{g}(\mathbf{x}_s, s)|\mathcal{Y}_s\big]ds, \tag{8.132}$$

where $\mathbb{E}\big[\mathbf{g}(\mathbf{x}_s, s|\mathcal{Y}_s\big]$ is described as

$$\widehat{\mathbf{g}}(\mathbf{x}_t) = \mathbb{E}\big[\mathbf{g}(\mathbf{x}_t, t)|\mathcal{Y}_t\big] \tag{8.133}$$

$$= \int_{-\infty}^{\infty} \mathbf{g}(\mathbf{x}_t)p(\mathbf{x}_t|\mathcal{Y}_t)d\mathbf{x}. \tag{8.134}$$

For any test function $\phi \in \mathbb{R}^{N_x}$, the *forward diffusion operator* $\widetilde{\mathbf{L}}$ is defined as

$$\widetilde{\mathbf{L}}\phi = -\sum_{i=1}^{N_x} \mathbf{f}_t^i \frac{\partial \phi}{\partial \mathbf{x}_i} + \frac{1}{2}\sum_{i,j=1}^{N_x} a_t^{ij} \frac{\partial^2 \phi}{\partial \mathbf{x}_i \partial \mathbf{x}_j}, \tag{8.135}$$

which essentially is the Fokker-Planck operator, and is the adjoint of the backward operator of Eq. (8.130).

Given initial condition $p(\mathbf{x}_0)$ at $t = 0$ as boundary condition, it turns out that the pdf of diffusion process satisfies the *Fokker- Planck-Kolmogorov equation* (FPK) [Horsthemke and Lefever, 1984]; a.k.a. *Kolmogorov forward equation* [Risken, 1989],

$$\frac{\partial p(\mathbf{x}_t)}{\partial t} = \widetilde{\mathbf{L}}_t p(\mathbf{x}_t) \ , \tag{8.136}$$

and also the *Kolmogorov backward equation*

$$\frac{\partial p(\mathbf{x}_t)}{\partial t} = -\mathbf{L}_t p(\mathbf{x}_t). \tag{8.137}$$

The stochastic process is determined equivalently by the FPK Eq. (8.136) or the SDE Eq. (8.129).

The FPK equation can be interpreted as follows:

- the first term is the equation of motion for a cloud of particles whose distribution is $p(\mathbf{x}_t)$, each point of which obeys the equation of motion $d\mathbf{x}/dt = \mathbf{f}(\mathbf{x}_t, t)$,
- the second term describes the disturbance due to Brownian motion.

The solution of Eq. (8.136) can be solved exactly by Fourier transform. By inverting the Fourier transform, we obtain:

$$p(\mathbf{x}, t+\Delta t|\mathbf{x}_0, t) = \frac{1}{\sqrt{2\pi\sigma_0\Delta t}} \exp\Big\{ - \frac{\big(\mathbf{x} - \mathbf{x}_0 - \mathbf{f}(\mathbf{x}_0)\Delta t\big)^2}{2\sigma_0\Delta t}\Big\}, \quad (8.138)$$

which is a Gaussian distribution of a deterministic path.

By involving the innovation process Eq. (8.132) and assuming $\mathbb{E}\big[\mathbf{v}_t\mathbf{v}_t^T\big] = \boldsymbol{\Sigma}_{\mathbf{v},t}$, we have the following Kushner's equation (e.g., [Kushner, 1967]):

$$dp(\mathbf{x}_t|\mathcal{Y}_t) = \widetilde{\mathbf{L}}_t p(\mathbf{x}_t|\mathcal{Y}_t)dt + p(\mathbf{x}_t|\mathcal{Y}_t)\mathbf{e}_t\boldsymbol{\Sigma}_{\mathbf{v},t}^{-1}dt, \quad (t \geq 0) \quad (8.139)$$

which reduces to the FPK Eq. (8.136) when there are no observations or filtration $\mathcal{Y}_t$.

Integrating Eq. (8.139), we have

$$p(\mathbf{x}_t|\mathcal{Y}_t) = p(\mathbf{x}_0) + \int_0^t \widetilde{\mathbf{L}}_s p(\mathbf{x}_s|\mathcal{Y}_s)ds + \int_0^t p(\mathbf{x}_s|\mathcal{Y}_s)\mathbf{e}_s\boldsymbol{\Sigma}_{\mathbf{v},s}^{-1}ds \quad (8.140)$$

Given conditional pdf Eq. (8.140), suppose we want to calculate $\widehat{\phi}(\mathbf{x}_t) = \mathbb{E}\big[\phi(\mathbf{x}_t)|\mathcal{Y}_t\big]$ for any nonlinear function $\phi \in R^{N_x}$.

By interchanging the order of integrations, we have

$$\begin{aligned}
\widehat{\phi}(\mathbf{x}_t) &= \int_{-\infty}^{\infty} \phi(\mathbf{x}) p(\mathbf{x}_t|\mathcal{Y}_t)d\mathbf{x} && (8.141)\\
&= \int_{-\infty}^{\infty} \phi(\mathbf{x}) p(\mathbf{x}_0)d\mathbf{x} \\
&\quad + \int_0^t \int_{-\infty}^{\infty} \phi(\mathbf{x})\widetilde{\mathbf{L}}_s p(\mathbf{x}_s|\mathcal{Y}_s)d\mathbf{x}ds \\
&\quad + \int_0^t \int_{-\infty}^{\infty} \phi(\mathbf{x}) p(\mathbf{x}_s|\mathcal{Y}_s)\mathbf{e}_s\boldsymbol{\Sigma}_{\mathbf{v},s}^{-1}d\mathbf{x}ds && (8.142)\\
&= \mathbb{E}\big[\phi(\mathbf{x}_0)\big] + \int_0^t \int_{-\infty}^{\infty} p(\mathbf{x}_s|\mathcal{Y}_s)\widetilde{\mathbf{L}}_s\phi(\mathbf{x})d\mathbf{x}ds \\
&\quad + \int_0^t \Big[\int_{-\infty}^{\infty} \phi(\mathbf{x})\mathbf{g}(\mathbf{x}, s)p(\mathbf{x}_s|\mathcal{Y}_s)d\mathbf{x} \\
&\quad\quad - \mathbf{g}(\mathbf{x}_s)\int_{-\infty}^{\infty} \phi(\mathbf{x})p(\mathbf{x}_s|\mathcal{Y}_s)d\mathbf{x}\Big]\boldsymbol{\Sigma}_{\mathbf{v},s}^{-1}ds. && (8.143)
\end{aligned}$$

The Kushner equation lends itself a recursive form of filtering solution, but the conditional mean requests all of higher-order conditional moments and thus

leads to an infinite-dimensional system.

On the other hand, under some mild conditions, the *unnormalized conditional* density of $\mathbf{x}_t$ given $\mathcal{Y}_s$, denoted as $\pi(\mathbf{x}_t|\mathcal{Y}_s)$, is the unique solution of the following stochastic partial differential equation (PDE), the so-called Zakai equation (see [Jazwinski, 1970], [Kushner, 1977], [Gardiner, 1990], [Horsthemke and Lefever, 1984]):

$$d\pi(\mathbf{x}_t|\mathcal{Y}_t) = \widetilde{\mathbf{L}}\pi(\mathbf{x}_t|\mathcal{Y}_t)dt + \mathbf{g}(\mathbf{x}_t, t)\pi(\mathbf{x}_t|\mathcal{Y}_t)d\mathbf{y}_t \tag{8.144}$$

with the same $\widetilde{\mathbf{L}}$ defined in Eq. 8.139.

Zakai equation and Kushner equation have a one-to-one correspondence, but Zakai equation is much simpler, hence we are usually turned to solve the Zakai equation instead of Kushner equation. This is true because Eq. (8.144) is linear w.r.t. $\pi(\mathbf{x}_t|\mathcal{Y}_t$ whereas [8.139] involves certain nonlinearity. In the early history of nonlinear filtering, the common way is to discretize the Zakai equation to seek the numerical solution.

We will discus these algorithms for forecasting in *Continuous-discrete*-time and pure *continuous*-time in Chapter [9].

8.9 Theoretical Issues

8.9.1 Convergence and Asymptotic Results

Under some mild conditions the particle methods converge to the solution of the Zakai equation [Crisan et al., 1999a] and Kushner-Stratonovich equation [Crisan et al., 1999b] and we can find the sufficient and necessary conditions for the a.s. convergence of particle filter to the true posterior in [Crisan , 2003].

Almost Sure Convergence

If the transition kernel $K(\mathbf{x}_k|\mathbf{x}_{k-1})$ is *Feller* [1] , importance weights are upper bounded, and the likelihood function is continuous, bounded, and strictly positive, then with $N_s \to \infty$ the filtered density given by particle filter converges asymptotically to the true posterior.

[1] A kernel is *Feller*, means that for any continous bounded function ϕ, $K\phi$ is also continous bounded function

Mean Square Convergence

If likelihood function is bounded, for any bounded function $\phi \in \mathbb{R}_{N_x}$ then for $t \geq 0$, there exists a $C_{k|k}$ independent of N_s s.t. [Del Moral, 2004]

$$\mathbb{E}\Big[\Big((\widehat{P}_{k|k}, \phi) - (P_{k|k}, \phi)\Big)^2\Big] = C_{k|k} \frac{\|\phi\|^2}{N_s}, \tag{8.145}$$

where $\big(\widehat{P}_{k|k}, \phi\big) = \int \phi(\mathbf{X}_{0:k}) P(d\mathbf{X}_{0:k}|\mathbf{Y}_{1:k})$, and $\|\phi\| = \sup_{\mathbf{X}_{0:k}} |\phi(\mathbf{X}_{0:k}|$.

In order to ensure Eq. (8.145) holds, the number of particles N_s needs to increase over time since it depends on $C_{k|k}$, a term that further relies on N_x. As discussed in [Crisan et al., 1999a], in order to assure the uniform convergence, both $C_{k|k}$ and the approximation error accumulates over the time.

In a high-dimensional space (order of tens or higher), particle filters still suffer the problem of curse of dimensionality [Bellman, 1961].

Suppose the minimum number is determined by the effective volume (variance) of the search space (proposal) against the target space (posterior). If the proposal and posterior are uniform in two N_x-dimensional hyperspheres with radii r and R ($R > r$) respectively, the effective particle number N_{eff} is approximately measured by the the volume ratio in the proposal space against posterior space, namely

$$N_{eff} \approx N_s * \big(\frac{r}{R}\big)^{N_x} \tag{8.146}$$

when the ratio is low ($r \ll R$), the effective number decreases exponentially as N_x increases; on the other hand, if we want to keep the effective number as a constant, we need to increase N_s exponentially as N_x increases [Chopin, 2002].

An important asymptotic result is the error bound of the filter. According to the Cramer-Rao theorem, the expected square error of an estimate is generally given by : [Simandl et al. , 2001]

$$\mathcal{E}(\mathbf{x}) = \mathbb{E}\big[(\mathbf{x} - \widehat{\mathbf{x}})^2\big] \tag{8.147}$$

$$\geq \frac{\Big[1 + \frac{d\mathbb{E}[\mathbf{x} - \widehat{\mathbf{x}}]}{d\mathbf{x}}\Big]^2}{\mathbf{J}(\mathbf{x})} + \big(\mathbb{E}\big[\mathbf{x} - \widehat{\mathbf{x}}\big]\big)^2, \tag{8.148}$$

where $\mathbf{J}(\mathbf{x})$ is the Fisher information matrix defined by

$$\mathbf{J}(\mathbf{x}) = \mathbb{E}\Big[\Big(\frac{\partial}{\partial \mathbf{x}} \log p(\mathbf{x}, \mathbf{y})\Big)\Big(\frac{\partial}{\partial \mathbf{x}} \log p(\mathbf{x}, \mathbf{y})\Big)^T\Big]. \tag{8.149}$$

If the estimate is unbiased $\mathbb{E}[\mathbf{x} - \widehat{\mathbf{x}}] = 0$, then $\mathcal{E}(\mathbf{x})$ is equal to the variance, and Eq. [8.146] reduces to

$$\mathcal{E}(x) \geq \mathbf{J}^{-1}(\mathbf{x}), \tag{8.150}$$

and the estimate satisfying Eq. [8.150] is called Fisher efficient.

8.9.2 Bias-Variance

Let's first consider the exact Monte Carlo sampling. The true and Monte Carlo state-error covariance matrices are defined by

$$\boldsymbol{\Sigma} = \mathbb{E}_p\big[(\mathbf{x} - \boldsymbol{\mu})(\mathbf{x} - \boldsymbol{\mu})^T\big], \tag{8.151}$$

$$\boldsymbol{\Sigma}_{\widehat{\mu}} = \mathbb{E}_p\big[(\mathbf{x} - \widehat{\boldsymbol{\mu}})(\mathbf{x} - \widehat{\boldsymbol{\mu}})^T\big], \tag{8.152}$$

where $\boldsymbol{\mu} = \mathbb{E}_p\big[\mathbf{x}\big]$, and $\widehat{\boldsymbol{\mu}} = \frac{1}{N_s}\sum_{i=1}^{N_s} \mathbf{x}^{(i)}$, where $\big\{\mathbf{x}^{(i)}\big\}$ are i.i.d. samples drawn from true pdf $p(\mathbf{x})$ and we get

$$\boldsymbol{\Sigma}_{\widehat{\mu}} = \big(1 + \frac{1}{N_s}\big)\boldsymbol{\Sigma} \tag{8.153}$$

$$= \boldsymbol{\Sigma} + \mathrm{Var}_p[\boldsymbol{\mu}], \tag{8.154}$$

where the second line follows the fact that

$$\mathbb{E}_p\big[(\boldsymbol{\mu} - \widehat{\boldsymbol{\mu}})(\boldsymbol{\mu} - \widehat{\boldsymbol{\mu}})^T\big] = \frac{1}{N_s}\boldsymbol{\Sigma}. \tag{8.155}$$

Hence the uncertainty from the exact Monte Carlo sampling part is the order of N_s^{-1}. In practice, we usually calculate the sample variance in place of the true variance, for Monte Carlo simulations we have :

$$\widehat{\boldsymbol{\Sigma}}_{\widehat{\mu}} = \frac{1}{N_s - 1}\sum_{i=1}^{N_s}\big(\widehat{\boldsymbol{\mu}} - \mathbf{x}^{(i)}\big)\big(\widehat{\boldsymbol{\mu}} - \mathbf{x}^{(i)}\big)^T, \tag{8.156}$$

and $\widehat{\boldsymbol{\Sigma}}_{\widehat{\mu}}$ is an unbiased estimate of $\boldsymbol{\Sigma}$ instead of $\boldsymbol{\Sigma}_{\widehat{\mu}}$, the unbiased estimate of $\boldsymbol{\Sigma}_{\widehat{\mu}}$ is given by $\big(1 + N_s^{-1}\big)\widehat{\boldsymbol{\Sigma}}_{\widehat{\mu}}$.

In practice, with moderate sample size the bias is not negligible, and this bias accounts for the following sources: limited simulated samples, limited computing power and limited memory (calculation of posterior $p(\mathbf{X}_{0:k}|\mathbf{Y}_{0:k})$ needs storing the data up to k), not to mention the sampling inaccuracy as well as the existence of noise [Kim , 2002].

Generally we can define the bias and the variance of importance sampling or MCMC estimate as [Mark and Baram , 2001]:

$$\text{Bias} = \mathbb{E}_\pi\big[\widehat{f}(\mathbf{x})\big] - \mathbb{E}_p\big[f(\mathbf{x})\big], \tag{8.157}$$

$$\text{Var} = \mathbb{E}_\pi\Big[\big(\widehat{f}(\mathbf{x}) - \mathbb{E}_\pi\big[\widehat{f}(\mathbf{x})\big]\big)^2\Big] \tag{8.158}$$

where $\widehat{f}(\mathbf{x})$ is given by the weighted importance sampling. The quality of approximation is measured by a loss function $\mathcal{E}$, as decomposed by

$$\mathcal{E} = \mathbb{E}_\pi\Big[\Big(\widehat{f}(\mathbf{x}) - \mathbb{E}_p\big[\widehat{f}(\mathbf{x})\big]\Big)^2\Big] \tag{8.159}$$

$$= \text{Bias}^2 + \text{Var}. \tag{8.160}$$

8.9.3 Robustness

Algorithmic robustness and numerical robustness are important for the discrete-time filtering. In many practical scenarios, the filter might encounter the possibility of divergence where the algorithmic assumption is violated or the numerical problem is encountered (e.g., ill-conditioned matrix factorization).

There are two fundamental problems concerning the robustness in particle filters,

- when there is an outlier the importance weights will be very unevenly distributed and it usually requires a large number of N_s particles to assure the accuracy of empirical density approximation. Hence the measurement density $p(\mathbf{y}_k|\mathbf{x}_k)$ is supposed to insensitive to the $\mathbf{x}_k$.

- the empirical distribution from the samples often approximates poorly for the long-tailed distribution, either for proposal distribution or for posterior. This is imaginable because the probability sampling from the tail part of distribution is very low, and resampling somehow makes this problem more severe.

9

Stochastic Differential Equations and Filtering

This chapter gives a description of Stochastic Differential Equations applied to curves, and Stochastic Bayes' Filters in Continuous-Discrete Time.

9.1 Introduction

In chapter [5] we built a "Model of Trading" in Continuous-Discrete time for Stopping Time prediction of an asset, with Stop-Loss. In section [5.3] we described the procedure of forecasting with Particle and Kalman-Bucy filters in continuous-discrete time.

The purpose of this chapter is to introduce the general concepts involved in Bayesian estimation for learning and generalization of non stationary, nonlinear systems in a Dynamic State Space representation with Particle and Kalman-Bucy filters for parameter estimation and forecasting in an auto-adaptive procedure in continuous-discrete time.

We consider a system described by a stochastic process called the state process whose behaviors is governed by a dynamic system containing a Brownian motion noise input. This is observed, together with unavoidable observation noise.

The goal of the stochastic filtering problem is to determine the state given the observations, and this must be formalized in the language of stochastic calculus.

This chapter is organized as follow:

- in section [9.2] we introduce the Stochastic Differential Equations,
- in section [9.3] we describe the Continuous-Discrete Time Filtering Kalman-Bucy filter algorithms for initial parameter estimation and Particle filter algorithms for forecasting of the first stopping-time of a

"Model of Trading" with stop-loss, in continuous-discrete time, in a joint filtering procedure, as used in section [5.3].

9.2 Stochastic Differential Equations (SDE)

9.2.1 Models

We will use Stochastic Differential Equations for the modeling of financial systems dynamics, Continuous-Discrete Time Filtering where we have a Stochastic Differential Equation, in continuous-time, for the state dynamics, and a Stochastic Difference Equation, in discrete-time, for the observations.

State Process

Let $\{\mathbf{x}(t), \mathcal{F}_t, t \geq 0\}$ be the state process. It is defined to be solution of the stochastic differential equation, in continuous time:

$$d\mathbf{x}(t) = \mathbf{f}(\mathbf{x}(t), t)dt + \mathbf{L}(\mathbf{x}(t), t)d\boldsymbol{\beta}(t), \tag{9.1}$$

where

- $\mathbf{x}(t) \in \mathbb{R}^n$ is the state,
- $\mathbf{f} : \mathbb{R}^n \times \mathbb{R}_+ \mapsto \mathbb{R}^n$ is the drift function,
- $\mathbf{L}(\mathbf{x}(t), t) \in \mathbb{R}^{n \times s}$ is the dispersion matrix,
- $\boldsymbol{\beta}(t) \in \mathbb{R}^s$ is Brownian motion (see Appendix: [A.1.8]), with diffusion matrix, $\mathbf{Q}_c(t) \in \mathbb{R}^{s \times s}$,
- $\mathcal{F}_t$ is a filtration (see Appendix: [A.1.4].

Observation Process

We consider an observation process which satisfies the stochastic difference equation in discrete time:

$$\mathbf{y}_k = \mathbf{h}_k(\mathbf{x}(t_k)) + \mathbf{n}_k \tag{9.2}$$

where:

- $\mathbf{x}(t) \in \mathbb{R}^n$ is the state,
- $\mathbf{y}_k$ is the observation at time t_k,
- $\mathbf{n}_k$ is the measurement noise at time t_k, with $\mathbf{n}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{R})$.

9.2.2 Filtering Problem

We consider a physical system whose state is represented by the vector $\mathbf{x}(t)$, and governed by a stochastic differential equation with a noise term $\boldsymbol{\beta}$ representing random perturbations to the system. We have also the observation process $\mathbf{y}(t)$ which includes new independent noise $\boldsymbol{\nu}$.

The filtering problem is to find the conditional law of the signal process given the observations, i.e. to find:

$$p\big(\mathbf{x}, t \mid \mathbf{y}_{1:k}\big) \quad \forall t \in [0, T]. \tag{9.3}$$

which is the complete solution of the filtering problem, because $p\big(\mathbf{x}, t \mid \mathbf{y}_{1:k}\big)$ embodies all statistical information about $\mathbf{x}(t)$ which is contained in the available observations, $\mathbf{y}_{1:k}$, and in the initial condition $\big(\mathbf{x}, t_0\big)$.

9.2.3 Financial Processes

For financial processes we have to use different representations of their dynamics, depending on their stationarity or non-stationarity, and if the stochastic terms depend on state.

1. **Non-Stationary Process in Continuous-discrete time I**

   For a non-stationary process in Continuous-Discrete time, and if the stochastic term does not depend on the state, we use the following formulations for a non-linear process and the linear Kalman filter:

   Continuous-Discrete State Space Models I

   $$d\mathbf{x}(t) = \mathbf{f}(\mathbf{x}, t)dt + \mathbf{L}(t)d\boldsymbol{\beta}(t) \tag{9.4}$$
   $$\mathbf{y}_k \sim p(\mathbf{y}_k|\mathbf{x}(t_k)). \tag{9.5}$$

   Continuous-Discrete Kalman Filter I

   $$d\mathbf{x}(t) = \mathbf{F}(t)\mathbf{x}(t)dt + \mathbf{L}(t)d\boldsymbol{\beta}(t) \tag{9.6}$$
   $$\mathbf{y}_k = \mathbf{H}_k\mathbf{x}(t_k) + \mathbf{r}_k. \tag{9.7}$$

2. **Non-Stationary Process in Continuous-discrete time II**

   For a non-stationary process in Continuous-Discrete time, and if the stochastic term depends on the state, we use the following formulations for a non-linear process and the linear Kalman filter:

Continuous-Discrete State Space Models II

$$d\mathbf{x}(t) = \mathbf{f}(\mathbf{x}, t)dt + \mathbf{L}(\mathbf{x}, t)d\boldsymbol{\beta}(t) \tag{9.8}$$
$$\mathbf{y}_k \sim p(\mathbf{y}_k|\mathbf{x}(t_k)). \tag{9.9}$$

Continuous-Discrete Kalman Filter II

$$d\mathbf{x}(t) = \mathbf{F}(t)\mathbf{x}(t)dt + \mathbf{L}(\mathbf{x}, t)d\boldsymbol{\beta}(t) \tag{9.10}$$
$$\mathbf{y}_k = \mathbf{H}_k\mathbf{x}(t_k) + \mathbf{r}_k. \tag{9.11}$$

9.2.4 White noise

In modeling any physical process we proceed by defining the variables describing the process and connecting them via certain causal relationships or physical laws. But we have also some unpredictable fluctuations in the process for which no causal relationships exist. Usually we represent these fluctuations by a *white noise*.

In Discrete-time:

We can define a white random sequence $\{x_k, k = 1, 2, \cdots\}$ as a Markov sequence for which

$$p(x_k|x_l) = p(x_k) \quad (k > l). \tag{9.12}$$

Knowing the realization x_l in no way helps in predicting what x_k will be. A white sequence is completely random or totally unpredictable. If the x_k are all normally distributed, the sequence $\{x_k\}$ is called a *white Gaussian random sequence*. Its distribution is specified by

$$\overline{\{x_k\}} = \mathbb{E}\Big[\{x_k\}\Big] \tag{9.13}$$
$$\begin{aligned} Cov\Big(\{x_k\}, \{x_l\}\Big) &= \mathbb{E}\Big[\Big(\{x_k\} - \overline{\{x_k\}}\Big)\Big(\{x_l\} - \overline{\{x_l\}}\Big)^T\Big] \\ &= \mathbf{Q}_k\delta_{k,l} \end{aligned} \tag{9.14}$$

In Continuous-time :

Since the white Gaussian sequence serves as a good model for the noise in discrete physical process, its continuous analog may serve as a good model for the noise in continuous dynamical systems. We might define a white process $\{x(t),\ t \in T\}$ as a Markov process for which

$$p\big(x(t)|x(\tau)\big) = p\big(x(t)\big) \quad t > \tau \in T. \tag{9.15}$$

If the $x(t)$ are normally distributed for each $t \in T$, then the process is a white Gaussian process with

$$\begin{aligned} Cov\Big(x(t), x(\tau)\Big) &= \mathbb{E}\Big[\Big(x(t) - \overline{x(t)}\Big)\Big(x(\tau) - \overline{x(\tau)}\Big)^T\Big] \\ &= \mathbf{Q}(t)\delta(t-\tau), \end{aligned} \tag{9.16}$$

where $\delta(t-\tau)$ is the Dirac delta function. But, since the Dirac delta function is not an ordinary function, in continuous-time, the white Gaussian process is a mathematical fiction.

Moreover, if we consider the power spectral density function of this process, it must be constant at all frequencies, that requires infinite power and, therefore, continuous-time white Gaussian noise is not physically realizable [Jazwinski, 1970].

Thus in continuous-time, we have to represent the unpredictable fluctuations by a Brownian motion process or Wiener process.

9.2.5 Motivation

Many dynamic processes in engineering, physics, finance,... can be modeled as Stochastic Differential Equations (SDE) with an unknown driving function $\mathbf{w}(t)$ as follows:

$$\frac{d\mathbf{x}}{dt} = \mathbf{f}(\mathbf{x}, t) + \mathbf{L}(\mathbf{x}, t)\mathbf{w}(t) \tag{9.17}$$

The unknown function $\mathbf{w}(t)$ would be ideally modeled as a process that is Gaussian and completely "white" in the sense that $\mathbf{w}(t)$ and $\mathbf{w}(s)$ are independent for all $t \neq s$. However, the problem is that this kind of process cannot exists in any mathematically or physically meaningful sense [Oksendal , 2003] [Karlin and Taylor, 1975].

The solution to this existence problem is that actually the white process does not need to exist as long as its integral exists. Integrating the Eq. (9.17) once with respect to time gives the stochastic integral equation

$$\mathbf{x}(t) - \mathbf{x}(s) = \int_s^t \mathbf{f}(\mathbf{x}, t')dt' + \int_s^t \mathbf{L}(\mathbf{x}, t')\mathbf{w}(t')dt' \tag{9.18}$$

The first integral is a Lebesgue integral and does not cause any problems, but the second integral is problematic because of the appearance of white noise

process. Fortunately, this integral can be defined to be an Ito integral with respect to the stochastic "measure" $\boldsymbol{\beta}(t)$, which has independent Gaussian increments:

$$\int_s^t \mathbf{L}(\mathbf{x}, t')\mathbf{w}(t')dt' \doteq \int_s^t \mathbf{L}(\mathbf{x}, t')d\boldsymbol{\beta}(t'). \tag{9.19}$$

White noise is then, at least in formal sense, the time derivative of the Brownian motion $\mathbf{w}(t) = d\boldsymbol{\beta}(t)/dt$.

Because by a stochastic differential equation it is actually meant the corresponding stochastic integral equation, this point is emphasized by writing stochastic differential equations in form:

$$d\mathbf{x}(t) = \mathbf{f}(\mathbf{x}, t)dt + \mathbf{L}(\mathbf{x}, t)d\boldsymbol{\beta}(t). \tag{9.20}$$

where the usage of the problematic white noise process is avoided.

9.3 Continuous-Discrete Time Filtering

9.3.1 Introduction

In chapter [5] we built a "Model of Trading" in Continuous-Discrete time for Stopping Time prediction of an asset, with Stop-Loss. In section [5.3] we described the procedure of forecasting with Particle and Kalman-Bucy filters in continuous-Discrete time.

The goal of this chapter is to introduce the general concepts involved in Stochastic Bayesian estimation for learning and generalization of non stationary, nonlinear systems in a Dynamic State Space representation with Particle and Kalman-Bucy filters for parameter estimation and forecasting in an auto-adaptive procedure.

This chapter is organized as follow:

- we describe the Model,
- we formulate the *Filtering distribution*,
- we review classical methods of continuous-discrete filtering, such as the *Continuous-Discrete Kalman-Bucy Filter*,
- we shown how the *Unscented Kalman-Bucy Filter* can be applied to nonlinear continuous-discrete filtering problems. This continuous-discrete Unscented Kalman-Bucy Filter is based on a matrix form of the *Unscented Transform* [Julier, 1997],

- we describe the Continuous-Discrete Sequential Importance Sampling algorithm, based on application of the *Girsanov* theorem (Appendix [A.5.2]),

The dynamics of processes are modeled as *Itô Stochastic Differential Equations* (SDE) driven by Brownian motions and measurements are modeled as non-linear functions of the state, which are corrupted by Gaussian measurement noises.

The main references for this section are:

- most of the continuous-discrete filtering problems considered in this section have the same form as in the classic book of [Jazwinski, 1970],
- the stochastic processes, with chemical and physical applications, are described in [Gardiner, 1990]; and [Horsthemke and Lefever, 1984], in an engineer formulation,
- an explicit description of UKF can be found in [Doucet, 1998]; [Doucet, et al., 2001]; [van der Merwe , 2004]; [Wan and van der Merwe , 2000]; and [Wan and van der Merwe , 2001],
- the description of UKF in matrix formulation, and algorithms comes from [Sarkka , 2006],
- an explicit description of Monte Carlo Filters is given in [Fearnhead , 1998],
- the numerical methods are described in the classical books of [Kloeden et al., 1992]; and [Kloeden and Platen, 1992],
- the stochastic algorithms are adapted from [Gilsing and Shardlow , 2005]; [Cyganowski et al. , 2005]; and [Picchini , 2007],
- the particle filter algorithms are adapted from [Rimmer et al. , 2005].

9.3.2 Model

A continuous-discrete state space model is as follow:

$$d\mathbf{x} = \mathbf{f}(\mathbf{x}, t)dt + \mathbf{L}(t)d\boldsymbol{\beta}(t) \tag{9.21}$$

$$\mathbf{y}_k \sim p(\mathbf{y}_k|\mathbf{x}(t_k)), \tag{9.22}$$

where

- $\mathbf{x}(t) \in \mathbb{R}^n$ is the state,
- $\mathbf{y}_k \in \mathbb{R}^m$ is the measurement obtained at time instance t_k.
- $\mathbf{f} : \mathbb{R}^n \times \mathbb{R}_+ \mapsto \mathbb{R}^n$ is the drift function,
- $\mathbf{L}(t) \in \mathbb{R}^{n \times s}$ is the dispersion matrix,
- $\boldsymbol{\beta}(t) \in \mathbb{R}^s$ is Brownian motion with diffusion matrix, $\mathbf{Q}_c(t) \in \mathbb{R}^{s \times s}$,

- $p(\mathbf{y}_k|\mathbf{x}(t_k))$ defines the likelihood distribution of measurement $\mathbf{y}_k$ given the state $\mathbf{x}(t_k)$. It depends on the measurement model (linear or nonlinear), and the distribution of the state (Gaussian or other).

In estimation context the Eq. (9.21) is often stated in terms of a white noise process $\mathbf{w}(t)$ as

$$\frac{d\mathbf{x}}{dt} = \mathbf{f}(\mathbf{x}, t) + \mathbf{L}(t)\mathbf{w}(t), \tag{9.23}$$

where the white noise is defined as the derivative of the Brownian motion $\mathbf{w}(t) = d\boldsymbol{\beta}/dt$.

But a continuous white noise cannot exist in the mathematical sense, because Brownian motion is nowhere differentiable. For this reason the integral equation formulation of the SDE as in Eq. (9.21) is often used in mathematical analysis.

However, in engineering and physics applications, models are much easier to formulate in terms of white noise, and fortunately all sensible models involving white noise can also be interpreted in terms of Brownian motion.

9.3.3 Filtering Equations

The purpose is to estimate the conditional filtering distribution $p(\mathbf{x}(t_k)|\mathbf{y}_{1:k})$, given the observations, from the state equation Eq. (9.21), and the likelihood distribution Eq. (9.22).

Given the state transition distribution $p(\mathbf{x}(t_k)|\mathbf{x}(t_{k-1}))$ and the likelihood distribution $p(\mathbf{y}_k|\mathbf{x}(t_k))$, we obtain the conditional filtering distribution $p(\mathbf{x}(t_k)|\mathbf{y}_{1:k})$.

The filtering distribution can be defined for all $t \in \mathbb{R}_+$, not only at times t_k, as follow:

1. At times t_k we know the conditional filtering distribution of the state $\mathbf{x}(t_k)$ given the measurements $\mathbf{y}_{1:k}$:

$$p\big(\mathbf{x}(t_k)|\mathbf{y}_{1:k}\big). \tag{9.24}$$

2. At any time t, such as $t_k < t < t_{k+1}$ the filtering distribution of $\mathbf{x}(t)$ is the distribution obtained from $p\big(\mathbf{x}(t_k)|\mathbf{y}_{1:k}\big)$ by prediction to time t:

$$p\big(\mathbf{x}(t)|\mathbf{y}_{1:k}\big) = \int p\big(\mathbf{x}(t)|\mathbf{x}(t_k)\big)\, p\big(\mathbf{x}(t_k|\mathbf{y}_{1:k}\big) d\mathbf{x}(t_k). \tag{9.25}$$

9.3.4 Continuous-Discrete Kalman-Bucy Filter

In the continuous-discrete Kalman-Bucy filter (see, e.g., [Jazwinski, 1970])

- the dynamic model is a linear stochastic differential equation (see, Appendix [A.4.2]),
- the measurements are observed at discrete instances of time.

and we use the continuous-discrete time model:

$$d\mathbf{x}(t) = \mathbf{F}(t)\mathbf{x}(t)dt + \mathbf{L}(t)d\boldsymbol{\beta}(t) \tag{9.26}$$

$$\mathbf{y}_k = \mathbf{H}_k\mathbf{x}(t_k) + \mathbf{r}_k, \tag{9.27}$$

where

- $\mathbf{F}(t)$ and $\mathbf{L}(t)$ are time dependent matrices,
- $\boldsymbol{\beta}(t)$ is a Brownian motion with diffusion matrix $\mathbf{Q}_c(t)$,
- $\mathbf{H}_k$ is a time dependent matrix,
- $\mathbf{r}_k \sim N(\mathbf{0}, \mathbf{R}_k)$,
- The stochastic process $\mathbf{x}(t)$ has the initial distribution $\mathbf{x}(0) \sim N(\mathbf{m}(0), \mathbf{P}(0))$.

As shown in Appendix: [A.4.2] the solution $\mathbf{x}(t)$ is a Gaussian process with its mean and covariance given by the differential equations Eq. (A.61) and Eq. (A.62).

We can convert the continuous-time linear dynamic model in Eq. (9.26) and Eq. (9.27) into the equivalent discrete model, using Theorem: [A.4.3], and we consider two algorithms.

We use these algorithms for initial parameter estimation for stationary, linearized "Model of Trading" in Continuous-Discrete time, as describes in section [5.3].

Algorithm 9.3.1 (Continuous-discrete Kalman-Bucy filter I) .

In this Kalman-Bucy filter algorithm, we have three steps based on Theorem [A.4.3]):

1. ***Discretization:***
   Solve the discrete-time model matrices
   - $\mathbf{A}_{k-1} \doteq \mathbf{A}(t_k)$,
   - $\mathbf{Q}_{k-1} \doteq \mathbf{Q}(t_k)$

from the differential equations

$$\frac{d\mathbf{A}(t)}{dt} = \mathbf{F}(t)\mathbf{A}(t) \tag{9.28}$$

$$\frac{d\mathbf{Q}(t)}{dt} = \mathbf{F}(t)\mathbf{Q}(t) + \mathbf{Q}(t)\mathbf{F}^T(t) + \mathbf{L}(t)\mathbf{Q}_c(t)\mathbf{L}^T(t). \tag{9.29}$$

with initial conditions $\mathbf{A}(t_{k-1}) = \mathbf{I}$ *and* $\mathbf{Q}(t_{k-1}) = \mathbf{0}$.

The transition density is given by: (see: [A.4.3])

$$p(\mathbf{x}(t_k)|\mathbf{x}(t_{k-1})) = \mathcal{N}(\mathbf{x}(t_k)|\mathbf{A}_{k-1}\mathbf{x}(t_{k-1}), \mathbf{Q}_{k-1}), \tag{9.30}$$

and we apply the discrete-time Kalman filter equations:

2. ***Prediction****:*

$$\mathbf{m}_{k|k-1} = \mathbf{A}_{k-1}\mathbf{m}_{k-1|k-1} \tag{9.31}$$

$$\mathbf{P}_{k,k-1} = \mathbf{A}_{k-1}\mathbf{P}_{k-1|k-1}\mathbf{A}_{k-1}^T + \mathbf{Q}_{k-1}. \tag{9.32}$$

3. ***Update****:*

$$\mathbf{e}_k = \mathbf{y}_k - \mathbf{H}_k\mathbf{m}_{k|k-1} \tag{9.33}$$

$$\mathbf{S}_k = \mathbf{H}_k\mathbf{P}_{k|k-1}\mathbf{H}_k^T + \mathbf{R}_k \tag{9.34}$$

$$\mathbf{K}_k = \mathbf{P}_k\mathbf{H}_k^T\mathbf{S}_k^{-1} \tag{9.35}$$

$$\mathbf{m}_{k|k} = \mathbf{m}_{k|k-1} + \mathbf{K}_k\mathbf{e}_k \tag{9.36}$$

$$\mathbf{P}_{k|k} = \mathbf{P}_{k|k-1} - \mathbf{K}_k\mathbf{S}_k\mathbf{K}_k^T. \tag{9.37}$$

If the dynamic model in Eq. (9.26) and Eq. (9.27) is linear and time invariant (LTI), the matrices $\mathbf{F}$ and $\mathbf{L}$ do not depend on time, and the discrete model matrices will depend only on the time difference, and they can be then solved in closed form or by numerical methods (see, Theorem [A.4.4]).

- $\Delta t_{k-1} = t_k - t_{k-1}$,
- $\mathbf{A}_{k-1} = \mathbf{A}(\Delta t_{k-1})$,
- $\mathbf{Q}_{k-1} = \mathbf{Q}(\Delta t_{k-1})$.

Algorithm 9.3.2 (Continuous-discrete Kalman-Bucy filter II) .

This continuous-discrete Kalman-Bucy filter consists in two steps based on Theorem [A.4.2]:

1. ***Prediction***:
   First, we realize the integrations of differential equations:

$$\frac{d\mathbf{m}(t)}{dt} = \mathbf{F}(t)\ \mathbf{m}(t) \tag{9.38}$$
$$\frac{d\mathbf{P}(t)}{dt} = \mathbf{F}(t)\mathbf{P}(t) + \mathbf{P}(t)\mathbf{F}^T(t) + \mathbf{L}(t)\ \mathbf{Q}_c(t)\ \mathbf{L}^T(t). \tag{9.39}$$

   with initial conditions

$$\mathbf{m}(t_{k-1}) \doteq \mathbf{m}_{k-1} \tag{9.40}$$
$$\mathbf{P}(t_{k-1}) \doteq \mathbf{P}_{k-1} \tag{9.41}$$

   to get the predicted mean and covariance to time instance t_k:

$$\mathbf{m}_{k|k-1} = \mathbf{m}(t_k) \tag{9.42}$$
$$\mathbf{P}_{k|k-1} = \mathbf{P}(t_k). \tag{9.43}$$

2. ***Update***:
   The updating step is the same as for Kalman filter.

$$\mathbf{e}_k = \mathbf{y}_k - \mathbf{H}_k\mathbf{m}_{k|k-1} \tag{9.44}$$
$$\mathbf{S}_k = \mathbf{H}_k\mathbf{P}_{k|k-1}\mathbf{H}_k^T + \mathbf{R}_k \tag{9.45}$$
$$\mathbf{K}_k = \mathbf{P}_k\mathbf{H}_k^T\mathbf{S}_k^{-1} \tag{9.46}$$
$$\mathbf{m}_{k|k} = \mathbf{m}_{k|k-1} + \mathbf{K}_k\mathbf{e}_k \tag{9.47}$$
$$\mathbf{P}_{k|k} = \mathbf{P}_{k|k-1} - \mathbf{K}_k\mathbf{S}_k\mathbf{K}_k^T. \tag{9.48}$$

Remark 9.3.1 () .

In these continuous-discrete Kalman-Bucy filter algorithms, the results of filtering are the mean $\mathbf{m}(t)$, and the covariance $\mathbf{P}(t)$, defined for all t, when the filtering result is interpreted in a generalized sense. But these functions are not continuous at the measurement times.

The filtering solution is then of the form,

$$p\big(\mathbf{x}(t)|\mathbf{y}_{1:k}\big) = \mathcal{N}\big(\mathbf{x}(t)|\mathbf{m}(t), \mathbf{P}(t)\big), \tag{9.49}$$

where k is such that $t \in [t_k, t_{k+1}]$.

9.3.5 Continuous-Discrete Unscented Kalman-Bucy Filter

The method is based on continuous-time version of the *Unscented Transform* (UT), (see [Sarkka , 2006]; [Wan and van der Merwe , 2001], and Algorithm [H.5.1]).

Theorem 9.3.1 (Unscented approximation of SDEs) .

The continuous-time unscented transform based Gaussian process approximation has the following differential equations for the mean and covariance

$$\frac{d\mathbf{m}}{dt} = \mathbf{f}(\mathbf{X}(t), t)\mathbf{w}_m, \tag{9.50}$$

$$\frac{d\mathbf{P}}{dt} = \mathbf{f}(\mathbf{X}(t), t)\mathbf{W}\mathbf{X}^T(t) + \mathbf{X}(t)\mathbf{W}\mathbf{f}^T(\mathbf{X}(t), t) + \mathbf{L}(t)\mathbf{Q}_c(t)\mathbf{L}^T(t). \tag{9.51}$$

The expression for the sigma point matrix $\mathbf{X}(t)$ is given as

$$\mathbf{X}(t) = \big[\mathbf{m}(t) \cdots \mathbf{m}(t)\big] + \sqrt{c}\big[0 \quad \sqrt{\mathbf{P}(t)} \quad -\sqrt{\mathbf{P}(t)}\big], \tag{9.52}$$

where $\sqrt{\mathbf{P}(t)}$ is a matrix square root of $\mathbf{P}(t)$ (e.g., Cholesky factor) and vector $\mathbf{w}_m$ and matrix $\mathbf{W}$ are defined as

$$\mathbf{w}_m = \Big[w_0^{(m)}\ w_1^{(m)} \cdots w_{2L}^{(m)}\Big]^T \tag{9.53}$$

$$\mathbf{W}_c = diag\Big(w_0^{(c)}\ ,\ w_1^{(m)}, \cdots, w_{2L}^{(m)}\Big) \tag{9.54}$$

$$\mathbf{W} = \Big[\mathbf{I} - \mathbf{I}\big(\mathbf{w}_m \cdots \mathbf{w}_m\big)\Big]\mathbf{W}_c\Big[\mathbf{I} - \mathbf{I}\big(\mathbf{w}_m \cdots \mathbf{w}_m\big)\Big]^T, \tag{9.55}$$

where the weights $w_k^{(m)}$ and $w_k^{(c)}$ are defined in Equations [8.46], [8.47], and [8.48].

see [Sarkka , 2006]) for proof and detailed description.

9.3.6 Continuous-Discrete Unscented Kalman-Bucy Filter

We use the Continuous-Discrete Unscented Kalman-Bucy Filter for initial parameter estimation for stationary, nonlinear "Model of Trading" in Continuous-Discrete time, as describes in section [5.3].

Algorithm 9.3.3 (Continuous-Discrete Unscented Kalman-Bucy Filter) .

The continuous-discrete unscented Kalman-Bucy filter consists of two steps:

1. ***Prediction***.
We integrate the differential equations:

$$\mathbf{X}(t) = \Big[\mathbf{m}(t) \cdots \mathbf{m}(t)\Big] + \sqrt{c}\Big[\mathbf{0} \quad \sqrt{\mathbf{P}(t)} \quad -\sqrt{\mathbf{P}(t)}\Big] \tag{9.56}$$

$$\frac{d\mathbf{m}}{dt} = \mathbf{f}(\mathbf{X}(t), t)\mathbf{w}_m \tag{9.57}$$

$$\frac{d\mathbf{P}}{dt} = \mathbf{f}(\mathbf{X}(t), t)\mathbf{W}\mathbf{X}^T(t) + \mathbf{X}(t)\mathbf{W}\mathbf{f}^T(\mathbf{X}(t), t) + \mathbf{L}(t)\mathbf{Q}_c(t)\mathbf{L}^T(t). \tag{9.58}$$

from the initial conditions to time instance t_k
- $\mathbf{m}(t_{k-1}) = \mathbf{m}_{k-1}$,
- $\mathbf{P}(t_{k-1}) = \mathbf{P}_{k-1}$

to get the predicted mean and covariance:
- $\mathbf{m}_{k|k-1} = \mathbf{m}(t_k)$,
- $\mathbf{P}_{k|k-1} = \mathbf{P}(t_k)$,

2. ***Update***.
The update step is the same as for discrete-time unscented Kalman filter (see: Algorithm [H.5.1]),

$$\mathbf{X}_{k|k-1} = \Big[\mathbf{m}_{k|k-1} \cdots \mathbf{m}_{k|k-1}\Big] + \sqrt{c}\Big[\mathbf{0} \quad \sqrt{\mathbf{P}_{k|k-1}} \quad -\sqrt{\mathbf{P}_{k|k-1}}\Big] \tag{9.59}$$

$$\mathbf{Y}_{k|k-1} = \mathbf{h}(\mathbf{X}_{k|k-1}, k) \tag{9.60}$$

$$\boldsymbol{\mu}_k = \mathbf{Y}_{k|k-1}\mathbf{w}_m \tag{9.61}$$

$$\mathbf{S}_k = \mathbf{Y}_{k|k-1}\mathbf{W}\mathbf{Y}^T_{k|k-1} + \mathbf{R}_k \tag{9.62}$$

$$\mathbf{C}_k = \mathbf{X}_{k|k-1}\mathbf{W}\mathbf{Y}^T_{k|k-1} \tag{9.63}$$

$$\mathbf{K}_k = \mathbf{C}_k\mathbf{S}_k^{-1} \tag{9.64}$$

$$\mathbf{m}_{k|k} = \mathbf{m}_{k|k-1} + \mathbf{K}_k[\mathbf{y}_k - \boldsymbol{\mu}_k] \tag{9.65}$$

$$\mathbf{P}_{k|k} = \mathbf{P}_{k|k-1} - \mathbf{K}_k\mathbf{S}_k\mathbf{K}_k^T. \tag{9.66}$$

9.3.7 Continuous-Discrete Sequential Importance Resampling

For an optimal estimation of continuous-discrete filtering models we have two methods:

- a the bootstrap filter procedure, which is a simple method for approximating the optimal solution,

- a measure transform based methods for more general continuous-discrete sequential importance resampling.

In case of 'Particle Filters' in continuous time, 'particles' are no more random samples drawn from a probability distribution as in discrete time, but will come from 'paths' of the stochastic differential equation, and we have the steps:

1. we use particles $\mathbf{x}_{k-1}^{(i)}$ from time-step $(k-1)$ as initial conditions $(\mathbf{x}^{(i)}(t_{k-1}) = \mathbf{x}_{k-1}^{(i)})$ for resolution of the stochastic differential equation,
2. from these particles, we simulate the trajectories $\mathbf{x}^{(i)}(t)$ for $(t \in [t_{k-1}, t_k])$,
3. we obtain the particles at time-step k as $\mathbf{x}_k^{(i)} = \mathbf{x}^{(i)}(t_k)$,
4. these particles give an approximation of the state transition distribution $p(\mathbf{x}_k|\mathbf{x}_{k-1}^{(i)})$
5. we must use 'Likelihood ratio' of stochastic differential equations, using the Girsanov theorem [A.5.1], for estimation of importance weights associated to these paths.

We use these algorithms for prediction of the first stopping-time for the "Model of Trading" with stop-loss in Continuous-Discrete time, in a joint filtering procedure as describes in section [5.3].

Continuous-Discrete Bootstrap Filter

Let's consider the system:

$$d\mathbf{x} = \mathbf{f}(\mathbf{x}, t)dt + \mathbf{L}(\mathbf{x}, t)d\boldsymbol{\beta} \tag{9.67}$$

$$\mathbf{y}_k \sim p(\mathbf{y}_k|\mathbf{x}(t_k)), \tag{9.68}$$

A bootstrap filter is very easy to implement for the continuous-discrete filtering problems of the general form because samples from the transition density of the dynamic model can be easily generated by numerically simulating the stochastic differential equation (see, Appendix: [C.6.1]).

Algorithm 9.3.4 (Continuous-discrete bootstrap filter) .

Bootstrap filtering for continuous-time stochastic differential equation, and discrete observations, can be performed as follows:

1. *from the state equation we simulate the paths:*
   $\{\mathbf{x}^{(i)}(t) : t_{k-1} \leq t \leq t_k, \quad i = 1, \cdots, N\}$, *using:*

$$d\mathbf{x}^{(i)} = \mathbf{f}(\mathbf{x}^{(i)}, t)dt + \mathbf{L}(\mathbf{x}^{(i)}, t)d\boldsymbol{\beta}^{(i)}(t) \tag{9.69}$$

$$\mathbf{x}^{(i)}(t_{k-1}) = \mathbf{x}_{k-1}^{(i)}. \tag{9.70}$$

   with independent Brownian motions $\boldsymbol{\beta}^{(i)}(t)$,
2. *we set* $\mathbf{x}_k^{(i)} = \mathbf{x}^{(i)}(t_k)$.
3. *each* $\mathbf{x}_k^{(i)}$ *is a random draw from the transition distribution* $p(\mathbf{x}_k|\mathbf{x}_{k-1}^{(i)})$.
4. *using the likelihood, we compute the weights: likelihood:*

$$w_k^{(i)} \propto w_{k-1}^{(i)} p(\mathbf{y}_k|\mathbf{x}_k^{(i)}). \tag{9.71}$$

5. *we resample the particles from* $\{\mathbf{x}_k^{(i)} : i = 1, \cdots, N\}$.

The bootstrap filter is fine for "*Filtering*", even with multiplicative system noises, but it does not work for "*Parameter Estimation*". It suffers from the problem that using the dynamic state model as the importance distribution is not very efficient, and the bootstrap filter is likely to produce degenerate approximations if the dynamic model is not very accurate.

Some more efficient importance processes can be used in the continuous-discrete filtering problem.

Sequential Importance Resampling Filter for Continuous SDEs

We consider a restricted class of algorithm, very useful for the sequential importance resampling of more general SDEs.

Let the state equation of the SDE system:

$$d\mathbf{x} = \mathbf{f}(\mathbf{x}, t)dt + \mathbf{L}d\boldsymbol{\beta}. \tag{9.72}$$

and an importance process given by equation:

$$d\mathbf{s} = \mathbf{g}(\mathbf{s}, t)dt + \mathbf{B}d\boldsymbol{\beta}, \tag{9.73}$$

where $\mathbf{L}$, and $\mathbf{B}$ are time independent and invertible. In this case, the probability measures of $\mathbf{x}$ and $\mathbf{s}$ are absolutely continuous with respect to the probability measure of the driving Brownian motion $\boldsymbol{\beta}$.

The likelihood ratio of the processes can be computed as follows:

Theorem 9.3.2 (Likelihood ratio of SDEs) .

Assume that the processes $\mathbf{x}(t)$ *and* $\mathbf{s}(t)$ *are generated by the stochastic differential equations*

$$d\mathbf{x} = \mathbf{f}(\mathbf{x}, t)dt + \mathbf{L}d\boldsymbol{\beta}, \tag{9.74}$$

$$d\mathbf{s} = \mathbf{g}(\mathbf{s}, t)dt + \mathbf{B}d\boldsymbol{\beta}, \tag{9.75}$$

with $\mathbf{x}(0) = \mathbf{x}_0$ *and* $\mathbf{s}(0) = \mathbf{x}_0$*, where*

- $\mathbf{f}(\mathbf{x}, t)$ *and* $\mathbf{g}(\mathbf{s}, t)$ *are bounded and measurable,*
- $\mathbf{L}$ *and* $\mathbf{B}$ *are invertible matrixes,*
- $\boldsymbol{\beta}(t)$ *is a Brownian motion with respect to measure* P*.*

where Eq. (9.74) *is the process and Eq.* (9.75) *is the corresponding (scaled) importance process.*

Then the expectations of $\mathbf{h}\big(\mathbf{x}(t)\big)$ *under measures* P*, and* Q*, such as* $Q(d\omega) = Z(t;\omega)P(d\omega)$ *can be expressed as*

$$\mathbb{E}_P\big[\mathbf{h}(\mathbf{x}(t))\big] = \mathbb{E}_P[Z(t;\omega)\mathbf{h}(\mathbf{s}^*(t))], \tag{9.76}$$

where the scaled version of the process $\mathbf{s}(t)$ *is defined as*

$$\mathbf{s}^*(t) = \mathbf{x}_0 + \mathbf{L}\mathbf{B}^{-1}(\mathbf{s}(t) - \mathbf{x}_0), \tag{9.77}$$

and the likelihood ratio

$$(\frac{dQ_{s^*}}{dP_x})(t;\omega) = Z(t;\omega) \tag{9.78}$$

is

$$\begin{aligned} Z(t;\omega) = \exp\Big(& \int_0^t \big[\mathbf{L}^{-1}\mathbf{f}(\mathbf{s}^*(t), t) - \mathbf{B}^{-1}\mathbf{g}(\mathbf{s}(t), t)\big]^T d\boldsymbol{\beta}(t,\omega) \\ & + \int_0^t \big[\mathbf{L}^{-1}\mathbf{f}(\mathbf{s}^*(t), t)\big]^T \big[\mathbf{B}^{-1}\mathbf{g}(\mathbf{s}(t), t)\big] dt \\ & - \frac{1}{2}\int_0^t \big(\|\mathbf{L}^{-1}\mathbf{f}(\mathbf{s}^*(t), t)\|^2 + \|\mathbf{B}^{-1}\mathbf{g}(\mathbf{s}(t), t)\|^2\big) dt\Big). \end{aligned} \tag{9.79}$$

Proof : see [Sarkka , 2006].

Theorem [9.3.2] actually states that given a set of samples from the process $\mathbf{s}(t)$ we can form a set of importance samples from $\mathbf{x}(t)$ by scaling $\mathbf{s}(t)$, computing the corresponding values $Z(t;\omega)$ and using them as the importance weights.

The values can be computed by using any numerical integration method as long as the method approximates the strong solution, which is needed to ensure that the weights $Z(t;\omega)$ are adapted to the same Brownian motion as $\mathbf{s}(t)$ (see Appendix [C.1].

Algorithm 9.3.5 (Importance Sampling of SDE) .

We get a set of importance samples sample $\{(\mathbf{x}^{(i)}, w^{(i)}) : i = 1, \cdots, N\}$ *using the importance process* $\mathbf{s}(t)$ *as follows:*

1. *randomly draw* N *Brownian motions* $\{\boldsymbol{\beta}^{(i)}(t), t_{k-1} \leq t \leq t_k, i = 1, \cdots, N\}$ *and, from* $t = 0$ *to* $t = T$*, simulate the corresponding (scaled) importance processes*

$$d\mathbf{s}^{(i)} = \mathbf{g}(\mathbf{s}^{(i)}, t)dt + \mathbf{B}d\boldsymbol{\beta}^{(i)} \tag{9.80}$$
$$\mathbf{s}^{(i)}(0) = \mathbf{x}_0, \tag{9.81}$$

2. *compute:*

$$\mathbf{s}^{*(i)}(t) = \mathbf{x}_0 + \mathbf{L}\mathbf{B}^{-1}(\mathbf{s}(t) - \mathbf{x}_0), \tag{9.82}$$

3. *set*

$$\mathbf{x}^{(i)} = \mathbf{s}^{*(i)}(T). \tag{9.83}$$

4. *for each* i *compute*

$$\begin{aligned} w^{(i)} = \exp\Big(& \int_0^T \Big[\mathbf{L}^{-1}\mathbf{f}(\mathbf{s}^{*(i)}(t), t) - \mathbf{B}^{-1}\mathbf{g}(\mathbf{s}^{(i)}(t), t)\Big]^T d\boldsymbol{\beta}^{(i)} \\ & + \int_0^T \big[L^{-1}\mathbf{f}(\mathbf{s}^{*(i)}(t), t)\big]^T \big[\mathbf{B}^{-1}\mathbf{g}(\mathbf{s}^{(i)}(t), t)\big] dt \\ & - \frac{1}{2}\int_0^T \big(\|\mathbf{L}^{-1}\mathbf{f}(\mathbf{s}^{*(i)}(t), t)\|^2 + \|\mathbf{B}^{-1}\mathbf{g}(\mathbf{s}^{(i)}(t), t)\|^2\big) dt\Big). \end{aligned} \tag{9.84}$$

5. $\{(\mathbf{x}^{(i)}, w^{(i)}) : i = 1, \cdots, N\}$ *is a set of importance samples such that for any function* $\mathbf{h}(.)$

$$\mathbb{E}\big[\mathbf{h}(\mathbf{x}(T))\big] \approx \sum_i w^{(i)}\mathbf{h}(\mathbf{x}^{(i)}), \tag{9.85}$$

where $\mathbf{x}(T)$ *is the solution to the stochastic differential equation*

$$d\mathbf{x} = \mathbf{f}(\mathbf{x}, t)dt + \mathbf{L}d\boldsymbol{\beta} \tag{9.86}$$
$$\mathbf{x}(0) = \mathbf{x}_0 \tag{9.87}$$

at time T.

Continuous-Discrete Sequential Importance Resampling Filter

Let consider the filtering model:

$$d\mathbf{x} = \mathbf{f}(\mathbf{x}, t)dt + \mathbf{L}d\boldsymbol{\beta} \tag{9.88}$$
$$\mathbf{y}_k \sim p(\mathbf{y}_k|\mathbf{x}(t_k)), \tag{9.89}$$

and, assume there exists importance process $\mathbf{s}(t)$ which is defined by:

$$d\mathbf{s} = \mathbf{g}(\mathbf{s}, t)dt + \mathbf{B}d\boldsymbol{\beta}, \tag{9.90}$$

which has the law that is a rough approximation to the filtering result of the model given by Eq. (9.88), and Eq. (9.89), at least at the measurement times, and the matrixes $\mathbf{L}$, and $\mathbf{B}$ are assumed to be invertible.

We generate a set of importance samples from the process $\mathbf{x}(t)$, conditioned to the measurements $\mathbf{Y}_{1:k}$ using $\mathbf{s}(t)$ as the importance process. This procedure reduces the degeneracy problem in the bootstrap filter.

Because the measures of both processes are absolutely continuous with respect to the measures of the driving Brownian motions it is possible to use the Algorithm [9.3.5] for generating the importance samples.

The continuous-discrete SIR filter for the model can be now constructed with a slight modification to the discrete-time SIR (Algorithm [H.6.1]) as follows:

Algorithm 9.3.6 (Continuous-Discrete Sequential Importance Resampling Filter)
.

Given the importance process $\mathbf{s}(t)$, a weighted set of samples $\{\mathbf{x}_{k-1}^{(i)}, w_{k-1}^{(i)}\}$ and the new measurement $\mathbf{y}_k$, a single step of continuous-discrete sequential importance resampling can be now performed as follows:

1. *draw N Brownian motions: $\{\boldsymbol{\beta}^{(i)}(t), t_{k-1} \leq t \leq t_{k-1} i = 1, \cdots, N\}$,*

2. *simulate the corresponding importance processes, from $t = t_{k-1}$ to $t = t_k$:*

$$d\mathbf{s}^{(i)} = \mathbf{g}(\mathbf{s}^{(i)}, t)dt + \mathbf{B}d\boldsymbol{\beta}^{(i)} \tag{9.91}$$

3. *put*

$$\mathbf{s}^{(i)}(t_{k-1}) = \mathbf{x}_{k-1}^{(i)} \tag{9.92}$$

4. *compute:*

$$\mathbf{s}^{*(i)}(t) = \mathbf{x}_{k-1}^{(i)} + \mathbf{L}\mathbf{B}^{-1}(\mathbf{s}^{(i)}(t) - \mathbf{x}_{k-1}^{(i)}), \tag{9.93}$$

5. *set:*

$$\mathbf{x}_k^{(i)} = \mathbf{s}^{*(i)}(t_k). \tag{9.94}$$

6. *for each i compute:*

$$\begin{aligned} w_k^{(i)} = w_{k-1}^{(i)} \exp\Big(& \int_{t_{k-1}}^{t_k} \big[\mathbf{L}^{-1}\mathbf{f}(\mathbf{s}^{*(i)}(t), t) - \mathbf{B}^{-1}\mathbf{g}(\mathbf{s}^{(i)}(t), t)\big]^T d\boldsymbol{\beta}^{(i)} \\ & + \int_{t_{k-1}}^{t_k} \big[\mathbf{L}^{-1}\mathbf{f}(\mathbf{s}^{*(i)}(t), t)\big]^T \big[\mathbf{B}^{-1}\mathbf{g}(\mathbf{s}^{(i)}(t), t)\big] dt \\ & - \frac{1}{2}\int_{t_{k-1}}^{t_k} \big(\|\mathbf{L}^{-1}\mathbf{f}(\mathbf{s}^{*(i)}(t), t)\|^2 + \|\mathbf{B}^{-1}\mathbf{g}(\mathbf{s}^{(i)}(t), t)\|^2\big) dt \Big)\, p(\mathbf{y}_k|\mathbf{x}_k^{(i)}), \end{aligned} \tag{9.95}$$

7. *re-normalize the weights to sum to unity,*
8. *compute the effective number of weights:*

$$\widehat{N}_{eff} = \frac{1}{\sum_{i=1}^{N} \left(w_k^{(i)}\right)^2}. \tag{9.96}$$

9. *if this effective number of weights is too low, perform resampling*

Part IV

Experiments Results

Part Four describes the Experimentations of asset Forecasting with a "Price Forecasting Model", and with a "Model of Trading".

10

Experiments

10.1 Introduction

This chapter describes the experiment results of financial time series forecasting, with the methods developed in Chapters [4], and [5], using high-frequency "tick-by-tick" time series without resampling, on an interval of three hours, for speculation.

Two models are tested:

- a "*Price Forecasting Model*", describes in chapter [4], forecasts the behavior of an asset on an interval of three hours, based on a *Functional Clustering* and a smoothing by *cubic-splines* in the training phase , and a *Functional Classification* for generalization (detailed in chapter [7].

- a "*Model of Trading*", describes in chapter [5], forecasts the first Stopping-time, the "high" and "low" of an asset on an interval of three hours, when this asset crosses for the first time a threshold selected by the trader. Two models are used:

  - a Model of Trading without stop-loss order, in a Discrete time representation (detailed in section [5.2]),

  - a Model of trading with stop-loss order, in a Continuous-Discrete time representation (detailed in section [5.3]),

  based on a *Dynamic State Space Representation*, using *Stochastic Bayes' Filters* for generalization, as detailed in chapter [8] for Discrete time representation, and in chapter [9] for Continuous-Discrete time representation.

For quality measurement of procedures, we realize tradings and we consider the cumulative returns in the generalization phase.

10.2 Price Forecasting Model

10.2.1 Data

The examples presented here deal with the IBM stock time series of "tick data" for the period starting on January 02, 1997 and ending on may 08, 1997 with more than 3000 transactions per day, on the New York Stock Exchange (NYSE). We use the periods : [January, 02 to Mars, 29] for Training, [Mars, 30, to April, 19] for Validation and [Mars, 20 to May, 08] for Testing.

10.2.2 Methodology

The Price Forecasting Model will use as inputs, inhomogeneous high-frequency time series of prices and homogeneous implied volatility series. Using such implied volatilities for price forecasting have been described in [Dablemont et al., 2003].

Volatility time series can be obtained from market data, (for example VIX, VXN or VDAX), or derived from option market prices. From eight call and put option contracts on the underlying asset, with strikes prices at-the-money, near-the-money, nearby and secondary nearby, in order to eliminate "smile" or "smirks", and with one to six months of maturity. With a distinction between American and European style option [Ruttiens, 2003].

Such implied volatilities series are difficult to estimate but they can be approximated by a tick-by-tick Zhou volatility estimator with long covariance ([Zumbach et al., 2002]) estimated from a k-tick returns. But this estimator can give negative values if the number of ticks k used for the returns is not large enough.

The pricing models use very high-frequency time series of tick by tick data, without resampling. 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, as described in Section 4.3.

10.2.3 Filtering

It is known by researchers that high-frequency series contain some bad quotes. Therefore the series have to be cleaned in order to eliminate these bad quotes (decimal errors, test ticks, repeated ticks, tick copying, ...) (see section [6.8]). If such bad quotes were used, the results of the simulation would be inevitably bad and unusable in case of aberrant outliers [Dacorogna et al., 1993]. When using tick data from a non-continuous market, we also eliminate the last half hours of the working hours of the market. It is better to consider the trades in these intervals of time as outliers.

10.2.4 Times of the trades

In a non-continuous market, we compress the weekends and other inactive periods (inactive hours, holidays) (see section [6.11]).

10.2.5 Training phase

The method described in the previous sections contains several meta-parameters. In order to fix these values in an appropriate way, we first select sets of possible values for each of them, and then train one model for each combination of the possible values. The latter are chosen as :

- the parameter Δt_{in}, (4hr, 8hr, 12hr, 16hr), interval of time for the regressors
- the order for cubic splines (4),
- the parameter q_{prices}, (3, 5, 7), number of knots for smoothing of prices,
- the parameter q_{vol}, (3, 5, 7), the number of knots for smoothing of volatilities,
- the parameter G_{in}, (3, 6, 9), number of clusters for IN map,
- the parameter G_{out}, (3, 6, 9), number of clusters for OUT map,
- the complexity of the RBFN (2, 4, 6) hidden nodes,
- the parameter Δt_{out} (1hr, 3hr), interval of time for prediction,
- the parameter Δt, (1hr), increment of time.

Best choose of Parameters

- For spline smoothing, we choose the number of internal knots, thus the dimension of the spline coefficient vector, according to the time interval for data, and the volatility of the asset.
- For RBF, we select the simplest model possible, in order to modelizing the process and not the noises,
- With a small number of parameters, the training model estimated from an asset can be used in forecasting, not only for this asset, but also for other assets of the same economical sector on the same market.

10.2.6 Validation phase

In the validation phase, we use the second part of tick data to test all models built in the training phase and we chose the best one according to a criterion of quality. For this pricing model we have chosen the quality of the prediction of the trend, the localization of "high" and "low" in time and their values.

10.2.7 Test phase

In the test phase, we use the last part of tick data for simulation, using the best pricing model estimated from the validation phase.

10.2.8 Performance Criteria

To estimate the performance of a forecasting method, researchers use common criteria as: Root Mean Square Error (RMSE), Mean absolute Error (MAE), Mean Square Error (MSE),....

For a speculator, the objective is to go short at "high" prices and to go long at "low" prices, in a very short-term (less than one day), in order to maximize the final net yield, after brokerage fees and taxes.

Thus to estimate the performances of our forecasting we choose the following criteria:

1. the similarity between predicted and real trend, to estimate if the market is going-down or going-up,
2. the localization of "high" and "low", to know when buying and selling a share,
3. the estimation of the difference between "high" and "low" values, to estimate if the trading will give a final net yield.

10.2.9 Forecasting

We realize the forecasting of future transactions for an interval of one hour at 10.00hr., 11.00hr., 12.00hr., 13.00hr., and 14.00hr., and the forecasting for an interval of three hours at 10.30hr., 11.30hr., and 12.30hr., from past transactions and implied volatilities series.

In the training phase, for clustering purposes, we shifted all curves in order to get first observations at $150.00 without distortion of curves. That means, forecasting prices are relative and not absolute, but delta prices are real.

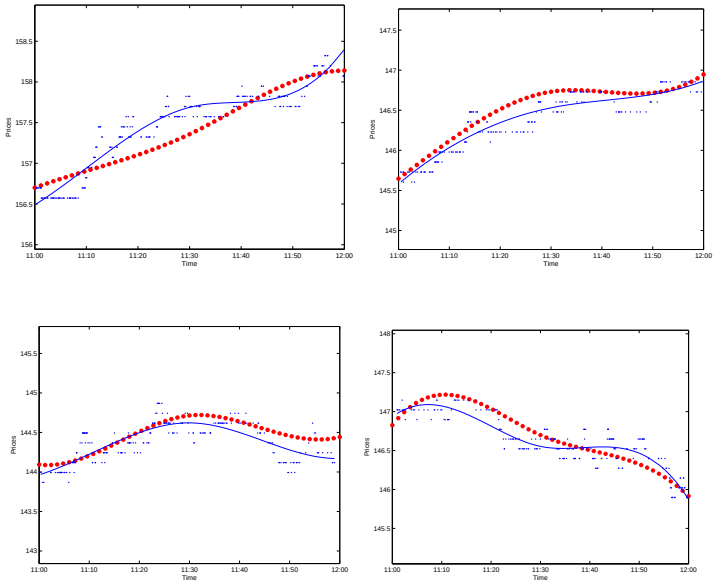

Figure 10.1: Price forecasting on an interval of one hour. Observations (Points); Smoothing splines (solid curve); Out-of-sample forecasting by the model (dotted curve)

Fig.(10.1) shows out-of-sample forecastings on an interval of one hour, superposed with observations and smoothing splines (not known by the model). We can observe good correlations between out-of-sample forecasting and observations. For a one hour forecasting, the trend prediction is very good, but this interval of prediction is not appropriated for speculation.

Fig.(10.2), (10.3), and (10.4) show out-of-sample forecastings on an interval of three hours, superposed with observations and smoothing splines (not known by the model). This interval of prediction is appropriated for speculation.

Fig.(10.2), shows "good" correlations between predicted and real trends, and accordingly, the trader can decide to go long and short to realize a profit.

Fig.(10.3), shows "good" correlations between localizations of "high" and "low", and their delta prices, between real and predicted curves. Thus the trader can trade at these optimal time-points to maximize his profit.

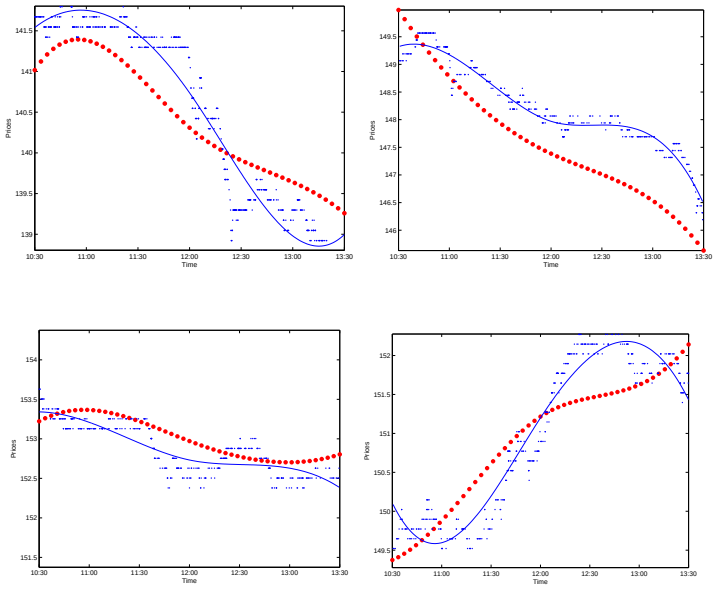

Figure 10.2: Price forecasting for an interval of three hours with good correlations between predicted and real trends. Observations (Points); Smoothing splines (solid curve); Out-of-sample forecasting by the model (dotted curve)

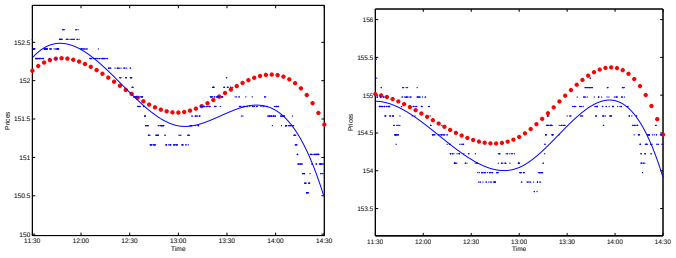

Fig.(10.4), shows correlations between localizations of "high" and "low", between real and predicted curves, but not the same delta prices. However, in many cases we observe than the predicted delta price is less than the real delta price. Thus the trader can also trade at these optimal time-points to maximize his profit.

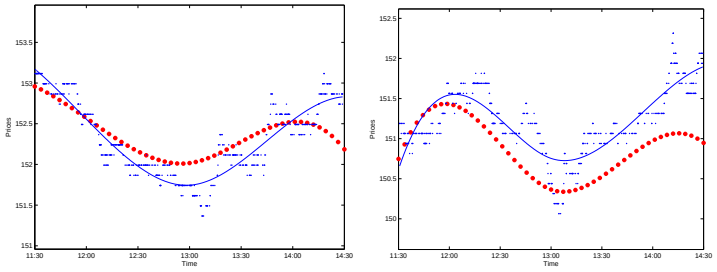

Figure 10.3: Price forecasting for an interval of three hours with good correlations between localizations of "high" and "low" and their difference of prices. Observations (Points); Smoothing splines (solid curve); Out-of-sample forecasting by the model (dotted curve)

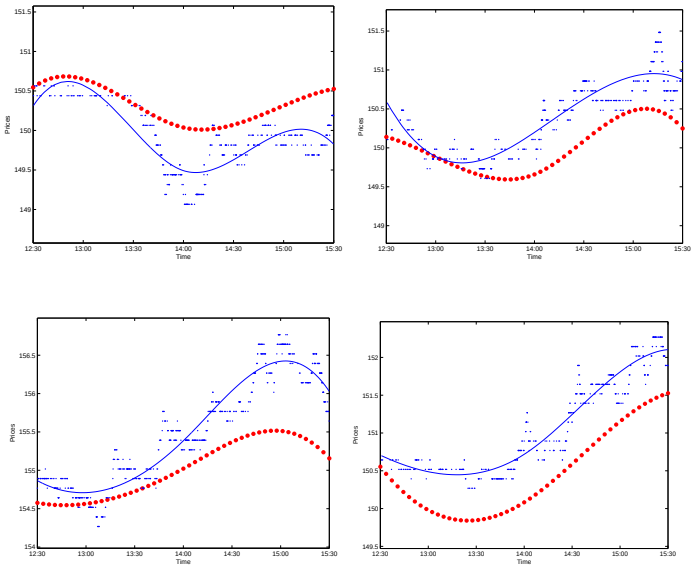

Figure 10.4: Price forecasting for an interval of three hours with good correlations between localizations of "high" and "low" but a difference between real and predicted prices. Observations (Points); Smoothing splines (solid curve); Out-of-sample forecasting by the model (dotted curve)

By inspection of 60 curves for a one hour interval of prediction, and 45 curves for a three hours interval of prediction.

- For a one hour forecasting, we estimate between 75% and 80% of global "good" predictions.
  - "Good" prediction of trends. The real and predicted trends are similar, and accordingly the trader can go long or go short with profit.
  - "Good" prediction of "high" and "low". The localizations of "high" and "low" in observation series and forecasting are at the "very similar time-points" and with the "very similar values". If the trader trades at these predicted times, he maximizes his profit.
- For a three hours forecasting, we estimate between 60% and 65% of global "good" predictions.
  - The prediction of trends is easy, and accordingly, the trader can go long and go short with profit. This is the main criteria, and we estimate 34 "good" predictions of the trend on 45 curves. Note than if the market is going-up or going-down we have no "high" and "low", but only an increasing or decreasing curve.
  - The localization of "high" and "low" seem possible, and the trader can trade at these optimal time-points. We estimate 28 "good" forecastings of these time-points on 45 curves.
  - The estimation of the difference between "high" and "low" values is a bit difficult, and the estimation of net yield is biased. We estimate only 18 "good" predictions of delta prices on 45 curves, but in some cases we could trade at these time-points and realize a final yield better than the prediction.
  - We observed 9 curves for which trends, localizations of "high" and "low", and delta prices were bad.
  - We observed 2 outliers for which the classification was impossible.

10.2.10 Tradings

To quantify the power of this algorithm, every day at 10.30hr, 11.30hr, and 12.30hr, we forecast curves of future prices for an interval of three hours. We decide to trade or to do nothing, depending of the trend forecasting. At time-points "high" and "low" predicted by the model we sell, eventually short, or buy one share, respectively.

Fig.(10.5), shows the predicted cumulative returns for tradings in these intervals of three hours. The figures include the losses due to transaction costs (bid-ask spreads), but no brokerage fees or taxes. We assume an investor with

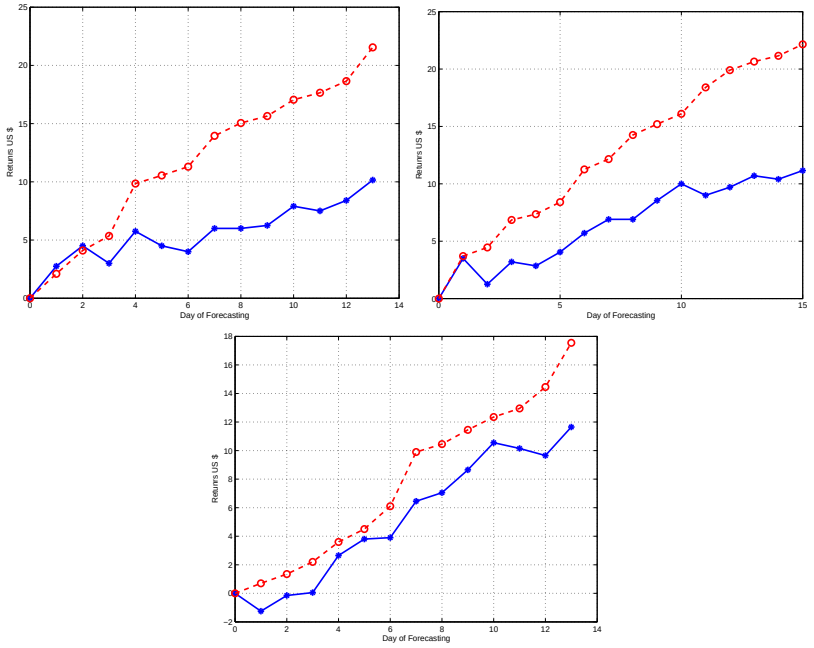

Figure 10.5: Cumulative Returns for intervals of three hours at 10:30, 11:30, and 12:30. Realized Cumulative Returns (solid curve), Predicted Cumulative Returns (dotted curve)

no credit limit but no capital.

For these intervals of tradings, we estimate a realization of 47%, 50%, and 66% of the predicted cumulative returns. We also note than a majority of realized local returns are positive even if the values are less than the predicted ones.

Predictions depend on the day, due to holidays and week-end in inputs, and also the localization of the interval of prediction in the day. Forecasting Wednesdays is easier than forecasting Mondays. Forecasting at 11:30hr. is easier than forecasting at 10:30hr, because we have more data from the opening of the market until the interval of prediction.

10.2.11 Pros and Cons

- **Pros**
  - The training phase is realized in batch,

  - The forecasting phase is very fast, only 5 sec. of CPU time on a PC,
  - We can use the model estimated from an asset to realize the forecasting of an other asset of the same economical sector on the same market.
  - We use the implied volatility of the index as the volatility of the asset.
- **Cons**
  - Training and Tuning:
    - are difficult,
    - the computational load is hight- one week of CPU time per model on a PC.
  - We have many models to manage:
    - we need one model per day,
    - we need one model per horizon of prediction.
  - We need high frequency time series of Prices and Implied Volatilities.

Conclusion

We have presented a functional method for the clustering, modeling and forecasting of time series by functional analysis and neural networks. This method can be applied to any types of time series but is particularly effective when the observations are sparse, irregularly spaced, occur at different time points for each curve, or when only fragments of the curves are observed. Standard methods completely fail in these circumstances. By the functional clustering, we can also forecast multiple dynamic processes.

At first sight, these "Price Forecasting Models" could be a good opportunity for speculators, to trade actions on a very short horizon.

The main difficulty with this procedure reminds the CPU time for the training and the tuning of a large set of models. Once training, validation and test phases have been done in an off-line and batch procedure, we can realize the forecasting in real time, with a very low computational load.

10.3 Model of Trading

10.3.1 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 and validation of the models, and the period : January, 04, 1999 to May, 26, 1999 for testing.

We use "bid-ask" spreads as exogenous inputs in the Pricing model, and transaction prices as observations in Pricing and Trading models

10.3.2 Methodology

To realize this experimentation, we build two models.

The first model (**Price forecasting 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 short, or to do nothing.
This model is based on a *Functional Clustering Analysis* (see Section [7.3]) and uses a Local Nonlinear *Radial Basis Function Network* (RBFN) [Benoudjit and Verleysen, 2003]

The second model (**Trading model**) estimates the first stopping time in order to close the position, depending of a threshold defined by the trader.

This model uses a *Dynamical State-Space Models* (DSSM),

$$\mathbf{x}_{k+1} = \mathbf{f}_k(\mathbf{x}_k, \mathbf{v}_k; \mathbf{w}) \tag{10.1}$$

$$\mathbf{y}_k = \mathbf{h}_k(\mathbf{x}_k, \mathbf{n}_k; \mathbf{w}) \ , \tag{10.2}$$

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

The 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 modeling purposes [Dablemont et al., 2007].

We realize the training phase, to estimate the parameters of the models, by *joint estimation*, but in order to curb 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.

10.3.3 Filtering

In the experience of many researchers, high-frequency series contain some bad quotes, thus they have to be cleaned in order to eliminate these bad quotes (decimal errors, test ticks, repeated ticks, tick copying, ...) (see section [6.8]). If such bad quotes were used, the results of the simulation would be inevitably bad and unusable in case of aberrant outliers [Dacorogna et al., 1993]. In case of using tick data from a non-continuous market, we also eliminate the last half hour of the working hours of the market. It is better to consider the trades in these intervals of time as outliers.

10.3.4 Times of the trades

In a non-continuous market, we compress the weekends and other inactive periods (inactive hours, holidays) (see section [6.11]).

10.3.5 Training phase

The method described in the previous sections contains several meta-parameters. In order to fix these values in an appropriate way, we first select sets of possible values for each of them, and then train one model for each combination of the possible values. The latter are chosen as :

- the interval of time for the regressors of the Price Forecasting Model for the Trend (6hr, 9hr)
- the interval of time for the Trading Model (3hr),
- the number of knots for the smoothing by splines (3,5,7),
- the dimension of the state ($\mathbb{R}^1$, $\mathbb{R}^2$, $\mathbb{R}^3$),
- the complexity of the RBF (2, 3, 4 hidden nodes),
- the complexity of the state equation $\mathbf{f}$, and the measurement equation $\mathbf{h}$

10.3.6 Best choose of Parameters

- For the Price Forecasting Model, we choose the number of internal knots of spline smoothing according to the time interval selected for input data.
- For the Trading Model, we choose one internal knot for spline smoothing, due to a time interval of three hours for data.
- For RBF, we select the simplest model possible, in order to modelizing the process and not the noises,

10.3.7 Validation phase

In the validation phase, we use the second part of tick data to test all models built in the training phase and we chose the best one according to a criterion of quality. For the pricing model, we have chosen the quality of the prediction of the trend, and for the trading model, we have chosen the precision of the prediction of the first stopping time.

10.3.8 Test phase

In the test phase, we use the last part of tick data for simulation, using the best pricing model and the best trading model estimated from the validation phase.

Simulation without Stop-Loss order

Every day, at 11.00 AM, we realize *one* trading. We decide to buy or to sell short *one* share or to do 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 this trading process is presented as a function of time in Figure [10.6]. The figure of the cumulative returns includes the losses due to transaction costs (the bid-ask spreads) but no brokerage fees or taxes. We assume an investor with credit limit but no capital.

On the interval of simulation, [Jan, 01, 1999 to May, 26, 1999], we have a positive cumulative yield, but sometimes we observe a negative local yield. To curb these negative yields we add a stop-loss process.

Simulation with Stop-Loss order

We realize one trading every day at 11.00 AM, but in this simulation, every ten minutes, we rerun the pricing model to estimate the new trend. If this new trend is similar to the former then we continue, but if we have a reversal in trend we immediately stop and close the position. We realize a reversal trading (to go short in place of to go long, or the opposite) and rerun the trading model to predict the new first stopping time in hope to close the position with a final positive yield.

The resulting behavior of this trading process with stop-loss order is presented as a function of time in Figure [10.7]. We also represent the cumulative returns.

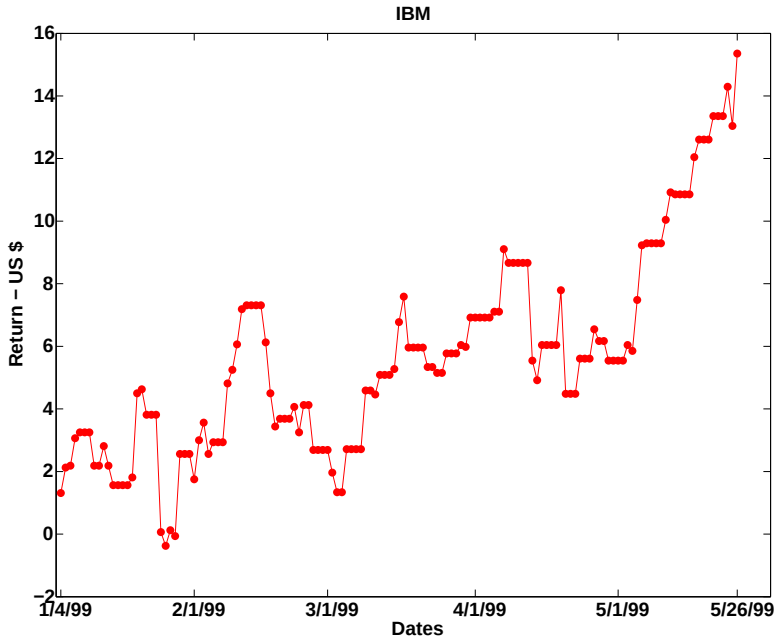

Figure 10.6: Cumulative Returns of trading models as function of time, without stop-loss order. Every day, at 11.00 AM, we realize *one* trading. We decide to buy or to sell short *one* share or to do nothing, depending of the trend forecasting, then we close the position according to the predicted stopping time.

In this simulation, with a stop-loss process, we have reduced the number of negative local yields and their amplitude. That gives a better final net yield, but with an very significant increasing of the computational load.

10.3.9 Pros and Cons

- **Pros**
  - We only use "Prices", and "Bid" and "Ask" high frequency series of data.
  - The Trading Model is auto-adaptive to slow modifications of dynamics.
- **Cons**

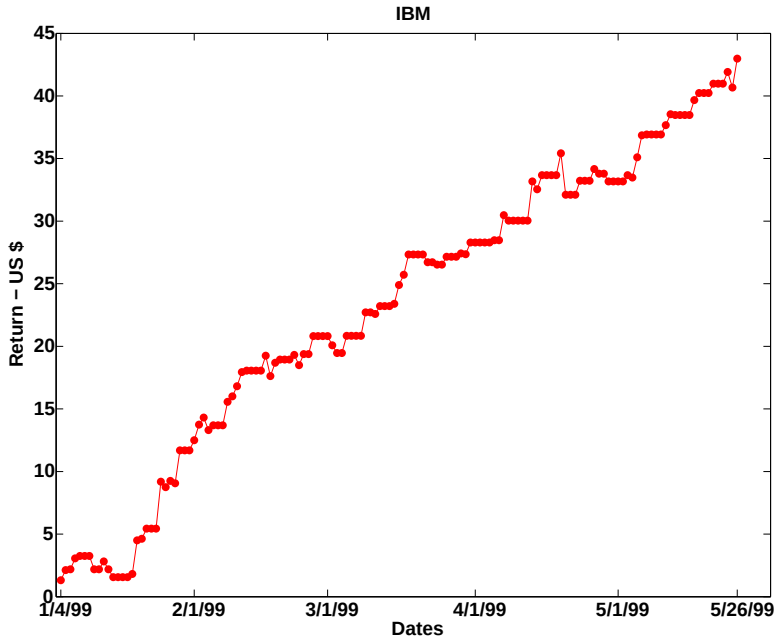

Figure 10.7: Cumulative Returns of trading models as function of time, with stop-loss order. Every ten minutes, we rerun the pricing model to estimate the new trend. If this new trend is similar to the former, we continue, but if we have a reversal in the trend, we immediately stop and close the position. We realize an opposite trading and rerun the trading model to predict the new first stopping time.

- The Training and Validation phases are realized as a real time procedure.
- The complexity of parameter estimation is very hight.
- In the Forecasting phase, periodically we have to reinitialize the parameter estimation in a batch procedure, to curb the degeneration of the algorithms.
- To limit the "curses" of dimensionality, we must split the procedures into two steps. First an off-line parameter estimation; afterwards, an on-line state forecasting.
- Periodically, we must reestimate the parameters in a batch procedure.

Conclusion

We have presented a functional method for the Trading of very high frequency time series by Functional Analysis, Neural Networks, and Bayesian Estimation, with Particle and Kalman Filters.

A Pricing model gives, at time t, a signal of buying or selling short an asset, or to do nothing. The trader goes long or short. A Trading model forecasts when this asset crosses for the first time a threshold, and the trader closes his position. A stop-loss process can detect bad forecastings in order to curb the negative local yields.

These processes 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 on five months to test the algorithms, and at first sight, the stable profitability of trading models and the quality of the pricing models could be a good opportunity for trading assets on a very short horizon for speculators.

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

These algorithms are written in Matlab, and they need a very computational load for forecastings, because of the recursive up-dating of the Particles. For real-time applications we could put them in a form suitable for parallelization in C++.

11

Conclusions

The world went through weeks of financial turbulence in stock markets and investors were overcome by fears fueled by more bad news, while countries continued their attempts to calm the markets with more injection of funds.

By these very disturbed times, even if traders hope extreme risk aversion has passed, an investor would like predict the future of the market in order to protect his portfolio and a speculator would like to optimize his tradings. But can we forecast the market?

The "Efficient Market Hypothesis" states that stock prices reflect all information.

If markets are efficient then new information is reflected quickly into market prices. Then past prices contain no information about future changes and price changes are random, that implied zero correlation between price change at time "t" and price change at time "t+1".

- If price cycles were predictable competition between investors would eliminate them.
  - Arbitrage/Speculation will force prices to their efficient values,
  - Simple trading rule: Buy undervalued assets - Sell overvalued assets.
- Prices will only change on the basis of new information which by definition is random
  - hence price changes are random.

Conversely, if markets are inefficient, information is reflected only slowly into market prices, if at all.

The empirical evidence surveyed in Fama generally supports this idea for long time interval but researches questioned this assertion for very short time interval, and they find anomalies such as: Small firms, January effect, Day of the Week effect, Holiday effects, Volatility tests/Predictability of long run returns, Autocorrelation properties, Contarrian/Values strategies, Momentum strategies, New Issue market.

Moreover, traders use Fundamental analysis based on statistical tools and Technical analysis which employs tools of geometry and pattern recognition as effective means for extracting useful information from market prices. They consider than past prices contain information for predicting future returns.

11.1 Purpose of the Thesis

The analysis of financial time series is of primary importance in the economic world. This thesis deals with a data-driven empirical analysis of financial time series. The goal is to obtain insights into the dynamics of series and out-of-sample forecasting.

Forecasting future returns on assets is of obvious interest in empirical finance. If one was able to forecast tomorrow's returns on an asset with some degree of precision, one could use this information in an investment today. Unfortunately, we are seldom able to generate a very accurate prediction for asset returns.

11.2 Models of Forecasting

This thesis describes the design of numerical models and algorithms, for the forecasting of financial time series, for speculation on a short time interval.

To this aim, we will use two models:

- a *Price Forecasting Model* predicts asset prices on an interval of three hours,
- a *Model of Trading* predicts "high" and "low" of an asset, on an interval of three hours,

using high-frequency "tick-by-tick" financial time series, without resampling, and without other economic information.

11.2.1 Price Forecasting Model

A "*Price Forecasting Model*" forecasts the behavior of an asset for an interval of three hours from high-frequency "tick-by-tick" financial time series without resampling.

Financial time series display typical nonlinear characteristics, it exists clusters within which returns and volatility display specific dynamic behavior. For this reason, we consider here nonlinear forecasting models, based on local analysis into clusters. Although financial theory does not provide many motivations for nonlinear models, analyzing data by nonlinear tools seems to be appropriate, and they are at least as much informative as an analysis by more restrictive linear methods.

This model is based on *Functional Clustering* and smoothing by *cubic-splines* in the training phase to build *local Neural models*, and *Functional Classification* for generalization.

For the data, we only use *Prices* and *Implied Volatility* of this asset but no other economical information.

The aim of this model is to, at least empirically, verify that a multiple switching process leads to better short-term forecasting based on an empirical functional analysis of the past of series.

An originality of this method is that it does not make the assumption that a single model is able to capture the dynamics of the whole series. On the contrary, it splits the past of the series into clusters, and generates a specific local neural model for each of them. The local models are then combined in a probabilistic way, according to the distribution of the series in the past.

This forecasting method can be applied to any time series forecasting problem, but is particularly suited for data showing nonlinear dependencies, cluster effects and observed at irregularly and randomly spaced times like high-frequency financial time series do.

Results

We tested this model on securities from NYSE and EURONEXT Paris, and we shown the results for IBM asset. Forecasting one to three hours is fine, but the prediction for a longer interval seems impossible, the quality of forecasting estimated from cumulative returns gives very poor results. This could be the limit of predictability for this market, due to the absence of structure of correlation for longer interval, as described by the "Efficient Market Hypothesis".

Pros

- The forecasting phase is very fast, only a few secondes of CPU time.
- We can use the model estimated from an asset to realize the forecasting of an other asset of the same economical sector on the same market.

Cons

- Training and Tuning are difficult, and the computational load is hight,
- We need high frequency time series of Prices and Implied Volatilities, and Implied volatility is difficult to estimate. We can only use Prices but the results are really not so good.
- This model is not auto-adaptive to changes of dynamics such as split, take-over bid,...

11.2.2 Model of Trading

The *Model of Trading* forecasts the *First Stopping time*, when an asset crosses for the first time a threshold defined by the trader, using high frequency time series of "Bid-Ask" spreads and "Prices" without other economical information.

This model combines a *Price Forecasting Model* for the prediction of market trend, and a *Trading Recommendation* for prediction of the first stopping time.

For prediction of the market trend, we use a Functional Clustering with smoothing by cubic splines and local Neural Networks forecasting models.

For the prediction of the first stopping time, we use an auto-adaptive *Dynamic State Space Model*, with *Particle Filters* and *Kalman/Bucy Filters* for parameter estimation.

Two models are tested:

- We use a *Discrete Dynamic State Space Model* with *Stochastic Difference Equations* and with *Particle Filters* and *Kalman Filters* for prediction of the first stopping time, without stop-loss order.
- We use a *Continuous/Discrete Dynamic State Space Model* with *Stochastic Differential Equations* and with *Stochastic Particle Filters* and *Kalman-Bucy Filters* for prediction of the first stopping time, with stop-loss order.

Results

We have tested these models on IBM asset from NYSE. The quality of forecasting was estimated by realizing tradings and measuring the cumulative returns. The results are less than with the Price forecasting model, but using a stop-loss order improve the results. The difficulty comes with the need of choosing a threshold. If we are optimistic, we choose a hight threshold and in many cases we could not find a first stopping time, and we lose some opportunities of gains. If we are pessimistic, we choose a low threshold and in many cases we lose opportunities of higher gains.

Pros

- We only use "Prices", and "Bid" and "Ask" high frequency series of data.
- The Trading Model is auto-adaptive to slow modifications of dynamics.

Cons

- The complexity of parameter estimation is very hight.
- In the Forecasting phase, periodically we have to reinitialize the parameter estimation in a batch procedure, to curb the degeneration of the algorithms.

11.3 Open issues

The main problem with particle filters is the CPU time due to the number of particles used to approximate the posterior, even if we use an adaptive Particle filter. In a real time application we should use Parallel Computing to curb this CPU time. But the difficulty comes with the risk of particle depletion at each local node and not to have enough diversity in the particle pool to properly track the state. Moreover, combining adaptive particle filter and parallel computing seems difficult.

To forecast "High" and "Low", we should test a full Continuous Time State Space Model with Stochastic Differential Equations for states and measurements, and multiplicative noises in both equations. In this case, the dispersion matrix must depend on time and state and Particle and Kalman-Bucy filters must be adapted to allow the more general dispersion matrices. This could be possible by using an Ito formula instead of a Stratonovich one for the derivation of Kalman-Bucy filters. This extension should not be straightforward because the Girsanov theorem can not be applied in this case. This theorem is useful to find weak solutions of stochastic differential equations that we need

for the Importance process of particle filters

It could be possible to generalize the algorithms presented in this thesis to more complex securities such as futures on Bund, euro stoxx 50, swap USD or EUR, Interest rates,... but in this case we have to use diffusion models and their extensions such as jump-diffusions and Markov models driven by Lévy processes. Using more general martingales we can model sudden changes in signals. The Brownian Motion should be a general Lévy process such as compound Poisson processes, but the main difficulty is that these processes do not have the excessively useful Markov property which is the foundation of filtering for Dynamic state space models. Moreover, according to the financial theory, noises of state and measurements equations should be correlated.

Part V

Appendix

Part Five clarifies the tools used in developments of applications, gives the demonstration of theorems, and algorithms.

A

Mathematical Definitions.

A.1 Preliminaries

Definition A.1.1 (Probability space) .

A probability space is defined by the elements $\{\Omega, \mathcal{F}, P\}$ where $\mathcal{F}$ is a σ-algebra of Ω and P is a complete, σ-additive probability measure on all $\mathcal{F}$. In other words, P is a set function whose arguments are random events (element of $\mathcal{F}$) such that axioms of probability hold.

Definition A.1.2 (σ-algebra) .

Let $\mathcal{S}$ be a set and $\mathcal{F}$ be a family of subsets of $\mathcal{S}$. $\mathcal{F}$ is a σ-algebra if

1. $\emptyset \in \mathcal{F}$
2. $A \in \mathcal{F}$ *implies* $A^c \in \mathcal{F}$
3. $A_1, A_2, \cdots \in \mathcal{F}$ *implies* $\bigcup_{i=1}^{\inf} A_i \in \mathcal{F}$.

Definition A.1.3 (Stochastic process) .

An indexed collection of random variables

$$\mathcal{X}(\omega) = \{\mathbf{x}(t;\omega),\ 0 \leq t < \infty\}, \tag{A.1}$$

is called a stochastic process. A stochastic process is really a function of two variables, the time parameter t and the probability parameter ω.

- *each $\omega \mapsto x(t;\omega)$ is a measurable function defined on a probability space $(\Omega, \mathcal{F}, P)$,*
- *for each ω, the function $t \mapsto \mathbf{x}(.;\omega)$ is called the sample path (or realization or trajectory) of the process,*
- *for each t, the function $t \mapsto \mathbf{x}(t;.)$ is a random variable*

Definition A.1.4 (Filtration) .

The increasing family of σ-algebras $\mathcal{X}_t \subset \mathcal{F}$ on Ω such that

$$\mathcal{X}_t = \sigma(\mathbf{x}(s),\ 0 \leq s < t) \ \Rightarrow \mathcal{X}_s \subset \mathcal{X}_t, \tag{A.2}$$

is called a filtration.

The natural filtration of a stochastic process is the smallest filtration such that the process is adapted to it.

The natural filtration $\mathcal{X}_t$ of stochastic process $\mathbf{x}(t;\omega)$, that is,

$$\mathcal{X}_t = \sigma(\mathbf{x}(s), 0 \leq s < t), \tag{A.3}$$

can be thought of as the history of the stochastic process up to the time t. The filtration contains all the information that can be known about the process at the time t.

Definition A.1.5 (Adapted process) .

The stochastic process $\mathbf{x}(t;\omega)$ is said to be adapted to the filtration $\mathcal{X}_t$ if for each $t \geq 0$ the function $\omega \mapsto \mathbf{x}(t;\omega)$ is $\mathcal{X}_t$-measurable.

Definition A.1.6 (Markov process) .

A stochastic process $\mathbf{x}(t)$ is a Markov process if its future is independent of its past given the present:

$$p(\mathbf{x}(s)|\mathcal{X}_t) = p(\mathbf{x}(s)|\mathbf{x}(t)), \textit{for all} s \geq t. \tag{A.4}$$

Definition A.1.7 (Martingale) .

An $\mathcal{X}_t$-adapted stochastic process $\mathbf{x}(t)$ with bounded expectation $\mathbb{E}[\mathbf{x}(t)] < \infty$ is called a martingale with respect to the filtration $\mathcal{X}_t$ if

$$\mathbb{E}[\mathbf{x}(s)|\mathcal{X}_t] = \mathbf{x}(t), \textit{for all } s \geq t. \tag{A.5}$$

Definition A.1.8 (Standard Brownian motion) .

A process $\beta(t)$ is called a standard Brownian motion, or Wiener process if it has the following properties:

1. $\beta(0) = 0$,
2. $\beta(t_1), \beta(t_2) - \beta(t_1), \cdots, \beta(t_3) - \beta(t_2)$ *are independent for all* $t_1 < t_2 \cdots < t_{k-1} < t_k < \infty$,

3. $\beta(t) - \beta(s) \sim \mathcal{N}(0, t-s)$ *for every* $0 < s < t < \infty$,
4. *The sample path* $t \mapsto \beta(t;\omega)$ *is continuous for all* $\omega \in \Omega$.

Definition A.1.9 (Brownian motion) .

A scalar Brownian motion with diffusion coefficient $q(t)$ *can be defined as the process:*

$$\beta(t) = \sqrt{q(t)}\beta_s(t) \tag{A.6}$$

where $\beta_s(t)$ *is a standard Brownian motion.*

An n-dimensional vector process $\boldsymbol{\beta}(t) = (\beta_1(t) \cdots \beta_n(t))^T$ *where* $\beta_i(t)$ *are independent Brownian motions with diffusion coefficients* $q_i(t)$ *is called n-dimensional Brownian motion with diffusion matrix* $\mathbf{Q}_c(t) = diag(q_1(t), ..., q_n(t))$.

Definition A.1.10 (Stochastic integral) .

An Itô integral is the stochastic integral of a function (or stochastic process) $f(t,\omega)$ *with respect to Brownian motion* $\beta(t)$

$$\mathcal{I}[f] = \int_S^T f(t,\omega)d\beta_t. \tag{A.7}$$

Definition A.1.11 (Simple Process) .

A stochastic process $\phi_n(t,\omega) : [S,T] \times \Omega \to \mathbb{R}$ *is called simple if there exists partition* $S = t_0 < t_1 < \cdots < t_n < t_{n+1} = T$ *such that* $\phi_n(s,\omega) = \theta_j(\omega), t_j < s \leq t_{j+1}$ *where* $\phi_n(s,\omega)$ *is a random variable.*

For technical reasons we shall also require that each $\theta_j(\omega)$ *is measurable with respect to a filtration* $\mathcal{X}_t$ *such the Brownian motion* $\beta(t)$ *is martingale with respect to the filtration.*

Thus, simple process is a piecewise constant stochastic process. For simple processes the Ito integral can be defined as follows:

$$\mathcal{I}[\phi_n] = \int_S^T \phi_n(t,\omega)d\beta_t \tag{A.8}$$

$$= \sum_{j=0}^{n} \theta_j(\omega)\big(\beta(t_{j+1} - \beta(t_j)\big). \tag{A.9}$$

Ito integral of a more general stochastic process can be now defined as limit of integrals of simple processes:

Definition A.1.12 (Itô Integral) .

Let $(\Omega, \mathcal{A}, P)$ be a probability space,

- *$\beta(t)$ a Brownian motion with natural filtration $\mathcal{F}_t \subset \mathcal{A}$,*
- *$f(t,\omega) : [0,\infty) \times \Omega \to \mathbb{R}$ a stochastic process with the following properties:*

1. *$(t,\omega) \mapsto f(t,\omega)$ is $\mathcal{B}[0,\infty) \times \mathcal{A}-$ measurable.*
2. *There exists filtration $\mathcal{X}_t$ such that $\beta(t))$ is martingale with respect to $\mathcal{X}_t$ and $f(t,\omega)$ is $\mathcal{X}_t$-adapted.*
3. *$\mathbb{E}[\int_S^T f(t,\omega)^2 dt] < \infty$.*

Then the Ito integral of $f(t,\omega)$ with respect to the Brownian motion $\beta(t)$ can be defined as

$$\int_S^T f(t,\omega) d\beta(t;\omega) = \lim_{n\to\infty} \int_S^T \phi_n(t,\omega) d\beta(t;\omega), \tag{A.10}$$

where $\{\phi_n\}$ is a sequence of simple processes such that

$$\lim_{n\to\infty} \mathbb{E}\Big[\int_S^T \big(f(t,\omega) - \phi_n(t,\omega)\big) dt \Big] = 0. \tag{A.11}$$

Note that the Ito integral is always a martingale.

A.2 Markov Process

Definition A.2.1 (Stochastic Process) .

We can mathematically describes a Stochastic Process as a system which evolves probabilistically in time, or a system in which a certain time-dependent random variable $x(t)$ exits. We can measure values $x_1, x_2, \cdots, x_n$ of $x(t)$ at times $t_1, t_2, \cdots, t_n$ and assume that a set of joint probability densities exits.

$$p\big(x_1, t_1; x_2, t_2; x_3, t_3; \cdots\big) \tag{A.12}$$

which describes the system completely. In terms of these joint probability density functions, we can also define conditional probability densities:

$$\begin{aligned} &p\big(x_1, t_1; x_2, t_2; x_3, t_3; \cdots | y_1, \tau_1; y_2, \tau_2; y_3, \tau_3; \cdots\big) \\ &\quad = \frac{p\big(x_1, t_1; x_2, t_2; x_3, t_3; \cdots; y_1, \tau_1; y_2, \tau_2; y_3, \tau_3; \cdots\big)}{p\big(y_1, \tau_1; y_2, \tau_2; y_3, \tau_3; \cdots\big)} \end{aligned} \tag{A.13}$$

Definition A.2.2 (Markov Process) .

The Markov Assumption is formulated in terms of the conditional probabilities. We require that if the times satisfy the ordering $t_1 \geq t_2 \geq t_3 \geq \cdots \geq \tau_1 \geq \tau_2 \cdots$, the conditional probability is determined entirely by the knowledge of the most recent condition

$$\begin{aligned} &p\big(x_1, t_1; x_2, t_2; x_3, t_3; \cdots | y_1, \tau_1; y_2, \tau_2; y_3, \tau_3; \cdots\big) \\ &\quad = p\big(x_1, t_1; x_2, t_2; x_3, t_3; \cdots | y_1, \tau_1\big) \end{aligned} \tag{A.14}$$

provided

$$t_1 \geq t_2 \geq t_3 \geq \cdots \geq \tau_1 \geq \tau_2 \cdots \tag{A.15}$$

By definition of the conditional probability density and using the Markov assumption, we find

$$\begin{aligned} &p\big(x_1, t_1; x_2, t_2; x_3, t_3; \cdots ; x_n, t_n\big) \\ &\quad = p\big(x_1, t_1 | x_2, t_2\big) p\big(x_2, t_2 | x_3, t_3\big) \cdots p\big(x_{n-1}, t_{n-1} | x_n, t_n\big) p\big(x_n, t_n\big) \end{aligned} \tag{A.16}$$

Definition A.2.3 (Chapman-Kolmogorov Equation) .

For unconditional probabilities, we have

$$p\big(x_1, t_1\big) = \int p\big(x_1, t_1; x_2, t_2\big) dx_2 \tag{A.17}$$

$$= \int p\big(x_1, t_1 | x_2, t_2\big) p\big(x_2, t_2\big) dx_2 \tag{A.18}$$

For conditional probabilities, we get

$$p\big(x_1, t_1 | x_3, t_3\big) = \int p\big(x_1, t_1; x_2, t_2 | x_3, t_3\big) dx_2 \tag{A.19}$$

$$= \int p\big(x_1, t_1 | x_2, t_2; x_3, t_3\big) p\big(x_2, t_2 | x_3, t_3\big) dx_2 \tag{A.20}$$

If we introduce the Markov assumption, we can drop the t_3 dependance in the doubly conditioned probability and we find

$$p\big(x_1, t_1 | x_3, t_3\big) = \int p\big(x_1, t_1 | x_2, t_2\big) p\big(x_2, t_2 | x_3, t_3\big) dx_2 \tag{A.21}$$

which is the Chapman-Kolmogorov Equation.

A.2.1 Continuity in Stochastic processes

Whether or not the random variable $\mathbf{x}_t$ has a continuous range of possible values is a completely different question from whether the sample path of $\mathbf{x}(t)$ is a continuous function of t.

A major question now arise. Do Markov processes with continuous sample paths actually exist in reality? It is almost certainly the case in a classical picture, all variables with a continuous range have continuous sample paths. But in many cases, if we observe the process on a fine time scale, the process will probably not Markovian. the immediate history of the whole system will almost certainly be required to predict even the probabilistic future. This is certainly born out in all attempts to derive Markovian probabilistic equations from mechanics. Equations which are derived are rarely truly Markovian, rather there is a certain characteristic memory time during which the previous history is important.

This means that there is really no such thing as a Markov process; rather, there may be systems whose memory time is so small that, on the time scale on which we carry out observations, it is fair to regard them as being well approximated by a Markov process. But in this case, the question of whether the sample paths are actually continuous is not relevant. The sample paths of the approximating Markov process certainly need not be continuous.

However, Markov processes with continuous sample paths do exist mathematically and are useful in describing reality.

Definition A.2.4 (Mathematical Definition of a Continuous Markov Process) .

For a Markov process, the sample paths are continuous functions of t, if for any $\epsilon > 0$ we have

$$\lim_{\Delta t \to 0} \frac{1}{\Delta t} \int_{|\mathbf{x}-\mathbf{z}|>\epsilon} p\big(\mathbf{x}, t+\Delta t|\mathbf{z}, t\big) d\mathbf{x} = 0 \tag{A.22}$$

uniformly in $\mathbf{z}$, t, and Δt. That means that the probability for the final position $\mathbf{x}$ to be finitely different from $\mathbf{z}$ goes to zero faster that Δt, as Δt goes to zero.

Definition A.2.5 (Differential Stochastic Equation) .

Differential Stochastic Equation is an equation of the form

$$d\mathbf{x}(t) = \mathbf{f}(\mathbf{x}, t)dt + \mathbf{L}(\mathbf{x}, t)d\boldsymbol{\beta}(t), \tag{A.23}$$

where

- $\mathbf{f} : \mathbb{R}^n \times [0, \omega) \mapsto \mathbb{R}^n$ *is the drift function ,*
- $\mathbf{L} : \mathbb{R}^n \times [0, \infty) \mapsto R^{n*d}$ *is the dispersion matrix,*
- $\boldsymbol{\beta}(t)$ *is a* $d-$*dimensional Brownian motion with diffusion matrix* $\mathbf{Q}_c(t)$*.*

The matrix $\mathbf{L}(\mathbf{x}, t)\mathbf{Q}_c(t)\mathbf{L}^T(\mathbf{x}, t)$ *is then the covariance diffusion matrix of the stochastic differential equation.*

Definition A.2.6 (Differential Chapman-Kolmogorov Equation) .

Under appropriate assumptions, the Chapman-Kolmogorov equation can be reduced to a differential equation, and for the Differential Stochastic Equation [A.23] we find the

$$\begin{aligned}\frac{\partial p\big(\mathbf{x}, t|\mathbf{y}, s\big)}{\partial t} &= -\sum_i \frac{\partial}{\partial x_i}\big[f_i(\mathbf{x}, t)p(\mathbf{x}, t|\mathbf{y}, s)\big] \\ &+ \sum_{i,j} \frac{1}{2}\frac{\partial^2}{\partial x_i \partial x_j}\big[\mathbf{L}_{i,j}(\mathbf{x}, t)p(\mathbf{x}, t|\mathbf{y}, s)\big] \\ &+ \int \big[W(\mathbf{x}|\mathbf{z}, t)p(\mathbf{z}, t|\mathbf{y}, s) - W(\mathbf{z}|\mathbf{x}, t)p(\mathbf{x}, t|\mathbf{y}, s)\big]d\mathbf{z},\end{aligned} \tag{A.24}$$

where

$$W(\mathbf{z}|\mathbf{x}, t) = \lim_{\Delta t \to 0} \frac{p(\mathbf{z}, t + \Delta t|\mathbf{x}, t)}{\Delta t} \tag{A.25}$$

$$W(\mathbf{x}|\mathbf{z}, t) = \lim_{\Delta t \to 0} \frac{p(\mathbf{x}, t + \Delta t|\mathbf{z}, t)}{\Delta t} \tag{A.26}$$

uniformly in $\mathbf{x}$*,* $\mathbf{z}$*, and* t*, and for* $|\mathbf{x} - \mathbf{z}| \geq \epsilon$*.*

Definition A.2.7 (Diffusion Processes) .

If we assume the quantities $W(\mathbf{z}|\mathbf{x}, t)$ *and* $W(\mathbf{x}|\mathbf{z}, t)$ *to be zero, the differential Chapman-Kolmogorov equation reduces to the Fokker-Planck equation:*

$$\begin{aligned}\frac{\partial p\big(\mathbf{x}, t|\mathbf{y}, s\big)}{\partial t} &= -\sum_i \frac{\partial}{\partial x_i}\big[f_i(\mathbf{x}, t)p(\mathbf{x}, t|\mathbf{y}, s)\big] \\ &+ \sum_{i,j} \frac{1}{2}\frac{\partial^2}{\partial x_i \partial x_j}\big[\mathbf{L}_{i,j}(\mathbf{x}, t)p(\mathbf{x}, t|\mathbf{y}, s)\big]\end{aligned} \tag{A.27}$$

and the corresponding process is known as a diffusion process. The vector $\mathbf{f}(\mathbf{x}, t)$ *is known as the drift vector and the matrix* $\mathbf{L}(\mathbf{x}, t)$ *as a the diffusion matrix.*

A.3 The Ito Calculus and Stochastic Differential Equations

Definition A.3.1 (Stochastic Differential Equation) .

Stochastic Differential Equation (SDE) is an equation of the form

$$d\mathbf{x}(t) = \mathbf{f}(\mathbf{x}, t)dt + \mathbf{L}(\mathbf{x}, t)d\boldsymbol{\beta}(t), \tag{A.28}$$

where

- $\mathbf{f} : \mathbb{R}^n \times [0, \omega) \mapsto \mathbb{R}^n$ *is the drift function ,*
- $\mathbf{L} : \mathbb{R}^n \times [0, \infty) \mapsto R^{n*d}$ *is the dispersion matrix,*
- $\boldsymbol{\beta}(t)$ *is a* $d-$*dimensional Brownian motion with diffusion matrix* $\mathbf{Q}_c(t)$*.*

The matrix $\mathbf{L}(\mathbf{x}, t)\mathbf{Q}_c(t)\mathbf{L}^T(\mathbf{x}, t)$ *is then the diffusion matrix of the stochastic differential equation.*

The stochastic differential equation [A.28] is actually a short hand notation to the stochastic integral equation

$$\mathbf{x}(t) - \mathbf{x}(s) = \int_s^t \mathbf{f}(\mathbf{x}, t')dt' + \int_s^t \mathbf{L}(\mathbf{x}, t')d\boldsymbol{\beta}(t'), \tag{A.29}$$

where the last integral is an Ito stochastic integral. The stochastic process solution $\mathbf{x}(t)$ *to the stochastic differential equation is called Ito process .*

The *Stratonovich stochastic differential equations* [Jazwinski, 1970] are similar to Ito differential equations, but instead of Ito integrals they involve stochastic integrals in the Stratonovich sense. A Stratonovich stochastic differential equation can always be converted into an equivalent Ito equation by using simple transformation formulas [Jazwinski, 1970]. If the dispersion term is independent of the state $\mathbf{L}(\mathbf{x}, t) = \mathbf{L}(t)$, then the Ito and Stratonovich interpretations of the stochastic differential equation are the same.

To distinguish between Ito and Stratonovich stochastic differential equations, the Stratonovich integral is denoted by a small circle before the Brownian differential as follow:

$$d\mathbf{x} = \mathbf{f}(\mathbf{x}, t)dt + \mathbf{L}(\mathbf{x}, t) \circ d\boldsymbol{\beta}. \tag{A.30}$$

The white noise interpretation of SDEs naturally leads to stochastic differential equations in Stratonovich sense. This is because the discrete-time approximation of white noise driven differential equations converge to stochastic differential equations in Stratonovich sense, not in Ito sense. For this reason higher order numerical integrations schemes also approximate the corresponding

Stratonovich equation when applied to stochastic differential equations.

A solution to a stochastic differential equation is called *strong* if for given Brownian motion $\boldsymbol{\beta}(t)$ with filtration $\mathcal{F}_t$ it is possible to construct a solution $\mathbf{x}(t)$ which is $\mathcal{F}_t$-adapted. Uniqueness of a strong solution means that the paths of the process are unique for given Brownian motion and for this reason strong uniqueness is also called path-wise uniqueness.

A solution is called *weak* if it is possible to construct some Brownian motion $\widehat{\boldsymbol{\beta}}(t)$ and a stochastic process $\widehat{\mathbf{x}}(t)$ such that the pair is a solution to the stochastic differential equation. Weak uniqueness means that the probability law of the solution is unique, that is, there cannot be two solutions with different finite-dimensional distributions.

The required conditions for drift function $\mathbf{f}$ and dispersion matrix $\mathbf{L}$, which guarantee existences of strong and weak solutions can be found in the books of [Kloeden et al., 1992] and [Oksendal , 2003].

The most important tool for computing strong solutions to stochastic differential equations is the Ito formula, which can be interpreted as counterpart of the chain rule in ordinary calculus.

Definition A.3.2 (Taylor Series) .

Let's $\boldsymbol{\phi}\big(\mathbf{x}_t, t\big)$ a twice differentiable function such as $\mathbf{x}_t \in \mathbb{R}^n$, then

$$d\boldsymbol{\phi} = \frac{\partial \boldsymbol{\phi}}{\partial t} dt + \nabla \otimes \boldsymbol{\phi} + \frac{1}{2} d\mathbf{x}^T \mathbf{H}\big(\boldsymbol{\phi}\big) d\mathbf{x} \tag{A.31}$$

with the Jacobian

$$\nabla \otimes \boldsymbol{\phi} \doteq \frac{\partial \boldsymbol{\phi}}{\partial \mathbf{x}^T} = \begin{pmatrix} \frac{\partial \boldsymbol{\phi}}{\partial x_1} & \frac{\partial \boldsymbol{\phi}}{\partial x_2} & \cdots & \frac{\partial \boldsymbol{\phi}}{\partial x_n} \end{pmatrix} \tag{A.32}$$

and the Hessian

$$\mathbf{H}\big(\boldsymbol{\phi}\big) \doteq \frac{\partial^2 \boldsymbol{\phi}}{\partial \mathbf{x} \partial \mathbf{x}^T} = \begin{pmatrix} \frac{\partial^2 \boldsymbol{\phi}}{\partial x_1^2} & \frac{\partial^2 \boldsymbol{\phi}}{\partial x_1 \partial x_2} & \cdots & \frac{\partial^2 \boldsymbol{\phi}}{\partial x_1 \partial x_n} \\ \frac{\partial^2 \boldsymbol{\phi}}{\partial x_2 \partial x_1} & \frac{\partial^2 \boldsymbol{\phi}}{\partial x_2^2} & \cdots & \frac{\partial^2 \boldsymbol{\phi}}{\partial x_2 \partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 \boldsymbol{\phi}}{\partial x_n \partial x_1} & \frac{\partial^2 \boldsymbol{\phi}}{\partial x_n \partial x_2} & \cdots & \frac{\partial^2 \boldsymbol{\phi}}{\partial x_n \partial x_n} \end{pmatrix} \tag{A.33}$$

Definition A.3.3 (Itô Formula) .

Assume that the process $\mathbf{x}(t)$, $\mathbf{x}_t \in \mathbb{R}^n$, *is generated by the stochastic differential equation*

$$d\mathbf{x} = \mathbf{f}(\mathbf{x}, t)dt + \mathbf{L}(\mathbf{x}, t)d\boldsymbol{\beta}(t). \tag{A.34}$$

where $\boldsymbol{\beta}(t)$ *is a Brownian motion with diffusion matrix* $\mathbf{Q}_c(t)$.

Let $\mathbf{g}(., t)$, $\mathbf{g} \in \mathbb{R}^m$, *be a twice differentiable function. Then, by Taylor, the components of the stochastic process*

$$\mathbf{y}(t) = \mathbf{g}(\mathbf{x}(t), t) \tag{A.35}$$

satisfy the stochastic differential equations

$$dy_k = \frac{\partial g_k}{\partial t}dt + \sum_i \frac{\partial g_k}{\partial x_i}dx_i + \frac{1}{2}\sum_{i,j}\frac{\partial^2 g_k}{\partial x_i \partial x_j}dx_i dx_j, \tag{A.36}$$

where the terms $dx_i dx_j$ *are computed according to the rules*

$$dt^2 = 0, \tag{A.37a}$$
$$dtd\boldsymbol{\beta} = 0, \tag{A.37b}$$
$$d\boldsymbol{\beta}^T dt = 0, \tag{A.37c}$$
$$d\boldsymbol{\beta}d\boldsymbol{\beta}^T = \mathbb{E}\big[d\boldsymbol{\beta}d\boldsymbol{\beta}^T\big] = \mathbf{Q}_c(t)dt. \tag{A.37d}$$

Thus

$$d\mathbf{g} = \frac{\partial \mathbf{g}}{\partial t}dt + \frac{\partial \mathbf{g}}{\partial x^T}d\mathbf{x} + \frac{1}{2}d\boldsymbol{\beta}^T\mathbf{L}^T\frac{\partial^2 \mathbf{g}}{\partial \mathbf{x}\partial \mathbf{x}^T}\mathbf{L}d\boldsymbol{\beta} \tag{A.38}$$

or in matrix formulation

$$d\mathbf{g} = \frac{\partial \mathbf{g}}{\partial t}dt + \big(\nabla \otimes \mathbf{g}\big)d\mathbf{x} + \frac{1}{2}Tr\big[\mathbf{L}\mathbf{Q}_c\mathbf{L}^T\mathbf{H}\big(\mathbf{g}\big)\big]dt \tag{A.39}$$

or

$$d\mathbf{g} = \Big(\frac{\partial \mathbf{g}}{\partial t} + \big(\nabla \otimes \mathbf{g}\big)\mathbf{f} + \frac{1}{2}Tr\big[\mathbf{L}\mathbf{Q}_c\mathbf{L}^T\mathbf{H}\big(\mathbf{g}\big)\big]\Big)dt + \big(\nabla \otimes \mathbf{g}\big)\mathbf{L}d\boldsymbol{\beta}(t) \tag{A.40}$$

with the Jacobian

$$\nabla \otimes \mathbf{g} \doteq \frac{\partial \mathbf{g}}{\partial \mathbf{x}^T} = \begin{pmatrix} \frac{\partial g_1}{\partial x_1} & \frac{\partial g_1}{\partial x_2} & \dots & \frac{\partial g_1}{\partial x_n} \\ \frac{\partial g_2}{\partial x_1} & \frac{\partial g_2}{\partial x_2} & \dots & \frac{\partial g_2}{\partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial g_m}{\partial x_1} & \frac{\partial g_m}{\partial x_2} & \dots & \frac{\partial g_m}{\partial x_n} \end{pmatrix} \tag{A.41}$$

and the Hessian

$$\mathbf{H}(\mathbf{g}) \doteq \frac{\partial^2 \mathbf{g}}{\partial \mathbf{x} \partial \mathbf{x}^T} = \begin{pmatrix} \frac{\partial^2 \mathbf{g}}{\partial x_1^2} & \frac{\partial^2 \mathbf{g}}{\partial x_1 \partial x_2} & \cdots & \frac{\partial^2 \mathbf{g}}{\partial x_1 \partial x_{n_x}} \\ \frac{\partial^2 \mathbf{g}}{\partial x_2 \partial x_1} & \frac{\partial^2 \mathbf{g}}{\partial x_2^2} & \cdots & \frac{\partial^2 \mathbf{g}}{\partial x_2 \partial x_{n_x}} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 \mathbf{g}}{\partial x_{n_x} \partial \mathbf{x}_1} & \frac{\partial^2 \mathbf{g}}{\partial \mathbf{x}_{n_x} \partial x_2} & \cdots & \frac{\partial^2 \mathbf{g}}{\partial x_{n_x} \partial x_{n_x}} \end{pmatrix} \tag{A.42}$$

Definition A.3.4 (Stratonovich Formula) .

If the Equation [A.34] was a stochastic differential equation in Stratonovich sense

$$d\mathbf{x} = \mathbf{f}(\mathbf{x}, t)dt + \mathbf{L}(\mathbf{x}, t) \circ d\boldsymbol{\beta}, \tag{A.43}$$

then the differential would be

$$dy_k = \frac{\partial g_k}{\partial t} dt + \sum_i \frac{\partial g_k}{\partial x_i} \circ dx. \tag{A.44}$$

That is, the familiar result from calculus.

In Bayesian inference all information about the unknown quantities is assumed to be contained in the probability distribution of the unknown quantities. For this reason, when doing Bayesian inference on stochastic differential equations weak solutions to stochastic differential equations are often enough, because we are only interested in the probability laws, not the actual paths of the processes.

The probability distribution, that is, the law of any weak solution to a stochastic differential equation can be computed by the Kolmogorov forward equation:

Theorem A.3.1 (Kolmogorov forward equation) .

The Kolmogorov forward equation, also known as the Fokker-Planck equation, describes the time evolution of the probability density function for a system.

The probability density of the stochastic process $\mathbf{x}(t)$ *which is generated by the differential equation*

$$d\mathbf{x} = \mathbf{f}(\mathbf{x}, t)dt + \mathbf{L}(\mathbf{x}, t)d\boldsymbol{\beta}(t). \tag{A.45}$$

satisfies

$$\frac{\partial p}{\partial t} = -\sum_i \frac{\partial}{\partial x_i}\Big(f_i(\mathbf{x},t)p\Big) + \frac{1}{2}\sum_{i,j}\frac{\partial^2}{\partial x_i \partial x_j}\Big(\Big[\mathbf{L}(\mathbf{x},t)\mathbf{Q}_c(t)\mathbf{L}^T(\mathbf{x},t)\Big]_{ij}p\Big), \tag{A.46}$$

where $\mathbf{f}(\mathbf{x},t)$ *is the drift vector and* $\mathbf{L}(\mathbf{x},t)\mathbf{Q}_c(t)\mathbf{L}^T(\mathbf{x},t)$ *the diffusion tensor. The probability density* $p(\mathbf{x}(t)) = p(\mathbf{x},t)$ *is interpreted as function of* $\mathbf{x}$ *and* t*.*

The equation can also be written in the operator form

$$\frac{\partial p}{\partial t} = \mathcal{A}_t^*\big[p\big], \tag{A.47}$$

where the operator $\mathcal{A}_t^*$ *is*

$$\mathcal{A}_t^* = -\sum_i \frac{\partial}{\partial x_i}\Big(f_i(\mathbf{x},t)(.)\Big) + \frac{1}{2}\sum_{i,j}\frac{\partial^2}{\partial x_i \partial x_j}\Big(\Big[\mathbf{L}(\mathbf{x},t)\mathbf{Q}_c(t)\mathbf{L}^T(\mathbf{x},t)\Big]_{ij}(.)\Big), \tag{A.48}$$

which is the formal adjoint of the characteristic operator $\mathcal{A}_t$ *of the diffusion process.*

A.4 Ornstein-Uhlenbeck Process

Definition A.4.1 (Time-Independent Ornstein-Uhlenbeck Process) .

We define the Time-Independent Ornstein-Uhlenbeck process by the SDE

$$d\mathbf{x}(t) = -\mathbf{F}\mathbf{x}(t)dt + \mathbf{L}d\boldsymbol{\beta}(t), \tag{A.49}$$

where $\mathbf{F}$ *and* $\mathbf{L}$ *are constant matrices. The solution is given by*

$$\mathbf{x}(t) = \mathbf{x}(0)\exp\big[-\mathbf{F}\,t\big] + \int_0^t \exp\big[-\mathbf{F}\,(t-s)\big]\,\mathbf{L}d\boldsymbol{\beta}(s). \tag{A.50}$$

The mean is

$$\mathbb{E}\big[\mathbf{x}(t)\big] = \mathbb{E}\big[\mathbf{x}(0)\big]\exp\big[-\mathbf{F}\,t\big] \tag{A.51}$$

and the correlation function is

$$\begin{aligned}\mathbb{E}\big[\mathbf{x}(t),\mathbf{x}^T(s)\big] &\equiv \mathbb{E}\Big[\big(\mathbf{x}(t)-\mathbb{E}[\mathbf{x}(t)]\big)\big(\mathbf{x}(s)-\mathbb{E}[\mathbf{x}(s)]\big)^T\Big]\\ &= \exp\big[-\mathbf{F}\,t\big]\,\mathbb{E}\big[\mathbf{x}(0),\mathbf{x}^T(0)\big]\,\exp\big[-\mathbf{F}\,s\big]\\ &+ \int_0^{min(t,s)}\exp\big[-\mathbf{F}\,(t-t')\big]\,\mathbf{L}\mathbf{Q}_c\mathbf{L}^T\,\exp\big[-\mathbf{F}^T\,(s-t')\big]\,dt'.\end{aligned} \tag{A.52}$$

The integral can be explicitly evaluated in certain special cases, and for particular low-dimensional problems, it is possible to simply multiply everything out term by term.

Definition A.4.2 (Time-Dependent Ornstein-Uhlenbeck Process) .

We define the Time-Dependent Ornstein-Uhlenbeck process by the SDE

$$d\mathbf{x}(t) = -\mathbf{F}(t)\mathbf{x}(t)dt + \mathbf{L}(t)d\boldsymbol{\beta}(t), \tag{A.53}$$

The solution is given by

$$\begin{aligned}\mathbf{x}(t) = \mathbf{x}(0)\exp\big[-\int_0^t \mathbf{F}(s)ds\big] \\ + \int_0^t \Big\{\exp\big[-\int_{t'}^t \mathbf{F}(s)ds\big]\Big\}\,\mathbf{L}(t')d\boldsymbol{\beta}(t').\end{aligned} \tag{A.54}$$

which is very similar to the solution of the Time-Independent Ornstein-Uhlenbeck Process,

and we find

$$\mathbb{E}\big[\mathbf{x}(t)\big] = \mathbb{E}\big[\mathbf{x}(0)\big]\exp\big[-\int_0^t \mathbf{F}(s)ds\big] \tag{A.55}$$

$$\begin{aligned}\mathbb{E}\big[\mathbf{x}(t),\mathbf{x}^T(t)\big] = \exp\big[-\int_0^t \mathbf{F}(t')\,dt'\big]\,\mathbb{E}\big[\mathbf{x}(0),\mathbf{x}^T(0)\big]\,\exp\big[-\int_0^t \mathbf{F}(t')\,dt'\big] \\ + \int_0^t \exp\big[-\int_{t'}^t \mathbf{F}(s)\,ds\big]\,\mathbf{L}(t')\mathbf{Q}_c(t')\mathbf{L}^T(t')\,\exp\big[-\int_{t'}^t \mathbf{F}^T(s)\,ds\big]\,dt'.\end{aligned} \tag{A.56}$$

The time-dependent Ornstein-Uhlenbeck process arise very naturally in connection with the development of asymptotic methods in low-noise systems.

Theorem A.4.1 (Solution of Time-Independent Linear Stochastic Process) .

The solution of Time-Independent Linear Stochastic Process of the form

$$d\mathbf{x} = \mathbf{F}\mathbf{x}(t)dt + \mathbf{L}d\boldsymbol{\beta}(t), \tag{A.57}$$

where

- *the initial conditions are* $\mathbf{x}(0) \sim N\big(\mathbf{m}(0),\mathbf{P}(0)\big)$,
- $\mathbf{F}$ *and* $\mathbf{L}$ *are constant matrices,*
- $\boldsymbol{\beta}(t)$ *is a Brownian motion with constant diffusion matrix* $\mathbf{Q}_c$,

is a Gaussian process with the following mean $\mathbf{m}(t)$ *and covariance* $\mathbf{P}(t)$*:*

$$\mathbf{m}(t) = \exp(\mathbf{F}\ t)\mathbf{m}(0) \tag{A.58}$$

$$\mathbf{P}(t) = \exp(\mathbf{F}\ t)\mathbf{P}(0)\exp(\mathbf{F}^T\ t) + \int_0^t \exp\big(\mathbf{F}\ (t-\tau)\big)\ \mathbf{L}\mathbf{Q}_c\mathbf{L}^T\ \exp\big(\mathbf{F}^T\ (t-\tau)\big)\ d\tau, \tag{A.59}$$

where $\exp(.)$ *is the matrix exponential function.*

Theorem A.4.2 (Solution of Time-Dependent Linear Stochastic Process) .

A Time-Dependent Linear Stochastic Process of the form

$$d\mathbf{x} = \mathbf{F}(t)\mathbf{x}(t)dt + \mathbf{u}(t) + \mathbf{L}(t)d\boldsymbol{\beta}(t), \tag{A.60}$$

where

- *the initial conditions are* $\mathbf{x}(0) \sim N\big(\mathbf{m}(0), \mathbf{P}(0)\big)$*,*
- $\mathbf{F}(t)$ *and* $\mathbf{L}(t)$ *are matrix valued functions,*
- $\mathbf{u}(t)$ *is a known deterministic (non-random) function,*
- $\boldsymbol{\beta}(t)$ *is a Brownian motion with diffusion matrix* $\mathbf{Q}_c(t)$*,*

can be solved exactly using the ordinary differential equations

$$\frac{d\mathbf{m}(t)}{dt} = \mathbf{F}(t)\mathbf{m}(t) + \mathbf{u}(t), \tag{A.61}$$

$$\frac{d\mathbf{P}(t)}{dt} = \mathbf{F}(t)\mathbf{P}(t) + \mathbf{P}(t)\mathbf{F}^T(t) + \mathbf{L}(t)\mathbf{Q}_c(t)\mathbf{L}^T(t). \tag{A.62}$$

The solution is a Gaussian process with mean $\mathbf{m}(t)$ *and covariance* $\mathbf{P}(t)$*:*

$$p\big(\mathbf{x}(t)\big) = \mathcal{N}\big(\mathbf{x}(t)|\mathbf{m}(t), \mathbf{P}(t)\big). \tag{A.63}$$

Proof.

We take the expectation of Eq. [A.60], and interchanging the operators d *and* $\mathbb{E}$*, we get for the mean, with* $\mathbb{E}[d\boldsymbol{\beta}(t)] = 0$*,*

$$d\mathbf{m}_t = \Big(\mathbf{F}(t)\ \mathbb{E}[\mathbf{x}(t)]\ +\ \mathbf{u}(t)\Big)\ dt, \tag{A.64}$$

$$= \Big(\mathbf{F}(t)\ \mathbf{m}(t)\ +\ \mathbf{u}(t)\Big)\ dt, \tag{A.65}$$

and for the covariance

$$d\mathbf{P}(t) = d\big(\mathbb{E}[\mathbf{x}(t)\mathbf{x}^T(t)] - \mathbb{E}[\mathbf{x}(t)]\ \mathbb{E}[\mathbf{x}^T(t)]\big) = \mathbb{E}\big[d\big(\mathbf{x}(t)\mathbf{x}^T(t)\big)\big] - d\big(\mathbb{E}\big[\widehat{\mathbf{x}}(t)\big]\mathbb{E}\big[\widehat{\mathbf{x}}^T(t)\big]\big). \tag{A.66}$$

Applying Ito Eq. [A.36] to the elements of $\Big(\mathbf{x}(t)\mathbf{x}^T(t)\Big)$, *and* $\Big(\widehat{\mathbf{x}}(t)\widehat{\mathbf{x}}^T(t)\Big)$ *gives*

$$d\big(\mathbf{x}(t)\mathbf{x}^T(t)\big) = \mathbf{x}(t)\; d\mathbf{x}^T(t) \;+\; \big(d\mathbf{x}(t)\big)\; \mathbf{x}^T(t) \;+\; \mathbf{L}\; \mathbf{Q}_c\; \mathbf{L}^T\; dt, \quad \text{(A.67)}$$

$$d\big(\widehat{\mathbf{x}}(t)\widehat{\mathbf{x}}^T(t)\big) = \widehat{\mathbf{x}}(t)\; d\widehat{\mathbf{x}}^T(t) \;+\; \big(d\widehat{\mathbf{x}}(t)\big)\; \widehat{\mathbf{x}}^T(t), \quad \text{(A.68)}$$

and using Eq. [A.60] and [A.65] we find

$$\begin{aligned} d\mathbf{P}(t) &= \mathbb{E}\Big[\mathbf{x}(t)\big(\mathbf{x}^T(t)\mathbf{F}^T dt + d\boldsymbol{\beta}^T(t)\mathbf{L}^T\big) + \big(\mathbf{F}\mathbf{x}(t)dt + \mathbf{L}d\boldsymbol{\beta}(t)\big)\mathbf{x}^T(t) + \;\mathbf{L}\; \mathbf{Q}_c\; \mathbf{L}^T\; dt\Big] \\ &\quad -\widehat{\mathbf{x}}(t)\widehat{\mathbf{x}}^T(t)\mathbf{F}^T dt - \mathbf{F}\widehat{\mathbf{x}}(t)\widehat{\mathbf{x}}^T(t)dt \\ &= \Big(\mathbb{E}\big[\mathbf{x}(t)\mathbf{x}^T(t)\big] - \mathbf{x}(t)\mathbf{x}^T(t)\Big)\mathbf{F}^T dt + \mathbf{F}\Big(\mathbb{E}\big[\mathbf{x}(t)\mathbf{x}^T(t)\big] - \widehat{\mathbf{x}}(t)\widehat{\mathbf{x}}^T(t)\Big)dt \\ &\quad +\mathbf{L}\; \mathbf{Q}_c\; \mathbf{L}^T\; dt, \\ &= \mathbf{P}(t)\; \mathbf{F}^T \;+\; \mathbf{F}\; \mathbf{P}(t) + \mathbf{L}\; \mathbf{Q}_c\; \mathbf{L}^T\; dt \end{aligned} \quad \text{(A.69)}$$

Remark A.4.1 (Matrix fraction decomposition) *The covariance of linear time invariant stochastic differential equation [A.59] can be solved by using matrix fractions. If we define matrices* $\mathbf{C}$ *and* $\mathbf{D}$ *such as* $\mathbf{P} = \mathbf{C}\mathbf{D}^{-1}$, $\mathbf{P}$ *solves the matrix Riccati differential equation*

$$\frac{d\mathbf{P}(t)}{dt} = \mathbf{F}\mathbf{P}(t) + \mathbf{P}(t)\mathbf{F}^T + \mathbf{L}\mathbf{Q}_c\mathbf{L}^T. \quad \text{(A.70)}$$

if matrices $\mathbf{C}$ *and* $\mathbf{D}$ *solve the differential equation*

$$\begin{pmatrix} d\mathbf{C}(t)/dt \\ d\mathbf{D}(t)/dt \end{pmatrix} = \begin{pmatrix} \mathbf{F} & \mathbf{L}\mathbf{Q}_c\mathbf{L}^T \\ \mathbf{0} & -\mathbf{F}^T \end{pmatrix} \begin{pmatrix} \mathbf{C}(t) \\ \mathbf{D}(t) \end{pmatrix}, \quad \text{(A.71)}$$

and

$$\mathbf{P}(0) = \mathbf{C}(0)\; \mathbf{D}(0)^{-1}. \quad \text{(A.72)}$$

We can select, for example,

$$\mathbf{C}(0) = \mathbf{P}(0) \quad \text{(A.73)}$$

$$\mathbf{D}(0) = \mathbf{I}. \quad \text{(A.74)}$$

Because the differential equation [A.71] is linear and time invariant, it can be solved using the matrix exponential function:

$$\begin{pmatrix} \mathbf{C}(t) \\ \mathbf{D}(t) \end{pmatrix} = \exp\Big\{ \begin{pmatrix} \mathbf{F} & \mathbf{L}\mathbf{Q}_c\mathbf{L}^T \\ \mathbf{0} & -\mathbf{F}^T \end{pmatrix}(t) \Big\} \begin{pmatrix} \mathbf{C}(0) \\ \mathbf{D}(0) \end{pmatrix}, \quad \text{(A.75)}$$

The final solution is given as

$$\mathbf{P}(t) = \mathbf{C}(t)\; \mathbf{D}^{-1}(t). \quad \text{(A.76)}$$

Theorem A.4.3 (Discretization Time-Dependent Ornstein-Uhlenbeck Process)
.

The transition density of the linear differential equation [A.60] with $\mathbf{u}(t) = 0$ *can be written in form*

$$p\big(\mathbf{x}(t_{k+1})|\mathbf{x}(t_k)\big) = \mathcal{N}\big(\mathbf{x}(t_{k+1})|\mathbf{A}_k\mathbf{x}(t_k), \mathbf{Q}_k\big), \tag{A.77}$$

where the matrices $\mathbf{A}_k$ *and* $\mathbf{Q}_k$ *are the solutions* $\mathbf{A}_k \doteq \mathbf{A}(t_{k+1})$ *and* $\mathbf{Q}_k \doteq \mathbf{Q}(t_{k+1})$ *to the differential equations*

$$\frac{d\mathbf{A}(t)}{dt} = \mathbf{F}(t)\mathbf{A}(t) \tag{A.78}$$

$$\frac{d\mathbf{Q}(t)}{dt} = \mathbf{F}(t)\mathbf{Q}(t) + \mathbf{Q}(t)\mathbf{F}^T(t) + \mathbf{L}(t)\mathbf{Q}_c(t)\mathbf{L}^T(t), \tag{A.79}$$

with the initial conditions $\mathbf{A}(t_k) = \mathbf{I}$ *and* $\mathbf{Q}(t_k) = \mathbf{0}$. *The mean and covariance of the Gaussian process solution to the equation [A.60] at discrete time instances* $t_1, t_2, \cdots$ *are exactly given by the recursion equations*

$$\mathbf{m}_{k+1} = \mathbf{A}_k\mathbf{m}_k \tag{A.80}$$

$$\mathbf{P}_{k+1} = \mathbf{A}_k\mathbf{P}_k\mathbf{A}_k^T + \mathbf{Q}_k, \tag{A.81}$$

where $\mathbf{m}_k \doteq \mathbf{m}(t_k)$ *and* $\mathbf{P}_k \doteq \mathbf{P}(t_k)$.

Theorem A.4.4 (Discretization of Time-Independent Ornstein-Uhlenbeck Process)
.

In the case of Time-Independent Ornstein-Uhlenbeck Process, equation [A.57], the discretization equations can be explicitly solved:

$$\mathbf{A}_k = \exp\left(\mathbf{F}\ \Delta t_k\right) \tag{A.82}$$

$$\mathbf{Q}_k = \int_0^{\Delta t_k} \exp\left(\mathbf{F}\ (\Delta t_k - \tau)\right)\ \mathbf{L}\mathbf{Q}_c\mathbf{L}^T \exp\left(\mathbf{F}^T\ (\Delta t_k - \tau)\right)\ d\tau, \tag{A.83}$$

where $\Delta t_k = t_{k+1} - t_k$.

The matrix $\mathbf{Q}_k$ can be efficiently computed by the matrix fraction decomposition (see [A.4.1]) if the integral [A.83] cannot be computed in closed form.

This idea of discretization above is particularly useful in the case of the Kalman filter, because the canonical form of the Kalman filter has this kind of discrete dynamical model. The conclusion is that it does not matter that the Kalman filter was originally designed for discrete models, it still is exact for linear continuous-time dynamical models with discrete measurements.

A.5 Girsanov

Girsanov's theorem is an element of stochastic calculus which does not have an analogue in standard calculus. It is very important in the theory of financial mathematics to convert from the physical measure which describe the probability that an underlying instrument will take a particular value to the risk-neutral measure. It is also very useful for the treatment of Stochastic Differential Equations that describe the dynamic of physical or financial systems.

A.5.1 Change of measure

Let two probability spaces $\{\Omega, \mathcal{F}, P\}$ and $\{\Omega, \mathcal{F}, Q\}$. When we wish to compare two measures P and Q, we do not want either of them simply to throw information away; since when they are positive they can be related by the Radon-Nikodym derivative. This motivates the following definition of equivalence of two measures.

Definition A.5.1 (Measures) .

Two measures P and Q are said to be equivalent if they operate on the same sample space, and if A is any event in the sample space then $P(A) > 0 \Leftrightarrow Q(A) > 0$. In other words P is absolutely continuous with respect to Q and Q is absolutely continuous with respect to P.

Theorem A.5.1 ( ) .

Let $\boldsymbol{\beta}(t)$ a measurable process adapted to the filtration $\mathcal{F}_t$. If P and Q are equivalent measures, then the following results hold

$$\mathbb{E}_Q\Big(\boldsymbol{\beta}(t)\Big) = \mathbb{E}_P\Big(\frac{dQ}{dP}\boldsymbol{\beta}(t)\Big) \tag{A.84}$$

$$\mathbb{E}_Q\Big(\boldsymbol{\beta}(t)|\mathcal{F}_s\Big) = Z^{-1}(s)\mathbb{E}_P\Big(Z(t)\boldsymbol{\beta}(t)|\mathcal{F}_s\Big), \tag{A.85}$$

where

$$Z(s) = \mathbb{E}_P\Big(\frac{dQ}{dP}\Big|\mathcal{F}_s\Big). \tag{A.86}$$

Here $Z(s)$ is the Radon-Nikodym derivative of Q with respect to P. The first result basically shows that this is a martingale, and the second is a continuous time version of Bayes theorem.

Theorem A.5.2 (Girsanov) .

Assume that $\Big\{\boldsymbol{\theta}(t) \in \mathbb{R}_n : 0 \leq t \leq T\Big\}$ *is a* $\mathcal{F}-$*measurable process, which is adapted to the natural filtration* $\mathcal{F}_t \subset \mathcal{F}$ *of* n*-dimensional standard Brownian motion* $\Big\{\boldsymbol{\beta}(t) \in \mathbb{R}_n : 0 \leq t \leq T\Big\}$ *with respect to measure* P*.*

A necessary and sufficient condition for $\boldsymbol{\theta}$ *to be a martingale is Novikov's condition which requires that*

$$\mathbb{E}_P\Big[\exp\big(\int_0^t \|\boldsymbol{\theta}(t)\|^2 dt\Big)\Big] < \infty, \tag{A.87}$$

then

$$Z(t) = \exp\Big(\int_0^t \boldsymbol{\theta}^T(s) d\boldsymbol{\beta}(s) - \frac{1}{2}\int_0^t \|\boldsymbol{\theta}(s)\|^2 ds\Big), \tag{A.88}$$

satisfies the equation

$$Z(t) = 1 + \int_0^t Z(s)\boldsymbol{\theta}^T(s) d\boldsymbol{\beta}(s), \tag{A.89}$$

and is a martingale.

Then under the measure $Q(d\omega) = Z(t;\omega)P(d\omega)$ *the process*

$$\widetilde{\boldsymbol{\beta}}(t) = \boldsymbol{\beta}(t) - \int_0^t \boldsymbol{\theta}(s) ds, \tag{A.90}$$

is $n-$*dimensional standard Brownian motion.*

The random variable $Z(t;\omega)$ *is the likelihood ratio between laws* Q *and* P

$$\frac{dQ}{dP}(\omega)\Big|_{\mathcal{F}_t} = Z(t;\omega). \tag{A.91}$$

The Girsanov theorem can readily be applied to finding weak solutions and for removing drift from stochastic differential equations of the form:

$$d\mathbf{x} = \mathbf{f}(\mathbf{x}, t)dt + \mathbf{L}d\boldsymbol{\beta} \tag{A.92}$$

$$\mathbf{x}(0) = \mathbf{x}_0 \tag{A.93}$$

Theorem A.5.3 (Weak solution of SDE) .

Assume that the process $\mathbf{x}(t)$ *is generated by the stochastic differential equation [A.92].*

If we now define

$$Z(t;\omega) = \exp\Big(\int_0^t \big[\mathbf{L}^{-1}\mathbf{f}\big(\mathbf{x}_0 + \mathbf{L}\boldsymbol{\beta}(t;\omega), t\big)\Big]^T d\boldsymbol{\beta}(t;\omega) \\ -\frac{1}{2}\int_0^t \|\mathbf{L}^{-1}\mathbf{f}\big(\mathbf{x}_0 + \mathbf{L}\boldsymbol{\beta}(t;\omega), t\big)\|^2 dt\Big), \tag{A.94}$$

then the expectation of any function (or functional) $\mathbf{h}(.)$ *can be expressed as*

$$\mathbb{E}_P\Big[\mathbf{h}\big(\mathbf{x}(t)\big)\Big] = \mathbb{E}_P\Big[Z(t;\omega)\ \mathbf{h}\big(\mathbf{x}_0 + \mathbf{L}\boldsymbol{\beta}(t;\omega)\big)\Big]. \tag{A.95}$$

and thus $Z(t;\omega)$ *is the likelihood ratio between processes* $\mathbf{x}(t)$ *and* $\mathbf{x}_0 + \mathbf{L}\boldsymbol{\beta}(t;\omega)$.

Proof.

If we define

$$\widetilde{\mathbf{x}}(t) = \mathbf{x}_0 + \mathbf{L}\boldsymbol{\beta}(t;\omega) \tag{A.96}$$
$$\boldsymbol{\theta}(t) = \mathbf{L}^{-1}\mathbf{f}\big(\mathbf{x}_0 + \mathbf{L}\boldsymbol{\beta}(t), t\big) \tag{A.97}$$
$$\widetilde{\boldsymbol{\beta}}(t) = \boldsymbol{\beta}(t) - \int_0^t \boldsymbol{\theta}(s)ds \\ = \mathbf{L}^{-1}\big(\widetilde{\mathbf{x}}(t) - \mathbf{x}_0\big) - \int_0^t \mathbf{L}^{-1}\mathbf{f}\big(\widetilde{\mathbf{x}}(s), s\big)ds. \tag{A.98}$$

then by arranging the latest equation, we get that the processes $\widetilde{\mathbf{x}}(t)$ and $\widetilde{\boldsymbol{\beta}}(t)$ satisfy

$$d\widetilde{\mathbf{x}} = \mathbf{f}\big(\widetilde{\mathbf{x}}, t\big)dt + \mathbf{L}d\widetilde{\boldsymbol{\beta}}(t) \tag{A.99}$$
$$\widetilde{\mathbf{x}}(0) = \mathbf{x}_0. \tag{A.100}$$

By the Girsanov theorem, under the measure $Q(d\omega) = Z(t;\omega)P(d\omega)$ the process $\widetilde{\boldsymbol{\beta}}(t)$ is a Brownian motion and thus the pair $\big(\widetilde{\mathbf{x}}(t), \widetilde{\boldsymbol{\beta}}(t)\big)$ is a weak solution to the SDE.

For any function $\mathbf{h}(.)$ we have

$$\widetilde{\mathbb{E}}_Q\Big[\mathbf{h}\big(\widetilde{\mathbf{x}}(t)\big)\Big] = \mathbb{E}_P\Big[Z(t;\omega)\mathbf{h}\big(\mathbf{x}_0 + \mathbf{L}\boldsymbol{\beta}(t;\omega)\big)\Big]. \tag{A.101}$$

But by definition of a weak solution, we should have

$$\widetilde{\mathbb{E}}_Q\Big[\mathbf{h}\big(\widetilde{\mathbf{x}}(t)\big)\Big] = \mathbb{E}_P\Big[\mathbf{h}\big(\mathbf{x}(t)\big)\Big]. \tag{A.102}$$

Theorem A.5.4 (Removal of drift) .

Assume that the process $\mathbf{x}(t)$ *is generated by the stochastic differential equation [A.92]. If we define*

$$Z(t;\omega) = \exp\Big(-\int_0^t \Big[\mathbf{L}^{-1}\mathbf{f}\big(\mathbf{x}(t),t\big)\Big]^T d\boldsymbol{\beta}(t) \\ -\frac{1}{2}\int_0^t \|\mathbf{L}^{-1}\mathbf{f}\big(\mathbf{x}(t),t\big)\|^2 dt\Big), \tag{A.103}$$

Then under the measure $Q(d\omega) = Z(t;\omega)P(d\omega)$ *the process* $\mathbf{x}(t) - \mathbf{x}_0$ *is a Brownian motion with diffusion matrix* $\mathbf{L}\mathbf{L}^T$ *and thus the law of* $\mathbf{x}(t)$ *is the same as the law of* $\mathbf{x}_0 + \mathbf{L}\boldsymbol{\beta}(t)$.

Proof.

If we define

$$\boldsymbol{\theta}(t) = -\mathbf{L}^{-1}\mathbf{f}\big(\mathbf{x}(t),t\big) \tag{A.104}$$

$$Q(d\omega) = Z(t;\omega)P(d\omega), \tag{A.105}$$

then by Theorem [A.5.2], we get that under the measure Q

$$\begin{aligned}\widetilde{\boldsymbol{\beta}}(t) &= \boldsymbol{\beta}(t) - \int_0^t \boldsymbol{\theta}(s)ds \\ &= \mathbf{L}^{-1}\big(\mathbf{x}(t) - \mathbf{x}_0\big) - \int_0^t \mathbf{L}^{-1}\mathbf{f}\big(\mathbf{x}(s),s\big)ds + \int_0^t \mathbf{L}^{-1}\mathbf{f}\big(\mathbf{x}(s),s\big)ds \\ &= \mathbf{L}^{-1}\big(\mathbf{x}(t) - \mathbf{x}_0\big),\end{aligned} \tag{A.106}$$

is a standard Brownian motion and thus, $\mathbf{x}(t) - \mathbf{x}_0$ is a Brownian motion with diffusion matrix $\mathbf{L}\mathbf{L}^T$.

B

Kullback-Leibler-Divergence Sampling.

B.1 Adaptive Particle Filters

The idea is to determine the optimal number of samples N to approximate the true posterior, $p(\mathbf{x}_k|\mathbf{y}_{1:k})$, such as the error between the true posterior and its sample-based approximation, $p(\mathbf{x}_k^{(i)}|\mathbf{y}_{1:k})$ is less than ϵ with probability $1-\delta$, and using the Kullback-Leibler divergence as the distance between the sample-based maximum likelihood estimate (MLE) and the true posterior .

B.1.1 Kullback-Leibler divergence - KLD

In probability theory, the Kullback-Leibler divergence is a measure of the difference between two probability distributions: a "true" probability distribution P and an arbitrary probability distribution Q, where P represents observations or a calculated probability distribution, and Q represents a theory or an approximation of P, (see [Niklaus , 2004]; [Kullback , 1959]):

$$D_{KL}(P\|Q) = KL(P,Q) = \sum_i P(i) \log \frac{P(i)}{Q(i)} \,. \tag{B.1}$$

For distributions of a continuous random variable we have

$$D_{KL}(P\|Q) = KL(p,q) = \int_X p(x) \log \frac{p(x)}{q(x)} \,. \tag{B.2}$$

where p and q denote the probability density functions associated to the cumulative distribution functions P and Q respectively.

B.1.2 Multivariate Kernel density estimation

We consider a d-dimensional random state $\mathbf{x}_k$, for $k = 1, \ldots, n$, having a conditional density $p(\mathbf{x}_k|\mathbf{y}_{1:k})$, and a generic vector $\mathbf{x} \in R^d$.

Let a representation of the true posterior distribution, $p(\mathbf{x}_k|\mathbf{y}_{1:k})$, by a d-dimensional kernel density estimator,

$$\widehat{p}(\mathbf{x}, \mathbf{H}) = n^{-1} \sum_{k=1}^{n} K_{\mathbf{H}}(\mathbf{x} - \mathbf{x}_k) \tag{B.3}$$

in order to determine the number of samples N so that the distance between the sample-based maximum likelihood estimate (MLE), $p(\mathbf{x}_k^{(i)}|\mathbf{y}_{1:k})$, and the true posterior, $p(\mathbf{x}_k|\mathbf{y}_{1:k})$, does not exceed a pre-specified threshold ϵ. (The following description is adapted from [Gijbels , 1998]; and [Wand and Jones , 1995]).

Where $\mathbf{H}$ is a symmetric positive definite $(d \times d)$ bandwidth matrix, and

$$K_{\mathbf{H}}(\mathbf{x}) = |\mathbf{H}|^{-\frac{1}{2}} \mathcal{K}(\mathbf{H}^{-\frac{1}{2}}\mathbf{x}) \ , \tag{B.4}$$

and $\mathcal{K}$ is a d-variate kernel function satisfying

$$\int \mathcal{K}(\mathbf{y}) d\mathbf{y} = 1 \ . \tag{B.5}$$

For $\mathcal{K}$ we choose the standard d-variate normal density

$$\mathcal{K}(\mathbf{y}) = (2\pi)^{-\frac{d}{2}} \exp\big(-\frac{1}{2}\mathbf{y}^T\mathbf{y}\big) \ , \tag{B.6}$$

in which case $K_{\mathbf{H}}(\mathbf{x} - \mathbf{x}_k)$ is the $\mathcal{N}(\mathbf{x}_k, \mathbf{H})$ density in the vector $\mathbf{x}$.

A simple form of multivariate kernel density is obtained by choosing

$$\mathbf{H} = h^2\mathbf{I} \ , \tag{B.7}$$

with $\mathbf{I}$ the $(d \times d)$ identity matrix, and we get

$$\widehat{p}(\mathbf{x}, \mathbf{H}) = N^{-1}h^{-d} \sum_{j=1}^{b} \mathcal{K}\big(\frac{\mathbf{x} - \mathbf{x}_j}{h}\big) \ , \tag{B.8}$$

In this case, the Asymptotic Mean Integrated Squared Error (AMISE) is

$$AMISE\Big\{ \widehat{p}\,(\ .\ ;\ \mathbf{H}\ ) \Big\} = \frac{1}{b\,h^d} R(\mathcal{K}) + \frac{1}{4}\, h^4\, \mu_2^2 \int_{\mathcal{X}} \big\{\nabla^2 p(\mathbf{x})\big\}^2 d\mathbf{x} \ , \tag{B.9}$$

where

$$\nabla^2 p(\mathbf{x}) = \sum_{j=1}^{d} \frac{\partial^2}{\partial x_j^2} p(\mathbf{x}) \tag{B.10a}$$

$$R(\mathcal{K}) = \int \mathcal{K}^2(\mathbf{u}) d\mathbf{u} \tag{B.10b}$$

$$\mu_2 = \int \mathbf{u}\mathbf{u}^T \, \mathcal{K}(\mathbf{u}) d\mathbf{u} \, , \tag{B.10c}$$

and the asymptotically optimal bandwidth is

$$h_{AMISE} = \Big\{ \frac{dR(\mathcal{K})}{b \, \mu_2^2 \int \big\{ \nabla^2 p(\mathbf{x}) \big\}^2 d\mathbf{x}} \Big\}^{\frac{1}{d+4}} \, . \tag{B.11}$$

A quick and simple bandwidth selection rule is obtained by assuming that p belongs to the family of multivariate normal densities.

B.1.3 Adaptation

Now, we draw N samples from this distribution with b different bins to get an approximate of the true posterior distribution. (The following description is adapted from [Fox et al. , 2000], and [Fox et al. , 2001]).

Let

$$\boldsymbol{\nu} = (\nu_1, \nu_2, \ldots, \nu_b) \tag{B.12}$$

where ν_j, with $j \in [1, b]$, represents the number of samples for bin j, and with the constraint $\sum_{j=1}^{b} \nu_j = N$.

In such a case, $\boldsymbol{\nu}$ is distributed according to a multinomial distribution (MN)

$$\boldsymbol{\nu} \sim \mathrm{MN}_b(N, \mathbf{p}) \tag{B.13}$$

where $\mathbf{p} = (p_1, p_2, \ldots, p_b)^T$ specifies the true probability of each bin, with $\sum_{j=1}^{b} p_j = 1$.

The maximum likelihood estimate of $\mathbf{p}$, using N samples is given by the expectation

$$\widehat{\mathbf{p}} = N^{-1} \boldsymbol{\nu} \, . \tag{B.14}$$

Now, to estimate an optimal value of N, we will use the likelihood-ratio test. This test is a statistical test in which the ratio is computed between the maximum of the likelihood function under the null hypothesis and the maximum

with that constraint relaxed. If the likelihood ratio is λ and the null hypothesis holds, then for commonly occurring families of probability distributions, $-2\log(\lambda)$ has an asymptotic chi-square distribution.

The likelihood ratio test is given by

$$\lambda_N(\mathbf{p}) = \frac{L_N(\boldsymbol{\nu}, \mathbf{p}_0)}{\sup_{\mathbf{p}} L_N(\boldsymbol{\nu}, \mathbf{p})} , \tag{B.15}$$

and for large N, $-2\log\lambda_N(\mathbf{p})$ has approximately a χ^2 distribution with $b-1$ degrees of freedom.

$$-2\log\lambda_N(\mathbf{p}) \rightarrow \chi^2_{b-1} \quad \text{as} \quad N \rightarrow \infty . \tag{B.16}$$

Thus we have

$$\log\lambda_N(\mathbf{p}) = \sum_{j=1}^{b} \nu_j \log\frac{p_j}{\widehat{p}_j} , \tag{B.17}$$

and with Eq. (B.14), we have :

$$-\log\lambda_N(\mathbf{p}) = N\sum_{j=1}^{b} \widehat{p}_j \log\frac{\widehat{p}_j}{p_j} . \tag{B.18}$$

From Eq. (B.1) we get

$$-\log\lambda_N(\mathbf{p}) = N * \mathrm{KL}(\widehat{\mathbf{p}}, \mathbf{p}) . \tag{B.19}$$

We suppose than $\mathbf{p}$ is the true distribution, than using Eq. (B.16), and Eq. (B.19) we get

$$\begin{aligned} P_{\mathbf{p}}(\mathrm{KL}(\mathbf{p}, \widehat{\mathbf{p}}) \ \leq \ \epsilon) &= P_{\mathbf{p}}(2N * KL(\mathbf{p}, \widehat{\mathbf{p}}) \ \leq \ 2N * \epsilon) && \text{(B.20a)} \\ &= P_{\mathbf{p}}(-2\log\lambda_N(\mathbf{p})) \ \leq \ 2N * \epsilon) && \text{(B.20b)} \\ &\doteq P_{\mathbf{p}}(\chi^2_{b-1} \ \leq \ 2N * \epsilon) . && \text{(B.20c)} \end{aligned}$$

The quantiles of the χ^2 distribution are given by

$$P(\chi^2_{b-1} \ \leq \ \chi^2_{b-1,1-\delta}) = 1 - \delta . \tag{B.21}$$

We choose N such as $(2N * \epsilon)$ is equal to $\chi^2_{b-1,1-\delta}$ and combining Eq. (B.20) and Eq. (B.21), we get

$$P_{\mathbf{p}}(\mathrm{KL}(\mathbf{p}, \widehat{\mathbf{p}}) \ \leq \ \epsilon) \doteq 1 \ - \ \delta . \tag{B.22}$$

Thus we have got a relationship between the number of samples and the approximation of the real posterior.

If we choose the number of samples N such as

$$N = \frac{1}{2\epsilon}\chi^2_{b-1,1-\delta}\ , \tag{B.23}$$

than the Kullback-Leibler divergence between the MLE and the true posterior is less than ϵ.

But in order to determine N, we need to compute the quantiles of the χ^2 distribution.

The Wilson-Hilferty approximation [Severo and Zelen , 1960] gives a good approximation of a χ^2 distribution by a Normal distribution for $b \geq 30$, such as

$$\Big[\big(\frac{\chi^2_b}{b}\big)^{\frac{1}{3}} - \frac{2}{9b} - 1\Big]\sqrt{\frac{9b}{2}} \simeq \mathcal{N}(0;1) \tag{B.24}$$

which gives for the determination of the number of samples N to approximate the true posterior distribution

$$N = \frac{1}{2\epsilon}\chi^2_{b-1,1-\delta} \tag{B.25a}$$

$$\doteq \frac{b-1}{2\epsilon}\Big(z_{1-\delta}\sqrt{\frac{2}{9(b-1)}} + \frac{2}{9(b-1)} + 1\Big)^3\ , \tag{B.25b}$$

where $z_{1-\delta}$ is the upper $1-\delta$ quantile of the normal distribution.

We can see that the required number of particles to approximate the true posterior distribution is proportional to the inverse of the error bound ϵ, and to the first order linear in the number of bins b with support. We assume that a bin of the multinomial distribution has support if its probability is above a threshold, that means it contains at least one particle.

C

Monte Carlo Simulations.

C.1 Stochastic Differential Equations

In discrete-time Particle Filters, we generate particles from the distribution a posteriori of the state, $p(\mathbf{x}_n|\mathbf{y}_{1:n})$. In continuous-time Particle Filters, we generate paths of the state $\mathbf{x}(t)$ from the distribution à posteriori of the paths of this state given by the state's equation. Thus, we are interested in forming a Monte Carlo approximation to the probability of the state $\mathbf{x}(t)$, which is generated by the stochastic differential equation

$$d\mathbf{x}(t) = \mathbf{f}\big(\mathbf{x}(t), t\big)dt + \mathbf{L}\big(\mathbf{x}(t), t\big)d\boldsymbol{\beta}(t) \tag{C.1}$$

where :

- $\mathbf{f}\big(\mathbf{x}(t), t\big)$ is the drift coefficient,
- $\mathbf{L}\big(\mathbf{x}(t), t\big)$ is the diffusion coefficient, or noise term,
- $\boldsymbol{\beta}(t)$ is a Brownian motion with diffusion matrix $\mathbf{Q}_c(t)$

If the diffusion coefficient does not depend on $\mathbf{x}(t)$ we say the equation has additive noise, otherwise the equation has multiplicative noise.

A Wiener process $\boldsymbol{\beta} = \boldsymbol{\beta}(t), \quad 0 \leq t \leq T$ is a Gaussian process that depends continuously on time such as

1. $\boldsymbol{\beta}(0) = 0$ with probability 1,
2. for $0 \leq t \leq T, \quad E(\boldsymbol{\beta}(t) = 0$,
3. for $0 \leq s \leq t \leq T, \quad \text{Var}(\boldsymbol{\beta}(t) - \boldsymbol{\beta}(s) = t - s$,
4. for $0 \leq s \leq t \leq u \leq v \leq T$, the increments $\boldsymbol{\beta}(t) - \boldsymbol{\beta}(s)$ and $\boldsymbol{\beta}(u) - \boldsymbol{\beta}(v)$ are independent.

Equation (C.1) is interpreted as the stochastic integral equation

$$\mathbf{x}(t) = \mathbf{x}(t_0) + \int_{t_0}^{t} \mathbf{f}\big(\mathbf{x}(s), s\big)ds + \int_{t_0}^{t} \mathbf{L}\big(\mathbf{x}(s), s\big)d\boldsymbol{\beta}(s) \tag{C.2}$$

where the first integral is a regular Lebesgue integral and the second integral is a stochastic integral, interpreted in the Ito or Stratonovich form. We can move between the two calculi by means of a simple transformation, and the solution of the Ito Eq. (C.1) can be written as the solution of the Stratanovich equation

$$d\mathbf{x}(t) = \Big(\mathbf{f}\big(\mathbf{x}(t), t\big) - \frac{1}{2}\mathbf{L}\big(\mathbf{x}(t), t\big)\frac{\partial \mathbf{L}\big(\mathbf{x}(t), t\big)}{\partial \mathbf{x}^T}\Big)dt + \mathbf{L}\big(\mathbf{x}(t), t\big) \circ d\boldsymbol{\beta}(t) \tag{C.3}$$

which has a modified drift function. The Stratanovich calculus follows the usual rules of deterministic calculus, whereas the Ito calculus conveniently relates to martingale theory but has its own stochastic chain rule, the Ito formula.

C.1.1 One Dimension

We particularize the stochastic differential equations in one dimension, and we will use the symbolic Ito's formulation

$$dx_t = a\big(x_t, t\big)dt + b\big(x_t, t\big)dW_t \tag{C.4}$$

and also the stochastic integral equation

$$x_t = x_{t_0} + \int_{t_0}^{t} a\big(x_s, s\big)ds + \int_{t_0}^{t} b\big(x_s, s\big)dW_s \tag{C.5}$$

or the Stratanovich's formulation

$$dx_t = \Big(a\big(x_t, t\big) - \frac{1}{2}b\big(x_t, t\big)\frac{\partial b\big(x_t, t\big)}{\partial x^T}\Big)dt + b\big(x_t, t\big) \circ dW_t \tag{C.6}$$

C.2 Strong and Weak Convergence

To estimate the quality of a numerical scheme it is useful to known how the method approximates the true solution. In the stochastic case, unlike in the deterministic case, we can use the *strong* and the *weak* convergence.

C.2.1 Strong Convergence

In strong convergence, we use the concept of absolute error, which is the expectation of the absolute value of the difference between the numerical approximation and the true Ito solution at a final terminal time T, i.e.

$$\epsilon = \mathbb{E}\big(|x_T - y_N|\big). \tag{C.7}$$

A discrete time approximation y of the exact solution x converge in a strong sense with order $\gamma > 0$ if there exists constant $C < \infty$, which does not depend on Δ, such that

$$\epsilon(\Delta) = \mathbb{E}\big(|x_T - y_N|\big) \leq C\Delta^{\gamma} \tag{C.8}$$

for all fixed step-sizes $\Delta \in (0, 1)$.

C.2.2 Weak Convergence

Strong convergence is computationally expensive and not a necessary condition when only expected value type information about the solution is of interest, when, for example we only need of a moment of the solution x, we are not required to approximate individual paths of x which leads to the concept of weak convergence.

A general discrete time approximation y of a solution x converges in the weak sense with order $\beta > 0$ if, for any polynomial test functions g, there exists a constant $C < \infty$, which does not depend on Δ, such that

$$\big|\mathbb{E}\big(g(x_T)\big) - \mathbb{E}\big(g(y_N)\big)\big| \leq C\Delta^{\beta} \tag{C.9}$$

for all fixed step-sizes $\Delta \in (0, 1)$, provided these functionals exist. In a sense this criterion provides an error of the mean, variance or whatever moment is required. Generally it is easier and faster to implement numerical methods subject to this weak convergence condition.

But it is important when dealing a problem to identify whether a good pathwise approximation is required or only an approximation of some functional of the solution.

C.3 Stochastic Taylor Expansion

Much of deterministic numerical analysis for ordinary differential equations is based on manipulating and truncating Taylor expansions. Analogously, for SDEs we use a stochastic Taylor expansion, with different versions corresponding to the Ito and Stratonovich forms of stochastic calculus. The Ito-Taylor expansion is based on repeated iterations of the Ito formula.

For any twice continuously differentiable function $f : \mathbb{R} \to \mathbb{R}$ Ito's formula gives

$$f(x_t) = f(x_{t_0}) + \int_{t_0}^{t} L^0\ f(x_s)ds + \int_{t_0}^{t} L^1\ f(x_s)dW_s, \tag{C.10}$$

with the operators

$$L^0\ f = a\frac{df}{dx} + \frac{1}{2}b^2\frac{d^2f}{dx^2} \tag{C.11}$$

$$L^1 = b\ \frac{df}{dx}. \tag{C.12}$$

If we now apply the Ito formula, we get

$$\begin{aligned} x_t = x_{t_0} &+ \int_{t_0}^{t} \Big(a(x_{t_0}) + \int_{t_0}^{s} L^0\ a(x_z)dz + \int_{t_0}^{s} L^1\ a(x_z)dW_z\Big)ds \\ &+ \int_{t_0}^{t} \Big(b(x_{t_0}) + \int_{t_0}^{s} L^0\ b(x_z)dz + \int_{t_0}^{s} L^1\ b(x_z)dW_z\Big)dW_s \qquad \text{(C.13)} \\ &= x_{t_0} + a(x_{t_0})\int_{t_0}^{t} ds + b(x_{t_0})\int_{t_0}^{t} dW_s + R. \qquad \text{(C.14)} \end{aligned}$$

We can also apply the Ito formula to $f = L^1\ b$ in Eq. (C.14) to get

$$x_t = x_{t_0} + a(x_{t_0})\int_{t_0}^{t} ds + b(x_{t_0})\int_{t_0}^{t} dW_s + b(x_{t_0})b'(x_{t_0})\int_{t_0}^{t}\int_{t_0}^{s} dW_z dW_s + R. \tag{C.15}$$

For the purposes of numerical schemes, these integrals have to be evaluated and expressed in terms of different random variables. Methods relating to the evaluation of such integrals can be found in [Gardiner, 1990].

C.4 Strong Approximation

To apply a numerical scheme to the SDE Eq. (C.4), we must first discretize our time interval $[0;T]$, using a fixed step-size $\Delta = T/N$. This gives us a set of equally spaced points

$$0 = \tau_0 < \tau_1 < \cdots < \tau_n < \cdots < \tau_N = T,$$

which we use to approximate our solution.

C.4.1 Strong Taylor Schemes

In principle, arbitrarily many terms can be used to create schemes of a desired level of convergence. For practical implementation, however, this is at

the expense of evaluating more and more derivatives and stochastic integrals, leading to expressions that become very complicated as the desired order of convergence increases.

Consider, for example, the Taylor order 1.5 scheme for the SDE Eq. (C.4) as found in section 4.1C of [Kloeden and Platen, 1992]

$$\begin{aligned} y_{n+1} = y_n &+ a(y_n)\Delta_n + b(y_n)\Delta W_n \\ &+ b(y_n)b'(y_n)\ I(1,1) \\ &+ b(y_n)a'(y_n)\ I(1,0) \\ &+ \frac{1}{2}\Delta^2\Big(a(y_n)a'(y_n) + \frac{1}{2}b^2(y_n)a''(y_n)\Big) \\ &+ \Big(a(y_n)b'(y_n) + \frac{1}{2}b^2b''(y_n)\Big)\ I(0,1) \\ &+ b\Big(b(y_n)b''(y_n) + \big(b'(y_n)\big)^2\Big)\ I(1,1,1), \end{aligned} \tag{C.16}$$

where

$$I(1) = \int_{\tau_n}^{\tau_{n+1}} dW_s = \Delta W, \tag{C.17}$$

$$I(1,1) = \int_{\tau_n}^{\tau_{n+1}} \int_{\tau}^{s} dW_z dW_s = \frac{1}{2}\big((\Delta W)^2 - \Delta\big), \tag{C.18}$$

$$I(0,1) = \int_{\tau_n}^{\tau_{n+1}} \int_{\tau}^{s} ds dW_s = (\Delta W)\Delta - \Delta Z, \tag{C.19}$$

$$I(1,0) = \int_{\tau_n}^{\tau_{n+1}} \int_{\tau}^{s} dW_s ds = \Delta Z, \tag{C.20}$$

$$I(1,1,1) = \int_{\tau_n}^{\tau_{n+1}} \int_{\tau}^{s_3} \int_{\tau}^{s_2} dW_{s_1} dW_{s_2} dW_{s_3} = \frac{1}{2}\Big(\frac{1}{3}(\Delta W)^2 - \Delta\Big)\Delta W, \tag{C.21}$$

Here $\tau_{n+1} - \tau_n$ is the length of the step-size used of a discrete time approximation. The term ΔW is just the increment of the Wiener process, and ΔZ is a another normally distributed random variable with the required properties

$$\mathbb{E}(\Delta Z) = 0, \quad \text{for the mean} \tag{C.22}$$

$$\mathbb{E}\big((\Delta Z)^2\big) = \frac{1}{3}\Delta^3, \quad \text{for the variance,} \tag{C.23}$$

$$\mathbb{E}(\Delta Z\ \Delta W) = \frac{1}{2}\Delta^2, \quad \text{for the covariance.} \tag{C.24}$$

Upon evaluation of the integrals we find,

$$
\begin{aligned}
y_{n+1} = y_n &+ a(y_n)\Delta_n + b(y_n)\Delta W_n \\
&+ b(y_n)b'(y_n)\frac{1}{2}\big((\Delta W_n)^2 - \Delta_n\big) \\
&+ b(y_n)a'(y_n)\Delta Z_n \\
&+ \frac{1}{2}\Delta_n^2\Big(a(y_n)a'(y_n) + \frac{1}{2}b^2(y_n)a''(y_n)\Big) \\
&+ \Big(a(y_n)b'(y_n) + \frac{1}{2}b^2 b''(y_n)\Big)\Big((\Delta W_n)\Delta_n - \Delta Z_n\Big) \\
&+ b\Big(b(y_n)b''(y_n) + \big(b'(y_n)\big)^2\Big)\frac{1}{2}\Big(\frac{1}{3}(\Delta W_n)^2 - \Delta_n\Big)\Delta W_n, \quad \text{(C.25)}
\end{aligned}
$$

So it becomes clear that this approach becomes computationally more expensive as higher order methods are required.

C.4.2 Euler-Maruyama Method

The simplest example of a strong approximation y of the solution of Eq. (C.4) is the Euler-Maruyama method, which is of the form

$$y_{n+1} = y_n + a(y_n)\Delta + b(y_n)\Delta W, \quad \text{(C.26)}$$

for $n = 0, 1, 2, \cdots, N-1$ with $y_0 = x_0$, and $\Delta W = W_{\tau_{n+1}} - W_{\tau_n}$.

The Euler-Maruyama method converges with strong order $\gamma = 1/2$, and in many cases, the method provides a poor estimate of the solution, particularly in cases where the coefficients are non-linear, as is well documented in the deterministic Euler case. For more satisfactory levels of accuracy higher order schemes are required.

C.4.3 Milstein Method

By including from the Ito-Taylor expansion Eq. (C.15) the additional term

$$b(x_{t_0})\; b'(x_{t_0}) \int_{t_0}^{t}\int_{t_0}^{s} dW_z dW_s = b(x_{t_0})\; b'(x_{t_0})\; \frac{1}{2}\; \big((\Delta W)^2 - \Delta\big), \quad \text{(C.27)}$$

we obtain the Milstein scheme

$$y_{n+1} = y_n + a(y_n)\Delta + b(y_n)\Delta W + \frac{1}{2}b(y_n)b'(y_n)\big((\Delta W)^2 - \Delta\big), \quad \text{(C.28)}$$

which has strong order of convergence $\gamma = 1$.

So we see a considerable gain in efficiency by adding just one term from the Ito-Taylor expansion. However that for the case $b = 0$ the Milstein scheme

again reduces to the deterministic Euler method. The drawbacks of the Euler method for the numerical integration of ordinary differential equations are well known, and is generally not recommended for any complicated or non-linear equations. So, unless the SDE in question contains a linear or constant drift term the scheme should probably be avoided.

C.4.4 Strong Runge-Kutta approximations

Clearly, one of the principal problems in the practical implementation of higher order Taylor approximations is that derivatives of higher orders have to be evaluated. To this end, we consider another class of strong approximation methods known as stochastic Runge-Kutta schemes.

In the numerical analysis of ordinary differential equations, Runge-Kutta type schemes constitute one of the main tools for the accurate and fast approximation of a solution. These cannot be naively extended to stochastic differential equations and in general must contain multiple integrals that appear in stochastic Taylor expansions.

There is a large variety of Runge-Kutta type methods for approximating SDEs, with varying levels of convergence that are all derivative free. One of the first such examples was suggested by Platen in 1984 which replaces the derivatives in the Milstein scheme Eq. (C.28) by finite differences. In Ito form it is given by

$$\begin{aligned} y_{n+1} = y_n + a(y_n)\Delta_n + b(y_n)\Delta W_n \\ + \Big(b(\widehat{y}_n) - b(y_n)\Big)\frac{1}{2\sqrt{\Delta_n}}\big((\Delta W_n)^2 - \Delta_n\big), \end{aligned} \tag{C.29}$$

where

$$\widehat{y}_n = y_n + b(y_n)\ \sqrt{\Delta_n}. \tag{C.30}$$

C.5 Weak Approximations

It is not always necessary to simulate individual trajectories of a solution of a SDE, sometimes only information about moments or other functionals may be all that is required, in which case weak convergence is all that is required.

C.5.1 Weak Taylor Schemes

We can theoretically construct weak Taylor schemes of arbitrary order by the inclusion of appropriately many terms from stochastic Taylor expansions.

If we want to construct an order 2.0 weak Taylor scheme in the Ito case all terms with single and double integrals need to be included from the Ito-Taylor expansion, resulting in the following scheme

$$\begin{aligned} y_{n+1} = y_n &+ a(y_n)\Delta_n + b(y_n)\Delta W_n \\ &+ b(y_n)b'(y_n)\ I(1,1) \\ &+ b(y_n)a'(y_n)\ I(1,0) \\ &+ \Big(a(y_n)b'(y_n) + \frac{1}{2}b^2 b''(y_n)\Big)\ I(0,1) \\ &+ \frac{1}{2}\Delta^2\Big(a(y_n)a'(y_n) + \frac{1}{2}b^2(y_n)a'(y_n)\Big) \end{aligned} \tag{C.31}$$

which gives

$$\begin{aligned} y_{n+1} = y_n &+ a(y_n)\Delta_n + b(y_n)\Delta W_n \\ &+ b(y_n)b'(y_n)\frac{1}{2}\big((\Delta W_n)^2 - \Delta_n\big) \\ &+ b(y_n)a'(y_n)\Delta Z_n \\ &+ \Big(a(y_n)b'(y_n) + \frac{1}{2}b^2 b''(y_n)\Big)(\Delta W_n \Delta_n - \Delta Z_n) \\ &+ \frac{1}{2}\Delta_n^2\Big(a(y_n)a'(y_n) + \frac{1}{2}b^2(y_n)a'(y_n)\Big) \end{aligned} \tag{C.32}$$

C.5.2 Weak Runge-Kutta Method

As in the strong case, the principal disadvantage of weak Taylor schemes is the need to evaluate derivatives of various orders of the drift and diffusion coefficients at each time step in addition to the coefficients themselves. This can circumvented, again by Runge-Kutta type methods which avoid such evaluations, but as before the methods are not just simple extensions of classical Runge-Kutta formulae.

We present just one of example of a derivative free method with weak order of convergence 2.0, which was also suggested by Platen, taking the form

$$\begin{aligned} y_{n+1} = y_n &+ \big(a(\widehat{y}_n) + a(y_n)\big)\frac{1}{2}\Delta_n + \big(b(y_n^+) + b(y_n^-)\big)\frac{1}{4}\Delta\widehat{W}_n \\ &+ \big(b(y_n^+) - b(y_n^-)\big)\big((\Delta\widehat{W})^2 - \Delta_n\big)\frac{1}{4\sqrt{\Delta_n}}, \end{aligned} \tag{C.33}$$

where

$$\widehat{y}_n = y_n + a(y_n)\Delta_n + b(y_n)\Delta\widehat{W}_n, \tag{C.34}$$
$$y_n^+ = y_n + a(y_n)\Delta_n + b(y_n)\sqrt{\Delta_n}, \tag{C.35}$$
$$y_n^- = y_n + a(y_n)\Delta_n - b(y_n)\sqrt{\Delta_n}. \tag{C.36}$$

C.6 Higher Dimensions

Most of the above theory can be extended to higher dimensional stochastic processes and equations, though some adjustments may have to be made to the numerical schemes. Multiple stochastic integrals in terms of several Wiener processes can become quite complicated and may have to be approximated, and obviously with more dimensions come more stochastic differential equations that have to be solved and so schemes can take more time to compute.

The N-dimensional vector version of the SDE, in Ito form, driven by an M-dimensional Wiener process $\beta_t = (\beta_t^1, \cdots, \beta_t^M)$ has the following form:

$$d\mathbf{x}(t) = \mathbf{f}\big(\mathbf{x}(t), t\big)dt + \mathbf{L}\big(\mathbf{x}(t), t\big)d\boldsymbol{\beta}(t) \tag{C.37}$$

and the operators

$$L^0 = \frac{\partial}{\partial t} + \sum_{k=1}^{N} f^k \frac{\partial}{\partial x_k} + \sum_{k,l=1}^{N}\sum_{j=1}^{M} L^{k,j}L^{l,j}\frac{\partial^2}{\partial x_k \partial x_l}, \tag{C.38}$$

$$L^j = \sum_{k=1}^{N} L^{k,j}\frac{\partial}{\partial x_k}. \tag{C.39}$$

This allows us to express the multi-dimensional version of the Ito formula in a compact way. For a sufficiently smooth transformation $\mathbf{U} : [t_0, T] \times \mathbb{R}^N \to \mathbf{R}^k$ of the solution $\mathbf{X} = \{\mathbf{x}(t), t_0 \le t \le T\}$ we get a k-dimensional process $\mathbf{Y} = \{\mathbf{y}(t) = \mathbf{U}(\mathbf{x}(t), t), t_0 \le t \le T\}$ with the vector stochastic differential

$$d\mathbf{y}(t) = L^0 U(\mathbf{x}, t)dt + \sum_{j=1}^{M} L^j U(\mathbf{x}, t)d\beta^j(t). \tag{C.40}$$

C.6.1 Strong Numerical Schemes

Let's the extension of the Euler-Maruyama method Eq. (C.26) as applied to Eq. (C.37), for a time discretization subinterval $[\tau_n, \tau_{n+1}]$. In this case, the kth component of Euler scheme will be

$$y_{n+1}^k = y_n^k + f^k(\mathbf{x}_n, t_n)\ \Delta_n + \sum_{j=1}^{M} L^{k,j}(\mathbf{x}_n, t_n)\ \Delta\beta_n^j, \tag{C.41}$$

where

$$\Delta\beta_n^j = \int_{\tau_n}^{\tau_{n+1}} d\beta^j(t) = \beta^j(\tau_{n+1}) - \beta^j(\tau_n). \tag{C.42}$$

C.6.2 Weak Numerical Schemes

The Weak Euler scheme, or weak Taylor order 1.0 scheme for Eq. (C.37) has the expected form similar to Eq. (C.41), having kth component

$$y_{n+1}^k = y_n^k + f^k(\mathbf{x}_n, t_n)\Delta_n + \sum_{j=1}^{M} L^{k,j}(\mathbf{x}_n, t_n)\Delta\widehat{\beta}_n^j, \tag{C.43}$$

where

$$P\big(\widehat{\beta}_n^j = \pm\sqrt{\Delta_n}\big) = 0.5. \tag{C.44}$$

C.6.3 Weak Runge-Kutta Method

Consider a finite set of SDE's,

$$dx_t^j = f^j(\mathbf{x}_t, t)dt + \sum_{k=1}^{M} L^{j,k}(\mathbf{x}_t, t)d\beta_t^k, \tag{C.45}$$

for $j \in [1, N]$, from Ito we get

$$x_{t+dt}^j = x_t^j\ +\ \sum_{k=1}^{M} \frac{\partial x_t^j}{\partial \beta_t^k}\ d\beta_t^k + \frac{1}{2}\sum_{k,l=1}^{M} \frac{\partial^2 x_t^k}{\partial\beta_t^k d\beta_t^l}\ d\beta_t^k\ d\beta_t^l. \tag{C.46}$$

In a mean-square sense the product of differentials $d\beta_t^k\ d\beta_t^l = \delta_{k,l}dt$, and we get

$$\begin{aligned} dx_{t+dt}^j &= x_{t+dt}^j - x_t^j \\ &= \Big[\frac{\partial x_t^j}{\partial t}\ +\ \frac{1}{2}\ \sum_{k=1}^{M}\ \frac{\partial^2 x_t^j}{\partial(\beta_t^k)^2}\Big]dt\ +\ \sum_{k=1}^{M}\ \frac{\partial x_t^j}{\partial\beta_t^k}\ d\beta_t^k, \end{aligned} \tag{C.47}$$

with

$$\frac{\partial x_t^j}{\partial \beta_t^k} = L^{j,k}(\mathbf{x}_t, t), \tag{C.48}$$

$$\frac{\partial x_t^j}{\partial t} = f^j(\mathbf{x}_t, t) \; - \; \frac{1}{2} \sum_{k=1}^{M} \sum_{i=1}^{N} \; L^{i,k}(\mathbf{x}_t, t) \; \frac{\partial L^{j,k}(\mathbf{x}_t, t)}{\partial x_t^i}. \tag{C.49}$$

Now that these first-order derivatives are expressed in terms of f^j and $L^{j,k}$, higher-order derivatives can be computed. Thus a Taylor expansion of the solutions

$$x_{t+\Delta t}^j = x_t^j \; + \; \frac{\partial x_k^j}{\partial t} \; \Delta t \; + \; \sum_{k=1}^{M} \; \frac{\partial x_k^j}{\partial \beta_t^k} \; \Delta \beta_t^k + \frac{1}{2} \; \sum_{k,l=1}^{M} \; \frac{\partial^2 x_t^j}{\partial \beta_t^k \; \partial \beta_t^l} \Delta \beta_t^k \; \Delta \beta_t^l \; + \; \cdots \tag{C.50}$$

can be obtained for finite displacements Δt and $\Delta \beta_t^k$.

For given displacements Δt and $\Delta \beta_t^k$ define

$$g_j(x_t, t) = \frac{\partial x_t^j}{\partial t} \; \Delta t \; + \; \sum_{k=1}^{M} \; \frac{\partial x_t^j}{\partial \beta_t^k} \; \Delta \beta_t^k \tag{C.51}$$

$$= \Big[f^j(\mathbf{x}_t, t) \; - \frac{1}{2} \; \sum_{k=1}^{M} \sum_{j=1}^{N} \; L^{k,j}(x_t, t) \; \frac{\partial L^{k,j}(\mathbf{x}_t, t)}{\partial x_t^j} \Big] \; \Delta t$$

$$+ \sum_{k=1}^{M} \; L^{k,j}(\mathbf{x}_t, t) \; \Delta \beta_t^k, \tag{C.52}$$

and consider the four-stage approximation

$$K_j^1 = g_j(\mathbf{x}_{t_k}, t_k), \tag{C.53}$$

$$K_j^2 = g_j\Big(\mathbf{x}_{t_k} + \frac{1}{2}\mathbf{K}^1 \; , \; t_k + \frac{1}{2}\Delta t\Big), \tag{C.54}$$

$$K_j^3 = g_j\Big(\mathbf{x}_{t_k} + \frac{1}{2}\mathbf{K}^2 \; , \; t_k + \frac{1}{2}\Delta t\Big), \tag{C.55}$$

$$K_j^4 = g_j\Big(\mathbf{x}_{t_k} + \mathbf{K}^3 \; , \; t_k + \Delta t\Big), \tag{C.56}$$

$$\mathbf{x}_{t_{k+1}} = \mathbf{x}_{t_k} \; + \; \frac{1}{6} \; \Big(\mathbf{K}^1 \; + \; 2\mathbf{K}^2 \; + \; 2\mathbf{K}^3 \; + \; \mathbf{K}^4\Big). \tag{C.57}$$

Algorithm C.6.1 (Stochastic Runge-Kutta method) .

The stochastic weak fourth order, strong second order Runge-Kutta method can be implemented by defining the function

$$\widehat{\mathbf{f}}_j(\mathbf{x}, t, \Delta\boldsymbol{\beta}) = f_j(\mathbf{x}, t) \; - \; \frac{1}{2} \sum_{i,k} L_{i,k}(\mathbf{x}, t) \; \frac{\partial L_{j,k}(\mathbf{x}, t)}{\partial x_i} \; + \; \sum_{k} L_{j,k}(\mathbf{x}, t) \; \frac{\Delta\beta_k}{\Delta t}. \tag{C.58}$$

On each step k do the following:

- *Draw random variable $\Delta\boldsymbol{\beta}_k$ from the distribution $(t_k = k\Delta t)$*

$$\Delta\beta_k \sim N\big(\mathbf{0}, \mathbf{Q}(t_k)\Delta t\big). \tag{C.59}$$

- *Compute*

$$\Delta\mathbf{x}^1 = \widehat{\mathbf{f}}\big(\mathbf{x}_k \; , \; t_k \; , \; \Delta\boldsymbol{\beta}_k\big) \; \Delta t \tag{C.60}$$

$$\Delta\mathbf{x}^2 = \widehat{\mathbf{f}}\big(\mathbf{x}_k + \frac{1}{2}\Delta\mathbf{x}^1 \; , \; t_k + \frac{1}{2}\Delta t \; , \; \Delta\boldsymbol{\beta}_k\big) \; \Delta t \tag{C.61}$$

$$\Delta\mathbf{x}^3 = \widehat{\mathbf{f}}\big(\mathbf{x}_k + \frac{1}{2}\Delta\mathbf{x}^2 \; , \; t_k + \frac{1}{2}\Delta t \; , \; \Delta\boldsymbol{\beta}_k\big) \; \Delta t \tag{C.62}$$

$$\Delta\mathbf{x}^4 = \widehat{\mathbf{f}}\big(\mathbf{x}_k + \Delta\mathbf{x}^3 \; , \; t_k + \Delta t \; , \; \Delta\boldsymbol{\beta}_k\big) \; \Delta t \tag{C.63}$$

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \frac{1}{6}\big(\Delta\mathbf{x}^1 + 2\Delta\mathbf{x}^2 + 2\Delta\mathbf{x}^3 + \Delta\mathbf{x}^4\big). \tag{C.64}$$

The idea of this algorithm is that the Ito SDE is actually converted into the corresponding Stratonovich differential equation (hence the correction term in $\widetilde{\mathbf{f}}$).
The fortunate property of this Stratonovich form is that the Taylor series for functions can be formed in the same way as in deterministic case. Thus the Runge-Kutta method can be derived in the same way as in deterministic case, but now the strong order is half, because $\Delta\boldsymbol{\beta}^2$ is of the order $\mathcal{O}\big(\Delta t\big)$.

D

Parameter Estimation of a Linear Model by Gauss-Newton.

In this appendix, the parameter estimation of a linear state Space Model is realized by Gauss-Newton optimization. In Section [5.3] this procedure was used for initial parameter estimation of a non stationary, linear model in Continuous-discrete time representation, and sketched without derivation. Derivations for these expressions are provided below.

Let consider a non stationary, linear dynamical system with a state space representation. The system matrices are represented by an hyperparameter θ. We can realize a parameter estimation by derivative of the maximum likelihood and optimization by Gauss-Newton procedure.

D.1 State Space Model

Let consider a non stationary, linear dynamical model:

$$\mathbf{x}_{k+1} = \mathbf{A}_k(\boldsymbol{\theta})\mathbf{x}_k + \mathbf{B}_k(\boldsymbol{\theta})\mathbf{u}_k + \mathbf{G}_k(\boldsymbol{\theta})\mathbf{v}_k \tag{D.1}$$

$$\mathbf{y}_k = \mathbf{C}_k(\boldsymbol{\theta})\mathbf{x}_k + \mathbf{D}_k(\boldsymbol{\theta})\mathbf{u}_k + \mathbf{n}_k, \tag{D.2}$$

where :

- $\mathbf{x}_k \in \mathbb{R}^n$, $\mathbf{y}_k \in \mathbb{R}^q$, $\mathbf{u}_k \in \mathbb{R}^m$, $\mathbf{v}_k \in \mathbb{R}^p$, $\mathbf{n}_k \in \mathbb{R}^q$,
- $\mathbf{A}_k(\boldsymbol{\theta})$ $[n, n]$, $\mathbf{B}_k(\boldsymbol{\theta})$ $[n, m]$, $\mathbf{G}_k(\boldsymbol{\theta})$ $[n, p]$, $\mathbf{C}_k(\boldsymbol{\theta})$ $[q, n]$, $\mathbf{D}_k(\boldsymbol{\theta})$ $[q, m]$,
- these system matrices depend on a set of unknown parameters $\boldsymbol{\theta} \in \mathbb{R}^J$ where J is the number of hyperparameters,
- $\mathbf{x}_k$ is the state at time k,
- $\mathbf{y}_k$ is the observation at time k,
- $\{\mathbf{u}_k\}$ a deterministic input,
- $\{\mathbf{v}_k\}$, $\{\mathbf{n}_k\}$ the system, and measurement white noises, with $\mathbf{v}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}(\boldsymbol{\theta}))$, $\mathbf{n}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{R}(\boldsymbol{\theta}))$,
- $\mathbf{x}_0 \sim \mathcal{N}(\overline{\mathbf{x}_0}, \mathbf{P}_0)$.

D.2 Kalman Filter

The Kalman filter will be used in the derivatives of the likelihood in next sections. It is recalled here to make easier the understanding of the procedure.

Once a model has been put in a state space form, we can use the Kalman filter, in a recursive procedure to compute the optima estimator of the state at time k, based on the information available at time k. This information consists of the observations up to and including $\mathbf{y}_k$.

Initialisation

$$\widehat{\mathbf{x}}_{0|-1} = \mathbb{E}\{\mathbf{x}_0\} = \overline{\mathbf{x}_0} \tag{D.3}$$
$$\mathbf{P}_{0|-1} = Var\{\mathbf{x}_0\} \tag{D.4}$$

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

$$\widehat{\mathbf{x}}_{k|k-1} = \mathbf{A}_{k-1}\widehat{\mathbf{x}}_{k-1|k-1} + \mathbf{B}_{k-1}\mathbf{u}_{k-1} \tag{D.5}$$
$$\mathbf{P}_{k|k-1} = \mathbf{A}_{k-1}\mathbf{P}_{k-1|k-1}\mathbf{A}_{k-1}^T + \mathbf{G}_{k-1}\mathbf{Q}_{k-1}\mathbf{G}_{k-1}^T \tag{D.6}$$

Update

$$\mathbf{\Sigma}_{k|k-1} = \mathbf{C}_k\mathbf{P}_{k|k-1}\mathbf{C}_k^T + \mathbf{R}_k \tag{D.7}$$
$$\mathbf{K}_k = \mathbf{P}_{k|k-1}\mathbf{C}_k^T\mathbf{\Sigma}_{k|k-1}^{-1} \tag{D.8}$$
$$\mathbf{e}_k = \mathbf{y}_k - \mathbf{D}_k\mathbf{u}_k - \mathbf{C}_k\widehat{\mathbf{x}}_{k|k-1} \tag{D.9}$$
$$\widehat{\mathbf{x}}_{k|k} = \widehat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k\mathbf{e}_k \tag{D.10}$$
$$\mathbf{P}_{k|k} = \mathbf{P}_{k|k-1} - \mathbf{K}_k\mathbf{C}_k\mathbf{P}_{k|k-1} \tag{D.11}$$

D.3 Likelihood

We can use the Kalman filter if we know, at each iteration, the value of hyperparameter $\boldsymbol{\theta}$ to define the system matrices, and we use the maximum likelihood to estimate this hyperparameter.

The classical theory of maximum likelihood is based on a situation in which the set of observations $\{\mathbf{y}_0, \mathbf{y}_1, \cdots, \mathbf{y}_N\}$ are independently and identically distributed. The joint density function is therefore given by:

$$L_N(\mathbf{y}; \boldsymbol{\theta}) = \prod_{k=1}^{N} p(\mathbf{y}_k), \tag{D.12}$$

where $p(\mathbf{y}_k)$ is the joint probability function of the k-th set of observations $\mathbf{y}_k$. Once the observations have been made, $L_N(\boldsymbol{\theta};\mathbf{y})$ is interpreted as a likelihood function and the Maximum Likelihood estimator is found by maximizing this function with respect to θ given the observations $\mathbf{y}$.

The principal characteristic of a time series is that the observations are not independent. Hence Eq. (D.12) is not applicable. Instead the definition of a conditional probability density function is used to write the joint density functions:

$$L_N(\mathbf{y};\boldsymbol{\theta}) = \prod_{k=1}^{N} p(\mathbf{y}_k|\mathbf{y}_{1:k-1}), \tag{D.13}$$

where $p(\mathbf{y}_k|\mathbf{y}_{k|k-1})$, denotes the distribution of y_k conditional on the information set at time $k-1$, $\{\mathbf{y}_0, \mathbf{y}_1, \cdots, \mathbf{y}_{k-1}\}$. If the noises and initial state have proper normal distributions, the conditional distribution $L_N(\mathbf{y};\theta)$ is itself normal, and the mean and covariance matrix of the conditional distribution are given by the Kalman filter.

D.3.1 Computation of the Likelihood function

We realize a computation of the likelihood from Kalman filter equations.

$$L_N(\boldsymbol{\theta};\mathbf{y}) = \prod_{k=1}^{N} p(\mathbf{y}_k|\mathbf{y}_{1:k-1}). \tag{D.14}$$

From measurement equation we have :

$$\begin{aligned}\mathbf{y}_k &= \mathbf{C}_k(\boldsymbol{\theta})\mathbf{x}_k + \mathbf{D}_k(\boldsymbol{\theta})\mathbf{u}_k + \mathbf{v}_k &\text{(D.15)}\\ &= \mathbf{C}_k\widehat{\mathbf{x}}_{k|k-1} + \mathbf{C}_k\big[\mathbf{x}_k - \widehat{\mathbf{x}}_{k|k-1}\big] + \mathbf{D}_k\mathbf{u}_k + \mathbf{v}_k. &\text{(D.16)}\end{aligned}$$

The conditional mean and covariance are given by:

$$\begin{aligned}\mathbb{E}\big\{\mathbf{y}_k|\mathbf{y}_{1:k-1}\big\} &= \mathbf{C}_k\widehat{\mathbf{x}}_{k|k-1} + \mathbf{D}_k\mathbf{u}_k \doteq \widehat{\mathbf{y}}_k &\text{(D.17)}\\ Var\big\{\mathbf{y}_k|\mathbf{y}_{1:k-1}\big\} &= \mathbb{E}\big[(\mathbf{y}_k - \widehat{\mathbf{y}}_{k|k-1})(\mathbf{y}_k - \widehat{\mathbf{y}}_{k|k-1})^T\big]. &\text{(D.18)}\end{aligned}$$

We define the innovations:

$$\mathbf{e}_k = \mathbf{y}_k - \widehat{\mathbf{y}}_{k|k-1}. \tag{D.19}$$

Hence we use these innovations in the likelihood equation to get:

$$\begin{aligned}Var\big\{\mathbf{y}_k|\mathbf{y}_{1:k-1}\big\} &= \boldsymbol{\Sigma}_{k|k-1} &\text{(D.20)}\\ p(\mathbf{y}_k|\mathbf{y}_{1:k-1}) &= \frac{1}{(2\pi)^{\frac{q}{2}}}\frac{1}{\mid\boldsymbol{\Sigma}_{k|k-1}\mid^{\frac{1}{2}}}exp\Big[-\frac{1}{2}\mathbf{e}_k^T\boldsymbol{\Sigma}_{k|k-1}^{-1}\mathbf{e}_k\Big] &\text{(D.21)}\end{aligned}$$

$$\log L = -\frac{Nq}{2}\log(2\pi) - \frac{1}{2}\sum_{k=1}^{N}\log \mid \mathbf{\Sigma}_{k|k-1} \mid -\frac{1}{2}\sum_{k=1}^{N}(\mathbf{e}_k^T\mathbf{\Sigma}_{k|k-1}^{-1}\mathbf{e}_k) \quad \text{(D.22)}$$

$$(\log L)_k \doteq l_k = -\frac{q}{2}\log(2\pi) - \frac{1}{2}\log \mid \mathbf{\Sigma}_{k|k-1} \mid -\frac{1}{2}(\mathbf{e}_k^T\mathbf{\Sigma}_{k|k-1}^{-1}\mathbf{e}_k). \quad \text{(D.23)}$$

By the Woodbury's lemma we get :

$$\mathbf{\Sigma}_{k|k-1} = \mathbf{C}_k\mathbf{P}_{k|k-1}\mathbf{C}_k^T + \mathbf{R}_k \quad \text{(D.24)}$$

$$\mathbf{\Sigma}_{k|k-1}^{-1} = \mathbf{R}_k^{-1} - \mathbf{R}_k^{-1}\mathbf{C}_k\Big[\mathbf{P}_{k|k-1}^{-1} + \mathbf{C}_k^T\mathbf{R}_k^{-1}\mathbf{C}_k\Big]^{-1}\mathbf{C}_k^T\mathbf{R}_k^{-1}. \quad \text{(D.25)}$$

Hence the last term of the likelihood is:

$$\mathbf{e}_k^T\mathbf{\Sigma}_{k|k-1}^{-1}\mathbf{e}_k = tr\big[\mathbf{R}_k^{-1}\mathbf{e}_k\mathbf{e}_k^T\big] - \big[\mathbf{C}_k^T\mathbf{R}_k^{-1}\mathbf{e}_k\big]^T\big[\mathbf{P}_{k|k-1}^{-1} + \mathbf{C}_k^T\mathbf{R}_k^{-1}\mathbf{C}_k\big]^{-1}\big[\mathbf{C}_k^T\mathbf{R}_k^{-1}\mathbf{e}_k\big]. \quad \text{(D.26)}$$

Hence the k-th contribution of the log-likelihood is given by:

$$\begin{aligned} l_k = &-\frac{q}{2}\log(2\pi) - \frac{1}{2}log|\mathbf{\Sigma}_{k|k-1}| - \frac{1}{2}tr\big[\mathbf{R}_k^{-1}\mathbf{e}_k\mathbf{e}_k^T\big] \\ &+\frac{1}{2}\big[\mathbf{C}_k^T\mathbf{R}_k^{-1}\mathbf{e}_k\big]^T\big[\mathbf{P}_{k|k-1}^{-1} + \mathbf{C}_k^T\mathbf{R}_k^{-1}\mathbf{C}_k\big]^{-1}\big[\mathbf{C}_k^T\mathbf{R}_k^{-1}\mathbf{e}_k\big]. \end{aligned} \quad \text{(D.27)}$$

D.3.2 Derivatives of the Likelihood function

We must realize the derivative of the likelihood, to this end we use the following equations, let a symmetric matrix $\mathbf{A}$ and a variable x, we have the expressions:

$$\frac{\partial|\mathbf{A}|}{\partial x} = |\mathbf{A}|tr\big[\mathbf{A}^{-1}\frac{\partial\mathbf{A}}{\partial x}\big] \quad \text{(D.28)}$$

$$\frac{\partial\mathbf{A}^{-1}}{\partial x} = -\mathbf{A}^{-1}\frac{\partial\mathbf{A}}{\partial x}\mathbf{A}^{-1}. \quad \text{(D.29)}$$

Differentiating the k-th component of the log-likelihood gives:

$$\frac{\partial l_k}{\partial\theta_i} = -\frac{1}{2}\frac{\partial\log \mid \mathbf{\Sigma}_{k|k-1} \mid}{\partial\theta_i} - \frac{1}{2}\frac{\partial\big(\mathbf{e}_k^T\mathbf{\Sigma}_{k|k-1}^{-1}\mathbf{e}_k\big)}{\partial\theta_i} \quad \text{(D.30)}$$

and for $i = 1, 2, \cdots, J$, we find:

$$\frac{\partial \log \mid \mathbf{\Sigma}_{k|k-1} \mid}{\partial \theta_i} = \mid \mathbf{\Sigma}_{k|k-1} \mid^{-1} \frac{\partial \mid \mathbf{\Sigma}_{k|k-1} \mid}{\partial \theta_i} = tr\Big[\mathbf{\Sigma}_{k|k-1}^{-1} \frac{\partial \mathbf{\Sigma}_{k|k-1}}{\partial \theta_i}\Big] \tag{D.31}$$

$$\frac{\partial \big(\mathbf{e}_k^T \mathbf{\Sigma}_{k|k-1}^{-1} \mathbf{e}_k\big)}{\partial \theta_i} = \frac{\partial \mathbf{e}_k^T}{\partial \theta_i} \mathbf{\Sigma}_{k|k-1}^{-1} \mathbf{e}_k + \mathbf{e}_k^T \frac{\partial \mathbf{\Sigma}_{k|k-1}^{-1}}{\partial \theta_i} \mathbf{e}_k + \mathbf{e}_k^T \mathbf{\Sigma}_{k|k-1}^{-1} \frac{\partial \mathbf{e}_k}{\partial \theta_i} \tag{D.32}$$

$$\frac{\partial l_k}{\partial \theta_i} = -\frac{1}{2} tr\Big[\mathbf{\Sigma}_{k|k-1}^{-1} \frac{\partial \mathbf{\Sigma}_{k|k-1}}{\partial \theta_i}\Big] - \frac{1}{2}\Big[\frac{\partial \mathbf{e}_k^T}{\partial \theta_i} \mathbf{\Sigma}_{k|k-1}^{-1} \mathbf{e}_k + \mathbf{e}_k^T \frac{\partial \mathbf{\Sigma}_{k|k-1}^{-1}}{\partial \theta_i} \mathbf{e}_k + \mathbf{e}_k^T \mathbf{\Sigma}_{k|k-1}^{-1} \frac{\partial \mathbf{e}_k}{\partial \theta_i}\Big] \tag{D.33}$$

$$\frac{\partial l_k}{\partial \theta_i} = -\frac{1}{2} tr\Big\{\Big[\mathbf{\Sigma}_{k|k-1}^{-1} \frac{\partial \mathbf{\Sigma}_{k|k-1}}{\partial \theta_i}\Big]\Big[\mathbf{I} - \mathbf{\Sigma}_{k|k-1}^{-1} \mathbf{e}_k \mathbf{e}_k^T\Big]\Big\} - \frac{\partial \mathbf{e}_k^T}{\partial \theta_i} \mathbf{\Sigma}_{k|k-1}^{-1} \mathbf{e}_k \tag{D.34}$$

$$\frac{\partial \log L}{\partial \theta_i} = -\sum_{k=1}^{N} \left\{ \frac{1}{2} tr\Big\{\Big[\mathbf{\Sigma}_{k|k-1}^{-1} \frac{\partial \mathbf{\Sigma}_{k|k-1}}{\partial \theta_i}\Big]\Big[\mathbf{I} - \mathbf{\Sigma}_{k|k-1}^{-1} \mathbf{e}_k \mathbf{e}_k^T\Big]\Big\} + \frac{\partial \mathbf{e}_k^T}{\partial \theta_i} \mathbf{\Sigma}_{k|k-1}^{-1} \mathbf{e}_k \right\} \tag{D.35}$$

for $i \in \{1, J\}$, where J is the number of hyperparamètres.

Now, we must estimate $\frac{\partial \mathbf{e}_k}{\partial \theta_i}$ et $\frac{\partial \mathbf{\Sigma}_{k|k-1}}{\partial \theta_i}$.

D.4 Derivative recursions

D.4.1 Estimation of $\frac{\partial \mathbf{e}_k}{\partial \theta_i}$

We have :

$$\mathbf{e}_k = \mathbf{y}_k - \mathbf{D}_k \mathbf{u}_k - \mathbf{C}_k \widehat{\mathbf{x}}_{k|k-1} \tag{D.36}$$

then:

$$\frac{\partial \mathbf{e}_k}{\partial \theta_i} = -\frac{\partial \mathbf{D}_k}{\partial \theta_i} u_k - \frac{\partial \mathbf{C}_k}{\partial \theta_i} \widehat{x}_{k|k-1} - \mathbf{C}_k \frac{\partial \widehat{\mathbf{x}}_{k|k-1}}{\partial \theta_i} \tag{D.37}$$

We have to estimate $\frac{\partial \widehat{\mathbf{x}}_{k|k-1}}{\partial \theta_i}$ from Kalman filter equations.

D.4.2 Estimation of $\frac{\partial \mathbf{\Sigma}_{k|k-1}}{\partial \theta_i}$

We have :

$$\mathbf{\Sigma}_{k|k-1} = \mathbf{C}_k \mathbf{P}_{k|k-1} \mathbf{C}_k^T + \mathbf{R}_k \tag{D.38}$$

then:

$$\frac{\partial \mathbf{\Sigma}_{k|k-1}}{\partial \theta_i} = \frac{\partial \mathbf{C}_k}{\partial \theta_i} \mathbf{P}_{k|k-1} \mathbf{C}_k^T + \mathbf{C}_k \frac{\partial \mathbf{P}_{k|k-1}}{\partial \theta_i} \mathbf{C}_k^T + \mathbf{C}_k \mathbf{P}_{k|k-1} \frac{\partial \mathbf{C}_k^T}{\partial \theta_i} + \frac{\partial \mathbf{R}_k}{\partial \theta_i} \tag{D.39}$$

We have to estimate $\frac{\partial \mathbf{P}_{k|k-1}}{\partial \theta_i}$ from Kalman filter equations.

D.4.3 Derivative recursions for the Prediction step

We have :

$$\widehat{\mathbf{x}}_{k|k-1} = \mathbf{A}_{k-1}\widehat{\mathbf{x}}_{k-1|k-1} + \mathbf{B}_{k-1}\mathbf{u}_{k-1} \tag{D.40}$$

$$\mathbf{P}_{k|k-1} = \mathbf{A}_{k-1}\mathbf{P}_{k-1|k-1}\mathbf{A}_{k-1}^T + \mathbf{G}_{k-1}\mathbf{Q}_{k-1}\mathbf{G}_{k-1}^T \tag{D.41}$$

then:

$$\frac{\partial \widehat{\mathbf{x}}_{k|k-1}}{\partial \theta_i} = \frac{\partial \mathbf{A}_{k-1}}{\partial \theta_i}\widehat{\mathbf{x}}_{k-1|k-1} + \mathbf{A}_{k-1}\frac{\partial \widehat{\mathbf{x}}_{k-1|k-1}}{\partial \theta_i} + \frac{\partial \mathbf{B}_{k-1}}{\partial \theta_i}\mathbf{u}_{k-1} \tag{D.42}$$

and:

$$\begin{aligned}\frac{\partial \mathbf{P}_{k|k-1}}{\partial \theta_i} = &\frac{\partial \mathbf{A}_{k-1}}{\partial \theta_i}\mathbf{P}_{k-1|k-1}\mathbf{A}_{k-1}^T + \mathbf{A}_{k-1}\frac{\partial \mathbf{P}_{k-1|k-1}}{\partial \theta_i}\mathbf{A}_{k-1}^T + \mathbf{A}_{k-1}\mathbf{P}_{k-1|k-1}\frac{\partial \mathbf{A}_{k-1}^T}{\partial \theta_i} \\ &+ \frac{\partial \mathbf{G}_{k-1}}{\partial \theta_i}\mathbf{Q}_{k-1}\mathbf{G}_{k-1}^T + \mathbf{G}_{k-1}\frac{\partial \mathbf{Q}_{k-1}}{\partial \theta_i}\mathbf{G}_{k-1}^T + \mathbf{G}_{k-1}\mathbf{Q}_{k-1}\frac{\partial \mathbf{G}_{k-1}^T}{\partial \theta_i}\end{aligned} \tag{D.43}$$

We must get analytical expressions for $\frac{\partial \widehat{\mathbf{x}}_{k-1|k-1}}{\partial \theta_i}$ and also for $\frac{\partial \mathbf{P}_{k-1|k-1}}{\partial \theta_i}$ from "update" equations of the Kalman filter.

D.4.4 Derivative recursions for the the Update step

We have :

$$\boldsymbol{\Sigma}_{k|k-1} = \mathbf{C}_k\mathbf{P}_{k|k-1}\mathbf{C}_k^T + \mathbf{R}_k \tag{D.44}$$

$$\mathbf{K}_k = \mathbf{P}_{k|k-1}\mathbf{C}_k^T\boldsymbol{\Sigma}_{k|k-1}^{-1} \tag{D.45}$$

$$\mathbf{e}_k = \mathbf{y}_k - \mathbf{D}_k\mathbf{u}_k - \mathbf{C}_k\widehat{\mathbf{x}}_{k|k-1} \tag{D.46}$$

$$\widehat{\mathbf{x}}_{k|k} = \widehat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k\mathbf{e}_k \tag{D.47}$$

$$\mathbf{P}_{k|k} = \mathbf{P}_{k|k-1} - \mathbf{K}_k\mathbf{C}_k\mathbf{P}_{k|k-1} \tag{D.48}$$

then:

$$\frac{\partial \mathbf{e}_k}{\partial \theta_i} = -\frac{\partial \mathbf{C}_k}{\partial \theta_i}\widehat{\mathbf{x}}_{k|k-1} - \mathbf{C}_k\frac{\partial \widehat{\mathbf{x}}_{k|k-1}}{\partial \theta_i} - \frac{\partial \mathbf{D}_k}{\partial \theta_i}\mathbf{u}_k \tag{D.49}$$

$$\frac{\partial \mathbf{\Sigma}_{k|k-1}}{\partial \theta_i} = \frac{\partial \mathbf{C}_k}{\partial \theta_i}\mathbf{P}_{k|k-1}\mathbf{C}_k^T + \mathbf{C}_k\frac{\partial \mathbf{P}_{k|k-1}}{\partial \theta_i}\mathbf{C}_k^T + \mathbf{C}_k\mathbf{P}_{k|k-1}\frac{\partial \mathbf{C}_k^T}{\partial \theta_i} + \frac{\partial \mathbf{R}_k}{\partial \theta_i} \tag{D.50}$$

$$\begin{aligned}\frac{\partial \mathbf{K}_k}{\partial \theta_i} = &\frac{\partial \mathbf{P}_{k|k-1}}{\partial \theta_i}\mathbf{C}_k^T\mathbf{\Sigma}_{k|k-1}^{-1} + \mathbf{P}_{k|k-1}\frac{\partial \mathbf{C}_k^T}{\partial \theta_i}\mathbf{\Sigma}_{k|k-1}^{-1} \\ &-\mathbf{P}_{k|k-1}\mathbf{C}_k^T\mathbf{\Sigma}_{k|k-1}^{-1}\frac{\partial \mathbf{\Sigma}_{k|k-1}}{\partial \theta_i}\mathbf{\Sigma}_{k|k-1}^{-1}\end{aligned} \tag{D.51}$$

$$\frac{\partial \widehat{\mathbf{x}}_{k|k}}{\partial \theta_i} = \frac{\partial \widehat{\mathbf{x}}_{k|k-1}}{\partial \theta_i} + \frac{\partial \mathbf{K}_k}{\partial \theta_i}\mathbf{e}_k + \mathbf{K}_k\frac{\partial \mathbf{e}_k}{\partial \theta_i} \tag{D.52}$$

$$\frac{\partial \mathbf{P}_{k|k}}{\partial \theta_i} = \frac{\partial \mathbf{P}_{k|k-1}}{\partial \theta_i} - \frac{\partial \mathbf{K}_k}{\partial \theta_i}\mathbf{C}_k\mathbf{P}_{k|k-1} - \mathbf{K}_k\frac{\partial \mathbf{C}_k}{\partial \theta_i}\mathbf{P}_{k|k-1} - \mathbf{K}_k\mathbf{C}_k\frac{\partial \mathbf{P}_{k|k-1}}{\partial \theta_i} \tag{D.53}$$

To realize these derivations, we must estimate the partial derivatives $\frac{\partial \widehat{\mathbf{x}}_{k|k-1}}{\partial \theta_i}$ and $\frac{\partial \widehat{\mathbf{x}}_{k|k}}{\partial \theta_i}$ but the functional relations between $\widehat{\mathbf{x}}_{k|k-1}$, $\widehat{\mathbf{x}}_{k|k}$ and θ_i are not explicitly defined.

D.4.5 Direct Estimation of $\widehat{\mathbf{x}}_{k|k}$

We must estimate :

$$\widehat{\mathbf{x}}_{k|k} = \widehat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k\mathbf{e}_k \tag{D.54}$$

with :

$$\widehat{\mathbf{x}}_{k|k-1} = \mathbf{A}_{k-1}\widehat{\mathbf{x}}_{k-1|k-1} + \mathbf{B}_{k-1}\mathbf{u}_{k-1} \tag{D.55}$$

$$\mathbf{P}_{k|k-1} = \mathbf{A}_{k-1}\mathbf{P}_{k-1|k-1}\mathbf{A}_{k-1}^T + \mathbf{G}_{k-1}\mathbf{Q}_{k-1}\mathbf{G}_{k-1}^T \tag{D.56}$$

$$\mathbf{\Sigma}_{k|k-1} = \mathbf{C}_k\mathbf{P}_{k|k-1}\mathbf{C}_k^T + \mathbf{R}_k \tag{D.57}$$

$$\mathbf{K}_k = \mathbf{P}_{k|k-1}\mathbf{C}_k^T\mathbf{\Sigma}_{k|k-1}^{-1} \tag{D.58}$$

$$\mathbf{e}_k = \mathbf{y}_k - \mathbf{D}_k\mathbf{u}_k - \mathbf{C}_k\widehat{\mathbf{x}}_{k|k-1} \tag{D.59}$$

$$\mathbf{P}_{k|k} = \mathbf{P}_{k|k-1} - \mathbf{K}_k\mathbf{C}_k\mathbf{P}_{k|k-1} \tag{D.60}$$

It is equal to solve $F_L(\mathbf{X})$ by the generalized mean square error method.

$$\begin{aligned}F_L(\mathbf{X}) = &[\mathbf{X} - \mathbf{B}_k\mathbf{u}_k - \widehat{\mathbf{x}}_{k|k-1}]^T\mathbf{P}_{k|k-1}^{-1}[\mathbf{X} - \mathbf{B}_k\mathbf{u}_k - \widehat{\mathbf{x}}_{k|k-1}] \\ &+ [\mathbf{y}_k - \mathbf{D}_k\mathbf{u}_k - \mathbf{C}_k\mathbf{X}]^T\mathbf{R}_k^{-1}[\mathbf{y}_k - \mathbf{D}_k\mathbf{u}_k - \mathbf{C}_k\mathbf{X}]\end{aligned} \tag{D.61}$$

let :

$$\widehat{\mathbf{x}}_{k|k} = \arg\min_{\mathbf{X}} F_L(\mathbf{X}) \tag{D.62}$$

thus, $\widehat{\mathbf{x}}_{k|k}$ is solution of:

$$\frac{\partial F_L(\mathbf{X})}{\partial \mathbf{X}}\Big|_{\widehat{\mathbf{x}}_{k|k}} = 0 \tag{D.63}$$

We have :

$$\frac{\partial(\mathbf{x}^T\mathbf{B}\mathbf{x})}{\partial \mathbf{x}} = (\mathbf{B}+\mathbf{B}^T)\mathbf{x} \tag{D.64}$$

$$\frac{\partial(\mathbf{x}-\mathbf{A}\mathbf{s})^T\mathbf{W}(\mathbf{x}-\mathbf{A}\mathbf{s})}{\partial \mathbf{s}} = -2\,\mathbf{A}^T\mathbf{W}(\mathbf{x}-\mathbf{A}\mathbf{s}) \tag{D.65}$$

hence :

$$0 = 2\mathbf{P}_{k|k-1}^{-1}[\mathbf{X}-\mathbf{B}_k\mathbf{u}_k-\widehat{\mathbf{x}}_{k|k-1}] - 2\mathbf{C}_k^T\mathbf{R}_k^{-1}[\mathbf{y}_k-\mathbf{D}_k\mathbf{u}_k-\mathbf{C}_k\mathbf{X}]\Big|_{\mathbf{X}=\widehat{\mathbf{x}}_{k|k}} \tag{D.66}$$

let :

$$\mathbf{M}_k = \mathbf{P}_{k|k-1}^{-1} + \mathbf{C}_k^T\mathbf{R}_k^{-1}\mathbf{C}_k \tag{D.67}$$

$$\mathbf{b}_k = \mathbf{P}_{k|k-1}^{-1}\widehat{\mathbf{x}}_{k|k-1} + \mathbf{C}_k^T\mathbf{R}_k^{-1}[\mathbf{y}_k-\mathbf{D}_k\mathbf{u}_k] \tag{D.68}$$

and we find :

$$\mathbf{M}_k\mathbf{X} = \mathbf{b}_k \tag{D.69}$$

as it is an equation in $\mathbf{X}$, by GLS we get:

$$\widehat{\mathbf{x}}_{k|k} = [\mathbf{M}_k^T\mathbf{P}_{k|k}^{-1}\mathbf{M}_k]^{-1}\mathbf{M}_k^T\mathbf{P}_{k|k}^{-1}\mathbf{b}_k \tag{D.70}$$

Now, we have to estimate $\mathbf{P}_{k|k}$.

D.4.6 Estimation of $\mathbf{P}_{k|k}$

The prediction is given by:

$$\widehat{\mathbf{x}}_{k|k-1} = \mathbf{A}_{k-1}\widehat{\mathbf{x}}_{k-1|k-1} + \mathbf{B}_{k-1}\mathbf{u}_{k-1} \tag{D.71}$$

$$\mathbf{P}_{k|k-1} = \mathbf{A}_{k-1}\mathbf{P}_{k-1|k-1}\mathbf{A}_{k-1}^T + \mathbf{G}_{k-1}\mathbf{Q}_{k-1}\mathbf{G}_{k-1}^T \tag{D.72}$$

The update is given by:

$$\widehat{\mathbf{x}}_{k|k} = \widehat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k\mathbf{e}_k \tag{D.73}$$

$$\mathbf{P}_{k|k} = \mathbf{P}_{k|k-1} - \mathbf{K}_k\mathbf{C}_k\mathbf{P}_{k|k-1} \tag{D.74}$$

We use the Woodburg and Welling lemma for a direct estimation of $\mathbf{P}_{k|k}$:

$$\left[\mathbf{A}+\mathbf{CBC}^T\right]^{-1}=\mathbf{A}^{-1}-\mathbf{A}^{-1}\mathbf{C}\left[\mathbf{B}^{-1}+\mathbf{C}^T\mathbf{A}^{-1}\mathbf{C}\right]^{-1}\mathbf{C}^T\mathbf{A}^{-1} \quad \text{(D.75)}$$

$$\left[\mathbf{P}^{-1}+\mathbf{C}^T\mathbf{R}^{-1}\mathbf{C}\right]^{-1}\mathbf{C}^T\mathbf{R}^{-1}=\mathbf{P}\mathbf{C}^T\left[\mathbf{CPC}^T+\mathbf{R}\right]^{-1} \quad \text{(D.76)}$$

we have :

$$\mathbf{P}_{k|k}=\mathbf{P}_{k|k-1}-\mathbf{K}_k\mathbf{C}_k\mathbf{P}_{k|k-1} \quad \text{(D.77)}$$

$$\mathbf{\Sigma}_{k|k-1}=\mathbf{C}_k\mathbf{P}_{k|k-1}\mathbf{C}_k^T+\mathbf{R}_k \quad \text{(D.78)}$$

$$\mathbf{K}_k=\mathbf{P}_{k|k-1}\mathbf{C}_k^T\mathbf{\Sigma}_{k|k-1}^{-1} \quad \text{(D.79)}$$

$$\mathbf{P}_{k|k}=\mathbf{P}_{k|k-1}-\mathbf{P}_{k|k-1}\mathbf{C}_k^T\left[\mathbf{C}_k\mathbf{P}_{k|k-1}\mathbf{C}_k^T+\mathbf{R}_k\right]^{-1}\mathbf{C}_k\mathbf{P}_{k|k-1} \quad \text{(D.80)}$$

hence:

$$\mathbf{P}_{k|k}=\left[\mathbf{P}_{k|k-1}^{-1}+\mathbf{C}_k^T\mathbf{R}_k^{-1}\mathbf{C}_k\right]^{-1} \quad \text{(D.81)}$$

D.4.7 Estimation of $\frac{\partial\widehat{\mathbf{x}}_{k|k}}{\partial\theta_i}$

$\widehat{\mathbf{x}}_{k|k}$ is the value of $\mathbf{X}$ which minimize the function $F_L(\mathbf{X})$, therefore, we find :

$$0=\mathbf{P}_{\mathbf{k}|k-1}^{-1}\left[\widehat{\mathbf{x}}_{k|k}-\widehat{\mathbf{x}}_{k|k-1}\right]-\mathbf{P}_{k|k-1}^{-1}\mathbf{B}_k\mathbf{u}_k-\mathbf{C}_k^T\mathbf{R}_k^{-1}\left[\mathbf{y}_k-\mathbf{D}_k\mathbf{u}_k-\mathbf{C}_k\widehat{\mathbf{x}}_{k|k}\right] \quad \text{(D.82)}$$

which is the value for every θ_i, then, we can take the differentiating of both members of the equation and solving to find $\frac{\partial\widehat{\mathbf{x}}_{k|k}}{\partial\theta_i}$.

We have :

$$\begin{aligned}0=&\frac{\partial\mathbf{P}_{k|k-1}^{-1}}{\partial\theta_i}\left[\widehat{\mathbf{x}}_{k|k}-\widehat{\mathbf{x}}_{k|k-1}\right]+\mathbf{P}_{k|k-1}^{-1}\left[\frac{\partial\widehat{\mathbf{x}}_{k|k}}{\partial\theta_i}-\frac{\partial\widehat{\mathbf{x}}_{k|k-1}}{\partial\theta_i}\right]\\&-\frac{\partial\mathbf{P}_{k|k-1}^{-1}}{\partial\theta_i}\mathbf{B}_k\mathbf{u}_k-\mathbf{P}_{k|k-1}^{-1}\frac{\partial\mathbf{B}_k}{\partial\theta_i}\mathbf{u}_k\\&-\frac{\partial\mathbf{C}_k^T}{\partial\theta_i}\mathbf{R}_k^{-1}\left[\mathbf{y}_k-\mathbf{D}_k\mathbf{u}_k-\mathbf{C}_k\widehat{\mathbf{x}}_{k|k}\right]-\mathbf{C}_k^T\frac{\partial\mathbf{R}_k^{-1}}{\partial\theta_i}\left[\mathbf{y}_k-\mathbf{D}_k\mathbf{u}_k-\mathbf{C}_k\widehat{\mathbf{x}}_{k|k}\right]\\&+\mathbf{C}_k^T\mathbf{R}_k^{-1}\frac{\partial\mathbf{D}_k}{\partial\theta_i}\mathbf{u}_k+\mathbf{C}_k^T\mathbf{R}_k^{-1}\frac{\partial\mathbf{C}_k}{\partial\theta_i}\widehat{\mathbf{x}}_{k|k}+\mathbf{C}_k^T\mathbf{R}_k^{-1}\mathbf{C}_k\frac{\partial\widehat{\mathbf{x}}_{k|k}}{\partial\theta_i}\end{aligned} \quad \text{(D.83)}$$

using :

$$\mathbf{M}(\widehat{\mathbf{x}}_{k|k},\theta_i) = \mathbf{P}_{k|k-1}^{-1} + \mathbf{C}_k^T\mathbf{R}_k^{-1}\mathbf{C}_k \tag{D.84}$$

$$\begin{aligned}\mathbf{b}(\widehat{\mathbf{x}}_{k|k},\theta_i) = {} & \mathbf{P}_{k|k-1}^{-1}\frac{\partial\mathbf{P}_{k|k-1}}{\partial\theta_i}\mathbf{P}_{k|k-1}^{-1}\big[\widehat{\mathbf{x}}_{k|k}-\widehat{\mathbf{x}}_{k|k-1}\big] + \mathbf{P}_{k|k-1}^{-1}\frac{\partial\widehat{\mathbf{x}}_{k|k-1}}{\partial\theta_i} \\ & - \mathbf{P}_{k|k-1}^{-1}\frac{\partial\mathbf{P}_{k|k-1}}{\partial\theta_i}\mathbf{P}_{k|k-1}^{-1}\mathbf{B}_k\mathbf{u}_k + \mathbf{P}_{k|k-1}^{-1}\frac{\partial\mathbf{B}_k}{\partial\theta_i}\mathbf{u}_k \\ & + \frac{\partial\mathbf{C}_k^T}{\partial\theta_i}\mathbf{R}_k^{-1}\big[\mathbf{y}_k-\mathbf{D}_k\mathbf{u}_k-\mathbf{C}_k\widehat{x}_{k|k}\big] + \mathbf{C}_k^T\frac{\partial\mathbf{R}_k^{-1}}{\partial\theta_i}\big[\mathbf{y}_k-\mathbf{D}_k\mathbf{u}_k-\mathbf{C}_k\widehat{\mathbf{x}}_{k|k}\big] \\ & - \mathbf{C}_k^T\mathbf{R}_k^{-1}\frac{\partial\mathbf{D}_k}{\partial\theta_i}\mathbf{u}_k - \mathbf{C}_k^T\mathbf{R}_k^{-1}\frac{\partial\mathbf{C}_k}{\partial\theta_i}\widehat{\mathbf{x}}_{k|k}\end{aligned} \tag{D.85}$$

hence :

$$\frac{\partial\widehat{\mathbf{x}}_{k|k}}{\partial\theta_i} = \mathbf{M}^{-1}(\widehat{\mathbf{x}}_{k|k},\theta_i)\mathbf{b}(\widehat{\mathbf{x}}_{k|k},\theta_i) \tag{D.86}$$

D.4.8 Estimation of $\frac{\partial\mathbf{P}_{k|k}}{\partial\boldsymbol{\theta}_i}$

We have :

$$\mathbf{P}_{k|k} = \big[\mathbf{P}_{k|k-1}^{-1} + \mathbf{C}_k^T\mathbf{R}_k^{-1}\mathbf{C}_k\big]^{-1} \tag{D.87}$$

hence :

$$\begin{aligned}\frac{\partial\mathbf{P}_{k|k}}{\partial\theta_i} = {} & -\big[\mathbf{P}_{k|k-1}^{-1} + \mathbf{C}_k^T\mathbf{R}_k^{-1}\mathbf{C}_k\big]^{-1}\Big[-\mathbf{P}_{k|k-1}^{-1}\frac{\partial\mathbf{P}_{k|k-1}}{\partial\theta_i}\mathbf{P}_{k|k-1}^{-1} + \frac{\partial\mathbf{C}_k^T}{\partial\theta_i}\mathbf{R}_k^{-1}\mathbf{C}_k \\ & -\mathbf{C}_k^T\mathbf{R}_k^{-1}\frac{\partial\mathbf{R}_k}{\partial\theta_i}\mathbf{R}_k^{-1}\mathbf{C}_k + \mathbf{C}_k^T\mathbf{R}_k^{-1}\frac{\partial\mathbf{C}_k}{\partial\theta_i}\Big]\big[\mathbf{P}_{k|k-1}^{-1} + \mathbf{C}_k^T\mathbf{R}_k^{-1}\mathbf{C}_k\big]^{-1}\end{aligned} \tag{D.88}$$

D.5 Parameter Optimization

To estimate $\boldsymbol{\theta}$ we choose the Gauss-Newton method, or "Method of Scoring".

Let consider:

$$\boldsymbol{\theta}^{(l+1)} = \boldsymbol{\theta}^{(l)} - \rho_l\ \mathbf{R}^{(l)}\ \mathbf{g}^{(l)} \tag{D.89}$$

where :

- $\boldsymbol{\theta}^{(l)}$ is the hyperparameter vector at iteration (l),

- $\mathbf{g}^{(l)}$ is the gradient vector of the negative log-likelihood,

$$\mathbf{J}(\boldsymbol{\theta}) = -logL(\boldsymbol{\theta}; Y) \tag{D.90}$$

$$\mathbf{g}^{(l)} = \frac{\partial \mathbf{J}}{\partial \boldsymbol{\theta}}\Big|_{\theta^{(l)}} \tag{D.91}$$

$$\mathbf{g}_i^{(l)} = -\frac{1}{2} tr\Big\{ \Big[\boldsymbol{\Sigma}_{k|k-1}^{-1} \frac{\partial \boldsymbol{\Sigma}_{k|k-1}}{\partial \theta_i} \Big] \Big[\mathbf{I} - \boldsymbol{\Sigma}_{k|k-1}^{-1} \mathbf{e}_k \mathbf{e}_k^T \Big] \Big\} - \frac{\partial \mathbf{e}_k^T}{\partial \theta_i} \boldsymbol{\Sigma}_{k|k-1}^{-1} \mathbf{e}_k \tag{D.92}$$

- $\mathbf{R}^{(l)}$ is an approximation to the second partial matrix:

$$\mathbf{R}^{(l)} = \Big[\frac{\partial^2 \mathbf{J}}{\partial \boldsymbol{\theta} \partial \boldsymbol{\theta}^T} \Big]^{-1}\Big|_{\theta^{(l)}} \tag{D.93}$$

In the Gauss-Newton algorithm, $\mathbf{R}^{(l)}$ is choose as the inverse of the Fisher information matrix $\mathbf{M}^{(l)}$, then :

$$\mathbf{M}^{(l)} = \mathbb{E}\Big[(\frac{\partial \mathbf{J}}{\partial \boldsymbol{\theta}}) (\frac{\partial \mathbf{J}}{\partial \boldsymbol{\theta}})^T \Big]\Big|_{\theta^{(l)}} \tag{D.94}$$

which we can estimate from:

$$\begin{aligned} \widehat{\mathbf{M}}^{(l)}(i,j) = \sum_{k=1}^{N} \Big\{ (\frac{\partial \mathbf{e}_k}{\partial \theta_i^{(l)}})^T \boldsymbol{\Sigma}_{k|k-1}^{-1} \frac{\partial \mathbf{e}_k}{\partial \theta_j^{(l)}} + \frac{1}{2} tr\Big[\boldsymbol{\Sigma}_{k|k-1}^{-1} \frac{\partial \boldsymbol{\Sigma}_{k|k-1}}{\partial \theta_i^{(l)}} \boldsymbol{\Sigma}_{k|k-1}^{-1} \frac{\partial \boldsymbol{\Sigma}_{k|k-1}}{\partial \theta_j^{(l)}} \Big] \\ + \frac{1}{4} tr\Big[\boldsymbol{\Sigma}_{k|k-1}^{-1} \frac{\partial \boldsymbol{\Sigma}_{k|k-1}}{\partial \theta_i^{(l)}} \Big] tr\Big[\boldsymbol{\Sigma}_{k|k-1}^{-1} \frac{\partial \boldsymbol{\Sigma}_{k|k-1}}{\partial \theta_j^{(l)}} \Big] \Big\} \end{aligned} \tag{D.95}$$

Estimating $\widehat{\mathbf{M}}$ does not need an estimation of second derivatives $\frac{\partial^2 e_k}{\partial \theta_i^{(l)} \partial \theta_j^{(l)}}$ neither $\frac{\partial^2 \Sigma_{k|k-1}}{\partial \theta_i^{(l)} \partial \theta_j^{(l)}}$.

As $\mathbf{M}^{(l)}$ is a non-negative matrix, we can always find ρ_l such as $\mathbf{J}(\theta^{(l+1)} < \mathbf{J}(\theta^{(l)}$

D.6 Equivalence Kalman - Projection

D.6.1 State Space Model

$$\mathbf{x}_{k+1} = \mathbf{A}_k(\boldsymbol{\theta}) \mathbf{x}_k + \mathbf{v}_k \tag{D.96}$$

$$\mathbf{y}_k = \mathbf{C}_k(\boldsymbol{\theta}) \mathbf{x}_k + \mathbf{n}_k \tag{D.97}$$

with : $\mathbf{v}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}_k(\boldsymbol{\theta}))$, $\mathbf{n}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{R}_k(\boldsymbol{\theta}))$,

D.6.2 Kalman Filter

Prediction

$$\widehat{\mathbf{x}}_{k|k-1} = \mathbf{A}_{k-1}\widehat{\mathbf{x}}_{k-1|k-1} \tag{D.98}$$
$$\mathbf{P}_{k|k-1} = \mathbf{A}_{k-1}\mathbf{P}_{k-1|k-1}\mathbf{A}_{k-1}^T + \mathbf{Q}_{k-1} \tag{D.99}$$

Update

$$\mathbf{\Sigma}_{k|k-1} = \mathbf{C}_k\mathbf{P}_{k|k-1}\mathbf{C}_k^T + \mathbf{R}_k \tag{D.100}$$
$$\mathbf{K}_k = \mathbf{P}_{k|k-1}\mathbf{C}_k^T\mathbf{\Sigma}_{k|k-1}^{-1} \tag{D.101}$$
$$\mathbf{e}_k = \mathbf{y}_k - \mathbf{C}_k\widehat{\mathbf{x}}_{k|k-1} \tag{D.102}$$
$$\widehat{\mathbf{x}}_{k|k} = \widehat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k\mathbf{e}_k \tag{D.103}$$
$$\mathbf{P}_{k|k} = \mathbf{P}_{k|k-1} - \mathbf{K}_k\mathbf{C}_k\mathbf{P}_{k|k-1} \tag{D.104}$$

We find :

$$\widehat{\mathbf{x}}_{k|k} = \widehat{\mathbf{x}}_{k|k-1} + \mathbf{P}_{k|k-1}\mathbf{C}_k^T\Big[\mathbf{C}_k\mathbf{P}_{k|k-1}\mathbf{C}_k^T + \mathbf{R}_k\Big]^{-1}\Big[\mathbf{y}_k - \mathbf{C}_k\widehat{\mathbf{x}}_{k|k-1}\Big] \tag{D.105}$$
$$= \widehat{\mathbf{x}}_{k|k-1} + \mathbf{P}_{k|k-1}\mathbf{C}_k^T\Big[\mathbf{C}_k\mathbf{P}_{k|k-1}\mathbf{C}_k^T + \mathbf{R}_k\Big]^{-1}\Big[\mathbf{y}_k - \widehat{\mathbf{y}}_{k|k-1}\Big] \tag{D.106}$$

D.6.3 Estimation by Linear Regression

Let consider the system :

$$\mathbf{x}_k = \widehat{\mathbf{x}}_{k|k-1} + \widetilde{\mathbf{x}}_{k|k-1} \tag{D.107}$$
$$\mathbf{y}_k = \mathbf{C}_k\mathbf{x}_k + \mathbf{n}_k \tag{D.108}$$

with :

- $\widetilde{\mathbf{x}}_{k|k-1} \sim \mathcal{N}(\mathbf{0}, \mathbf{P}_{k|k-1})$
- $\mathbf{n}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{R}_k)$

We obtain the matrix formulation:

$$\begin{pmatrix}\widehat{\mathbf{x}}_{k|k-1}\\ \mathbf{y}_k\end{pmatrix} = \begin{pmatrix}\mathbf{I}\\ \mathbf{C}_k\end{pmatrix}\mathbf{x}_k + \begin{pmatrix}-\widetilde{\mathbf{x}}_{k|k-1}\\ \mathbf{n}_k\end{pmatrix} \tag{D.109}$$

let :

$$\widetilde{\mathbf{y}} = \widetilde{\mathbf{C}}\mathbf{x}_k + \boldsymbol{\epsilon}_k \tag{D.110}$$

with :

- $\widetilde{\mathbf{y}} \doteq \mathbf{y} - \widehat{\mathbf{y}}$
- $\boldsymbol{\epsilon}_k \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma})$
- $\boldsymbol{\Sigma} = \begin{pmatrix} \mathbf{P}_{k|k-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{R}_k \end{pmatrix}$

by GLS we find :

$$\widehat{\mathbf{x}}_k = \left[\widetilde{\mathbf{C}}^T \boldsymbol{\Sigma}^{-1} \widetilde{\mathbf{C}}\right]^{-1} \widetilde{\mathbf{C}}^T \boldsymbol{\Sigma}^{-1} \widetilde{\mathbf{y}} \tag{D.111}$$

let :

$$\widehat{\mathbf{x}}_k = \left(\begin{pmatrix} \mathbf{I} & \mathbf{C}_k^T \end{pmatrix} \begin{pmatrix} \mathbf{P}_{k|k-1}^{-1} & 0 \\ 0 & \mathbf{R}_k^{-1} \end{pmatrix} \begin{pmatrix} \mathbf{I} \\ \mathbf{C}_k \end{pmatrix} \right)^{-1} \begin{pmatrix} \mathbf{I} & \mathbf{C}_k^T \end{pmatrix} \begin{pmatrix} \mathbf{P}_{k|k-1}^{-1} & 0 \\ 0 & \mathbf{R}_k^{-1} \end{pmatrix} \begin{pmatrix} \widetilde{\mathbf{x}}_{k|k-1} \\ \mathbf{y}_k \end{pmatrix} \tag{D.112}$$

$$= \left[\mathbf{P}_{k|k-1}^{-1} + \mathbf{C}_k^T \mathbf{R}_k^{-1} \mathbf{C}_k\right]^{-1} \left[\mathbf{P}_{k|k-1}^{-1} \widehat{\mathbf{x}}_{k|k-1} + \mathbf{C}_k^T \mathbf{R}_k^{-1} \mathbf{y}_k\right] \tag{D.113}$$

We use the Woodburg's lemma :

$$\left[\mathbf{A} + \mathbf{C}\mathbf{B}\mathbf{C}^T\right]^{-1} = \mathbf{A}^{-1} - \mathbf{A}^{-1}\mathbf{C}\left[\mathbf{B}^{-1} + \mathbf{C}^T\mathbf{A}^{-1}\mathbf{C}\right]^{-1}\mathbf{C}^T\mathbf{A}^{-1} \tag{D.114}$$

where :

$$\widehat{\mathbf{x}}_k = \left[\mathbf{P}_{k|k-1} - \mathbf{P}_{k|k-1}\mathbf{C}_k^T\mathbf{D}_k^{-1}\mathbf{C}_k\mathbf{P}_{k|k-1}\right]\left[\mathbf{P}_{k|k-1}^{-1}\widehat{\mathbf{x}}_{k|k-1} + \mathbf{C}_k^T\mathbf{R}_k^{-1}\mathbf{y}_k\right] \tag{D.115}$$

with:

$$\mathbf{D}_k = \mathbf{R}_k + \mathbf{C}_k\mathbf{P}_{k|k-1}\mathbf{C}_k^T \tag{D.116}$$

hence :

$$\widehat{\mathbf{x}}_k = \widehat{\mathbf{x}}_{k|k-1} + \mathbf{P}_{k|k-1}\mathbf{C}_k^T\mathbf{D}_k^{-1}\left[\mathbf{y}_k - \mathbf{C}_k\widehat{\mathbf{x}}_{k|k-1}\right] \tag{D.117}$$

$$= \widehat{\mathbf{x}}_{k|k-1} + \mathbf{P}_{k|k-1}\mathbf{C}_k^T\left[\mathbf{R}_k + \mathbf{C}_k\mathbf{P}_{k|k-1}\mathbf{C}_k^T\right]^{-1}\left[\mathbf{y}_k - \widehat{\mathbf{y}}_{k|k-1}\right] \tag{D.118}$$

which is the same as given by Kalman filter.

D.6.4 Estimation of $F_L(\mathbf{X})$

We have:

$$\begin{pmatrix} \widehat{\mathbf{x}}_{k|k-1} \\ \mathbf{y}_k \end{pmatrix} = \begin{pmatrix} \mathbf{I} \\ \mathbf{C}_k \end{pmatrix} \mathbf{x}_k + \begin{pmatrix} -\widetilde{\mathbf{x}}_{k|k-1} \\ \mathbf{n}_k \end{pmatrix} \tag{D.119}$$

let :

$$\widetilde{\mathbf{y}} = \widetilde{\mathbf{C}}\mathbf{x}_k + \boldsymbol{\epsilon}_k \tag{D.120}$$

where :

$$\left(\widetilde{\mathbf{y}} - \widetilde{\mathbf{C}}\mathbf{x}_k\right) \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}) \tag{D.121}$$

$$\boldsymbol{\Sigma} = \begin{pmatrix} \mathbf{P}_{k|k-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{R}_k \end{pmatrix} \tag{D.122}$$

hence :

$$\mathbf{x}_k = \arg\min_{\mathbf{X}} \left(\widetilde{\mathbf{y}} - \widetilde{\mathbf{C}}\mathbf{X}\right)^T \boldsymbol{\Sigma}^{-1} \left(\widetilde{\mathbf{y}} - \widetilde{\mathbf{C}}\mathbf{X}\right) \tag{D.123}$$

hence :

$$F_L(\mathbf{X}) = \left(\widetilde{\mathbf{y}} - \widetilde{\mathbf{C}}\mathbf{X}\right)^T \boldsymbol{\Sigma}^{-1} \left(\widetilde{\mathbf{y}} - \widetilde{\mathbf{C}}\mathbf{X}\right) \tag{D.124}$$

$$\begin{aligned} &= \begin{pmatrix} \widehat{\mathbf{x}}_{k|k-1} - \mathbf{X} \\ \mathbf{y}_k - \mathbf{C}_k\mathbf{X} \end{pmatrix}^T \begin{pmatrix} \mathbf{P}_{k|k-1}^{-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{R}_k^{-1} \end{pmatrix} \begin{pmatrix} \widehat{\mathbf{x}}_{k|k-1} - \mathbf{X} \\ \mathbf{y}_k - \mathbf{C}_k\mathbf{X} \end{pmatrix} \\ &= \left[\widehat{\mathbf{x}}_{k|k-1} - \mathbf{X}\right]^T \mathbf{P}_{k|k-1}^{-1} \left[\widehat{\mathbf{x}}_{k|k-1} - \mathbf{X}\right] + \left[\mathbf{y}_k - \mathbf{C}_k\mathbf{X}\right]^T \mathbf{R}_k^{-1} \left[\mathbf{y}_k - \mathbf{C}_k\mathbf{X}\right] \\ &= \left[\mathbf{X} - \widehat{\mathbf{x}}_{k|k-1}\right]^T \mathbf{P}_{k|k-1}^{-1} \left[\mathbf{X} - \widehat{\mathbf{x}}_{k|k-1}\right] + \left[\mathbf{y}_k - \mathbf{C}_k\mathbf{X}\right]^T \mathbf{R}_k^{-1} \left[\mathbf{y}_k - \mathbf{C}_k\mathbf{X}\right] \end{aligned} \tag{D.125}$$

hence :

$$\widehat{\mathbf{x}}_k = \arg\min_{\mathbf{X}} F_L(\mathbf{X}) \tag{D.126}$$

$F_L(\mathbf{X})$ is a quadratic function, then we $\widehat{\mathbf{x}}_k$ comes from:

$$\frac{\partial F_L(\mathbf{X})}{\partial \mathbf{X}}\Big|_{\mathbf{X}=\widehat{\mathbf{x}}_k} = 0 \tag{D.127}$$

In conclusion:

- in Kalman filter, $\widehat{\mathbf{x}}_{k|k}$ is the conditional expectation of $\mathbf{x}_k$ given $\mathbf{y}_{1:k}$.
- in the expression of $F_L(\mathbf{X})$, $\widehat{\mathbf{x}}_k$ is a linear projection.

E

Parameter Estimation of a non Linear Model by Gauss-Newton.

In this appendix, the parameter estimation of a nonlinear state Space Model is realized by Gauss-Newton optimization. In Section [5.3] this procedure was used for initial parameter estimation, and sketched without derivation. Derivations for these expressions are provided below.

Let consider a non stationary, nonlinear state space model, with an hyperparameter θ. We develop an estimation of this hyperparameter by direct optimization.

E.1 State Space Model

Let consider the state space model :

$$\mathbf{x}_{k+1} = \mathbf{f}_k(\mathbf{x}_k, \boldsymbol{\theta}) + \mathbf{B}_k(\boldsymbol{\theta})\mathbf{u}_k + \mathbf{G}_k(\boldsymbol{\theta})\mathbf{v}_k \tag{E.1}$$

$$\mathbf{y}_k = \mathbf{h}_k(\mathbf{x}_k, \boldsymbol{\theta}) + \mathbf{n}_k \tag{E.2}$$

with nonlinear RBF functions in State and Measurement equations:

$$\mathbf{f}_k(\mathbf{x}_k) = \sum_{i=1}^{I} \lambda_{ik}^{(f)} \exp\big[-\frac{1}{\sigma_{ik}^{(f)}} (\mathbf{x}_k - \mathbf{c}_{ik}^{(f)})^T (\mathbf{x}_k - \mathbf{c}_{ik}^{(f)}) \big] \tag{E.3}$$

$$\mathbf{h}_k(\mathbf{x}_k) = \sum_{j=1}^{J} \lambda_{jk}^{(h)} \exp\big[-\frac{1}{\sigma_{jk}^{(h)}} (\mathbf{x}_k - \mathbf{c}_{jk}^{(h)})^T (\mathbf{x}_k - \mathbf{c}_{jk}^{(h)}) \big]. \tag{E.4}$$

where

- $\mathbf{x}_k \in \mathbb{R}^n$, $\mathbf{y}_k \in \mathbb{R}^q$, $\mathbf{u}_k \in \mathbb{R}^m$, $\mathbf{v}_k \in \mathbb{R}^p$, $\mathbf{n}_k \in \mathbb{R}^q$,
- $\mathbf{B}_k(\boldsymbol{\theta})$ $[n, m]$, $\mathbf{G}_k(\boldsymbol{\theta})$ $[n, p]$,

- these system matrices depend on a set of unknown parameters $\boldsymbol{\theta} \in \mathbb{R}^J$ where J is the number of hyperparameters,
- $\mathbf{f}_k$ is a nonlinear function,
- $\mathbf{h}_k$ is a nonlinear function,
- $\mathbf{x}_k$ is the state at time k,
- $\mathbf{y}_k$ is the observation at time k,
- $\{\mathbf{u}_k\}$ a deterministic input at time k,
- $\{\mathbf{v}_k\}$, $\{\mathbf{n}_k\}$ the system, and measurement white noises, with $\mathbf{v}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}(\boldsymbol{\theta}))$, $\mathbf{n}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{R}(\boldsymbol{\theta}))$,
- $\mathbf{x}_0 \sim \mathcal{N}(\overline{\mathbf{x}_0}, \mathbf{P}_0)$,
- I is the number of neurons in the hidden layer for function $\mathbf{f}_k$,
- $\mathbf{c}_{ik}^{(f)}$ is the centroid vector for neuron i,
- $\lambda_{ik}^{(f)}$ are the weights of the linear output neuron,
- $\sigma_{ik}^{(f)}$ are the variances of clusters,
- J is the number of neurons in the hidden layer for function $\mathbf{h}_k$,
- $\mathbf{c}_{jk}^{(h)}$ is the centroid vector for neuron j,
- $\lambda_{jk}^{(h)}$ are the weights of the linear output neuron,
- $\sigma_{jk}^{(h)}$ are the variances of clusters.

We realize the linearization of the model as follow:

$$\mathbf{f}_k(\mathbf{x}_k) \simeq \mathbf{f}_k(\widehat{\mathbf{x}}_k) + \mathbf{A}_k\big[\mathbf{x}_k - \widehat{\mathbf{x}}_k\big] \tag{E.5}$$

$$\mathbf{A}_k = \Big[\frac{\partial \mathbf{f}_k(\mathbf{x}_k)}{\partial \mathbf{x}_k}\Big]_{\widehat{\mathbf{x}}_k} \tag{E.6}$$

$$= -2\sum_{i=1}^{I} \frac{\lambda_{ik}^{(f)}}{\sigma_{ik}^{(f)}}[\widehat{\mathbf{x}}_k - \mathbf{c}_{ik}^{(f)}] \exp\Big[-\frac{1}{\sigma_{ik}^{(f)}}(\widehat{\mathbf{x}}_k - \mathbf{c}_{ik}^{(f)})(\widehat{\mathbf{x}}_k - \mathbf{c}_{ik}^{(f)})^T\Big] \tag{E.7}$$

$$\mathbf{h}_k(\mathbf{x}_k) \simeq \mathbf{h}_k(\widehat{\mathbf{x}}_k) + \mathbf{C}_k\big[\mathbf{x}_k - \widehat{\mathbf{x}}_k\big] \tag{E.8}$$

$$\mathbf{C}_k = \Big[\frac{\partial \mathbf{h}_k(\mathbf{x}_k)}{\partial \mathbf{x}_k}\Big]_{\widehat{\mathbf{x}}_k} \tag{E.9}$$

$$= -2\sum_{j=1}^{J} \frac{\lambda_{jk}^{(h)}}{\sigma_{jk}^{(h)}}[\widehat{\mathbf{x}}_k - \mathbf{c}_{jk}^{(h)}] \exp\Big[-\frac{1}{\sigma_{jk}^{(h)}}(\widehat{\mathbf{x}}_k - \mathbf{c}_{jk}^{(h)})(\widehat{\mathbf{x}}_k - \mathbf{c}_{jk}^{(h)})^T\Big]. \tag{E.10}$$

We get the linearized model:

$$\mathbf{x}_k = \mathbf{A}_k\mathbf{x}_{k-1} + \mathbf{B}_k\mathbf{u}_k + \mathbf{v}_k + \mathbf{a}_k \tag{E.11}$$

$$\mathbf{y}_k = \mathbf{C}_k\mathbf{x}_k + \mathbf{n}_k + \mathbf{c}_k, \tag{E.12}$$

with "residues" of linearization:

$$\mathbf{a}_k = \mathbf{f}_k(\widehat{\mathbf{x}}_{k|k}) - \mathbf{A}_k\widehat{\mathbf{x}}_{k|k} \tag{E.13}$$
$$\mathbf{c}_k = \mathbf{h}_k(\widehat{\mathbf{x}}_{k|k-1}) - \mathbf{C}_k\widehat{\mathbf{x}}_{k|k-1}. \tag{E.14}$$

If Δt small, we have small "residues" of linearization, and states of nonlinear and linearized models are similar.

E.2 Extended Kalman filter

The Extended Kalman filter will be used in the derivatives of the likelihood in next sections. It is recalled here to make easier the understanding of this procedure.

Filtering for $k = 1, 2, \cdots$ N, (Forwards)

Prediction

$$\mathbf{f}_k(\mathbf{x}_k) = \sum_{i=1}^{I} \lambda_{ik}^{(f)} \exp\big[-\frac{1}{\sigma_{ik}^{(f)}}(\mathbf{x}_k - \mathbf{c}_{ik}^{(f)})^T(\mathbf{x}_k - \mathbf{c}_{ik}^{(f)})\big] \tag{E.15}$$
$$\widehat{\mathbf{x}}_{k|k-1} = \mathbf{f}_{k-1}(\widehat{\mathbf{x}}_{k-1|k-1}) \tag{E.16}$$
$$\mathbf{A}_{k-1} = \Big[\frac{\partial \mathbf{f}_k(\mathbf{x}_k)}{\partial \mathbf{x}_k}\Big]_{\widehat{\mathbf{x}}_{k-1|k-1}} \tag{E.17}$$
$$\mathbf{P}_{k|k-1} = \mathbf{A}_{k-1}\mathbf{P}_{k-1|k-1}\mathbf{A}_{k-1}^T + \mathbf{Q} \tag{E.18}$$

Update

$$\mathbf{h}_k(\mathbf{x}_k) = \sum_{i=1}^{I} \lambda_{ik}^{(h)} \exp\big[-\frac{1}{\sigma_{ik}^{(h)}}(\mathbf{x}_k - \mathbf{c}_{ik}^{(h)})^T(\mathbf{x}_k - \mathbf{c}_{ik}^{(h)})\big] \tag{E.19}$$
$$\mathbf{C}_k = \Big[\frac{\partial \mathbf{h}_k(\mathbf{x}_k)}{\partial \mathbf{x}_k}\Big]_{\widehat{\mathbf{x}}_{k|k-1}} \tag{E.20}$$
$$\boldsymbol{\Sigma}_{k|k-1} = \mathbf{C}_k\mathbf{P}_{k|k-1}\mathbf{C}_k^T + \mathbf{R} \tag{E.21}$$
$$\mathbf{K}_k = \mathbf{P}_{k|k-1}\mathbf{C}_k^T\boldsymbol{\Sigma}_{k|k-1}^{-1} \tag{E.22}$$
$$\mathbf{e}_k = \mathbf{y}_k - \mathbf{h}_k(\widehat{\mathbf{x}}_{k|k-1}) \tag{E.23}$$
$$\widehat{\mathbf{x}}_{k|k} = \widehat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k\mathbf{e}_k \tag{E.24}$$
$$\mathbf{P}_{k|k} = \mathbf{P}_{k|k-1} - \mathbf{K}_k\mathbf{C}_k\mathbf{P}_{k|k-1} \tag{E.25}$$

E.3 Likelihood

The likelihood is given by :

$$\log L = \sum_{k=1}^{N} \{ -\frac{N_k}{2}\log(2\pi) - \frac{1}{2}\log \mid \mathbf{\Sigma}_{k|k-1} \mid -\frac{1}{2}\sum_{k=1}^{N}(\mathbf{e}_k^T \mathbf{\Sigma}_{k|k-1}^{-1}\mathbf{e}_k)\} \tag{E.26}$$

The k-th component of the log-likelihood is given by :

$$l_k = -\frac{N_k}{2}\log(2\pi) - \frac{1}{2}\log \mid \mathbf{\Sigma}_{k|k-1} \mid -\frac{1}{2}(\mathbf{e}_k^T \mathbf{\Sigma}_{k|k-1}^{-1}\mathbf{e}_k) \tag{E.27}$$

E.4 Computation of the Likelihood function

We have :

$$l_k = -\frac{N_k}{2}\log(2\pi) - \frac{1}{2}\log \mid \mathbf{\Sigma}_{k|k-1} \mid -\frac{1}{2}(\mathbf{e}_k^T \mathbf{\Sigma}_{k|k-1}^{-1}\mathbf{e}_k) \tag{E.28}$$

by Woodbury's lemma, we get:

$$\mathbf{\Sigma}_{k|k-1} = \mathbf{C}_k \mathbf{P}_{k|k-1}\mathbf{C}_k^T + \mathbf{R}_k \tag{E.29}$$

$$\mathbf{\Sigma}_{k|k-1}^{-1} = \mathbf{R}_k^{-1} - \mathbf{R}_k^{-1}\mathbf{C}_k\Big[\mathbf{P}_{k|k-1}^{-1} + \mathbf{C}_k^T\mathbf{R}_k^{-1}\mathbf{C}_k\Big]^{-1}\mathbf{C}_k^T\mathbf{R}_k^{-1} \tag{E.30}$$

hence :

$$\mathbf{e}_k^T\mathbf{\Sigma}_{k|k-1}^{-1}\mathbf{e}_k = tr\big[\mathbf{R}_k^{-1}\mathbf{e}_k\mathbf{e}_k^T\big] - \big[\mathbf{C}_k^T\mathbf{R}_k^{-1}\mathbf{e}_k\big]^T\big[\mathbf{P}_{k|k-1}^{-1} + \mathbf{C}_k^T\mathbf{R}_k^{-1}\mathbf{C}_k\big]^{-1}\big[\mathbf{C}_k^T\mathbf{R}_k^{-1}\mathbf{e}_k\big] \tag{E.31}$$

hence the k-th component of the log-likelihood is given by:

$$\begin{aligned} l_k = &-\frac{N_k}{2}\log(2\pi) - \frac{1}{2}\log|\mathbf{\Sigma}_{k|k-1}| - \frac{1}{2}tr\big[\mathbf{R}_k^{-1}\mathbf{e}_k\mathbf{e}_k^T\big] \\ &+\frac{1}{2}\big[\mathbf{C}_k^T\mathbf{R}_k^{-1}\mathbf{e}_k\big]^T\big[\mathbf{P}_{k|k-1}^{-1} + \mathbf{C}_k^T\mathbf{R}_k^{-1}\mathbf{C}_k\big]^{-1}\big[\mathbf{C}_k^T\mathbf{R}_k^{-1}\mathbf{e}_k\big] \end{aligned} \tag{E.32}$$

E.5 Derivatives computations

E.5.1 Derivatives of the likelihood function

Let consider a symmetric matrix $\mathbf{A}$ and a scalar x , then we have:

$$\frac{\partial|\mathbf{A}|}{\partial x} = \big|\mathbf{A}\big| tr\big[\mathbf{A}^{-1}\frac{\partial \mathbf{A}}{\partial x}\big] \tag{E.33}$$

$$\frac{\partial \mathbf{A}^{-1}}{\partial x} = -\mathbf{A}^{-1}\frac{\partial \mathbf{A}}{\partial x}\mathbf{A}^{-1} \tag{E.34}$$

the derivation of the k-th component of the log-likelihood is given by:

$$\frac{\partial l_k}{\partial \theta_i} = -\frac{1}{2}\frac{\partial \log \mid \mathbf{\Sigma}_{k|k-1} \mid}{\partial \theta_i} - \frac{1}{2}\frac{\partial\big(\mathbf{e}_k^T \mathbf{\Sigma}_{k|k-1}^{-1}\mathbf{e}_k\big)}{\partial \theta_i} \tag{E.35}$$

for $i = 1, 2, \cdots, J$.

let :

$$\frac{\partial \log \mid \mathbf{\Sigma}_{k|k-1} \mid}{\partial \theta_i} = \mid \mathbf{\Sigma}_{k|k-1} \mid^{-1} \frac{\partial \mid \mathbf{\Sigma}_{k|k-1} \mid}{\partial \theta_i} = tr\Big[\mathbf{\Sigma}_{k|k-1}^{-1}\frac{\partial \mathbf{\Sigma}_{k|k-1}}{\partial \theta_i}\Big] \tag{E.36}$$

$$\frac{\partial\big(\mathbf{e}_k^T \mathbf{\Sigma}_{k|k-1}^{-1}\mathbf{e}_k\big)}{\partial \theta_i} = \frac{\partial \mathbf{e}_k^T}{\partial \theta_i}\mathbf{\Sigma}_{k|k-1}^{-1}\mathbf{e}_k + \mathbf{e}_k^T\frac{\partial \mathbf{\Sigma}_{k|k-1}^{-1}}{\partial \theta_i}\mathbf{e}_k + \mathbf{e}_k^T\mathbf{\Sigma}_{k|k-1}^{-1}\frac{\partial \mathbf{e}_k}{\partial \theta_i} \tag{E.37}$$

$$\frac{\partial l_k}{\partial \theta_i} = -\frac{1}{2}tr\Big[\mathbf{\Sigma}_{k|k-1}^{-1}\frac{\partial \mathbf{\Sigma}_{k|k-1}}{\partial \theta_i}\Big] - \frac{1}{2}\Big[\frac{\partial \mathbf{e}_k^T}{\partial \theta_i}\mathbf{\Sigma}_{k|k-1}^{-1}\mathbf{e}_k + \mathbf{e}_k^T\frac{\partial \mathbf{\Sigma}_{k|k-1}^{-1}}{\partial \theta_i}\mathbf{e}_k + \mathbf{e}_k^T\mathbf{\Sigma}_{k|k-1}^{-1}\frac{\partial \mathbf{e}_k}{\partial \theta_i}\Big] \tag{E.38}$$

$$\frac{\partial l_k}{\partial \theta_i} = -\frac{1}{2}tr\Big\{\Big[\mathbf{\Sigma}_{k|k-1}^{-1}\frac{\partial \mathbf{\Sigma}_{k|k-1}}{\partial \theta_i}\Big]\Big[\mathbf{I} - \mathbf{\Sigma}_{k|k-1}^{-1}\mathbf{e}_k\mathbf{e}_k^T\Big]\Big\} - \frac{\partial \mathbf{e}_k^T}{\partial \theta_i}\mathbf{\Sigma}_{k|k-1}^{-1}\mathbf{e}_k \tag{E.39}$$

$$\frac{\partial \log L}{\partial \theta_i} = -\sum_{k=1}^{N}\left\{\frac{1}{2}tr\Big\{\Big[\mathbf{\Sigma}_{k|k-1}^{-1}\frac{\partial \mathbf{\Sigma}_{k|k-1}}{\partial \theta_i}\Big]\Big[\mathbf{I} - \mathbf{\Sigma}_{k|k-1}^{-1}\mathbf{e}_k\mathbf{e}_k^T\Big]\Big\} + \frac{\partial \mathbf{e}_k^T}{\partial \theta_i}\mathbf{\Sigma}_{k|k-1}^{-1}\mathbf{e}_k\right\} \tag{E.40}$$

for $i \in \{1, J\}$, where J is the number of hyperparameters.

Now, we have to estimate $\frac{\partial}{\partial \theta_i}\mathbf{e}_k$ and $\frac{\partial}{\partial \theta_i}\mathbf{\Sigma}_{k|k-1}$

E.5.2 Estimation of $\frac{\partial \mathbf{e}_k}{\partial \boldsymbol{\theta}_i}$

We have :

$$\mathbf{e}_k = \mathbf{y}_k - \mathbf{h}_k(\widehat{\mathbf{x}}_{k|k-1}(\boldsymbol{\theta}), \boldsymbol{\theta}) \tag{E.41}$$

hence :

$$\begin{aligned}\frac{\partial \mathbf{e}_k}{\partial \theta_i} &= -\frac{\partial \mathbf{h}_k(\mathbf{X}(\boldsymbol{\theta}), \boldsymbol{\theta})}{\partial \theta_i}\Big|_{\widehat{\mathbf{x}}_{k|k-1}} \\ &= -\frac{\partial \mathbf{h}_k(\mathbf{X}, \boldsymbol{\theta})}{\partial \theta_i}\Big|_{\widehat{\mathbf{x}}_{k|k-1}} - \frac{\partial \mathbf{h}_k(\mathbf{X}(\boldsymbol{\theta}), \boldsymbol{\theta})}{\partial \mathbf{X}^{\mathbf{T}}}\frac{\partial \mathbf{X}(\boldsymbol{\theta})}{\partial \theta_i}\Big|_{\widehat{\mathbf{x}}_{k|k-1}} \\ &= -\frac{\partial \mathbf{h}_k(\mathbf{X}, \boldsymbol{\theta})}{\partial \theta_i}\Big|_{\widehat{\mathbf{x}}_{k|k-1}} - \mathbf{C}_k(\widehat{\mathbf{x}}_{k|k-1})\frac{\partial \mathbf{X}(\boldsymbol{\theta})}{\partial \theta_i}\Big|_{\widehat{\mathbf{x}}_{k|k-1}}\end{aligned} \tag{E.42}$$

Now, we have to estimate $\frac{\partial \widehat{\mathbf{x}}_{k|k-1}}{\partial \theta_i}$ from Kalman filter equations.

E.5.3 Estimation of $\frac{\partial \mathbf{\Sigma}_{k|k-1}}{\partial \boldsymbol{\theta}_i}$

We have :

$$\mathbf{\Sigma}_{k|k-1} = \mathbf{C}_k \mathbf{P}_{k|k-1} \mathbf{C}_k^T + \mathbf{R}_k \tag{E.43}$$

$$\mathbf{C}_k = \Big[\frac{\partial \mathbf{h}_k(\mathbf{x}_k)}{\partial \mathbf{x}_k^T}\Big]_{\widehat{\mathbf{x}}_{k|k-1}} \tag{E.44}$$

hence :

$$\begin{aligned}\frac{\partial \mathbf{\Sigma}_{k|k-1}}{\partial \theta_i} = &\Big[\frac{\partial \mathbf{C}_k(\mathbf{X},\boldsymbol{\theta})}{\partial \theta_i} + \frac{\partial \mathbf{C}_k(\mathbf{X}(\boldsymbol{\theta}),\boldsymbol{\theta})}{\partial \mathbf{X^T}} \frac{\partial \mathbf{X}(\boldsymbol{\theta})}{\partial \theta_i}\Big]_{\widehat{\mathbf{x}}_{k|k-1}} \mathbf{P}_{k|k-1} \mathbf{C}_k^T \\ &+ \mathbf{C}_k(\widehat{\mathbf{x}}_{k|k-1},\boldsymbol{\theta}) \frac{\partial \mathbf{P}_{k|k-1}}{\partial \theta_i} \mathbf{C}_k^T(\widehat{\mathbf{x}}_{k|k-1},\boldsymbol{\theta}) \\ &+ \mathbf{C}_k(\widehat{\mathbf{x}}_{k|k-1},\boldsymbol{\theta}) \mathbf{P}_{k|k-1} \Big[\frac{\partial \mathbf{C}_k(\mathbf{X},\boldsymbol{\theta})}{\partial \theta_i} + \frac{\partial \mathbf{C}_k(\mathbf{X}(\boldsymbol{\theta}),\boldsymbol{\theta})}{\partial \mathbf{X^T}} \frac{\partial \mathbf{X}(\boldsymbol{\theta})}{\partial \theta_i}\Big]^T_{\widehat{\mathbf{x}}_{k|k-1}} + \frac{\partial \mathbf{R}_k}{\partial \theta_i}\end{aligned} \tag{E.45}$$

Now, we have to estimate $\frac{\partial \mathbf{P}_{k|k-1}}{\partial \theta_i}$ from Kalman filter equations.

E.5.4 Estimations of $\frac{\partial \widehat{\mathbf{x}}_{k|k-1}}{\partial \boldsymbol{\theta}_i}$ and $\frac{\partial \mathbf{P}_{k|k-1}}{\partial \boldsymbol{\theta}_i}$

We have :

$$\widehat{\mathbf{x}}_{k|k-1} = \mathbf{A}_{k-1} \widehat{\mathbf{x}}_{k-1|k-1} + \mathbf{B}_{k-1} \mathbf{u}_{k-1} \tag{E.46}$$

$$\mathbf{P}_{k|k-1} = \mathbf{A}_{k-1} \mathbf{P}_{k-1|k-1} \mathbf{A}_{k-1}^T + \mathbf{G}_{k-1} \mathbf{Q}_{k-1} \mathbf{G}_{k-1}^T \tag{E.47}$$

hence :

$$\frac{\partial \widehat{\mathbf{x}}_{k|k-1}}{\partial \theta_i} = \frac{\partial \mathbf{A}_{k-1}}{\partial \theta_i} \widehat{\mathbf{x}}_{k-1|k-1} + \mathbf{A}_{k-1} \frac{\partial \widehat{\mathbf{x}}_{k-1|k-1}}{\partial \theta_i} + \frac{\partial \mathbf{B}_{k-1}}{\partial \theta_i} \mathbf{u}_{k-1} \tag{E.48}$$

$$\begin{aligned}\frac{\partial \mathbf{P}_{k|k-1}}{\partial \theta_i} = &\frac{\partial \mathbf{A}_{k-1}}{\partial \theta_i} \mathbf{P}_{k-1|k-1} \mathbf{A}_{k-1}^T + \mathbf{A}_{k-1} \frac{\partial \mathbf{P}_{k-1|k-1}}{\partial \theta_i} \mathbf{A}_{k-1}^T \\ &+ \mathbf{A}_{k-1} \mathbf{P}_{k-1|k-1} \frac{\partial \mathbf{A}_{k-1}^T}{\partial \theta_i} + \frac{\partial \mathbf{G}_{k-1}}{\partial \theta_i} \mathbf{Q}_{k-1} \mathbf{G}_{k-1}^T \\ &+ \mathbf{G}_{k-1} \frac{\partial \mathbf{Q}_{k-1}}{\partial \theta_i} \mathbf{G}_{k-1}^T + \mathbf{G}_{k-1} \mathbf{Q}_{k-1} \frac{\partial \mathbf{G}_{k-1}^T}{\partial \theta_i}\end{aligned} \tag{E.49}$$

Ne need analytical expressions for $\frac{\partial \widehat{\mathbf{x}}_{k-1|k-1}}{\partial \theta_i}$ and for $\frac{\partial \mathbf{P}_{k-1|k-1}}{\partial \theta_i}$ from "Update" equations of the Kalman filter.

E.5.5 Estimation of $\widehat{\mathbf{x}}_{k|k}$

We have :

$$F_{NL}(\mathbf{X}) = [\mathbf{X} - \widehat{\mathbf{x}}_{k|k-1}]^T \mathbf{P}_{k|k-1}^{-1} [\mathbf{X} - \widehat{\mathbf{x}}_{k|k-1}] + [\mathbf{y}_k - \mathbf{h}_k(\mathbf{X})]^T \mathbf{R}_k^{-1} [\mathbf{y}_k - \mathbf{h}_k(\mathbf{X})] \tag{E.50}$$

let :

$$\widehat{\mathbf{x}}_{k|k} = \arg\min_{\mathbf{X}} F_{NL}(\mathbf{X}) \tag{E.51}$$

Then $\widehat{\mathbf{x}}_{k|k}$ is solution of

$$\frac{\partial F_{NL}(\mathbf{X})}{\partial \mathbf{X}}\Big|_{\widehat{\mathbf{x}}_{k|k}} = 0 \tag{E.52}$$

Equation $F_{NL}(\mathbf{X})$ must be minimized at each time step d k and for every component of the hyperparameter θ_i.

Let $\mathbf{P}_{k|k}$ the mean square error matrix of $\widehat{\mathbf{x}}_{k|k}$, then we find :

$$\mathbf{P}_{k|k} = \left[\mathbf{P}_{k|k-1}^{-1} + \mathbf{C}_k^T \mathbf{R}_k^{-1} \mathbf{C}_k\right]^{-1} \tag{E.53}$$

By Gauss-Newton algorithm, we have the iterative procedure:

$$\widehat{\mathbf{x}}_{k|k}^{(j+1)} = \widehat{\mathbf{x}}_{k|k}^{(j)} - \rho_j \mathbf{R}_j \mathbf{g}_j \tag{E.54}$$

with ρ_j an optimal value to insure the decrease of $F_{NL}(\mathbf{X})$

$$\mathbf{R}_j = \left[\mathbf{P}_{k|k-1}^{-1} + \mathbf{C}_k^T(\widehat{\mathbf{x}}_{k|k}^{(j)}) \mathbf{R}_k^{-1} \mathbf{C}_k(\widehat{\mathbf{x}}_{k|k}^{(j)})\right]^{-1} \tag{E.55}$$

$$\mathbf{g}_j = \mathbf{P}_{k|k}^{-1} \left[\widehat{\mathbf{x}}_{k|k}^{(j)} - \widehat{\mathbf{x}}_{k|k-1}\right] - \mathbf{C}_k^T(\widehat{\mathbf{x}}_{k|k}^{(j)}) \mathbf{R}_k^{-1} \left[\mathbf{y}_k - \mathbf{h}_k(\widehat{\mathbf{x}}_{k|k}^{(j)})\right] \tag{E.56}$$

The procedure starts with former values $\widehat{\mathbf{x}}_{k|k}^{(0)} = \widehat{\mathbf{x}}_{k|k-1}$ coming from prediction step.

E.5.6 Estimation of $\frac{\partial \widehat{\mathbf{x}}_{k|k}}{\partial \theta_i}$

We cannot estimate $\frac{\partial \widehat{\mathbf{x}}_{k|k}}{\partial \theta_i}$ because we do not know the functional relation between $\widehat{\mathbf{x}}_{k|k}$ and θ from Kalman filter.

To cope with this problem, we can represent $\widehat{\mathbf{x}}_{k|k}$ as a value that minimizes the function:

$$F_{NL}(\mathbf{X}) = [\mathbf{X} - \widehat{\mathbf{x}}_{k|k-1}]^T \mathbf{P}_{k|k-1}^{-1} [\mathbf{X} - \widehat{\mathbf{x}}_{k|k-1}] + [\mathbf{y}_k - \mathbf{h}_k(\mathbf{X})]^T \mathbf{R}_k^{-1} [\mathbf{y}_k - \mathbf{h}_k(\mathbf{X})] \tag{E.57}$$

Let :

$$\widehat{\mathbf{x}}_{k|k} = \arg\min_{\mathbf{X}} F_{NL}(\mathbf{X}); \tag{E.58}$$

hence $\widehat{\mathbf{x}}_{k|k}$ is the root of:

$$\frac{\partial F_{NL}(\mathbf{X})}{\partial \mathbf{X}} = 0, \tag{E.59}$$

Let :

$$0 = \mathbf{P}_{k|k-1}^{-1} \big[\mathbf{X} - \widehat{\mathbf{x}}_{k|k-1}\big]\Big|_{X=\widehat{\mathbf{x}}_{k|k}} - \frac{\partial \mathbf{h}_k^T(\mathbf{X})}{\partial \mathbf{X}} \mathbf{R}_k^{-1} \big[\mathbf{y}_k - \mathbf{h}_k(\mathbf{X})\big]\Big|_{X=\widehat{\mathbf{x}}_{k|k}} \tag{E.60}$$

Thus θ_i is solution of:

$$0 = \frac{\partial \Big\{ \mathbf{P}_{k|k-1}^{-1} \big[\mathbf{X} - \widehat{\mathbf{x}}_{k|k-1}\big] \Big\}}{\partial \theta_i}\Big|_{\mathbf{X}=\widehat{\mathbf{x}}_{k|k}} - \frac{\partial \Big\{ \frac{\partial \mathbf{h}_k^T(\mathbf{X})}{\partial \mathbf{X}} \mathbf{R}_k^{-1} \big[\mathbf{y}_k - \mathbf{h}_k(\mathbf{X})\big] \Big\}}{\partial \theta_i}\Big|_{\mathbf{X}=\widehat{\mathbf{x}}_{k|k}} \tag{E.61}$$

We must take the derivatives of

- the vector $\mathbf{h}_k(\widehat{\mathbf{x}}_{k|k}, \boldsymbol{\theta})$
- the matrix $\mathbf{C}_k(\widehat{\mathbf{x}}_{k|k}, \boldsymbol{\theta}) = \frac{\partial \mathbf{h}_k(\mathbf{X}, \boldsymbol{\theta})}{\partial \mathbf{X}}\big|_{\widehat{\mathbf{x}}_{k|k-1}}$

which depend on $\boldsymbol{\theta}$ thru:

- a direct relationship in $\boldsymbol{\theta}$ for $\mathbf{h}_k(\mathbf{X}, \boldsymbol{\theta})$ and $\mathbf{C}_k(\mathbf{X}, \boldsymbol{\theta})$
- an indirect relationship in $\boldsymbol{\theta}$ thru $\mathbf{X}(\boldsymbol{\theta})$ for $\mathbf{h}_k(\mathbf{X}(\boldsymbol{\theta}), \boldsymbol{\theta})$ and $\mathbf{C}_k(\mathbf{X}(\boldsymbol{\theta}), \boldsymbol{\theta})$

hence we have :

$$\frac{\partial \mathbf{h}_k(\mathbf{X}(\boldsymbol{\theta}), \boldsymbol{\theta})}{\partial \theta_i}\big|_{\widehat{\mathbf{x}}_{k|k}} = \frac{\partial \mathbf{h}_k(\mathbf{X}, \boldsymbol{\theta})}{\partial \theta_i}\big|_{\widehat{\mathbf{x}}_{k|k}} + \frac{\partial \mathbf{h}_k(\mathbf{X}(\boldsymbol{\theta}), \boldsymbol{\theta})}{\partial \mathbf{X}^T} \frac{\partial \mathbf{X}(\boldsymbol{\theta})}{\partial \theta_i}\big|_{\widehat{\mathbf{x}}_{k|k}} \tag{E.62}$$

$$\frac{\partial \mathbf{C}_k(\mathbf{X}(\theta), \theta)}{\partial \theta_i}\big|_{\widehat{\mathbf{x}}_{k|k}} = \frac{\partial \mathbf{C}_k(\mathbf{X}, \boldsymbol{\theta})}{\partial \theta_i}\big|_{\widehat{\mathbf{x}}_{k|k}} + \frac{\partial \mathbf{C}_k(\mathbf{X}(\boldsymbol{\theta}), \boldsymbol{\theta})}{\partial \mathbf{X}^T} \frac{\partial \mathbf{X}(\boldsymbol{\theta})}{\partial \theta_i}\big|_{\widehat{\mathbf{x}}_{k|k}} \tag{E.63}$$

We find :

$$
\begin{aligned}
0 = \quad & - \quad \mathbf{P}_{k|k-1}^{-1} \frac{\partial \mathbf{P}_{k|k-1}}{\partial \theta_i} \mathbf{P}_{k|k-1}^{-1} \Big[\widehat{\mathbf{x}}_{k|k} - \widehat{\mathbf{x}}_{k|k-1} \Big] \\
& + \quad \mathbf{P}_{k|k-1}^{-1} \frac{\partial \widehat{\mathbf{x}}_{k|k}}{\partial \theta_i} - \mathbf{P}_{k|k-1}^{-1} \frac{\partial \widehat{\mathbf{x}}_{k|k-1}}{\partial \theta_i} \\
& - \quad \frac{\partial \mathbf{C}_k^T(\widehat{\mathbf{x}}_{k|k}, \boldsymbol{\theta})}{\partial \theta_i} \mathbf{R}_k^{-1} \Big[\mathbf{y}_k - \mathbf{h}_k(\widehat{\mathbf{x}}_{k|k}) \Big] \\
& + \quad \mathbf{C}_k^T(\widehat{\mathbf{x}}_{k|k}, \boldsymbol{\theta}) \mathbf{R}_k^{-1} \frac{\partial \mathbf{R}_k^{-1}}{\partial \theta_i} \mathbf{R}_k^{-1} \Big[\mathbf{y}_k - \mathbf{h}_k(\widehat{\mathbf{x}}_{k|k}) \Big] \\
& + \quad \mathbf{C}_k^T(\widehat{\mathbf{x}}_{k|k}, \boldsymbol{\theta}) \mathbf{R}_k^{-1} \Big[\frac{\partial \mathbf{h}_k(\mathbf{X}, \boldsymbol{\theta})}{\partial \theta_i} \big|_{\widehat{x}_{k|k}} + \frac{\partial \mathbf{h}_k(\mathbf{X}(\boldsymbol{\theta}), \boldsymbol{\theta})}{\partial \mathbf{X}^T} \frac{\partial \mathbf{X}(\boldsymbol{\theta})}{\partial \theta_i} \big|_{\widehat{x}_{k|k}} \Big]
\end{aligned}
\tag{E.64}
$$

Let :

$$
\mathbf{M}(\widehat{x}_{k|k}, \boldsymbol{\theta}) = \mathbf{P}_{k|k-1}^{-1} + \mathbf{C}_k^T(\widehat{\mathbf{x}}_{k|k}, \boldsymbol{\theta}) \mathbf{R}_k^{-1} \mathbf{C}_k(\widehat{\mathbf{x}}_{k|k}, \boldsymbol{\theta}) \tag{E.65}
$$

$$
\begin{aligned}
\mathbf{b}(\widehat{x}_{k|k}, \boldsymbol{\theta}) = & \mathbf{P}_{k|k-1}^{-1} \frac{\partial \mathbf{P}_{k|k-1}}{\partial \theta_i} \mathbf{P}_{k|k-1}^{-1} \Big[\widehat{\mathbf{x}}_{k|k} - \widehat{\mathbf{x}}_{k|k-1} \Big] + \mathbf{P}_{k|k-1}^{-1} \frac{\partial \widehat{\mathbf{x}}_{k|k-1}}{\partial \theta_i} \\
& + \quad \frac{\partial \mathbf{C}_k^T(\widehat{\mathbf{x}}_{k|k}, \boldsymbol{\theta})}{\partial \theta_i} \mathbf{R}_k^{-1} \Big[\mathbf{y}_k - \mathbf{h}_k(\widehat{\mathbf{x}}_{k|k}) \Big] \\
& - \quad \mathbf{C}_k^T(\widehat{\mathbf{x}}_{k|k}, \boldsymbol{\theta}) \mathbf{R}_k^{-1} \frac{\partial \mathbf{R}_k^{-1}}{\partial \theta_i} \mathbf{R}_k^{-1} \Big[\mathbf{y}_k - \mathbf{h}_k(\widehat{\mathbf{x}}_{k|k}) \Big] \\
& - \quad \mathbf{C}_k^T(\widehat{\mathbf{x}}_{k|k}, \boldsymbol{\theta}) \mathbf{R}_k^{-1} \frac{\partial \mathbf{h}_k(\mathbf{X}, \boldsymbol{\theta})}{\partial \theta_i} \big|_{\widehat{x}_{k|k}}
\end{aligned}
\tag{E.66}
$$

hence :

$$
\frac{\partial \widehat{\mathbf{x}}_{k|k}}{\partial \theta_i} = \mathbf{M}(\widehat{x}_{k|k}, \boldsymbol{\theta})^{-1} \mathbf{b}(\widehat{x}_{k|k}, \boldsymbol{\theta}) \tag{E.67}
$$

E.5.7 Estimation of $\frac{\partial \mathbf{P}_{k|k}}{\partial \boldsymbol{\theta}_i}$

We have :

$$
\mathbf{P}_{k|k} = \Big[\mathbf{P}_{k|k-1}^{-1} + \mathbf{C}_k^T \mathbf{R}_k^{-1} \mathbf{C}_k \Big]^{-1} \tag{E.68}
$$

hence :

$$\begin{aligned}\frac{\partial \mathbf{P}_{k|k}}{\partial \theta_i} = -\Big[\mathbf{P}_{k|k-1}^{-1} + \mathbf{C}_k^T \mathbf{R}_k^{-1} \mathbf{C}_k\Big]^{-1} \\ \Big\{ \quad - \quad \mathbf{P}_{k|k-1}^{-1} \frac{\partial \mathbf{P}_{k|k-1}}{\partial \theta_i} \mathbf{P}_{k|k-1}^{-1} - \mathbf{C}_k^T \mathbf{R}_k^{-1} \frac{\partial \mathbf{R}_k}{\partial \theta_i} \mathbf{R}_k^{-1} \mathbf{C}_k \\ + \quad \Big[\frac{\partial \mathbf{C}_k^T(\widehat{\mathbf{x}}_{k|k-1}, \boldsymbol{\theta})}{\partial \theta_i} + \frac{\partial \mathbf{C}_k(\mathbf{X}(\boldsymbol{\theta}), \boldsymbol{\theta})}{\partial \mathbf{X}^T} \frac{\partial \mathbf{X}(\boldsymbol{\theta})}{\partial \theta_i}\Big|_{\widehat{x}_{k|k-1}}\Big] \mathbf{R}_k^{-1} \mathbf{C}_k(\widehat{\mathbf{x}}_{k|k-1}, \boldsymbol{\theta}) \\ + \quad \mathbf{C}_k^T(\widehat{\mathbf{x}}_{k|k-1}, \boldsymbol{\theta}) \mathbf{R}_k^{-1} \Big[\frac{\partial \mathbf{C}_k(\widehat{\mathbf{x}}_{k|k-1}, \boldsymbol{\theta})}{\partial \theta_i} + \frac{\partial \mathbf{C}_k(\mathbf{X}(\boldsymbol{\theta}), \boldsymbol{\theta})}{\partial \mathbf{X}^T} \frac{\partial \mathbf{X}(\boldsymbol{\theta})}{\partial \theta_i}\Big|_{\widehat{\mathbf{x}}_{k|k-1}}\Big] \quad \Big\} \\ \Big[\mathbf{P}_{k|k-1}^{-1} + \mathbf{C}_k^T \mathbf{R}_k^{-1} \mathbf{C}_k\Big]^{-1} \end{aligned} \tag{E.69}$$

E.6 Parameter optimization

To estimate θ we choose the Gauss-Newton method, or "Method of Scoring".

Let consider:

$$\boldsymbol{\theta}^{(l+1)} = \boldsymbol{\theta}^{(l)} - \rho_l \, \mathbf{R}^{(l)} \, \mathbf{g}^{(l)} \tag{E.70}$$

where :

- $\boldsymbol{\theta}^{(l)}$ is the hyperparameter vector at iteration (l),
- $\mathbf{g}^{(l)}$ is the gradient vector of the negative log-likelihood,

$$\mathbf{J}(\theta) = -logL(\boldsymbol{\theta}; Y) \tag{E.71}$$

$$\mathbf{g}^{(l)} = \frac{\partial \mathbf{J}}{\partial \boldsymbol{\theta}}\Big|_{\boldsymbol{\theta}^{(l)}} \tag{E.72}$$

$$\mathbf{g}_i^{(l)} = -\frac{1}{2} tr\Big\{\Big[\boldsymbol{\Sigma}_{k|k-1}^{-1} \frac{\partial \boldsymbol{\Sigma}_{k|k-1}}{\partial \theta_i}\Big]\Big[\mathbf{I} - \boldsymbol{\Sigma}_{k|k-1}^{-1} \mathbf{e}_k \mathbf{e}_k^T\Big]\Big\} - \frac{\partial \mathbf{e}_k^T}{\partial \theta_i} \boldsymbol{\Sigma}_{k|k-1}^{-1} \mathbf{e}_k \tag{E.73}$$

- $\mathbf{R}^{(l)}$ is an approximation to the second partial matrix:

$$\mathbf{R}^{(l)} = \Big[\frac{\partial^2 \mathbf{J}}{\partial \boldsymbol{\theta} \partial \boldsymbol{\theta}^T}\Big]^{-1}\Big|_{\theta^{(l)}} \tag{E.74}$$

In the Gauss-Newton algorithm, $\mathbf{R}^{(l)}$ is choose as the inverse of the Fisher information matrix $\mathbf{M}^{(l)}$, then :

$$\mathbf{M}^{(l)} = \mathbb{E}\Big[(\frac{\partial \mathbf{J}}{\partial \boldsymbol{\theta}})(\frac{\partial \mathbf{J}}{\partial \boldsymbol{\theta}})^T\Big]\Big|_{\theta^{(l)}} \tag{E.75}$$

which we can estimate from:

$$\widehat{\mathbf{M}}^{(l)}(i,j) = \sum_{k=1}^{N} \Big\{ \big(\frac{\partial \mathbf{e}_k}{\partial \theta_i^{(l)}}\big)^T \mathbf{\Sigma}_{k|k-1}^{-1} \frac{\partial \mathbf{e}_k}{\partial \theta_j^{(l)}} + \frac{1}{2}\, tr\Big[\mathbf{\Sigma}_{k|k-1}^{-1} \frac{\partial \mathbf{\Sigma}_{k|k-1}}{\partial \theta_i^{(l)}} \mathbf{\Sigma}_{k|k-1}^{-1} \frac{\partial \mathbf{\Sigma}_{k|k-1}}{\partial \theta_j^{(l)}}\Big] + \frac{1}{4}\, tr\Big[\mathbf{\Sigma}_{k|k-1}^{-1} \frac{\partial \mathbf{\Sigma}_{k|k-1}}{\partial \theta_i^{(l)}}\Big] tr\Big[\mathbf{\Sigma}_{k|k-1}^{-1} \frac{\partial \mathbf{\Sigma}_{k|k-1}}{\partial \theta_j^{(l)}}\Big] \Big\} \tag{E.76}$$

Estimating $\widehat{\mathbf{M}}$ does not need an estimation of second derivatives $\frac{\partial^2 e_k}{\partial \theta_i^{(l)} \partial \theta_j^{(l)}}$ neither $\frac{\partial^2 \Sigma_{k|k-1}}{\partial \theta_i^{(l)} \partial \theta_j^{(l)}}$.

As $\mathbf{M}^{(l)}$ is a non-negative matrix, we can always find ρ_l such as $\mathbf{J}(\theta^{(l+1)} < \mathbf{J}(\theta^{(l)}$

F

Parameter Estimation of a Linear Model by EM algorithm.

In this appendix, the parameter estimation of a nonlinear state Space Model is realized by EM algorithm. In Section [5.2] this procedure was used for initial parameter estimation, and sketched without derivation. Derivations for these expressions are provided below.

Let consider a stationary, linear dynamic state space model, with a vector of hyperparameter θ. Estimating of these parameters of a dynamical system from training data actually reduces to the problem of maximum likelihood (ML) identification of a general linear dynamical system. The classical method to obtain parameters of a linear state space system involves the construction of a time-varying Kalman predictor and the expression of the likelihood function in terms of the prediction error.

This identification could be simple if the state of the system were observable. This observation can be combined with the Expectation-Maximization (EM) algorithm to provide a conceptually simple approach to the ML identification of dynamical systems.

F.1 State Space Model

Let consider the stationary linear model:

$$\mathbf{x}_{k+1} = \mathbf{A}(\boldsymbol{\theta})\mathbf{x}_k + \mathbf{v}_k \tag{F.1}$$

$$\mathbf{y}_k = \mathbf{C}(\boldsymbol{\theta})\mathbf{x}_k + \mathbf{n}_k \tag{F.2}$$

where :

- $\mathbf{x}_k \in R^n$, $\mathbf{y}_k \in R^q$, $\mathbf{v}_k \in R^n$, $\mathbf{n}_k \in R^q$,
- $\mathbf{A}(\boldsymbol{\theta})$ $[n, n]$, $\mathbf{C}(\boldsymbol{\theta})$ $[q, n]$,

- these system matrices depend on a set of unknown parameters $\boldsymbol{\theta} \in \mathbb{R}^J$ where J is the number of hyperparameters,
- $\mathbf{x}_k$ is the state at time k,
- $\mathbf{y}_k$ is the observation at time k,
- $\{\mathbf{v}_k\}$, $\{\mathbf{n}_k\}$ system and measurement noises, with $\mathbf{v}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}(\boldsymbol{\theta}))$, $\mathbf{n}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{R}(\boldsymbol{\theta}))$,
- $\mathbf{x}_0 \sim N(\overline{\mathbf{x}_0}, \mathbf{P}_0)$.

Algorithm F.1.1 (Kalman Smoother) .

The Kalman smoother will be used in the derivatives of the likelihood in next sections. It is recalled here to make easier the understanding of the procedure.

Initialisation

$$\widehat{\mathbf{x}}_{0|-1} = E\{\mathbf{x}_0\} = \overline{\mathbf{x}_0} \tag{F.3}$$
$$\mathbf{P}_{0|-1} = Var\{\mathbf{x}_0\} \tag{F.4}$$

Filtering *for $k = 1, 2, \cdots, N$, (Forwards)*

Prediction

$$\widehat{\mathbf{x}}_{k|k-1} = \mathbf{A}\widehat{\mathbf{x}}_{k-1|k-1} \tag{F.5}$$
$$\mathbf{P}_{k|k-1} = \mathbf{A}\mathbf{P}_{k-1|k-1}\mathbf{A}^T + \mathbf{Q} \tag{F.6}$$

Update

$$\boldsymbol{\Sigma}_{k|k-1} = \mathbf{C}\mathbf{P}_{k|k-1}\mathbf{C}^T + \mathbf{R} \tag{F.7}$$
$$\mathbf{K}_k = \mathbf{P}_{k|k-1}\mathbf{C}^T\boldsymbol{\Sigma}_{k|k-1}^{-1} \tag{F.8}$$
$$\mathbf{e}_k = \mathbf{y}_k - \mathbf{C}\widehat{\mathbf{x}}_{k|k-1} \tag{F.9}$$
$$\widehat{\mathbf{x}}_{k|k} = \widehat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k\mathbf{e}_k \tag{F.10}$$
$$\mathbf{P}_{k|k} = \mathbf{P}_{k|k-1} - \mathbf{K}_k\mathbf{C}\mathbf{P}_{k|k-1} \tag{F.11}$$

One-step ahead Prediction

$$\widehat{\mathbf{x}}_{k+1|k} = \left[\mathbf{A}_k - \mathbf{A}_k\mathbf{K}_k\mathbf{C}_k^T\right] + \mathbf{K}_k\mathbf{y}_k \tag{F.12}$$
$$\mathbf{P}_{k+1|k} = \mathbf{A}_k\mathbf{P}_{k|k}\mathbf{A}_k^T + \mathbf{Q}_k \tag{F.13}$$

Smoothing *for $N > k$ (Backwards)*

$$\mathbf{x}_{k-1|N} = \widehat{\mathbf{x}}_{k-1|k-1} + \mathbf{J}_{k-1}\left[\widehat{\mathbf{x}}_{k|N} - \widehat{\mathbf{x}}_{k|k-1}\right] \tag{F.14}$$

$$\mathbf{P}_{k-1|N} = \mathbf{P}_{k-1|k-1} + \mathbf{J}_{k-1}\left[\mathbf{P}_{k|N} - \mathbf{P}_{k|k-1}\right]\mathbf{J}_{k-1}^T \tag{F.15}$$

$$\mathbf{J}_{k-1} = \mathbf{P}_{k-1|k-1}\mathbf{A}_{k-1}^T\mathbf{P}_{k|k-1}^{-1} \tag{F.16}$$

F.2 Likelihood

Maximum likelihood estimates of the unknown parameters θ can be obtained by maximizing the likelihood:

$$L(\boldsymbol{\theta}|\mathbf{y}_k) = p(\mathbf{y}_k|\boldsymbol{\theta}). \tag{F.17}$$

Let consider the set of observations: $\mathbf{y}_{1:N} = \{\mathbf{y}_0, \mathbf{y}_1, \cdots, \mathbf{y}_{1:N}\}$, then the log-likelihood is given by:

$$\log L_N = -\frac{Nq}{2}\log(2\pi) - \frac{1}{2}\sum_{k=1}^{N}\log \mid \boldsymbol{\Sigma}_{k|k-1}(\boldsymbol{\theta}) \mid -\frac{1}{2}\sum_{k=1}^{N}(\mathbf{e}_k(\boldsymbol{\theta})^T\boldsymbol{\Sigma}_{k|k-1}(\boldsymbol{\theta})^{-1}\mathbf{e}_k(\boldsymbol{\theta})), \tag{F.18}$$

where $\mathbf{e}_k(\boldsymbol{\theta})$, and $\boldsymbol{\Sigma}_{k|k-1}(\boldsymbol{\theta})$ are given by the Kalman filter.

Let consider:

$$J(\boldsymbol{\theta}|\mathbf{y}_{1:N}) = -2\log L_N(\boldsymbol{\theta}|Y_{1:N}) \tag{F.19}$$

than:

$$J(\boldsymbol{\theta}|\mathbf{y}_{1:N}) = \frac{Nq}{2}\log(2\pi) + \frac{1}{2}\sum_{k=1}^{N}\log \mid \boldsymbol{\Sigma}_{k|k-1}(\boldsymbol{\theta}) \mid +\frac{1}{2}\sum_{k=1}^{N}\mathbf{e}_k^T(\boldsymbol{\theta})\boldsymbol{\Sigma}_{k|k-1}^{-1}(\boldsymbol{\theta})\mathbf{e}_k(\boldsymbol{\theta}) \tag{F.20}$$

To derive the EM algorithm we need to develop a probabilistic model for Eq. (F.1), and (F.2).

We assume the likelihood of the data given the states, initial conditions and evolution of states to be represented by Gaussian distributions.

We have the following distributions:

$$p(\mathbf{x}_0|\boldsymbol{\theta}) = \frac{1}{(2\pi)^{\frac{N}{2}}} \frac{1}{|\mathbf{P}_0|^{\frac{1}{2}}} \exp\left[(\mathbf{x}_0 - \overline{\mathbf{x}}_0)^T \mathbf{P}_0^{-1} (\mathbf{x}_0 - \overline{\mathbf{x}}_0)\right] \tag{F.21}$$

$$p(\mathbf{x}_k|\mathbf{x}_{k-1}, \boldsymbol{\theta}) = \frac{1}{(2\pi)^{\frac{N}{2}}} \frac{1}{|\mathbf{Q}|^{\frac{1}{2}}} \exp\left[(\mathbf{x}_k - \mathbf{A}\mathbf{x}_{k-1})^T \mathbf{Q}^{-1} (\mathbf{x}_k - \mathbf{A}\mathbf{x}_{k-1})\right] \tag{F.22}$$

$$p(\mathbf{y}_k|\mathbf{x}_k, \boldsymbol{\theta}) = \frac{1}{(2\pi)^{\frac{N}{2}}} \frac{1}{|\mathbf{R}|^{\frac{1}{2}}} \exp\left[(\mathbf{y}_k - \mathbf{C}\mathbf{x}_k)^T \mathbf{R}^{-1} (\mathbf{y}_k - \mathbf{C}\mathbf{x}_k)\right], \tag{F.23}$$

Under the initial model assumptions of uncorrelated noises, Markov assumption for states evolution, and independence of measurements given the state, the joint distribution is given by:

$$p(\mathbf{x}_{0:N}, \mathbf{y}_{1:N}|\boldsymbol{\theta}) = p(\mathbf{x}_0|\boldsymbol{\theta}) \prod_{k=1}^{N} p(\mathbf{x}_k|\mathbf{x}_{k-1}, \boldsymbol{\theta}) \prod_{k=0}^{N} p(\mathbf{y}_k|\mathbf{x}_k, \boldsymbol{\theta}) \tag{F.24}$$

and the log-likelihood:

$$\begin{aligned} \log L_N(\theta|\mathbf{x}_{0:N}, \mathbf{y}_{1:N}) = & - \quad \text{constante} \\ & - \quad \frac{1}{2}(\mathbf{x}_0 - \overline{\mathbf{x}}_0)^T \mathbf{P}_0^{-1} (\mathbf{x}_0 - \overline{\mathbf{x}}_0) - \frac{1}{2} \log |\mathbf{P}_0| \\ & - \quad \frac{1}{2} \sum_{k=1}^{N} \Big[\log |\mathbf{Q}| + (\mathbf{x}_k - \mathbf{A}\mathbf{x}_{k-1})^T \mathbf{Q}^{-1} (\mathbf{x}_k - \mathbf{A}\mathbf{x}_{k-1}) \Big] \\ & - \quad \frac{1}{2} \sum_{k=0}^{N} \Big[\log |\mathbf{R}| + (\mathbf{y}_k - \mathbf{C}\mathbf{x}_k)^T \mathbf{R}^{-1} (\mathbf{y}_k - \mathbf{C}\mathbf{x}_k) \Big] \end{aligned} \tag{F.25}$$

than :

$$\begin{aligned} J(\boldsymbol{\theta}|\mathbf{x}_{0:N}, \mathbf{y}_{1:N}) = & (\mathbf{x}_0 - \overline{\mathbf{x}}_0)^T \mathbf{P}_0^{-1} (\mathbf{x}_0 - \overline{\mathbf{x}}_0) + \log |\mathbf{P}_0| \\ & + \quad \sum_{k=1}^{N} \Big\{ \log \|\mathbf{Q}\| + (\mathbf{x}_k - \mathbf{A}\mathbf{x}_{k-1})^T \mathbf{Q}^{-1} (\mathbf{x}_k - \mathbf{A}\mathbf{x}_{k-1}) \Big\} \\ & + \quad \sum_{k=0}^{N} \Big\{ \log \|\mathbf{R}\| + (\mathbf{y}_k - \mathbf{C}\mathbf{x}_k)^T \mathbf{R}^{-1} (\mathbf{y}_k - \mathbf{C}\mathbf{x}_k) \Big\} \end{aligned} \tag{F.26}$$

All we need to do is to compute the expectation of the log-likelihood and then differentiate the result with respect to the parameters so as to maximise it.

The EM algorithm involves computing the expected values of the states and covariances with the extended Kalman smoother and then estimating the parameters with the formulae obtained by differentiating the expected log-likelihood.

F.3 Derivatives of the likelihood function

F.3.1 Estimation of A

Differentiating with respect to $\mathbf{A}$ yields:

$$\frac{\partial J}{\partial \mathbf{A}} = 0 \tag{F.27}$$

where :

$$\frac{\partial J}{\partial \mathbf{A}} = \frac{\partial}{\partial \mathbf{A}} \sum_{k=1}^{N} (\mathbf{x}_k - \mathbf{A}\mathbf{x}_{k-1})^T \mathbf{Q}^{-1} (\mathbf{x}_k - \mathbf{A}\mathbf{x}_{k-1}) \tag{F.28}$$

but we have :

$$(\mathbf{x}_k - \mathbf{A}\mathbf{x}_{k-1})^T \mathbf{Q}^{-1} (\mathbf{x}_k - \mathbf{A}\mathbf{x}_{k-1}) = tr\Big[\mathbf{Q}^{-1} (\mathbf{x}_k - \mathbf{A}\mathbf{x}_{k-1})(\mathbf{x}_k - \mathbf{A}\mathbf{x}_{k-1})^T\Big] \tag{F.29}$$

and the results:

$$\frac{\partial}{\partial \mathbf{A}} tr[\mathbf{A}\mathbf{B}] = \mathbf{B}^T \tag{F.30}$$

$$\frac{\partial}{\partial \mathbf{A}} tr[\mathbf{A}^T\mathbf{B}] = \mathbf{B} \tag{F.31}$$

$$tr[\mathbf{A} + \mathbf{B}] = tr[\mathbf{A}] + tr[\mathbf{B}] \tag{F.32}$$

$$tr[\mathbf{A}\mathbf{B}] = tr[\mathbf{A}^T + \mathbf{B}^T] = tr[\mathbf{B}\mathbf{A}] = tr[\mathbf{B}^T + \mathbf{A}^T] \tag{F.33}$$

hence :

$$\frac{\partial}{\partial \mathbf{A}} tr\big[\mathbf{C}\mathbf{A}\mathbf{B}\mathbf{A}^T\big] = \frac{\partial}{\partial \mathbf{A}} tr\big[\mathbf{A}^T\mathbf{C}^T\mathbf{A}^*\mathbf{B}^T\big] + \frac{\partial}{\partial \mathbf{A}} tr\big[\mathbf{A}\mathbf{B}^T\mathbf{A}^{*T}\mathbf{C}^T\big] \tag{F.34}$$

$$= \mathbf{C}^T\mathbf{A}\mathbf{B}^T + \mathbf{C}\mathbf{A}\mathbf{B} \tag{F.35}$$

with $\mathbf{A}^* = \mathbf{A}$ fixed.
Then :

$$tr\Big[\mathbf{Q}^{-1} (\mathbf{x}_k - \mathbf{A}\mathbf{x}_{k-1})(\mathbf{x}_k - \mathbf{A}\mathbf{x}_{k-1})^T\Big] = tr\big[\mathbf{Q}^{-1}\mathbf{x}_k\mathbf{x}_k^T\big]$$
$$- tr\big[\mathbf{Q}^{-1}\mathbf{A}\mathbf{x}_{k-1}\mathbf{x}_k^T\big] - tr\big[\mathbf{Q}^{-1}\mathbf{x}_k\mathbf{x}_{k-1}^T\mathbf{A}^T\big] + tr\big[\mathbf{Q}^{-1}\mathbf{A}\mathbf{x}_{k-1}\mathbf{x}_{k-1}^T\mathbf{A}^T\big] \tag{F.36}$$

we find :

$$\frac{\partial}{\partial \mathbf{A}} tr\left[\mathbf{Q}^{-1}\mathbf{A}\mathbf{x}_{k-1}\mathbf{x}_{k-1}^T\mathbf{A}^T\right] = \mathbf{Q}^{-1}\mathbf{A}\mathbf{x}_{k-1}\mathbf{x}_{k-1}^T + \mathbf{Q}^{-1}\mathbf{A}\mathbf{x}_{k-1}\mathbf{x}_{k-1}^T \quad \text{(F.37)}$$

$$\frac{\partial}{\partial \mathbf{A}} tr\left[\mathbf{Q}^{-1}\mathbf{A}\mathbf{x}_{k-1}\mathbf{x}_{k}^T\right] = \frac{\partial}{\partial \mathbf{A}} tr\left[\mathbf{A}^T\mathbf{Q}^{-1}\mathbf{x}_{k}\mathbf{x}_{k-1}^T\right] \quad \text{(F.38)}$$

$$= \mathbf{Q}^{-1}\mathbf{x}_{k}\mathbf{x}_{k-1}^T \quad \text{(F.39)}$$

$$\frac{\partial}{\partial \mathbf{A}} tr\left[\mathbf{Q}^{-1}\mathbf{x}_{k}\mathbf{x}_{k-1}^T\mathbf{A}^T\right] = \frac{\partial}{\partial \mathbf{A}} tr\left[\mathbf{A}\mathbf{x}_{k-1}\mathbf{x}_{k}^T\mathbf{Q}^{-1}\right] \quad \text{(F.40)}$$

$$= \mathbf{Q}^{-1}\mathbf{x}_{k}\mathbf{x}_{k-1}^T \quad \text{(F.41)}$$

that gives :

$$\frac{\partial}{\partial \mathbf{A}} tr\Big[\mathbf{Q}^{-1}(\mathbf{x}_k - \mathbf{A}\mathbf{x}_{k-1})(\mathbf{x}_k - \mathbf{A}\mathbf{x}_{k-1})^T\Big] = -2\mathbf{Q}^{-1}\mathbf{x}_{k}\mathbf{x}_{k-1}^T + 2\mathbf{Q}^{-1}\mathbf{A}\mathbf{x}_{k-1}\mathbf{x}_{k-1}^T \quad \text{(F.42)}$$

and :

$$\frac{\partial J}{\partial \mathbf{A}} = -2\mathbf{Q}^{-1}\sum_{k=1}^{N}\mathbf{x}_{k}\mathbf{x}_{k-1}^T + 2\mathbf{Q}^{-1}\sum_{k=1}^{N}\mathbf{A}\mathbf{x}_{k-1}\mathbf{x}_{k-1}^T \quad \text{(F.43)}$$

but we want to have:

$$\frac{\partial J}{\partial \mathbf{A}} = 0 \quad \text{(F.44)}$$

hence :

$$\sum_{k=1}^{N}\mathbf{x}_{k}\mathbf{x}_{k-1}^T = \mathbf{A}\sum_{k=1}^{N}\mathbf{x}_{k-1}\mathbf{x}_{k-1}^T \quad \text{(F.45)}$$

so we find :

$$\Gamma_4 = \frac{1}{N}\sum_{k=1}^{N}\mathbf{x}_{k}\mathbf{x}_{k-1}^T \quad \text{(F.46)}$$

$$\Gamma_3 = \frac{1}{N}\sum_{k=1}^{N}\mathbf{x}_{k-1}\mathbf{x}_{k-1}^T \quad \text{(F.47)}$$

and we have :

$$\widehat{\mathbf{A}} = \Gamma_4\Gamma_3^{-1} \quad \text{(F.48)}$$

F.3.2 Estimation of C

Differentiating with respect to **C** yields:

$$\frac{\partial J}{\partial \mathbf{C}} = 0 \tag{F.49}$$

where :

$$\frac{\partial J}{\partial \mathbf{C}} = \frac{\partial}{\partial \mathbf{C}} \sum_{k=0}^{N} (\mathbf{y}_k - \mathbf{C}\mathbf{x}_k)^T \mathbf{R}^{-1} (\mathbf{y}_k - \mathbf{C}\mathbf{x}_k) \tag{F.50}$$

but we have :

$$(\mathbf{y}_k - \mathbf{C}\mathbf{x}_k)^T \mathbf{R}^{-1} (\mathbf{y}_k - \mathbf{C}\mathbf{x}_k) = tr\Big[\mathbf{R}^{-1} (\mathbf{y}_k - \mathbf{C}\mathbf{x}_k)(\mathbf{y}_k - \mathbf{C}\mathbf{x}_k)^T\Big] \tag{F.51}$$

let :

$$\begin{aligned} &tr\Big[\mathbf{R}^{-1} (\mathbf{y}_k - \mathbf{C}\mathbf{x}_k)(\mathbf{y}_k - \mathbf{C}\mathbf{x}_k)^T\Big] = \\ &tr\big[\mathbf{R}^{-1}\mathbf{y}_k\mathbf{y}_k^T\big] - tr\big[\mathbf{R}^{-1}\mathbf{C}\mathbf{x}_k\mathbf{y}_k^T\big] - tr\big[\mathbf{R}^{-1}\mathbf{y}_k\mathbf{x}_k^T\mathbf{C}^T\big] + tr\big[\mathbf{R}^{-1}\mathbf{C}\mathbf{x}_k\mathbf{x}_k^T\mathbf{C}^T\big] \end{aligned} \tag{F.52}$$

we find :

$$\frac{\partial}{\partial \mathbf{C}} tr\big[\mathbf{R}^{-1}\mathbf{C}\mathbf{x}_k\mathbf{x}_k^T\mathbf{C}^T\big] = \mathbf{R}^{-1}\mathbf{C}\mathbf{x}_k\mathbf{x}_k^T + \mathbf{R}^{-1}\mathbf{C}\mathbf{x}_k\mathbf{x}_k^T \tag{F.53}$$

$$\begin{aligned} \frac{\partial}{\partial \mathbf{C}} tr\big[\mathbf{R}^{-1}\mathbf{C}\mathbf{x}_k\mathbf{y}_k^T\big] &= \frac{\partial}{\partial \mathbf{C}} tr\big[\mathbf{C}^T\mathbf{R}^{-1}\mathbf{y}_k\mathbf{x}_k^T\big] \\ &= \mathbf{R}^{-1}\mathbf{y}_k\mathbf{x}_k^T \end{aligned} \tag{F.54}$$

$$\begin{aligned} \frac{\partial}{\partial \mathbf{C}} tr\big[\mathbf{R}^{-1}\mathbf{y}_k\mathbf{x}_k^T\mathbf{C}^T\big] &= \frac{\partial}{\partial \mathbf{C}} tr\big[\mathbf{C}\mathbf{x}_k\mathbf{y}_k^T\mathbf{R}^{-1}\big] \\ &= \mathbf{R}^{-1}\mathbf{y}_k\mathbf{x}_k^T \end{aligned} \tag{F.55}$$

that gives :

$$\frac{\partial}{\partial \mathbf{C}} tr\Big[\mathbf{R}^{-1} (\mathbf{y}_k - \mathbf{C}\mathbf{y}_k)(\mathbf{y}_k - \mathbf{C}\mathbf{y}_k)^T\Big] = -2\mathbf{R}^{-1}\mathbf{y}_k\mathbf{x}_k^T + 2\mathbf{R}^{-1}\mathbf{C}\mathbf{x}_k\mathbf{x}_k^T \tag{F.56}$$

and :

$$\frac{\partial J}{\partial \mathbf{C}} = -2\mathbf{R}^{-1} \sum_{k=0}^{N} \mathbf{y}_k\mathbf{x}_k^T + 2\mathbf{R}^{-1} \sum_{k=0}^{N} \mathbf{C}\mathbf{x}_k\mathbf{x}_k^T \tag{F.57}$$

we want to have:

$$\frac{\partial J}{\partial \mathbf{C}} = 0 \tag{F.58}$$

hence :

$$\sum_{k=0}^{N} \mathbf{y}_k \mathbf{x}_k^T = \mathbf{C} \sum_{k=0}^{N} \mathbf{x}_k \mathbf{x}_k^T \tag{F.59}$$

so we find :

$$\Gamma_1 = \frac{1}{N+1} \sum_{k=0}^{N} \mathbf{x}_k \mathbf{x}_k^T \tag{F.60}$$

$$\Gamma_6 = \frac{1}{N+1} \sum_{k=0}^{N} \mathbf{y}_k \mathbf{x}_k^T \tag{F.61}$$

and we have :

$$\widehat{\mathbf{C}} = \Gamma_6 \Gamma_1^{-1} \tag{F.62}$$

F.3.3 Estimation of Q

Differentiating with respect to $\mathbf{Q}$ yields:

$$\frac{\partial J}{\partial \mathbf{Q}} = \frac{\partial J}{\partial \mathbf{Q^{-1}}} = 0 \tag{F.63}$$

where :

$$\frac{\partial J}{\partial \mathbf{Q}} = \frac{\partial}{\partial \mathbf{Q}} \sum_{k=1}^{N} \Big\{ \log \|\mathbf{Q}\| + (\mathbf{x}_k - \mathbf{A}\mathbf{x}_{k-1})^T \mathbf{Q}^{-1} (\mathbf{x}_k - \mathbf{A}\mathbf{x}_{k-1}) \Big\} \tag{F.64}$$

but we have:

$$\log |\mathbf{Q}| = -\log |\mathbf{Q}^{-1}| \tag{F.65}$$

hence :

$$\frac{\partial}{\partial \mathbf{Q}} \log |\mathbf{Q}| = -\frac{\partial}{\partial \mathbf{Q^{-1}}} \log |\mathbf{Q}^{-1}| = -\mathbf{Q} \tag{F.66}$$

we have :

$$\begin{aligned} &\frac{\partial}{\partial \mathbf{Q^{-1}}} \Big[(\mathbf{x}_k - \mathbf{A}\mathbf{x}_{k-1})^T \mathbf{Q}^{-1} (\mathbf{x}_k - \mathbf{A}\mathbf{x}_{k-1}) \Big] \\ &\quad = \quad \frac{\partial}{\partial \mathbf{Q^{-1}}} tr \Big[\mathbf{Q}^{-1} (\mathbf{x}_k - \mathbf{A}\mathbf{x}_{k-1}) (\mathbf{x}_k - \mathbf{A}\mathbf{x}_{k-1})^T \Big] \\ &\quad = \quad (\mathbf{x}_k - \mathbf{A}\mathbf{x}_{k-1}) (\mathbf{x}_k - \mathbf{A}\mathbf{x}_{k-1})^T \end{aligned} \tag{F.67}$$

hence :

$$\frac{\partial J}{\partial \mathbf{Q^{-1}}} = \sum_{k=1}^{N} -\mathbf{Q} + \sum_{k=1}^{N} (\mathbf{x}_k - \mathbf{A}\mathbf{x}_{k-1})(\mathbf{x}_k - \mathbf{A}\mathbf{x}_{k-1})^T \tag{F.68}$$

that gives :

$$\begin{aligned}\widehat{\mathbf{Q}} = \frac{1}{N}\sum_{k=1}^{N} \mathbf{x}_k\mathbf{x}_k^T - \mathbf{A}\Big[\frac{1}{N}\sum_{k=1}^{N}\mathbf{x}_{k-1}\mathbf{x}_k^T\Big] \\ -\Big[\frac{1}{N}\sum_{k=1}^{N}\mathbf{x}_k\mathbf{x}_{k-1}^T\Big]\mathbf{A}^T + \mathbf{A}\Big[\frac{1}{N}\sum_{k=1}^{N}\mathbf{x}_{k-1}\mathbf{x}_{k-1}^T\Big]\mathbf{A}^T\end{aligned} \tag{F.69}$$

let :

$$\Gamma_2 = \frac{1}{N}\sum_{k=1}^{N}\mathbf{x}_k\mathbf{x}_k^T \tag{F.70}$$

$$\Gamma_3 = \frac{1}{N}\sum_{k=1}^{N}\mathbf{x}_{k-1}\mathbf{x}_{k-1}^T \tag{F.71}$$

$$\Gamma_4 = \frac{1}{N}\sum_{k=1}^{N}\mathbf{x}_k\mathbf{x}_{k-1}^T \tag{F.72}$$

that gives :

$$\widehat{\mathbf{Q}} = \Gamma_2 - \widehat{\mathbf{A}}\Gamma_4^T - \Gamma_4\widehat{\mathbf{A}}^T + \widehat{\mathbf{A}}\Gamma_3\widehat{\mathbf{A}}^T \tag{F.73}$$

and we have :

$$\widehat{\mathbf{Q}} = \Gamma_2 - \widehat{\mathbf{A}}\Gamma_4^T \tag{F.74}$$

F.3.4 Estimation of R

Differentiating with respect to $\mathbf{R}$ yields:

$$\frac{\partial J}{\partial \mathbf{R}} = \frac{\partial J}{\partial \mathbf{R^{-1}}} = 0 \tag{F.75}$$

where :

$$\frac{\partial J}{\partial \mathbf{R}} = \frac{\partial}{\partial \mathbf{R}}\sum_{k=0}^{N}\Big\{\log\|\mathbf{R}\| + (\mathbf{y}_k - \mathbf{C}\mathbf{x}_k)^T\mathbf{R}^{-1}(\mathbf{y}_k - \mathbf{C}\mathbf{x}_k)\Big\} \tag{F.76}$$

we have:

$$\log|\mathbf{R}| = -\log|\mathbf{R}^{-1}| \tag{F.77}$$

hence :

$$\frac{\partial}{\partial \mathbf{R}} \log |\mathbf{R}| = -\frac{\partial}{\partial \mathbf{R^{-1}}} \log |\mathbf{R}^{-1}| = -\mathbf{R} \tag{F.78}$$

we have :

$$\begin{aligned} \frac{\partial}{\partial \mathbf{R^{-1}}} \Big[(\mathbf{y}_k - \mathbf{C}\mathbf{x}_k)^T \mathbf{R}^{-1} (\mathbf{y}_k - \mathbf{C}\mathbf{x}_k) \Big] \\ &= \frac{\partial}{\partial \mathbf{R^{-1}}} tr \Big[\mathbf{R}^{-1} (\mathbf{y}_k - \mathbf{C}\mathbf{x}_k)(\mathbf{y}_k - \mathbf{C}\mathbf{x}_k)^T \Big] \\ &= (\mathbf{y}_k - \mathbf{C}\mathbf{x}_k)(\mathbf{y}_k - \mathbf{C}\mathbf{x}_k)^T \end{aligned} \tag{F.79}$$

hence :

$$\frac{\partial J}{\partial \mathbf{R^{-1}}} = \sum_{k=0}^{N} -\mathbf{R} + \sum_{k=0}^{N} (\mathbf{y}_k - \mathbf{C}\mathbf{x}_k)(\mathbf{y}_k - \mathbf{C}\mathbf{x}_k)^T \tag{F.80}$$

we find :

$$\begin{aligned} \widehat{\mathbf{R}} = \frac{1}{N+1} \sum_{k=0}^{N} \mathbf{y}_k \mathbf{y}_k^T - \mathbf{C} \Big[\frac{1}{N+1} \sum_{k=0}^{N} \mathbf{x}_k \mathbf{y}_k^T \Big] \\ - \Big[\frac{1}{N+1} \sum_{k=0}^{N} \mathbf{y}_k \mathbf{x}_k^T \Big] \mathbf{C}^T + \mathbf{C} \Big[\frac{1}{N+1} \sum_{k=0}^{N} \mathbf{x}_k \mathbf{x}_k^T \Big] \mathbf{C^T} \end{aligned} \tag{F.81}$$

let :

$$\Gamma_5 = \frac{1}{N+1} \sum_{k=0}^{N} \mathbf{y}_k \mathbf{y}_k^T \tag{F.82}$$

$$\Gamma_6 = \frac{1}{N+1} \sum_{k=0}^{N} \mathbf{y}_k \mathbf{x}_k^T \tag{F.83}$$

$$\Gamma_1 = \frac{1}{N+1} \sum_{k=0}^{N} \mathbf{x}_k \mathbf{x}_k^T \tag{F.84}$$

hence :

$$\widehat{\mathbf{R}} = \Gamma_5 - \widehat{\mathbf{C}} \Gamma_6^T - \Gamma_6 \widehat{\mathbf{C}}^T + \widehat{\mathbf{C}} \Gamma_1 \widehat{\mathbf{C}}^T \tag{F.85}$$

we get :

$$\widehat{\mathbf{R}} = \Gamma_5 - \widehat{\mathbf{C}} \Gamma_6^T \tag{F.86}$$

F.3.5 Conclusions

If we had the sufficient statistics:

$$\Gamma_1 = \frac{1}{N+1}\sum_{k=0}^{N}\mathbf{x}_k\mathbf{x}_k^T \tag{F.87}$$

$$\Gamma_2 = \frac{1}{N}\sum_{k=1}^{N}\mathbf{x}_k\mathbf{x}_k^T \tag{F.88}$$

$$\Gamma_3 = \frac{1}{N}\sum_{k=1}^{N}\mathbf{x}_{k-1}\mathbf{x}_{k-1}^T \tag{F.89}$$

$$\Gamma_4 = \frac{1}{N}\sum_{k=1}^{N}\mathbf{x}_k\mathbf{x}_{k-1}^T \tag{F.90}$$

$$\Gamma_5 = \frac{1}{N+1}\sum_{k=0}^{N}\mathbf{y}_k\mathbf{y}_k^T \tag{F.91}$$

$$\Gamma_6 = \frac{1}{N+1}\sum_{k=0}^{N}\mathbf{y}_k\mathbf{x}_k^T \tag{F.92}$$

we could estimate the parameters:

$$\widehat{\mathbf{A}} = \Gamma_4\Gamma_3^{-1} \tag{F.93}$$

$$\widehat{\mathbf{C}} = \Gamma_6\Gamma_1^{-1} \tag{F.94}$$

$$\widehat{\mathbf{Q}} = \Gamma_2 - \widehat{\mathbf{A}}\Gamma_4^T \tag{F.95}$$

$$\widehat{\mathbf{R}} = \Gamma_5 - \widehat{\mathbf{C}}\Gamma_6^T \tag{F.96}$$

F.4 Parameter Estimation by EM Algorithm

In this section we derive the sufficient statistics by EM algorithm.

The EM algorithm is an iterative procedure for finding a mode of the likelihood function $p(\mathbf{y}|\theta)$.

The EM algorithm involves at each iteration the computation of the sufficient statistics described previously using the recursions above and the old estimates of the model parameters in the E-step. The new estimates for the system parameters can then be obtained from these statistics as the simple multivariate regression coefficients in the M-step.

By EM-algorithm, we maximize:

$$Q(\boldsymbol{\theta}^{(p+1)}|\boldsymbol{\theta}^{(p)}) = \mathbb{E}_{\theta^{(p)}}\left\{J\big(\mathbf{x}_{0:N}, \mathbf{y}_{1:N}; \boldsymbol{\theta}^{(p+1)}|\mathbf{y}_{1:N}\big)\right\} \tag{F.97}$$

with $J\big(\mathbf{x}_{0:N}, \mathbf{y}_{1:N}; \boldsymbol{\theta}\big)$ given by :

$$\begin{aligned} J(\mathbf{x}_{0:N}, \mathbf{y}_{1:N}; \boldsymbol{\theta}) = & (\mathbf{x}_0 - \overline{\mathbf{x}}_0)^T \mathbf{P}_0^{-1} (\mathbf{x}_0 - \overline{\mathbf{x}}_0) + \log|\mathbf{P}_0| \\ + & \sum_{k=1}^{N} \Big\{ \log\|\mathbf{Q}\| + (\mathbf{x}_k - \mathbf{A}\mathbf{x}_{k-1})^T \mathbf{Q}^{-1} (\mathbf{x}_k - \mathbf{A}\mathbf{x}_{k-1}) \Big\} \\ + & \sum_{k=0}^{N} \Big\{ \log\|\mathbf{R}\| + (\mathbf{y}_k - \mathbf{C}\mathbf{x}_k)^T \mathbf{R}^{-1} (\mathbf{y}_k - \mathbf{C}\mathbf{x}_k) \Big\} \end{aligned} \tag{F.98}$$

F.4.1 E-step

In the E-stet we estimate the sufficient statistics:

$$E_{\theta^{(p)}}\left\{\mathbf{y}_k \mathbf{y}_k^T | Y_{1:N}\right\} \tag{F.99}$$
$$E_{\theta^{(p)}}\left\{\mathbf{y}_k \mathbf{x}_k^T | Y_{1:N}\right\} \tag{F.100}$$
$$E_{\theta^{(p)}}\left\{\mathbf{x}_k \mathbf{x}_k^T | Y_{1:N}\right\} \tag{F.101}$$
$$E_{\theta^{(p)}}\left\{\mathbf{x}_k \mathbf{x}_{k-1}^T | Y_{1:N}\right\} \tag{F.102}$$

from former estimations, and Kalman filter.

We consider Gaussian non correlated noises, Gaussian initial state, than the distribution of states will be Gaussian, suc as:

$$p(\widehat{\mathbf{x}}_k|Y_N) \simeq N\big(\mathbf{x}_{k|N}, \mathbf{P}_{k|N}\big) \tag{F.103}$$

with :

$$\widehat{\mathbf{x}}_k|Y_N = E_{\theta^{(p)}}\left\{\mathbf{x}_k|Y_N\right\} \tag{F.104}$$
$$\mathbf{P}_{k|N} = Var_{\theta^{(p)}}\left\{\mathbf{x}_k|Y_N\right\} \tag{F.105}$$
$$\mathbf{P}_{k|N} = E_{\theta^{(p)}}\left\{\mathbf{x}_k \mathbf{x}_k^T|Y_N\right\} - E_{\theta^{(p)}}\left\{\mathbf{x}_k|Y_N\right\} E_{\theta^{(p)}}\left\{\mathbf{x}_k|Y_N\right\} \tag{F.106}$$

Estimation of $\mathbb{E}_{\boldsymbol{\theta^{(p)}}}\left\{\mathbf{y_k} \mathbf{y_k^T} | \boldsymbol{Y_{1:N}}\right\}$

$$E_{\theta^{(p)}}\left\{\mathbf{y}_k \mathbf{y}_k^T|Y_N\right\} = \mathbf{y}_k \mathbf{y}_{k^T} \tag{F.107}$$

Estimation of $\mathbb{E}_{\boldsymbol{\theta}^{(p)}}\left\{\mathbf{y}_{\boldsymbol{k}}\mathbf{x}_{\boldsymbol{k}}^{\boldsymbol{T}}|\boldsymbol{Y}_{\mathbf{1:N}}\right\}$

$$E_{\theta^{(p)}}\left\{\mathbf{y}_k\mathbf{x}_k^T|Y_N\right\} = \mathbf{y}_k E_{\theta^{(p)}}\left\{\mathbf{x}_k^T|Y_N\right\} \tag{F.108}$$

so :

$$E_{\theta^{(p)}}\left\{\mathbf{y}_k\mathbf{x}_k^T|Y_N\right\} = \mathbf{y}_k\widehat{\mathbf{x}}_{k|N}^T \tag{F.109}$$

Estimation of $\mathbb{E}_{\boldsymbol{\theta}^{(p)}}\left\{\mathbf{x}_{\boldsymbol{k}}\mathbf{x}_{\boldsymbol{k}}^{\boldsymbol{T}}|\boldsymbol{Y}_{\mathbf{1:N}}\right\}$

$$E_{\theta^{(p)}}\left\{\mathbf{x}_k\mathbf{x}_k^T|Y_N\right\} = P_{k|N} + \widehat{\mathbf{x}}_{k|N}\widehat{\mathbf{x}}_{k|N}^T \tag{F.110}$$

Estimation of $\mathbb{E}_{\boldsymbol{\theta}^{(p)}}\left\{\mathbf{x}_{\boldsymbol{k}}\mathbf{x}_{\boldsymbol{k-1}}^{\boldsymbol{T}}|\boldsymbol{Y}_{\mathbf{1:N}}\right\}$

by definition :

$$\mathbf{P}_{k,k-1|l} = E\big\{(\mathbf{x}_k - \widehat{\mathbf{x}}_{k|l})(\mathbf{x}_{k-1} - \widehat{\mathbf{x}}_{k-1|l})^T|Y_l\big\} \tag{F.111}$$

$$\begin{aligned} E_{\theta^{(p)}}\left\{\mathbf{x}_k\mathbf{x}_{k-1}^T|Y_N\right\} &= E_{\theta^{(p)}}\left\{(\mathbf{x}_k - \widehat{\mathbf{x}}_{k|N})(\mathbf{x}_{k-1} - \widehat{\mathbf{x}}_{k-1|N})^T|Y_N\right\} \\ &+ E_{\theta^{(p)}}\left\{\mathbf{x}_k\widehat{\mathbf{x}}_{k-1|N}^T|Y_N\right\} \\ &+ E_{\theta^{(p)}}\left\{\widehat{\mathbf{x}}_{k|N}\mathbf{x}_{k-1}^T|Y_N\right\} \\ &- E_{\theta^{(p)}}\left\{\widehat{\mathbf{x}}_{k|N}\widehat{\mathbf{x}}_{k-1|N}^T|Y_N\right\} \end{aligned} \tag{F.112}$$

$$\begin{aligned} E_{\theta^{(p)}}\left\{\mathbf{x}_k\mathbf{x}_{k-1}^T|Y_N\right\} &= E_{\theta^{(p)}}\left\{(\mathbf{x}_k - \widehat{\mathbf{x}}_{k|N})(\mathbf{x}_{k-1} - \widehat{\mathbf{x}}_{k-1|N})^T|Y_N\right\} \\ &+ E_{\theta^{(p)}}\left\{\mathbf{x}_k|Y_N\right\}\widehat{\mathbf{x}}_{k-1|N}^T \\ &+ \widehat{\mathbf{x}}_{k|N}E_{\theta^{(p)}}\left\{\mathbf{x}_{k-1}^T|Y_N\right\} \\ &- \widehat{\mathbf{x}}_{k|N}\widehat{\mathbf{x}}_{k-1|N}^T \end{aligned} \tag{F.113}$$

$$\begin{aligned} E_{\theta^{(p)}}\left\{\mathbf{x}_k\mathbf{x}_{k-1}^T|Y_N\right\} &= E_{\theta^{(p)}}\left\{(\mathbf{x}_k - \widehat{\mathbf{x}}_{k|N})(\mathbf{x}_{k-1} - \widehat{\mathbf{x}}_{k-1|N})^T|Y_N\right\} \\ &+ \widehat{\mathbf{x}}_{k|N}\widehat{\mathbf{x}}_{k-1|N}^T \\ &+ \widehat{\mathbf{x}}_{k|N}\widehat{\mathbf{x}}_{k-1|N}^T \\ &- \widehat{\mathbf{x}}_{k|N}\widehat{\mathbf{x}}_{k-1|N}^T \end{aligned} \tag{F.114}$$

$$E_{\theta^{(p)}}\{\mathbf{x}_k\mathbf{x}_{k-1}^T|Y_N\} = E_{\theta^{(p)}}\{(\mathbf{x}_k - \widehat{\mathbf{x}}_{k|N})(\mathbf{x}_{k-1} - \widehat{\mathbf{x}}_{k-1|N})^T|Y_N\} + \widehat{\mathbf{x}}_{k|N}\widehat{\mathbf{x}}_{k-1|N}^T \tag{F.115}$$

so :

$$E_{\theta^{(p)}}\{\mathbf{x}_k\mathbf{x}_{k-1}^T|Y_N\} = \mathbf{P}_{k,k-1|N} + \widehat{\mathbf{x}}_{k|N}\widehat{\mathbf{x}}_{k-1|N}^T \tag{F.116}$$

using Eq. (F.146) and Eq. (F.188) we find :

$$\mathbf{P}_{k,k-1|N} = \mathbf{P}_{k,k-1|k} + \big[\mathbf{P}_{k|N} - \mathbf{P}_{k,|k}\big]\mathbf{P}_{k|k}^{-1}\mathbf{P}_{k,k-1|k} \tag{F.117}$$

using Eq. (F.131) we find :

$$\mathbf{P}_{k,k-1|k} = \big[\mathbf{I} - \mathbf{K}_k\mathbf{C}\big]A\mathbf{P}_{k-1|k-1} \tag{F.118}$$

Estimation of $\mathbf{P}_{k,k-1|k}$

$$\mathbf{P}_{k,k-1|k} = E_{\theta^{(p)}}\{(\mathbf{x}_k - \widehat{\mathbf{x}}_{k|k})(\mathbf{x}_{k-1} - \widehat{\mathbf{x}}_{k-1|k})^T|Y_k\} \tag{F.119}$$

$\widehat{\mathbf{x}}_{k-1|k}$ is given by Kalman filtering:

$$\widehat{\mathbf{x}}_{k-1|k} = \widehat{\mathbf{x}}_{k-1|k-1} + \mathbf{J}_{k-1}\big[\widehat{\mathbf{x}}_{k|k} - \widehat{\mathbf{x}}_{k|k-1}\big] \tag{F.120}$$

with the gain in filtering:

$$\mathbf{J}_{k-1} = \mathbf{P}_{k-1|k-1}\mathbf{A}^T\mathbf{P}_{k|k-1}^T \tag{F.121}$$

so :

$$\begin{aligned}\mathbf{P}_{k,k-1|k} &= E\{[\mathbf{x}_k - \widehat{\mathbf{x}}_{k|k}][\mathbf{x}_{k-1} - \widehat{\mathbf{x}}_{k-1|k-1} - \mathbf{J}_k(\widehat{\mathbf{x}}_{k|k} - \widehat{\mathbf{x}}_{k|k-1})]^T\} \\ &= E\{[\mathbf{x}_k - \widehat{\mathbf{x}}_{k|k}][\mathbf{x}_{k-1} - \widehat{\mathbf{x}}_{k-1|k-1}]^T\} \\ &\quad -E\{[\mathbf{x}_k - \widehat{\mathbf{x}}_{k|k}][\widehat{\mathbf{x}}_{k|k} - \widehat{\mathbf{x}}_{k|k-1}]^T\mathbf{J}_k^T\}\end{aligned} \tag{F.122}$$

but $[\mathbf{x}_k - \widehat{\mathbf{x}}_{k|k}] \perp [\widehat{\mathbf{x}}_{k|k} - \widehat{\mathbf{x}}_{k|k-1}]$, so we have:

$$\mathbf{P}_{k,k-1|k} = E\{[\mathbf{x}_k - \widehat{\mathbf{x}}_{k|k}][\mathbf{x}_{k-1} - \widehat{\mathbf{x}}_{k-1|k-1}]^T\} \tag{F.123}$$

and by Kalman, we have :

$$\widehat{\mathbf{x}}_{k|k} = \widehat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k\mathbf{e}_k \tag{F.124}$$

$$\mathbf{e}_k = \mathbf{y}_k - \mathbf{C}\widehat{\mathbf{x}}_{k|k-1} \tag{F.125}$$

$$\mathbf{y}_k = \mathbf{C}\mathbf{x}_k + \mathbf{v}_k \tag{F.126}$$

hence :

$$\begin{aligned}\mathbf{P}_{k,k-1|k} &= E\big\{[\mathbf{x}_k - \widehat{\mathbf{x}}_{k|k-1} - \mathbf{K}_k\mathbf{e}_k][\mathbf{x}_{k-1} - \widehat{\mathbf{x}}_{k-1|k-1}]^T\big\} \\ &= E\big\{[\mathbf{x}_k - \widehat{\mathbf{x}}_{k|k-1}][\mathbf{x}_{k-1} - \widehat{\mathbf{x}}_{k-1|k-1}]^T - \mathbf{K}_k E\big\{\mathbf{e}_k[\mathbf{x}_{k-1} - \widehat{\mathbf{x}}_{k-1|k-1}]^T\big\}\end{aligned} \tag{F.127}$$

using Eq. (F.135) we have :

$$E\big\{\mathbf{e}_k[\mathbf{x}_{k-1} - \widehat{\mathbf{x}}_{k-1|k-1}]^T\big\} = \mathbf{C}\mathbf{P}_{k,k-1|k-1} \tag{F.128}$$

then :

$$\mathbf{P}_{k,k-1|k} = \mathbf{P}_{k,k-1|k-1} - \mathbf{K}_k\mathbf{C}\mathbf{P}_{k,k-1|k-1} \tag{F.129}$$

and using Eq. (F.137) we have :

$$\mathbf{P}_{k,k-1|k-1} = \mathbf{A}\mathbf{P}_{k-1|k-1} \tag{F.130}$$

that gives :

$$\mathbf{P}_{k,k-1|k} = \big[\mathbf{I} - \mathbf{K}_k\mathbf{C}\big]\mathbf{A}\mathbf{P}_{k-1|k-1} \tag{F.131}$$

Estimation of $\mathbb{E}\big\{\mathbf{e}_{\boldsymbol{k}}[\mathbf{x}_{\boldsymbol{k-1}} - \widehat{\mathbf{x}}_{\boldsymbol{k-1|k-1}}]^{\boldsymbol{T}}\big\}$

we have :

$$\mathbf{e}_k = \mathbf{y}_k - \mathbf{C}\widehat{\mathbf{x}}_{k|k-1} \tag{F.132}$$

$$\mathbf{y}_k = \mathbf{C}\mathbf{x}_k + \mathbf{v}_k \tag{F.133}$$

hence :

$$\begin{aligned}E\big\{\mathbf{e}_k[\mathbf{x}_{k-1} - \widehat{\mathbf{x}}_{k-1|k-1}]^T &= E\big\{[\mathbf{C}\mathbf{x}_k + \mathbf{v}_k - \mathbf{C}\widehat{\mathbf{x}}_{k|k-1}][\mathbf{x}_{k-1} - \widehat{\mathbf{x}}_{k-1|k-1}]^T\big\} \\ &= \mathbf{C}\; E\big\{\mathbf{x}_k - \widehat{\mathbf{x}}_{k|k-1}][\mathbf{x}_{k-1} - \widehat{\mathbf{x}}_{k-1|k-1}]^T\big\}\end{aligned} \tag{F.134}$$

hence :

$$E\big\{\mathbf{e}_k[\mathbf{x}_{k-1} - \widehat{\mathbf{x}}_{k-1|k-1}]^T = \mathbf{C}\mathbf{P}_{k,k-1|k-1} \tag{F.135}$$

Estimation of $\mathbf{P}_{\boldsymbol{k,k-1|k-1}}$

we have :

$$\begin{aligned}\mathbf{P}_{k,k-1|k-1} &= E\big\{[\mathbf{x}_k - \widehat{\mathbf{x}}_{k|k-1}][\mathbf{x}_{k-1} - \widehat{\mathbf{x}}_{k-1|k-1}]^T\big\} \\ &= E\big\{[\mathbf{A}\mathbf{x}_{k-1} - \mathbf{A}\widehat{\mathbf{x}}_{k-1|k-1}][\mathbf{x}_{k-1} - \widehat{\mathbf{x}}_{k-1|k-1}]^T\big\}\end{aligned} \tag{F.136}$$

hence :

$$\mathbf{P}_{k,k-1|k-1} = \mathbf{A}\mathbf{P}_{k-1|k-1} \tag{F.137}$$

Estimation of $\mathbf{P}_{k,k-1|N}$

$$\mathbf{P}_{k,k-1|N} = E\big\{[\mathbf{x}_k - \widehat{\mathbf{x}}_{k|N}][\mathbf{x}_{k-1} - \widehat{\mathbf{x}}_{k-1|N}]^T\big\} \tag{F.138}$$

we write :

$$[\mathbf{x}_k - \widehat{\mathbf{x}}_{k|N}] = [\mathbf{x}_k - \widehat{\mathbf{x}}_{k|k}] - [\widehat{\mathbf{x}}_{k|N} - \widehat{\mathbf{x}}_{k|k}] \tag{F.139}$$

hence :

$$\begin{aligned}\mathbf{P}_{k,k-1|N} = E\big\{[\mathbf{x}_k - \widehat{\mathbf{x}}_{k|k}][\mathbf{x}_{k-1} - \widehat{\mathbf{x}}_{k-1|N}]^T\big\} \\ -E\big\{[\widehat{\mathbf{x}}_{k|N} - \widehat{\mathbf{x}}_{k|k}][\mathbf{x}_{k-1} - \widehat{\mathbf{x}}_{k-1|N}]^T\big\}\end{aligned} \tag{F.140}$$

but $[\widehat{\mathbf{x}}_{k|N} - \widehat{\mathbf{x}}_{k|k}]$ only depends on innovations at time points $\{k+1, \cdots, N\}$, then $[\widehat{\mathbf{x}}_{k|N} - \widehat{\mathbf{x}}_{k|k}] \perp [\mathbf{x}_{k-1} - \widehat{\mathbf{x}}_{k-1|N}]$,
then we have :

$$\mathbf{P}_{k,k-1|N} = E\big\{[\mathbf{x}_k - \widehat{\mathbf{x}}_{k|k}][\mathbf{x}_{k-1} - \widehat{\mathbf{x}}_{k-1|N}]^T\big\} \tag{F.141}$$

using Eq. (F.158) we have :

$$[\mathbf{x}_{k-1} - \widehat{\mathbf{x}}_{k-1|N}] = [\mathbf{x}_{k-1} - \widehat{\mathbf{x}}_{k-1|k}] + \mathbf{P}^T_{k,k-1|k-1}\mathbf{P}^{-1}_{k|k-1}[\widehat{\mathbf{x}}_{k|k} - \widehat{\mathbf{x}}_{k|N}] \tag{F.142}$$

so :

$$\begin{aligned}\mathbf{P}_{k,k-1|N} = E\big\{[\mathbf{x}_k - \widehat{\mathbf{x}}_{k|k}][\mathbf{x}_{k-1} - \widehat{\mathbf{x}}_{k-1|k}]\big\} \\ +E\big\{[\mathbf{x}_k - \widehat{\mathbf{x}}_{k|k}][\widehat{\mathbf{x}}_{k|k} - \widehat{\mathbf{x}}_{k|N}]^T\big\}\mathbf{P}^{-1}_{k|k-1}\mathbf{P}_{k,k-1|k-1}\end{aligned} \tag{F.143}$$

but :

$$\mathbf{P}_{k,k-1|k} = E\big\{[\mathbf{x}_k - \widehat{\mathbf{x}}_{k|k}][\mathbf{x}_{k-1} - \widehat{\mathbf{x}}_{k-1|k}]^T\big\} \tag{F.144}$$

using (F.177) we have :

$$E\big\{[\mathbf{x}_k - \widehat{\mathbf{x}}_{k|k}][\widehat{\mathbf{x}}_{k|k} - \widehat{\mathbf{x}}_{k|N}]^T\big\} = \big[\mathbf{P}_{k|N} - \mathbf{P}_{k|k}\big] \tag{F.145}$$

let :

$$\mathbf{P}_{k,k-1|N} = \mathbf{P}_{k,k-1|k} + \big[\mathbf{P}_{k|N} - \mathbf{P}_{k|k}\big]\mathbf{P}^{-1}_{k|k-1}\mathbf{P}_{k,k-1|k-1} \tag{F.146}$$

using Eq. (F.188) we have :

$$\mathbf{P}_{k,k-1|N} = \mathbf{P}_{k,k-1|k} + \big[\mathbf{P}_{k|N} - \mathbf{P}_{k|k}\big]\mathbf{P}^{-1}_{k|k}\mathbf{P}_{k,k-1|k} \tag{F.147}$$

Estimation of $\left[\mathbf{x}_{k-1} - \widehat{\mathbf{x}}_{k-1|N}\right]$

by Kalman smoothing, we have:

$$\widehat{\mathbf{x}}_{k-1|N} = \widehat{\mathbf{x}}_{k-1|k-1} + \mathbf{J}_{k-1}\left[\widehat{\mathbf{x}}_{k|N} - \widehat{\mathbf{x}}_{k|k-1}\right] \tag{F.148}$$

with the smoothing Kalman gain:

$$\mathbf{J}_{k-1} = \mathbf{P}_{k-1|k-1}\mathbf{A}^T\mathbf{P}_{k|k-1}^T \tag{F.149}$$

using Eq. (F.164) :

$$\mathbf{J}_{k-1} = \mathbf{P}_{k,k-1|k-1}^T\mathbf{P}_{k|k-1}^{-1} \tag{F.150}$$

hence :

$$\widehat{\mathbf{x}}_{k-1|N} = \widehat{\mathbf{x}}_{k-1|k-1} + \mathbf{P}_{k,k-1|k-1}^T\mathbf{P}_{k|k-1}^{-1}\left[\widehat{\mathbf{x}}_{k|N} - \widehat{\mathbf{x}}_{k|k-1}\right] \tag{F.151}$$

hence :

$$\begin{aligned}&\left[\mathbf{x}_{k-1} - \widehat{\mathbf{x}}_{k-1|N}\right]\\ &\quad= \mathbf{x}_{k-1} - \widehat{\mathbf{x}}_{k-1|k-1} - \mathbf{P}_{k,k-1|k-1}^T\mathbf{P}_{k|k-1}^{-1}\left[\widehat{\mathbf{x}}_{k|N} - \widehat{\mathbf{x}}_{k|k-1}\right]\end{aligned} \tag{F.152}$$

using Eq. (F.172) :

$$\widehat{\mathbf{x}}_{k-1|k-1} = \widehat{\mathbf{x}}_{k-1|k} - \mathbf{P}_{k,k-1|k-1}^T\mathbf{C}^T\mathbf{\Sigma}_{k|k-1}^{-1}\mathbf{e}_k \tag{F.153}$$

so we find :

$$\begin{aligned}\left[\mathbf{x}_{k-1} - \widehat{\mathbf{x}}_{k-1|N}\right] &= \mathbf{x}_{k-1} - \widehat{\mathbf{x}}_{k-1|k} + \mathbf{P}_{k,k-1|k-1}^T\mathbf{C}^T\mathbf{\Sigma}_{k|k-1}^{-1}\mathbf{e}_k\\ &\quad - \mathbf{P}_{k,k-1|k-1}^T\mathbf{P}_{k|k-1}^{-1}\left[\widehat{\mathbf{x}}_{k|N} - \widehat{\mathbf{x}}_{k|k-1}\right]\end{aligned} \tag{F.154}$$

$$\begin{aligned}\left[\mathbf{x}_{k-1} - \widehat{\mathbf{x}}_{k-1|N}\right] &= \mathbf{x}_{k-1} - \widehat{\mathbf{x}}_{k-1|k}\\ &\quad + \mathbf{P}_{k,k-1|k-1}^T\mathbf{P}_{k|k-1}^{-1}\left[\widehat{\mathbf{x}}_{k|k-1} - \widehat{\mathbf{x}}_{k|N} + \mathbf{P}_{k|k-1}\mathbf{C}^T\mathbf{\Sigma}_{k|k-1}^{-1}\mathbf{e}_k\right]\end{aligned} \tag{F.155}$$

but :

$$\mathbf{K}_k = \mathbf{P}_{k|k-1}\mathbf{C}^T\mathbf{\Sigma}_{k|k-1}^{-1} \tag{F.156}$$

$$\widehat{\mathbf{x}}_{k|k} = \widehat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k\mathbf{e}_k \tag{F.157}$$

then we have :

$$\left[\mathbf{x}_{k-1} - \widehat{\mathbf{x}}_{k-1|N}\right] = \left[\mathbf{x}_{k-1} - \widehat{\mathbf{x}}_{k-1|k}\right] + \mathbf{P}_{k,k-1|k-1}^T\mathbf{P}_{k|k-1}^{-1}\left[\widehat{\mathbf{x}}_{k|k} - \widehat{\mathbf{x}}_{k|N}\right] \tag{F.158}$$

Estimation of $\mathbf{J_{k-1}}$

we have the smoothing Kalman gain :

$$\mathbf{J}_{k-1} = \mathbf{P}_{k-1|k-1}\mathbf{A}^T\mathbf{P}_{k|k-1}^T \tag{F.159}$$

but :

$$\begin{aligned}\mathbf{P}_{k,k-1|k-1} &= \mathbb{E}\big\{\big[\mathbf{x}_k - \widehat{\mathbf{x}}_{k|k-1}\big]\big[\mathbf{x}_{k-1} - \widehat{\mathbf{x}}_{k-1|k-1}\big]^T|Y_{k-1}\big\}\\ \mathbf{P}_{k-1,k-1|k-1} &= \mathbb{E}\big\{\big[\mathbf{x}_{k-1} - \widehat{\mathbf{x}}_{k-1|k-1}\big]\big[\mathbf{x}_{k-1} - \widehat{\mathbf{x}}_{k-1|k-1}\big]^T|Y_{k-1}\big\}\end{aligned}$$

and :

$$\mathbf{x}_k = \mathbf{A}\mathbf{x}_{k-1} + \mathbf{w}_{k-1} \tag{F.160}$$

$$\widehat{\mathbf{x}}_{k|k-1} = \mathbf{A}\widehat{\mathbf{x}}_{k-1|k-1} \tag{F.161}$$

hence :

$$\begin{aligned}\mathbf{P}_{k,k-1|k-1} &= \mathbb{E}\big\{\big[\mathbf{A}\mathbf{x}_{k-1} - \mathbf{A}\widehat{\mathbf{x}}_{k-1|k-1} + \mathbf{w}_{k-1}\big]\big[\mathbf{x}_{k-1} - \widehat{\mathbf{x}}_{k-1|k-1}\big]^T|Y_{k-1}\big\}\\ &= \mathbf{A}\ \mathbb{E}\big\{\big[\mathbf{x}_{k-1} - \widehat{\mathbf{x}}_{k-1|k-1}\big]\big[\mathbf{x}_{k-1} - \widehat{\mathbf{x}}_{k-1|k-1}\big]^T|Y_{k-1}\big\}\\ &= \mathbf{A}\mathbf{P}_{k-1,k-1|k-1}\end{aligned} \tag{F.162}$$

hence :

$$\mathbf{P}_{k,k-1|k-1} = \mathbf{A}\mathbf{P}_{k-1|k-1} \tag{F.163}$$

and we have :

$$\mathbf{J}_{k-1} = \mathbf{P}_{k,k-1|k-1}^T\mathbf{P}_{k|k-1}^{-1} \tag{F.164}$$

Estimation of $\widehat{\mathbf{x}}_{\boldsymbol{k-1|k-1}}$

we have :

$$\widehat{\mathbf{x}}_{k-1|k} = \widehat{\mathbf{x}}_{k-1|k-1} + \mathbf{J}_{k-1}\big[\widehat{\mathbf{x}}_{k|k} - \widehat{\mathbf{x}}_{k|k-1}\big] \tag{F.165}$$

$$\mathbf{J}_{k-1} = \mathbf{P}_{k,k-1|k-1}^T\mathbf{P}_{k|k-1}^{-1} \tag{F.166}$$

so :

$$\widehat{\mathbf{x}}_{k-1|k} = \widehat{\mathbf{x}}_{k-1|k-1} + \mathbf{P}_{k,k-1|k-1}^T\mathbf{P}_{k|k-1}^{-1}\big[\widehat{\mathbf{x}}_{k|k} - \widehat{\mathbf{x}}_{k|k-1}\big] \tag{F.167}$$

hence :

$$\widehat{\mathbf{x}}_{k-1|k-1} = \widehat{\mathbf{x}}_{k-1|k} - \mathbf{P}^T_{k,k-1|k-1}\mathbf{P}^{-1}_{k|k-1}\big[\widehat{\mathbf{x}}_{k|k} - \widehat{\mathbf{x}}_{k|k-1}\big] \tag{F.168}$$

but we have :

$$\widehat{\mathbf{x}}_{k|k} = \widehat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k\mathbf{e}_k \tag{F.169}$$
$$\mathbf{K}_k = \mathbf{P}_{k|k-1}\mathbf{C}^T\mathbf{\Sigma}^{-1}_{k|k-1} \tag{F.170}$$

then we find :

$$\widehat{\mathbf{x}}_{k-1|k-1} = \widehat{\mathbf{x}}_{k-1|k} - \mathbf{P}^T_{k,k-1|k-1}\mathbf{P}^{-1}_{k|k-1}\mathbf{K}_k\mathbf{e}_k \tag{F.171}$$

and we have :

$$\widehat{\mathbf{x}}_{k-1|k-1} = \widehat{\mathbf{x}}_{k-1|k} - \mathbf{P}^T_{k,k-1|k-1}\mathbf{C}^T\mathbf{\Sigma}^{-1}_{k|k-1}\mathbf{e}_k \tag{F.172}$$

Estimation of $\mathbb{E}\big\{[\mathbf{x}_{\boldsymbol{k}} - \widehat{\mathbf{x}}_{\boldsymbol{k|k}}][\widehat{\mathbf{x}}_{\boldsymbol{k|k}} - \widehat{\mathbf{x}}_{\boldsymbol{k|N}}]^{\boldsymbol{T}}\big\}$

we have :

$$\big[\mathbf{x}_k - \widehat{\mathbf{x}}_{k|N}\big] = \big[\mathbf{x}_k - \widehat{\mathbf{x}}_{k|k}\big] - \big[\widehat{\mathbf{x}}_{k|N} - \widehat{\mathbf{x}}_{k|k}\big] \tag{F.173}$$
$$\big[\widehat{\mathbf{x}}_{k|k} - \widehat{\mathbf{x}}_{k|N}\big] = \big[\mathbf{x}_k - \widehat{\mathbf{x}}_{k|N}\big] - \big[\mathbf{x}_k - \widehat{\mathbf{x}}_{k|k}\big] \tag{F.174}$$
$$\big[\mathbf{x}_k - \widehat{\mathbf{x}}_{k|k}\big] = \big[\mathbf{x}_k - \widehat{\mathbf{x}}_{k|N}\big] + \big[\widehat{\mathbf{x}}_{k|N} - \widehat{\mathbf{x}}_{k|k}\big] \tag{F.175}$$

then we find :

$$\begin{aligned}
&\mathbb{E}\big\{\big[\mathbf{x}_k - \widehat{\mathbf{x}}_{k|k}\big]\big[\widehat{\mathbf{x}}_{k|k} - \widehat{\mathbf{x}}_{k|N}\big]^T\big\} \\
&\quad= \mathbb{E}\big\{\big[\mathbf{x}_k - \widehat{\mathbf{x}}_{k|k}\big]\big[\mathbf{x}_k - \widehat{\mathbf{x}}_{k|N}\big]^T\big\} - \big\{\big[\mathbf{x}_k - \widehat{\mathbf{x}}_{k|k}\big]\big[\mathbf{x}_k - \widehat{\mathbf{x}}_{k|N}\big]^T\big\} \\
&\quad= \mathbb{E}\big\{\big[\mathbf{x}_k - \widehat{\mathbf{x}}_{k|N}\big]\big[\mathbf{x}_k - \widehat{\mathbf{x}}_{k|N}\big]^T\big\} - \mathbb{E}\big\{\big[\mathbf{x}_k - \widehat{\mathbf{x}}_{k|k}\big]\big[\mathbf{x}_k - \widehat{\mathbf{x}}_{k|k}\big]^T\big\} \\
&\quad+\mathbb{E}\big\{\big[\widehat{\mathbf{x}}_{k|N} - \widehat{\mathbf{x}}_{k|k}\big]\big[\mathbf{x}_k - \widehat{\mathbf{x}}_{k|N}\big]^T\big\} \\
&\quad= \mathbf{P}_{k|N} - \mathbf{P}_{k|k} - \mathbb{E}\big\{\big[\widehat{\mathbf{x}}_{k|N} - \widehat{\mathbf{x}}_{k|k}\big]\big[\mathbf{x}_k - \widehat{\mathbf{x}}_{k|N}\big]^T\big\}
\end{aligned} \tag{F.176}$$

but $\big[\widehat{\mathbf{x}}_{k|N} - \widehat{\mathbf{x}}_{k|k}\big]$ only depends on innovations at time points $\big\{k+1, \cdots, N\big\}$ then $\big[\widehat{\mathbf{x}}_{k|N} - \widehat{\mathbf{x}}_{k|k}\big] \perp \big[\mathbf{x}_k - \widehat{\mathbf{x}}_{k|N}\big]$ and we only have:

$$\mathbb{E}\big\{\big[\mathbf{x}_k - \widehat{\mathbf{x}}_{k|k}\big]\big[\mathbf{x}_{k|k} - \widehat{\mathbf{x}}_{k|N}\big]^T\big\} = \mathbf{P}_{k|N} - \mathbf{P}_{k|k} \tag{F.177}$$

Estimation of $\mathbf{P}^{-1}_{\boldsymbol{k|k-1}}\mathbf{P}_{\boldsymbol{k,k-1|k-1}}$

we have :

$$\mathbf{P}_{k|k} = \big[\mathbf{I} - \mathbf{K}_k\mathbf{C}\big]\mathbf{P}_{k|k-1} \tag{F.178}$$
$$\mathbf{P}_{k|k-1} = \big[\mathbf{I} - \mathbf{K}_k\mathbf{C}\big]^{-1}\mathbf{P}_{k|k} \tag{F.179}$$
$$\mathbf{P}^{-1}_{k|k-1} = \mathbf{P}^{-1}_{k|k}\big[\mathbf{I} - \mathbf{K}_k\mathbf{C}\big] \tag{F.180}$$
$$\mathbf{K}_k = \mathbf{P}_{k|k}\mathbf{C}^T\mathbf{R}^{-1} \tag{F.181}$$

so :

$$\mathbf{P}^{-1}_{k|k-1} = \mathbf{P}^{-1}_{k|k} - \mathbf{C}^T\mathbf{R}^{-1}\mathbf{C} \tag{F.182}$$

we have :

$$\mathbf{P}_{k,k-1|k} = \mathbf{P}_{k,k-1|k-1} - \mathbf{K}_k\mathbf{C}\mathbf{P}_{k,k-1|k-1} \tag{F.183}$$
$$\mathbf{P}_{k,k-1|k-1} = \mathbf{P}_{k,k-1|k} + \mathbf{K}_k\mathbf{C}\mathbf{P}_{k,k-1|k-1} \tag{F.184}$$

then :

$$\begin{aligned}\mathbf{P}^{-1}_{k|k-1}\mathbf{P}_{k,k-1|k-1} &= \big[\mathbf{P}^{-1}_{k|k} - \mathbf{C}^T\mathbf{R}^{-1}\mathbf{C}\big]\big[\mathbf{P}_{k,k-1|k} + \mathbf{K}_k\mathbf{C}\mathbf{P}_{k,k-1|k-1}\big] \\ &= \mathbf{P}^{-1}_{k|k}\mathbf{P}_{k,k-1|k} - \mathbf{C}^T\mathbf{R}^{-1}\mathbf{C}\mathbf{P}_{k,k-1|k} \\ &\quad + \mathbf{P}^{-1}_{k|k}\mathbf{K}_k\mathbf{C}\mathbf{P}_{k,k-1|k-1} - \mathbf{C}^T\mathbf{R}^{-1}\mathbf{C}\mathbf{K}_k\mathbf{C}\mathbf{P}_{k,k-1|k-1} \\ &= \mathbf{P}^{-1}_{k|k}\mathbf{P}_{k,k-1|k} + \mathbf{P}^{-1}_{k|k}\mathbf{K}_k\mathbf{C}\mathbf{P}_{k,k-1|k-1} \\ &\quad - \mathbf{C}^T\mathbf{R}^{-1}\mathbf{C}\big[\mathbf{P}_{k,k-1|k} + \mathbf{K}_k\mathbf{C}\mathbf{P}_{k,k-1|k-1}\big]\end{aligned} \tag{F.185}$$

using Eq. (F.183) we have :

$$\begin{aligned}\mathbf{P}^{-1}_{k|k-1}P_{k,k-1|k-1} &= \mathbf{P}^{-1}_{k|k}\mathbf{P}_{k,k-1|k} + \mathbf{P}^{-1}_{k|k}\mathbf{K}_k\mathbf{C}\mathbf{P}_{k,k-1|k-1} \\ &\quad - \mathbf{C}^T\mathbf{R}^{-1}\mathbf{C}\mathbf{P}_{k,k-1|k-1} \\ &= \mathbf{P}^{-1}_{k|k}\mathbf{P}_{k,k-1|k} + \big[\mathbf{P}^{-1}_{k|k}\mathbf{K}_k\mathbf{C} - \mathbf{C}^T\mathbf{R}^{-1}\mathbf{C}\big]\mathbf{P}_{k,k-1|k-1}\end{aligned} \tag{F.186}$$

using Eq. (F.181) we have :

$$\mathbf{P}^{-1}_{k|k-1}\mathbf{P}_{k,k-1|k-1} = \mathbf{P}^{-1}_{k|k}\mathbf{P}_{k,k-1|k} + \big[\mathbf{C}^T\mathbf{R}^{-1}\mathbf{C} - \mathbf{C}^T\mathbf{R}^{-1}\mathbf{C}\big]\mathbf{P}_{k,k-1|k-1} \tag{F.187}$$

so we find :

$$\mathbf{P}^{-1}_{k|k-1}\mathbf{P}_{k,k-1|k-1} = \mathbf{P}^{-1}_{k|k}\mathbf{P}_{k,k-1|k} \tag{F.188}$$

F.4.2 M-step

In the M-step, we find the parameter estimation from the sufficient statistics given by the E-step, as simple multivariate regression coefficients.

Let consider the sufficient statistics:

$$\Gamma_1 = \frac{1}{N+1}\sum_{k=0}^{N}\left[\mathbf{P}_{k|N} + \widehat{\mathbf{x}}_{k|N}\widehat{\mathbf{x}}_{k|N}^T\right] \tag{F.189}$$

$$\Gamma_2 = \frac{1}{N}\sum_{k=1}^{N}\left[\mathbf{P}_{k|N} + \widehat{\mathbf{x}}_{k|N}\widehat{\mathbf{x}}_{k|N}^T\right] \tag{F.190}$$

$$\Gamma_3 = \frac{1}{N}\sum_{k=1}^{N}\left[\mathbf{P}_{k-1|N} + \widehat{\mathbf{x}}_{k-1|N}\widehat{\mathbf{x}}_{k-1|N}^T\right] \tag{F.191}$$

$$\Gamma_4 = \frac{1}{N}\sum_{k=1}^{N}\left[\mathbf{P}_{k,k-1|N} + \widehat{\mathbf{x}}_{k|N}\widehat{\mathbf{x}}_{k-1|N}^T\right] \tag{F.192}$$

$$\Gamma_5 = \frac{1}{N+1}\sum_{k=0}^{N}\left[\mathbf{y}_k\mathbf{y}_k^T\right] \tag{F.193}$$

$$\Gamma_6 = \frac{1}{N+1}\sum_{k=0}^{N}\left[\mathbf{y}_k\widehat{\mathbf{x}}_{k|N}^T\right] \tag{F.194}$$

and we find the parameter estimations:

$$\widehat{\mathbf{A}} = \Gamma_4\Gamma_3^{-1} \tag{F.195}$$

$$\widehat{\mathbf{C}} = \Gamma_6\Gamma_1^{-1} \tag{F.196}$$

$$\widehat{\mathbf{Q}} = \Gamma_2 - \widehat{\mathbf{A}}\Gamma_4^T \tag{F.197}$$

$$\widehat{\mathbf{R}} = \Gamma_5 - \widehat{\mathbf{C}}\Gamma_6^T \tag{F.198}$$

F.5 State Estimation by Kalman Smoother

We have identified the set of parameters in the learning phase $\{\widehat{\mathbf{A}}, \widehat{\mathbf{C}}, \widehat{\mathbf{Q}}, \widehat{\mathbf{R}}\}$, now we can use the Kalman filter in filtering, prediction, and smoothing to get the states in the inference phase.

F.5.1 Initialisation

$$\widehat{\mathbf{x}}_{0|-1} = E\{\mathbf{x}_0\} = \overline{\mathbf{x}_0} \tag{F.199}$$

$$\mathbf{P}_{0|-1} = Var\{\mathbf{x}_0\} \tag{F.200}$$

F.5.2 Filtering

for $k = 1, 2, \cdots, N$, (Forwards)

Prediction

$$\widehat{\mathbf{x}}_{k|k-1} = \mathbf{A}\widehat{\mathbf{x}}_{k-1|k-1} \tag{F.201}$$
$$\mathbf{P}_{k|k-1} = \mathbf{A}\mathbf{P}_{k-1|k-1}\mathbf{A}^T + \mathbf{Q} \tag{F.202}$$

Update

$$\mathbf{\Sigma}_{k|k-1} = \mathbf{C}\mathbf{P}_{k|k-1}\mathbf{C}^T + \mathbf{R} \tag{F.203}$$
$$\mathbf{K}_k = \mathbf{P}_{k|k-1}\mathbf{C}^T\mathbf{\Sigma}_{k|k-1}^{-1} \tag{F.204}$$
$$\mathbf{e}_k = \mathbf{y}_k - \mathbf{C}\widehat{\mathbf{x}}_{k|k-1} \tag{F.205}$$
$$\widehat{\mathbf{x}}_{k|k} = \widehat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k\mathbf{e}_k \tag{F.206}$$
$$\mathbf{P}_{k|k} = \mathbf{P}_{k|k-1} - \mathbf{K}_k\mathbf{C}\mathbf{P}_{k|k-1} \tag{F.207}$$

F.5.3 Prediction

$$\widehat{\mathbf{x}}_{k+1|k} = \left[\mathbf{A}_k - \mathbf{A}_k\mathbf{K}_k\mathbf{C}_k^T\right] + \mathbf{K}_k\mathbf{y}_k \tag{F.208}$$
$$\mathbf{P}_{k+1|k} = \mathbf{A}_k\mathbf{P}_{k|k}\mathbf{A}_k^T + \mathbf{Q}_k \tag{F.209}$$

F.5.4 Smoothing

for $N > k$ (Backwards)

$$\mathbf{x}_{k-1|N} = \widehat{\mathbf{x}}_{k-1|k-1} + \mathbf{J}_{k-1}\left[\widehat{\mathbf{x}}_{k|N} - \widehat{\mathbf{x}}_{k|k-1}\right] \tag{F.210}$$
$$\mathbf{P}_{k-1|N} = \mathbf{P}_{k-1|k-1} + \mathbf{J}_{k-1}\left[\mathbf{P}_{k|N} - \mathbf{P}_{k|k-1}\right]\mathbf{J}_{k-1}^T \tag{F.211}$$
$$\mathbf{J}_{k-1} = \mathbf{P}_{k-1|k-1}\mathbf{A}_{k-1}^T\mathbf{P}_{k|k-1}^{-1} \tag{F.212}$$

At the end of this procedure, we have the set of states

$$\{\mathbf{x}_{0|N}, \mathbf{x}_{1|N}, \mathbf{x}_{2|N}, \cdots, \mathbf{x}_{N-1|N}, \mathbf{x}_{N|N}\}$$

We can also realize an estimation of the initial state and its covariance in order to improve the M-step.

F.5.5 Estimation of $\overline{\mathbf{x}}_\mathbf{0}$

We have :

$$\frac{\partial J}{\partial \overline{\mathbf{x}}_\mathbf{0}} = 0 \tag{F.213}$$

then:

$$\begin{aligned}\frac{\partial J}{\partial \overline{\mathbf{x}}_\mathbf{0}} &= \frac{\partial}{\partial \overline{\mathbf{x}}_\mathbf{0}} tr\Big[\mathbf{P}_0^{-1}[\mathbf{P}_{0|N} + (\widehat{\mathbf{x}}_{0|N} - \overline{\mathbf{x}}_\mathbf{0})(\widehat{\mathbf{x}}_{0|N} - \overline{\mathbf{x}}_\mathbf{0})^T]\Big] \\ &= \mathbf{P}_0^{-1}[-2\widehat{\mathbf{x}}_{0|N} + 2\overline{\mathbf{x}}_\mathbf{0}]\end{aligned} \tag{F.214}$$

let :

$$\widehat{\overline{\mathbf{x}}_\mathbf{0}} = \widehat{\mathbf{x}}_{0|N} \tag{F.215}$$

F.5.6 Estimation of $\mathbf{P_0}$

we have :

$$\frac{\partial J}{\partial \mathbf{P}_0^{-1}} = 0 \tag{F.216}$$

so we find :

$$\frac{\partial J}{\partial \mathbf{P}_0^{-1}} = \frac{\partial}{\partial \mathbf{P}_0^{-1}}\Big[\log||P_0|| + tr\big[\mathbf{P}_0^{-1}[\mathbf{P}_{0|N} + (\widehat{\mathbf{x}}_{0|N} - \overline{\mathbf{x}}_\mathbf{0})(\widehat{\mathbf{x}}_{0|N} - \overline{\mathbf{x}}_\mathbf{0})^T]\big]\Big] \tag{F.217}$$

and we have :

$$\widehat{\mathbf{P}}_0 = \mathbf{P}_{0|N} \tag{F.218}$$

F.6 Conclusion

The learning procedure, gives the parameter estimations:

$$\widehat{\mathbf{A}} = \Gamma_4\Gamma_3^{-1} \tag{F.219}$$
$$\widehat{\mathbf{C}} = \Gamma_6\Gamma_1^{-1} \tag{F.220}$$
$$\widehat{\mathbf{Q}} = \Gamma_2 - \widehat{\mathbf{A}}\Gamma_4^T \tag{F.221}$$
$$\widehat{\mathbf{R}} = \Gamma_5 - \widehat{\mathbf{C}}\Gamma_6^T \tag{F.222}$$
$$\widehat{\overline{\mathbf{x}}_\mathbf{0}} = \widehat{\mathbf{x}}_{0|N} \tag{F.223}$$
$$\widehat{\mathbf{P}}_0 = \mathbf{P}_{0|N} \tag{F.224}$$

The inference procedure gives the states:

$$\{\mathbf{x}_{0|N}, \mathbf{x}_{1|N}, \mathbf{x}_{2|N}, \cdots, \mathbf{x}_{N-1|N}, \mathbf{x}_{N|N}\}$$

G

Parameter Estimation of a Non Linear Model by EM algorithm.

In this appendix, the parameter estimation of a nonlinear state Space Model is realized by EM algorithm. In Section [5.2] this procedure was used for initial parameter estimation, and sketched without derivation. Derivations for these expressions are provided below.

G.1 Introduction

Let consider a non stationary, nonlinear Discrete Time Model in a Dynamic State Space representation:

$$\mathbf{x}_{k+1} = \mathbf{f}_k(\mathbf{x}_k) + \mathbf{v}_k \tag{G.1}$$

$$\mathbf{y}_k = \mathbf{h}_k(\mathbf{x}_k) + \mathbf{n}_k, \tag{G.2}$$

with non stationary, nonlinear RBF functions in State and Measurement equations:

$$\mathbf{f}_k(\mathbf{x}_k) = \sum_{i=1}^{I} \lambda_{ik}^{(f)} \exp\big[-\frac{1}{\sigma_{ik}^{(f)}} (\mathbf{x}_k - \mathbf{c}_{ik}^{(f)})^T (\mathbf{x}_k - \mathbf{c}_{ik}^{(f)}) \big] \tag{G.3}$$

$$\mathbf{h}_k(\mathbf{x}_k) = \sum_{j=1}^{J} \lambda_{jk}^{(h)} \exp\big[-\frac{1}{\sigma_{jk}^{(h)}} (\mathbf{x}_k - \mathbf{c}_{jk}^{(h)})^T (\mathbf{x}_k - \mathbf{c}_{jk}^{(h)}) \big]. \tag{G.4}$$

where

- $\mathbf{x}_k \in R^n$, $\mathbf{y}_k \in R^q$, $\mathbf{v}_k \in R^n$, $\mathbf{n}_k \in R^q$,
- $\mathbf{x}_k$ is the state, without economic signification,
- $\mathbf{y}_k$ is the observation, the spline coefficient vector of the smoothing of past raw three hours data,
- Eq. (G.1) is the state equation,

- Eq. (G.2) is the measurement equation,
- $\mathbf{v}_k$ is the process noise, with $\mathbf{v}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{Q})$,
- $\mathbf{n}_k$ is the measurement noise, with $\mathbf{n}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{R})$.
- $\mathbf{x}_0 \sim N(\overline{\mathbf{x}_0}, \mathbf{P}_0)$,
- I is the number of neurons in the hidden layer for function $\mathbf{f}_k$,
- $\mathbf{c}_{ik}^{(f)}$) is the centroid vector for neuron i,
- $\lambda_{ik}^{(f)}$ are the weights of the linear output neuron,
- $\sigma_{ik}^{(f)}$ are the variances of clusters,
- J is the number of neurons in the hidden layer for function $\mathbf{h}_k$,
- $\mathbf{c}_{jk}^{(h)}$) is the centroid vector for neuron i,
- $\lambda_{jk}^{(h)}$ are the weights of the linear output neuron,
- $\sigma_{jk}^{(h)}$ are the variances of clusters.

In each Training, Validation, and Test phase, to realize the stopping time prediction, we need a state forecasting carries out an inference step. This inference step comes after a parameter estimation in a learning step.

We must estimate two sets of parameters for non stationary processes, such as:

- estimation of : $\{\widehat{\lambda}_{ik}^{(f)},\ \widehat{\sigma}_{ik}^{(f)},\ \mathbf{c}_{ik}^{(f)}\}$ for function $\mathbf{f}_k(\mathbf{x}_k)$,
- estimation of : $\{\widehat{\lambda}_{jk}^{(h)},\ \widehat{\sigma}_{jk}^{(h)},\ \mathbf{c}_{jk}^{(h)}\}$ for function $\mathbf{h}_k(\mathbf{x}_k)$.

But, for parameter estimation we need two sets of input and output data:

- for parameter estimation of $\mathbf{f}_k(\mathbf{x}_k)$, we need :
  - input data $\{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_{k-1}, \cdots, \mathbf{x}_{N-1}\}$
  - output data $\{\mathbf{x}_2, \mathbf{x}_3, \cdots, \mathbf{x}_k, \cdots, \mathbf{x}_N\}$
- for parameter estimation of $\mathbf{h}_k(\mathbf{x}_k)$, we need:
  - input data $\{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_{k-1}, \cdots, \mathbf{x}_N\}$
  - output data $\{\mathbf{y}_1, \mathbf{y}_2, \cdots, \mathbf{y}_{k-1}, \cdots, \mathbf{y}_N\}$

We know the measurements $\{\mathbf{y}_1, \mathbf{y}_2, \cdots, \mathbf{y}_{k-1}, \cdots, \mathbf{y}_N\}$, but we do not know the states
$\{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_{k-1}, \cdots, \mathbf{x}_N\}$. Thus, we cannot directly realize a parameter estimation.

We could use a *dual filtering* procedure to estimate, in a recursive procedure, parameters and states. But when the number of parameters and state dimension are high, this procedure does not stabilize.

To cope with this problem, we realize an initial parameter estimation with EM-algorithm in a "*learning*" procedure, and a state estimation by Kalman smoothers in an "*inference*" procedure, into separate steps as follow:

- Learning procedure for stationary, linearized model,
  - we consider a stationary, nonlinear system,
  - we realize a linearization of equations,
  - we use the EM-algorithm to estimate the parameters of the linearized model,
- Inference procedure for stationary, linearized model,
  - we use a Kalman smoother for state estimation, using the parameters of the linearized model,
- Learning procedure for stationary, nonlinear model,
  - if we have a small Δt, the state behaviors of nonlinear and linearized models are similar, and we can use the state values coming from the linearized model as input for the nonlinear model, in order to realize its parameter estimation,

Afterwards, we use these initial parameters for the non stationary, nonlinear model, with $\mathbf{f}_k$, and $\mathbf{h}_k$ functions, in a joint filter procedure that gives a state prediction and an adaptive parameter estimation in the Training, Validation and Test phases..

G.2 Parameter Estimation

In this section we give a detailed description of the method for initial parameter estimation of a dynamic state space model in discrete time.

G.2.1 Stationary nonlinear model

We consider a stationary, nonlinear State Space Model with Radial Basis Functions (RBF) in state and measurement equations,

$$\mathbf{x}_{k+1} = \mathbf{f}(\mathbf{x}_k) + \mathbf{v}_k \tag{G.5}$$

$$\mathbf{y}_k = \mathbf{h}(\mathbf{x}_k) + \mathbf{n}_k, \tag{G.6}$$

with nonlinear RBF functions:

$$\mathbf{f}(\mathbf{x}_k) = \sum_{i=1}^{I} \lambda_i^{(f)} \exp\big[-\frac{1}{\sigma_i^{(f)}} (\mathbf{x}_k - \mathbf{c}_i^{(f)})^T (\mathbf{x}_k - \mathbf{c}_i^{(f)}) \big] \tag{G.7}$$

$$\mathbf{h}(\mathbf{x}_k) = \sum_{j=1}^{J} \lambda_j^{(h)} \exp\big[-\frac{1}{\sigma_j^{(h)}} (\mathbf{x}_k - \mathbf{c}_j^{(h)})^T (\mathbf{x}_k - \mathbf{c}_j^{(h)}) \big]. \tag{G.8}$$

We have to estimate the hyperparameters θ:

- $\widehat{\lambda}_j^{(f)}$, $\widehat{\sigma}_j^{(f)}$, $\mathbf{c}_i^{(f)}$ for state equation $\mathbf{f}(\mathbf{x}_k)$
- $\widehat{\lambda}_j^{(f)}$, $\widehat{\sigma}_j^{(f)}$, $\mathbf{c}_i^{(f)}$ for measurement equation $\mathbf{h}(\mathbf{x}_k)$.

We can estimate the parameters θ by the Expectation-Maximization (EM) algorithm. The EM algorithm is an iterative method for finding a mode of the likelihood function $p(\theta|\mathbf{y}_{1:N})$, where $\{\mathbf{y}_{1:N}\}$ represents the set of observations, as follow:

- Expectation-step : we estimate the states given an estimation of parameters,
- Maximization-step : we estimate new parameters given the states,
- re-estimate the states with the new set of parameters,
- and so forth.

EM is particularly useful when models can be re-expressed in augmented parameter spaces, when the extra parameter $\mathbf{x}$ can be thought of as missing data, in situation where it is hard to maximise $p(\theta|\mathbf{y}_{1:N})$. EM will allow us to accomplish this by working with $p(\theta|\mathbf{x}_{0:N}, \mathbf{y}_{1:N})$.

G.2.2 Linearization of the stationary model

We realize the linearization of the model as follow:

$$\mathbf{f}(\mathbf{x}_k) \simeq \mathbf{f}(\widehat{\mathbf{x}}_{k|N}) + \mathbf{A}\big[\mathbf{x}_k - \widehat{\mathbf{x}}_{k|N}\big] \tag{G.9}$$

$$\mathbf{A} = \Big[\frac{\partial \mathbf{f}(\mathbf{x}_k)}{\partial \mathbf{x}_k}\Big]_{\widehat{\mathbf{x}}_{k|N}} \tag{G.10}$$

$$= -2\sum_{i=1}^{I} \frac{\lambda_i^{(f)}}{\sigma_i^{(f)}}[\widehat{\mathbf{x}}_{k|N} - \mathbf{c}_i^{(f)}] \exp\Big[-\frac{1}{\sigma_i^{(f)}}(\widehat{\mathbf{x}}_{k|N} - \mathbf{c}_i^{(f)})(\widehat{\mathbf{x}}_{k|N} - \mathbf{c}_i^{(f)})^T\Big] \tag{G.11}$$

$$\mathbf{h}(\mathbf{x}_k) \simeq \mathbf{h}(\widehat{\mathbf{x}}_{k|N}) + \mathbf{C}\big[\mathbf{x}_k - \widehat{\mathbf{x}}_{k|N}\big] \tag{G.12}$$

$$\mathbf{C} = \Big[\frac{\partial \mathbf{h}(\mathbf{x}_k)}{\partial \mathbf{x}_k}\Big]_{\widehat{\mathbf{x}}_{k|N}} \tag{G.13}$$

$$= -2\sum_{j=1}^{J} \frac{\lambda_j^{(h)}}{\sigma_j^{(h)}}[\widehat{\mathbf{x}}_{k|N} - \mathbf{c}_j^{(h)}] \exp\Big[-\frac{1}{\sigma_j^{(h)}}(\widehat{\mathbf{x}}_{k|N} - \mathbf{c}_j^{(h)})(\widehat{\mathbf{x}}_{k|N} - \mathbf{c}_j^{(h)})^T\Big], \tag{G.14}$$

where $\widehat{\mathbf{x}}_{k|N}$ is a state estimation after filtering and smoothing by Kalman filter, such as:

$$\widehat{\mathbf{x}}_{k|N} = \mathbb{E}[\mathbf{x}_k|\mathbf{y}_{1:N}] \qquad \text{for } N > k. \tag{G.15}$$

We get the linearized model:

$$\mathbf{x}_k = \mathbf{A}\mathbf{x}_{k-1} + \mathbf{v}_k + \mathbf{a}_k \tag{G.16}$$
$$\mathbf{y}_k = \mathbf{C}\mathbf{x}_k + \mathbf{n}_k + \mathbf{c}_k, \tag{G.17}$$

with "residues" of linearization:

$$\mathbf{a}_k = \mathbf{f}(\widehat{\mathbf{x}}_{k|k}) - \mathbf{A}\widehat{\mathbf{x}}_{k|k} \tag{G.18}$$
$$\mathbf{c}_k = \mathbf{h}(\widehat{\mathbf{x}}_{k|k-1}) - \mathbf{C}\widehat{\mathbf{x}}_{k|k-1}. \tag{G.19}$$

If Δt small, we have small "residues" of linearization, and states of nonlinear and linearized models are similar.

G.2.3 Parameter estimation of a stationary linear model

We consider a linear stationary process, and we realize a parameter initialization, as follow:

- Learning procedure for stationary linearized model:

  from likelihood of linearized model, by EM-algorithm in a batch procedure we estimate:
  - noises covariances matrixes $\mathbf{R}$ and $\mathbf{Q}$,
  - initial state $\mathbf{x}_0$ and its covariance $\mathbf{P}_0$,
  - $\mathbf{A}$ and $\mathbf{C}$ matrixes of the linearized system,

- Inference procedure for stationary linearized model:

  from the parameters estimated in the learning phase, in an inference phase, by Extended Kalman Smoothing we estimate the states $\{\mathbf{x}_{1|N}, \mathbf{x}_{2|N}, \cdots, \mathbf{x}_{N|N}\}$ of the linearized system,

We detailed these procedures in next sections.

Likelihood

To derive the EM algorithm we need to develop a probabilistic model for Eq. (G.5), and (G.6).

We assume the likelihood of the data given the states, initial conditions and evolution of states can be represented by Gaussian distributions.

Let suppose Gaussian distributions for state and measurement noises, than we have:

$$p(\mathbf{x}_0|\boldsymbol{\theta}) = \frac{1}{(2\pi)^{\frac{N}{2}}} \frac{1}{|\mathbf{P}_0|^{\frac{1}{2}}} \exp\left\{(\mathbf{x}_0 - \overline{\mathbf{x}}_0)^T \mathbf{P}_0^{-1} (\mathbf{x}_0 - \overline{\mathbf{x}}_0)\right\} \tag{G.20}$$

$$p(\mathbf{x}_k|\mathbf{x}_{k-1}|\boldsymbol{\theta}) = \frac{1}{(2\pi)^{\frac{N}{2}}} \frac{1}{|\mathbf{Q}|^{\frac{1}{2}}} \exp\left\{\left[\mathbf{x}_k - \widehat{\mathbf{f}}(\mathbf{x}_k)\right]^T \mathbf{Q}^{-1} \left[\mathbf{x}_k - \widehat{\mathbf{f}}(\mathbf{x}_k\right]\right\} \tag{G.21}$$

$$p(\mathbf{y}_k|\mathbf{x}_k|\boldsymbol{\theta}) = \frac{1}{(2\pi)^{\frac{N}{2}}} \frac{1}{|\mathbf{R}|^{\frac{1}{2}}} \exp\left\{\left[\mathbf{y}_k - \widehat{\mathbf{h}}(\mathbf{x}_k)\right]^T \mathbf{R}^{-1} \left[\mathbf{y}_k - \widehat{\mathbf{h}}(\mathbf{x}_k)\right]\right\}. \tag{G.22}$$

Under the initial model assumptions of uncorrelated noises, Markov assumption for states evolution, and independence of measurements given the state, the joint distribution is given by:

$$p(\mathbf{x}_{0:N}, \mathbf{y}_{1:N}|\boldsymbol{\theta}) = p(\mathbf{x}_0|\boldsymbol{\theta}) \prod_{k=1}^{N} p(\mathbf{x}_k|\mathbf{x}_{k-1}, \boldsymbol{\theta}) \prod_{k=0}^{N} p(\mathbf{y}_k|\mathbf{x}_k, \boldsymbol{\theta}), \tag{G.23}$$

and the log-likelihood of the complete data is given by:

$$\begin{aligned} J(\boldsymbol{\theta}|\mathbf{x}_{0:N}, \mathbf{y}_{1:N}) = & \\ & (\mathbf{x}_0 - \overline{\mathbf{x}}_0)^T \mathbf{P}_0^{-1} (\mathbf{x}_0 - \overline{\mathbf{x}}_0) + \log ||\mathbf{P}_0|| \\ + & \sum_{k=1}^{N} \left\{ \log \|\mathbf{Q}\| + (\mathbf{x}_k - \widehat{\mathbf{f}}(\mathbf{x}_k))^T \mathbf{Q}^{-1} (\mathbf{x}_k - \widehat{\mathbf{f}}(\mathbf{x}_k)) \right\} \\ + & \sum_{k=0}^{N} \left[\log \|\mathbf{R}\| + [\mathbf{y}_k - \widehat{\mathbf{h}}(\mathbf{x}_k)]^T \mathbf{R}^{-1} [\mathbf{y}_k - \widehat{\mathbf{h}}(\mathbf{x}_k)] \right] \end{aligned} \tag{G.24}$$

Learning phase

All we need to do is to compute the expectation of the log-likelihood and then differentiate the result with respect to the parameters so as to maximise the log-likelihood.

The EM algorithm involves computing the expected values of the states and covariances with the Extended Kalman Smoother and then estimating the parameters with the formulae obtained by differentiating the expected log-likelihood.

We maximize:

$$Q(\boldsymbol{\theta}^{(p+1)}|\boldsymbol{\theta}^{(p)}) = \mathbb{E}_{\boldsymbol{\theta}^{(p)}} \left\{ J\left(\boldsymbol{\theta}^{(p+1)}|\mathbf{x}_{0:N}, \mathbf{y}_{1:N}\right) \right\} \tag{G.25}$$

in an iterative procedure; where $\boldsymbol{\theta}$ is the set of hyperparameters $\{\mathbf{A}, \mathbf{C}, \mathbf{R}, \mathbf{Q}, \mathbf{x}_0, \mathbf{P}_0\}$.

- in the **Expectation-step** we estimate the states given a parameter estimation,
- in the **Maximization-step** we estimate new parameters given states.

We apply this algorithm to:

$$\begin{aligned}
&\mathbb{E}\big\{ J\big(\boldsymbol{\theta}^{(p+1)}|\mathbf{x}_{0:N}, \mathbf{y}_{1:N}\big)\big\} = \\
&\\
&(N+1)\log||\mathbf{R}|| + N\log||\mathbf{Q}|| + \log||\mathbf{P}_0|| \\
&+ \sum_{k=1}^{N} tr\Big[\mathbf{Q}^{-1}\big[(\widehat{\mathbf{x}}_{k|N} - \widehat{\mathbf{f}}(\widehat{\mathbf{x}}_{k|N}))(\widehat{\mathbf{x}}_{k|N} - \widehat{\mathbf{f}}(\widehat{\mathbf{x}}_{k|N}))^T + \mathbf{P}_{k|N} + \mathbf{A}\mathbf{P}_{k|N}\mathbf{A}^T\big]\Big] \\
&+ \sum_{k=0}^{N} tr\Big[\mathbf{R}^{-1}\big[(\mathbf{y}_k - \widehat{\mathbf{h}}(\widehat{\mathbf{x}}_{k|N}))(\mathbf{y}_k - \widehat{\mathbf{h}}(\widehat{\mathbf{x}}_{k|N}))^T + \mathbf{C}\mathbf{P}_{k|N}\mathbf{C}^T\big]\Big] \\
&+ tr\Big[\mathbf{P}_0^{-1}[\mathbf{P}_{0|N} + (\widehat{\mathbf{x}}_{0|N} - \overline{\mathbf{x}}_{\mathbf{0}})(\widehat{\mathbf{x}}_{0|N} - \overline{\mathbf{x}}_{\mathbf{0}})^T]\Big], \qquad \text{(G.26)}
\end{aligned}$$

to get the hyperparameters: $\{\mathbf{A}, \mathbf{C}, \mathbf{R}, \mathbf{Q}, \mathbf{x}_0, \mathbf{P}_0\}$.

This procedure is detailed in Appendix [F] for State and Parametric Estimations of a Linear Model by EM algorithm..

Algorithm G.2.1 (EM Algorithm III) .

E-step

At each iteration, we estimate a minimal sufficient statistics.

$$E_{\theta^{(p)}}\big\{\mathbf{y}_k\mathbf{y}_k^T|Y_{1:N}\big\} \qquad \text{(G.27)}$$
$$E_{\theta^{(p)}}\big\{\mathbf{y}_k\mathbf{x}_k^T|Y_{1:N}\big\} \qquad \text{(G.28)}$$
$$E_{\theta^{(p)}}\big\{\mathbf{x}_k\mathbf{x}_k^T|Y_{1:N}\big\} \qquad \text{(G.29)}$$
$$E_{\theta^{(p)}}\big\{\mathbf{x}_k\mathbf{x}_{k-1}^T|Y_{1:N}\big\} \qquad \text{(G.30)}$$

*Using **Kalman filtering**, (Forwards)*
for $k = 1, \cdots, N$

$$\boldsymbol{\Sigma}_{k|k-1} = \mathbf{C}\mathbf{P}_{k|k-1}\mathbf{C}^T + \mathbf{R} \tag{G.31}$$
$$\mathbf{K}_k = \mathbf{P}_{k|k-1}\mathbf{C}^T\boldsymbol{\Sigma}_{k|k-1}^{-1} \tag{G.32}$$
$$\mathbf{e}_k = \mathbf{y}_k - \mathbf{C}\widehat{\mathbf{x}}_{k|k-1} \tag{G.33}$$
$$\widehat{\mathbf{x}}_{k|k} = \widehat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k\mathbf{e}_k \tag{G.34}$$
$$\mathbf{P}_{k|k} = \mathbf{P}_{k|k-1} - \mathbf{K}_k\mathbf{C}\mathbf{P}_{k|k-1} \tag{G.35}$$
$$\widehat{\mathbf{x}}_{k+1|k} = \mathbf{A}\widehat{\mathbf{x}}_{k|k} \tag{G.36}$$
$$\mathbf{P}_{k+1|k} = \mathbf{A}\mathbf{P}_{k|k}\mathbf{A}^T + \mathbf{Q} \tag{G.37}$$
$$\mathbf{P}_{k,k-1|k} = \big[\mathbf{I} - \mathbf{K}_k\mathbf{C}\big]\mathbf{A}\mathbf{P}_{k-1|k-1} \tag{G.38}$$

*Using **Kalman smoothing**, (Backwards)*
for $k = N, \cdots, 1$

$$\widehat{\mathbf{x}}_{k-1|N} = \widehat{\mathbf{x}}_{k-1|k-1} + \mathbf{J}_{k-1}\big[\widehat{\mathbf{x}}_{k|N} - \widehat{\mathbf{x}}_{k|k-1}\big] \tag{G.39}$$
$$\mathbf{P}_{k-1|N} = \mathbf{P}_{k-1|k-1} + \mathbf{J}_{k-1}\big[\mathbf{P}_{k|N} - \mathbf{P}_{k|k-1}\big]\mathbf{J}_{k-1}^T \tag{G.40}$$
$$\mathbf{J}_{k-1} = \mathbf{P}_{k-1|k-1}\mathbf{A}^T\mathbf{P}_{k|k-1}^T \tag{G.41}$$
$$\mathbf{P}_{k,k-1|N} = \mathbf{P}_{k,k-1|k} + \big[\mathbf{P}_{k|N} - \mathbf{P}_{k|k}\big]\mathbf{P}_{k|k}^{-1}\mathbf{P}_{k,k-1|k} \tag{G.42}$$

To get the Sufficient Statistics

$$\Gamma_1 = \frac{1}{N+1}\sum_{k=0}^{N}\big[\mathbf{P}_{k|N} + \widehat{\mathbf{x}}_{k|N}\widehat{\mathbf{x}}_{k|N}^T\big] \tag{G.43}$$
$$\Gamma_2 = \frac{1}{N}\sum_{k=1}^{N}\big[\mathbf{P}_{k|N} + \widehat{\mathbf{x}}_{k|N}\widehat{\mathbf{x}}_{k|N}^T\big] \tag{G.44}$$
$$\Gamma_3 = \frac{1}{N}\sum_{k=1}^{N}\big[\mathbf{P}_{k-1|N} + \widehat{\mathbf{x}}_{k-1|N}\widehat{\mathbf{x}}_{k-1|N}^T\big] \tag{G.45}$$
$$\Gamma_4 = \frac{1}{N}\sum_{k=1}^{N}\big[\mathbf{P}_{k,k-1|N} + \widehat{\mathbf{x}}_{k|N}\widehat{\mathbf{x}}_{k-1|N}^T\big] \tag{G.46}$$

$$\Gamma_5 = \frac{1}{N+1}\sum_{k=0}^{N}\big[\mathbf{y}_k\mathbf{y}_k^T\big] \tag{G.47}$$
$$\Gamma_6 = \frac{1}{N+1}\sum_{k=0}^{N}\big[\mathbf{y}_k\widehat{\mathbf{x}}_{k|N}^T\big] \tag{G.48}$$

M-step
Estimating model parameters from sufficient statistics given by E-step, as coefficients

of regression.
We find

$$\widehat{\mathbf{A}} = \mathit{\Gamma}_4 \mathit{\Gamma}_3^{-1} \tag{G.49}$$
$$\widehat{\mathbf{C}} = \mathit{\Gamma}_6 \mathit{\Gamma}_1^{-1} \tag{G.50}$$
$$\widehat{\mathbf{Q}} = \mathit{\Gamma}_2 - \widehat{\mathbf{A}} \mathit{\Gamma}_4^T \tag{G.51}$$
$$\widehat{\mathbf{R}} = \mathit{\Gamma}_5 - \widehat{\mathbf{C}} \mathit{\Gamma}_6^T \tag{G.52}$$
$$\widehat{\mathbf{x_0}} = \widehat{\mathbf{x}}_{0|N} \tag{G.53}$$
$$\widehat{\mathbf{P}}_0 = \mathbf{P}_{0|N} \tag{G.54}$$

State Estimation

From the hyperparameters estimated by EM-algorithm, we use the Extended Kalman smoother to estimate the state values:

$$\{\mathbf{x}_{0|N}, \mathbf{x}_{1|N}, \cdots, \mathbf{x}_{N-1|N}, \mathbf{x}_{N|N}\}$$

where $\mathbf{x}_{k|N}$ represents the estimation of $\mathbf{x}_k$ at time step k, based on all past, present, and future observations:

$$\{\mathbf{y}_1, \mathbf{y}_2, \cdots, \mathbf{y}_{N-1}, \mathbf{y}_N\}$$

by forwards and backwards Kalman filter procedures, in order to improve state estimation.

Algorithm G.2.2 (Extended Kalman Smoother) .

Filtering *for $k = 1, 2, \cdots N$, (Forwards)*

Prediction

$$\mathbf{f}_k(\mathbf{x}_k) = \sum_{i=1}^{I} \lambda_i^{(f)} \exp\big[-\frac{1}{\sigma_i^{(f)}} (\mathbf{x}_k - \mathbf{c}_i^{(f)})^T (\mathbf{x}_k - \mathbf{c}_i^{(f)}) \big] \tag{G.55}$$
$$\widehat{\mathbf{x}}_{k|k-1} = \mathbf{f}_{k-1}(\widehat{\mathbf{x}}_{k-1|k-1}) \tag{G.56}$$
$$\mathbf{A}_{k-1} = \Big[\frac{\partial \mathbf{f}_k(\mathbf{x}_k)}{\partial \mathbf{x}_k} \Big]_{\widehat{\mathbf{x}}_{k-1|k-1}} \tag{G.57}$$
$$\mathbf{P}_{k|k-1} = \mathbf{A}_{k-1} \mathbf{P}_{k-1|k-1} \mathbf{A}_{k-1}^T + \mathbf{Q} \tag{G.58}$$

Update

$$\mathbf{h}_k(\mathbf{x}_k) = \sum_{i=1}^{I} \lambda_i^{(h)} \exp\big[-\frac{1}{\sigma_i^{(h)}}(\mathbf{x}_k - \mathbf{c}_i^{(h)})^T(\mathbf{x}_k - \mathbf{c}_i^{(h)})\big] \tag{G.59}$$

$$\mathbf{C}_k = \Big[\frac{\partial \mathbf{h}_k(\mathbf{x}_k)}{\partial \mathbf{x}_k}\Big]_{\widehat{\mathbf{x}}_{k|k-1}} \tag{G.60}$$

$$\boldsymbol{\Sigma}_{k|k-1} = \mathbf{C}_k \mathbf{P}_{k|k-1} \mathbf{C}_k^T + \mathbf{R} \tag{G.61}$$

$$\mathbf{K}_k = \mathbf{P}_{k|k-1} \mathbf{C}_k^T \boldsymbol{\Sigma}_{k|k-1}^{-1} \tag{G.62}$$

$$\mathbf{e}_k = \mathbf{y}_k - \mathbf{h}_k(\widehat{\mathbf{x}}_{k|k-1}) \tag{G.63}$$

$$\widehat{\mathbf{x}}_{k|k} = \widehat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k \mathbf{e}_k \tag{G.64}$$

$$\mathbf{P}_{k|k} = \mathbf{P}_{k|k-1} - \mathbf{K}_k \mathbf{C}_k \mathbf{P}_{k|k-1} \tag{G.65}$$

One-step ahead Prediction

$$\widehat{\mathbf{x}}_{k+1|k} = \big[\mathbf{A}_k - \mathbf{A}_k \mathbf{K}_k \mathbf{C}_k^T\big] + \mathbf{K}_k \mathbf{y}_k \tag{G.66}$$

$$\mathbf{P}_{k+1|k} = \mathbf{A}_k \mathbf{P}_{k|k} \mathbf{A}_k^T + \mathbf{Q}_k \tag{G.67}$$

Smoothing *for $N > k$ (Backwards)*

$$\mathbf{x}_{k-1|N} = \widehat{\mathbf{x}}_{k-1|k-1} + \mathbf{J}_{k-1}\big[\widehat{\mathbf{x}}_{k|N} - \widehat{\mathbf{x}}_{k|k-1}\big] \tag{G.68}$$

$$\mathbf{P}_{k-1|N} = \mathbf{P}_{k-1|k-1} + \mathbf{J}_{k-1}\big[\mathbf{P}_{k|N} - \mathbf{P}_{k|k-1}\big]\mathbf{J}_{k-1}^T \tag{G.69}$$

$$\mathbf{J}_{k-1} = \mathbf{P}_{k-1|k-1} A_{k-1}^T \mathbf{P}_{k|k-1}^{-1} \tag{G.70}$$

G.2.4 Parameter Estimation of a stationary nonlinear model

In previous sections we have realized a state estimation of a stationary linearized model.

We can use this state estimation $\{\mathbf{x}_{0|N}, \mathbf{x}_{1|N}, \cdots, \mathbf{x}_{N|N}\}$ and the measurement set $\{\mathbf{y}_1, \mathbf{y}_2, \cdots, \mathbf{y}_N\}$ to realize a parameter estimation of the stationary nonlinear model:

$$\mathbf{f}(\mathbf{x}_k) = \sum_{i=1}^{I} \lambda_i^{(f)} \exp\big[-\frac{1}{\sigma_i^{(f)}}(\mathbf{x}_k - \mathbf{c}_i^{(f)})^T(\mathbf{x}_k - \mathbf{c}_i^{(f)})\big] \tag{G.71}$$

$$\mathbf{h}(\mathbf{x}_k) = \sum_{j=1}^{J} \lambda_j^{(h)} \exp\big[-\frac{1}{\sigma_j^{(h)}}(\mathbf{x}_k - \mathbf{c}_j^{(h)})^T(\mathbf{x}_k - \mathbf{c}_j^{(h)})\big] \tag{G.72}$$

to get the parameter estimations:

- $\widehat{\lambda}_j^{(f)}, \widehat{\sigma}_j^{(f)}, \mathbf{c}_i^{(f)}$ for $\mathbf{f}(\mathbf{x}_k)$
- $\widehat{\lambda}_j^{(f)}, \widehat{\sigma}_j^{(f)}, \mathbf{c}_i^{(f)}$ for $\mathbf{h}(\mathbf{x}_k)$

by RBF methods.

Afterwards, we will use these parameter estimations as initial values in joint filtering procedure, for state estimation and adaptive parameter estimation, with Particle and Kalman filters, of a non stationary nonlinear system in Training, Validation and Test phase.

H

Algorithms

H.1 Functional Clustering

Algorithm H.1.1 (Functional Clustering by EM Algorithm) .

Let $g_i(t)$ the hidden true value for the curve i at time t and $\mathbf{g}_i$, $\mathbf{y}_i$ and $\boldsymbol{\epsilon}_i$, the random vectors of, respectively, hidden true values, measurements and errors. We have:

$$\boldsymbol{y}_i = \boldsymbol{g}_i + \boldsymbol{\epsilon}_i, \qquad i = 1, \cdots, N \ , \tag{H.1}$$

where N is the number of curves. The random errors $\boldsymbol{\epsilon}_i$ are assumed i.i.d., uncorrelated with each other and with $\boldsymbol{g}_i$.

For the trend forecasting, we use a functional clustering model on a q-dimensional space as:

$$\mathbf{y}_i = \mathbf{S}_i \left(\boldsymbol{\lambda}_0 \ + \ \boldsymbol{\Lambda} \, \boldsymbol{\alpha}_{ki} \ + \ \boldsymbol{\gamma}_i \right) + \boldsymbol{\epsilon}_i, \tag{H.2}$$

for $k = 1, \cdots K$ clusters.

with:

- $\boldsymbol{\epsilon}_i \sim N(\boldsymbol{0}, \mathbf{R})$,
- $\boldsymbol{\gamma}_i \sim N(\boldsymbol{0}, \boldsymbol{\Gamma})$,
- $\boldsymbol{\Lambda}$, *a projection matrix onto a h-dimensional subspace, with $h \leq q$.*

$\boldsymbol{S}_i = \left[\boldsymbol{s}(t_{i1}), \cdots, \boldsymbol{s}(t_{in_i}) \right]^T$ *is the spline basis matrix for curve i :*

$$\boldsymbol{S}_i = \begin{pmatrix} s_1(t_1) & \cdots & s_q(t_1) \\ \vdots & \ddots & \vdots \\ s_1(t_{n_i}) & \cdots & s_q(t_{n_i}) \end{pmatrix}. \tag{H.3}$$

The term $\boldsymbol{\alpha}_{ki}$ is defined as: $\boldsymbol{\alpha}_{ki} = \boldsymbol{\alpha}_k$ if curve i belongs to cluster k, where $\boldsymbol{\alpha}_k$ is a representation of the centroid of cluster k in a reduced h-dimensional subspace:

$$\boldsymbol{\alpha}_k = \left(\alpha_{k1}, \alpha_{k2} \ \ldots \ \alpha_{kh} \right)^T. \tag{H.4}$$

Then we have :

- $\boldsymbol{s}(t)^T \boldsymbol{\lambda}_0$*: the representation of the global mean curve,*
- $\boldsymbol{s}(t)^T (\boldsymbol{\lambda}_0 + \boldsymbol{\Lambda}\boldsymbol{\alpha}_k)$*: the global representation of the centroid of cluster k,*
- $\boldsymbol{s}(t)^T \boldsymbol{\Lambda}\boldsymbol{\alpha}_k$*: the local representation of the centroid of cluster k in connection with the global mean curve,*
- $\boldsymbol{s}(t)^T \boldsymbol{\gamma}_i$*: the local representation of the curve i in connection with the centroid of its cluster k.*

See Chap. [7] for a detailed description.

H.1.1 Model

$$\boldsymbol{y}_i = \boldsymbol{S}_i(\boldsymbol{\lambda}_0 + \boldsymbol{\Lambda}\boldsymbol{\alpha}_{ki} + \boldsymbol{\gamma}_i) + \boldsymbol{\epsilon}_i, \tag{H.5}$$

for curve $i = 1, \cdots, m^{(k)}$*, belonging to cluster* $k = 1, \cdots G$

H.1.2 Log-likelihood

$$\begin{aligned} l(\pi_k, \boldsymbol{\lambda}_0, \boldsymbol{\Lambda}, \boldsymbol{\alpha}_k, \boldsymbol{\Gamma}, \sigma^2 | \boldsymbol{y}_{1:N}, \boldsymbol{z}_{1:N}, \boldsymbol{\gamma}_{1:N}) = \\ -\frac{1}{2}\sum_{i=1}^{N}(n_i + q)\log(2\pi) \\ +\sum_{i=1}^{N}\sum_{k=1}^{G} z_{ki}\log(\pi_k) \qquad \text{(H.6)} \\ -\frac{1}{2}\sum_{i=1}^{N}\left[\log(|\boldsymbol{\Gamma}|) + \boldsymbol{\gamma}_i^T\boldsymbol{\Gamma}^{-1}\boldsymbol{\gamma}_i\right] \qquad \text{(H.7)} \\ -\frac{1}{2}\sum_{i=1}^{N}\sum_{k=1}^{G} z_{ki}\left[n_i\log(\sigma^2) + \frac{1}{\sigma^2}\left\|\boldsymbol{y}_i - \boldsymbol{S}_i(\boldsymbol{\lambda}_0 + \boldsymbol{\Lambda}\boldsymbol{\alpha}_k + \boldsymbol{\gamma}_i)\right\|^2\right]. \qquad \text{(H.8)} \end{aligned}$$

After initialization of all parameters, the EM algorithm consists in iteratively maximizing the expected values of (H.6)*,* (H.7) *and* (H.8) *given* $\mathbf{y}_i$ *and the current parameters estimates. As these three parts involve separate parameters, we can optimize them separately.*

H.1.3 Initialization

1. *let* $\mathbf{y}_i = \mathbf{S}_i\boldsymbol{\eta}_i$*,*
2. *estimating* $\boldsymbol{\eta}_i$ *by OLS, for* $i = 1, \cdots, N$*,*
3. *clustering all* $\boldsymbol{\eta}_i$ *by K-means,*
4. *estimating the mean curve of each cluster,* $\mathbf{m}_k$*, as centroid, we have:*

$$\mathbf{m}_k = \boldsymbol{\lambda}_0 + \mathbf{c}_k, \tag{H.9}$$

5. *estimating* $\boldsymbol{\lambda}_0$ *as the global mean curve of all clusters,*
6. *removing the global mean curve from* $\mathbf{m}_k$*:*

$$\mathbf{c}_k = \mathbf{m}_k - \boldsymbol{\lambda}_0, \tag{H.10}$$

7. *estimating* $\boldsymbol{\Lambda}$ *by SVD on* $\mathbf{c}_k$*,*

8. estimating $\boldsymbol{\alpha}_k$ such as:

$$\mathbf{c}_k = \boldsymbol{\Lambda}\ \boldsymbol{\alpha}_k \tag{H.11}$$

9. estimating probability π_k, such as:

$$\pi_k = \frac{n_i^{(k)}}{N} \tag{H.12}$$

10. choosing $\sigma^2 = 0$,
11. choosing $\boldsymbol{\Gamma} = \mathbf{0}$

H.1.4 E step

The E step consists in :

$$\widehat{\boldsymbol{\gamma}}_i = E\Big\{\boldsymbol{\gamma}_i | \boldsymbol{y}_i, \boldsymbol{\lambda}_0, \boldsymbol{\Lambda}, \boldsymbol{\alpha}, \boldsymbol{\Gamma}, \sigma^2, z_{ik}\Big\} \ . \tag{H.13}$$

For curve i we have the model :

$$\boldsymbol{y}_i = \boldsymbol{S}_i(\boldsymbol{\lambda}_0 + \boldsymbol{\Lambda}\boldsymbol{\alpha}_k + \boldsymbol{\gamma}_i) + \boldsymbol{\epsilon}_i \ . \tag{H.14}$$

Let :

$$\mathbf{u}_i = \boldsymbol{y}_i - \boldsymbol{S}_i(\boldsymbol{\lambda}_0 + \boldsymbol{\Lambda}\boldsymbol{\alpha}_k) \ . \tag{H.15}$$

Then, the joint distribution of $\mathbf{u}_i$ and $\boldsymbol{\gamma_i}$ is written as :

$$\begin{pmatrix} \mathbf{u}_i \\ \boldsymbol{\gamma}_i \end{pmatrix} \sim N\left(\begin{bmatrix} \mathbf{0} \\ \mathbf{0} \end{bmatrix}, \begin{bmatrix} \mathbf{S}\boldsymbol{\Gamma}\mathbf{S}^T + \sigma^2\mathbf{I} & \mathbf{S}\boldsymbol{\Gamma} \\ \boldsymbol{\Gamma}\mathbf{S}^T & \boldsymbol{\Gamma} \end{bmatrix} \right), \tag{H.16}$$

and the conditional distribution of $\boldsymbol{\gamma_i}$ given $\mathbf{u}_i$ is :

$$\boldsymbol{\gamma_i}|\mathbf{u}_i = N\Big(\widetilde{\boldsymbol{\gamma}_i}; \boldsymbol{\Sigma}_{\gamma_i}\Big) \ , \tag{H.17}$$

where :

$$\widetilde{\boldsymbol{\gamma}_i} = \big(\mathbf{S}_i^T\mathbf{S}_i + \boldsymbol{\sigma}^2\boldsymbol{\Gamma}^{-1}\big)^{-1}\mathbf{S}_i^T\mathbf{u}_i \ , \tag{H.18}$$

and :

$$\boldsymbol{\Sigma}_{\boldsymbol{\gamma_i}} = \sigma^2\big(\mathbf{S}_i^T\mathbf{S}_i + \sigma^2\boldsymbol{\Gamma}^{-1}\big)^{-1} \ . \tag{H.19}$$

Then, we have the conditional distribution for $(\widehat{\boldsymbol{\gamma}}_i|\mathbf{y}_i, z_{ik} = 1)$:

$$(\widehat{\boldsymbol{\gamma}}_i|\mathbf{y}_i, z_{ik} = 1) \sim N\big(\widetilde{\boldsymbol{\gamma}_i}; \boldsymbol{\Sigma}_{\widehat{\boldsymbol{\gamma}}_i}\big) \ , \tag{H.20}$$

with :

$$\widetilde{\boldsymbol{\gamma_i}} = \left(\mathbf{S}_i^T\mathbf{S}_i + \sigma^2\boldsymbol{\Gamma}^{-1}\right)^{-1}\mathbf{S}_i^T\left(\mathbf{y}_i - \mathbf{S}_i\boldsymbol{\lambda}_0 - \mathbf{S}_i\boldsymbol{\Lambda}\boldsymbol{\alpha}_k\right) , \tag{H.21}$$

$$\boldsymbol{\Sigma}_{\widehat{\boldsymbol{\gamma}_i}} = \sigma^2\left(\mathbf{S}_i^T\mathbf{S}_i + \sigma^2\boldsymbol{\Gamma}^{-1}\right)^{-1} , \tag{H.22}$$

and we find :

$$\widehat{\boldsymbol{\gamma}}_i = \left(\mathbf{S}_i^T\mathbf{S}_i + \sigma^2\boldsymbol{\Gamma}^{-1}\right)^{-1}\mathbf{S}_i^T\left(\mathbf{y}_i - \mathbf{S}_i\boldsymbol{\lambda}_0 - \mathbf{S}_i\boldsymbol{\Lambda}\boldsymbol{\alpha}_k\right) . \tag{H.23}$$

$$\widehat{\boldsymbol{\gamma}}_i = \mathbb{E}\Big\{\boldsymbol{\gamma}_i|\boldsymbol{y}_i, \boldsymbol{\lambda}_0, \boldsymbol{\Lambda}, \boldsymbol{\alpha}, \boldsymbol{\Gamma}, \sigma^2, z_{ik}\Big\} , \tag{H.24}$$

with :

$$\widehat{\boldsymbol{\gamma}}_i = \left(\mathbf{S}_i^T\mathbf{S}_i + \sigma^2\boldsymbol{\Gamma}^{-1}\right)^{-1}\mathbf{S}_i^T\left(\mathbf{y}_i - \mathbf{S}_i\boldsymbol{\lambda}_0 - \mathbf{S}_i\boldsymbol{\Lambda}\boldsymbol{\alpha}_k\right) . \tag{H.25}$$

H.1.5 M step

The M step involve maximizing :

$$Q = E\Big\{l(\pi_k, \boldsymbol{\lambda}_0, \boldsymbol{\Lambda}, \boldsymbol{\alpha}_k, \boldsymbol{\Gamma}, \sigma^2|\boldsymbol{y}_{1:N}, \boldsymbol{z}_{1:N}, \boldsymbol{\gamma}_{1:N})\Big\} , \tag{H.26}$$

holding $\boldsymbol{\gamma}_{1:N}$ fixed.

The sequential optimization of π_k, $\boldsymbol{\lambda}_0$, $\boldsymbol{\Lambda}$, $\boldsymbol{\alpha}_k$, $\boldsymbol{\Gamma}$, and σ^2 will be respectively detailed in the next subsections.

Estimation of $\widehat{\boldsymbol{\pi_k}}$

The expected value of (H.6) *is maximized by setting :*

$$\widehat{\pi_k} = \frac{1}{N}\sum_{i=1}^{N}\pi_{k|i} , \tag{H.27}$$

with :

$$\pi_{k|i} = P\left(z_{ik} = 1|\mathbf{y}_i\right) , \tag{H.28}$$

$$= \frac{f(y|z_{ik} = 1)\pi_k}{\sum_{j=1}^{G} f(y|z_{ij} = 1)\pi_j} , \tag{H.29}$$

with $f(y|z_{ik} = 1)$ given by :

$$\boldsymbol{y}_i \sim N(\boldsymbol{S}_i(\boldsymbol{\lambda}_0 + \boldsymbol{\Lambda}\boldsymbol{\alpha}_{z_i}), \boldsymbol{\Sigma}_i) , \tag{H.30}$$

where

$$\boldsymbol{\Sigma}_i = \sigma^2\mathbf{I} + \mathbf{S}_i\boldsymbol{\Gamma}\mathbf{S}_i^T . \tag{H.31}$$

Estimation of $\widehat{\mathbf{\Gamma}}$

The expected value of (H.7) *is maximized by setting :*

$$\widehat{\mathbf{\Gamma}} = \frac{1}{N}\sum_{i=1}^{N} E\Big[\widehat{\boldsymbol{\gamma}}_i\widehat{\boldsymbol{\gamma}}_i^T|\mathbf{Y}_i\Big] \;, \tag{H.32}$$

$$= \frac{1}{N}\sum_{i=1}^{N}\sum_{k=1}^{G} E\Big[\widehat{\boldsymbol{\gamma}}_i\widehat{\boldsymbol{\gamma}}_i^T|\mathbf{y}_i, z_{ik}=1\Big] \;, \tag{H.33}$$

with $(\widehat{\boldsymbol{\gamma}}_i|\mathbf{y}_i, z_{ik}=1)$ *given by the E step.*

To maximize (H.8)*, we need an iterative procedure where* $\boldsymbol{\lambda}_0$*,* $\boldsymbol{\alpha}_k$*, and the columns of* $\mathbf{\Lambda}$ *are repeatedly optimized while holding all other parameters fixed.*

Estimation of $\boldsymbol{\lambda_0}$

From the functional model :

$$\boldsymbol{y}_i = \boldsymbol{S}_i(\boldsymbol{\lambda}_0 + \mathbf{\Lambda}\boldsymbol{\alpha}_{\boldsymbol{z}_i} + \boldsymbol{\gamma}_i) + \boldsymbol{\epsilon}_i \;, \qquad i = 1, \cdots, N \;, \tag{H.34}$$

we have got, by Generalized Least Squares (GLS) :

$$\widehat{\boldsymbol{\lambda}}_0 = \Big(\sum_{i=1}^{N}\mathbf{S}_i^T\mathbf{S}_i\Big)^{-1}\sum_{i=1}^{N}\mathbf{S}_i^T\Big[\mathbf{y}_i - \sum_{k=1}^{G}\pi_{k|i}\mathbf{S}_i(\mathbf{\Lambda}\boldsymbol{\alpha}_k + \widehat{\boldsymbol{\gamma}}_{ik})\Big] \;, \tag{H.35}$$

with $\widehat{\boldsymbol{\gamma}}_{ik} = E\Big\{\boldsymbol{\gamma}_{ik}|z_{ik}=1, \mathbf{y}_i\Big\}$ *given by the E step.*

Estimation of $\boldsymbol{\alpha_k}$

The $\widehat{\boldsymbol{\alpha}}_k$ *are estimated from :*

$$\widehat{\boldsymbol{\alpha}}_k = \Big(\sum_{i=1}^{N}\pi_{k|i}\mathbf{\Lambda}^T\mathbf{S}_i^T\mathbf{S}_i\mathbf{\Lambda}\Big)^{-1}\sum_{i=1}^{N}\pi_{k|i}\mathbf{\Lambda}^T\mathbf{S}_i^T\Big[\mathbf{y}_i - \mathbf{S}_i\widehat{\boldsymbol{\lambda}_0} - \mathbf{S}_i\widehat{\boldsymbol{\gamma}}_{ik}\Big] \;. \tag{H.36}$$

Estimation of $\mathbf{\Lambda}$

By GLS, we only have the possibility of estimating vectors and not matrices, thus we will have to optimize each column of $\mathbf{\Lambda}$ *separately, holding all other fixed using :*

$$\boldsymbol{\Lambda}_m = \Big(\sum_{i=1}^{N} \sum_{k=1}^{G} \pi_{k|i} \widehat{\alpha}_{km}^2 \mathbf{S}_i^T \mathbf{S}_i \Big)^{-1} \sum_{i=1}^{N} \sum_{k=1}^{G} \pi_{k|i} \widehat{\alpha}_{km} \mathbf{S}_i^T \Big(\overline{\mathbf{y}}_i - \sum_{l \neq m}^{G} \widehat{\alpha}_{km} \mathbf{S}_i \widehat{\boldsymbol{\Lambda}}_l - \mathbf{S}_i \widehat{\gamma}_{ik} \Big) , \tag{H.37}$$

where :

- $\boldsymbol{\Lambda}_m$ *is the column* m *of* $\boldsymbol{\Lambda}$*,*
- $\widehat{\alpha}_{km}$ *is the component* m *of* $\widehat{\boldsymbol{\alpha}}_k$*,*
- $\overline{\mathbf{y}}_i = \mathbf{y}_i - \mathbf{S}_i \widehat{\boldsymbol{\lambda}}_0$.

We iterate through (H.35) (H.36) (H.37) *until all parameters have converged; then we can optimize* σ^2 *.*

Estimation of $\boldsymbol{\sigma^2}$

We have :

$$\widehat{\sigma^2} = \frac{1}{N} \sum_{i=1}^{N} \sum_{k=1}^{G} \pi_k E\Big[\big(\overline{\mathbf{y}}_i - \mathbf{S}_i \boldsymbol{\Lambda} \boldsymbol{\alpha}_k - \mathbf{S}_i \boldsymbol{\gamma}_i \big)^T \big(\overline{\mathbf{y}}_i - \mathbf{S}_i \boldsymbol{\Lambda} \boldsymbol{\alpha}_k - \mathbf{S}_i \boldsymbol{\gamma}_i \big) | \mathbf{y}_i, z_{ik} = 1 \Big] \tag{H.38}$$

Let :

$$\widehat{\sigma^2} = \frac{1}{N} \sum_{i=1}^{N} \sum_{k=1}^{G} \pi_k \Big\{ \big(\overline{\mathbf{y}}_i - \mathbf{S}_i \boldsymbol{\Lambda} \boldsymbol{\alpha}_k - \mathbf{S}_i \boldsymbol{\gamma}_i \big)^T \big(\overline{\mathbf{y}}_i - \mathbf{S}_i \boldsymbol{\Lambda} \boldsymbol{\alpha}_k - \mathbf{S}_i \boldsymbol{\gamma}_i \big) + \mathbf{S}_i \, Cov[\boldsymbol{\gamma}_i | \mathbf{y}_i, z_{ik} = 1] \mathbf{S}_i^T \Big\} . \tag{H.39}$$

The algorithm iterates until all the parameters have converged.

H.1.6 Constraints

We normalize $\boldsymbol{\alpha}_k$ *such as*

$$\sum_k \boldsymbol{\alpha}_k = \mathbf{0} , \tag{H.40}$$

and we apply the restriction

$$\boldsymbol{\Lambda}^T \mathbf{S}^T \boldsymbol{\Sigma}^{-1} \mathbf{S} \boldsymbol{\Lambda} = \mathbf{I} , \tag{H.41}$$

with :

$$\boldsymbol{\Sigma} = \sigma^2 \mathbf{I} + \mathbf{S} \boldsymbol{\Gamma} \mathbf{S}^T , \tag{H.42}$$

in order to define $\boldsymbol{\Lambda}$*.*

H.2 Functional Classification

Algorithm H.2.1 (Functional Classification by EM Algorithm) .

Let the Bayes's formula:

$$P(class = k|\boldsymbol{x}) = \frac{f_{(k)}(\boldsymbol{x})\ \pi_k}{\sum_{j=1}^{G} f_j(\boldsymbol{x})\ \pi_j}\ , \tag{H.43}$$

and the functional model:

$$\boldsymbol{y}_i^{(k)} = \boldsymbol{S}_i^{(k)}\Big(\boldsymbol{\lambda}_0 + \boldsymbol{\Lambda}\boldsymbol{\alpha}^{(k)} + \boldsymbol{\gamma}_i^{(k)}\Big) + \boldsymbol{\epsilon}_i^{(k)}\ . \tag{H.44}$$

Using the Bayes's formula, the a posteriori probability for curve $\mathbf{y}_i$ to belong to class (k) is proportional to:

$$\Big(\mathbf{y}_i^{(k)} - \mathbf{S}_i^{(k)}\boldsymbol{\lambda}_0 - \mathbf{S}_i^{(k)}\boldsymbol{\Lambda}\boldsymbol{\alpha}^{(k)}\Big)^T \boldsymbol{\Sigma}_i^{-1}\Big(\mathbf{y}_i^{(k)} - \mathbf{S}_i^{(k)}\boldsymbol{\lambda}_0 - \mathbf{S}_i^{(k)}\boldsymbol{\Lambda}\boldsymbol{\alpha}^{(k)}\Big) - 2\log\pi_k, \tag{H.45}$$

where $\mathbf{S}_i^{(k)}$ is the spline basis matrix for curve $\mathbf{y}_i^{(k)}$, and

$$\boldsymbol{\Sigma}_i = \sigma^2\mathbf{I} + \mathbf{S}_i^{(k)}\boldsymbol{\Gamma}(\mathbf{S}_i^{(k)})^T\ . \tag{H.46}$$

So the curve $\left(\mathbf{y}_i^{(k)}\right)$ will be classified to class k that minimizes:

$$k = \arg\min_{(k)}\Big(\|\mathbf{y}_i^{(k)} - \mathbf{S}_i^{(k)}\boldsymbol{\lambda}_0 - \mathbf{S}_i^{(k)}\boldsymbol{\Lambda}\boldsymbol{\alpha}^{(k)}\|^2_{\boldsymbol{\Sigma}_i^{-1}} - 2\log\pi_k\Big)\ . \tag{H.47}$$

H.3 Kalman Filter (KF)

Algorithm H.3.1 (Kalman Filter (KF)) .

H.3.1 System

$$\mathbf{x}_k = \mathbf{A}_k\mathbf{x}_{k-1} + \mathbf{B}_k\mathbf{u}_k + \mathbf{G}_k\mathbf{v}_k \tag{H.48}$$
$$\mathbf{y}_k = \mathbf{C}_k\mathbf{x}_k + \mathbf{D}_k\mathbf{n}_k \tag{H.49}$$

where :

- $\mathbf{x}_k \in R^n$, $\mathbf{y}_k \in R^q$, $\mathbf{u}_k \in R^m$, $\mathbf{v}_k \in R^p$, $\mathbf{n}_k \in R^q$,
- *Eq.* (H.48) *is the state equation,*
- *Eq.* (H.49) *is the measurement equation,*
- $\mathbf{A}_k\,[n,n]$, $\mathbf{B}_k\,[n,m]$, $\mathbf{G}_k\,[n,p]$, $\mathbf{C}_k\,[q,n]$, $\mathbf{D}_k\,[q,q]$ *(these matrices may be time-variant but are known),*
- $\{\mathbf{u}_k\}$ *is a deterministic sequence,*
- $\{\mathbf{v}_k\}$, $\{\mathbf{n}_k\}$ *are the process and measurement noise sequences, with* $\mathbf{v}_k \sim \mathcal{N}(0, \mathbf{Q})$, $\mathbf{n}_k \sim \mathcal{N}(0, \mathbf{R})$,
- $\mathbf{x}_0 \sim \mathcal{N}(\overline{\mathbf{x}_0}, \mathbf{P}_0)$

H.3.2 Initialization

$$\widehat{\mathbf{x}}_0 = E\{\mathbf{x}_0\} = \overline{\mathbf{x}_0} \tag{H.50a}$$
$$\mathbf{P}_0 = E[(\mathbf{x}_0 - \widehat{\mathbf{x}}_0)(\mathbf{x}_0 - \widehat{\mathbf{x}}_0)^T] \tag{H.50b}$$
$$\mathbf{R} = E[(\mathbf{v} - \overline{\mathbf{v}})(\mathbf{v} - \overline{\mathbf{v}})^T] \tag{H.50c}$$
$$\mathbf{Q} = E[(\mathbf{n} - \overline{\mathbf{n}})(\mathbf{n} - \overline{\mathbf{n}})^T] \tag{H.50d}$$

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

H.3.3 Prediction step

Compute the predicted state mean and covariance (time update)

$$\widehat{\mathbf{x}}_{k|k-1} = \mathbf{A}_{k-1}\widehat{\mathbf{x}}_{k-1|k-1} + \mathbf{B}_{k-1}\mathbf{u}_{k-1} + \mathbf{G}_k\overline{\mathbf{v}} \tag{H.51a}$$
$$\mathbf{P}_{k|k-1} = \mathbf{A}_{k-1}\mathbf{P}_{k-1|k-1}\mathbf{A}_{k-1}^T + \mathbf{G}_{k-1}\mathbf{Q}_{k-1}\mathbf{G}_{k-1}^T \tag{H.51b}$$

H.3.4 Correction step

Update estimates with latest observation (measurement update)

$$\boldsymbol{\Sigma}_{k|k-1} = \mathbf{C}_k \mathbf{P}_{k|k-1} \mathbf{C}_k^T + \mathbf{R}_k \tag{H.52a}$$

$$\mathbf{K}_k = \mathbf{P}_{k|k-1} \mathbf{C}_k^T \boldsymbol{\Sigma}_{k|k-1}^{-1} \tag{H.52b}$$

$$\mathbf{e}_k = \mathbf{y}_k - \mathbf{C}_k \widehat{\mathbf{x}}_{k|k-1} - \mathbf{D}_k \overline{\mathbf{n}} \tag{H.52c}$$

$$\widehat{\mathbf{x}}_{k|k} = \widehat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k \mathbf{e}_k \tag{H.52d}$$

$$\mathbf{P}_{k|k} = \mathbf{P}_{k|k-1} - \mathbf{K}_k \mathbf{C}_k \mathbf{P}_{k|k-1} \tag{H.52e}$$

H.4 Extended Kalman Filter (EKF)

Algorithm H.4.1 (Extended Kalman Filter (EKF)) .

H.4.1 System

$$\mathbf{x}_k = \mathbf{f}_k(\mathbf{x}_{k-1}, \mathbf{u}_k, \mathbf{v}_k) \tag{H.53}$$
$$\mathbf{y}_k = \mathbf{h}_k(\mathbf{x}_k, \mathbf{n}_k) \tag{H.54}$$

where :

- $\mathbf{x}_k \in R^n$, $\mathbf{y}_k \in R^q$, $\mathbf{u}_k \in R^m$, $\mathbf{v}_k \in R^p$, $\mathbf{n}_k \in R^q$,
- $\mathbf{f}_k$ *is the state equation,* $\mathbf{h}_k$ *is the measurement equation, these parametric functions may be time-variant but are known,*
- $\{\mathbf{u}_k\}$ *is a deterministic sequence,*
- $\{\mathbf{v}_k\}$, $\{\mathbf{n}_k\}$ *are process and measurement noise sequences; with* $\mathbf{v}_k \sim \mathcal{N}(0, \mathbf{Q})$, $\mathbf{n}_k \sim \mathcal{N}(0, \mathbf{R})$,
- $\mathbf{x}_0 \sim \mathcal{N}(\overline{\mathbf{x}_0}, \mathbf{P}_0)$.

H.4.2 Initialization

$$\widehat{\mathbf{x}}_0 = E\{\mathbf{x}_0\} = \overline{\mathbf{x}_0} \tag{H.55a}$$
$$\mathbf{P}_0 = E[(\mathbf{x}_0 - \widehat{\mathbf{x}}_0)(\mathbf{x}_0 - \widehat{\mathbf{x}}_0)^T] \tag{H.55b}$$
$$\mathbf{R} = E[(\mathbf{v} - \overline{\mathbf{v}})(\mathbf{v} - \overline{\mathbf{v}})^T] \tag{H.55c}$$
$$\mathbf{Q} = E[(\mathbf{n} - \overline{\mathbf{n}})(\mathbf{n} - \overline{\mathbf{n}})^T] \tag{H.55d}$$

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

H.4.3 Prediction step

Compute the process model Jacobians :

$$\mathbf{F}_k = \nabla_{\mathbf{x}} \mathbf{f}_k(\mathbf{x}, \overline{\mathbf{v}}, \mathbf{u}_k)|_{\mathbf{x}=\widehat{\mathbf{x}}_{k-1}} \tag{H.56a}$$
$$\mathbf{G}_k = \nabla_{\mathbf{v}} \mathbf{f}_k(\widehat{\mathbf{x}}_{\mathbf{k-1}}, \mathbf{v}, \mathbf{u}_k)|_{\mathbf{v}=\widehat{\mathbf{v}}} \tag{H.56b}$$

Compute the predicted state mean and covariance (time update)

$$\widehat{\mathbf{x}}_{k|k-1} = \mathbf{f}_k(\widehat{\mathbf{x}}_{k|k-1}, \overline{\mathbf{v}}, \mathbf{u}_k) \tag{H.57a}$$
$$\mathbf{P}_{k|k-1} = \mathbf{F}_k \mathbf{P}_{k-1|k-1} \mathbf{F}_k^T + \mathbf{G}_k \mathbf{R} \mathbf{G}_k^T \tag{H.57b}$$

H.4.4 Correction step

Compute the observation model Jacobians :

$$\mathbf{H}_k = \nabla \mathbf{h}_k(\mathbf{x}, \mathbf{n})|_{\mathbf{x}=\widehat{\mathbf{x}}_{k|k-1}} \tag{H.58a}$$
$$\mathbf{D}_k = \nabla \mathbf{h}_k(\mathbf{x}_{k|k-1}, \mathbf{n})|_{\mathbf{n}=\overline{\mathbf{n}}} \tag{H.58b}$$

Update estimates with latest observation (measurement update)

$$\mathbf{\Sigma}_{k|k-1} = \mathbf{H}_k \mathbf{P}_{k|k-1} \mathbf{H}_k^T + \mathbf{D}_k \mathbf{R} \mathbf{D}_k^T \tag{H.59a}$$
$$\mathbf{K}_k = \mathbf{P}_{k|k-1} \mathbf{H}_k^T \mathbf{\Sigma}_{k|k-1}^{-1} \tag{H.59b}$$
$$\mathbf{e}_k = \mathbf{y}_k - \mathbf{h}_k(\widehat{\mathbf{x}}_{k|k-1}, \overline{\mathbf{n}}) \tag{H.59c}$$
$$\widehat{\mathbf{x}}_{k|k} = \widehat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k \mathbf{e}_k \tag{H.59d}$$
$$\mathbf{P}_{k|k} = \mathbf{P}_{k|k-1} - \mathbf{K}_k \mathbf{H}_k \mathbf{P}_{k|k-1} \tag{H.59e}$$

H.5 Unscented Kalman Filter (UKF)

Algorithm H.5.1 (Unscented Kalman Filter (UKF)) .

H.5.1 System

$$\mathbf{x}_k = \mathbf{f}_k(\mathbf{x}_{k-1}, \mathbf{u}_k, \mathbf{v}_k) \tag{H.60}$$
$$\mathbf{y}_k = \mathbf{h}_k(\mathbf{x}_k, \mathbf{n}_k) \tag{H.61}$$

where :

- $\mathbf{x}_k \in R^n$, $\mathbf{y}_k \in R^q$, $\mathbf{u}_k \in R^m$, $\mathbf{v}_k \in R^p$, $\mathbf{n}_k \in R^q$,
- $\mathbf{f}_k$ *is the state equation,* $\mathbf{h}_k$ *is the measurement equation, these parametric functions may be time-variant but are known,*
- $\{\mathbf{u}_k\}$ *is a deterministic sequence,*
- $\{\mathbf{v}_k\}$, $\{\mathbf{n}_k\}$ *are process and measurement noise sequences; with* $\mathbf{v}_k \sim \mathcal{N}(0, \mathbf{Q})$, $\mathbf{n}_k \sim \mathcal{N}(0, \mathbf{R})$,
- $\mathbf{x}_0 \sim \mathcal{N}(\overline{\mathbf{x}_0}, \mathbf{P}_0)$.

H.5.2 Initialization

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

$$w_0^{(m)} = \frac{\lambda}{L+\lambda} \tag{H.62a}$$
$$w_0^{(c)} = \frac{\lambda}{L+\lambda} + (1 - \alpha^2 + \beta) \tag{H.62b}$$
$$w_i^{(m)} = w_i^{(c)} = \frac{1}{2(L+\lambda)} \quad \textit{for } i = 1, \cdots, 2L, \tag{H.62c}$$

$$\widehat{\mathbf{x}}_0 = \mathbb{E}[\mathbf{x}_0] \tag{H.63a}$$
$$\mathbf{P}_{\mathbf{x}_0} = \mathbb{E}[(\mathbf{x}_0 - \widehat{\mathbf{x}}_0)(\mathbf{x}_0 - \widehat{\mathbf{x}}_0)^T] \tag{H.63b}$$
$$\widehat{\mathbf{x}}_0^a = \mathbb{E}[\mathbf{x}^a] = \mathbb{E}[\big((\mathbf{x}_0)^T \ (\mathbf{0})^T \ (\mathbf{0})^T \big)^T] \tag{H.63c}$$
$$\mathbf{P}_0^a = \mathbb{E}[(\mathbf{x}_0^a - \widehat{\mathbf{x}}_0^a)(\mathbf{x}_0^a - \widehat{\mathbf{x}}_0^a)^T] = \begin{pmatrix} \mathbf{P}_{\mathbf{x}_0} & 0 & 0 \\ 0 & \mathbf{Q} & 0 \\ 0 & 0 & \mathbf{R} \end{pmatrix} \tag{H.63d}$$

H.5.3 Sigma-points

$$\mathcal{X}^a_{k-1} = \left(\widehat{\mathbf{x}}^a_{k-1} \; \widehat{\mathbf{x}}^a_{k-1} + \gamma \sqrt{\mathbf{P}^a_{k-1}} \; \widehat{\mathbf{x}}^a_{k-1} - \gamma \sqrt{\mathbf{P}^a_{k-1}} \right) \tag{H.64}$$

H.5.4 Prediction step

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

$$\mathcal{X}^x_{k|k-1} = \mathbf{f}(\mathcal{X}^x_{k-1}, \mathcal{X}^v_{k-1}, \mathbf{u}_{k-1}) \tag{H.65a}$$

$$\widehat{\mathbf{x}}_{k|k-1} = \sum_{i=0}^{2L} w_i^{(m)} \mathcal{X}^x_{i,k|k-1} \tag{H.65b}$$

$$\mathbf{P}_{\mathbf{x}_{k|k-1}} = \sum_{i=0}^{2L} w_i^{(c)} (\mathcal{X}^x_{i,k|k-1} - \widehat{\mathbf{x}}_{k|k-1})(\mathcal{X}^x_{i,k|k-1} - \widehat{\mathbf{x}}_{k|k-1})^T \tag{H.65c}$$

H.5.5 Innovation

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

$$\mathcal{Y}_{k|k-1} = \mathbf{h}(\mathcal{X}^x_{k|k-1}, \mathcal{X}^n_{k-1}) \tag{H.66a}$$

$$\widehat{\mathbf{y}}_{k|k-1} = \sum_{i=0}^{2L} w_i^{(m)} \mathcal{Y}_{i,k|k-1} \tag{H.66b}$$

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

H.5.6 Measurement Update step

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

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

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

$$\mathbf{K}_k = \mathbf{P}_{\mathbf{x}_k \mathbf{y}_k} \mathbf{P}^{-1}_{\widetilde{\mathbf{y}}_k} \tag{H.67c}$$

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

$$\widehat{\mathbf{x}}_{k|k} = \widehat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k \mathbf{e}_k \tag{H.68a}$$

$$\mathbf{P}_{\mathbf{x}_{k|k}} = \mathbf{P}_{\mathbf{x}_{k|k-1}} - \mathbf{K}_k \mathbf{P}_{\widetilde{\mathbf{y}}_k} \mathbf{K}_k^T \tag{H.68b}$$

Parameters

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

H.6 Bootstrap Particle Filter(BPF)

Algorithm H.6.1 (Bootstrap Particle Filter (BPF)) .

H.6.1 System

$$\mathbf{x}_k = \mathbf{f}_k(\mathbf{x}_{k-1}, \mathbf{u}_k, \mathbf{v}_k) \tag{H.69}$$
$$\mathbf{y}_k = \mathbf{h}_k(\mathbf{x}_k, \mathbf{n}_k) \tag{H.70}$$

where :

- $\mathbf{x}_k \in R^n$, $\mathbf{y}_k \in R^q$, $\mathbf{u}_k \in R^m$, $\mathbf{v}_k \in R^p$, $\mathbf{n}_k \in R^q$,
- $\mathbf{f}_k$ *is the state equation,* $\mathbf{h}_k$ *is the measurement equation, these parametric functions may be time-variant but are known,*
- $\{\mathbf{u}_k\}$ *is a deterministic sequence,*
- $\{\mathbf{v}_k\}$, $\{\mathbf{n}_k\}$ *are process and measurement noise sequences; with* $\mathbf{v}_k \sim \mathcal{N}(0, \mathbf{Q})$, $\mathbf{n}_k \sim \mathcal{N}(0, \mathbf{R})$,
- $\mathbf{x}_0 \sim \mathcal{N}(\overline{\mathbf{x}_0}, \mathbf{P}_0)$.

H.6.2 Initialization

We draw N *particles* $\{\mathbf{x}_0^{(i)}\}$ *from the estimated distribution* $p(\mathbf{x}_0)$ *of the initial state* $\mathbf{x}_0$.

at time $\mathbf{k-1}$, *we have particles* $\{\mathbf{x}_{k-1}^{(i)}\}$ *and weights* $\{w_k^{(i)}\}$ *that give us an approximation of the posterior distribution* $p(\mathbf{x}_{k-1}^{(i)}|\mathbf{Y}_{1:k-1})$.

H.6.3 Prediction step

We generate process noise $\mathbf{v}_k^{(i)}$ *according to its distribution and we estimate* N *new particles, using the state equation*

$$\mathbf{x}_k^{(i)} \;=\; \mathbf{f}_k(\mathbf{x}_{k-1}^{(i)}, \mathbf{v}_k^{(i)}) \tag{H.71}$$

at time $\mathbf{k}$, *we observe* $\mathbf{y}_k$

H.6.4 Updating step

we estimate

$$\widehat{\mathbf{y}}_k^{(i)} \;=\; \mathbf{h}_k(\mathbf{x}_k^{(i)}) \tag{H.72}$$

and we have

$$\mathbf{e}_k^{(i)} \;=\; \mathbf{y}_k \;-\; \widehat{\mathbf{y}}_k^{(i)} \tag{H.73}$$

we estimate the likelihood function

$$Llh_k^{(i)} \;= \frac{1}{(2\pi)^{\frac{q}{2}}(detR)^{\frac{1}{2}}} \exp\Big[-\frac{1}{2}\big(\mathbf{e}_k^{(i)}-\boldsymbol{\mu}_n\big)^T\mathbf{R}^{-1}\big(\mathbf{e}_k^{(i)}-\boldsymbol{\mu}_n\big)\Big] \tag{H.74}$$

and we find the weights

$$w_k^{(i)} \;=\; \widetilde{w}_{k-1}^{(i)} \;*\; Llh_k^{(i)} \tag{H.75}$$

that we normalize

$$\widetilde{w}_k^{(i)} \;=\; w_k^{(i)}\;\frac{1}{\sum_{j=1}^N w_k^{(j)}} \tag{H.76}$$

and we get $\{(\mathbf{x}_k^{(i)}, \widetilde{w}_k^{(i)})\}$ *to approximate the new posterior distribution* $p(\mathbf{x}_k|\mathbf{y}_k)$.

H.6.5 Resampling step

From $\{\mathbf{x}_k^{(i)}\}$ *and* $\{\widetilde{w}_k^{(i)}\}$ *we eliminate particles with lower weights and duplicate particles with higher weights to give a new set of N particles* $\{\widetilde{\mathbf{x}}_k^{(i)}\}$ *with the same weights* $\{\widetilde{w}_k^{(i)} = \frac{1}{N}\}$.

H.6.6 Expectation step

We have got

$$E[\mathbf{x}_k] \;=\; \sum_{i=1}^{N} \widetilde{w}_k^{(i)}\widetilde{\mathbf{x}}_k^{(i)}\;. \tag{H.77}$$

H.7 Sigma-Point Particle Filter (SPPF)

Algorithm H.7.1 (Sigma-Point Particle Filter) .

H.7.1 System

$$\mathbf{x}_k = \mathbf{f}_k(\mathbf{x}_{k-1}, \mathbf{u}_k, \mathbf{v}_k) \tag{H.78}$$
$$\mathbf{y}_k = \mathbf{h}_k(\mathbf{x}_k, \mathbf{n}_k) \tag{H.79}$$

where :

- $\mathbf{x}_k \in R^n$, $\mathbf{y}_k \in R^q$, $\mathbf{u}_k \in R^m$, $\mathbf{v}_k \in R^p$, $\mathbf{n}_k \in R^q$,
- $\mathbf{f}_k$ *is the state equation,* $\mathbf{h}_k$ *is the measurement equation, these parametric functions may be time-variant but are known,*
- $\{\mathbf{u}_k\}$ *is a deterministic sequence,*
- $\{\mathbf{v}_k\}$, $\{\mathbf{n}_k\}$ *are process and measurement noise sequences; with* $\mathbf{v}_k \sim \mathcal{N}(0, \mathbf{Q})$, $\mathbf{n}_k \sim \mathcal{N}(0, \mathbf{R})$,
- $\mathbf{x}_0 \sim \mathcal{N}(\overline{\mathbf{x}_0}, \mathbf{P}_0)$.

H.7.2 Initialization

We draw N *particles* $\{\mathbf{x}_0^{(i)}\}$ *from the estimated distribution* $p(\mathbf{x}_0)$ *of the initial state* $\mathbf{x}_0$.

at time $k-1$,

we know $\widehat{\mathbf{x}}_{k-1}^{(i)}$ *the estimate of the state and* $\widehat{\mathbf{S}}_{\mathbf{x}_{k-1}}^{(i)}$ *the covariance of the state, for all* $i = 1 \cdots N$.

at time k,

we observe $\mathbf{y}_k$,

H.7.3 Importance sampling Step

1. *Update the gaussian prior distribution for each particle with the UKF algorithm, that gives an estimate of the state at time* k, $\overline{\mathbf{x}}_k^{(i)}$ *and an estimate of the covariance of the state* $\widehat{\mathbf{S}}_{\mathbf{x}_k}^{(i)}$ *:*

$$\left(\overline{\mathbf{x}}_k^{(i)}, \widehat{\mathbf{S}}_{\mathbf{x}_k}^{(i)}\right) \leftarrow UKF\left(\widehat{\mathbf{x}}_{k-1}^{(i)}, \widehat{\mathbf{S}}_{\mathbf{x}_{k-1}}^{(i)}, \mathbf{y}_k, \mathbf{v}_k, \mathbf{n}_k\right), \tag{H.80}$$

2. *sample the new estimate of the state* $\widehat{\mathbf{x}}_k^{(i)}$ *from :*

$$\widehat{\mathbf{x}}_k^{(i)} \sim q_{\mathcal{N}}\big(\mathbf{x}_k|\mathbf{Y}_{1:k}\big) \;=\; \mathcal{N}\big(\mathbf{x}_k;\overline{\mathbf{x}}_k^{(i)},\widehat{\mathbf{S}}_{\mathbf{x}_k}^{(i)}\big) \tag{H.81}$$

$$\widehat{\mathbf{x}}_k^{(i)} \;=\; \overline{\mathbf{x}}_k^{(i)} \;+\; \widehat{\mathbf{S}}_{\mathbf{x}_k}^{(i)}\,\boldsymbol{\nu}_k \tag{H.82}$$

where $\boldsymbol{\nu}_k \sim \mathcal{N}(\mathbf{0},\mathbf{1})$*,*

3. *estimate the transition prior distribution*

$$\mathbf{v}_k^{(i)} = \widehat{\mathbf{x}}_k^{(i)} \;-\; \mathbf{f}_k\big(\widehat{\mathbf{x}}_{k-1}^{(i)}\big) \tag{H.83}$$

$$tpr_k^{(i)} = \frac{1}{(2\pi)^{\frac{p}{2}}(det\mathbf{Q})^{\frac{1}{2}}}\exp\Big[-\frac{1}{2}\big(\mathbf{v}_k^{(i)}-\boldsymbol{\mu}_v\big)^T\mathbf{Q}^{-1}\big(\mathbf{v}_k^{(i)}-\boldsymbol{\mu}_v\big)\Big] \tag{H.84}$$

4. *estimate the likelihood distribution*

$$\mathbf{e}_k^{(i)} = \mathbf{y}_k \;-\; \mathbf{h}_k\big(\widehat{\mathbf{x}}_k^{(i)}\big) \tag{H.85}$$

$$llh_k^{(i)} = \frac{1}{(2\pi)^{\frac{q}{2}}(det\mathbf{R})^{\frac{1}{2}}}\exp\Big[-\frac{1}{2}\big(\mathbf{e}_k^{(i)}-\boldsymbol{\mu}_n\big)^T\mathbf{R}^{-1}\big(\mathbf{e}_k^{(i)}-\boldsymbol{\mu}_n\big)\Big] \tag{H.86}$$

5. *estimate the proposal distribution*

$$ppd_k^{(i)} \;=\; \frac{1}{(2\pi)^{\frac{n}{2}}(det\mathbf{S}_{\mathbf{x_k}}^{(\mathbf{i})})^{\frac{1}{2}}}\exp\Big[-\frac{1}{2}\big(\widehat{\mathbf{x}}_k^{(i)}-\overline{\mathbf{x}}_k^{(i)}\big)^T\big(\mathbf{S}_{\mathbf{x_k}}^{(\mathbf{i})}\big)^{-1}\big(\widehat{\mathbf{x}}_k^{(i)}-\overline{\mathbf{x}}_k^{(i)}\big)\Big] \tag{H.87}$$

6. *estimate of the weights*

$$w_k^{(i)} = w_{k-1}^{(i)}\;\frac{llh_k^{(i)}\;*\;tpr_k^{(i)}}{ppd_k^{(i)}} \tag{H.88}$$

$$\widetilde{w}_k^{(i)} = w_k^{(i)}\;\frac{1}{\sum_{j=1}^{N} w_k^{(j)}} \tag{H.89}$$

H.7.4 Resampling step

From $\{\widehat{\mathbf{x}}_k^{(i)}\}$ *and* $\{\widetilde{w}_k^{(i)}\}$ *we eliminate particles with lower weights and duplicate particles with higher weights to give a new set of N particles* $\{\widetilde{\mathbf{x}}_k^{(i)}\}$ *with the same weights* $\{\widetilde{w}_k^{(i)}=\frac{1}{N}\}$*.*

H.7.5 Expectation step

We have got

$$E[\mathbf{x}_k] \;=\; \sum_{i=1}^{N}\widetilde{w}_k^{(i)}\widetilde{\mathbf{x}}_k^{(i)}\;. \tag{H.90}$$

H.8 Parameter Estimation for a Linear Model by Gauss-Newton

Algorithm H.8.1 (Parameter Estimation for a Linear Model by Gauss-Newton)
.

H.8.1 System

Let consider a Dynamic State Space model:

$$\mathbf{x}_k = \mathbf{A}(\boldsymbol{\theta})\mathbf{x}_{k-1} + \mathbf{v}_k \tag{H.91}$$
$$\mathbf{y}_k = \mathbf{C}(\boldsymbol{\theta})\mathbf{x}_k + \mathbf{n}_k \tag{H.92}$$

where :

- $\mathbf{x}_k \in \mathbb{R}^n$, $\mathbf{y}_k \in \mathbb{R}^q$, $\mathbf{v}_k \in \mathbb{R}^p$, $\mathbf{n}_k \in \mathbb{R}^q$,
- $\mathbf{A}_k(\boldsymbol{\theta})$ $[n, n]$, $\mathbf{C}_k(\boldsymbol{\theta})$ $[q, n]$,
- *these system matrices depend on a set of unknown parameters* $\boldsymbol{\theta} \in \mathbb{R}^J$ *where* J *is the number of hyperparameters,*
- $\mathbf{x}_k$ *is the state at time* k,
- $\mathbf{y}_k$ *is the observation at time* k,
- $\{\mathbf{v}_k\}$, $\{\mathbf{n}_k\}$ *the system, and measurement white noises, with* $\mathbf{v}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}(\boldsymbol{\theta}))$, $\mathbf{n}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{R}(\boldsymbol{\theta}))$,
- $\mathbf{x}_0 \sim \mathcal{N}(\overline{\mathbf{x}_0}, \mathbf{P}_0)$.

H.8.2 Kalman Filter

Initialisation

$$\widehat{\mathbf{x}}_{0|-1} = E\{\mathbf{x}_0\} = \overline{\mathbf{x}_0} \tag{H.93}$$
$$\mathbf{P}_{0|-1} = Var\{\mathbf{x}_0\} \tag{H.94}$$
$$\theta_0 = E(\theta) \tag{H.95}$$

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

Prediction

$$\widehat{\mathbf{x}}_{k|k-1} = \mathbf{A}\widehat{\mathbf{x}}_{k-1|k-1} \tag{H.96}$$
$$\mathbf{P}_{k|k-1} = \mathbf{A}\mathbf{P}_{k-1|k-1}\mathbf{A}^T + \mathbf{Q} \tag{H.97}$$

Updating

$$\boldsymbol{\Sigma}_{k|k-1} = \mathbf{C}\mathbf{P}_{k|k-1}\mathbf{C}^T + \mathbf{R} \tag{H.98}$$
$$\mathbf{K}_k = \mathbf{P}_{k|k-1}\mathbf{C}^T\boldsymbol{\Sigma}_{k|k-1}^{-1} \tag{H.99}$$
$$\mathbf{e}_k = \mathbf{y}_k - \mathbf{C}\widehat{\mathbf{x}}_{k|k-1} \tag{H.100}$$
$$\widehat{\mathbf{x}}_{k|k} = \widehat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k\mathbf{e}_k \tag{H.101}$$
$$\mathbf{P}_{k|k} = \mathbf{P}_{k|k-1} - \mathbf{K}_k\mathbf{C}\mathbf{P}_{k|k-1} \tag{H.102}$$

H.8.3 Likelihood

the k-ieme contribution of the log-likelihood is given by :

$$l_k = -\frac{q}{2}\log(2\pi) - \frac{1}{2}\log \mid \boldsymbol{\Sigma}_{k|k-1} \mid -\frac{1}{2}(\mathbf{e}_k^T\boldsymbol{\Sigma}_{k|k-1}^{-1}\mathbf{e}_k) \tag{H.103}$$

the likelihood is given by:

$$\log L = -\frac{Nq}{2}\log(2\pi) - \frac{1}{2}\sum_{k=1}^{N}\log \mid \boldsymbol{\Sigma}_{k|k-1} \mid -\frac{1}{2}\sum_{k=1}^{N}(\mathbf{e}_k^T\boldsymbol{\Sigma}_{k|k-1}^{-1}\mathbf{e}_k) \tag{H.104}$$

H.8.4 log-likelihood

$$\frac{\partial l_k}{\partial \theta_i} = -\frac{1}{2}tr\Big\{\Big[\boldsymbol{\Sigma}_{k|k-1}^{-1}\frac{\partial \boldsymbol{\Sigma}_{k|k-1}}{\partial \theta_i}\Big]\Big[\mathbf{I} - \boldsymbol{\Sigma}_{k|k-1}^{-1}\mathbf{e}_k\mathbf{e}_k^T\Big]\Big\} - \frac{\partial \mathbf{e}_k^T}{\partial \theta_i}\boldsymbol{\Sigma}_{k|k-1}^{-1}\mathbf{e}_k \tag{H.105}$$

the derivation of the log-likelihood is:

$$\frac{\partial \log L}{\partial \theta_i} = -\sum_{k=1}^{N}\left\{\frac{1}{2}tr\Big\{\Big[\boldsymbol{\Sigma}_{k|k-1}^{-1}\frac{\partial \boldsymbol{\Sigma}_{k|k-1}}{\partial \theta_i}\Big]\Big[\mathbf{I} - \boldsymbol{\Sigma}_{k|k-1}^{-1}\mathbf{e}_k\mathbf{e}_k^T\Big]\Big\} + \frac{\partial \mathbf{e}_k^T}{\partial \theta_i}\boldsymbol{\Sigma}_{k|k-1}^{-1}\mathbf{e}_k\right\} \tag{H.106}$$

for $i \in \{1, J\}$, where J is the number of hyperparameters.

$$\frac{\partial \mathbf{e}_k}{\partial \theta_i} = -\frac{\partial \mathbf{D}_k}{\partial \theta_i} u_k - \frac{\partial \mathbf{C}_k}{\partial \theta_i} \widehat{x}_{k|k-1} - \mathbf{C}_k \frac{\partial \widehat{\mathbf{x}}_{k|k-1}}{\partial \theta_i} \tag{H.107}$$

$$\frac{\partial \mathbf{\Sigma}_{k|k-1}}{\partial \theta_i} = \frac{\partial \mathbf{C}_k}{\partial \theta_i} \mathbf{P}_{k|k-1} \mathbf{C}_k^T + \mathbf{C}_k \frac{\partial \mathbf{P}_{k|k-1}}{\partial \theta_i} \mathbf{C}_k^T + \mathbf{C}_k \mathbf{P}_{k|k-1} \frac{\partial \mathbf{C}_k^T}{\partial \theta_i} + \frac{\partial \mathbf{R}_k}{\partial \theta_i} \tag{H.108}$$

$$\frac{\partial \widehat{\mathbf{x}}_{k|k-1}}{\partial \theta_i} = \frac{\partial \mathbf{A}_{k-1}}{\partial \theta_i} \widehat{\mathbf{x}}_{k-1|k-1} + \mathbf{A}_{k-1} \frac{\partial \widehat{\mathbf{x}}_{k-1|k-1}}{\partial \theta_i} + \frac{\partial \mathbf{B}_{k-1}}{\partial \theta_i} \mathbf{u}_{k-1} \tag{H.109}$$

$$\begin{aligned} \frac{\partial \mathbf{P}_{k|k-1}}{\partial \theta_i} = & \frac{\partial \mathbf{A}_{k-1}}{\partial \theta_i} \mathbf{P}_{k-1|k-1} \mathbf{A}_{k-1}^T + \mathbf{A}_{k-1} \frac{\partial \mathbf{P}_{k-1|k-1}}{\partial \theta_i} \mathbf{A}_{k-1}^T \\ & + \mathbf{A}_{k-1} \mathbf{P}_{k-1|k-1} \frac{\partial \mathbf{A}_{k-1}^T}{\partial \theta_i} + \frac{\partial \mathbf{G}_{k-1}}{\partial \theta_i} \mathbf{Q}_{k-1} \mathbf{G}_{k-1}^T \\ & + \mathbf{G}_{k-1} \frac{\partial \mathbf{Q}_{k-1}}{\partial \theta_i} \mathbf{G}_{k-1}^T + \mathbf{G}_{k-1} \mathbf{Q}_{k-1} \frac{\partial \mathbf{G}_{k-1}^T}{\partial \theta_i} \end{aligned} \tag{H.110}$$

$$\mathbf{M}(\widehat{\mathbf{x}}_{k|k}, \theta_i) = \mathbf{P}_{k|k-1}^{-1} + \mathbf{C}_k^T \mathbf{R}_k^{-1} \mathbf{C}_k \tag{H.111}$$

$$\begin{aligned} \mathbf{b}(\widehat{\mathbf{x}}_{k|k}, \theta_i) = & \mathbf{P}_{k|k-1}^{-1} \frac{\partial \mathbf{P}_{k|k-1}}{\partial \theta_i} \mathbf{P}_{k|k-1}^{-1} \big[\widehat{\mathbf{x}}_{k|k} - \widehat{\mathbf{x}}_{k|k-1} \big] + \mathbf{P}_{k|k-1}^{-1} \frac{\partial \widehat{\mathbf{x}}_{k|k-1}}{\partial \theta_i} \\ & - \mathbf{P}_{k|k-1}^{-1} \frac{\partial \mathbf{P}_{k|k-1}}{\partial \theta_i} \mathbf{P}_{k|k-1}^{-1} \mathbf{B}_k \mathbf{u}_k + \mathbf{P}_{k|k-1}^{-1} \frac{\partial \mathbf{B}_k}{\partial \theta_i} \mathbf{u}_k \\ & + \frac{\partial \mathbf{C}_k^T}{\partial \theta_i} \mathbf{R}_k^{-1} \big[\mathbf{y}_k - \mathbf{D}_k \mathbf{u}_k - \mathbf{C}_k \widehat{x}_{k|k} \big] + \mathbf{C}_k^T \frac{\partial \mathbf{R}_k^{-1}}{\partial \theta_i} \big[\mathbf{y}_k - \mathbf{D}_k \mathbf{u}_k - \mathbf{C}_k \widehat{\mathbf{x}}_{k|k} \big] \\ & - \mathbf{C}_k^T \mathbf{R}_k^{-1} \frac{\partial \mathbf{D}_k}{\partial \theta_i} \mathbf{u}_k - \mathbf{C}_k^T \mathbf{R}_k^{-1} \frac{\partial \mathbf{C}_k}{\partial \theta_i} \widehat{\mathbf{x}}_{k|k} \end{aligned} \tag{H.112}$$

$$\frac{\partial \widehat{\mathbf{x}}_{k|k}}{\partial \theta_i} = \mathbf{M}^{-1}(\widehat{\mathbf{x}}_{k|k}, \theta_i) \mathbf{b}(\widehat{\mathbf{x}}_{k|k}, \theta_i) \tag{H.113}$$

$$\begin{aligned} \frac{\partial \mathbf{P}_{k|k}}{\partial \theta_i} = & -\big[\mathbf{P}_{k|k-1}^{-1} + \mathbf{C}_k^T \mathbf{R}_k^{-1} \mathbf{C}_k \big]^{-1} \Big[- \mathbf{P}_{k|k-1}^{-1} \frac{\partial \mathbf{P}_{k|k-1}}{\partial \theta_i} \mathbf{P}_{k|k-1}^{-1} + \frac{\partial \mathbf{C}_k^T}{\partial \theta_i} \mathbf{R}_k^{-1} \mathbf{C}_k \\ & - \mathbf{C}_k^T \mathbf{R}_k^{-1} \frac{\partial \mathbf{R}_k}{\partial \theta_i} \mathbf{R}_k^{-1} \mathbf{C}_k + \mathbf{C}_k^T \mathbf{R}_k^{-1} \frac{\partial \mathbf{C}_k}{\partial \theta_i} \Big] \big[\mathbf{P}_{k|k-1}^{-1} + \mathbf{C}_k^T \mathbf{R}_k^{-1} \mathbf{C}_k \big]^{-1} \end{aligned} \tag{H.114}$$

H.8.5 Optimisation of $\boldsymbol{\theta}$

$$\boldsymbol{\theta}^{(l+1)} = \boldsymbol{\theta}^{(l)} - \rho_l \, \mathbf{R}^{(l)} \, \mathbf{g}^{(l)} \tag{H.115}$$

$$\mathbf{g}_i^{(l)} = -\frac{1}{2} tr\Big\{ \Big[\mathbf{\Sigma}_{k|k-1}^{-1} \frac{\partial \mathbf{\Sigma}_{k|k-1}}{\partial \theta_i} \Big] \Big[\mathbf{I} - \mathbf{\Sigma}_{k|k-1}^{-1} \mathbf{e}_k \mathbf{e}_k^T \Big] \Big\} - \frac{\partial \mathbf{e}_k^T}{\partial \theta_i} \mathbf{\Sigma}_{k|k-1}^{-1} \mathbf{e}_k \tag{H.116}$$

$$\begin{aligned} \widehat{\mathbf{R}}^{(l)}(i,j) = \sum_{k=1}^{N} \Big\{ & \big(\frac{\partial \mathbf{e}_k}{\partial \theta_i^{(l)}} \big)^T \mathbf{\Sigma}_{k|k-1}^{-1} \frac{\partial \mathbf{e}_k}{\partial \theta_j^{(l)}} \\ & + \frac{1}{2} tr \Big[\mathbf{\Sigma}_{k|k-1}^{-1} \frac{\partial \mathbf{\Sigma}_{k|k-1}}{\partial \theta_i^{(l)}} \mathbf{\Sigma}_{k|k-1}^{-1} \frac{\partial \mathbf{\Sigma}_{k|k-1}}{\partial \theta_j^{(l)}} \Big] \\ & + \frac{1}{4} tr \Big[\mathbf{\Sigma}_{k|k-1}^{-1} \frac{\partial \mathbf{\Sigma}_{k|k-1}}{\partial \theta_i^{(l)}} \Big] tr \Big[\mathbf{\Sigma}_{k|k-1}^{-1} \frac{\partial \mathbf{\Sigma}_{k|k-1}}{\partial \theta_j^{(l)}} \Big] \Big\} \end{aligned} \tag{H.117}$$

H.9 Parameter Estimation for a non Linear Model by Gauss-Newton

Algorithm H.9.1 (Parameter Estimation for a non Linear Model by Gauss-Newton)
.

H.9.1 System

Let consider the state space model :

$$\mathbf{x}_{k+1} = \mathbf{f}_k(\mathbf{x}_k, \boldsymbol{\theta}) + \mathbf{B}_k(\boldsymbol{\theta})\mathbf{u}_k + \mathbf{G}_k(\boldsymbol{\theta})\mathbf{v}_k \tag{H.118}$$

$$\mathbf{y}_k = \mathbf{h}_k(\mathbf{x}_k, \boldsymbol{\theta}) + \mathbf{n}_k \tag{H.119}$$

with nonlinear RBF functions in State and Measurement equations:

$$\mathbf{f}_k(\mathbf{x}_k) = \sum_{i=1}^{I} \lambda_{ik}^{(f)} \exp\big[-\frac{1}{\sigma_{ik}^{(f)}}(\mathbf{x}_k - \mathbf{c}_{ik}^{(f)})^T(\mathbf{x}_k - \mathbf{c}_{ik}^{(f)})\big] \tag{H.120}$$

$$\mathbf{h}_k(\mathbf{x}_k) = \sum_{j=1}^{J} \lambda_{jk}^{(h)} \exp\big[-\frac{1}{\sigma_{jk}^{(h)}}(\mathbf{x}_k - \mathbf{c}_{jk}^{(h)})^T(\mathbf{x}_k - \mathbf{c}_{jk}^{(h)})\big]. \tag{H.121}$$

where

- $\mathbf{x}_k \in \mathbb{R}^n$, $\mathbf{y}_k \in \mathbb{R}^q$, $\mathbf{u}_k \in \mathbb{R}^m$, $\mathbf{v}_k \in \mathbb{R}^p$, $\mathbf{n}_k \in \mathbb{R}^q$,
- $\mathbf{B}_k(\boldsymbol{\theta})$ $[n, m]$, $\mathbf{G}_k(\boldsymbol{\theta})$ $[n, p]$,
- *these system matrices depend on a set of unknown parameters* $\boldsymbol{\theta} \in \mathbb{R}^J$ *where* J *is the number of hyperparameters,*
- $\mathbf{f}_k$ *is a nonlinear function,*
- $\mathbf{h}_k$ *is a nonlinear function,*
- $\mathbf{x}_k$ *is the state at time* k*,*
- $\mathbf{y}_k$ *is the observation at time* k*,*
- $\{\mathbf{u}_k\}$ *a deterministic input at time* k*,*
- $\{\mathbf{v}_k\}$, $\{\mathbf{n}_k\}$ *the system, and measurement white noises, with* $\mathbf{v}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}(\boldsymbol{\theta}))$, $\mathbf{n}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{R}(\boldsymbol{\theta}))$,
- $\mathbf{x}_0 \sim \mathcal{N}(\overline{\mathbf{x}_0}, \mathbf{P}_0)$,
- I *is the number of neurons in the hidden layer for function* $\mathbf{f}_k$*,*
- $\mathbf{c}_{ik}^{(f)}$ *is the centroid vector for neuron* i*,*
- $\lambda_{ik}^{(f)}$ *are the weights of the linear output neuron,*
- $\sigma_{ik}^{(f)}$ *are the variances of clusters,*
- J *is the number of neurons in the hidden layer for function* $\mathbf{h}_k$*,*
- $\mathbf{c}_{jk}^{(h)}$ *is the centroid vector for neuron* j*,*
- $\lambda_{jk}^{(h)}$ *are the weights of the linear output neuron,*

- $\sigma_{jk}^{(h)}$ *are the variances of clusters.*

We realize the linearization of the model as follow:

$$\mathbf{f}_k(\mathbf{x}_k) \simeq \mathbf{f}_k(\widehat{\mathbf{x}}_k) + \mathbf{A}_k\big[\mathbf{x}_k - \widehat{\mathbf{x}}_k\big] \tag{H.122}$$

$$\mathbf{A}_k = \Big[\frac{\partial \mathbf{f}_k(\mathbf{x}_k)}{\partial \mathbf{x}_k}\Big]_{\widehat{\mathbf{x}}_k} \tag{H.123}$$

$$= -2\sum_{i=1}^{I} \frac{\lambda_{ik}^{(f)}}{\sigma_{ik}^{(f)}}[\widehat{\mathbf{x}}_k - \mathbf{c}_{ik}^{(f)}] \exp\big[-\frac{1}{\sigma_{ik}^{(f)}}(\widehat{\mathbf{x}}_k - \mathbf{c}_{ik}^{(f)})(\widehat{\mathbf{x}}_k - \mathbf{c}_{ik}^{(f)})^T\big] \tag{H.124}$$

$$\mathbf{h}_k(\mathbf{x}_k) \simeq \mathbf{h}_k(\widehat{\mathbf{x}}_k) + \mathbf{C}_k\big[\mathbf{x}_k - \widehat{\mathbf{x}}_k\big] \tag{H.125}$$

$$\mathbf{C}_k = \Big[\frac{\partial \mathbf{h}_k(\mathbf{x}_k)}{\partial \mathbf{x}_k}\Big]_{\widehat{\mathbf{x}}_k} \tag{H.126}$$

$$= -2\sum_{j=1}^{J} \frac{\lambda_{jk}^{(h)}}{\sigma_{jk}^{(h)}}[\widehat{\mathbf{x}}_k - \mathbf{c}_{jk}^{(h)}] \exp\big[-\frac{1}{\sigma_{jk}^{(h)}}(\widehat{\mathbf{x}}_k - \mathbf{c}_{jk}^{(h)})(\widehat{\mathbf{x}}_k - \mathbf{c}_{jk}^{(h)})^T\big], \tag{H.127}$$

We get the linearized model:

$$\mathbf{x}_k = \mathbf{A}_k\mathbf{x}_{k-1} + \mathbf{B}_k\mathbf{u}_k + \mathbf{v}_k + \mathbf{a}_k \tag{H.128}$$

$$\mathbf{y}_k = \mathbf{C}_k\mathbf{x}_k + \mathbf{n}_k + \mathbf{c}_k, \tag{H.129}$$

with "residues" of linearization:

$$\mathbf{a}_k = \mathbf{f}_k(\widehat{\mathbf{x}}_{k|k}) - \mathbf{A}_k\widehat{\mathbf{x}}_{k|k} \tag{H.130}$$

$$\mathbf{c}_k = \mathbf{h}_k(\widehat{\mathbf{x}}_{k|k-1}) - \mathbf{C}_k\widehat{\mathbf{x}}_{k|k-1}. \tag{H.131}$$

If Δt small, we have small "residues" of linearization, and states of nonlinear and linearized models are similar.

H.9.2 Extended Kalman filter

The Extended Kalman filter will be used in the derivatives of the likelihood in next sections. It is recalled here to make easier the understanding of the procedure.

Filtering *for $k = 1, 2, \cdots N$, (Forwards)*

Prediction

$$\mathbf{f}_k(\mathbf{x}_k) = \sum_{i=1}^{I} \lambda_{ik}^{(f)} \exp\big[-\frac{1}{\sigma_{ik}^{(f)}} (\mathbf{x}_k - \mathbf{c}_{ik}^{(f)})^T (\mathbf{x}_k - \mathbf{c}_{ik}^{(f)}) \big] \tag{H.132}$$

$$\widehat{\mathbf{x}}_{k|k-1} = \mathbf{f}_{k-1}(\widehat{\mathbf{x}}_{k-1|k-1}) \tag{H.133}$$

$$\mathbf{A}_{k-1} = \Big[\frac{\partial \mathbf{f}_k(\mathbf{x}_k)}{\partial \mathbf{x}_k} \Big]_{\widehat{\mathbf{x}}_{k-1|k-1}} \tag{H.134}$$

$$\mathbf{P}_{k|k-1} = \mathbf{A}_{k-1} \mathbf{P}_{k-1|k-1} \mathbf{A}_{k-1}^T + \mathbf{Q} \tag{H.135}$$

Update

$$\mathbf{h}_k(\mathbf{x}_k) = \sum_{i=1}^{I} \lambda_{ik}^{(h)} \exp\big[-\frac{1}{\sigma_{ik}^{(h)}} (\mathbf{x}_k - \mathbf{c}_{ik}^{(h)})^T (\mathbf{x}_k - \mathbf{c}_{ik}^{(h)}) \big] \tag{H.136}$$

$$\mathbf{C}_k = \Big[\frac{\partial \mathbf{h}_k(\mathbf{x}_k)}{\partial \mathbf{x}_k} \Big]_{\widehat{\mathbf{x}}_{k|k-1}} \tag{H.137}$$

$$\boldsymbol{\Sigma}_{k|k-1} = \mathbf{C}_k \mathbf{P}_{k|k-1} \mathbf{C}_k^T + \mathbf{R} \tag{H.138}$$

$$\mathbf{K}_k = \mathbf{P}_{k|k-1} \mathbf{C}_k^T \boldsymbol{\Sigma}_{k|k-1}^{-1} \tag{H.139}$$

$$\mathbf{e}_k = \mathbf{y}_k - \mathbf{h}_k(\widehat{\mathbf{x}}_{k|k-1}) \tag{H.140}$$

$$\widehat{\mathbf{x}}_{k|k} = \widehat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k \mathbf{e}_k \tag{H.141}$$

$$\mathbf{P}_{k|k} = \mathbf{P}_{k|k-1} - \mathbf{K}_k \mathbf{C}_k \mathbf{P}_{k|k-1} \tag{H.142}$$

H.9.3 Likelihood

$$\log L = \sum_{k=1}^{N} \big\{ -\frac{N_k}{2} \log(2\pi) - \frac{1}{2} \log \mid \boldsymbol{\Sigma}_{k|k-1} \mid -\frac{1}{2} \sum_{k=1}^{N} (\mathbf{e}_k^T \boldsymbol{\Sigma}_{k|k-1}^{-1} \mathbf{e}_k) \big\} \tag{H.143}$$

the k-ieme contribution of the log-likelihood is given by :

$$l_k = -\frac{N_k}{2} \log(2\pi) - \frac{1}{2} \log \mid \boldsymbol{\Sigma}_{k|k-1} \mid -\frac{1}{2} (\mathbf{e}_k^T \boldsymbol{\Sigma}_{k|k-1}^{-1} \mathbf{e}_k) \tag{H.144}$$

$$\begin{aligned} l_k = &-\frac{N_k}{2} \log(2\pi) - \frac{1}{2} \log |\mathbf{R}_k| - \frac{1}{2} \log |\mathbf{P}_{k|k-1}| - \frac{1}{2} \log |\mathbf{P}_{k|k-1}^{-1} + \mathbf{C}_k^T \mathbf{R}_k^{-1} \mathbf{C}_k| \\ &-\frac{1}{2} tr\big[\mathbf{R}_k^{-1} \mathbf{e}_k \mathbf{e}_k^T\big] + \frac{1}{2} \big[\mathbf{C}_k^T \mathbf{R}_k^{-1} \mathbf{e}_k\big]^T \big[\mathbf{P}_{k|k-1}^{-1} + \mathbf{C}_k^T \mathbf{R}_k^{-1} \mathbf{C}_k\big]^{-1} \big[\mathbf{C}_k^T \mathbf{R}_k^{-1} \mathbf{e}_k\big] \end{aligned} \tag{H.145}$$

H.9.4 Derivation of the likelihood

$$\frac{\partial l_k}{\partial \theta_i} = -\frac{1}{2} tr\Big\{ \Big[\mathbf{\Sigma}_{k|k-1}^{-1} \frac{\partial \mathbf{\Sigma}_{k|k-1}}{\partial \theta_i} \Big] \Big[\mathbf{I} - \mathbf{\Sigma}_{k|k-1}^{-1} \mathbf{e}_k \mathbf{e}_k^T \Big] \Big\} - \frac{\partial \mathbf{e}_k^T}{\partial \theta_i} \mathbf{\Sigma}_{k|k-1}^{-1} \mathbf{e}_k \tag{H.146}$$

the derivation of the log-likelihood is:

$$\frac{\partial \log L}{\partial \theta_i} = -\sum_{k=1}^{N} \left\{ \frac{1}{2} tr\Big\{ \Big[\mathbf{\Sigma}_{k|k-1}^{-1} \frac{\partial \mathbf{\Sigma}_{k|k-1}}{\partial \theta_i} \Big] \Big[\mathbf{I} - \mathbf{\Sigma}_{k|k-1}^{-1} \mathbf{e}_k \mathbf{e}_k^T \Big] \Big\} + \frac{\partial \mathbf{e}_k^T}{\partial \theta_i} \mathbf{\Sigma}_{k|k-1}^{-1} \mathbf{e}_k \right\} \tag{H.147}$$

for $i \in \{1, J\}$, where J is the number of hyperparameters.

Estimation of $\frac{\partial \mathbf{e}_k}{\partial \theta_i}$

$$\frac{\partial \mathbf{e}_k}{\partial \theta_i} = -\frac{\partial \mathbf{h}_k(\mathbf{X}, \theta)}{\partial \theta_i}\Big|_{\widehat{\mathbf{x}}_{k|k-1}} - \mathbf{C}_k(\widehat{\mathbf{x}}_{k|k-1}) \frac{\partial \mathbf{X}(\theta)}{\partial \theta_i}\Big|_{\widehat{\mathbf{x}}_{k|k-1}} \tag{H.148}$$

Estimation of $\frac{\partial \mathbf{\Sigma}_{k|k-1}}{\partial \theta_i}$

$$\begin{aligned} \frac{\partial \mathbf{\Sigma}_{k|k-1}}{\partial \theta_i} = & \Big[\frac{\partial \mathbf{C}_k(\mathbf{X}, \theta)}{\partial \theta_i} + \frac{\partial \mathbf{C}_k(\mathbf{X}(\theta), \theta)}{\partial \mathbf{X^T}} \frac{\partial \mathbf{X}(\theta)}{\partial \theta_i} \Big]_{\widehat{\mathbf{x}}_{k|k-1}} \mathbf{P}_{k|k-1} \mathbf{C}_k^T \\ & + \mathbf{C}_k(\widehat{\mathbf{x}}_{k|k-1}, \theta) \frac{\partial \mathbf{P}_{k|k-1}}{\partial \theta_i} \mathbf{C}_k^T(\widehat{\mathbf{x}}_{k|k-1}, \theta) \\ & + \mathbf{C}_k(\widehat{\mathbf{x}}_{k|k-1}, \theta) \mathbf{P}_{k|k-1} \Big[\frac{\partial \mathbf{C}_k(\mathbf{X}, \theta)}{\partial \theta_i} + \frac{\partial \mathbf{C}_k(\mathbf{X}(\theta), \theta)}{\partial \mathbf{X^T}} \frac{\partial \mathbf{X}(\theta)}{\partial \theta_i} \Big]_{\widehat{\mathbf{x}}_{k|k-1}}^T + \frac{\partial \mathbf{R}_k}{\partial \theta_i} \end{aligned} \tag{H.149}$$

H.9.5 Derivation of the prediction

$$\frac{\partial \widehat{\mathbf{x}}_{k|k-1}}{\partial \theta_i} = \frac{\partial \mathbf{A}_{k-1}}{\partial \theta_i} \widehat{\mathbf{x}}_{k-1|k-1} + \mathbf{A}_{k-1} \frac{\partial \widehat{\mathbf{x}}_{k-1|k-1}}{\partial \theta_i} + \frac{\partial \mathbf{B}_{k-1}}{\partial \theta_i} \mathbf{u}_{k-1} \tag{H.150}$$

$$\begin{aligned} \frac{\partial \mathbf{P}_{k|k-1}}{\partial \theta_i} = & \frac{\partial \mathbf{A}_{k-1}}{\partial \theta_i} \mathbf{P}_{k-1|k-1} \mathbf{A}_{k-1}^T + \mathbf{A}_{k-1} \frac{\partial \mathbf{P}_{k-1|k-1}}{\partial \theta_i} \mathbf{A}_{k-1}^T \\ & + \mathbf{A}_{k-1} \mathbf{P}_{k-1|k-1} \frac{\partial \mathbf{A}_{k-1}^T}{\partial \theta_i} + \frac{\partial \mathbf{G}_{k-1}}{\partial \theta_i} \mathbf{Q}_{k-1} \mathbf{G}_{k-1}^T \\ & + \mathbf{G}_{k-1} \frac{\partial \mathbf{Q}_{k-1}}{\partial \theta_i} \mathbf{G}_{k-1}^T + \mathbf{G}_{k-1} \mathbf{Q}_{k-1} \frac{\partial \mathbf{G}_{k-1}^T}{\partial \theta_i} \end{aligned} \tag{H.151}$$

Estimation of $\widehat{\mathbf{x}}_{k|k}$

We use the iterative Gauss-Newton procedure:

$$\widehat{\mathbf{x}}_{k|k}^{(j+1)} = \widehat{\mathbf{x}}_{k|k}^{(j)} - \rho_j \mathbf{R}_j \mathbf{g}_j \tag{H.152}$$

with

$$\mathbf{R}_j = \Big[\mathbf{P}_{k|k-1}^{-1} + \mathbf{C}_k^T(\widehat{\mathbf{x}}_{k|k}^{(j)})\mathbf{R}_k^{-1}\mathbf{C}_k(\widehat{\mathbf{x}}_{k|k}^{(j)})\Big]^{-1} \tag{H.153}$$

$$\mathbf{g}_j = \mathbf{P}_{k|k}^{-1}\big[\widehat{\mathbf{x}}_{k|k}^{(j)} - \widehat{\mathbf{x}}_{k|k-1}\big] - \mathbf{C}_k^T(\widehat{\mathbf{x}}_{k|k}^{(j)})\mathbf{R}_k^{-1}\big[\mathbf{y}_k - \mathbf{h}_k(\widehat{\mathbf{x}}_{k|k}^{(j)})\big] \tag{H.154}$$

We initialize the procedure with $\widehat{\mathbf{x}}_{k|k}^{(0)} = \widehat{\mathbf{x}}_{k|k-1}$ given by the prediction step.

Estimation of $\frac{\partial \widehat{\mathbf{x}}_{k|k}}{\partial \theta_i}$

$$\mathbf{M}(\widehat{x}_{k|k},\theta) = \mathbf{P}_{k|k-1}^{-1} + \mathbf{C}_k^T(\widehat{\mathbf{x}}_{k|k},\theta)\mathbf{R}_k^{-1}\mathbf{C}_k(\widehat{\mathbf{x}}_{k|k},\theta) \tag{H.155}$$

$$\begin{aligned}\mathbf{b}(\widehat{x}_{k|k},\theta) = {} & \mathbf{P}_{k|k-1}^{-1}\frac{\partial \mathbf{P}_{k|k-1}}{\partial \theta_i}\mathbf{P}_{k|k-1}^{-1}\Big[\widehat{\mathbf{x}}_{k|k} - \widehat{\mathbf{x}}_{k|k-1}\Big] \\ & + \mathbf{P}_{k|k-1}^{-1}\frac{\partial \widehat{\mathbf{x}}_{k|k-1}}{\partial \theta_i} + \frac{\partial \mathbf{C}_k^T(\widehat{\mathbf{x}}_{k|k},\theta)}{\partial \theta_i}\mathbf{R}_k^{-1}\Big[\mathbf{y}_k - \mathbf{h}_k(\widehat{\mathbf{x}}_{k|k})\Big] \\ & - \mathbf{C}_k^T(\widehat{\mathbf{x}}_{k|k},\theta)\mathbf{R}_k^{-1}\frac{\partial \mathbf{R}_k^{-1}}{\partial \theta_i}\mathbf{R}_k^{-1}\Big[\mathbf{y}_k - \mathbf{h}_k(\widehat{\mathbf{x}}_{k|k})\Big] \\ & - \mathbf{C}_k^T(\widehat{\mathbf{x}}_{k|k},\theta)\mathbf{R}_k^{-1}\frac{\partial \mathbf{h}_k(\mathbf{X},\theta)}{\partial \theta_i}\Big|_{\widehat{x}_{k|k}}\end{aligned} \tag{H.156}$$

$$\frac{\partial \widehat{\mathbf{x}}_{k|k}}{\partial \theta_i} = \mathbf{M}(\widehat{x}_{k|k},\theta)^{-1}\mathbf{b}(\widehat{x}_{k|k},\theta) \tag{H.157}$$

Estimation of $\frac{\partial \mathbf{P}_{k|k}}{\partial \theta_i}$

$$\begin{aligned}\frac{\partial \mathbf{P}_{k|k}}{\partial \theta_i} = {} & -\Big[\mathbf{P}_{k|k-1}^{-1} + \mathbf{C}_k^T\mathbf{R}_k^{-1}\mathbf{C}_k\Big]^{-1} \\ & \times \Big\{ -\mathbf{P}_{k|k-1}^{-1}\frac{\partial \mathbf{P}_{k|k-1}}{\partial \theta_i}\mathbf{P}_{k|k-1}^{-1} - \mathbf{C}_k^T\mathbf{R}_k^{-1}\frac{\partial \mathbf{R}_k}{\partial \theta_i}\mathbf{R}_k^{-1}\mathbf{C}_k \\ & \quad + \Big[\frac{\partial \mathbf{C}_k^T(\widehat{\mathbf{x}}_{k|k-1},\theta)}{\partial \theta_i} + \frac{\partial \mathbf{C}_k(\mathbf{X}(\theta),\theta)}{\partial \mathbf{X}^T}\frac{\partial \mathbf{X}(\theta)}{\partial \theta_i}\Big|_{\widehat{x}_{k|k-1}}\Big]\mathbf{R}_k^{-1}\mathbf{C}_k(\widehat{\mathbf{x}}_{k|k-1},\theta) \\ & \quad + \mathbf{C}_k^T(\widehat{\mathbf{x}}_{k|k-1},\theta)\mathbf{R}_k^{-1}\Big[\frac{\partial \mathbf{C}_k(\widehat{\mathbf{x}}_{k|k-1},\theta)}{\partial \theta_i} + \frac{\partial \mathbf{C}_k(\mathbf{X}(\theta),\theta)}{\partial \mathbf{X}^T}\frac{\partial \mathbf{X}(\theta)}{\partial \theta_i}\Big|_{\widehat{\mathbf{x}}_{k|k-1}}\Big] \quad \Big\} \\ & \times \quad \Big[\mathbf{P}_{k|k-1}^{-1} + \mathbf{C}_k^T\mathbf{R}_k^{-1}\mathbf{C}_k\Big]^{-1}\end{aligned} \tag{H.158}$$

H.9.6 Optimisation of $\boldsymbol{\theta}$

$$\boldsymbol{\theta}^{(l+1)} = \boldsymbol{\theta}^{(l)} - \rho_l \, \mathbf{R}^{(l)} \, \mathbf{g}^{(l)} \tag{H.159}$$

$$\mathbf{g}_i^{(l)} = -\frac{1}{2} \, tr\Big\{ \Big[\boldsymbol{\Sigma}_{k|k-1}^{-1} \frac{\partial \boldsymbol{\Sigma}_{k|k-1}}{\partial \theta_i} \Big] \Big[\mathbf{I} - \boldsymbol{\Sigma}_{k|k-1}^{-1} \mathbf{e}_k \mathbf{e}_k^T \Big] \Big\} - \frac{\partial \mathbf{e}_k^T}{\partial \theta_i} \boldsymbol{\Sigma}_{k|k-1}^{-1} \mathbf{e}_k \tag{H.160}$$

$$\begin{aligned} \widehat{\mathbf{R}}^{(l)}(i,j) = \sum_{k=1}^{N} \Big\{ \big(\frac{\partial \mathbf{e}_k}{\partial \theta_i^{(l)}} \big)^T \boldsymbol{\Sigma}_{k|k-1}^{-1} \frac{\partial \mathbf{e}_k}{\partial \theta_j^{(l)}} + \frac{1}{2} \, tr\Big[\boldsymbol{\Sigma}_{k|k-1}^{-1} \frac{\partial \boldsymbol{\Sigma}_{k|k-1}}{\partial \theta_i^{(l)}} \boldsymbol{\Sigma}_{k|k-1}^{-1} \frac{\partial \boldsymbol{\Sigma}_{k|k-1}}{\partial \theta_j^{(l)}} \Big] \\ + \frac{1}{4} \, tr\Big[\boldsymbol{\Sigma}_{k|k-1}^{-1} \frac{\partial \boldsymbol{\Sigma}_{k|k-1}}{\partial \theta_i^{(l)}} \Big] tr\Big[\boldsymbol{\Sigma}_{k|k-1}^{-1} \frac{\partial \boldsymbol{\Sigma}_{k|k-1}}{\partial \theta_j^{(l)}} \Big] \Big\} \end{aligned} \tag{H.161}$$

H.10 Parameter Estimation for Linear Model by EM algorithm

Algorithm H.10.1 (Parameter Estimation for Linear Model by EM algorithm)
.

H.10.1 State Space Model

$$\mathbf{x}_k = \mathbf{A}(\theta)\mathbf{x}_{k-1} + \mathbf{v}_k \tag{H.162}$$
$$\mathbf{y}_k = \mathbf{C}(\theta)\mathbf{x}_k + \mathbf{n}_k \tag{H.163}$$

where:

- $\mathbf{x}_k \in \mathbb{R}^n$, $\mathbf{y}_k \in \mathbb{R}^q$, $\mathbf{v}_k \in \mathbb{R}^p$, $\mathbf{n}_k \in \mathbb{R}^q$,
- $\mathbf{A}_k(\boldsymbol{\theta})$ $[n, n]$, $\mathbf{C}_k(\boldsymbol{\theta})$ $[q, n]$,
- *these system matrices depend on a set of unknown parameters* $\boldsymbol{\theta} \in \mathbb{R}^J$ *where* J *is the number of hyperparameters,*
- $\mathbf{x}_k$ *is the state,*
- $\mathbf{y}_k$ *is the observation,*
- *Eq.* (H.162) *is the state equation,*
- *Eq.* (H.163) *is the measurement equation,*
- $\mathbf{v}_k$ *is the process noise, with* $\mathbf{v}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}(\boldsymbol{\theta}))$,
- $\mathbf{n}_k$ *is the measurement noise, with* $\mathbf{n}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{R}(\boldsymbol{\theta}))$.
- $\mathbf{x}_0 \sim N(\overline{\mathbf{x}_0}, \mathbf{P}_0)$.

H.10.2 Kalman Filter

Initialization

$$\widehat{\mathbf{x}}_{0|-1} = E\{\mathbf{x}_0\} = \overline{\mathbf{x}_0} \tag{H.164}$$
$$\mathbf{P}_{0|-1} = Var\{\mathbf{x}_0\} \tag{H.165}$$

for $k = 1, 2, \cdots N$ *(Forwards)*

Prediction

$$\widehat{\mathbf{x}}_{k|k-1} = \mathbf{A}\widehat{\mathbf{x}}_{k-1|k-1} \tag{H.166}$$
$$\mathbf{P}_{k|k-1} = \mathbf{A}\mathbf{P}_{k-1|k-1}\mathbf{A}^T + \mathbf{Q} \tag{H.167}$$

Update

$$\boldsymbol{\Sigma}_{k|k-1} = \mathbf{C}\mathbf{P}_{k|k-1}\mathbf{C}^T + \mathbf{R} \tag{H.168}$$
$$\mathbf{K}_k = \mathbf{P}_{k|k-1}\mathbf{C}^T\boldsymbol{\Sigma}^{-1}_{k|k-1} \tag{H.169}$$
$$\mathbf{e}_k = \mathbf{y}_k - \mathbf{C}\widehat{\mathbf{x}}_{k|k-1} \tag{H.170}$$
$$\widehat{\mathbf{x}}_{k|k} = \widehat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k\mathbf{e}_k \tag{H.171}$$
$$\mathbf{P}_{k|k} = \mathbf{P}_{k|k-1} - \mathbf{K}_k\mathbf{C}\mathbf{P}_{k|k-1} \tag{H.172}$$

H.10.3 Likelihood

$$\log L_N = -\frac{Nq}{2}\log(2\pi) - \frac{1}{2}\sum_{k=1}^{N}\log \mid \boldsymbol{\Sigma}_{k|k-1} \mid -\frac{1}{2}\sum_{k=1}^{N}(\mathbf{e}_k^T\boldsymbol{\Sigma}^{-1}_{k|k-1}\mathbf{e}_k) \tag{H.173}$$

H.10.4 Learning

In the learning step we estimate the parameters by EM-algorithm.

E-step

Forward Procedure

for each $k = 1, \cdots, N$ *we calculate :*

$$\boldsymbol{\Sigma}_{k|k-1} = \mathbf{C}\mathbf{P}_{k|k-1}\mathbf{C}^T + \mathbf{R} \tag{H.174}$$
$$\mathbf{K}_k = \mathbf{P}_{k|k-1}\mathbf{C}^T\boldsymbol{\Sigma}^{-1}_{k|k-1} \tag{H.175}$$
$$\mathbf{e}_k = \mathbf{y}_k - \mathbf{C}\widehat{\mathbf{x}}_{k|k-1} \tag{H.176}$$
$$\widehat{\mathbf{x}}_{k|k} = \widehat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k\mathbf{e}_k \tag{H.177}$$
$$\mathbf{P}_{k|k} = \mathbf{P}_{k|k-1} - \mathbf{K}_k\mathbf{C}\mathbf{P}_{k|k-1} \tag{H.178}$$
$$\widehat{\mathbf{x}}_{k+1|k} = \mathbf{A}\widehat{\mathbf{x}}_{k|k} \tag{H.179}$$
$$\mathbf{P}_{k+1|k} = \mathbf{A}\mathbf{P}_{k|k}\mathbf{A}^T + \mathbf{Q} \tag{H.180}$$
$$\mathbf{P}_{k,k-1|k} = \big[\mathbf{I} - \mathbf{K}_k\mathbf{C}\big]\mathbf{A}\mathbf{P}_{k-1|k-1} \tag{H.181}$$

Backward Procedure

for each $k = N, \cdots, 1$ *we calculate :*

$$\widehat{\mathbf{x}}_{k-1|N} = \widehat{\mathbf{x}}_{k-1|k-1} + \mathbf{J}_{k-1}\big[\widehat{\mathbf{x}}_{k|N} - \widehat{\mathbf{x}}_{k|k-1}\big] \tag{H.182}$$
$$\mathbf{P}_{k-1|N} = \mathbf{P}_{k-1|k-1} + \mathbf{J}_{k-1}\big[\mathbf{P}_{k|N} - \mathbf{P}_{k|k-1}\big]\mathbf{J}^T_{k-1} \tag{H.183}$$
$$\mathbf{J}_{k-1} = \mathbf{P}_{k-1|k-1}\mathbf{A}^T\mathbf{P}^T_{k|k-1} \tag{H.184}$$
$$\mathbf{P}_{k,k-1|N} = \mathbf{P}_{k,k-1|k} + \big[\mathbf{P}_{k|N} - \mathbf{P}_{k|k}\big]\mathbf{P}^{-1}_{k|k}\mathbf{P}_{k,k-1|k} \tag{H.185}$$

M-step

The sufficient statistics are given by:

$$\Gamma_1 = \frac{1}{N+1}\sum_{k=0}^{N}\left[\mathbf{P}_{k|N} + \widehat{\mathbf{x}}_{k|N}\widehat{\mathbf{x}}_{k|N}^T\right] \tag{H.186}$$

$$\Gamma_2 = \frac{1}{N}\sum_{k=1}^{N}\left[\mathbf{P}_{k|N} + \widehat{\mathbf{x}}_{k|N}\widehat{\mathbf{x}}_{k|N}^T\right] \tag{H.187}$$

$$\Gamma_3 = \frac{1}{N}\sum_{k=1}^{N}\left[\mathbf{P}_{k-1|N} + \widehat{\mathbf{x}}_{k-1|N}\widehat{\mathbf{x}}_{k-1|N}^T\right] \tag{H.188}$$

$$\Gamma_4 = \frac{1}{N}\sum_{k=1}^{N}\left[\mathbf{P}_{k,k-1|N} + \widehat{\mathbf{x}}_{k|N}\widehat{\mathbf{x}}_{k-1|N}^T\right] \tag{H.189}$$

$$\Gamma_5 = \frac{1}{N+1}\sum_{k=0}^{N}\left[\mathbf{y}_k\mathbf{y}_k^T\right] \tag{H.190}$$

$$\Gamma_6 = \frac{1}{N+1}\sum_{k=0}^{N}\left[\mathbf{y}_k\widehat{\mathbf{x}}_{k|N}^T\right] \tag{H.191}$$

the parameter estimated are given by:

$$\widehat{\mathbf{A}} = \Gamma_4\Gamma_3^{-1} \tag{H.192}$$

$$\widehat{\mathbf{C}} = \Gamma_6\Gamma_1^{-1} \tag{H.193}$$

$$\widehat{\mathbf{Q}} = \Gamma_2 - \widehat{\mathbf{A}}\Gamma_4^T \tag{H.194}$$

$$\widehat{\mathbf{R}} = \Gamma_5 - \widehat{\mathbf{C}}\Gamma_6^T \tag{H.195}$$

$$\widehat{\overline{\mathbf{x}}_{\mathbf{0}}} = \widehat{\mathbf{x}}_{0|N} \tag{H.196}$$

$$\widehat{\mathbf{P}}_0 = \mathbf{P}_{0|N} \tag{H.197}$$

H.10.5 Inference

We use the EKF for filtering and prediction to estimate the state values at each time step, as follow:

Filtering

for $k = 1, 2, \cdots N$, (Forwards)

Prediction

$$\widehat{\mathbf{x}}_{k|k-1} = \mathbf{A}\widehat{\mathbf{x}}_{k-1|k-1} \tag{H.198}$$
$$\mathbf{P}_{k|k-1} = \mathbf{A}\mathbf{P}_{k-1|k-1}\mathbf{A}^T + \mathbf{Q} \tag{H.199}$$

Update

$$\mathbf{\Sigma}_{k|k-1} = \mathbf{C}\mathbf{P}_{k|k-1}\mathbf{C}^T + \mathbf{R} \tag{H.200}$$
$$\mathbf{K}_k = \mathbf{P}_{k|k-1}\mathbf{C}^T\mathbf{\Sigma}_{k|k-1}^{-1} \tag{H.201}$$
$$\mathbf{e}_k = \mathbf{y}_k - \mathbf{C}\widehat{\mathbf{x}}_{k|k-1} \tag{H.202}$$
$$\widehat{\mathbf{x}}_{k|k} = \widehat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k\mathbf{e}_k \tag{H.203}$$
$$\mathbf{P}_{k|k} = \mathbf{P}_{k|k-1} - \mathbf{K}_k\mathbf{C}\mathbf{P}_{k|k-1} \tag{H.204}$$

Conclusion

The learning procedure, gives the parameters:

$$\widehat{\mathbf{A}} = \mathit{\Gamma}_4\mathit{\Gamma}_3^{-1} \tag{H.205}$$
$$\widehat{\mathbf{C}} = \mathit{\Gamma}_6\mathit{\Gamma}_1^{-1} \tag{H.206}$$
$$\widehat{\mathbf{Q}} = \mathit{\Gamma}_2 - \widehat{\mathbf{A}}\mathit{\Gamma}_4^T \tag{H.207}$$
$$\widehat{\mathbf{R}} = \mathit{\Gamma}_5 - \widehat{\mathbf{C}}\mathit{\Gamma}_6^T \tag{H.208}$$
$$\widehat{\overline{\mathbf{x}}_{\mathbf{0}}} = \widehat{\mathbf{x}}_{0|N} \tag{H.209}$$
$$\widehat{\mathbf{P}}_0 = \mathbf{P}_{0|N} \tag{H.210}$$

The inference procedure gives the states:

$$\{\mathbf{x}_{0|N}, \mathbf{x}_{1|N}, \mathbf{x}_{2|N}, \cdots, \mathbf{x}_{N-1|N}, \mathbf{x}_{N|N}\}$$

H.11 Parameter Estimation for Non Linear Model by EM algorithm

Algorithm H.11.1 (Parameter Estimation for Non Linear Model by EM algorithm)
.

H.11.1 State Space Model

$$\mathbf{x}_k = \mathbf{f}(\mathbf{x}_{k-1}, \theta) + \mathbf{v}_k \tag{H.211}$$
$$\mathbf{y}_k = \mathbf{h}(\mathbf{x}_k, \theta) + \mathbf{n}_k \tag{H.212}$$

with RBF in State and Measurement equations, without reference to θ to simplify:

$$\mathbf{f}(\mathbf{x}_k) = \sum_{i=1}^{I} \lambda_i^{(f)} \exp\big[-\frac{1}{\sigma_i^{(f)}}(\mathbf{x}_k - \mathbf{c}_i^{(f)})^T(\mathbf{x}_k - \mathbf{c}_i^{(f)})\big] \tag{H.213}$$

$$\mathbf{h}(\mathbf{x}_k) = \sum_{j=1}^{J} \lambda_j^{(h)} \exp\big[-\frac{1}{\sigma_j^{(h)}}(\mathbf{x}_k - \mathbf{c}_j^{(h)})^T(\mathbf{x}_k - \mathbf{c}_j^{(h)})\big] \tag{H.214}$$

where

- $\mathbf{x}_k \in R^n$, $\mathbf{y}_k \in R^q$, $\mathbf{v}_k \in R^n$, $\mathbf{n}_k \in R^q$,
- $\mathbf{x}_k$ *is the state, without economic signification,*
- $\mathbf{y}_k$ *is the observation, the spline coefficient vector of the smoothing of past raw three hours data,*
- *Eq.* (G.1) *is the state equation,*
- *Eq.* (G.2) *is the measurement equation,*
- $\mathbf{v}_k$ *is the process noise, with* $\mathbf{v}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{Q})$,
- $\mathbf{n}_k$ *is the measurement noise, with* $\mathbf{n}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{R})$.
- $\mathbf{x}_0 \sim N(\overline{\mathbf{x}_0}, \mathbf{P}_0)$,
- I *is the number of neurons in the hidden layer for function* $\mathbf{f}_k$,
- $\mathbf{c}_{ik}^{(f)}$) *is the centroid vector for neuron* i,
- $\lambda_{ik}^{(f)}$ *are the weights of the linear output neuron,*
- $\sigma_{ik}^{(f)}$ *are the variances of clusters,*
- J *is the number of neurons in the hidden layer for function* $\mathbf{h}_k$,
- $\mathbf{c}_{jk}^{(h)}$) *is the centroid vector for neuron* i,
- $\lambda_{jk}^{(h)}$ *are the weights of the linear output neuron,*
- $\sigma_{jk}^{(h)}$ *are the variances of clusters.*
- θ *is an hyperparameter.*

H.11.2 Linearization

Linearization of the model:

$$\mathbf{f}(\mathbf{x}_k) \simeq \mathbf{f}(\widehat{\mathbf{x}}_{k|N}) + \mathbf{A}\big[\mathbf{x}_k - \widehat{\mathbf{x}}_{k|N}\big] \tag{H.215}$$

$$\mathbf{A} = \Big[\frac{\partial \mathbf{f}(\mathbf{x}_k)}{\partial \mathbf{x}_k}\Big]_{\widehat{\mathbf{x}}_{k|N}} \tag{H.216}$$

$$= -2\sum_{i=1}^{I} \frac{\lambda_i^{(f)}}{\sigma_i^{(f)}}[\widehat{\mathbf{x}}_{k|N} - \mathbf{c}_i^{(f)}] \exp\big[-\frac{1}{\sigma_i^{(f)}}(\widehat{\mathbf{x}}_{k|N} - \mathbf{c}_i^{(f)})(\widehat{\mathbf{x}}_{k|N} - \mathbf{c}_i^{(f)})^T\big] \tag{H.217}$$

$$\mathbf{h}(\mathbf{x}_k) \simeq \mathbf{h}(\widehat{\mathbf{x}}_{k|N}) + \mathbf{C}\big[\mathbf{x}_k - \widehat{\mathbf{x}}_{k|N}\big] \tag{H.218}$$

$$\mathbf{C} = \Big[\frac{\partial \mathbf{h}(\mathbf{x}_k)}{\partial \mathbf{x}_k}\Big]_{\widehat{\mathbf{x}}_{k|N}} \tag{H.219}$$

$$= -2\sum_{j=1}^{J} \frac{\lambda_j^{(h)}}{\sigma_j^{(h)}}[\widehat{\mathbf{x}}_{k|N} - \mathbf{c}_j^{(h)}] \exp\big[-\frac{1}{\sigma_j^{(h)}}(\widehat{\mathbf{x}}_{k|N} - \mathbf{c}_j^{(h)})(\widehat{\mathbf{x}}_{k|N} - \mathbf{c}_j^{(h)})^T\big] \tag{H.220}$$

H.11.3 Gaussian distributions

Let suppose white noises for state and measurements.

$$p(\mathbf{x}_0|\theta) = \frac{1}{(2\pi)^{\frac{N}{2}}} \frac{1}{|\mathbf{P}_0|^{\frac{1}{2}}} \exp\Big\{(\mathbf{x}_0 - \overline{\mathbf{x}}_0)^T \mathbf{P}_0^{-1}(\mathbf{x}_0 - \overline{\mathbf{x}}_0)\Big\} \tag{H.221}$$

$$p(\mathbf{x}_k|\mathbf{x}_{k-1}|\theta) = \frac{1}{(2\pi)^{\frac{N}{2}}} \frac{1}{|\mathbf{Q}|^{\frac{1}{2}}} \exp\Big\{\big[\mathbf{x}_k - \widehat{\mathbf{f}}(\mathbf{x}_k)\big]^T \mathbf{Q}^{-1}\big[\mathbf{x}_k - \widehat{\mathbf{f}}(\mathbf{x}_k)\big]\Big\} \tag{H.222}$$

$$p(\mathbf{y}_k|\mathbf{x}_k|\theta) = \frac{1}{(2\pi)^{\frac{N}{2}}} \frac{1}{|\mathbf{R}|^{\frac{1}{2}}} \exp\Big\{\big[\mathbf{y}_k - \widehat{\mathbf{h}}(\mathbf{x}_k)\big]^T \mathbf{R}^{-1}\big[\mathbf{y}_k - \widehat{\mathbf{h}}(\mathbf{x}_k)\big]\Big\} \tag{H.223}$$

where $\overline{\mathbf{x}}_0$ *and* $\mathbf{P}_0$ *are the initial state and covariance.*

H.11.4 Likelihood

The joint distribution is given by:

$$p(\mathbf{x}_{0:N}, \mathbf{y}_{1:N}|\theta) = p(\mathbf{x}_0|\theta) \prod_{k=1}^{N} p(\mathbf{x}_k|\mathbf{x}_{k-1}, \theta) \prod_{k=0}^{N} p(\mathbf{y}_k|\mathbf{x}_k, \theta) \tag{H.224}$$

The likelihood is given by

$$
\begin{aligned}
J(\theta|\mathbf{x}_{0:N}, \mathbf{y}_{1:N}) = \\
&(\mathbf{x}_0 - \overline{\mathbf{x}}_0)^T \mathbf{P}_0^{-1} (\mathbf{x}_0 - \overline{\mathbf{x}}_0) + \log ||\mathbf{P}_0|| \\
+ &\sum_{k=1}^{N} \Big\{ \log \|\mathbf{Q}\| + (\mathbf{x}_k - \widehat{\mathbf{f}}(\mathbf{x}_k))^T \mathbf{Q}^{-1} (\mathbf{x}_k - \widehat{\mathbf{f}}(\mathbf{x}_k)) \Big\} \\
+ &\sum_{k=0}^{N} \Big[\log \|\mathbf{R}\| + [\mathbf{y}_k - \widehat{\mathbf{h}}(\mathbf{x}_k)]^T \mathbf{R}^{-1} [\mathbf{y}_k - \widehat{\mathbf{h}}(\mathbf{x}_k)] \Big]
\end{aligned} \tag{H.225}
$$

We have to maximize:

$$
\begin{aligned}
&E\big\{ J\big(\theta^{(p+1)}|\mathbf{x}_{0:N}, \mathbf{y}_{1:N}\big) \big\} = \\
&(N+1) \log ||\mathbf{R}|| + N \log ||\mathbf{Q}|| + \log ||\mathbf{P}_0|| \\
+ &\sum_{k=1}^{N} tr\Big[\mathbf{Q}^{-1} \big[(\widehat{\mathbf{x}}_{k|N} - \widehat{\mathbf{f}}(\widehat{\mathbf{x}}_{k|N}))(\widehat{\mathbf{x}}_{k|N} - \widehat{\mathbf{f}}(\widehat{\mathbf{x}}_{k|N}))^T + \mathbf{P}_{k|N} + \mathbf{A}\mathbf{P}_{k|N}\mathbf{A}^T \big] \Big] \\
+ &\sum_{k=0}^{N} tr\Big[\mathbf{R}^{-1} \big[(\mathbf{y}_k - \widehat{\mathbf{h}}(\widehat{\mathbf{x}}_{k|N}))(\mathbf{y}_k - \widehat{\mathbf{h}}(\widehat{\mathbf{x}}_{k|N}))^T + \mathbf{C}\mathbf{P}_{k|N}\mathbf{C}^T \big] \Big] \\
+ &tr\Big[\mathbf{P}_0^{-1} [\mathbf{P}_{0|N} + (\widehat{\mathbf{x}}_{0|N} - \overline{\mathbf{x}}_{\mathbf{0}})(\widehat{\mathbf{x}}_{0|N} - \overline{\mathbf{x}}_{\mathbf{0}})^{T}] \Big]
\end{aligned} \tag{H.226}
$$

H.11.5 Learning

In the learning step we estimate the parameters by EM-algorithm.

E-step

At each iteration, we estimate a minimal sufficient statistics.

$$E_{\theta^{(p)}}\big\{ \mathbf{y}_k \mathbf{y}_k^T | Y_N \big\} \tag{H.227}$$
$$E_{\theta^{(p)}}\big\{ \mathbf{y}_k \mathbf{x}_k^T | Y_N \big\} \tag{H.228}$$
$$E_{\theta^{(p)}}\big\{ \mathbf{x}_k \mathbf{x}_k^T | Y_N \big\} \tag{H.229}$$
$$E_{\theta^{(p)}}\big\{ \mathbf{x}_k \mathbf{x}_{k-1}^T | Y_N \big\} \tag{H.230}$$

Kalman filtering - Forwards

for $k = 1, \cdots, N$

$$\boldsymbol{\Sigma}_{k|k-1} = \mathbf{C}\mathbf{P}_{k|k-1}\mathbf{C}^T + \mathbf{R} \tag{H.231}$$
$$\mathbf{K}_k = \mathbf{P}_{k|k-1}\mathbf{C}^T\boldsymbol{\Sigma}_{k|k-1}^{-1} \tag{H.232}$$
$$\mathbf{e}_k = \mathbf{y}_k - \mathbf{C}\widehat{\mathbf{x}}_{k|k-1} \tag{H.233}$$
$$\widehat{\mathbf{x}}_{k|k} = \widehat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k\mathbf{e}_k \tag{H.234}$$
$$\mathbf{P}_{k|k} = \mathbf{P}_{k|k-1} - \mathbf{K}_k\mathbf{C}\mathbf{P}_{k|k-1} \tag{H.235}$$
$$\widehat{\mathbf{x}}_{k+1|k} = \mathbf{A}\widehat{\mathbf{x}}_{k|k} \tag{H.236}$$
$$\mathbf{P}_{k+1|k} = \mathbf{A}\mathbf{P}_{k|k}\mathbf{A}^T + \mathbf{Q} \tag{H.237}$$
$$\mathbf{P}_{k,k-1|k} = \big[\mathbf{I} - \mathbf{K}_k\mathbf{C}\big]\mathbf{A}\mathbf{P}_{k-1|k-1} \tag{H.238}$$

Kalman smoothing - Backwards

for $k = N, \cdots, 1$

$$\widehat{\mathbf{x}}_{k-1|N} = \widehat{\mathbf{x}}_{k-1|k-1} + \mathbf{J}_{k-1}\big[\widehat{\mathbf{x}}_{k|N} - \widehat{\mathbf{x}}_{k|k-1}\big] \tag{H.239}$$
$$\mathbf{P}_{k-1|N} = \mathbf{P}_{k-1|k-1} + \mathbf{J}_{k-1}\big[\mathbf{P}_{k|N} - \mathbf{P}_{k|k-1}\big]\mathbf{J}_{k-1}^T \tag{H.240}$$
$$\mathbf{J}_{k-1} = \mathbf{P}_{k-1|k-1}\mathbf{A}^T\mathbf{P}_{k|k-1}^T \tag{H.241}$$
$$\mathbf{P}_{k,k-1|N} = \mathbf{P}_{k,k-1|k} + \big[\mathbf{P}_{k|N} - \mathbf{P}_{k|k}\big]\mathbf{P}_{k|k}^{-1}\mathbf{P}_{k,k-1|k} \tag{H.242}$$

Sufficient Statistics

$$\mathit{\Gamma}_1 = \frac{1}{N+1}\sum_{k=0}^{N}\big[\mathbf{P}_{k|N} + \widehat{\mathbf{x}}_{k|N}\widehat{\mathbf{x}}_{k|N}^T\big] \tag{H.243}$$
$$\mathit{\Gamma}_2 = \frac{1}{N}\sum_{k=1}^{N}\big[\mathbf{P}_{k|N} + \widehat{\mathbf{x}}_{k|N}\widehat{\mathbf{x}}_{k|N}^T\big] \tag{H.244}$$
$$\mathit{\Gamma}_3 = \frac{1}{N}\sum_{k=1}^{N}\big[\mathbf{P}_{k-1|N} + \widehat{\mathbf{x}}_{k-1|N}\widehat{\mathbf{x}}_{k-1|N}^T\big] \tag{H.245}$$
$$\mathit{\Gamma}_4 = \frac{1}{N}\sum_{k=1}^{N}\big[\mathbf{P}_{k,k-1|N} + \widehat{\mathbf{x}}_{k|N}\widehat{\mathbf{x}}_{k-1|N}^T\big] \tag{H.246}$$
$$\mathit{\Gamma}_5 = \frac{1}{N+1}\sum_{k=0}^{N}\big[\mathbf{y}_k\mathbf{y}_k^T\big] \tag{H.247}$$
$$\mathit{\Gamma}_6 = \frac{1}{N+1}\sum_{k=0}^{N}\big[\mathbf{y}_k\widehat{\mathbf{x}}_{k|N}^T\big] \tag{H.248}$$

H.11.6 M-step

Estimating model parameters from sufficient statistics given by E-step, as coefficients of regression.
We find

$$\widehat{\mathbf{A}} = \Gamma_4 \Gamma_3^{-1} \tag{H.249}$$
$$\widehat{\mathbf{C}} = \Gamma_6 \Gamma_1^{-1} \tag{H.250}$$
$$\widehat{\mathbf{Q}} = \Gamma_2 - \widehat{\mathbf{A}} \Gamma_4^T \tag{H.251}$$
$$\widehat{\mathbf{R}} = \Gamma_5 - \widehat{\mathbf{C}} \Gamma_6^T \tag{H.252}$$
$$\widehat{\overline{\mathbf{x}_\mathbf{0}}} = \widehat{\mathbf{x}}_{0|N} \tag{H.253}$$
$$\widehat{\mathbf{P}}_0 = \mathbf{P}_{0|N} \tag{H.254}$$

H.11.7 Inference

We use the EKF for filtering and prediction to estimate the state values at each time step, as follow:

Filtering

*for $k = 1, 2, \cdots N$, (**Forwards**)*

Prediction

$$\mathbf{f}_k(\mathbf{x}_k) = \sum_{i=1}^{I} \lambda_i^{(f)} \exp\big[-\frac{1}{\sigma_i^{(f)}} (\mathbf{x}_k - \mathbf{c}_i^{(f)})^T (\mathbf{x}_k - \mathbf{c}_i^{(f)}) \big] \tag{H.255}$$
$$\widehat{\mathbf{x}}_{k|k-1} = \mathbf{f}_{k-1}(\widehat{\mathbf{x}}_{k-1|k-1}) \tag{H.256}$$
$$\mathbf{A}_{k-1} = \Big[\frac{\partial \mathbf{f}_k(\mathbf{x}_k)}{\partial \mathbf{x}_k} \Big]_{\widehat{\mathbf{x}}_{k-1|k-1}} \tag{H.257}$$
$$\mathbf{P}_{k|k-1} = \mathbf{A}_{k-1} \mathbf{P}_{k-1|k-1} \mathbf{A}_{k-1}^T + \mathbf{Q} \tag{H.258}$$

Update

$$\mathbf{h}_k(\mathbf{x}_k) = \sum_{i=1}^{I} \lambda_i^{(h)} \exp\big[-\frac{1}{\sigma_i^{(h)}} (\mathbf{x}_k - \mathbf{c}_i^{(h)})^T (\mathbf{x}_k - \mathbf{c}_i^{(h)}) \big] \tag{H.259}$$

$$\mathbf{C}_k = \Big[\frac{\partial \mathbf{h}_k(\mathbf{x}_k)}{\partial \mathbf{x}_k} \Big]_{\widehat{\mathbf{x}}_{k|k-1}} \tag{H.260}$$

$$\boldsymbol{\Sigma}_{k|k-1} = \mathbf{C}_k \mathbf{P}_{k|k-1} \mathbf{C}_k^T + \mathbf{R} \tag{H.261}$$

$$\mathbf{K}_k = \mathbf{P}_{k|k-1} \mathbf{C}_k^T \boldsymbol{\Sigma}_{k|k-1}^{-1} \tag{H.262}$$

$$\mathbf{e}_k = \mathbf{y}_k - \mathbf{h}_k(\widehat{\mathbf{x}}_{k|k-1}) \tag{H.263}$$

$$\widehat{\mathbf{x}}_{k|k} = \widehat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k \mathbf{e}_k \tag{H.264}$$

$$\mathbf{P}_{k|k} = \mathbf{P}_{k|k-1} - \mathbf{K}_k \mathbf{C}_k \mathbf{P}_{k|k-1} \tag{H.265}$$

H.11.8 Estimation of RBF parameters

From states given by Inference step:

$$\{\mathbf{x}_{0|N}, \mathbf{x}_{1|N}, \cdots, \mathbf{x}_{N-1|N}, \mathbf{x}_{N|N}\}$$

and observations:

$$\{\mathbf{y}_1, \mathbf{y}_2, \cdots, \mathbf{y}_N\}$$

by a learning procedure, we estimate the parameters of the RBF functions.

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.

Albrecht F., "Time Varying Exposure in Hedge Funds.", *HEC Lausanne*, Paper October 2005

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.

T.W. Anderson,(1951), "Estimated linear restrictions on regression coefficients for multivariate normal distributions ", *The Annals of Mathematical Statistics* , Vol. 22, pp. 327-351.

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.

Bashi, A.S., V.P. Jilkov, X.R. Li, and H. Chen, "Distributed Implementation of Particle Filters", in ,

Bellman R., "*introduction to the Mathematical Theory of Control Processes*", Academic Press, 1961.

Benoudjit, N. and M. Verleysen, "On the kernel widths in Radial-Basis Function Networks", in 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", in Official Monography, Edited by EUROSTAT.

Billio, M., (2000), "Neural Networks for Pattern Recognition", (*Oxford University Press).*

Bishwal, M., (2007), "Parameter Estimation in Stochastic Differential Equations" (Springer).

Bjorck, A., (2007), "Numerical methods for least squares problems" (SIAM)

Bolland, P.J., and J.T. Connor, ,(1997), "A Constrained Neural Network Kalman Filter for Price Estimation High Frequency Financial data", International Journal of Neural Systems, Vol. 8, N° 4, pp 399-415.

Box, G. and G.M. Jenkins,(1970), " Time series Forecasting and Control ", Holden-Day ,

Breymann, W., F. Zumbach, M. Dacarogna, and U. Muller (2000), "Dynamical deseasonalization in otc and localized exchange-traded markets", Olsen & Associates, Internal document.

Bruckner, U., and I. Nolte, (2002), "Intrinsic Time", Olsen & Associates, Internal document June 30,2002.

N.A. Campbell,(1980), "Shrunken estimators in discriminant and canonical variate analysis ", Applied Statistics , Vol. 29, pp. 5-14.

Carter, C.K., and R. Kohn,(1994), "On Gibbs sampling for state space models", Biometrika, Vol. 81/3, pp. 541-553.

Casella, G., and E. George,(1992), "Explaining the gibbs sampler", Am. Statist., Vol. 46, pp. 249-344.

Chen, Z., and S. Haykin,(2002), "On different facets of regularization theory", Neural Computing, Vol. 14, pp. 2791-2846.

Chien, Y.T., and K.S. Fu,(1992), "On Bayesian Learning and stochastic Approximation", IEEE Trans. Syst. Sci. Cybern., Vol. 3/1, pp. 28-38.

Chopin, N., "A sequentila particle filter method for static models", Biometrika, Vol. 89(3), pp. 539-552, Aug 2002.

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.

Chordia, T., R. Roll, and A. Subrahmanyan,(2000), "Commonality in liquidity", Journal of Financial economics, Vol. 56, pp. 3-28.

Choy, R., and A. Edelman,(2003), "Parallel MATLAB: doing it Right", Massachusetts Institut of Technology, Cambridge, MA 02139.

Chui, C.K., and G. Chen,(1990), "Kalman Filtering with Real-Time Applications", Springer.

Crisan, D., P. Del Moral, and T. Lyons,(1999), "Non linear filtering using branching and interacting particle systems", Markov processes Related Fields, Vol. 5/3, pp. 293-319.

Crisan, D., P. Del Moral, and T. Lyons,(1999), "Interacting particules systems approximation of the Kushner Stratonovitch equation", Adv. Appl. Prob., Vol. 31/3, pp. 819-838.

Crisan, D., (2003), "Exacts rates of convergence for a branching paricle approximation to the solution of Zakai equation", Ann. Prob., Vol. 32.

Cunningham, E.P. and C.R. Henderson, (1968), "An Iterative Procedure for Estimating Fixed Effects and Variance Components in Mixed Model Situations", Biometrics, Vol. 24, N° 1 (Mar., 1968), pp. 13-25.

Cyganowski, S., L. Grüne, and P.E. Kloeden,(2005), "MAPLE for Stochastic Differential Equations", Fachbereich Mathematik, Johan Wolfgang Goethe-Universität.

Dablemont, S., G. Simon, A. Lendasse, A. Ruttiens, F. Blayo, M. Verleysen(2003), "Time series forecasting with SOM and local non-linear models - Application to the DAX30 index prediction", WSOM 2003, Workshop on Self-Organizing Maps, pp. 340-345, 2003.

Dablemont, S., S. Van Bellegem, M. Verleysen,(2007), "Modelling and Forecasting Financial Time Series of "tick data" by Functional Analysis and Neural Networks", submited to European Journal of Finance.

Dablemont, S., S. Van Bellegem, M. Verleysen,(2007), "Forecasting "High" and "Low" of financial Time Series by particle Filters and Kalman Filters", submited to European journal of Finance.

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)

Dacorogna, M.M., U A. Muller, R.J. Nagler, R.B. Olsen and O.V. Pictet,(1993), "A geographical model for the daily and week seasonal volatility in foreign exchange market", Journal of International Money and Finance, Vol. 12, pp. 413-438, 1993.

de Boor, C. (1978), " A Practical Guide to Splines ", New York : Springer,

Deck, T., and T.G. Theting,(2004), "Robust Parameter estimation for Stochastic Differential Equations", Acta Applicandae Mathematicae , Vol. 84 N°3, pp. 279-314.

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

DeGroot, M. "Probability and statistics", Addison Wesley, 1989.

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

Dempster,A.P., N.M. Laird, and D.B. Rubin,(1977), "Maximum likelihood from incomplete data via the EM algorithm ", Jounal of the Royal Statistical Society, Ser. B , Vol. 39, pp. 1-22.

Dempster,A.P., M.R. Selwyn, C.M. Patel, and A.J. Roth,(1984), "Statistical and Computational Aspects of Mixed Model Analysis ", Applied Statistics , Vol. 33, N° 2, pp. 203-214.

Diebold, F., J.H. Lee and G.Weinbach, "Regime switching with time-varying transition probabilities, in non stationary time series analysis and cointegration", ed. by C. Hargreaves, 1994, pp 283-302, Oxford University Press.

DiPillo,P.J. (1976), "The application of bias to discriminant analysis ", Communiucation in Statistics, Part A - Theory and Methods , Vol. A5, pp. 843-854.

Do, B., "estimating the Heston Stochastic Volaty Model from Option Prices by Filtering: A Simulation Exercice", 2005, Monash University Paper October 4,2005.

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

Doucet, A., S. godsill, and C. Andrieu "On Sequential Monte Carlo sampling Methods for Bayesian Filtering", 2000, Statist. Comput., Vol. 10, pp. 197-208, 2000.

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", Seminar on Hogh frequency Finance, Konstanz, June 29,2002

Engel, R.F., "Autoregressive Conditional Heteroscedasticity with Estimates of the Variance of United Kingdom Inflation", Econometrica, Vol. 50, pp. 987-1007, 1982.

Engel, R.F., and J. Russel "Autoregressive Conditional Duration: A New model for Irregularly Spaced Data", Econometrica, Vol. 66, 5 pp. 1127-1162, 1998.

Engel, R.F., and J. Russel "Analysis of High Frequency Financial Data", December 21, 2004. http://home.uchicago.edu/ lhansen/survey.pdf

Epps, T., "Comovements in stocks prices in a very short run", Journal of the American Statistical Association, Vol. 74, pp. 291-298, 1979.

Fama, E.F., "Efficient capital markets: a review of theory and empirical work", Journal of Finance, Vol. 25, pp. 383-417, 1970.

Fama, E.F., "Efficient Capital Markets : II", The Journal of Finance, Vol. XLVI, N° 5, pp. 1515-1617, 1991.

Fearnhead, P. " Sequential Monte Carlo methods in filter theory", PhD thesis, University of Oxford, England, 1998.

Bjorck, A., (2007), "Practical methods of optimization" (John Wiley & Sons)

Fox, D., W. Burgard, H. kruppa, and S. Thrun (2000), "A Probabilistic Approach to Collaborative Multi-Robot Localization", Autonomous Robots vol N° 8/2, pp. 1-25, 2000.

Fox, D., W. Burgard, F. Dellaert, and S. Thrun (2001), "Particle Filters for Mobile Robot Localization", in Sequential Monte Carlo Methods in Practice, Ed; A. Doucet, N. de Freitas, and N. Gordon. Springer 2001.

Franke G., R. Olsen and W. Pohlmeier, "Overview of Forecasting models", PhD thesis, University Paris-Sud, Orsay, France, 1997

Franke, G., R. Olsen and W. Pohlmeier,(June 2002), "Overview of Forecasting Models", University of Konstanz, Seminar on High Frequency Finance, paper SS 2002.

Fruhwirth-Schnatter, S., and S. Kaufmann,(June 2004), "Model-based lustering of Multiple Time Series", journal of Business & Economic Statistics, vol. N° 26, N° 1, Jan 2008, pp. 78-89.

Gardiner, C.W., (1990), "Handbook of Stochastic Methods for Physics, Chemistry and the Natural Sciences", Springer-Verlag.

Geweke, J., (1989), "Bayesian inference in econometric models using Monte Carlo integration", Econometrica vol N° 57, pp.1317-1339, 1989.

Gijbels, I., (1998), "Advanced Nonparametric Statistics", Université catholique de Louvain, 1998.

Gilsing, H., and T. Shardlow,(2005), "SDELab - Stochastic differential equations with MATLAB", Institut für Mathematik, Humboldt Universität zu Berlin.

Grewal, M.S., and A.P. Andrews,(2001), "Kalman Filtering: Theory and Practice. Using MATLAB", John Wiley & Sons .

Guo, W., (2002), "Functional Mixed Effects models", Biometrics, Vol. 58, pp. 121-128.

Hamilton, J.D.,(1994), " Time Series Analysis ", Princeton University Press,

Harrison, J.M., S.R. Pliska "Martingales and Stochastic integrals in the Theory of Continous trading", Stochastic Processes and their Applicartions, Vol. 11, Pp. 215-260, 1981.

Harvey, A.C., (1989), "Forecasting, Structural Time Series Models and the Kalman filter", Cambridge University Press

Harville, D., (1976), "Extension of the Gauss-Markov Theorem to include the Estimation of Random Effects", The Annals of Statistics, Vol. 4, N° 1 (1976), pp. 384-395.

Harville, D., and A.L. carriquiry (1992), "Classical and Bayesian Prediction as Applied to an Unbalanced Mixed Linear Model", Biometrics, Vol. 48,(Dec. 1992), pp. 987-1003.

Harville, D., (1977), "Maximum Likelihood Approaches to Variance Component Estimation and to Related Problems", Journal of the American Statiscal Association, Vol. 72,(Jun. 1977), Vol. 72, N° 358, pp. 320-338.

Hastie, T., R. Tibshirani, and J. Friedman ,(2001), " The Elements of Statistical Learning. dta Mining, Inference, and prediction ", Springer ,

Hastie, T., A. Buja and R. Tibshirani,(1995), "Penalized discriminant anazlysis", Annals of Statistics, Vol. 23, pp. 73-102.

T. Hastie and R. Tibshirani,(1996), "Discriminant analysis by Gaussian mixtures", Jounal of the Royal Statistical Society, Serie B, Methodological, Vol. 58, pp.155-176.

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

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

Haykin, S.,(2002), "Adaptative Filter theory", Prentice Hall,

C.R. Henderson, (1982), "Analysis of Covariance in Mixed Model: Higher-Level, Nonhomogeneous, and Random Regressions", Biometrics, Vol. 38, N° 3, Special Issue: Analysis of Covariance (Sep., 1992), pp. 623-640.

Heston S. "A closed-Form Solution for Options with Stochastic Volatility with Applications to Bonds and Currency options", in Review of Financial studies, Vol. 6, pp. 327-343, 1993.

Hillmer, S., P. Yu "The market speed of adjustment to new information", Journal of inancial Economics, Vol. 7, pp.321-345, 1979.

Horsthemke, W. and R. Lefever, (1984), "Noise-Induced Transitions", Springer-Verlag

Hull, C.J., (2000), "Option, Futurs, and Other Derivatives", International Edition

James, G.M., T.H. Hastie and C.A. Sugar,(2000), "Principal component models for sparse functional data", Biometrika, Vol. 87, pp. 587-602.

James, G.M., C.A. Sugar,(2003), "Clustering for sparcely sampled functional data", Journal of the Americal Association, Vol. 98, pp. 397-408.

Jimenez, J.C., and T. Ozaki,(2005), "An Approximate Innovation method for the Estimation of diffusion processes from Discrete Dta", Journal of the Time Series Analysis, Vol. 27, pp. 76-97.

Javahery, A., D. Lautier, and A. galli (2003), "Filtering in Finance", Wilmott magazine.

Jazwinski A.H.,(1970), "Stochastic Processes and Filtering Theory", Academic Press.

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

Kalman, R.E. and R.S. Bucy (1961) "New results in linear filtering and prediction theory.", Transactions of the ASME, Journal of basic Engineering, Vol. 83, pp. 95-108, 1961

Karlin, S. and H.M. Taylor, "A First Course in Stochastic Processes", Academic Press.

Kim, C.J., "A note on approximation Bayesian bootsrap imputation", in Biometrika, Vol. 89, N°. 2 (2002), pp. 470-477.

Kim, C.J.,and M. Nelson "State-space Models with regime-Switching: Classical and Gibbs Sampling Approach with Applications", MIT Press 1999.

Kloeden, P.E. and E. Platen, "Numerical Solution of Stochastic Differential Equations", Springer-Verlag.

Kloeden, P.E., E. Platen and H. Schurz, "Numerical Solution of SDE Through Computer Experiments", Springer-Verlag.

Kullback, S.,(1959), "Information Theory and Statistics", John Wiley Publications, 1959.

Kushner, H.J.,(1967), "Dynamic equations for optimal nonlinear filtering", Jr. Differential Equations, Vol. 3, pp. 179-190.

Kushner, H.J.,(1977), "Probability Methods for Approximations in Stochastic Control and for Elliptic Equations", Academic Press.

Laird, N.M., J.H. Ware,(1982), "Random-Effects Models for Longitudinal Data", Biometrics, Vol. 38, pp. 963-974.

Lo, A.W., H. Mamaysky, and J. Wang "Foundations of Technical Analysis : computation algorithms, statistical inference, and empirical implementation", Journal of finance, Vol. 55, pp. 1705-1765, 2000

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

Lynch, P.E., and G.O. Zumbach,(2003), "Market heterogeneities and the causal structure of volatility", Working paper, Olsen Group.

Mark, Z.,and Y. Baram "The bias-varinace dilemma of the Monte Carlo method", in Artificial neural Networks, ICANN2001 (2001).

Maslowski, B., and J. Pospisil "Parameter Estimates for linear Differential Equations with Fractional Boundary Noise", in Communication and Information and Systems, Vol 27, N° 1, pp.1-20 (2007).

Muller, U.A., M.M. Dacorogna, R.D. Davé, O.V. Pictet, R.B. Olsen and J.R. Ward,(1995), "Fractals and Intrinsic Time - A Challenge to Econometricians", Working paper, O and A research Group - Paper UMA.1993-08-16 .

Muller, U.A., (2001), "The Olsen Filter for Data in Finance", Working paper, O & A research Group - Paper UMA.1999-04-27.

Murphy, K.P., (2002), "Dynamic Bayesian Networks: representation, Inference and Learning", PhD thesis, University of California, Berkeley, Computer Science, 2002.

Nabney, I., (2001), "Advance in Pattern Recognition", Springer series in Advances in Pattern Recognition.

Nelson, A.T.. "Nonlinear Estimation and modeling of Noisy Time-Series by dual Kalman Filtering Methods", PhD thesis, OGI School of Science and Engineering, Oregon Health and Science University, 2000.

Nielsen, J.N., H. Madsen and H. Melgaard,(2000), "Estimating Parameters in Discretely, Partially Observed Stochastic Differential Equations.", Technical University of Denmark.

Niklaus, B.(2004), "La Divergence de Kullback-Leibler ", Ecole Polytechnique Fédérale de Lausannes,

Norgaard, M., (2000), "Neural Networks for Modelling and Control of Dynamic Systems", Springer-Verlag.

O'Hara, M.(1998), " Market Microstructure Theory ", Blackwelll Publishers,

Oksendal, B.(1998), " Stochastic Differential Equations ", Springer,

Oksendal, B.,(2003), "Stochastic Differential equations: An Introduction with Applications", Springer.

Olsen, R.B., M.M. Dacorogna, U.A. Muller and O.V. Pictet ,(1992), "Going Back to the Basics Rethinking Market Efficiency", Olsen and Associates Research Group, RBO. 1992-09-07 December 2, 1992.

Pattel, J., and M. Wolson ,(1984), "The intraday speed of adjustment of stock prices to earnings and dividend announcements", Journal of Financial economics, Vol. 13, pp. 223-252.

Peters, E.E.,(1991), "Chaos and Order in the Capital markets: A New view of Cycles, Prices, and Market Volatility", Wiley, New York.

Picchini, U.,(2007), "SDE Toolbox - Simulation and estimation of Stochastic Differential Equations with MATLAB", http: sdetoolbox.sourceforge.net.

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

Plummer, T.,(1991), "Forecasting Financial Markets: A Technical analysis and the Dynamic of Prices", Wiley, New York.

Ramsay, J.O., G. Hooker, D. Campbell and J. Cao, "Parameter Estimation for Differential Equations: A Generalized Smoothing Approach", in Journal of the Royal Statistical Society: Series B, vol 69, N° 5, November 2007, pp. 741-796.

Ramsay, J.O., and B.W. Silverman,(1997), " Functional Data Analysis ", Springer Series in Statistics,

Ramsay, J.O. and Silverman,B.W.(2002), "Applied Functional Data Analysis" Springer Series in Statistics,

Refenes, A.N.., A. Zapranis, and G. Francis ,(1994), "Stock performance modeling using neural networks: a comparative study with regression models", Neural Networks, Vol. 5, pp. 961-970.

Rice, J.A. and C.O. Wu,(March 2001), "Nonpoarametric Mixed Effects models for Unequally Sampled Noisy Curves", Biometrics, Vol. 57, pp. 253-259.

Rimmer, D., A. Doucet, and W.J. Fitzgerald, "Particle filters for stochastic diffeential equations of nonlinear diffusions", Technical Report, Cambrdge University Engineering Dept..

Risken, H., and B.W. Silverman,(1989), "The Fokker-Planck Equation", Springer-Verlag,

Ruttiens, A.,(2003), "Futures, Swaps, Options, Les produits financiers dérivés", édipero,

Sarkka, S. "Recursive Bayesian Inference on Stochastic Differential Equations", PhD thesis, Helsinki University of Technology, Laboratory of Computational Engineering , 2006.

Sevelo, N.C.,and M. Zelen "Normal Approximation to the Chi-square and Non-Central F Probability Functions", in biometrika, Vol. 47, N°. 3/4 (Dec;, 1960), pp. 411-416.

Sparreman, G. "Simulation and Parameter Estimation of Stochastic Volatility Models", PhD thesis, Stockholm University, 2006.

Stein, E.,and J. Stein "Stock Price Distributions with Stochastic Volatility: An Analitical Approach", in Review of Financial studies, Vol. 4, pp. 727-752, 1991.

Simandl, M., J. Kralovec, and P. Tichavsky, "Filtering, prediction, and smoothing Cramer-Rao bounds for discrete-time nonlinear dynamic systems", in Automatica, Vol. 37, (2001), pp. 1703-1716.

Taylan, P., and G.W. Weber, "Parameter Estimation for Stochastic Differential Equations by Additive Models Using Nonlinear Regression, Splines and Conic Programming", Middle East Technical University, Institut of Applied mathematics, Ankara.

van der Merwe, R., , J.F.G. de Freitas, A. Doucet and E.A. 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.

Vapnik, V., (1998), "Statistical Learning Theory", Wiley, (1998).

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

Wand, M.P., and M.C. Jones, "Kernel Smoothing", Chapman & Hall, 1995

White, H., (1989), "Learning in artificial neural networks: a statistical perspective", Neural Computing, Vol. 1, pp. 425-464.

Xu, J., (2005), "Pricing and Hedging Options under Stochastic Volatilty", The University of British Columbia, March 2005

Zumbach, G., F. Corsi, and A. Trapletti (2002), "Efficient estimation of volatility using High Frequency Data", Olsen & Associates, Internal document.

Index