MASTER'S DEGREE THESIS

Master of Science in Advanced Mathematics and Mathematical Engineering

Title: Sequential estimation of neural models by Bayesian filtering

Author: Pau Closas

Advisor: Antoni Guillamon

Department: Department de Matemàtica Aplicada I

Academic year: 2013-2014

UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH
Facultat de Matemàtiques i Estadística

UNIVERSITAT POLITÈCNICA
DE CATALUNYA

MSc Thesis

Sequential estimation of neural models by Bayesian filtering

Author: Dr. Pau Closas

Thesis advisor: Prof. Antoni Guillamon

Department de Matemàtica Aplicada I
Universitat Politècnica de Catalunya

Barcelona, June 2014

A la Míriam i al Magí,

Abstract

One of the most challenging problems in neuroscience is the to unveil brain's connectivity. This problem might be treated from several perspectives, here we focus on the local phenomena occurring at a single neuron. The ultimate goal is thus to understand the dynamics of neurons and how the interconnection with other neurons affects its state.

Measurements of membrane potential traces constitute the main observables to derive a biophysical neuron model. In particular, the dynamics of auxiliary variables and the model parameters are inferred from voltage traces, in a costly process that typically entails a variety of channel blocks and clamping techniques, as well as some uncertainty in the parameter values due to noise in the signal. Moreover, voltage traces are also useful to obtain valuable information about synaptic input, an inverse problem with no satisfactory solution yet.

In this Thesis, we are interested in methods that can provide on-line estimation and avoid the need of repetitions that could be contaminated by neuronal variability. Particularly, we concentrate on methods to extract intrinsic activity of ionic channels, namely the probabilities of opening and closing ionic channels, and the contribution of synaptic conductances. We design a method based on Bayesian theory to sequentially infer these quantities from single-trace, noisy membrane potentials. The proposed estimation method highly relies on the fact that the neuron model is known. This is true to some extent, but most of the parameters in the model are to be estimated beforehand (this holds for any model). Therefore, the method is enhanced to the case of unknown model parameters, thus augmenting the algorithm with a method to jointly estimate the parameters using the same single-trace voltage measure.

We validate the proposed inference methods in realistic computer simulation experiments. The error performance is compared to the theoretical lower bound of accuracy that has been derived in the framework of this Thesis.

Resum

Un dels reptes més difícils de la neurociència és el d'entendre la connectivitat del cervell. Aquest problema es pot tractar des de diverses perspectives, aquí ens centrem en els fenòmens locals que ocorren en una sola neurona. L'objectiu final és, doncs, entendre la dinàmica de les neurones i com la interconnexió amb altres neurones afecta al seu estat.

Les observacions de traces del potencial de membrana constitueixen la principal font d'informació per a derivar models matemàtics d'una neurona, amb cert sentit biofísic. En particular, la dinàmica de les variables auxiliars i els paràmetres del model són estimats a partir d'aquestes traces de voltatge. El procés és en general costós i típicament implica una gran varietat de blocatges químics de canals iònics, així com una certa incertesa en els valors dels paràmetres a causa del soroll de mesura. D'altra banda, les traces de potencial de membrana també són útils per obtenir informació valuosa sobre l'entrada sinàptica, un problema invers sense solució satisfactòria a hores d'ara.

En aquesta Tesi, estem interessats en mètodes d'estimació seqüencial, que permetin evitar la necessitat de repeticions que podrien ser contaminades per la variabilitat neuronal. En particular, ens concentrem en mètodes per extreure l'activitat intrínseca dels canals iònics, és a dir, les probabilitats d'obertura i tancament de canals iònics, i la contribució de les conductàncies sinàptiques. Hem dissenyat un mètode basat en la teoria Bayesiana de filtrat per inferir seqüencialment aquestes quantitats a partir d'una única traa de voltatge, potencialment sorollosa. El mètode d'estimació proposat està basat en la suposició d'un model de neurona conegut. Això és cert fins a cert punt, però la majoria dels paràmetres en el model han de ser estimats per endavant (això és valid per a qualsevol model). Per tant, el mètode s'ha millorat pel cas de models amb paràmetres desconeguts, incloent-hi un procediment per estimar conjuntament els paràmetres i les variables dinàmiques.

Hem validat els mètodes d'inferència proposats mitjançant simulacions realistes. Les prestacions en termes d'error d'estimació s'han comparat amb el límit teòric, que s'ha derivat també en el marc d'aquesta Tesi.

Acknowledgements

"*Be yourself, everyone else is already taken.*"
Oscar Wilde (1854 - 1900)

Ja hi tornem a ser. Sembla que va ser ahir que estava escrivint uns agraïments i ja han passat cinc anys. De debò, com pot passar el temps tan ràpid i tan lent alhora? Suposo que ajuda el fet que hagi estat una etapa vital ben intensa, plena de grans notícies i passos endavant.

Al 2011 em vaig matricular al Master in Advanced Mathematics and Mathematical Engineering, tot complint una idea que ja feia molt de temps que em rondava. Després de tres anys he aconseguit completar aquests estudis, que conclouen amb el depòsit d'aquest projecte que tens a les mans. Ha estat una experiència enriquidora i que ha assolit les expectatives que tenia en matricular-me. De totes maneres és just dir que el camí ha estat llarg, m'han faltat hores al dia i dies al mes.

En primer lloc, vull agrair al meu director de projecte, l'Antoni Guillamon, el temps dedicat i la possibilitat de treballar amb ell. He après moltíssim al seu costat. A nivell tècnic és algú privilegiat que té una visió dels problemes envejable. En el camp personal, crec honestament que no podria haver triat un director amb qui congeniar millor. Gràcies per totes les reunions (gairebé furtives, doncs els dos tenim seriosos problemes de temps) i discussions engrescadores que han generat mil idees. Sort que finalment les hem reconduït en un projecte que es podia acabar en un temps finit. Toni, això és un punt i seguit.

No vull passar per alt la gran qualitat de l'equip docent de la Facultat de Matemàtiques i Estadística. He tingut la sort de creuar-me amb molt bons professionals. Deixeu-me agrair-vos la dedicació i paciència davant la tabarra que us he donat amb els dubtes més esotèrics. En ordre d'aparició: Josep Fàbrega (gràcies per fer-me redescobrir la probabilitat, un altre cop); Quim Puig i Jesús Fernández; Oriol Serra i Guillem Perarnau; Lupe Gómez i Àlex Sànchez; Josep Ginebra i Xavi Puig (Visca el Bayesianisme!); Jorge Villar i Javier Herranz; Pilar Muñoz i Josep A. Sánchez.

Míriam, ho hem tornat a fer. Gràcies per l'enorme comprensió. Ets el meu pal de paller i les paraules queden buides per agrair-te tantes coses. Ets la millor de les companyies, la meva musa. Fa poc més de dos anys va néixer el Magí, el flamant nou membre de la família. Ha fet que aquesta etapa fos encara més èpica. Amb el Magí he trobat les forces per, un cop més, donar-ho tot.

Pau Closas
Barcelona, Juny de 2014

Contents

List of Figures

Notation

Boldface upper-case letters denote matrices and boldface lower-case letters denote column vectors.

$\mathbb{R}$, $\mathbb{C}$	The set of real and complex numbers, respectively.
$\mathbb{R}^{N\times M}$, $\mathbb{C}^{N\times M}$	The set of $N \times M$ matrices with real- and complex-valued entries, respectively.
$\hat{x}$	Estimation and true value of parameter x.
$f(x)\|_{x=a}$	Function $f(x)$ evaluated at $x = a$.
$\|x\|$	Absolute value (modulus) of scalar x.
$\|\mathbf{x}\|$	ℓ^2-norm of vector $\mathbf{x}$, defined as $\|\mathbf{x}\| = \left(\mathbf{x}^H\mathbf{x}\right)^{\frac{1}{2}}$.
$\dim\{\mathbf{x}\}$	Dimension of vector $\mathbf{x}$.
$[\mathbf{x}]_r$	The r-th vector element.
$[\mathbf{X}]_{r,c}$	The matrix element located in row r and column c.
$[\mathbf{X}]_{r,:}$	The r-th row of matrix $\mathbf{X}$.
$[\mathbf{X}]_{:,c}$	The c-th column of matrix $\mathbf{X}$.
$\mathrm{Tr}\{\mathbf{X}\}$	Trace of matrix $\mathbf{X}$. $\mathrm{Tr}\{\mathbf{X}\} = \sum_{n=1}^{N}[\mathbf{X}]_{nn}$.
$\det(\mathbf{X})$	Determinant of matrix $\mathbf{X}$.
$\mathrm{diag}(\mathbf{x})$	A diagonal matrix whose diagonal entries are given by $\mathbf{x}$.
$\|\mathbf{X}\|_F$	Frobenius norm of matrix $\mathbf{X}$. If $\mathbf{X}$ is $N \times N$, $\|\mathbf{X}\|_F = \left(\sum_{u=1}^{N}\sum_{v=1}^{N}\|x_{uv}\|^2\right)^{\frac{1}{2}} = \left(\mathrm{Tr}\left\{\mathbf{X}^H\mathbf{X}\right\}\right)^{\frac{1}{2}}$
$\mathrm{Chol}\,(\mathbf{A})$	Cholesky factorization of an Hermitian positive-definite matrix $\mathbf{A}$ such that $\mathbf{A} = \mathbf{S}\mathbf{S}^\top$ if $\mathbf{S} = \mathrm{Chol}\,(\mathbf{A})$.

$\mathbf{I}$ Identity matrix. A subscript can be used to indicate the dimension.

$\mathbf{X}^*$ Complex conjugate of matrix $\mathbf{X}$ (also applied to scalars).

$\mathbf{X}^T$ Transpose of matrix $\mathbf{X}$.

$\mathbf{X}^H$ Complex conjugate and transpose (Hermitian) of matrix $\mathbf{X}$.

$\mathbf{X}^\dagger$ Moore-Penrose pseudoinverse of matrix $\mathbf{X}$. If $\mathbf{X}$ is $M \times N$,
$\mathbf{X}^\dagger = \mathbf{X}^H \left(\mathbf{X}\mathbf{X}^H\right)^{-1}$ if $M \leq N$,
$\mathbf{X}^\dagger = \mathbf{X}^{-1}$ if $M = N$, and
$\mathbf{X}^\dagger = \left(\mathbf{X}^H\mathbf{X}\right)^{-1} \mathbf{X}^H$ if $M \geq N$.

$\odot$ Schur-Hadamard (elementwise) product of matrices.
If $\mathbf{A}$ and $\mathbf{B}$ are two $N \times M$ matrices:

$$\mathbf{A} \odot \mathbf{B} = \begin{pmatrix} a_{11}b_{11} & a_{12}b_{12} & \cdots & a_{1M}b_{1M} \\ a_{21}b_{21} & a_{22}b_{22} & \ddots & a_{2M}b_{2M} \\ \vdots & \vdots & \cdots & \vdots \\ a_{N1}b_{N1} & a_{N2}b_{N2} & \cdots & a_{NM}b_{NM} \end{pmatrix}$$

$\otimes$ The Kronecker or tensor product. If $\mathbf{A}$ is $m \times n$, then

$$\mathbf{A} \otimes \mathbf{B} = \begin{pmatrix} [\mathbf{A}]_{11}\mathbf{B} & \cdots & [\mathbf{A}]_{1m}\mathbf{B} \\ \vdots & & \vdots \\ [\mathbf{A}]_{n1}\mathbf{B} & \cdots & [\mathbf{A}]_{nm}\mathbf{B} \end{pmatrix}$$

$\mathbf{P}_\mathbf{X}$ Orthogonal projector onto the subspace spanned by the columns of $\mathbf{X}$.
$\mathbf{P}_\mathbf{X} = \mathbf{X}\left(\mathbf{X}^H\mathbf{X}\right)^{-1}\mathbf{X}^H$.

$\mathbf{P}_\mathbf{X}^\perp$ $\mathbf{I} - \mathbf{P}_\mathbf{X}$, orthogonal projector onto the orthogonal complement to the columns of $\mathbf{X}$.

$\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ Multivariate Gaussian distribution with mean $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$.

$\mathcal{U}(a, b)$ Uniform distribution in the interval $[a, b]$.

$\mathbb{E}\{\cdot\}$ Statistical expectation. When used with a subindex, it specifies the distribution over which the expectation is taken, e.g.,
$\mathbb{E}_x\{\cdot\}$ over the distribution of a random variable x;
$\mathbb{E}_{x,y}\{\cdot\}$ over the joint distribution of x and y, $p(x, y)$;
$\mathbb{E}_{x|y}\{\cdot\}$ over the distribution of x conditioned to y, $p(x|y)$.

$\ln(\cdot)$ Natural logarithm (base e).

$\delta(n - m)$ Kronecker's delta function, defined as:

$$\delta(n-m) = \delta_{n,m} \triangleq \begin{cases} 1, & \text{if } n = m \\ 0, & \text{if } n \neq m \end{cases}$$

$\Re\{\cdot\}$, $\Im\{\cdot\}$	Real and imaginary parts, respectively.
$o_p(f_N)$	A sequence of random variables X_N is $X_N = o_p(f_N)$, for $f_N > 0 \ \ \forall N$, when $\frac{X_N}{f_N}$ converges to zero in probability, i.e., $\lim_{N\to\infty} P\left\{\left\lvert\frac{X_N}{f_N}\right\rvert > \delta\right\} = 0 \ \ \forall \delta > 0$
$f(t) * g(t)$	Convolution between $f(t)$ and $g(t)$.
$\arg\max_x f(x)$	Value of x that maximizes $f(x)$.
$\arg\min_x f(x)$	Value of x that minimizes $f(x)$.
$\frac{\partial f(\mathbf{x})}{\partial x_i}$	Partial derivative of function $f(\mathbf{x})$ with respect to the variable x_i.
$\frac{\partial f(\mathbf{x})}{\partial \mathbf{x}}$	Gradient of function $f(\mathbf{x})$ with respect to vector $\mathbf{x}$.
$\frac{\partial^2 f(\mathbf{x})}{\partial \mathbf{x}^2}$	Hessian matrix of function $f(\mathbf{x})$ with respect to vector $\mathbf{x}$.
$\nabla_{\mathbf{x}} f(\mathbf{x})$	Gradient of function $f(\mathbf{x})$ with respect to vector $\mathbf{x}$.
$\mathcal{H}_{\mathbf{x}} f(\mathbf{x})$	Hessian matrix of function $f(\mathbf{x})$ with respect to vector $\mathbf{x}$.
$\triangle_{\mathbf{x}_1}^{\mathbf{x}_2} f(\mathbf{x})$	second-order partial derivatives operator of function $f(\mathbf{x})$ with respect to vectors $\mathbf{x}_1$ and $\mathbf{x}_2$. Notice that $\mathcal{H}_{\mathbf{x}} f(\mathbf{x}) \triangleq \triangle_{\mathbf{x}}^{\mathbf{x}} f(\mathbf{x})$ and $\triangle_{\mathbf{x}_1}^{\mathbf{x}_2} = \nabla_{\mathbf{x}_1}\left[\nabla_{\mathbf{x}_2}^T\right]$.
$\dot{f}(t), \ddot{f}(t)$	*d*erivatives *o*f *t*ime of function $f(t)$, equivalent to $\frac{\partial f(\mathbf{t})}{\partial t}$ and $\frac{\partial^2 f(\mathbf{t})}{\partial t^2}$ respectively.
a.s.	almost surely convergence.
i.i.d.	independent identically distributed.
q.e.d.	quod erat demonstrandum.
r.v.	random variable.
*w.p.*1.	convergence with probability one.

Acronyms

ADC	Analog-to-Digital Converter.
AMPA	α-amino-3-hydroxy-5-methyl-4-isoxazolepropionic.
AR	autoregressive.
AWGN	Additive White Gaussian Noise.
BIM	Bayesian Information Matrix.
BLUE	Best Linear Unbiased Estimator.
CKF	cubature Kalman filter.
CNS	Central Nervous System.
CRB	Cramér Rao Bound.
DSS	discrete state-space.
EKF	extended Kalman filter.
EM	Expectation Maximization algorithm.
FIM	Fisher Information Matrix.
GABA	γ-aminobutyric acid.
HMM	Hidden Markov Model.
IS	Importance Sampling.
QKF	quadrature Kalman filter.
KF	Kalman filter.
LS	Least Squares.
ODE	ordinary differential equation.
OU	Ornstein-Uhlenbeck.
MAP	Maximum a posteriori.
MCMC	Markov-Chain Monte-Carlo.
ML	Maximum Likelihood.
MLE	Maximum Likelihood Estimator.

MMSE	Minimum Mean Square Error.
MSE	Mean Square Error.
PCRB	Posterior Cramér-Rao Bound.
PDF	Probability Density Function.
PF	particle filter.
PMCMC	particle Markov-Chain Monte-Carlo.
RAM	Robust Adaptive Metropolis.
RMSE	root mean square error.
SIR	Sampling Importance Resampling.
SIS	Sequential Importance Sampling.
SMC	Sequential Monte-Carlo.
SNR	Signal to Noise Ratio.
SPKF	sigma–point Kalman filter.
SRUKF	square–root unscented Kalman filter.
SRCKF	square–root cubature Kalman filter.
SRQKF	square–root quadrature Kalman filter.
SRKF	square–root Kalman filter.
SRSPKF	square–root sigma–point Kalman filter.
SS	state-space.
UKF	unscented Kalman filter.
UT	unscented transform.

1 Introduction

THIS dissertation deals with the problem of inferring the signals and parameters that cause neural activity to occur. The focus is on a microscopic vision of the problem, where single-neuron models (potentially connected to a network of peers) are in the core of the Thesis. The sole observation available are noisy, sampled voltage traces obtained from intracellular recordings. We design algorithms and inference methods using the tools provided by Bayesian filtering, that allow a probabilistic interpretation and treatment of the problem.

In this chapter, we glance at the structure of the document, serving as a guide to the reader. For the sake of clarity, the mathematical notation and the acronyms used along the dissertation can be consulted at the beginning of the document.

1.1 Motivation and Objectives of the Thesis

One of the most challenging problems in neuroscience is to unveil brain's connectivity. This problem might be treated from several perspectives, here we focus on the local phenomena occurring at a single neuron. The ultimate goal is thus to understand the dynamics of neurons and how the interconnection with other neurons affects its state.

Measurements of membrane potential traces constitute the main observables to derive a biophysical neuron model. In particular, the dynamics of auxiliary variables and the model parameters are inferred from voltage traces, in a costly process that typically entails a variety of channel blocks and clamping techniques [Bre12], as well as some uncertainty in the parameter values due to noise in the signal. Moreover, voltage traces are also useful to obtain valuable information about synaptic input, an inverse problem with no satisfactory solution yet (see for instance [Piw04, Béd11]).

In this Thesis, we are interested in methods that can provide on-line estimation and avoid the need of repetitions that could be contaminated by neuronal variability. Particularly, we concentrate on methods to extract intrinsic activity of ionic channels, namely the probabilities of opening and closing ionic channels, and the contribution of synaptic conductances. We built a method based on Bayesian theory to sequentially infer these quantities from single-trace, noisy membrane potentials. The proposed estimation method highly relies on the fact that the neuron model is known. This is true to some extent, but most of the parameters in the model are to be estimated beforehand (this holds for any model). Therefore, the method is enhanced to the case of unknown model parameters, thus augmenting the algorithm with a method to jointly estimate the parameters using the same single-trace voltage measure.

In conclusion, we propose a method that is able to sequentially infer the time-course of the membrane potential and its intrinsic/extrinsic activity from noisy observations of a voltage trace. The main features of the envisaged algorithm are:

Single-trial: the method should be able to estimate the desired signals and parameters from a single voltage trace, thus avoiding the experimental variability among trials.

Sequential: the algorithm should provide estimates each time a new observation is recorded, thus avoiding re-processing of all data stream each time.

Spike regime: contrary to most solutions operating under the sub-threshold assumption, the method should be able to operate in the presence of spikes as well. This has a twofold reason: the estimation can account for ionic channel activity; and we avoid the well-known sub-threshold misestimation problem in which removing spikes might produce errors in the dynamics estimation.

Robust: the method is model-dependent, thus implying knowledge of the model parameters. This might be a strong assumption and thus the algorithm should be provided with enhancements to adaptively estimate these parameters.

Efficient: the performance of the method should be close to the theoretical lower bounds, meaning that the estimation error is close to the bounds.

Notice that the focus in this Thesis is not on computational reduction techniques, as we thought that other requirements (like performance) were prioritized in this application. The results show the validity of the approach and its statistical efficiency. Although we used the Morris-Lecar neuron model in the computer simulations, the proposed procedure can be applied to any neuron model without loss of generality.

1.2 Thesis Outline and Reading Directions

The dissertation consists of 5 Chapters, where review material and novel contributions are presented. The thesis might be of interest to two groups of people: those working in the neuroscience field and to signal-processing oriented researchers with the objective of learning new fancy applications. The document is organized according to this premise, providing the basics of each topic. For the sake of clarity, the main ideas of the chapters are summarized herein:

Chapter 1 This chapter summarizes the main problem addressed and our research objectives for the rest of the Thesis. We provide as well the reader with reading directions.

Chapter 2 Basic material on neuroscience is provided for the non-specialist to get the biophysical meaning of the problem addressed. From the vast amount of information on the numerous disciplines related to neuroscience, we focus on a microscopic vision where single-neuron models are the core concept. In the chapter we sketch the basic modeling procedures to mimic neuron dynamics.

Note: if you have expertise in neuroscience you can skip this chapter, and please forgive me for the rather vague introduction.

Chapter 3 Similarly to the neuroscience material in Chapter 2, we provide a textbook review of Bayesian filtering methodology. The reason being its paramount importance in the derivation of the type of algorithms we are interested in this Thesis. At a glance, we present the theoretical Bayesian solution to recursive filtering and detail the most popular algorithms and go beyond what is strictly necessary for the comprehension of the methods derived in Chapter 4. We also comment on the derivation of theoretical estimation bounds under this framework, which is not always tackled in the literature where neuroscience and Bayesian filtering collide.

Note: if you know what a Kalman filter or a particle filter is you might be tempted to skip this chapter, please do it.

Chapter 4 This constitutes the core chapter in the Thesis. The material therein includes discussion of the discrete state-space representation of the problem and the model

inaccuracies due to missmodeling effects. In this chapter we present 2 sequential inference algorithms: *i*) a method based on particle filtering to estimate the time-evolving states of a neuron under the assumption of perfect model knowledge; and *ii*) an enhanced version where model parameters are jointly estimated, and thus the rather strong assumption of perfect model knowledge is relaxed. We provide exhaustive computer simulation results to validate the algorithms and observe that they are consistent, in the sense of attaining the theoretical lower bounds derived in the chapter as well.

Chapter 5 This concluding chapter summarizes the main results obtained in this Thesis and points out some interesting open problems which constitute the future work after the Thesis defence.

The work presented in this Thesis has been partially published in the form of scientific publications and talks. We list them here.

[1] C. Vich, P. Closas, and A. Guillamon, "Data treatment in estimating synaptic conductances: wrong procedures and new proposals," Barcelona Computational and Systems Neuroscience (BARCSYN), June 16 and 17, 2014.

[2] P. Closas, A. Guillamon, "Estimation of neural voltage traces and associated variables in uncertain models," BMC Neuroscience, Vol. 14, No. 1, pp. 1151, July 2013.

[3] P. Closas, A. Guillamon, "Estimation of neural voltage traces and associated variables in uncertain models," in Proceedings of the 22nd Annual Computational Neuroscience Meeting (CNS 2013), 13-18 July 2013, Paris (France).

[4] P. Closas, A. Guillamon, "Sequential estimation of gating variables from voltage traces in single-neuron models by particle filtering," in Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2013) 26-31 May 2013, Vancouver (Canada).

[5] P. Closas, and A. Guillamon, "Inference of voltage traces and gating variables in uncertain models," Barcelona Computational and Systems Neuroscience (BARCSYN), June 13 and 14, 2013.

2

Fundamentals of Neuroscience

NEUROSCIENCE is the science that delves into the understanding of the nervous system. It is one of the most interdisciplinary sciences, gathering together experts from a vast variety of fields of knowledge including biology, chemistry, medicine, psychology, physics, mathematics, statistics, engineering, and computer science.

Neuroscience is a rather broad discipline and encompasses many aspects related to the Central Nervous System (CNS). The different topics in neuroscience can be studied from various perspectives depending on the prism used to focus the problem. This ranges from understanding the internal mechanisms that cause a single cell (a neuron) to spike, to explaining the dynamics occurring in populations of neurons that are interconnected. Going further, more macroscopic analysis are important like those treating pools of neuron as an anatomically meaningful function. From microscopic to macroscopic, the CNS research could be classified into molecular neuroscience, cellular neuroscience, neural circuits, systems neuroscience, and cognitive neuroscience.

This Thesis deals with single neuron models, with the rest of the Chapter being devoted to explaining the basic ideas behind a neuron physiology and related mathematical models. Recommended textbooks are [Day05, Izh06, Kee09] and more recently [Erm10], from which we extracted part of the material presented in this chapter.

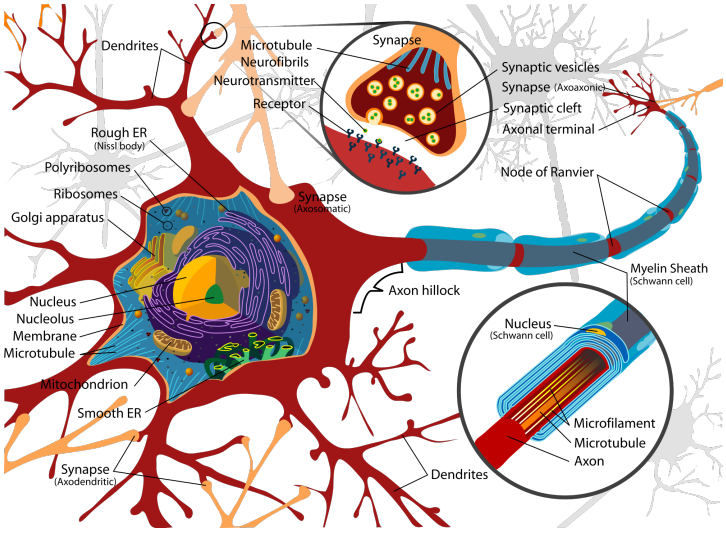


Figure 2.1: Diagram of a neuron (Source: http://en.wikipedia.org/wiki/Neuron).

2.1 Electrophysiology of neurons

Neurons are the basic information processing structures in the CNS. The main function of a neuron is to receive input information from other neurons, to process that information, and to send output information to other neurons. Synapses are connections between neurons, through which they communicate this information. It is controversial how this information is encoded, but it is quite accepted that information produces changes in the electrical activity of the neurons, seen as voltage changes in the membrane potential (i.e., the difference in electrical potential between the interior and the exterior of a biological cell).

To put some numbers, the human brain has *only* around 10^{11} neurons and 10^{15} connections among them (a.k.a. synapses). The basic constituents of a neuron can be seen in Fig. 2.1, where we can identify:

Soma: contains the nucleus of the cell, it is the body of the neuron where most of the information processing is carried.

Dendrites: are extensions of the soma which connect the neuron to neighboring neurons. Dendrites are capturing the stimuli from the rest of neurons.

Axon: is the largest part of a neuron where the information is transmitted in form of an electrical current. A cell might have only one axon or more. The physiological meaning for the propagation of the voltage through the axon can be understood

in terms of voltage-gated ionic channels located in the axon membrane. This is paramount in the topic treated in this Thesis, and thus we provide some further details in this section.

Synapses: located at the axon terminal, are in charge of the electrochemical reactions that cause neuron communications to happen. More precisely, the membrane potential (electrical phenomena) traveling through the axon, when reaching the synapse, activates the emission of neurotransmitters (chemical phenomena) from the neuron to the receptors of the target neurons. This chemical reaction is transformed again into electrical impulses in the dendrites of the receiving neurons.

2.2 Ionic currents

We are specially interested in understanding the phenomena through which an electrical voltage travels the axon from the soma to the synapse. We concentrate here on the biophysical meaning and its mathematical modeling. The basic idea is that the membrane covering the axon is essentially impermeable to most charged molecules. This makes the axon to act as a capacitor (in terms of electrical circuits) that separates the inner and outer parts of the neuron's axon. This is combined with the so-called ionic-channels, that allow the exchange of intracellular/extracellular ions through electrochemical gradients. This exchange of ions is responsible for the generation of an electrical pulse called action potential, that travels along the neuron's axon. Ionic-channels are found throughout the axon and are typically voltage-dependent, which is primarily how the action potential propagates.

The most common ionic species involves in the generation of the action potential are sodium (Na^+), potassium (K^+), chloride (Cl^-), and calcium (Ca^{2+}). For each ionic species, the corresponding ionic-channel aims at balancing the concentration and electrical potential gradients, which are opposite forces regulating the exchange of ions through the gate. The point at which both forces counterbalance is known as *Nernst equilibrium potential* and given by

$$E_{\text{ion}} \approx 62 \ \log_{10} \frac{[\text{Ion}]_{\text{out}}}{[\text{Ion}]_{\text{in}}} \tag{2.1}$$

in mV, where $[\text{Ion}]_{\text{out}}$ and $[\text{Ion}]_{\text{in}}$ are the concentrations of the ion inside and outside the cell, respectively. Nernst equilibrium potential is value for which the cross-membrane potential is zero for a given ionic channel. In the sequel, we denote by E_{Na}, E_{K}, E_{Cl}, and E_{Ca} the Nernst equilibrium potential of the typical ionic species. The time-varying net current of the i-th ionic species, $i \in \mathcal{I} = \{\text{Na}, \text{K}, \text{Cl}, \text{Ca}, \dots\}$, is thus

$$I_i = g_{\text{I}}(v - E_i) \tag{2.2}$$

where $g_{\mathrm{I}} \triangleq g_{\mathrm{I}}(t)$ is the time-varying conductance of the ionic channel in $\mathrm{mS/cm^2}$, and $(v - E_i)$ is the driving force that zeroes when the voltage is equal to the equilibrium potential as argued earlier. The time dependent conductances are responsible for spike (or action potential) generation.

2.3 Conductance-based models

The voltage travels through the axon over time, which results in the so-called compartmental neuron models. For the sake of simplicity and without loss of generality, we consider the evolution of the membrane potential at a specific site of the axon. Therefore, $v \triangleq v(t)$ denotes the continuous-time membrane potential at a point in the axon. Accounting that the membrane potential is seen as a capacitor, the current-voltage relation allows us to express the total current flowing in the membrane as proportional to the time derivative of the voltage. Then, we have that the mathematical model for the evolution of membrane potentials is of the basic form

$$C_m \dot{v} = -\sum_{i \in \mathcal{I}} I_i - \bar{g}_L (v - E_L) - I_{\mathrm{syn}} + I_{\mathrm{app}} \tag{2.3}$$

where C_m is the membrane capacitance and I_{app} represents the externally applied currents, for instance injected via an electrode and used to perform a controlled experiment.

In (2.3) we have introduced two additional terms. The first one is referred to as the *leakage term*. The leakage is mathematically used to gather all ionic channels that are not explicitly modeled. The maximal conductance of the leakage, $\bar{g}_L$, is considered constant and it is adjusted to match the membrane potential at resting state. Similarly, E_L has to be estimated at rest. The second new term in (2.3) gathers the contribution of neighboring neurons and it is referred to as the synaptic current, I_{syn}. We will see later how to model the synaptic current.

Strictly speaking, (2.3) is a dynamical system, although we have to provide more details for the different components before using it as a reliable model for neural activity. There are several models [Izh04, Izh06] that explain the membrane potential evolution using dynamical systems theory. Roughly speaking, there are terms which model intrinsic activity of the neuron (mostly, chemical reactions in the cell) and terms related to the contribution of synaptic noise to the neuron's activity (that is the voltages received from neighboring neurons) [Hod52, Piw04, Béd11, Kob11, Bre12]. Section 2.4 comments on the former mechanisms, mainly discussing the ionic conductances.

Ionic channels are the responsible for electrochemical gradient stabilization of ionic species. The conductance of the i-th channel, g_i, can be seen as a switch that opens or

closes the pump. Recall that $i \in \mathcal{I} = \{\mathrm{Na}, \mathrm{K}, \mathrm{Cl}, \mathrm{Ca}, \dots\}$. A refinement of (2.2) is then,

$$I_i = \bar{g}_i \; p_i(v - E_i) \tag{2.4}$$

where $\bar{g}_i$ is a constant called the maximal conductance, which is fixed for each ionic species. The variable $p_i \triangleq p_i(v)$ is the average proportion of channels of the type i in the open state. Notice that the proportion is in general voltage-dependent (i.e., sensitive to the membrane potential) and thus they are said to be voltage-gated.

The proportion p_i can be further classified into gates that activate (i.e., gates that open the i-th ionic channel) and those that inactivate (i.e., gates that close the i-th ionic channel). Mathematically, omitting the dependence on i,

$$p = m^a h^b \tag{2.5}$$

where a is the number of activation gates and $0 < m \triangleq m_i(v) < 1$ the probability of activating gate being in the open state. Similarly, b is the number of inactivation gates and $0 < h \triangleq h_i(v) < 1$ the probability of inactivating gate being in the open state.

We refer to m and h as the gating variables of the ionic channel. The dynamics of these gating variables are responsible for the membrane potential generation and can be expressed by a first-order differential equation:

$$\dot{m} = \frac{m_\infty(v) - m}{\tau_m(v)} \tag{2.6}$$

$$\dot{h} = \frac{h_\infty(v) - h}{\tau_h(v)} \tag{2.7}$$

where the activation/inactivation functions ($m_\infty(v)$ and $h_\infty(v)$, respectively) and time constants ($\tau_m(v)$ and $\tau_m(v)$) can be measured experimentally. The activation and inactivation functions are typically a sigmoid. The parameters defining the sigmoids are then related to the type of gate. As seen in Fig. 2.2, for large membrane potentials activating gates are more likely to be open. Conversely for inactivating gates. The time constants resemble a Gaussian, and smaller values of the time constant imply faster dynamics of the corresponding gating variable.

As a result, we have seen that neural activity can be modeled as a system of differential equations. Precisely, one for the voltage as in (2.3), and then as many first-order differential equations as gating variables and ionic channels. In the literature one might find many models following the basic formulae in (2.3) and (2.4), mostly varying on the number and type of gating variables and the activation/inactivation functions defining the dynamics of the gating variables. The pioneer work by [Hod52] has been followed by a plethora of alternative models such as [Mor81, Fit61, Nag62, Izh04]. Without loss of generality, in the sequel we consider one of the most popular single-neuron models: the Morris-Lecar model, which is introduced in the following section.

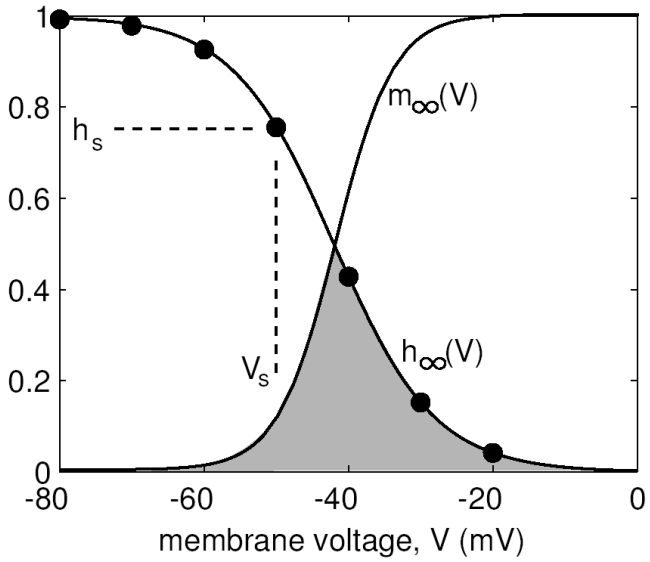


Figure 2.2: Sigmoid activation and inactivation functions (Source: [Izh06]).

2.3.1 Morris-Lecar model

From the myriad of existing single-neuron models, we consider the Morris-Lecar model proposed in [Mor81]. The model can be related (see [Izh06]) to the $I_{\mathrm{Na,p}} + I_{\mathrm{K}}$-model (pronounced persistent sodium plus potassium model). The dynamics of the neuron is modeled by a continuous-time dynamical system composed of the current-balance equation for the membrane potential, $v = v(t)$, and the K^+ gating variable $0 \leq n = n(t) \leq 1$, which represents the probability of the K^+ ionic channel to be active. Then, the system of differential equations is

$$C_m \dot{v} = -I_{\mathrm{L}} - I_{\mathrm{Ca}} - I_{\mathrm{K}} + I_{\mathrm{app}} \tag{2.8}$$

$$\dot{n} = \phi \frac{n_\infty(v) - n}{\tau_n(v)} \ , \tag{2.9}$$

where C_m is the membrane capacitance and ϕ a non-dimensional constant. I_{app} represents the (externally) applied current. For the time being, we have neglected I_{syn} in (2.8). The leakage, calcium, and potassium currents are of the form

$$I_{\mathrm{L}} = \bar{g}_{\mathrm{L}}(v - E_{\mathrm{L}}) \tag{2.10}$$

$$I_{\mathrm{Ca}} = \bar{g}_{\mathrm{Ca}} m_\infty(v)(v - E_{\mathrm{Ca}}) \tag{2.11}$$

$$I_{\mathrm{K}} = \bar{g}_{\mathrm{K}} n (v - E_{\mathrm{K}}) \ , \tag{2.12}$$

respectively. $\bar{g}_{\mathrm{L}}$, $\bar{g}_{\mathrm{Ca}}$, and $\bar{g}_{\mathrm{K}}$ are the maximal conductances of each current. E_{L}, E_{Ca}, and E_{K} denote the Nernst equilibrium potentials, for which the corresponding current is zero, a.k.a. reverse potentials.

The dynamics of the activation variable m is considered at the steady state, and thus we write $m = m_\infty(v)$. On the other hand, the time constant $\tau_n(v)$ for the gating variable n cannot be considered that fast and the corresponding differential equation needs to be considered. The formulae for these functions is

$$m_\infty(v) = \tfrac{1}{2} \cdot (1 + \tanh[\tfrac{v - V_1}{V_2}]) \tag{2.13}$$

$$n_\infty(v) = \tfrac{1}{2} \cdot (1 + \tanh[\tfrac{v - V_3}{V_4}]) \tag{2.14}$$

$$\tau_n(v) = 1/(\cosh[\tfrac{v - V_3}{2V_4}]) \ , \tag{2.15}$$

which parameters V_1, V_2, V_3, and V_4 can be measured experimentally [Izh06].

The knowledgeable reader would have noticed that the Morris-Lecar model is a Hodgin-Huxley type-model with the usual considerations, where the following two extra assumptions were made: the depolarizing current is generated by Ca^{2+} ionic channels (or Na^+ depending on the type of neuron modeled), whereas hyperpolarization is carried by K^+ ions; and that $m = m_\infty(v)$. The Morris-Lecar model is very popular in computational neuroscience as it models a large variety of neural dynamics while its phase-plane analysis is more manageable as it involves only two states [Rin98].

The Morris Lecar, although simple to formulate, results in a very interesting model as it can produce a number of different dynamics. For instance, for given values of its parameters, we encounter a subcritic Hopf bifurcation for $I_{\mathrm{app}} = 93.86\ \mu\mathrm{A/cm}^2$. On the other hand, for another set of parameter values, the system of equations has a Saddle-Node on an Invariant Circle (SNIC) bifurcation at $I_{\mathrm{app}} = 39.96\ \mu\mathrm{A/cm}^2$.

2.4 Synaptic inputs

The synaptic current, I_{syn}, gathers the contribution of neighboring neurons. I_{syn} is responsible for activating spike generation in neurons, without externally applying currents (e.g., via I_{app}). The most general model for I_{syn} considers decomposition in 2 independent components:

$$I_{\mathrm{syn}} = g_{\mathrm{E}}(t)(v(t) - E_{\mathrm{E}}) + g_{\mathrm{I}}(t)(v(t) - E_{\mathrm{I}}) \tag{2.16}$$

corresponding to excitatory (α-amino-3-hydroxy-5-methyl-4-isoxazolepropionic (AMPA) neuroreceptors) and inhibitory (γ-aminobutyric acid (GABA) neuroreceptors) terms, respectively. Roughly speaking, whereas the excitatory synaptic term makes the postsynaptic neuron more likely to generate a spike, the inhibitory term makes the postsynaptic

neuron less likely to generate an action potential. E_{E} and E_{I} are the corresponding reverse potentials. A longstanding problem is to characterize the time-varying global excitatory and inhibitory conductances $g_{\mathrm{E}}(t)$ and $g_{\mathrm{I}}(t)$. We present here 3 mathematical models for the synaptic current.

1. A classical model is to consider each of the 2 synapses as a whole, gathering the contributions of all the synapses of that type. In this case, similarly as for the voltage-dependent conductances, the synaptic conductance can be written as the product of its maximal conductance ($\bar{g}_u$) and the channel open probability (p_u),

$$g_u(t) = \bar{g}_u p_u \tag{2.17}$$

   where $u = \{\mathrm{E}, \mathrm{I}\}$. In turn, $p_u = s_u r_u$ can be expressed in terms of the process occurring on the pre- and post-synaptic sides: the probability that a postsynaptic channel is open state (s_u) and the probability that a transmitter is released by the pre-synaptic terminal (r). A simple model for s_u is similar to the model for opening/closing gating variables:

$$\dot{s}_u = \alpha_u(1 - s_u) - \beta_u s_u \tag{2.18}$$

   where α_u and β_u are the opening and closing rates of the channel u, respectively.

2. A further refinement of the above is to account for multiple postsynaptic channels. Particularly, if N_{E} and N_{I} denote the total number of excitatory and inhibitory synapses, then a plausible model is

$$I_{\mathrm{syn}} = \sum_{n_E=1}^{N_{\mathrm{E}}} g_{\mathrm{AMPA}} m_{\mathrm{E}}^{(n_E)}(t)(v(t) - E_{\mathrm{E}}) + \sum_{n_I=1}^{N_{\mathrm{I}}} g_{\mathrm{GABA}} m_{\mathrm{I}}^{(n_I)}(t)(v(t) - E_{\mathrm{I}}) \tag{2.19}$$

   where g_{AMPA} and g_{GABA} are maximal conductances, and $m_u^{(n)}(t)$ is the fraction of open post-synaptic receptors of the type u at each individual synapse n for a given time t. The kinetic equations for these variable are very similar to those in (2.18) with corresponding parameters. Notice that these individual dynamics differ in the time constant, defined by $1/\alpha_u$. This model is realistic, although might be intractable for estimation and simulation purposes if the number of synapses increases.

3. A third alternative is fundamentally different, it is referred to as *effective point-conductance* model in [Rud03, Piw04]. In this model, the excitatory/inhibitory global conductances are treated as Ornstein-Uhlenbeck (OU) processes

$$\dot{g}_u(t) = -\frac{1}{\tau_u}(g_u(t) - g_{u,0}) + \sqrt{\frac{2\sigma_u^2}{\tau_u}}\chi(t) \tag{2.20}$$

where $\chi(t)$ is a zero-mean, white noise, Gaussian process with unit variance. Then, the OU process has mean $g_{u,0}$, standard deviation σ_u, and time constant τ_u. Recall that we used the notation $u = \{\mathrm{E}, \mathrm{I}\}$. Although this is a much simpler model than (2.19), it was shown in [Rud03] that the OU model yields to a valid description of synaptic noise, capturing the properties of the more complex model.

Estimation of the synaptic conductances and its parameters is one of the main challenges in modern neuroscience. There is a lack for efficient algorithms able to estimate these quantities on-line, from a single trial, and robustly adapting to model uncertainties. These are goals addressed in this Thesis.

2.5 Intracellular recordings

The membrane potential, obtained from intracellular recordings, is one of the most valuable signals of neurons' activity. Most of the neuron models have been derived from fine measurements and allow the progress of "in silico" experiments. The recording of the membrane potential is a physical process, which involves two main issues not taken into account in the ideal model (2.3):

1. Voltage observations are noisy. This is due to the thermal noise at the sensing device, non-ideal conditions in experimental setups, etc.

2. Recorded observations are discrete. All sensing devices record data by sampling at regular time intervals the continuous-time natural phenomena. This is the task of an Analog-to-Digital Converter (ADC). Moreover, these samples are typically digitized, i.e. expressed by a finite number of bits. This latter issue is not tackled in the Thesis as we assume that modern computer capabilities allow us to sample with relatively large number of bits per sample.

Therefore, the problem investigated in this Thesis considers recordings of noisy voltage traces to infer the hidden gating variables of the neuron model, as well as filtered voltage estimates. Data is recorded at discrete time-instants k at a sampling frequency $f_s = 1/\mathrm{T}_s$. The problem can thus be posed in the form of a discrete-time, state-space model. The observations are

$$y_k \sim \mathcal{N}(v_k, \sigma^2_{y,k}) \ , \tag{2.21}$$

with $\sigma^2_{y,k}$ modeling the noise variance due to the sensor or the instrumentation inaccuracies when performing the experiment. To provide comparable results, we define the signal-to-noise ratio (SNR) as $\mathrm{SNR} = P_s/P_n$, with P_s being the average signal power and $P_n = \sigma^2_{y,k}$ the noise power.

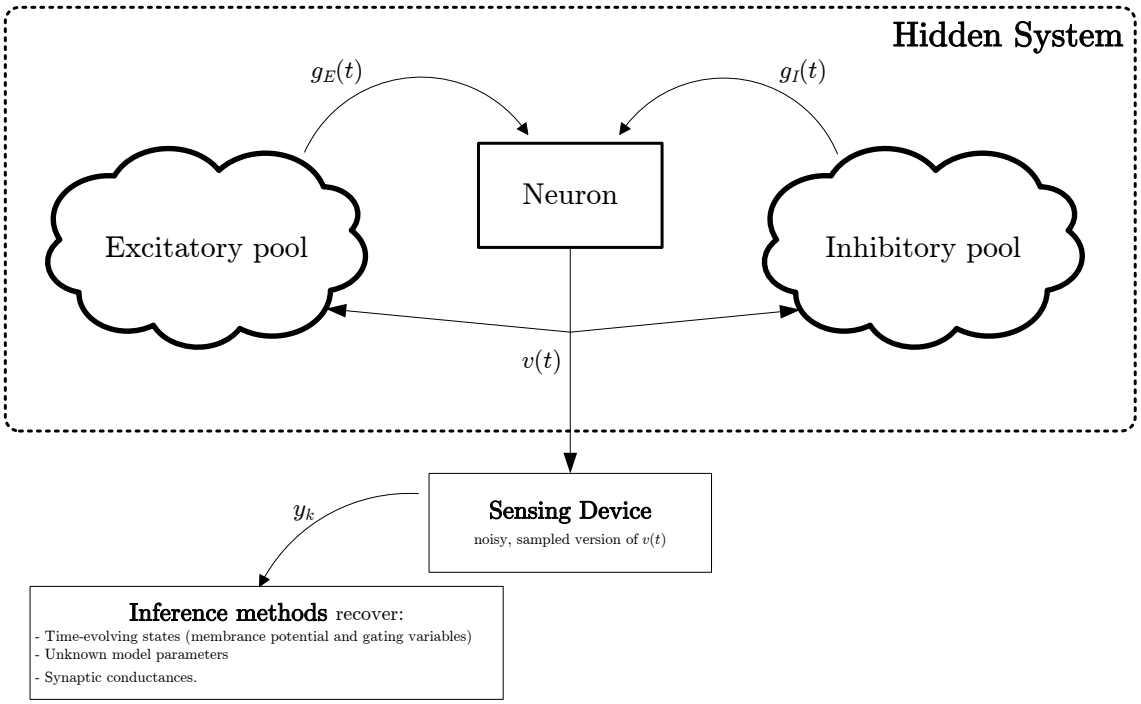


Figure 2.3: The experimental setup of interest in this Thesis.

Fig. 2.3 shows the basic setup we are dealing with in this Thesis. The neuron under observation has its own dynamics, producing electrical voltage patterns. It was discussed that the generation of action potentials is regulated by internal drivers (e.g., the active gating variables of the neuron) as well as exogenous factors like excitatory and inhibitory synaptic conductances produced by pools of connected neurons. This system is unobservable, in the sense that we cannot measure it directly. The sole observation from this system are the noisy membrane potentials y_k. In this experimental scenario, we aim at applying sequential inference methods to extract the following quantities:

1. The time-evolving states characterizing the neuron dynamics, including a filtered membrane potential and the dynamics of the gating variables.

2. The parameters defining the neuron model. It was seen that conductance-based models require the knowledge (or adjustment) of a number of static parameters. It is desirable to have autonomous inference algorithms that estimate these parameters as well, on the top of the time-evolving states.

3. The dynamics of synaptic conductances and its parameters. The final goal being to Discern the temporal contributions of global excitation from those of global inhibition, $g_{\mathrm{E}}(t)$ and $g_{\mathrm{I}}(t)$ respectively.

2.6 Summary

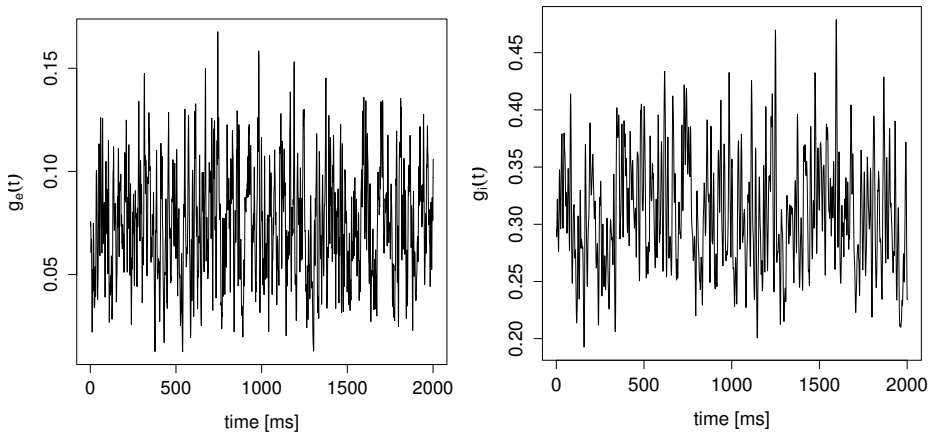
In this chapter, we gained some insight on the biological components of a neuron and the problem we aim at addressing. We presented conductance-based models for single-neurons, the biophysical meaning of its parameters, and explained in more detail a particular model named after Morris and Lecar. Without loss of generality, the latter model is used along the Thesis to validate the algorithms. In this discussion we left integrate-and-fire models on purpose [Izh06], although the methodology followed in the Thesis could be applied as well. Finally, we briefly sketched the practical implications of working with intracellular records which, basically, imply that observations are noisy and sampled.

Appendix 2.A Validation of the effective point-conductance model for synaptic conductances

This Appendix validates the OU model introduced in Section 2.4 with a set of realistic conductance measures. To do so we apply the tools from Bayesian modeling.

Particularly, we are interested in analyzing the suitability of two statistical models to explain synaptic conductances, $g_{\mathrm{E}}(t)$ and $g_{\mathrm{I}}(t)$. The first approach is based on the work in [Rud03, Piw04], where these terms were assumed to follow an OU process. The second approach considers this synaptic noise as a white noise process. Since most of the models are analogous for excitatory and inhibitory cells, in the sequel we use the generic index $u = \{\mathrm{E}, \mathrm{I}\}$ to simplify the equations. The results of the fitting can be consulted in Fig. 2.5(a)-2.6(b), and the model comparison metrics were in favor of the OU-based model. Basically, we looked at the model comparison metrics (i.e. Akaike's Information Criteria, AIC), and the posterior variances of the model parameters.

We have access to a set of conductances profiles from a computational network model of V1 with 128^2 integrate-and-fire neurons, which are representative to the cells of the layer $4C\alpha$ of the primary visual cortex [Tao04, McL00, Wie01]. See Figure 2.4 for the time-series of the data, where $T = 2000$ ms where recorded at a sampling period of $\mathrm{T}_s = 0.05$ seconds between consecutive samples.



(a) excitatory (b) inhibitory

Figure 2.4: Time-series of the excitatory and inhibitory synaptic conductances.

2.A.1 Synaptic conductances as an Ornstein-Uhlenbeck process

According to [Rud03, Piw04], a OU process is reasonable for the modeling of synaptic noise and has similar properties to more detailed (and thus complex) models. This appendix seeks validation of this statement. The OU process, in its discrete version using Euler's method, can be expressed

$$g_{u,k+1} = g_{u,k} - \frac{\mathrm{T}_s}{\tau_u}(g_{u,k} - g_{u,0}) + \mathrm{T}_s\sqrt{\frac{2\sigma_u^2}{\tau_u}}\chi \tag{2.22}$$

where χ is a zero-mean, independent white noise, Gaussian process with unit variance. Then, the OU process has mean $g_{u,0}$, standard deviation σ_u, and time constant τ_u.

The model in (2.22) can be rearranged as the following linear/Gaussian model:

$$g_{u,k+1} = \beta_{u0} + \beta_{u1}g_{u,k} + \sigma\chi \tag{2.23}$$

which is more convenient for the model fitting purposes we aim at. In (2.23) we defined:

$$\beta_{u0} = 1 - \frac{\mathrm{T}_s}{\tau_u} \tag{2.24}$$

$$\beta_{u1} = \mathrm{T}_s\frac{g_{u0}}{\tau_u} \tag{2.25}$$

$$\sigma = \mathrm{T}_s\sqrt{\frac{2\sigma_u^2}{\tau_u}}\ . \tag{2.26}$$

The observed data are the time-series of $g_{\mathrm{E}}(t)$ and $g_{\mathrm{I}}(t)$ that we plotted in Figure 2.4. Therefore, we can define a vector of observations as

$$\mathbf{g}_e = (g_{\mathrm{E},1}, \ldots, g_{\mathrm{E},T})^\top \tag{2.27}$$

$$\mathbf{g}_i = (g_{\mathrm{I},1}, \ldots, g_{\mathrm{I},T})^\top \tag{2.28}$$

and the vectors of explanatory variables as

$$\mathbf{x}_e = (0, g_{\mathrm{E},1}, \ldots, g_{\mathrm{E},T-1})^\top \tag{2.29}$$

$$\mathbf{x}_i = (0, g_{\mathrm{I},1}, \ldots, g_{\mathrm{I},T-1})^\top\ , \tag{2.30}$$

then, the model in (2.23) can be expressed in vector form and we come up with an statistical model such as:

$$\mathcal{M}_1 = \{\mathbf{g}_u|\beta_{u0}, \beta_{u1}, \tau \sim \mathcal{N}(\beta_{u0} + \beta_{u1}\mathbf{x}_u, \tau\mathbf{I}), (\beta_{u0}, \beta_{u1}, \tau) \in \mathbb{R}_+^3\}\ , \tag{2.31}$$

with $\mathbf{I}$ being the $n \times n$ identity matrix, and $\tau \doteq \sigma^{-2}$ the precision of the normal distribution (we are using WinBUGS notation here).

For the Bayesian model to be completed, we need to set a priori distributions for each of the unknown quantities in the statistical model $\mathcal{M}_1$, we select:

$$\begin{aligned} \beta_{u0} &\sim \pi(\beta_{u0}) = \mathcal{N}(0, 10^{-5}) && (2.32)\\ \beta_{u1} &\sim \pi(\beta_{u1}) = \mathcal{N}(0, 10^{-5}) && (2.33)\\ \tau &\sim \pi(\tau) = \Gamma(10^{-3}, 10^{-3}) \ . && (2.34) \end{aligned}$$

Finally, it is worth mentioning that we are able to recover the desired parameters of the OU model by undoing the transformations. It is straightforward to see from (2.24)–(2.26) that

$$\begin{aligned} \tau_u &= \frac{\mathrm{T}_s}{1-\beta_{u1}} && (2.35)\\ g_{u0} &= \frac{\beta_{u0}}{1-\beta_{u1}} && (2.36)\\ \sigma_u &= \frac{1}{2\tau \mathrm{T}_s}\frac{1}{(1-\beta_{u1})} \ , && (2.37) \end{aligned}$$

and it is even easier to obtain the a posteriori distributions of each parameter with WinBUGS (see code below).

The chunk of WinBUGS code implementing the OU model is:

```
model {

# statistical model:
for ( i in 1:N) {
          y[i] ~ dnorm(mu[i], tau.y)

      E[i] <- y[i]- mu[i]
      E2[i] <-  pow(E[i], 2)
}

mu[1] <- b0
for ( j in 2:N) {
      mu[j] <- b1*x[j] + b0
  }

# a priori distributions:
```

```
b0 ~ dnorm(0, 0.00001)
b1 ~ dnorm(0, 0.00001)
tau.y ~ dgamma(0.001, 0.001)

# obtaining the desired parameters:
tau <- Ts/(1-b1)
g0 <- b0/(1-b1)
sigma2 <- 1/(2*tau.y*Ts*(1-b1))

SQE <- sum(E2[])

}
```

2.A.2 Synaptic conductances as a white noise process

At the light of the time-series in Figure 2.4, one could be tempted to model the synaptic conductances as a white noise process of the form:

$$g_{u,k} = \alpha_{u0} + \sigma_u \chi \tag{2.38}$$

where χ is a zero-mean, white noise, Gaussian process with unit variance. The model has two unknown quantities α_{u0} and σ_u^2. Therefore, the statistical model might be described in this case as:

$$\mathcal{M}_2 = \{\mathbf{g}_u | \alpha_{u0}, \tau \sim \prod_{i=1}^{n} \mathcal{N}(\alpha_{u0}, \tau \doteq \sigma_u^{-2}), (\alpha_{u0}, \tau) \in \mathbb{R}_+^2\} \tag{2.39}$$

where we assumed i.i.d. observations. We chose the a priori distributions

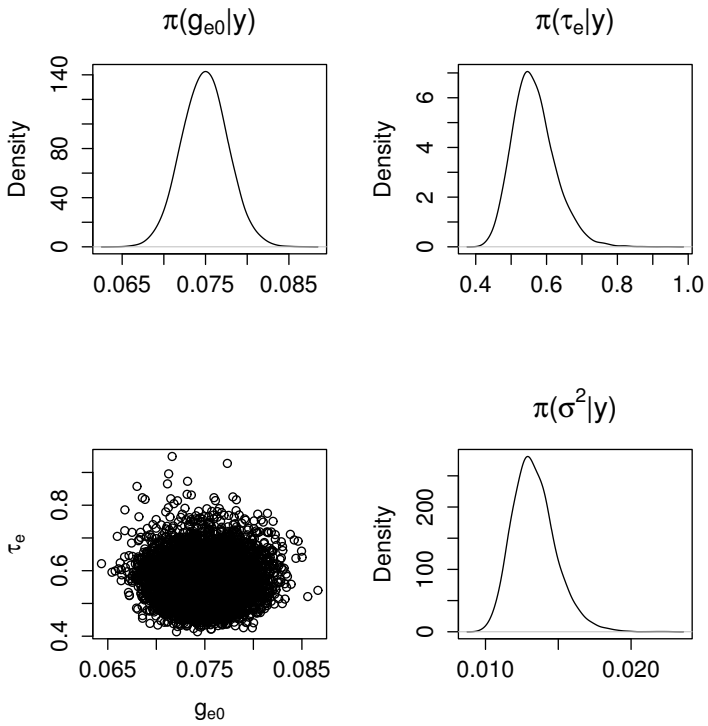
$$\alpha_{u0} \sim \pi(\alpha_{u0}) = \mathcal{N}(0, 10^{-5}) \tag{2.40}$$

$$\tau \sim \pi(\tau) = \Gamma(10^{-3}, 10^{-3}) \ . \tag{2.41}$$

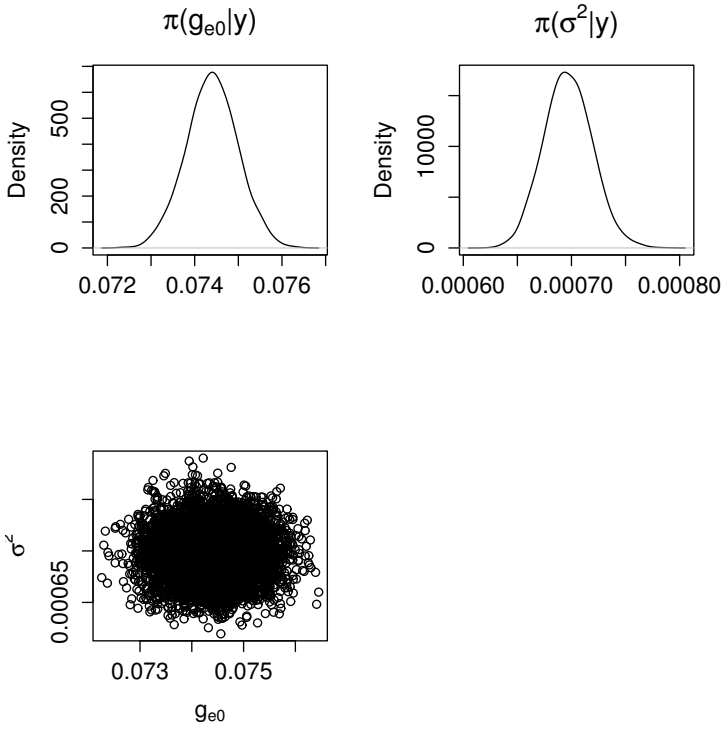
The chunk of WinBUGS code implementing the WN model is:

```
model {

# statistical model:
for ( i in 1:N) {
          y[i] ~ dnorm(a0, tau.y)
```

```
      #E[i] <- y[i]- a0
      #E2[i] <-  pow(E[i], 2)
}

# a priori distributions:
a0 ~ dnorm(0, 0.00001)
tau.y ~ dgamma(0.001, 0.001)

# obtaining the desired parameters:
sigma2 <- 1/tau.y

#SQE <- sum(E2[])

}
```

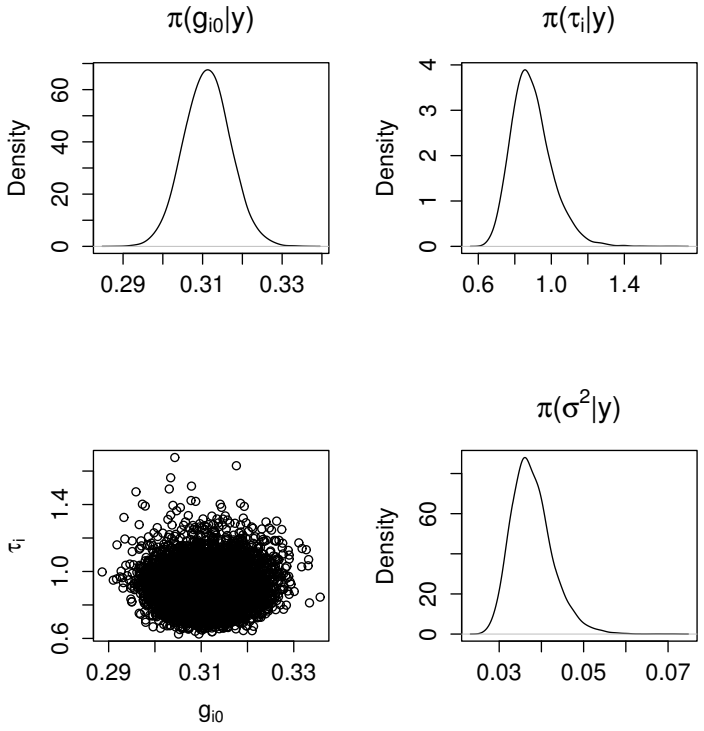


(a) OU-based model

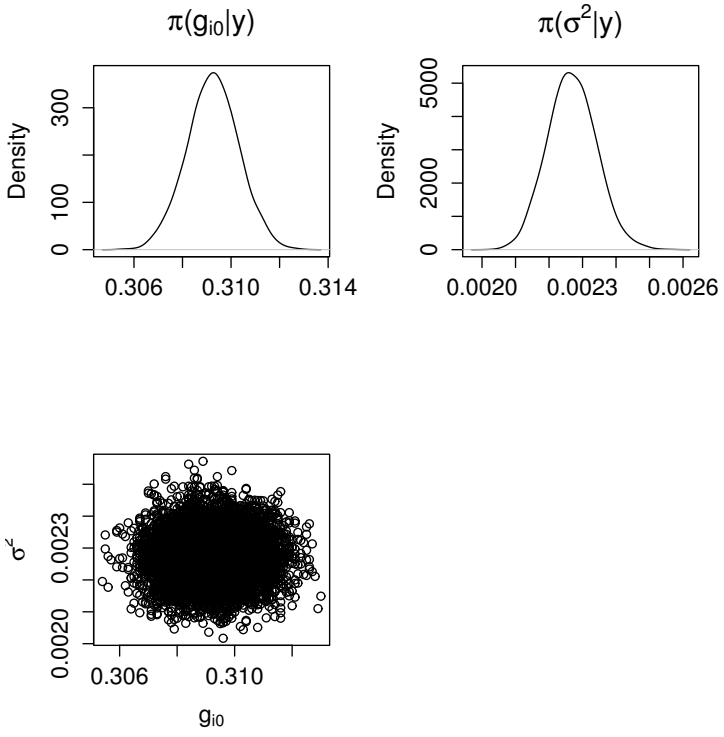


(b) WN-based model

Figure 2.5: A posteriori distributions for the parameters of $g_{\mathrm{E}}(t)$.



(a) OU-based model



(b) WN-based model

Figure 2.6: A posteriori distributions for the parameters of $g_{\mathrm{I}}(t)$.

3

Fundamentals of Bayesian Filtering

THE type of problems we are interested in involve the estimation of time-evolving parameters that can be expressed through a state-space (SS) formulation. Particularly, estimation of the states in a single-neuron model from noisy voltage traces can be readily seen as a filtering (even smoothing) problem. Bayesian theory provides the mathematical tools to deal with such problems in a systematic manner following the axioms of probability. The focus is on sequential methods that can incorporate new available measurements as they are recorded data without the need for reprocessing all past measurements. This is accomplished by the Bayesian filtering methodology and the related algorithms.

Ultimately, in Bayes theory we are interested in computing the marginal posterior distribution of having the system in a state $\boldsymbol{x}_k$ at discrete time k after observing data from 1 to k, that is $p(\boldsymbol{x}_k|\boldsymbol{y}_1, \ldots, \boldsymbol{y}_k)$. For instance, in the case of the Morris-Lecar model the states are the values of voltage traces and the gating variable.

This chapter presents an introductory overview on the optimal filtering framework and the algorithms that a practitioner has at hand. More precisely, we have organized the Chapter as follows. First, the optimal Bayesian filtering framework is presented in Section 3.1. Then, Section 3.2 reviews fundamental estimation bounds of filtering methods and Section 3.3 briefly sketches the algorithms one can use to implement Bayesian filters depending on the type of system under study. We have included an appendix with more details on the particle filtering methodology, which is of paramount importance in this

Thesis since methods designed in Chapter 4 are based on this approach. The details on the rest of algorithms are given here for the sake of completeness.

3.1 Bayesian nonlinear filtering over general state-space models

In general, the natural way to account for prior information is to consider a SS model. The SS representation provides a twofold modeling. On the one han, *state equation* illustrates the evolution of states with time. In other words, state equation mathematically expresses the prior information that the algorithm has regarding the state[1]. On the other hand, *measurement equation* models the dependency of measurements with unknown states. In this section, we introduce the general discrete state-space (DSS) model and the Bayesian conceptual solution, which is only analytically tractable when some assumptions hold. The section discusses both optimal and state-of-the-art suboptimal algorithms to obtain the Bayesian solution. We restrict ourselves to the discrete version of the SS model since it is the one required along this dissertation. Although similar, the continuous SS model has its own particularities that an interested reader can explore in detail in [And79].

3.1.1 Considering Prior information: the Bayesian recursion

The DSS approach deals with the nonlinear filtering problem: recursively compute estimates of states $\boldsymbol{x}_k \in \mathbb{R}^{n_x}$ given measurements $\boldsymbol{y}_k \in \mathbb{C}^{n_y}$ at time index k based on all available measurements, $\boldsymbol{y}_{1:k} = \{\boldsymbol{y}_1, \ldots, \boldsymbol{y}_k\}$. State equation models the evolution in time of target states as a discrete-time stochastic model, in general

$$\boldsymbol{x}_k = \mathbf{f}_{k-1}(\boldsymbol{x}_{k-1}, \boldsymbol{\nu}_k) \ , \tag{3.1}$$

where $\mathbf{f}_{k-1}(\cdot)$ is a known, possibly nonlinear, function of the state $\boldsymbol{x}_k$ and $\boldsymbol{\nu}_k$ is referred to as process noise which gathers any mismodeling effect or disturbances in the state characterization. In our case, for instance, $\mathbf{f}_{k-1}(\cdot)$ is defined by the ordinary differential equations (ODEs) describing the Morris-Lecar model where $\boldsymbol{x}_k$ includes the voltage traces and the gating variable. The relation between measurements and states is modeled by

$$\boldsymbol{y}_k = \mathbf{h}_k(\boldsymbol{x}_k, \mathbf{e}_k) \ , \tag{3.2}$$

where $\mathbf{h}_k(\cdot)$ is a known possibly nonlinear function and $\mathbf{e}_k$ is referred to as measurement noise. Both process and measurement noise are assumed with known statistics and mu-

[1]We understand by *state* the evolving (vector) *r.v.* that drives measurements and which is the estimation objective.

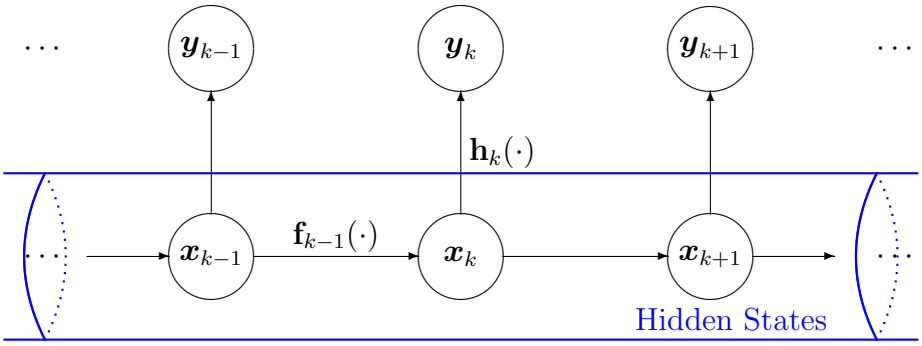


Figure 3.1: Graphical interpretation of the discrete state-space model as a Markov process of order one.

tually independent. The initial *a priori* distribution of the state vector is assumed to be known, $p(\boldsymbol{x}_0)$.

The methodology described in the sequel assumes that the DSS model describes a Markov process of order one, that is, the state at time instant k depends only on the previous state. This can be observed in (3.1) and (3.2). The Hidden Markov Model (HMM) is a statistical model where one Markov process, representing the underlying system, is observed through a stochastic process. Meaning that states are not directly observable, but measurements. The idea behind the HMM model is graphically shown in Figure 3.1, where it appears as evident that states are hidden and that the algorithm has access to measurements. In addition, the algorithm must have perfect knowledge of state and measurement equations, represented in Figure 3.1 by vertical and horizontal arrows respectively.

Alternatively, it might be convenient to express the DSS in terms of states and measurement distributions, i.e., prior and likelihood distributions respectively. We refer to this equivalent representation as the probabilistic DSS. This interpretation is equivalent to that in (3.1) and (3.2), but is useful in some problems. In this case, state equation is written as

$$\boldsymbol{x}_k \sim p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1}) \quad \text{for } k \geq 1 \ , \tag{3.3}$$

where $p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1})$ is referred to as the transitional prior. The relationship between measurements and states is generically modeled by the probability distribution

$$\boldsymbol{y}_k \sim p(\boldsymbol{y}_k|\boldsymbol{x}_k) \quad \text{for } k \geq 1 \ , \tag{3.4}$$

referred to as the likelihood function. Similarly as in the functional interpretation of the DSS , $p(\boldsymbol{x}_0)$ is assumed known.

From a Bayesian standpoint, the posterior distribution $p(\boldsymbol{x}_{0:k}|\boldsymbol{y}_{1:k})$ provides all necessary information about the state of the system $\boldsymbol{x}_{0:k}$, given all measurements $\boldsymbol{y}_{1:k}$ and the

prior $p(\boldsymbol{x}_{0:k})$. The *Bayes' theorem* allows to express the posterior in terms of the likelihood and prior distributions:

$$p(\boldsymbol{x}_{0:k}|\boldsymbol{y}_{1:k}) = \frac{p(\boldsymbol{y}_{1:k}|\boldsymbol{x}_{0:k})p(\boldsymbol{x}_{0:k})}{p(\boldsymbol{y}_{1:k})} \ , \tag{3.5}$$

which can be written as

$$p(\boldsymbol{x}_{0:k}|\boldsymbol{y}_{1:k}) = \frac{p(\boldsymbol{x}_0)\prod_{t=1}^{k} p(\boldsymbol{y}_t|\boldsymbol{x}_t)p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1})}{p(\boldsymbol{y}_{1:k})} \tag{3.6}$$

where we take into account that measurements are independent given $\boldsymbol{x}_{0:k}$ and consider the Markov state evolution depicted in Figure 3.1.

We are interested in the marginal distribution $p(\boldsymbol{x}_k|\boldsymbol{y}_{1:k})$ since, as will be seen later in this section, it allows the estimation of the realization of the target state vector $\boldsymbol{x}_k$. $p(\boldsymbol{x}_k|\boldsymbol{y}_{1:k})$ can be obtained by marginalization of (3.6), being the dimension of the integral growing with k. Alternatively the desired density[2] can be computed sequentially in two stages: prediction and update. The basic idea is to, assuming the filtering distribution known at $k-1$, first predict the new state and then incorporate the new measurement to obtain the distribution at k:

$$\cdots \longrightarrow p(\boldsymbol{x}_{k-1}|\boldsymbol{y}_{1:k-1}) \longrightarrow \underbrace{p(\boldsymbol{x}_k|\boldsymbol{y}_{1:k-1})}_{\text{prediction}} \longrightarrow \underbrace{p(\boldsymbol{x}_k|\boldsymbol{y}_{1:k})}_{\text{update}} \longrightarrow \cdots$$

First we notice from (3.6) that the posterior distribution can be recursively expressed as:

$$p(\boldsymbol{x}_{0:k}|\boldsymbol{y}_{1:k}) = \frac{p(\boldsymbol{y}_k|\boldsymbol{x}_k)p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1})}{p(\boldsymbol{y}_k|\boldsymbol{y}_{1:k-1})} p(\boldsymbol{x}_{0:k-1}|\boldsymbol{y}_{1:k-1}) \ , \tag{3.7}$$

where the marginal $p(\boldsymbol{x}_k|\boldsymbol{y}_{1:k})$ also satisfies the recursion [Sor88]. Given that $p(\boldsymbol{x}_0) \triangleq p(\boldsymbol{x}_0|\boldsymbol{y}_0)$ is known, where $\boldsymbol{y}_0$ is the set of no measurements, we can assume that the required density at time $k-1$ is available, $p(\boldsymbol{x}_{k-1}|\boldsymbol{y}_{1:k-1})$. In the prediction stage the predicted distribution is obtained by considering that $p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1}, \boldsymbol{y}_{1:k-1}) = p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1})$, due to the first-order Markovian SS model considered. Using the Chapman-Kolmogorov equation (see Appendix 3.A) to remove $\boldsymbol{x}_{k-1}$ we obtain,

$$p(\boldsymbol{x}_k|\boldsymbol{y}_{1:k-1}) = \int p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1})p(\boldsymbol{x}_{k-1}|\boldsymbol{y}_{1:k-1})d\boldsymbol{x}_{k-1} \ . \tag{3.8}$$

[2]In the sequel, $p(\boldsymbol{x}_k|\boldsymbol{y}_{1:k})$ is referred to as the filtering distribution. In contrast to $p(\boldsymbol{x}_{0:k}|\boldsymbol{y}_{1:k})$, which is the posterior distribution.

Whenever a new measurement becomes available at instant k, the predicted distribution in (3.8) is updated via the Bayes' rule (see Appendix 3.A)

$$\begin{aligned} p(\boldsymbol{x}_k|\boldsymbol{y}_{1:k}) &= p(\boldsymbol{x}_k|\boldsymbol{y}_k, \boldsymbol{y}_{1:k-1}) \\ &= \frac{p(\boldsymbol{y}_k|\boldsymbol{x}_k, \boldsymbol{y}_{1:k-1})p(\boldsymbol{x}_k|\boldsymbol{y}_{1:k-1})}{p(\boldsymbol{y}_k|\boldsymbol{y}_{1:k-1})} \\ &= \frac{p(\boldsymbol{y}_k|\boldsymbol{x}_k)p(\boldsymbol{x}_k|\boldsymbol{y}_{1:k-1})}{p(\boldsymbol{y}_k|\boldsymbol{y}_{1:k-1})} \ , \end{aligned} \tag{3.9}$$

being the normalizing factor

$$p(\boldsymbol{y}_k|\boldsymbol{y}_{1:k-1}) = \int p(\boldsymbol{y}_k|\boldsymbol{x}_k)p(\boldsymbol{x}_k|\boldsymbol{y}_{1:k-1})d\boldsymbol{x}_k \ . \tag{3.10}$$

Now the recursion is enclosed by equations (3.8) and (3.9), assuming some knowledge about the state evolution and the relation between measurements and states, described by $p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1})$ and $p(\boldsymbol{y}_k|\boldsymbol{x}_k)$ respectively. This recursion form the basis of the optimal Bayesian solution.

To sum up, the interest on characterizing the filtering distribution is that it enables one to compute optimal state estimates with respect to any criterion [Kay93], conditional upon measurements up to time k'. For example, the Minimum Mean Square Error (MMSE) estimator is extensively used in engineering applications, which is the conditional mean of the state with respect to available measurements,

$$\hat{\boldsymbol{x}}_k^{\text{MMSE}} = \mathbb{E}\left\{\boldsymbol{x}_k|\boldsymbol{y}_{1:k'}\right\} = \int \boldsymbol{x}_k p(\boldsymbol{x}_k|\boldsymbol{y}_{1:k'})d\boldsymbol{x}_k \ . \tag{3.11}$$

Another approach is to compute the Maximum a posteriori (MAP) estimate, which reduces to find the state value which maximizes the filtering distribution,

$$\hat{\boldsymbol{x}}_k^{\text{MAP}} = \arg\max_{\boldsymbol{x}_k} \left\{p(\boldsymbol{x}_k|\boldsymbol{y}_{1:k'})\right\} \ , \tag{3.12}$$

among many other criterions which can be used[3]. In general, we would like to compute any function $g(\cdot)$ of the state:

$$\widehat{g(\boldsymbol{x}_k)}^{\text{MMSE}} = \mathbb{E}\left\{g(\boldsymbol{x}_k)|\boldsymbol{y}_{1:k'}\right\} = \int g(\boldsymbol{x}_k)p(\boldsymbol{x}_k|\boldsymbol{y}_{1:k'})d\boldsymbol{x}_k \ , \tag{3.13}$$

conditional upon measurements up to time k'.

Depending on the value of k' we identify three different problems:

[3]Intuitively, the non-Bayesian counterparts to the MMSE and MAP estimators are the Least Squares (LS) and Maximum Likelihood (ML) estimators, respectively. In the latter, prior information is omitted or, equivalently, a noninformative prior is used.

Smoothing. It is the case of $k' > k$, were the state $\boldsymbol{x}_{0:k}$ is estimated using future measurements.

Filtering. Corresponds to the case $k' = k$. It is the problem considered in the sequel and the approach that is followed along the dissertation.

Prediction. In this case, one predicts values of states with measurements of previous instants: $k' < k$.

What the reader has read so far, corresponds to the conceptual solution of Bayesian filtering (smoothing and prediction) that is endowed with the sequential equations (3.8) and (3.9). In some particular cases, the recursion can be solved analytically and thus optimally. However, in general it cannot be solved and thus we need to resort to suboptimal algorithms. The most popular alternatives are discussed in Section 3.3. Before delving into the details of Bayesian filters, we devote Section 3.2 to understand the computation of theoretical lower bounds on estimation accuracy. The latter is of paramount importance to establish benchmarks and comparison among different estimation algorithms.

3.2 Posterior Cramér-Rao Bound

The Posterior Cramér-Rao Bound (PCRB) provides a lower bound on the MSE matrix for random parameters[4]. The Bayesian paradigm considers that the parameter of interest $\boldsymbol{\xi}$ is random with a given a priori distribution, denoted by $p(\boldsymbol{\xi})$. Then, the estimation error

$$\mathbf{C}(\hat{\boldsymbol{\xi}}) \triangleq \mathbb{E}_{\boldsymbol{y},\boldsymbol{\xi}} \left\{ \left(\hat{\boldsymbol{\xi}} - \boldsymbol{\xi} \right) \left(\hat{\boldsymbol{\xi}} - \boldsymbol{\xi} \right)^{\top} \right\} \tag{3.14}$$

is bounded as

$$\mathbf{C}(\hat{\boldsymbol{\xi}}) \geq \mathbf{J}_B^{-1}(\boldsymbol{\xi}) \ , \tag{3.15}$$

where $\mathbf{J}_B(\boldsymbol{\xi})$ is referred to as the Bayesian Information Matrix (BIM), its inverse provides the PCRB matrix [Tre07]. The matrix inequality in (3.15) means that $\mathbf{C}(\hat{\boldsymbol{\xi}}) - \mathbf{J}_B^{-1}(\boldsymbol{\xi})$ is a non-negative definite matrix. The BIM elements are computed as

$$\begin{aligned} \left[\mathbf{J}_B(\boldsymbol{\xi})\right]_{u,v} &\triangleq \mathbb{E}_{\boldsymbol{y},\boldsymbol{\xi}} \left\{ \frac{\partial \ln p(\boldsymbol{y},\boldsymbol{\xi})}{\partial \xi_u} \frac{\partial \ln p(\boldsymbol{y},\boldsymbol{\xi})}{\partial \xi_v} \right\} \\ &= -\mathbb{E}_{\boldsymbol{y},\boldsymbol{\xi}} \left\{ \frac{\partial^2 \ln p(\boldsymbol{y},\boldsymbol{\xi})}{\partial \xi_u \partial \xi_v} \right\} \ , \end{aligned} \tag{3.16}$$

[4]We can also find the Posterior Cramér-Rao Bound in the literature under the name of Bayesian Cramér-Rao Bound. Along the dissertation we used the former, as it is widespread used in nonlinear filtering literature.

with the expectations being over the joint distribution of data $\boldsymbol{y}$ and the parameter $\boldsymbol{\xi}$, i.e., $p(\boldsymbol{y}, \boldsymbol{\xi})$. Similarly as in the CRB, an Bayesian estimator of a random parameter is said to be Bayesian efficient when its variance attains the PCRB.

The BIM can be expressed as the summation of two terms: $\mathbf{J}_D(\boldsymbol{\xi})$ and $\mathbf{J}_P(\boldsymbol{\xi})$ [Tre68]. The former corresponds to the contribution of data measurements and the latter represents the information provided by prior, i.e.,

$$\mathbf{J}_B(\boldsymbol{\xi}) = \mathbf{J}_D(\boldsymbol{\xi}) + \mathbf{J}_P(\boldsymbol{\xi}) \ , \tag{3.17}$$

where the u, v-th element of each term is obtained as

$$\begin{aligned} \left[\mathbf{J}_D(\boldsymbol{\xi})\right]_{u,v} &\triangleq \mathbb{E}_{\boldsymbol{y},\boldsymbol{\xi}} \left\{ \frac{\partial \ln p(\boldsymbol{y}|\boldsymbol{\xi})}{\partial \xi_u} \frac{\partial \ln p(\boldsymbol{y}|\boldsymbol{\xi})}{\partial \xi_v} \right\} \\ &= -\mathbb{E}_{\boldsymbol{y},\boldsymbol{\xi}} \left\{ \frac{\partial^2 \ln p(\boldsymbol{y}|\boldsymbol{\xi})}{\partial \xi_u \partial \xi_v} \right\} \\ &= \mathbb{E}_{\boldsymbol{\xi}} \left\{ -\mathbb{E}_{\boldsymbol{y}|\boldsymbol{\xi}} \left\{ \frac{\partial^2 \ln p(\boldsymbol{y}|\boldsymbol{\xi})}{\partial \xi_u \partial \xi_v} \right\} \right\} \\ &= \mathbb{E}_{\boldsymbol{\xi}} \left\{ \left[\mathbf{J}_F(\boldsymbol{\xi})\right]_{u,v} \right\} \end{aligned} \tag{3.18}$$

and

$$\begin{aligned} \left[\mathbf{J}_P(\boldsymbol{\xi})\right]_{u,v} &\triangleq \mathbb{E}_{\boldsymbol{\xi}} \left\{ \frac{\partial \ln p(\boldsymbol{\xi})}{\partial \xi_u} \frac{\partial \ln p(\boldsymbol{\xi})}{\partial \xi_v} \right\} \\ &= -\mathbb{E}_{\boldsymbol{\xi}} \left\{ \frac{\partial^2 \ln p(\boldsymbol{\xi})}{\partial \xi_u \partial \xi_v} \right\} \ . \end{aligned} \tag{3.19}$$

From (3.18) we observe that the contribution of data to the Bayesian bound is equivalent to the expected value of $\mathbf{J}_F(\boldsymbol{\xi})$ over the distribution $p(\boldsymbol{\xi})$. Thus, the data contribution to the Bayesian estimation process corresponds to the averaged information matrix of the deterministic case, i.e., the Fischer Information Matrix. Moreover, when no prior data is considered, $\mathbf{J}_D(\boldsymbol{\xi})$ equals $\mathbf{J}_F(\boldsymbol{\xi})$ and the term $\mathbf{J}_P(\boldsymbol{\xi})$ disappears, which reduces to the deterministic case [Kay93].

So far, the PCRB was discussed from a static parameter estimation point of view. In other words, parameter $\boldsymbol{\xi}$ was the realization of a *r.v.* which we wanted to estimate. Another setup is possible. Dealing with the nonlinear filtering problem, tracking the time evolution of the parameter of interest (a.k.a. state vector) is the objective. As extensively discussed earlier, a state-space model can be used to characterize the evolution of the system and, particularly, we considered the discrete state-space model. We will see in Section 3.2.1 that, in that case, the evaluation of the PCRB can be demanding due to a dimensionality growth with time. Fortunately, a result due to [Tic98] allows one to compute the bound recursively. The result was extended in [Sim01] to the prediction and smoothing problems.

3.2.1 Recursive computation of the PCRB for nonlinear filtering

Let $\boldsymbol{y}_k \in \mathbb{C}^{n_y}$ be a vector of measured data, $\boldsymbol{x}_k \in \mathbb{R}^{n_x}$ an unknown random parameter and $\hat{\boldsymbol{x}}_k(\boldsymbol{y}_{1:k})$ an estimator of $\boldsymbol{x}_k$ considering available data at time instant k, $\boldsymbol{y}_{1:k} = \{\boldsymbol{y}_1, \ldots, \boldsymbol{y}_k\}$. The discrete state-space model provides a twofold characterization of the system under study, i.e., evolution of states and measurement dependence with states:

$$\boldsymbol{x}_k = \mathbf{f}_{k-1}(\boldsymbol{x}_{k-1}, \boldsymbol{\nu}_k) \tag{3.20}$$

$$\boldsymbol{y}_k = \mathbf{h}_k(\boldsymbol{x}_k, \mathbf{e}_k) \ , \tag{3.21}$$

respectively. $\mathbf{f}_{k-1}(\cdot)$ and $\mathbf{h}_k(\cdot)$ are known, possibly nonlinear, functions of the state $\boldsymbol{x}_k$. $\boldsymbol{\nu}_k$ and $\mathbf{e}_k$ are referred to as process and measurement noises, respectively. Both noises are assumed with known statistics and mutually independent. The initial *a priori* distribution of the state vector is assumed known, $p(\boldsymbol{x}_0)$.

For the filtering problem, the minimum theoretical achievable error variance is given by the PCRB [Ris04, Tre07, Ber01]. The PCRB states that the covariance matrix of the estimation error is bounded by the inverse of the Bayesian Information Matrix[5], $\mathbf{J}_k \in \mathbb{R}^{n_x \times n_x}$, i.e.,

$$\mathbf{C}_k(\boldsymbol{x}_k) \triangleq \mathbb{E}_{\boldsymbol{y}_k, \boldsymbol{x}_k} \left\{ (\hat{\boldsymbol{x}}_k(\boldsymbol{y}_{1:k}) - \boldsymbol{x}_k)(\hat{\boldsymbol{x}}_k(\boldsymbol{y}_{1:k}) - \boldsymbol{x}_k)^\top \right\} \geq \mathbf{J}_k^{-1} \ , \tag{3.22}$$

where the expectation is with respect to both measurements and states. The inequality in (3.22) means that the difference $\mathbf{C}_k(\boldsymbol{x}_k) - \mathbf{J}_k^{-1}$ is a positive semidefinite matrix and, if the equality holds, the estimator is said to be statistically efficient. Let the Trajectory Information Matrix, $\mathbf{J}(\boldsymbol{x}_{0:k}) \in \mathbb{R}^{(k+1)n_x \times (k+1)n_x}$, be the information matrix derived from the joint distribution for estimating $\boldsymbol{x}_{0:k}$ and defined as

$$\mathbf{J}(\boldsymbol{x}_{0:k}) = \mathbb{E}_{\boldsymbol{y}_k, \boldsymbol{x}_k} \left\{ -\triangle_{\boldsymbol{x}_{0:k}}^{\boldsymbol{x}_{0:k}} \ln p(\boldsymbol{y}_{1:k}, \boldsymbol{x}_{0:k}) \right\} \ , \tag{3.23}$$

where we define $\boldsymbol{x}_{0:k} = \{\boldsymbol{x}_0, \ldots, \boldsymbol{x}_k\}$ as the entire trajectory of state-vectors. We are interested in the problem of computing the BIM for estimating $\boldsymbol{y}_k$, to compute the bound in (3.22). Decomposing $\boldsymbol{x}_{0:k}$ and $\mathbf{J}(\boldsymbol{x}_{0:k})$ as

$$\boldsymbol{x}_{0:k} = \begin{pmatrix} \boldsymbol{x}_{0:k-1} \\ \boldsymbol{x}_k \end{pmatrix} \tag{3.24}$$

[5] where we dropped the subindex B for the sake of clarity, while keeping in mind that

$$\begin{aligned} \mathbf{J}_k &\triangleq \mathbf{J}_B(\boldsymbol{x}_k) \\ &= \mathbf{J}_D(\boldsymbol{x}_k) + \mathbf{J}_P(\boldsymbol{x}_k) \\ &= \mathbb{E}_{\boldsymbol{y}_k, \boldsymbol{x}_k} \left\{ \frac{\partial \ln p(\boldsymbol{y}_k|\boldsymbol{x}_k)}{\partial \boldsymbol{x}_k} \frac{\partial \ln p(\boldsymbol{y}_k|\boldsymbol{x}_k)}{\partial \boldsymbol{x}_k} \right\} + \mathbb{E}_{\boldsymbol{x}_k} \left\{ \frac{\partial \ln p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1})}{\partial \boldsymbol{x}_k} \frac{\partial \ln p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1})}{\partial \boldsymbol{x}_k} \right\} \ . \end{aligned}$$

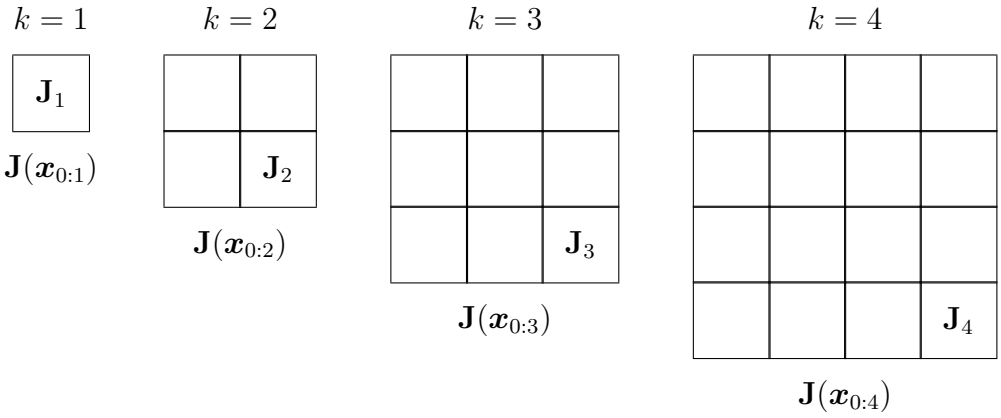


Figure 3.2: Dimensionality growth of the Trajectory Information Matrix with k.

and

$$\begin{aligned}\mathbf{J}(\boldsymbol{x}_{0:k}) &= \begin{pmatrix} \mathbf{A}_k & \mathbf{B}_k \\ \mathbf{B}_k^\top & \mathbf{C}_k \end{pmatrix} \\ &\triangleq \begin{pmatrix} \mathbb{E}_{\boldsymbol{y}_k,\boldsymbol{x}_k}\left\{-\triangle^{\boldsymbol{x}_{0:k-1}}_{\boldsymbol{x}_{0:k-1}} \ln p(\boldsymbol{y}_{1:k},\boldsymbol{x}_{0:k})\right\} & \mathbb{E}_{\boldsymbol{y}_k,\boldsymbol{x}_k}\left\{-\triangle^{\boldsymbol{x}_k}_{\boldsymbol{x}_{0:k-1}} \ln p(\boldsymbol{y}_{1:k},\boldsymbol{x}_{0:k})\right\} \\ \mathbb{E}_{\boldsymbol{y}_k,\boldsymbol{x}_k}\left\{-\triangle^{\boldsymbol{x}_{0:k-1}}_{\boldsymbol{x}_k} \ln p(\boldsymbol{y}_{1:k},\boldsymbol{x}_{0:k})\right\} & \mathbb{E}_{\boldsymbol{y}_k,\boldsymbol{x}_k}\left\{-\triangle^{\boldsymbol{x}_k}_{\boldsymbol{x}_k} \ln p(\boldsymbol{y}_{1:k},\boldsymbol{x}_{0:k})\right\} \end{pmatrix},\end{aligned} \tag{3.25}$$

respectively, we can compute the PCRB, $\mathbf{J}_k^{-1} \in \mathbb{R}^{n_x \times n_x}$, as the lower-right corner of $\mathbf{J}^{-1}(\boldsymbol{x}_{0:k})$, i.e.,

$$\mathbf{J}_k = \mathbf{C}_k - \mathbf{B}_k^\top \mathbf{A}_k^{-1} \mathbf{B}_k \ . \tag{3.26}$$

Notice that the computation of the $n_x \times n_x$ BIM involves either the inversion of $\mathbf{A}_k \in \mathbb{R}^{kn_x \times kn_x}$ or the inversion of $\mathbf{J}^{-1}(\boldsymbol{x}_{0:k})$. Clearly, this can imply a high computational cost.

As depicted in Figure 3.2, the dimensionality of $\mathbf{J}(\boldsymbol{x}_{0:k})$ grows with k. This poses a computational problem to the computation of the PCRB, which conveys the idea of deriving a recursive computation of the bound. Some papers proposed to relate the nonlinear filtering problem to an *equivalent* linear system, e.g., [Bob75]. However, the problem is partially solved in these approaches and still the recursion was to be found by [Tic98]. The latter provides a recipe for computing $\mathbf{J}_k$ without manipulating large matrices, such as $\mathbf{J}(\boldsymbol{x}_{0:k})$. Proposition 3.1 states the main result of that work.

Proposition 3.1. *The sequence* $\{\mathbf{J}_k\}$ *of posterior information submatrices for estimating state vectors* $\{\boldsymbol{x}_k\}$ *can be obtained using the following recursion:*

$$\mathbf{J}_{k+1} = \mathbf{D}_k^{22} - \mathbf{D}_k^{21} \left(\mathbf{J}_k + \mathbf{D}_k^{11}\right)^{-1} \mathbf{D}_k^{12} \ , \tag{3.27}$$

where

$$\begin{aligned}
\mathbf{D}_k^{11} &= \mathbb{E}_{\boldsymbol{x}_k,\boldsymbol{x}_{k+1}}\left\{-\triangle_{\boldsymbol{x}_k}^{\boldsymbol{x}_k}\ln p(\boldsymbol{x}_{k+1}|\boldsymbol{x}_k)\right\}\\
\mathbf{D}_k^{12} &= \mathbb{E}_{\boldsymbol{x}_k,\boldsymbol{x}_{k+1}}\left\{-\triangle_{\boldsymbol{x}_k}^{\boldsymbol{x}_{k+1}}\ln p(\boldsymbol{x}_{k+1}|\boldsymbol{x}_k)\right\}\\
\mathbf{D}_k^{21} &= \mathbb{E}_{\boldsymbol{x}_k,\boldsymbol{x}_{k+1}}\left\{-\triangle_{\boldsymbol{x}_{k+1}}^{\boldsymbol{x}_k}\ln p(\boldsymbol{x}_{k+1}|\boldsymbol{x}_k)\right\} = \left[\mathbf{D}_k^{12}\right]^\top\\
\mathbf{D}_k^{22} &= \mathbb{E}_{\boldsymbol{x}_k,\boldsymbol{x}_{k+1}}\left\{-\triangle_{\boldsymbol{x}_{k+1}}^{\boldsymbol{x}_{k+1}}\ln p(\boldsymbol{x}_{k+1}|\boldsymbol{x}_k)\right\}\\
&+ \mathbb{E}_{\boldsymbol{x}_{k+1},\boldsymbol{y}_{k+1}}\left\{-\triangle_{\boldsymbol{x}_{k+1}}^{\boldsymbol{x}_{k+1}}\ln p(\boldsymbol{y}_{k+1}|\boldsymbol{x}_{k+1})\right\}
\end{aligned} \tag{3.28}$$

and the initialization is done considering the prior density of the states:

$$\mathbf{J}_0 = \mathbb{E}_{\boldsymbol{x}_0}\left\{-\triangle_{\boldsymbol{x}_0}^{\boldsymbol{x}_0}\ln p(\boldsymbol{x}_0)\right\}\ . \tag{3.29}$$

Proof. See Appendix 3.B. □

The recursion in (3.27) is extremely useful in many cases where the computation of the PCRB is mathematically untractable. In addition, matrices involved in the recursive formula are $n_x \times n_x$, in contrast to the problem in equation (3.23) which has a dimension that increases with k (see Figure 3.2).

When the general DSS model described by (3.20) and (3.21) is particularized, some simplifications apply to the recursive computation of the PCRB in (3.27) and (3.28). The rest of the section is devoted to present those particularizations.

a) Additive Gaussian noise

In this case, the general DSS model is expressed as:

$$\begin{aligned}
\boldsymbol{x}_k &= \mathbf{f}_{k-1}(\boldsymbol{x}_{k-1}) + \boldsymbol{\nu}_k\\
\boldsymbol{y}_k &= \mathbf{h}_k(\boldsymbol{x}_k) + \mathbf{e}_k\ ,
\end{aligned} \tag{3.30}$$

where both process and measurement noise are zero-mean and Gaussian distributed, with covariance matrices being $\boldsymbol{\Sigma}_{x,k}$ and $\boldsymbol{\Sigma}_{y,k}$ respectively. Then, we have that

$$\begin{aligned}
-\ln p(\boldsymbol{x}_{k+1}|\boldsymbol{x}_k) &= c_1 + \frac{1}{2}\left(\boldsymbol{x}_{k+1} - \mathbf{f}_k(\boldsymbol{x}_k)\right)^\top \boldsymbol{\Sigma}_{x,k}^{-1}\left(\boldsymbol{x}_{k+1} - \mathbf{f}_k(\boldsymbol{x}_k)\right) \\
-\ln p(\boldsymbol{y}_{k+1}|\boldsymbol{x}_{k+1}) &= c_2 + \frac{1}{2}\left(\boldsymbol{y}_{k+1} - \mathbf{h}_{k+1}(\boldsymbol{x}_{k+1})\right)^\top \boldsymbol{\Sigma}_{y,k+1}^{-1}\left(\boldsymbol{y}_{k+1} - \mathbf{h}_{k+1}(\boldsymbol{x}_{k+1})\right)\ ,
\end{aligned} \tag{3.31}$$

where c_1 and c_2 are constants, and

$$\begin{aligned}\mathbf{D}_k^{11} &= \mathbb{E}_{\boldsymbol{x}_k}\left\{\tilde{\mathbf{F}}_k^{\top}\boldsymbol{\Sigma}_{x,k}^{-1}\tilde{\mathbf{F}}_k\right\}\\ \mathbf{D}_k^{12} &= -\mathbb{E}_{\boldsymbol{x}_k}\left\{\tilde{\mathbf{F}}_k^{\top}\right\}\boldsymbol{\Sigma}_{x,k}^{-1}\\ \mathbf{D}_k^{22} &= \boldsymbol{\Sigma}_{x,k}^{-1}+\mathbb{E}_{\boldsymbol{x}_{k+1}}\left\{\tilde{\mathbf{H}}_{k+1}^{\top}\boldsymbol{\Sigma}_{y,k+1}^{-1}\tilde{\mathbf{H}}_{k+1}\right\}\ .\end{aligned} \tag{3.32}$$

In (3.32) we use the definitions of Jacobian of $\mathbf{h}_k(\boldsymbol{x}_k)$ and $\mathbf{h}_k(\boldsymbol{x}_k)$ evaluated at the true value of $\boldsymbol{x}_k$:

$$\begin{aligned}\tilde{\mathbf{H}}_k &= \left[\nabla_{\boldsymbol{x}_k}\mathbf{h}_k^{\top}(\boldsymbol{x}_k)\right]^{\top}\\ \tilde{\mathbf{F}}_k &= \left[\nabla_{\boldsymbol{x}_k}\mathbf{f}_k^{\top}(\boldsymbol{x}_k)\right]^{\top}\ ,\end{aligned} \tag{3.33}$$

respectively.

The difficulty in evaluating (3.32) comes due to the need of performing the expectation over $\boldsymbol{x}_k$ and $\boldsymbol{x}_{k+1}$. The common approach is to approximate such expectations using Monte-Carlo simulation, i.e., create a significative number of state-vector trajectories, calculate the corresponding PCRB and average them to obtain the theoretical PCRB of the system under study. In certain cases, where the process noise is small, the expectation can be dropped out as a good approximation.

b) Linear systems under additive Gaussian noise

The linear DSS model corrupted by additive Gaussian noise reduces to :

$$\begin{aligned}\boldsymbol{x}_k &= \mathbf{F}_{k-1}\boldsymbol{x}_{k-1}+\boldsymbol{\nu}_k\\ \boldsymbol{y}_k &= \mathbf{H}_k\boldsymbol{x}_k+\mathbf{e}_k\ ,\end{aligned} \tag{3.34}$$

where $\mathbf{F}_{k-1}$ and $\mathbf{H}_k$ are known matrices that represent linear functions, referred to as *transitional* and *measurement* matrices respectively. $\boldsymbol{\nu}_k$ and $\mathbf{e}_k$ are mutually independent random variables drawn from a zero-mean white Gaussian probability density function with known covariance matrices, $\boldsymbol{\Sigma}_{x,k}$ and $\boldsymbol{\Sigma}_{y,k}$ respectively.

In this case, it is straightforward to show that the recursion can be expressed as:

$$\mathbf{J}_{k+1}=\boldsymbol{\Sigma}_{x,k}^{-1}-\boldsymbol{\Sigma}_{x,k}^{-1}\mathbf{F}_k\left(\mathbf{J}_k+\mathbf{F}_k^{\top}\boldsymbol{\Sigma}_{x,k}^{-1}\mathbf{F}_k\right)^{-1}\mathbf{F}_k^{\top}\boldsymbol{\Sigma}_{x,k}^{-1}+\mathbf{H}_{k+1}^{\top}\boldsymbol{\Sigma}_{y,k+1}^{-1}\mathbf{H}_{k+1}\ , \tag{3.35}$$

and using the matrix inversion lemma

$$\mathbf{J}_{k+1}=\left(\boldsymbol{\Sigma}_{x,k}+\mathbf{F}_k^{\top}\mathbf{J}_k^{-1}\mathbf{F}_k\right)^{-1}+\mathbf{H}_{k+1}^{\top}\boldsymbol{\Sigma}_{y,k+1}^{-1}\mathbf{H}_{k+1} \tag{3.36}$$

we obtain an expression with two terms: one corresponds to the process prediction and the other to the measurement update.

c) Linear states and nonlinear measurements under additive Gaussian noise

Consider a DSS model as:

$$\begin{aligned} \boldsymbol{x}_k &= \mathbf{F}_{k-1}\boldsymbol{x}_{k-1} + \boldsymbol{\nu}_k \\ \boldsymbol{y}_k &= \mathbf{h}_k(\boldsymbol{x}_k) + \mathbf{e}_k \ , \end{aligned} \tag{3.37}$$

where the previous definitions hold. Some simplifications apply to (3.32 under that DSS model. Considering that states are drawn from a Gaussian pdf and $p(\boldsymbol{x}_0) = \mathcal{N}(\bar{\boldsymbol{x}}_0, \boldsymbol{\Sigma}_{x,0})$, after mathematical manipulation of equation (3.29) we obtain that $\mathbf{J}_0 = \boldsymbol{\Sigma}_{x,0}^{-1}$. In addition,

$$\begin{aligned} \mathbf{D}_k^{11} &= \mathbf{F}_k^\top \boldsymbol{\Sigma}_{x,k}^{-1} \mathbf{F}_k \\ \mathbf{D}_k^{12} &= -\mathbf{F}_k^\top \boldsymbol{\Sigma}_{x,k}^{-1} \\ \mathbf{D}_k^{22} &= \boldsymbol{\Sigma}_{x,k}^{-1} + \mathbb{E}_{\boldsymbol{x}_{k+1}} \left\{ \tilde{\mathbf{H}}_{k+1}^\top \boldsymbol{\Sigma}_{y,k+1}^{-1} \tilde{\mathbf{H}}_{k+1} \right\} \ , \end{aligned} \tag{3.38}$$

where we use the definitions in (3.33). Notice that, after simplifications due to the model at hand, matrices $\mathbf{D}_k^{11}$, $\mathbf{D}_k^{12}$ and $\mathbf{D}_k^{21}$ are deterministic and can be easily obtained. However, due to the non-linearity in the measurement model, the expectation operator in the computation of $\mathbf{D}_k^{22}$ cannot be dropped out. In order to compute this expectation, a Monte-Carlo approximation can be performed as previously commented.

d) Nonlinear states and linear measurements under additive Gaussian noise

Consider a DSS model as:

$$\begin{aligned} \boldsymbol{x}_k &= \mathbf{f}_{k-1}(\boldsymbol{x}_{k-1}) + \boldsymbol{\nu}_k \\ \boldsymbol{y}_k &= \mathbf{H}_k\boldsymbol{x}_k + \mathbf{e}_k \ , \end{aligned} \tag{3.39}$$

where the transition equation is nonlinear/Gaussian and the measurements are expressed as a linear/Gaussian equation. Using (3.33), the components of the recursive PCRB formulation are

$$\begin{aligned} \mathbf{D}_k^{11} &= \mathbb{E}_{\boldsymbol{x}_k} \left\{ \tilde{\mathbf{F}}_k^\top \boldsymbol{\Sigma}_{x,k}^{-1} \tilde{\mathbf{F}}_k \right\} \\ \mathbf{D}_k^{12} &= -\mathbb{E}_{\boldsymbol{x}_k} \left\{ \tilde{\mathbf{F}}_k^\top \right\} \boldsymbol{\Sigma}_{x,k}^{-1} \\ \mathbf{D}_k^{22} &= \boldsymbol{\Sigma}_{x,k}^{-1} + \mathbf{H}_{k+1}^\top \boldsymbol{\Sigma}_{y,k+1}^{-1} \mathbf{H}_{k+1} \ . \end{aligned} \tag{3.40}$$

where we $\mathbf{D}_k^{22}$ becomes deterministic, but the rest of terms involving expectations have to be computed by Monte Carlo approximations.

This last case is the situation we will face in this Thesis: noisy/linear observations of states with nonlinear state evolution, everything under the Gaussian assumption.

3.3 Algorithms implementing Bayesian filtering

There are few cases where the Bayesian filtering problem can be analytically solved. An important case is when the system can be considered linear and Gaussian, in which case the Kalman filter (KF) [Kal60, And79] is known to be optimal. Another interesting case is when the SS is discrete and finite, where the optimal solution exists and can be found via optimal grid based methods [Ris04].

In more general models one must resort to suboptimal algorithms. A number of methods can be found in the literature [Che03]. A popular tool are particle filters (PFs) [Dou01a, Aru02, Dju03, Ris04] , a set of simulation-based methods which are applicable in very general nonlinear/non-Gaussian setups. Another appealing class of filters is composed of those variants of the KF which could operate with nonlinearities under the Gaussian assumption: extended Kalman filter (EKF) [And79, Kay93], unscented Kalman filter (UKF) [Jul97, Jul00, Wan00], central difference filter (CDF) [Ito00], divided difference filter (DDF) [Nor00], quadrature Kalman filter (QKF) [Ara08], or cubature Kalman filter (CKF) [Ara09]. Due to the Gaussian assumption, these algorithms only require the estimation of the first two moments of the distribution. They basically differ in the way they perform this estimation. Unlike the EKF, where only mean and covariance are propagated through the nonlinearity by means of a linearization process, the rest of mentioned KF–like algorithms resort to a set of so-called sigma-points to efficiently characterize the propagation of the normal distribution over the nonlinear system. Therefore, these filters are referred to with the general term of sigma–point Kalman filters (SPKFs) [Mer03]. In this section, we review the most popular Bayesian filtering algorithms.

3.3.1 The Kalman filter

Nobody can question that the KF has been one of the most used algorithms since its conception [Kal60]. Simple in its formulation, the KF provides an optimal Bayesian solution when the SS model is linear/Gaussian in both state and measurement equations. Thus, the model considered is

$$\begin{aligned} \boldsymbol{x}_k &= \mathbf{F}_{k-1}\boldsymbol{x}_{k-1} + \boldsymbol{\nu}_k \\ \boldsymbol{y}_k &= \mathbf{H}_k\boldsymbol{x}_k + \mathbf{e}_k \end{aligned} \tag{3.41}$$

where $\mathbf{F}_{k-1}$ and $\mathbf{H}_k$ are known matrices that represent linear functions, referred to as *transitional* and *measurement* matrices respectively. $\boldsymbol{\nu}_k$ and $\mathbf{e}_k$ are mutually independent random variables drawn from a zero-mean white Gaussian probability density function with known covariance matrices, $\boldsymbol{\Sigma}_{x,k}$ and $\boldsymbol{\Sigma}_{y,k}$ respectively.

The KF considers the posterior pdf as a Gaussian distribution, being completely characterized by its mean and covariance. Then, the prediction and update steps in equations (3.8) and (3.9) result in

$$\begin{aligned} p(\boldsymbol{x}_{k-1}|\boldsymbol{y}_{1:k-1}) &= \mathcal{N}\left(\boldsymbol{x}_{k-1};\ \hat{\boldsymbol{x}}_{k-1|k-1}, \mathbf{P}_{k-1|k-1}\right) \\ p(\boldsymbol{x}_{k}|\boldsymbol{y}_{1:k-1}) &= \mathcal{N}\left(\boldsymbol{x}_{k};\ \hat{\boldsymbol{x}}_{k|k-1}, \mathbf{P}_{k|k-1}\right) \\ p(\boldsymbol{x}_{k}|\boldsymbol{y}_{1:k}) &= \mathcal{N}\left(\boldsymbol{x}_{k};\ \hat{\boldsymbol{x}}_{k|k}, \mathbf{P}_{k|k}\right) \ . \end{aligned} \tag{3.42}$$

The KF provides the mean and covariance of each step in an iterative way [And79]:

$$\begin{aligned} \hat{\boldsymbol{x}}_{k|k-1} &= \mathbf{F}_{k-1}\hat{\boldsymbol{x}}_{k-1|k-1} \\ \mathbf{P}_{k|k-1} &= \boldsymbol{\Sigma}_{x,k} + \mathbf{F}_{k-1}\mathbf{P}_{k-1|k-1}\mathbf{F}_{k-1}^{\top} \\ \hat{\boldsymbol{x}}_{k|k} &= \hat{\boldsymbol{x}}_{k|k-1} + \mathbf{K}_k\left(\boldsymbol{y}_k - \mathbf{H}_k\hat{\boldsymbol{x}}_{k|k-1}\right) \\ \mathbf{P}_{k|k} &= \mathbf{P}_{k|k-1} - \mathbf{K}_k\mathbf{S}_k\mathbf{K}_k^{\top} \ , \end{aligned} \tag{3.43}$$

where it is defined the Kalman gain matrix as

$$\mathbf{K}_k = \mathbf{P}_{k|k-1}\mathbf{H}_k^{\top}\mathbf{S}_k^{-1} \tag{3.44}$$

and the variance of the innovation term as

$$\mathbf{S}_k = \mathbb{E}\left\{\left|\boldsymbol{y}_k - \mathbf{H}_k\hat{\boldsymbol{x}}_{k|k-1}\right|^2\right\} = \mathbf{H}_k\mathbf{P}_{k|k-1}\mathbf{H}_k^{\top} + \boldsymbol{\Sigma}_{y,k} \ . \tag{3.45}$$

Intuitively, a state prediction $\hat{\boldsymbol{x}}_{k|k-1}$ is done considering state equation in (3.41), i.e., a priori information. The state estimation at k is given by updating $\hat{\boldsymbol{x}}_{k|k-1}$ with a term that depends on the innovation error $\boldsymbol{y}_k - \mathbf{H}_k\hat{\boldsymbol{x}}_{k|k-1}$, which corrects the state prediction. The innovation error refers to the misadjusting between actual and predicted measurements, being controlled by the Kalman gain matrix. Initially, this matrix takes large values, since the main source of information are the measurements. Conversely, for increasing k, the updated values of $\mathbf{K}_k$ decrease since the algorithm gives more importance to prior data.

As said, the KF computes the mean and covariance matrices of the densities involved in (3.42) in a sequential way. In the case of linear/Gaussian models, the KF is the optimal solution. However, the assumptions might be too tight. They may not hold in some applications where the dependence of measurements on states is nonlinear or noises cannot be considered normally distributed or zero-biased. This is one of the main criticisms made against the use of such algorithm, which is nowadays still (widely) used.

3.3.2 Extended Kalman Filter

The KF can be extended to the case of nonlinear models of the form of:

$$\begin{aligned} \boldsymbol{x}_k &= \mathbf{f}_{k-1}(\boldsymbol{x}_{k-1}) + \boldsymbol{\nu}_k \\ \boldsymbol{y}_k &= \mathbf{h}_k(\boldsymbol{x}_k) + \mathbf{e}_k \end{aligned} \tag{3.46}$$

where $\mathbf{f}_{k-1}(\cdot)$ and $\mathbf{h}_k(\cdot)$ are known nonlinear functions. $\boldsymbol{\nu}_k$ and $\mathbf{e}_k$ are mutually independent random variables drawn from a zero-mean white Gaussian probability density function with known covariance matrices $\boldsymbol{\Sigma}_{x,k}$ and $\boldsymbol{\Sigma}_{y,k}$, respectively.

The approach taken by the EKF is based on a local linearization of the model, while maintaining the Gaussian constraint on the involved density functions [Jaz70]. The EKF approximates the posterior pdf as a Gaussian in the vein of (3.42). Thus, the posterior characterization is provided by its estimated mean and covariances. Similarly as done in the KF algorithm, the mean and covariance of both predictive and updated distributions are obtained in a sequential way [And79]:

$$\begin{aligned} \hat{\boldsymbol{x}}_{k|k-1} &= \mathbf{f}_{k-1}(\hat{\boldsymbol{x}}_{k-1|k-1}) \\ \mathbf{P}_{k|k-1} &= \boldsymbol{\Sigma}_{x,k} + \hat{\mathbf{F}}_{k-1}\mathbf{P}_{k-1|k-1}\hat{\mathbf{F}}_{k-1}^{\top} \\ \hat{\boldsymbol{x}}_{k|k} &= \hat{\boldsymbol{x}}_{k|k-1} + \mathbf{K}_k\left(\boldsymbol{y}_k - \mathbf{h}_k(\hat{\boldsymbol{x}}_{k|k-1})\right) \\ \mathbf{P}_{k|k} &= \mathbf{P}_{k|k-1} - \mathbf{K}_k\mathbf{S}_k\mathbf{K}_k^{\top} \end{aligned} \tag{3.47}$$

where

$$\begin{aligned} \mathbf{K}_k &= \mathbf{P}_{k|k-1}\hat{\mathbf{H}}_k^{\top}\mathbf{S}_k^{-1} \\ \mathbf{S}_k &= \mathbb{E}\left\{\left|\boldsymbol{y}_k - \hat{\mathbf{H}}_k\hat{\boldsymbol{x}}_{k|k-1}\right|^2\right\} = \hat{\mathbf{H}}_k\mathbf{P}_{k|k-1}\hat{\mathbf{H}}_k^{\top} + \boldsymbol{\Sigma}_{y,k}\ . \end{aligned} \tag{3.48}$$

The computation of this first and second order statistics is done after linearizing measurement and/or state evolution functions in (3.46). Local linearizations of $\mathbf{f}_{k-1}(\hat{\boldsymbol{x}}_{k-1|k-1})$ and $\mathbf{h}_k(\hat{\boldsymbol{x}}_{k|k-1})$ are obtained by the Gradient evaluated at the point of interest as

$$\begin{aligned} \hat{\mathbf{F}}_{k-1} &= \left[\nabla_{\boldsymbol{x}_{k-1}}\mathbf{f}_{k-1}^{\top}(\boldsymbol{x}_{k-1})\right]^{\top}\Big|_{\boldsymbol{x}_{k-1}=\hat{\boldsymbol{x}}_{k-1|k-1}} \\ \hat{\mathbf{H}}_k &= \left[\nabla_{\boldsymbol{x}_k}\mathbf{h}_k^{\top}(\boldsymbol{x}_k)\right]^{\top}\Big|_{\boldsymbol{x}_k=\hat{\boldsymbol{x}}_{k|k-1}}, \end{aligned} \tag{3.49}$$

respectively.

As said the EKF deals with nonlinearities in the model. However, the posterior density is still modeled as being Gaussian. Thus, it might fail in applications where the Gaussian

assumption is not valid or the nonlinearity is severe, which causes the true posterior to differ from Gaussianity. In addition, the EKF requires an accurate initialization of states and covariances in order to converge to the optimal solution, since the approximations in (3.49) are local.

3.3.3 The family of sigma-point Kalman filters

It is known that the EKF fails under severe nonlinearities. However, if the Gaussian assumption is still valid both for process and measurement noises we can seriously consider the family of SPKFs to infer the states of our system. As shown in [Ito00], for an arbitrary nonlinearity, we can derive a recursive algorithm by equating the Bayesian formulae with respect to the first and second moments of the distributions, that is, means and covariances of the conditional densities involved. The Bayesian filtering problem is then cast into a numerical evaluation of the involved integrals in (3.8) and (3.9).

As stated in the previous Section, there are a number of alternatives to evaluate such integrals under the Gaussian assumption. Here we are interested in SPKFs, a family of derivative-free algorithms which are based on a weighted sum of function values at specified (*i.e.*, deterministic) points within the domain of integration, as opposite to the stochastic sampling performed by particle filtering methods. A pseudocode description of a generic SPKF is given in Algorithm 3.1, where it is easy to recognize the underlying KF-like structure. This algorithm is valid for any SPKF–like algorithm, only differing on the generation of sigma-points in step 1. Notice that weights are normalized such that $\sum_i \omega_i^{(m)} = \sum_i \omega_i^{(c)} = 1$.

A further improvement of the standard SPKF scheme comes from the fact that, when we propagate the covariance matrix through a nonlinear function, the filter should preserve the properties of a covariance matrix, namely, its symmetry, and positive-definiteness. In practice, however, due to lack of arithmetic precision, numerical errors may lead to a loss of these properties. To circumvent this problem, a square-root filter can be considered to propagate the square root of the covariance matrix instead of the covariance itself. Even more, it avoids the inversion of the updated covariance matrix, entailing additional computational saving [Pot63]. The idea has been applied to the UKF [Van01] (although it does not guarantee positive definiteness of the covariance matrix) and, more successfully, to the Square–Root Quadrature Kalman Filter (SQKF)[Ara08] and the Square–Root Cubature Kalman Filter (SCKF) [Ara09].

As an outcome of Algorithm 3.1 one obtains the two distributions in (3.8) and (3.9) as

$$p(\mathbf{x}_k|\mathbf{y}_{1:k-1}) \approx \mathcal{N}\left(\hat{\mathbf{x}}_{k|k-1}, \boldsymbol{\Sigma}_{x,k|k-1}\right) \tag{3.50}$$

$$p(\mathbf{x}_k|\mathbf{y}_{1:k}) \approx \mathcal{N}\left(\hat{\mathbf{x}}_{k|k}, \boldsymbol{\Sigma}_{x,k|k}\right) , \tag{3.51}$$

respectively, which can be directly used to obtain an estimate of the states of interest.

Here we discuss the most popular algorithms falling in the SPKF category. As said, their sole difference is on the generation and weighting of sigma-points. At a glance:

UKF: It is based on the unscented transform (UT), whose sigma-points generation and weighting can be consulted[6] in Algorithm 3.3. The UT uses $2n_x + 1$ points to optimally characterize mean and covariance propagation of up to third order nonlinearities, under the Gaussian assumption. The square–root unscented Kalman filter (SRUKF) was first introduced in [Van01]. Intuitively, the role of parameter λ is to adjust the distance among the generated sigma–points and the mean. Therefore, if the dimension of the state–space increases, the distance also increases, forcing the transform to take into account non-local effects. The parameters required for the UT to operate are: α is a positive scaling parameter which can be made arbitrarily small to minimize higher order effects (*e.g.* 10^{-3}), and it determines the spread of sigma-points around a mean value; κ is a secondary parameter typically set to 0; and $\beta = 2$ was seen to be optimal in the Gaussian case [Jul04].

CKF: Recently, the Cubature Kalman filter (CKF) and its square-root version (SR-CKF) were proposed in [Ara09]. The CKF is based on a spherical–radial cubature rule [Str71] to generate and weight the sigma-points in Algorithm 3.1 and thus approximate the integrals in (3.8) and (3.9). Specifically, the authors claim that a third–degree cubature rule is enough in many filtering problems over higher–degree rules. Moreover, a third-degree spherical-radial cubature rule is said to be an optimal approximation to the Bayesian filter when the Gaussian assumption holds. This setup yields a total of $2n_x$ sigma–points, details found in Algorithm 3.4.

QKF: Another alternative for numerically solving the integrals in (3.8) and (3.9) is to resort to the Gauss–Hermite quadrature rules [Gol73]. Recently, this procedure was used to develop the so-called Quadrature Kalman filter (QKF) [Ito00, Ara07] and the corresponding square-root version (SRQKF) in [Ara08], exhibiting remarkable performance results. A single parameter is required for the use of Gauss–Hermite quadrature rule, which is the number of sigma–points per dimension γ. The pseudocode can be consulted in Algorithm 3.5. The square–root quadrature Kalman

[6] where $\mathbf{1}_i$ denotes the vector having a 1 at the i-th element and 0 otherwise.

Algorithm 3.1 Sigma-point Kalman Filter

Require: $\hat{\mathbf{x}}_0$, $\mathbf{\Sigma}_{x,0} = \mathbf{S}_{x,0|0}\mathbf{S}_{x,0|0}^\top$, $\mathbf{Q}_0$, $\mathbf{R}_0$.

1: Define sigma–points and weights $\{\boldsymbol{\xi}_i, \omega_i^{(m)}, \omega_i^{(c)}\}_{i=1,...,L}$
2: Set $k \Leftarrow 1$

Time update

3: Factorize the estimation covariance matrix:
$$\mathbf{\Sigma}_{x,k-1|k-1} = \mathbf{S}_{x,k-1|k-1}\mathbf{S}_{x,k-1|k-1}^\top.$$
4: Evaluate the sigma points:
$$\mathbf{x}_{i,k-1|k-1} = \mathbf{S}_{x,k-1|k-1}\boldsymbol{\xi}_i + \hat{\mathbf{x}}_{k-1|k-1},\ i = 1, ..., L.$$
5: Evaluate the propagated sigma-points:
$$\tilde{\mathbf{x}}_{i,k|k-1} = \mathbf{f}_{k-1}(\mathbf{x}_{i,k-1|k-1}).$$
6: Estimate the predicted state:
$$\hat{\mathbf{x}}_{k|k-1} = \textstyle\sum_{i=1}^{L} \omega_i^{(m)} \tilde{\mathbf{x}}_{i,k|k-1}.$$
7: Estimate the predicted error covariance:
$$\mathbf{\Sigma}_{x,k|k-1} = \textstyle\sum_{i=1}^{L} \omega_i^{(c)} \tilde{\mathbf{x}}_{i,k|k-1}\tilde{\mathbf{x}}_{i,k|k-1}^\top - \mathbf{\Delta}_{x,k},$$
with $\mathbf{\Delta}_{x,k} = \hat{\mathbf{x}}_{k|k-1}\hat{\mathbf{x}}_{k|k-1}^\top - \mathbf{\Sigma}_{\boldsymbol{\nu},k-1}$.

Measurement update

8: Factorize the predicted error covariance matrix:
$$\mathbf{\Sigma}_{x,k|k-1} = \mathbf{S}_{x,k|k-1}\mathbf{S}_{x,k|k-1}^\top.$$
9: Evaluate the sigma points:
$$\mathbf{x}_{i,k|k-1} = \mathbf{S}_{x,k|k-1}\boldsymbol{\xi}_i + \hat{\mathbf{x}}_{k|k-1},\ i = 1, ..., L.$$
10: Evaluate the propagated sigma-points:
$$\tilde{\mathbf{y}}_{i,k|k-1} = \mathbf{h}_k(\mathbf{x}_{i,k|k-1}).$$
11: Estimate the predicted measurement:
$$\hat{\mathbf{y}}_{k|k-1} = \textstyle\sum_{i=1}^{L} \omega_i^{(m)} \tilde{\mathbf{y}}_{i,k|k-1}.$$
12: Estimate the innovation covariance matrix:
$$\mathbf{\Sigma}_{y,k|k-1} = \textstyle\sum_{i=1}^{L} \omega_i^{(c)} \tilde{\mathbf{y}}_{i,k|k-1}\tilde{\mathbf{y}}_{i,k|k-1}^\top - \mathbf{\Delta}_{y,k},$$
with $\mathbf{\Delta}_{y,k} = \hat{\mathbf{y}}_{k|k-1}\hat{\mathbf{y}}_{k|k-1}^\top - \mathbf{\Sigma}_{n,k}$.
13: Estimate the cross–covariance matrix
$$\mathbf{\Sigma}_{xy,k|k-1} = \textstyle\sum_{i=1}^{L} \omega_i^{(c)} \tilde{\mathbf{x}}_{i,k|k-1}\tilde{\mathbf{y}}_{i,k|k-1}^\top - \hat{\mathbf{x}}_{k|k-1}\hat{\mathbf{y}}_{k|k-1}^\top.$$
14: Estimate the Kalman gain
$$\mathbf{K}_k = \mathbf{\Sigma}_{xy,k|k-1}\mathbf{\Sigma}_{y,k|k-1}^{-1}.$$
15: Estimate the updated state
$$\hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k\left(\mathbf{y}_k - \hat{\mathbf{y}}_{k|k-1}\right).$$
16: Estimate the corresponding error covariance:
$$\mathbf{\Sigma}_{x,k|k} = \mathbf{\Sigma}_{x,k|k-1} - \mathbf{K}_k\mathbf{\Sigma}_{y,k|k-1}\mathbf{K}_k^\top.$$
17: Set $k \Leftarrow k+1$ and go to step 4.

filter (SRQKF) algorithm has the appealing feature of coping with arbitrary nonlinearities since a γ-point Gauss–Hermite quadrature rule is exact for nonlinearities of order $2\gamma - 1$. The total number of generated points is γ^{n_x}. Thus, the SRQKF has one major drawback: the curse of dimensionality [Clo12b]. It is clear that for high–dimensional systems the implementation of the method is unfeasible for computational reasons (even for $\gamma = 3$), and alternatives should be explored.

3.3.4 Particle filtering for nonlinear/nonGaussian systems

In the recent years PFs played an important role in many research areas such as signal detection and demodulation, target tracking, positioning, Bayesian inference, audio processing, financial modeling, computer vision, robotics, control or biology [Spe02, Dju03, Pun03, Kar05, Hen08, Clo09]. The reason being its ability to deal with general nonlinear/nonGaussian systems. PFs are also referred to as Sequential Monte-Carlo (SMC) methods, since these tools are simulation-based methods to suboptimally (although close to it) sample from the marginal filtering distribution in situations where analytical solutions are hard to work out, or simply impossible. Many interesting textbooks on the topic can be consulted [Dou01a, Aru02, Che03, Ris04, Sär13]. For the sake of completeness, we have included a tutorial introduction to PFs in Appendix 3.D. In this section, only the basic algorithm is discussed.

Notice at this point that PFs constitute the core methods in this Thesis, and thus we devote here some effort in its understanding. The rest of algorithms presented earlier are provided for the sake of completeness, but bear in mind that the methods designed in Chapter 4 are based on the PF methodology.

Recall that Bayesian filtering involves the recursive estimation of states $\boldsymbol{x}_k \in \mathbb{R}^{n_x}$ given measurements $\boldsymbol{y}_k \in \mathbb{R}^{n_y}$ at time t based on all available measurements, $\boldsymbol{y}_{1:k} = \{\boldsymbol{y}_1, \ldots, \boldsymbol{y}_k\}$. To that aim, we are interested in the marginal filtering distribution $p(\boldsymbol{x}_k|\boldsymbol{y}_{1:k})$, which can be recursively expressed as

$$p(\boldsymbol{x}_k|\boldsymbol{y}_{1:k}) = \frac{p(\boldsymbol{y}_k|\boldsymbol{x}_k)p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1})}{p(\boldsymbol{y}_k|\boldsymbol{y}_{1:k-1})} p(\boldsymbol{x}_{k-1}|\boldsymbol{y}_{1:k-1}) \ , \tag{3.52}$$

with $p(\boldsymbol{y}_k|\boldsymbol{x}_k)$ and $p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1})$ referred to as the likelihood and the prior distributions, respectively. Unfortunately, (3.52) can only be obtained in closed-form in few special cases. Under the Gaussian assumption, we have seen in previous sections that Kalman-like methods provide interesting solutions. In more general setups – nonlinear and/or non-Gaussian – we should resort to more sophisticated methods such as PFs [Aru02, Dju03].

PFs approximate the filtering distribution by a set of N weighted random samples, forming the so-called set of particles $\left\{\boldsymbol{x}_k^{(i)}, w_k^{(i)}\right\}_{i=1}^{N}$. These random samples are drawn

Algorithm 3.2 Sequential Importance Sampling with systematic resampling

Require: $\left\{\boldsymbol{x}_{k-1}^{(i)}, w_{k-1}^{(i)}\right\}_{i=1}^{N}$ and $\boldsymbol{y}_k$

Ensure: $\left\{\boldsymbol{x}_{k}^{(i)}, w_{k}^{(i)}\right\}_{i=1}^{N}$ and $\hat{\boldsymbol{x}}_k$

1: **for** $i = 1$ to N **do**

2: Generate $\boldsymbol{x}_k^{(i)} \sim \pi(\boldsymbol{x}_k|\boldsymbol{x}_{0:k-1}^{(i)}, \boldsymbol{y}_{1:k})$

3: Calculate $\tilde{w}_k^{(i)} = w_{k-1}^{(i)} \frac{p(\boldsymbol{y}_k|\boldsymbol{x}_{0:k}^{(i)}, \boldsymbol{y}_{1:k-1}) p(\boldsymbol{x}_k^{(i)}|\boldsymbol{x}_{k-1}^{(i)})}{\pi(\boldsymbol{x}_k^{(i)}|\boldsymbol{x}_{0:k-1}^{(i)}, \boldsymbol{y}_{1:k})}$

4: **end for**

5: **for** $i = 1$ to N **do**

6: Normalize weights: $w_k^{(i)} = \frac{\tilde{w}_k^{(i)}}{\sum_{j=1}^{N} \tilde{w}_k^{(j)}}$

7: **end for**

8: MMSE state estimation: $\hat{\boldsymbol{x}}_k = \sum_{i=1}^{N} w_k^{(i)} \boldsymbol{x}_k^{(i)}$

9: $\left\{\boldsymbol{x}_{k}^{(i)}, 1/N\right\}_{i=1}^{N} = \text{Resample}(\left\{\boldsymbol{x}_{k}^{(i)}, w_{k}^{(i)}\right\}_{i=1}^{N})$

from the importance density distribution, $\pi(\cdot)$,

$$\boldsymbol{x}_k^{(i)} \sim \pi(\boldsymbol{x}_k|\boldsymbol{x}_{0:k-1}^{(i)}, \boldsymbol{y}_{1:k}) \tag{3.53}$$

and weighted according to the general formulation

$$w_k^{(i)} \propto w_{k-1}^{(i)} \frac{p(\boldsymbol{y}_k|\boldsymbol{x}_{0:k}^{(i)}, \boldsymbol{y}_{1:k-1}) p(\boldsymbol{x}_k^{(i)}|\boldsymbol{x}_{k-1}^{(i)})}{\pi(\boldsymbol{x}_k^{(i)}|\boldsymbol{x}_{0:k-1}^{(i)}, \boldsymbol{y}_{1:k})} \ . \tag{3.54}$$

The derivation of the weights expression can be consulted in Appendix 3.D, here we remark the idea that the random particles are weighted according its probability of occurrence a posteriori. Notice that the expression accounts for the likelihood and the a priori distributions, as well as the previous posterior given by the weights at $k-1$. This resembles the expression in (3.52).

Algorithm 3.2 outlines the operation of the standard PF when a new measurement $\boldsymbol{y}_k$ becomes available. After particle generation, weighting and normalization, a Minimum Mean Square Error (MMSE) estimate can be obtained by a weighted sum of particles. A typical problem of PFs is the degeneracy of particles, where all but one weight tend to zero. This situation causes the particle to collapse to a single state point. To avoid the degeneracy problem, we apply resampling, consisting in eliminating particles with low importance weights and replicating those in high-probability regions [Bol04b, Dou05]. In this work, we consider a multinomial sampling scheme for the resampling step.

There are many types of PFs based on the aforementioned concepts of Sequential Importance Sampling (SIS), the most popular being Sampling Importance Resampling (SIR) filter [Gor93], Auxiliary SIR (ASIR) filter [Pit01, God01], Regularized particle filter [Mus00], Local Linearization particle filter [Mer00], Gaussian particle filter [Kot03a, Kot03b], Multiple Model particle filter [McG00], Rao-Blackwellized particle filter [Sch05], Cost Reference particle filter [Míg04], and multiple particle filtering [Bug07, Clo12a].

3.4 Summary

This chapter presented the basic ideas and tools in Bayesian filtering theory. At a glance, these tools become handy when inferring the evolution of signals that can be formulated in state-space. The conceptual solution is not always analytically tractable, and thus we analyzed the most popular approximation to the problem. We also discussed on theoretical lower bounds of accuracy and recursive formulae to compute these benchmark curves.

Appendix 3.A Useful equalities

The Chapman-Kolmogorov equation

Conditional densities can be manipulated in order to obtain more tractable expressions by removing some variables. Denote some random variables as x_1, x_2 and x_3. If we want to remove x_2 from the joint pdf $f(x_1, x_2|x_3)$, we integrate with respect to this variable,

$$f(x_1|x_3) = \int_{-\infty}^{\infty} f(x_1|x_2, x_3)dx_2 \ . \tag{3.55}$$

On the other hand, if x_2 has to be removed from $f(x_1|x_2, x_3)$, the *Chapman-Kolmogorov* equation [Pap01] is extensively used:

$$f(x_1|x_3) = \int_{-\infty}^{\infty} f(x_1, x_2|x_3)f(x_2|x_3)dx_2 \ . \tag{3.56}$$

The Bayes' rule

Bayes' rule [Bay63] states that the probability density function of an event x conditioned to the event y can be expressed as

$$p(x|y) = \frac{p(y|x)p(x)}{p(y)} \ , \tag{3.57}$$

where the denominator can be computed using the *Total probability theorem*:

$$p(y) = \int_{-\infty}^{\infty} p(y|x)p(x)dx \ . \tag{3.58}$$

Appendix 3.B Proof of Proposition 3.1

Proceeding as in [Tic98], let us write the joint distribution of measurements and states as

$$\begin{aligned} p(\boldsymbol{y}_{1:k+1}, \boldsymbol{x}_{0:k+1}) &= p(\boldsymbol{y}_{1:k}, \boldsymbol{x}_{0:k}) p(\boldsymbol{x}_{k+1}|\boldsymbol{x}_{0:k}, \boldsymbol{y}_{1:k}) p(\boldsymbol{y}_{k+1}|\boldsymbol{x}_{k+1}, \boldsymbol{x}_{0:k}, \boldsymbol{y}_{1:k}) \\ &= p(\boldsymbol{y}_{1:k}, \boldsymbol{x}_{0:k}) p(\boldsymbol{x}_{k+1}|\boldsymbol{x}_k) p(\boldsymbol{y}_{k+1}|\boldsymbol{x}_{k+1}) \ . \end{aligned} \tag{3.59}$$

Using the decomposition of $\mathbf{J}(\boldsymbol{x}_{0:k})$ made in (3.25), the definitions in (3.28) and the expression for the joint distribution in (3.59), the Trajectory Information Matrix can be written in block form as

$$\mathbf{J}(\boldsymbol{x}_{0:k+1}) = \begin{pmatrix} \mathbf{A}_k & \mathbf{B}_k & \mathbf{0} \\ \mathbf{B}_k^\top & \mathbf{C}_k + \mathbf{D}_k^{11} & \mathbf{D}_k^{12} \\ \mathbf{0} & \mathbf{D}_k^{21} & \mathbf{D}_k^{22} \end{pmatrix} \ . \tag{3.60}$$

The desired BIM can be found as the $n_z \times n_z$ lower-right submatrix of $\mathbf{J}^{-1}(\boldsymbol{x}_{0:k+1})$:

$$\begin{aligned} \mathbf{J}_{k+1} &= \mathbf{D}_k^{22} - (\mathbf{0}, \mathbf{D}_k^{21}) \begin{pmatrix} \mathbf{A}_k & \mathbf{B}_k \\ \mathbf{B}_k^\top & \mathbf{C}_k + \mathbf{D}_k^{11} \end{pmatrix}^{-1} \begin{pmatrix} \mathbf{0} \\ \mathbf{D}_k^{21} \end{pmatrix} \\ &= \mathbf{D}_k^{22} - \mathbf{D}_k^{21} \left(\mathbf{C}_k + \mathbf{D}_k^{11} - \mathbf{B}_k^\top \mathbf{A}_k^{-1} \mathbf{B}_k \right)^{-1} \mathbf{D}_k^{12} \ , \end{aligned} \tag{3.61}$$

which follows from basic algebra [Gol96]. □

Appendix 3.C Sigma-point generation

Algorithm 3.3 Generation of sigma-points and weights using the unscented transformation

Require: n_x, α, κ, β

1: Set $L = 2n_x + 1$.
2: Set scaling parameter $\lambda = \alpha^2(n_x + \kappa) - n_x$.
3: Set sigma-points:
$\boldsymbol{\xi}_0 = \mathbf{0}$
$\boldsymbol{\xi}_i = \sqrt{(n_x + \lambda)}\ \mathbf{1}_i, \quad i = 1, \ldots, n_x$
$\boldsymbol{\xi}_i = \sqrt{(n_x + \lambda)}\ \mathbf{1}_{i-n_x}, \quad i = n_x + 1, \ldots, 2n_x$
4: Set mean weights:
$\omega_0^{(m)} = \lambda/(n_x + \lambda)$
$\omega_i^{(m)} = 0.5/(n_x + \lambda), \quad i = 1, \ldots, 2n_x$
5: Set covariance weights:
$\omega_0^{(c)} = \lambda/(n_x + \lambda) + (1 - \alpha^2 + \beta)$
$\omega_i^{(c)} = 0.5/(n_x + \lambda), \quad i = 1, \ldots, 2n_x$

Algorithm 3.4 Generation of sigma-points and weights for third-degree spherical-radial cubature rule

Require: n_x

1: Set $L = 2n_x$.
2: Set the cubature points $\boldsymbol{\xi}_i = \sqrt{n_x}\left[\mathbf{I}_{n_x \times n_x} \,\vdots\, -\mathbf{I}_{n_x \times n_x}\right]_i$, where $[\cdot]_{i=1,\ldots,L}$ indicates the i–th column.
3: Set the cubature weights $\omega_i = \frac{1}{2n_x}$, $i = 1, \ldots, L$.
4: Same mean and covariance weights:
$$\left.\begin{array}{l} \omega_i^{(m)} = \omega_i \\ \omega_i^{(c)} = \omega_i \end{array}\right\} \quad i = 1, \ldots, L$$

Algorithm 3.5 Generation of sigma-points and weights for multi-dimensional Gauss-Hermite quadrature rule

Require: n_x, γ

Set $L = \gamma^{n_x}$

Set $J_{i,i+1} = \sqrt{\frac{i}{2}}$, where $i = 1, \ldots, (\gamma - 1)$

Compute λ_i, the eigenvalues of $\mathbf{J}$

Set $\boldsymbol{\xi}_i = \sqrt{2}\lambda_i$

Set $\omega_i = (e_i)_1^2$, where $(e_i)_1$ is the first element of the i-th normalized eigenvector of $\mathbf{J}$

if $n_x > 1$ **then**

  Set $\boldsymbol{\zeta}_{1,:} = \boldsymbol{\xi}_i$ and $\boldsymbol{\varpi}_{1,:} = \omega_i$

  for $m = 2$ to n_x **do**

    Set $\boldsymbol{\xi}_{1:m-1,:} = \boldsymbol{\zeta} \otimes \mathbf{1}_{1\times\gamma}$

    Set $\boldsymbol{\xi}_{m,:} = \mathbf{1}_{1\times\gamma} \otimes \boldsymbol{\zeta}_{m-1,:}$

    Set $\boldsymbol{\Omega}_{1:m-1,:} = \boldsymbol{\varpi} \otimes \mathbf{1}_{1\times\gamma}$

    Set $\boldsymbol{\Omega}_{m,:} = \mathbf{1}_{1\times\gamma} \otimes \boldsymbol{\varpi}_{m-1,:}$

    Set $\boldsymbol{\zeta} = \boldsymbol{\xi}$ and $\boldsymbol{\varpi} = \boldsymbol{\Omega}$

  end for

  $\boldsymbol{\xi}_i = \boldsymbol{\xi}_{:,i}$, where $i = 1, \ldots, L$

  $\omega_i = \prod_{l=1}^{n_x} \boldsymbol{\Omega}_{l,i}$

end if

Same mean and covariance weights:

$$\left.\begin{array}{l} \omega_i^{(m)} = \omega_i \\ \omega_i^{(c)} = \omega_i \end{array}\right\} \quad i = 1, \ldots, L$$

Appendix 3.D A Brief Introduction to Particle Filters

The basic ideas of SMC methods date back to [Ham54], but the widespread use in nonlinear filtering theory is much more recent, the main reason being the high computational cost required, which is only affordable in modern computing facilities. This appendix presents a short overview of the fundamental concepts in particle filtering.

3.D.1 Monte-Carlo integration

The aim of PFs is to recursively estimate the posterior distribution $p(\boldsymbol{x}_{0:k}|\boldsymbol{y}_{1:k})$, the marginal filtering distribution $p(\boldsymbol{x}_k|\boldsymbol{y}_{1:k})$ and its associated expectations

$$I(g_k) = \mathbb{E}\left\{g_k(\boldsymbol{x}_{0:k})|\boldsymbol{y}_{1:k}\right\} = \int g_k(\boldsymbol{x}_{0:k}) p(\boldsymbol{x}_{0:k}|\boldsymbol{y}_{1:k}) d\boldsymbol{x}_{0:k} \ , \tag{3.62}$$

with $g_k(\cdot)$ a function of the state vector and the DSS model being as defined by equations (3.1) and (3.2) – or alternatively by (3.3) and (3.4).

PFs rely on the Monte-Carlo integration concept. Assume that we can simulate N_s independent identically distributed ($i.i.d.$) random samples from $p(\boldsymbol{x}_{0:k}|\boldsymbol{y}_{1:k})$, this set is denoted as $\{\boldsymbol{x}_{0:k}^{(i)}, i = 1, \ldots, N_s\}$. Then, the estimates of the distribution and its expectation are

$$\begin{aligned} p_s(\boldsymbol{x}_{0:k}|\boldsymbol{y}_{1:k}) &= \frac{1}{N_s}\sum_{i=1}^{N_s} \delta(\boldsymbol{x}_{0:k} - \boldsymbol{x}_{0:k}^{(i)}) \\ I_s(g_k) &= \int g_k(\boldsymbol{x}_{0:k}) p_s(\boldsymbol{x}_{0:k}|\boldsymbol{y}_{1:k}) d\boldsymbol{x}_{0:k} = \frac{1}{N_s}\sum_{i=1}^{N_s} g_k(\boldsymbol{x}_{0:k}^{(i)}) \ , \end{aligned} \tag{3.63}$$

respectively. The estimate $I_s(g_k)$ is unbiased and

$$P\left\{\lim_{N_s\to\infty} I_s(g_k) = I(g_k)\right\} = 1 \ , \tag{3.64}$$

according to the law of large numbers [Cri02]. Moreover, if the variance of g_k is finite, i.e.,

$$\begin{aligned} \sigma_{g_k}^2 &= \int (g_k(\boldsymbol{x}_{0:k}) - I(g_k))^2 p(\boldsymbol{x}_{0:k}|\boldsymbol{y}_{1:k}) d\boldsymbol{x}_{0:k} \\ &= \int g_k^2(\boldsymbol{x}_{0:k}) p(\boldsymbol{x}_{0:k}|\boldsymbol{y}_{1:k}) d\boldsymbol{x}_{0:k} - I^2(g_k) < \infty \ , \end{aligned} \tag{3.65}$$

then by the *central limit theorem* the estimation error converges in distribution, that is

$$\lim_{N_s \to \infty} \sqrt{N_s}(I_s(g_k) - I(g_k)) \sim \mathcal{N}\left(0, \sigma_{g_k}^2\right) \ . \tag{3.66}$$

Notice also that the rate of convergence is independent of the dimension of the integral, at least in theory. However, it is well known that PFs suffer from the curse of dimensionality [Dau03]. The latter convergence results are the basis of the success of PFs, compared to other suboptimal algorithms that lack of theoretical foundations to ensure convergence to the true posterior.

At this point one can wonder why such a method is named Monte-Carlo *integration*. Actually, it is a historically nomenclature since this method was first used to numerically solve integrals [Pap01]. For example, if we want to integrate a function $f(\boldsymbol{x})$, we can split it into two components: one plays the role of the probability density function $\vartheta(\boldsymbol{x})$ while the other is the function $g(\boldsymbol{x})$. Thus, the integral is obtained as

$$I = \int f(\boldsymbol{x})d\boldsymbol{x} = \int g(\boldsymbol{x})\vartheta(\boldsymbol{x})d\boldsymbol{x} \approx \frac{1}{N_s}\sum_{i=1}^{N_s} g(\boldsymbol{x}^{(i)}) \ , \tag{3.67}$$

where N_s points are generated from $\vartheta(\boldsymbol{x})$.

3.D.2 Importance Sampling and Sequential Importance Sampling

Unfortunately, usually it is not possible to sample effectively from the posterior distribution as required in (3.63), since it can be a complicated/unknown distribution. A classical alternative is the *Importance Sampling* (IS) method [Gew89]. Imagine we can only generate samples from a density $\pi(\boldsymbol{x}_{0:k}|\boldsymbol{y}_{1:k})$ which is *similar* to $p(\boldsymbol{x}_{0:k}|\boldsymbol{y}_{1:k})$, which means that both functions have the same support (in equation (3.79) and the corresponding text, further details will be given on this issue). We refer to $\pi(\boldsymbol{x}_{0:k}|\boldsymbol{y}_{1:k})$ as the *importance* or *proposal* density function. We can write equation (3.62) as

$$\begin{aligned} I(g_k) &= \int g_k(\boldsymbol{x}_{0:k})p(\boldsymbol{x}_{0:k}|\boldsymbol{y}_{1:k})d\boldsymbol{x}_{0:k} \\ &= \int g_k(\boldsymbol{x}_{0:k})\frac{p(\boldsymbol{x}_{0:k}|\boldsymbol{y}_{1:k})}{\pi(\boldsymbol{x}_{0:k}|\boldsymbol{y}_{1:k})}\pi(\boldsymbol{x}_{0:k}|\boldsymbol{y}_{1:k})d\boldsymbol{x}_{0:k} \\ &= \int g_k(\boldsymbol{x}_{0:k})\tilde{w}(\boldsymbol{x}_{0:k})\pi(\boldsymbol{x}_{0:k}|\boldsymbol{y}_{1:k})d\boldsymbol{x}_{0:k} \ , \end{aligned} \tag{3.68}$$

provided that $\tilde{w}(\boldsymbol{x}_{0:k}) = \frac{p(\boldsymbol{x}_{0:k}|\boldsymbol{y}_{1:k})}{\pi(\boldsymbol{x}_{0:k}|\boldsymbol{y}_{1:k})}$ is upper bounded. Applying the Monte-Carlo integration method, we draw N_s independent samples from the importance density function,

$\left\{\boldsymbol{x}_{0:k}^{(i)}\right\}_{i=1}^{N_s}$, then an estimate of $I(g_k)$ is

$$I_s(g_k) = \frac{1}{N_s}\sum_{i=1}^{N_s} g_k(\boldsymbol{x}_{0:k}^{(i)})w(\boldsymbol{x}_{0:k}^{(i)}) \ , \tag{3.69}$$

where

$$w(\boldsymbol{x}_{0:k}^{(i)}) = \frac{\tilde{w}(\boldsymbol{x}_{0:k}^{(i)})}{\sum_{j=1}^{N_s}\tilde{w}(\boldsymbol{x}_{0:k}^{j})} \tag{3.70}$$

are known as the normalized importance weights, and

$$\tilde{w}(\boldsymbol{x}_{0:k}^{(i)}) = \frac{p(\boldsymbol{x}_{0:k}|\boldsymbol{y}_{1:k})}{\pi(\boldsymbol{x}_{0:k}^{(i)}|\boldsymbol{y}_{1:k})} \tag{3.71}$$

are the unnormalized importance weights, which are normalized in (3.70) to obtain a proper probability density function.

The IS method has one main drawback which makes it inadequate for recursive filtering purposes: to estimate $p(\boldsymbol{x}_{0:k}|\boldsymbol{y}_{1:k})$ one needs to have all data $\boldsymbol{y}_{1:k}$ available. Then, when new data is available, one has to compute the importance weights over the entire state trajectory. Thus, the computational complexity increases with time. Instead, we are interested in an algorithm that is able to include new data in the estimation process without recomputing weights from the scratch.

In an attempt to obtain a sequential algorithm relying on IS method, the SIS algorithm is obtained. It is a Monte-Carlo method that forms the basis of most SMC-based filters, being the natural recursive version of IS to approach the optimal Bayesian solution. Recalling from the IS method, we have that the posterior distribution is characterized by the N_s generated points and its associated normalized weights, $\left\{\boldsymbol{x}_{0:k}^{(i)}, w_{0:k}^{(i)}\right\}_{i=1}^{N_s}$, as

$$\begin{aligned} p(\boldsymbol{x}_{0:k}|\boldsymbol{y}_{1:k}) &\approx \sum_{i=1}^{N_s} w_k^{(i)}\delta(\boldsymbol{x}_{0:k} - \boldsymbol{x}_{0:k}^{(i)}) \\ w_k^{(i)} &\propto \frac{p(\boldsymbol{x}_{0:k}^{(i)}|\boldsymbol{y}_{1:k})}{\pi(\boldsymbol{x}_{0:k}^{(i)}|\boldsymbol{y}_{1:k})} \ . \end{aligned} \tag{3.72}$$

Under the assumption that we have a discrete approximation of $p(\boldsymbol{x}_{0:k-1}|\boldsymbol{y}_{1:k-1})$, the aim of SIS is to obtain a set of particles that characterize the distribution at k when new

measurements are received, $\boldsymbol{y}_k$. If we choose the importance function to factorize as

$$\begin{aligned}\pi(\boldsymbol{x}_{0:k}|\boldsymbol{y}_{1:k}) &= \pi(\boldsymbol{x}_k|\boldsymbol{x}_{0:k-1},\boldsymbol{y}_{1:k})\pi(\boldsymbol{x}_{0:k-1}|\boldsymbol{y}_{1:k-1}) \\ &= \pi(\boldsymbol{x}_0)\prod_{t=1}^{k}\pi(\boldsymbol{x}_t|\boldsymbol{y}_{0:t-1},\boldsymbol{y}_{1:t}) \ , \end{aligned} \tag{3.73}$$

then the generation of samples can be done by augmenting the existing samples

$$\left\{\boldsymbol{x}_{0:k-1}^{(i)} \sim \pi(\boldsymbol{x}_{0:k-1}|\boldsymbol{y}_{1:k-1})\right\}_{i=1}^{N_s} \tag{3.74}$$

with the new state being

$$\left\{\boldsymbol{x}_{k}^{(i)} \sim \pi(\boldsymbol{x}_{k}|\boldsymbol{x}_{0:k-1},\boldsymbol{y}_{1:k})\right\}_{i=1}^{N_s} \ . \tag{3.75}$$

The associated weights are computed from the following posterior recursion:

$$\begin{aligned} p(\boldsymbol{x}_{0:k}|\boldsymbol{y}_{1:k}) &= \frac{p(\boldsymbol{y}_k|\boldsymbol{x}_{0:k},\boldsymbol{y}_{1:k-1})p(\boldsymbol{x}_{0:k}|\boldsymbol{y}_{1:k-1})}{p(\boldsymbol{y}_k|\boldsymbol{y}_{1:k-1})} \\ &= \frac{p(\boldsymbol{y}_k|\boldsymbol{x}_{0:k},\boldsymbol{y}_{1:k-1})p(\boldsymbol{x}_k|\boldsymbol{x}_{0:k-1},\boldsymbol{y}_{1:k-1})p(\boldsymbol{x}_{0:k-1}|\boldsymbol{y}_{1:k-1})}{p(\boldsymbol{y}_k|\boldsymbol{y}_{1:k-1})} \\ &= \frac{p(\boldsymbol{y}_k|\boldsymbol{x}_k)p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1})}{p(\boldsymbol{y}_k|\boldsymbol{y}_{1:k-1})}p(\boldsymbol{x}_{0:k-1}|\boldsymbol{y}_{1:k-1}) \\ &\propto p(\boldsymbol{y}_k|\boldsymbol{x}_k)p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1})p(\boldsymbol{x}_{0:k-1}|\boldsymbol{y}_{1:k-1}) \ , \end{aligned} \tag{3.76}$$

which only depends on the posterior at time $k-1$ and the likelihood and the prior at time k. Given that $p(\boldsymbol{x}_0) \triangleq p(\boldsymbol{x}_0|\boldsymbol{y}_0)$, where $\boldsymbol{y}_0$ is the set of no measurements, we can assume that the required density at time $k-1$ is available in the sequential approach. Substitution of (3.73) and (4.15) in (3.72) yields to the weight update equation,

$$w_k^{(i)} \propto w_{k-1}^{(i)}\frac{p(\boldsymbol{y}_k|\boldsymbol{x}_k^{(i)})p(\boldsymbol{x}_k^{(i)}|\boldsymbol{x}_{k-1}^{(i)})}{\pi(\boldsymbol{x}_k^{(i)}|\boldsymbol{x}_{0:k-1}^{(i)},\boldsymbol{y}_{1:k})} \ . \tag{3.77}$$

Now the recursion is closed and we are able to find an approximation of the filtering distribution given by

$$\hat{p}(\boldsymbol{x}_k|\boldsymbol{y}_{1:k}) = \sum_{i=1}^{N_s} w_k^{(i)}\delta(\boldsymbol{x}_k - \boldsymbol{x}_k^{(i)}) \ , \tag{3.78}$$

being $\delta(\cdot)$ the Kronecker's delta function. This approximation converges *a.s.* to the true posterior[7] as $N_s \to \infty$ under weak assumptions, according to the strong law of large

[7]Notice that *a.s.* convergence is equivalent to convergence *w.p.*1. Then, the convergence of the SIS estimate of the filtering distribution is expressed as

$$P\left\{\hat{p}(\boldsymbol{x}_k|\boldsymbol{y}_{1:k}) \to p(\boldsymbol{x}_k|\boldsymbol{y}_{1:k})\right\} = 1 \quad \text{as} \quad N_s \to \infty \ .$$

numbers [Dou98, Cri02]. These assumptions hold if the support of the chosen importance density ($\breve{\pi}$) include the support of the filtering distribution ($\breve{p}$), i.e.,

$$\begin{aligned} \breve{\pi} &= \{\boldsymbol{x}_k \in \mathbb{R}^{n_x} |\ \pi(\boldsymbol{x}_k|\boldsymbol{y}_{1:k}) > 0\} \\ \breve{p} &= \{\boldsymbol{x}_k \in \mathbb{R}^{n_x} |\ p(\boldsymbol{x}_k|\boldsymbol{y}_{1:k}) > 0\} \\ &\quad \text{and} \qquad \breve{p} \subseteq \breve{\pi}\ . \end{aligned} \tag{3.79}$$

Thus, for the convergence results to hold we have to ensure that the importance function has the same support as the true posterior, meaning that the set closure of the set of arguments of these functions for which the value is not zero is the same.

From the new set of sample points, one can compute a number of statistics. As conceptually presented in (3.11) and (3.12), the MMSE and MAP estimators can be obtained as

$$\hat{\boldsymbol{x}}_k^{\text{MMSE}} = \sum_{i=1}^{N_s} w_k^{(i)} \boldsymbol{x}_k^{(i)} \tag{3.80}$$

$$\hat{\boldsymbol{x}}_k^{\text{MAP}} = \arg\max_{\boldsymbol{x}_k^{(i)}} \left\{ w_k^{(i)} \right\}\ , \tag{3.81}$$

respectively. The covariance (or uncertainty region) of an estimate $\hat{\boldsymbol{x}}_k$ can be calculated as

$$\boldsymbol{\Sigma}_{\hat{x}} = \sum_{i=1}^{N_s} w_k^{(i)} \left(\boldsymbol{x}_k^{(i)} - \hat{\boldsymbol{x}}_k \right) \left(\boldsymbol{x}_k^{(i)} - \hat{\boldsymbol{x}}_k \right)^\top\ . \tag{3.82}$$

A pseudocode description of the SIS algorithm is shown in Algorithm 3.6. Notice that SIS is an algorithm that approximates the posterior by sequentially updating the measurement vector and propagating the importance weights. Basically, to sum up, it involves the approximation of the posterior by a set of N_s random samples taken from an *importance density function*, $\boldsymbol{x}_k^{(i)} \sim \pi(\boldsymbol{x}_k|\boldsymbol{x}_{0:k-1}^{(i)}, \boldsymbol{y}_{1:k})$, with associated importance weights $w_k^{(i)}$. The choice of $\pi(\cdot)$ is a critical issue in the design of any PF, which still remains as an open topic for statisticians [Dou00] and is usually an application-dependent issue. For a set of generated particles, $\left\{ \boldsymbol{x}_k^{(i)},\ w_k^{(i)} \right\}_{i=1}^{N_s}$, the approximation of the filtering distribution as given by PFs is obtained via equation (3.78).

3.D.3 Resampling

Everything has its payback, and PFs are not the exception. The main drawback of the SIS algorithm is that it suffers from the so-called *degeneracy phenomenon*, which states

Algorithm 3.6 Sequential Importance Sampling (SIS) algorithm

Require: $\left\{\boldsymbol{x}_{k-1}^{(i)},\ w_{k-1}^{(i)}\right\}_{i=1}^{N_s}$ and $\boldsymbol{y}_k$

Ensure: $\left\{\boldsymbol{x}_{k}^{(i)},\ w_{k}^{(i)}\right\}_{i=1}^{N_s}$

1: **for** $i = 1$ to N_s **do**

2: Generate $\boldsymbol{x}_k^{(i)} \sim \pi(\boldsymbol{x}_k|\boldsymbol{x}_{k-1}^{(i)}, \boldsymbol{y}_{1:k})$

3: Calculate $\tilde{w}_k^{(i)} = w_{k-1}^{(i)} \frac{p(\boldsymbol{y}_k|\boldsymbol{x}_k^{(i)})p(\boldsymbol{x}_k^{(i)}|\boldsymbol{x}_{k-1}^{(i)})}{\pi(\boldsymbol{x}_k^{(i)}|\boldsymbol{x}_{0:k-1}^{(i)},\boldsymbol{y}_{1:k})}$

4: **end for**

5: **for** $i = 1$ to N_s **do**

6: Normalize weights: $w_k^{(i)} = \frac{\tilde{w}_k^{(i)}}{\sum_{j=1}^{N_s}\tilde{w}_k^{j}}$

7: **end for**

that the variance of importance weights can only increase over time [Dou00]. In other words, after a certain number of sequential steps, one finds that the value of one of the normalized weights tends to 1 while the rest tend to 0. The problem then is to keep particle trajectories with *significant* weights and remove those which hardly contribute to the estimation of the filtering distribution. The solution was proposed in [Rub88] and is known as *resampling*: discarding samples with low importance weights and keep/multiply those with high importance weights. Resampling was proposed for SIS in a number of works [Gor93, Liu95, Ber97].

A measure of the degeneracy is the effective sample size N_{eff}, introduced in [Kon94], which is estimated as

$$\hat{N}_{eff} = \frac{1}{\sum_{i=1}^{N_s}\left(w_k^{(i)}\right)^2} \tag{3.83}$$

where $1 \leq \hat{N}_{eff} \leq N_s$ and values close to the lower bound indicate high degeneracy. The common approach is to specify a threshold such that, when the effective sample size is below, indicates the algorithm to apply resampling (see Algorithm 3.7). Although highly dependent on the chosen threshold, this approach is the most used in the literature. In [Ber99] a suggested threshold is $N_{th} = \frac{2}{3}N_s$.

The process of resampling particles can be implemented in several ways. Common schemes include multinomial resampling [Gor93], residual resampling [Liu98] and stratified/systematic resampling [Kit96, Cri01]. It is out of the scope of the dissertation the study of novel resampling strategies, in the sequel a multinomial resampling is considered which is one of the easiest to implement.

Although it solves the degeneracy phenomenon, resampling can result in a *sample impoverishment*, that is particles with high weights are selected many times. Notice that

Algorithm 3.7 Generic Particle Filtering with Resampling

Require: $\left\{\boldsymbol{x}_{k-1}^{(i)},\ w_{k-1}^{(i)}\right\}_{i=1}^{N_s}$ and $\boldsymbol{y}_k$

Ensure: $\left\{\boldsymbol{x}_{k}^{(i)},\ w_{k}^{(i)}\right\}_{i=1}^{N_s}$

1: $\left[\left\{\boldsymbol{x}_{k}^{(i)},\ w_{k}^{(i)}\right\}_{i=1}^{N_s}\right] = \mathrm{SIS}(\left\{\boldsymbol{x}_{k-1}^{(i)},\ w_{k-1}^{(i)}\right\}_{i=1}^{N_s},\ \boldsymbol{y}_k)$

2: Calculate $\hat{N}_{eff}$

3: **if** $\hat{N}_{eff} < N_{th}$ **then**

4: Resample Particles: $\left[\left\{\boldsymbol{x}_{k}^{(i)},\ w_{k}^{(i)}\right\}_{i=1}^{N_s}\right] = \mathrm{Resample}(\left\{\boldsymbol{x}_{k}^{(i)},\ w_{k}^{(i)}\right\}_{i=1}^{N_s})$

5: **end if**

this loss of diversity is severe when process noise is small [Gor93, Hig95]. Thus, convergence results of the algorithm should be re-established [Ber97]. Another undesired aspect of resampling is that it constitutes the bottleneck in any parallel implementation of PFs [Bol05], since all particles must be combined in this process.

Nevertheless, resampling is a key step in the design of a PF since degeneracy jeopardizes posterior estimates in a way that cannot be tolerated. Many work has been directed in the design of efficient resampling strategies and architectures [Bol04a].

3.D.4 Selection of the importance density

One of the key points of a PF algorithm is the choice of a *good* importance density function, $\pi(\boldsymbol{x}_k|\boldsymbol{x}_{0:k-1}^{(i)}, \boldsymbol{y}_{1:k})$. Actually, a bad choice can lead the algorithm to a poor characterization in (3.78) and, thus, to a bad performance. Some common alternatives might be found in the literature:

The optimal choice

Since the aim of a PF is to characterize the filtering distribution, it is quite intuitive to say that this distribution is the *best* choice for particle generation. Indeed, the optimal distribution is the target distribution that we wish to estimate:

$$\pi(\boldsymbol{x}_k|\boldsymbol{x}_{0:k-1}^{(i)}, \boldsymbol{y}_{1:k}) = p(\boldsymbol{x}_k|\boldsymbol{x}_{0:k-1}^{(i)}, \boldsymbol{y}_{1:k})\ , \tag{3.84}$$

which is optimal in the sense that it minimizes the variance of importance weights conditional upon the trajectory and the observations [Dou00]. The use of such an optimal

importance density reduces the degeneracy effect discussed previously. In this case, the calculation of weights reduces to:

$$w_k^{(i)} \propto w_{k-1}^{(i)} p(\boldsymbol{y}_k|\boldsymbol{x}_{k-1}^{(i)}) \ , \tag{3.85}$$

which does not depend on the current particle value $\boldsymbol{x}_k^{(i)}$, facilitating parallelization of the PF [Bol04a].

Thus, in order to use the optimal importance density, one has to *i)* be able to draw samples from $p(\boldsymbol{x}_k|\boldsymbol{x}_{0:k-1}^{(i)}, \boldsymbol{y}_{1:k})$ and to *ii)* evaluate

$$p(\boldsymbol{y}_k|\boldsymbol{x}_{k-1}) = \int p(\boldsymbol{y}_k|\boldsymbol{x}_k) p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1}) d\boldsymbol{x}_k \ . \tag{3.86}$$

Few special cases can be found that allow the generation/evaluation from these distributions. One case is when the states $\boldsymbol{x}_k$ are finite, as in [Dou01b]. Another case is when the DSS model is Gaussian with linear measurements [Dou00, Ris04]. Unfortunately, these two requirements may not hold in general and one has to resort to suboptimal choices.

Suboptimal choices

Since the only condition imposed on $\pi(\boldsymbol{x}_k|\boldsymbol{x}_{0:k-1}^{(i)}, \boldsymbol{y}_{1:k})$ is to fulfill (3.73), the amount of possible suboptimal importance densities seems huge[8]. Indeed, a general methodology for selecting an importance density is still missing, being several alternatives proposed in the literature. Actually, in most of the cases this choice highly depends on the application itself and has to be carefully designed.

The simplest approach, and the most popular, is to consider the transitional prior as the importance function. In this case, weights are proportional to the likelihood function:

$$\begin{aligned} \pi(\boldsymbol{x}_k|\boldsymbol{x}_{0:k-1}^{(i)}, \boldsymbol{y}_{1:k}) &= p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1}^{(i)}) \\ w_k^{(i)} &\propto w_{k-1}^{(i)} p(\boldsymbol{y}_k|\boldsymbol{x}_k^{(i)}) \ . \end{aligned} \tag{3.87}$$

The popularity of the transitional prior choice is due to its simple implementation and the lack of computationally intensive calculations. Note that in the Gaussian case, generation in (3.87) reduces to sample from a Gaussian distribution. Nevertheless, this choice was shown to be inefficient as it requires a large number of samples to effectively characterize the posterior distribution [Ris04]. The main reason is that, since measurements are not taken into consideration when generating particles, the algorithm is likely to exhibit a high degeneracy.

[8] Keeping in mind that the importance density has to share the same support as the target distribution, in order to claim convergence [Cri02].

More sophisticated, though also suboptimal, alternatives aim at approximating the optimal importance density in (3.84). Although they require a higher computational burden, their performance is typically better than the one provided by the transitional prior. Some of the strategies reported in the literature include local model linearization [Dou00], Gaussian approximations [Kot03a] or the use of the unscented transform [Mer00]. In Section 3.3.4 of this dissertation, we consider another importance density function. The method was originally proposed in [Cev07] for an acoustic multitarget tracking problem. Basically it is based on a Gaussian approximation of the optimal distribution.

4

Sequential estimation of neural activity from voltage traces in single-neuron models

THE membrane potential, obtained from intracellular recordings, is one of the most valuable signals of neurons' activity. Most of the neuron models have been derived from fine measurements and allow the progress of "in silico" experiments. However, other interesting quantities informing about the neuron's intrinsic activity or synaptic connections [Piw04, Béd11] are either costly to obtain (channel blocking and clamping techniques [Hod52, Bre12]) or impossible to measure explicitly with today's techniques. Thus, estimation methods can be very useful, mostly those that can be applied to obtain time evolution on-line; that is, avoiding the need of repetitions that could be contaminated by neuronal variability. In this chapter, as a first step, we concentrate on methods to extract intrinsic activity of ionic channels, namely the probabilities of opening and closing channels. To this purpose, we consider a neuron model and, using a PF algorithm with optimal importance density, we recover both the membrane potential and the activity of the potassium channel with the minimum attainable error. Moreover, we design an enhanced algorithm that can cope with model uncertainties and thus relaxing the assumption of perfect model knowledge of the former algorithm.

In the open literature similar solutions could be found. [Kob11] deals with a similar inference problem, under correct model assumptions for the ionic channels [Pos08], and solve the filtering problem by a Kalman filter. The authors assume synaptic inputs as white noise and estimate the mean and variances of the process.

Another similar setup is treated in [Pan12], where a PF is used to filter voltage and synaptic conductances. In this case, the voltage observations are assumed subthreshold, as opposite to our approach in this Thesis. A similar solution, this time based on the Gaussian mixture Kalman filtering, was proposed in [Lan13]. More recently, the same authors proposed in [Lan14] a Kalman-like solution to estimate voltage and gating variables, as well as the maximal conductances of the ionic channels (thus implicitly assuming the latter as time-varying). Our approach is similar in essence, although we go further in the method and present a solution that can cope with the estimation of any time-varying signal in the system (gathered in the state vector) and those deterministic parameters that describe the model (gathered in an unknown parameter vector).

In [Rud03, Béd11], it is shown how to extract synaptic conductances from noisy voltage traces. The work considers subthreshold activity. This approach might introduce errors in situations where spiking is present or when there is sub-threshold ionic activity [Gui06]. The approach we follow in this Thesis is to work in the spike regime, without the sub-threshold assumption [Clo13b, Clo13a].

The remainder of the chapter is organized as follows. The problem is stated in discrete state-space in Section 4.1, for which we focused in the model named after Morris and Lecar introduced in Section 2.3.1. The main causes of model inaccuracies are enumerated in Section 4.2, as well as hints to account for them in the method. We present a PF solution for the filtering problem under perfect model knowledge in Section 4.3 and design an enhanced algorithm that can estimate the parameters of the model in Section 4.4. Section 4.5 discusses computer simulations, with comparison to the theoretical estimation bounds derived in Appendix 4.A, and Section 4.6 wraps up the main conclusions of the study.

4.1 Problem statement

As introduced earlier in Chapter 2, the problem investigated in this Thesis considers recordings of noisy voltage traces to infer the hidden gating variables of the neuron model, as well as filtered voltage estimates. Data is recorded at discrete time-instants k at a sampling frequency $f_s = 1/\mathrm{T}_s$. The problem can thus be posed in the form of a discrete-time, state-space model. The observations are

$$y_k \sim \mathcal{N}(v_k, \sigma^2_{y,k}) \ , \tag{4.1}$$

with $\sigma^2_{y,k}$ modeling the noise variance due to the sensor or the instrumentation inaccuracies when performing the experiment. To provide comparable results, we define the signal-to-noise ratio (SNR) as $\text{SNR} = P_s/P_n$, with P_s being the average signal power and $P_n = \sigma^2_{y,k}$ the noise power.

On the other hand, we have models for the evolution of the voltage-traces and the hidden variables of a neuron. For instance, the Morris-Lecar model presented in Section 2.3.1. The unknown state vector in this case is

$$\boldsymbol{x}_k = \begin{pmatrix} v_k \\ n_k \end{pmatrix} . \tag{4.2}$$

Notice that the Morris-Lecar neuron model is defined by a system of continuous-time, ODEs. In general, all the mathematical models for neurons are of this type. However, we are interested in expressing the model in the form of a discrete state-space,

$$\boldsymbol{x}_k = f_k(\boldsymbol{x}_{k-1}) + \nu_k , \tag{4.3}$$

where $\nu_k \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}_{x,k})$ is the process noise which includes the model inaccuracies. The covariance matrix $\boldsymbol{\Sigma}_{x,k}$ is used to quantify our confidence in the model that maps $f_k : \{v_{k-1}, n_{k-1}\} \mapsto \{v_k, n_k\}$. In general, obtaining a closed-form analytical expression for f_k without approximations is not possible. An alternative option are numerical methods, which aim at solving the differential equation at discrete instants. One simple, yet popular, method to approximate $\dot{\boldsymbol{x}} = f(\boldsymbol{x})$ is the Euler method (a finite difference approximation) that is based on the approximation

$$\dot{\boldsymbol{x}} \doteq \frac{d\boldsymbol{x}}{dt} \approx \frac{\Delta \boldsymbol{x}}{\Delta t} = f(\boldsymbol{x}) \tag{4.4}$$

where $\Delta t = \mathrm{T}_s$ is the sampling period and thus,

$$\frac{\Delta \boldsymbol{x}}{\Delta t} = \frac{\boldsymbol{x}(t + \mathrm{T}_s) - \boldsymbol{x}(t)}{\mathrm{T}_s} = f(\boldsymbol{x}(t)) \tag{4.5}$$

or equivalently in discrete notation

$$\dot{\boldsymbol{x}} \approx \frac{\boldsymbol{x}_k - \boldsymbol{x}_{k-1}}{\mathrm{T}_s} = f(\boldsymbol{x}_{k-1}) , \tag{4.6}$$

from which it follows that

$$\boldsymbol{x}_k = \boldsymbol{x}_{k-1} + \mathrm{T}_s f(\boldsymbol{x}_{k-1}) , \tag{4.7}$$

which is of the Markovian type in (3.1).

If we focus on the Morris-Lecar model, the resulting discrete version of the ODE system in (2.8)-(2.9) is:

$$\begin{aligned} v_k &= v_{k-1} - \frac{\mathrm{T}_s}{C_m}\Big(\bar{g}_{\mathrm{L}}(v_{k-1} - E_{\mathrm{L}}) \\ &+ \bar{g}_{\mathrm{Ca}} m_\infty(v_{k-1})(v_{k-1} - E_{\mathrm{Ca}}) + \bar{g}_{\mathrm{K}} n_{k-1}(v_{k-1} - E_{\mathrm{K}}) - I_{\mathrm{app}}\Big) \qquad (4.8) \\ n_k &= n_{k-1} + \mathrm{T}_s \phi \frac{n_\infty(v_{k-1}) - n_{k-1}}{\tau_n(v_{k-1})} \ , \qquad (4.9) \end{aligned}$$

with $m_\infty(v_k)$, $n_\infty(v_k)$, and $\tau_n(v_k)$ defined as in (2.13)-(2.15). Then, (4.8) and (4.9) can be interpreted as $\boldsymbol{x}_k = f_k(\boldsymbol{x}_{k-1})$.

The goal is to express the inference problem in state-space formulation and apply the tools learned in Chapter 3. The final ingredient to do so is to introduce the so-called process noise in the state equation

$$\boldsymbol{x}_k = f_k(\boldsymbol{x}_{k-1}) + \begin{pmatrix} \nu_{v,k} \\ \nu_{n,k} \end{pmatrix} \ , \qquad (4.10)$$

where the noise terms $\nu_{v,k}$ and $\nu_{n,k}$ are jointly Gaussian with covariance matrix $\boldsymbol{\Sigma}_{x,k}$. Further details of this matrix are discussed in Section 4.2.

4.2 Model inaccuracies

The proposed estimation method highly relies on the fact that the neuron model is known. This is true to some extend, but most of the parameters in the model discussed in Section 2.3.1 are to be estimated beforehand. Therefore, the robustness of the method to possible inaccuracies should be assessed. In this section, we point out possible causes of missmodeling. In next section, computer simulations are used to characterize the performance of the method under these impairments.

In the single-neuron model considered, three major sources of inaccuracies can be identified:

1. The applied current I_{app} can be itself noisy, with a variance depending on the quality of the instrumentation used and the experiment itself. We model the actual applied current as the random variable

$$I_{\mathrm{app}} = I_o + \nu_{I,k} \ , \ \nu_{I,k} \sim \mathcal{N}(0, \sigma_I^2) \ , \qquad (4.11)$$

where I_o is the nominal current applied and σ_I^2 the variance around this value. Plugging (4.11) into (4.8) we obtain that the contribution of I_{app} to the noise term is $\frac{\mathrm{T}_s}{C_m}\nu_{I,k} \sim \mathcal{N}(0, (\mathrm{T}_s/C_m)^2\sigma_I^2)$.

2. The conductance of the leakage term has to be estimated beforehand. In general, this term is considered constant although it gathers relatively distinct phenomena that can potentially be time-varying. The maximal conductance of the leakage term is therefore inaccurate and modeled as

$$\bar{g}_L = \bar{g}_L^o + \nu_{g,k} \ , \ \nu_{g,k} \sim \mathcal{N}(0, \sigma_g^2) \ , \tag{4.12}$$

where $\bar{g}_L^o$ is the nominal, estimated conductance and σ_g^2 the variance of this estimate. Similarly, inserting (4.12) into (4.8) we see that the contribution of $\bar{g}_L$ to the noise term is $\frac{\mathrm{T}_s}{C_m}\nu_{g,k} \sim \mathcal{N}(0, (\mathrm{T}_s/C_m)^2(v_{k-1} - E_\mathrm{L})\sigma_g^2)$.

3. The parameters in $m_\infty(v_k)$, $n_\infty(v_k)$, and $\tau_n(v_k)$ are to be estimated. In general, these parameters are properly obtained by standard methods [Izh06]. However, as they are estimates, a residual error typically remains. To account for these inaccuracies, we consider that the equation governing the evolution of gating variables is corrupted by a zero-mean additive white Gaussian process with variance σ_n^2.

At the end of the day, we came up with a way of constructing the model covariance matrix, as the contribution of the aforementioned inaccuracies. In a practical setup, in order to compute the noise variance due to leakage, we need to use the approximation $\hat{v}_{k-1} \approx v_{k-1}$, where $\hat{v}_{k-1}$ is computed by the filter. We construct the covariance matrix of the model as

$$\mathbf{\Sigma}_{x,k} = \begin{pmatrix} \sigma_v^2 & 0 \\ 0 & \sigma_n^2 \end{pmatrix} \ , \tag{4.13}$$

where we used that the overall noise in the voltage model is $\frac{\mathrm{T}_s}{C_m}(\nu_{I,k} - \nu_{g,k}) \sim \mathcal{N}(0, \sigma_v^2)$ and

$$\sigma_v^2 = \left(\frac{\mathrm{T}_s}{C_m}\right)^2 \left(\sigma_I^2 + (\hat{v}_{k-1} - E_\mathrm{L})^2\sigma_g^2\right) \tag{4.14}$$

could be an estimate of σ_v^2, provided accurate knowledge of σ_I^2 and σ_g^2. Otherwise, the covariance matrix of the process has to be estimated by other means, as the ones presented in this Chapter for mixed state-parameter estimation in nonlinear filtering problems.

4.3 Sequential estimation of voltage traces and gating variables by particle filtering

Bayesian filtering involves the recursive estimation of states $\boldsymbol{x}_k \in \mathbb{R}^{n_x}$ given measurements $y_k \in \mathbb{R}$ at time t based on all available measurements, $y_{1:k} = \{y_1, \ldots, y_k\}$. To that aim, we

are interested in the filtering distribution $p(\boldsymbol{x}_k|y_{1:k})$, which can be recursively expressed as

$$p(\boldsymbol{x}_k|y_{1:k}) = \frac{p(y_k|\boldsymbol{x}_k)p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1})}{p(y_k|y_{1:k-1})} p(\boldsymbol{x}_{k-1}|y_{1:k-1}) \ , \tag{4.15}$$

with $p(y_k|\boldsymbol{x}_k)$ and $p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1})$ referred to as the likelihood and the prior distributions, respectively. Unfortunately, (4.15) can only be obtained in closed-form in some special cases and in more general setups we should resort to more sophisticated methods. In this paper we consider PF to overcome the nonlinearity of the model [Dju03].

As explained in Section 3.3.4, PFs approximate the filtering distribution by a set of N weighted random samples, forming the random measure $\left\{\boldsymbol{x}_k^{(i)}, w_k^{(i)}\right\}_{i=1}^{N}$. These random samples are drawn from the importance density distribution, $\pi(\cdot)$,

$$\boldsymbol{x}_k^{(i)} \sim \pi(\boldsymbol{x}_k|\boldsymbol{x}_{0:k-1}^{(i)}, y_{1:k}) \tag{4.16}$$

and weighted according to the general formulation

$$w_k^{(i)} \propto w_{k-1}^{(i)} \frac{p(y_k|\boldsymbol{x}_{0:k}^{(i)}, y_{1:k-1}) p(\boldsymbol{x}_k^{(i)}|\boldsymbol{x}_{k-1}^{(i)})}{\pi(\boldsymbol{x}_k^{(i)}|\boldsymbol{x}_{0:k-1}^{(i)}, y_{1:k})} \ . \tag{4.17}$$

The importance density from which particles are drawn is a key issue in designing efficient PFs. It is well-known that the optimal importance density is $\pi(\boldsymbol{x}_k|\boldsymbol{x}_{0:k-1}^{(i)}, y_{1:k}) = p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1}^{(i)}, y_k)$, in the sense that it minimizes the variance of importance weights. Weights are computed using (4.17) as $w_k^{(i)} \propto w_{k-1}^{(i)} p(y_k|\boldsymbol{x}_{k-1}^{(i)})$. This choice requires the ability to draw from $p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1}^{(i)}, y_k)$ and to evaluate $p(y_k|\boldsymbol{x}_{k-1}^{(i)})$. In general, the two requirements cannot be met and one needs to resort to suboptimal choices. However, we are able to use the optimal importance density since the state-space model assumed here is Gaussian, with nonlinear process equations but related linearly to observations [Dou00]. The equations are:

$$p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1}^{(i)}, y_k) = \mathcal{N}(\boldsymbol{\mu}_{\pi,k}^{(i)}, \boldsymbol{\Sigma}_{\pi,k}) \tag{4.18}$$

with

$$\boldsymbol{\mu}_{\pi,k}^{(i)} = \boldsymbol{\Sigma}_{\pi,k} \left(\boldsymbol{\Sigma}_{x,k}^{-1} f_k(\boldsymbol{x}_{k-1}^{(i)}) + \frac{y_k}{\sigma_{y,k}^2} \right) \tag{4.19}$$

$$\boldsymbol{\Sigma}_{\pi,k} = \left(\boldsymbol{\Sigma}_{x,k}^{-1} + \sigma_{y,k}^{-2} \mathbf{I} \right)^{-1} \ , \tag{4.20}$$

and the importance weights can be updated using

$$p(y_k|\boldsymbol{x}_{k-1}^{(i)}) = \mathcal{N}(\mathbf{h} f_k(\boldsymbol{x}_{k-1}^{(i)}), \mathbf{h}\boldsymbol{\Sigma}_{x,k}\mathbf{h}^\top + \sigma_{y,k}^2) \ , \tag{4.21}$$

with $\mathbf{h} = (1, 0)$. The PF provides a discrete approximation of the filtering distribution of the form $p(\boldsymbol{x}_k|y_{1:k}) \approx \sum_{i=1}^{N} w_k^{(i)} \delta(\boldsymbol{x}_k - \boldsymbol{x}_k^{(i)})$, which gather all information from $\boldsymbol{x}_k$ that the measurements up to time t provide. This was discussed when introducing (3.78). For instance, the minimum mean square error estimator can be obtained as

$$\hat{\boldsymbol{x}}_k = \sum_{i=1}^{N} w_k^{(i)} \boldsymbol{x}_k^{(i)} \ , \tag{4.22}$$

where $\hat{\boldsymbol{x}}_k = (\hat{v}_k, \hat{n}_k)^\top$. Recall that the method discussed in this section could be easily adapted to other neuron models simply substituting by the corresponding transition function f_k and constructing the state vector $\boldsymbol{x}_k$ conveniently.

As a final step, PFs incorporate a resampling strategy to avoid collapse of particles into a single state point. As introduced earlier in Section 3.3.4, resampling consists in eliminating particles with low importance weights and replicating those in high-probability regions [Dou05]. The overall algorithm can be consulted in Algorithm 4.1 at instance k. Notice that this version of the algorithm requires knowledge of noise statistics and all the model parameters

$$\boldsymbol{\Theta} = (\bar{g}_{\mathrm{L}}, E_{\mathrm{L}}, \bar{g}_{\mathrm{Ca}}, E_{\mathrm{Ca}}, \bar{g}_{\mathrm{K}}, E_{\mathrm{K}}, \phi, V_1, V_2, V_3, V_4)^\top \ . \tag{4.23}$$

Algorithm 4.1 Particle filtering with optimal importance density

Require: $\boldsymbol{\Sigma}_{x,k}$, $\sigma_{y,k}^2$, $\boldsymbol{\Theta}$, $\left\{\boldsymbol{x}_{k-1}^{(i)}, w_{k-1}^{(i)}\right\}_{i=1}^{N}$ and y_k

Ensure: $\left\{\boldsymbol{x}_k^{(i)}, w_k^{(i)}\right\}_{i=1}^{N}$ and $\hat{\boldsymbol{x}}_k$

1: Calculate $\boldsymbol{\Sigma}_{\pi,k} = \left(\boldsymbol{\Sigma}_{x,k}^{-1} + \sigma_{y,k}^{-2}\mathbf{I}\right)^{-1}$
2: **for** $i = 1$ to N **do**
3: Calculate $\boldsymbol{\mu}_{\pi,k}^{(i)} = \boldsymbol{\Sigma}_{\pi,k}\left(\boldsymbol{\Sigma}_{x,k}^{-1} f_k(\boldsymbol{x}_{k-1}^{(i)}) + \frac{y_k}{\sigma_{y,k}^2}\right)$
4: Generate $\boldsymbol{x}_k^{(i)} \sim \mathcal{N}(\boldsymbol{\mu}_{\pi,k}^{(i)}, \boldsymbol{\Sigma}_{\pi,k})$
5: Calculate $\tilde{w}_k^{(i)} = w_{k-1}^{(i)} \frac{p(y_k|\boldsymbol{x}_{0:k}^{(i)}, y_{1:k-1}) p(\boldsymbol{x}_k^{(i)}|\boldsymbol{x}_{k-1}^{(i)})}{\mathcal{N}(\boldsymbol{\mu}_{\pi,k}^{(i)}, \boldsymbol{\Sigma}_{\pi,k})}$
6: **end for**
7: **for** $i = 1$ to N **do**
8: Normalize weights: $w_k^{(i)} = \frac{\tilde{w}_k^{(i)}}{\sum_{j=1}^{N} \tilde{w}_k^{(j)}}$
9: **end for**
10: MMSE state estimation: $\hat{\boldsymbol{x}}_k = \sum_{i=1}^{N} w_k^{(i)} \boldsymbol{x}_k^{(i)}$
11: $\left\{\boldsymbol{x}_k^{(i)}, 1/N\right\}_{i=1}^{N} = \text{Resample}(\left\{\boldsymbol{x}_k^{(i)}, w_k^{(i)}\right\}_{i=1}^{N})$

4.4 Joint estimation of states and model parameters by particle filtering

In practice the parameters in (4.23) might not be known. It is reasonable to assume that $\boldsymbol{\Theta}$, or a subset of these parameters $\boldsymbol{\theta} \subseteq \boldsymbol{\Theta}$, are unknown and need to be estimated at the same time the filtering method in Algorithm 4.1 is executed. Therefore, the ultimate goal in this case is to estimate jointly the time evolving states and the unknown parameters of the model, $\boldsymbol{x}$ and $\boldsymbol{\theta}$ respectively.

We follow the approach in [Sär13] to present one of the possible methodologies to enhance the PF presented in Algorithm 4.1 with parameter estimation capabilities. Joint estimation of states and parameters is a longstanding problem in Bayesian filtering, and specially hard to handle in the context of PFs. Refer to [And04, And05, Poy11] and the references for a complete survey.

Following the Bayesian philosophy adopted in this Thesis, the problem fundamentally reduces to assigning an a priori distribution for the unknown parameter $\boldsymbol{\theta} \in \mathbb{R}^{n_\theta}$ and extending the state-space model in (3.3)-(3.4) with

$$\begin{aligned} \boldsymbol{\theta} &\sim p(\boldsymbol{\theta}) && (4.24)\\ \boldsymbol{x}_k &\sim p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1}, \boldsymbol{\theta}) \quad \text{for } k \geq 1 && (4.25)\\ \boldsymbol{y}_k &\sim p(\boldsymbol{y}_k|\boldsymbol{x}_k, \boldsymbol{\theta}) \quad \text{for } k \geq 1 && (4.26) \end{aligned}$$

and initial state distribution $\boldsymbol{x}_0 \sim p(\boldsymbol{x}_0|\boldsymbol{\theta})$. Applying Bayes rule, the full posterior distribution can be expressed as

$$p(\boldsymbol{x}_{0:T}, \boldsymbol{\theta}|\boldsymbol{y}_{1:T}) = \frac{p(\boldsymbol{y}_{1:T}|\boldsymbol{x}_{0:T}, \boldsymbol{\theta})p(\boldsymbol{x}_{0:T}|\boldsymbol{\theta})p(\boldsymbol{\theta})}{p(\boldsymbol{y}_{1:T})} \tag{4.27}$$

with

$$\begin{aligned} p(\boldsymbol{y}_{1:T}|\boldsymbol{x}_{0:T}, \boldsymbol{\theta}) &= \prod_{k=1}^{T} p(\boldsymbol{y}_k|\boldsymbol{x}_k, \boldsymbol{\theta}) && (4.28)\\ p(\boldsymbol{x}_{0:T}|\boldsymbol{\theta}) &= p(\boldsymbol{x}_0|\boldsymbol{\theta}) \prod_{k=1}^{T} p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1}, \boldsymbol{\theta}) \ . && (4.29) \end{aligned}$$

Notice here that we are dealing with a finite horizon of observations T. Then, from a Bayesian perspective, the estimation of $\boldsymbol{\theta}$ is equivalent to obtaining its marginal posterior distribution

$$p(\boldsymbol{\theta}|\boldsymbol{y}_{1:T}) = \int p(\boldsymbol{x}_{0:T}, \boldsymbol{\theta}|\boldsymbol{y}_{1:T}) d\boldsymbol{x}_{0:T} \ ; \tag{4.30}$$

however, this is in general extremely hard to compute analytically and one needs to find workarounds. Evaluation of the full posterior turns to be not only computationally prohibitive, but useless if states cannot be marginalized out analytically. Alternative methods resort on the factorization of the parameter marginal distribution as

$$p(\boldsymbol{\theta}|\boldsymbol{y}_{1:T}) = p(\boldsymbol{y}_{1:T}|\boldsymbol{\theta})p(\boldsymbol{\theta}) \tag{4.31}$$

and how Bayesian filters can be transformed to provide characterizations of the marginal likelihood distribution $p(\boldsymbol{y}_{1:T}|\boldsymbol{\theta})$ and related quantities. The marginal likelihood distribution can be recursively factorized in terms of the predictive distributions of the observations:

$$p(\boldsymbol{y}_{1:T}|\boldsymbol{\theta}) = \prod_{k=1}^{T} p(\boldsymbol{y}_k|\boldsymbol{y}_{1:k-1}, \boldsymbol{\theta}) \ , \tag{4.32}$$

with

$$\begin{aligned} p(\boldsymbol{y}_k|\boldsymbol{y}_{1:k-1}, \boldsymbol{\theta}) &= \mathbb{E}_{x_k|y_{1:k-1},\theta}\Big\{p(\boldsymbol{y}_k|\boldsymbol{x}_k, \boldsymbol{\theta})\Big\} & (4.33)\\ &= \int p(\boldsymbol{y}_k|\boldsymbol{x}_k, \boldsymbol{\theta})p(\boldsymbol{x}_k|\boldsymbol{y}_{1:k-1}, \boldsymbol{\theta})d\boldsymbol{x}_k & (4.34) \end{aligned}$$

obtained straightforwardly as a byproduct of any of the Bayesian filtering methods discussed earlier in Section 3.3.

A useful transformation of the marginal likelihood is the so-called *energy function*, which is sometimes more convenient when implementing the solution. The energy function is defined as

$$\begin{aligned} \varphi_T(\boldsymbol{\theta}) &= -\ln p(\boldsymbol{y}_{1:T}|\boldsymbol{\theta}) & (4.35)\\ &= -\ln p(\boldsymbol{y}_{1:T}|\boldsymbol{\theta}) - \ln p(\boldsymbol{\theta}) & (4.36) \end{aligned}$$

or, equivalently, $p(\boldsymbol{y}_{1:T}|\boldsymbol{\theta}) \propto \exp(-\varphi_T(\boldsymbol{\theta}))$. The energy function can then be recursively computed as a function of the predictive distribution

$$\begin{aligned} \varphi_0(\boldsymbol{\theta}) &= -\ln p(\boldsymbol{\theta}) & (4.37)\\ \varphi_k(\boldsymbol{\theta}) &= \varphi_{k-1}(\boldsymbol{\theta}) - \ln p(\boldsymbol{y}_k|\boldsymbol{y}_{1:k-1}, \boldsymbol{\theta}) \quad \text{for } k \geq 1 \ . & (4.38) \end{aligned}$$

Then, the basic problem is to obtain an estimate of the predictive distribution $p(\boldsymbol{y}_k|\boldsymbol{y}_{1:k-1}, \boldsymbol{\theta})$ from the PF we have designed in Section 4.3 and use it in conjunction with $p(\boldsymbol{\theta})$ to infer the marginal distribution $p(\boldsymbol{\theta}|\boldsymbol{y}_{1:T})$ of interest. This latter step can be performed in several ways, from which we choose to use the Markov-Chain Monte-Carlo (MCMC) methodology to continue with a fully Bayesian solution. Besides, it is known to be the solution that provides best results when used in a PF. Next, we detail how $\varphi_k(\boldsymbol{\theta})$ can be obtained from a PF algorithm, present the MCMC method for parameter estimation, and finally we sketch the overall algorithm.

Computing the energy function from particle filters:

The modification needed is very small. Actually, it is non-invasive in the sense that the algorithm remains the same and the energy function can be computed adding some extra equations. Recall that the predictive distribution is

$$p(\boldsymbol{y}_k|\boldsymbol{y}_{1:k-1},\boldsymbol{\theta}) = \int p(\boldsymbol{y}_k|\boldsymbol{x}_k,\boldsymbol{\theta})p(\boldsymbol{x}_k|\boldsymbol{y}_{1:k-1},\boldsymbol{\theta}) \tag{4.39}$$

and that the PF provides characterizations of the two distributions in the integral. Then, one can get an approximation by

$$\begin{aligned} p(\boldsymbol{y}_k|\boldsymbol{y}_{1:k-1},\boldsymbol{\theta}) &\approx \hat{p}(\boldsymbol{y}_k|\boldsymbol{y}_{1:k-1},\boldsymbol{\theta}) && (4.40)\\ &= \sum_{i=1}^{N} w_{k-1}^{(i)}\zeta_k^{(i)} && (4.41)\end{aligned}$$

with $w_{k-1}^{(i)}$ as in the original algorithm and

$$\zeta_k^{(i)} = \frac{p(\boldsymbol{y}_k|\boldsymbol{x}_k^{(i)},\boldsymbol{\theta})p(\boldsymbol{x}_k^{(i)}|\boldsymbol{x}_{k-1}^{(i)},\boldsymbol{\theta})}{\pi(\boldsymbol{x}_k^{(i)}|\boldsymbol{x}_{0:k-1}^{(i)},\boldsymbol{y}_{1:k},\boldsymbol{\theta})} \ . \tag{4.42}$$

Then, it is straightforward to identify the energy function approximation as

$$\begin{aligned} \varphi_T(\boldsymbol{\theta}) &= -\ln p(\boldsymbol{\theta}) - \ln p(\boldsymbol{y}_{1:T}|\boldsymbol{\theta}) && (4.43)\\ &= -\ln p(\boldsymbol{\theta}) - \ln \prod_{k=1}^{T} p(\boldsymbol{y}_k|\boldsymbol{y}_{1:k-1},\boldsymbol{\theta}) && (4.44)\\ &= -\ln p(\boldsymbol{\theta}) - \sum_{k=1}^{T} \ln p(\boldsymbol{y}_k|\boldsymbol{y}_{1:k-1},\boldsymbol{\theta}) && (4.45)\\ &\approx -\ln p(\boldsymbol{\theta}) - \sum_{k=1}^{T} \ln \hat{p}(\boldsymbol{y}_k|\boldsymbol{y}_{1:k-1},\boldsymbol{\theta}) && (4.46)\\ &= -\ln p(\boldsymbol{\theta}) - \sum_{k=1}^{T} \ln \sum_{i=1}^{N} w_{k-1}^{(i)}\zeta_k^{(i)} = \hat{\varphi}_T(\boldsymbol{\theta}) \ , && (4.47)\end{aligned}$$

which can be computed recursively in the PF algorithm.

The Particle Markov-Chain Monte-Carlo algorithm:

Once an approximation of the energy function is available, we can apply MCMC to infer the marginal distribution of the parameters. MCMC methods constitute a general methodology to generate samples recursively from a given distribution by randomly simulating

from a Markov chain [Gil96, Ber97, Liu08, Bro11]. There are many algorithms implementing the MCMC concept, being one of the most popular the Metropolis-Hastings (MH) algorithm. At the j-th iteration, the MH algorithm samples a candidate point $\boldsymbol{\theta}^*$ from a proposal distribution $q(\boldsymbol{\theta}^*|\boldsymbol{\theta}^{(j-1)})$ based on the previous sample $\boldsymbol{\theta}^{(j-1)}$. Starting from an arbitrary value $\boldsymbol{\theta}^{(0)}$, the MH algorithm accepts the new candidate point (meaning that it was generated from the target distribution, $p(\boldsymbol{\theta}|\boldsymbol{y}_{1:T})$) using the rule

$$\boldsymbol{\theta}^{(j)} = \begin{cases} \boldsymbol{\theta}^*, & \text{if } u \leq \alpha^{(j)} \\ \boldsymbol{\theta}^{(j)}, & \text{otherwise} \end{cases} \tag{4.48}$$

where u is drawn randomly from a uniform distribution, $u \sim \mathcal{U}(0,1)$, and

$$\alpha^{(j)} = \min\left\{1, \exp(\varphi_T(\boldsymbol{\theta}^{(j-1)}) - \varphi_T(\boldsymbol{\theta}^*))\frac{q(\boldsymbol{\theta}^{(j-1)}|\boldsymbol{\theta}^*)}{q(\boldsymbol{\theta}^*|\boldsymbol{\theta}^{(j-1)})}\right\} \tag{4.49}$$

is referred to as the acceptance probability.

It is critical for the performance of the algorithm the choice of the proposal density. A common choice is

$$q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(j-1)}) = \mathcal{N}(\boldsymbol{\theta}; \boldsymbol{\theta}^{(j-1)}, \boldsymbol{\Sigma}^{(j-1)}) \tag{4.50}$$

with the selection of the transitional covariance remaining as the tuning $\boldsymbol{\Sigma}^{(j-1)}$ parameter. This covariance can be adapted as iterations of the MCMC method progress. In this Thesis we have adopted the Robust Adaptive Metropolis (RAM) algorithm [Vih12]. The RAM algorithm can be consulted in Algorithm 4.2. We use the notation that $\mathbf{S} = \text{Chol}\,(\mathbf{A})$ denotes the Cholesky factorization of an Hermitian positive-definite matrix $\mathbf{A}$ such that $\mathbf{A} = \mathbf{S}\mathbf{S}^\top$, and $\mathbf{S}$ is a lower triangular matrix [Gol96]. The RAM algorithm outputs a set of samples $\left\{\boldsymbol{\theta}^{(j)}\right\}_{j=1}^{M}$, where M is the number of iteration of the MCMC procedure. This samples are originated from the target distribution

$$\left\{\boldsymbol{\theta}^{(j)} \sim p(\boldsymbol{\theta}|\boldsymbol{y}_{1:T})\right\}_{j=1}^{M} , \tag{4.51}$$

which can be used to approximate (after neglecting the first samples corresponding to the transient phase of the algorithm) it as

$$p(\boldsymbol{\theta}|\boldsymbol{y}_{1:T}) \approx \frac{1}{M}\sum_{j=1}^{M}\delta(\boldsymbol{\theta} - \boldsymbol{\theta}^{(j)}) , \tag{4.52}$$

and one can obtain the desired statistics from the characterization of the marginal distribution. For instance, point estimates of the parameter like these 2 options

$$\hat{\boldsymbol{\theta}}^{\text{MMSE}} = \frac{1}{M}\sum_{j=1}^{M}\boldsymbol{\theta}^{(j)} \qquad \text{or} \qquad \hat{\boldsymbol{\theta}} = \boldsymbol{\theta}^{(M)} . \tag{4.53}$$

The main assumption in Algorithm 4.2 is the ability of evaluating the energy function, $\varphi_T(\cdot)$. We have seen earlier how this can be done in a PF. Roughly speaking, the particle Markov-Chain Monte-Carlo (PMCMC) algorithm consists on putting together these two algorithms [And10]. The resulting PMCMC method can be consulted in Algorithm 4.3

Algorithm 4.2 Robust Adaptive Metropolis

Require: M, $\boldsymbol{\theta}^{(0)}$, $\boldsymbol{\Sigma}^{(0)}$, $\gamma \in (\frac{1}{2}, 1]$, $\bar{\alpha}_*$, and $\varphi_T(\cdot)$

Ensure: $\left\{\boldsymbol{\theta}^{(j)}\right\}_{j=1}^{M}$

Initialize: $\mathbf{S}_0 = \text{Chol}\left(\boldsymbol{\Sigma}^{(0)}\right)$ and $\varphi_T(\boldsymbol{\theta}^{(0)}) = 0$
for $j = 1$ to M **do**
  Draw $\mathbf{a} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
  Compute $\boldsymbol{\theta}^* = \boldsymbol{\theta}^{(j-1)} + \mathbf{S}_{j-1}\mathbf{a}$
  Compute $\alpha^{(j)} = \min\left\{1, \exp(\varphi_T(\boldsymbol{\theta}^{(j-1)}) - \varphi_T(\boldsymbol{\theta}^*))\right\}$
  Draw $u \sim \mathcal{U}(0, 1)$
  if $u \leq \alpha^{(j)}$ **then**
    $\boldsymbol{\theta}^{(j)} = \boldsymbol{\theta}^*$
  else
    $\boldsymbol{\theta}^{(j)} = \boldsymbol{\theta}^{(j-1)}$
  end if
  Compute $\eta^{(j)} = j^{-\gamma}$
  Compute $\mathbf{S}_j = \text{Chol}\left(\mathbf{S}_{j-1}\left(\mathbf{I} + \eta^{(j)}(\alpha^{(j)} - \bar{\alpha}_*)\frac{\mathbf{a}\mathbf{a}^\top}{||\mathbf{a}||^2}\right)\mathbf{S}_{j-1}^\top\right)$
end for

4.5 Computer simulation results

We simulated data of a neuron of the type described in Section 2.3.1, i.e. following a Morris-Lecar model. Particularly, we generated 500 ms of data, sampled at $f_s = 4$ kHz. The model parameters were set to $C_m = 20$ μF/cm^2, $\phi = 0.04$, $V_1 = -1.2$ mV, $V_2 = 18$ mV, $V_3 = 2$ mV, and $V_4 = 30$ mV; the reverse potentials were $E_{\text{L}} = -60$ mV, $E_{\text{Ca}} = 120$ mV, and $E_{\text{K}} = -84$ mV; and the maximal conductances were $\bar{g}_{\text{Ca}} = 4.4$ mS/cm^2 and $\bar{g}_{\text{K}} = 8.0$ mS/cm^2. We considered a measurement noise with a standard deviation of 1 mV and model inaccuracies generated as in Section 4.2. Three sets of simulations are discussed:

- First, we validate the filtering method in a nominal model with perfect knowledge of the model and its inaccuracies gathered in the process noise random term. In this case, the method in Algorithm 4.1 was used.

Algorithm 4.3 Joint state-parameter estimation by Particle MCMC

Require: $\boldsymbol{y}_{1:T}$, M, $\boldsymbol{\theta}^{(0)}$, $\boldsymbol{\Sigma}^{(0)}$, $\gamma \in (1/2, 1]$, $\bar{\alpha}_*$, and $\varphi_T(\cdot)$
Ensure: $\hat{\boldsymbol{x}}_{1:T}$ and $\hat{\boldsymbol{\theta}}$

Initialize: $\mathbf{S}_0 = \text{Chol}\left(\boldsymbol{\Sigma}^{(0)}\right)$ and $\hat{\varphi}_T(\boldsymbol{\theta}^{(0)}) = 0$
for $j = 1$ to M **do**
  Draw $\mathbf{a} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
  Compute $\boldsymbol{\theta}^* = \boldsymbol{\theta}^{(j-1)} + \mathbf{S}_{j-1}\mathbf{a}$
  Run the PF in Algorithm 4.1 with model parameters set to $\boldsymbol{\theta}^*$.
  Required outputs:
    State filtering $\hat{\boldsymbol{x}}^*_{1:T}$ as in (4.22)
    Energy function $\hat{\varphi}_T(\boldsymbol{\theta}^*)$ as in (4.47)
  Compute $\alpha^{(j)} = \min\left\{1, \exp(\hat{\varphi}_T(\boldsymbol{\theta}^{(j-1)}) - \hat{\varphi}_T(\boldsymbol{\theta}^*))\right\}$
  Draw $u \sim \mathcal{U}(0, 1)$
  if $u \leq \alpha^{(j)}$ **then**
    $\boldsymbol{\theta}^{(j)} = \boldsymbol{\theta}^*$
    $\hat{\boldsymbol{x}}_{1:T} = \hat{\boldsymbol{x}}^*_{1:T}$
  else
    $\boldsymbol{\theta}^{(j)} = \boldsymbol{\theta}^{(j-1)}$
  end if
  Compute $\eta^{(j)} = j^{-\gamma}$
  Compute $\mathbf{S}_j = \text{Chol}\left(\mathbf{S}_{j-1}\left(\mathbf{I} + \eta^{(j)}(\alpha^{(j)} - \bar{\alpha}_*)\frac{\mathbf{a}\mathbf{a}^\top}{||\mathbf{a}||^2}\right)\mathbf{S}_{j-1}^\top\right)$
end for
State filtering $\Rightarrow \hat{\boldsymbol{x}}_{1:T}$
Parameter estimation with $\left\{\boldsymbol{\theta}^{(j)}\right\}_{j=1}^{M}$ as in (4.53) $\Rightarrow \hat{\boldsymbol{\theta}}$

- Secondly, the model assumptions are relaxed in the sense that the parameters of the model are not known by the method. We analyze the capabilities of the proposed method to infer both the time-evolving states of the system and some of the parameters defining the model in (4.10). In this case, the method in Algorithm 4.3 was used.

- Finally, we validated the performance of the proposed methods in inferring the synaptic conductances. We used the Morris-Lecar model and a OU model for the synaptic contributions. We tested both PF and PMCMC methods.

4.5.1 Correct model parameters

In the simulations we considered the aforementioned model inaccuracies. To excite the neuron into spiking activity a nominal applied current was injected with $I_o = 110\ \mu\text{A/cm}^2$ and two values for σ_I where considered, namely 1% and 10% of I_o. The nominal conductance used in the model was $\bar{g}_\text{L} = 2\ \text{mS/cm}^2$, whereas the underlying neuron had a zero-mean Gaussian error with standard deviation $\sigma_{\bar{g}_\text{L}}$. Two variance values where considered as well, 1% and 10% of $\bar{g}_\text{L}$. Finally, we considered $\sigma_n = 10^{-3}$ in the dynamics of the gating variable.

In order to evaluate the efficiency of the proposed estimation method, we computed the Bayesian Cramér-Rao Bound (PCRB) according to the recursive formulation given in [Tic98], which we plot as a benchmark for the root mean square error (RMSE) curves obtained by computer simulations after 200 independent Monte Carlo trials. The derivation of the PCRB for the Morris-Lecar model can be consulted in Appendix 4.A. For a generic time series w_k, the RMSE of an estimator $\hat{w}_k$ is defined as

$$\text{RMSE}(w_k) = \sqrt{\mathbb{E}\{(w_k - \hat{w}_k)^2\}} \approx \sqrt{\frac{1}{M}\sum_{j=1}^{M}(w_k - \hat{w}_{j,k})^2}\ , \tag{4.54}$$

where $\hat{w}_{j,k}$ denotes the estimate of w_k at the j-th realization and M the number of independent Monte Carlo trials used to approximate the mathematical expectation.

Figures 4.1 and 4.2 show the time course of the RMSE using $N = \{500, 1000\}$ particles and the PCRB. We see that in both scenarios, our method attains the PCRB and thus is efficient. We measure the efficiency ($\eta \geq 1$) of the method as the quotient between the RMSE and the PCRB, averaged over the entire simulation time. The worse efficiency on estimating v_k was 1.43 corresponding to 500 particles and 10% of inaccuracies, the best was 1.11 for 1000 particles and 1% of errors. In estimating n_k the discrepancy was even lower, 1.06 and 1.03 for maximum and minimum η. To sum up, the PF approaches the PCRB with the number of particles and the performance (both theoretical and empirical) can be improved if model inaccuracies are reduced, i.e., if the model parameters are better estimated at a previous stage. For the sake of clarity, we summarize the results in Table 4.1, where the average RMSE and PCRB along the 500 ms simulation can be consulted. It is apparent that increasing the number of particles from $N = 500$ to $N = 1000$ does not improve significantly the performance of the method.

To give some intuition on the operation and performance of the PF method in Algorithm 4.1, we show the results for a single realization. The results are for 500 particles and two different values for $\sigma^2_{y,k}$, corresponding to 0 and 32 dB respectively. The latter corresponds to $\sigma_{y,k} = 1$ mV, which is considered a reasonable value in today's intracellular

	$\sigma_I = 0.01 \cdot I_o$, $\sigma_{g_\mathrm{L}} = 0.01 \cdot \bar{g}_\mathrm{L}^o$		$\sigma_I = 0.1 \cdot I_o$, $\sigma_{g_\mathrm{L}} = 0.1 \cdot \bar{g}_\mathrm{L}^o$	
	$N = 500$	$N = 1000$	$N = 500$	$N = 1000$
avg. RMSE(v_k)	0.3344	0.3211	0.4269	0.4203
avg. PCRB(v_k)	0.2325	0.2325	0.3777	0.3777
avg. RMSE(n_k)	0.0046	0.0045	0.0056	0.0055
avg. PCRB(n_k)	0.0043	0.0043	0.0053	0.0053

Table 4.1: Averaged results over simulation time.

sensing devices. Even in very low SNR regimes, the method is able to operate and provide reliable filtering results.

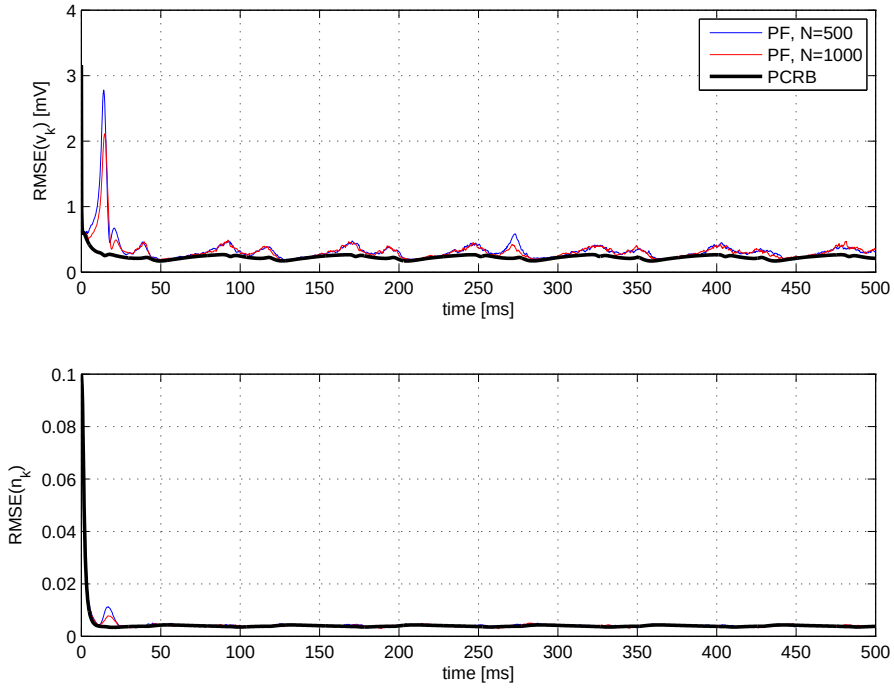


Figure 4.1: Evolution of the RMSE and the PCRB over time. Model inaccuracies where $\sigma_I = 0.01 \cdot I_o$ and $\sigma_{g_\mathrm{L}} = 0.01 \cdot \bar{g}_\mathrm{L}^o$.

4.5.2 Unknown model parameters

In this section we validate the Algorithm presented in Section 4.4. According to the previous analysis, we deem that 500 particles are enough for the filter to provide reliable results. The parameters of the PMCMC algorithm where set to $\gamma = 0.9$, $\bar{\alpha}_* = 0.234$.

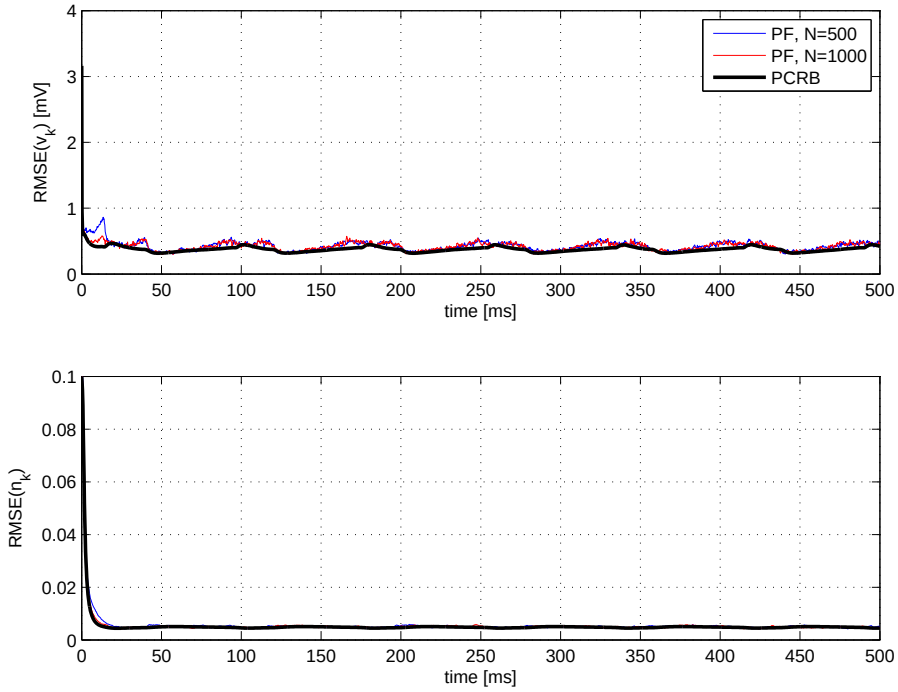
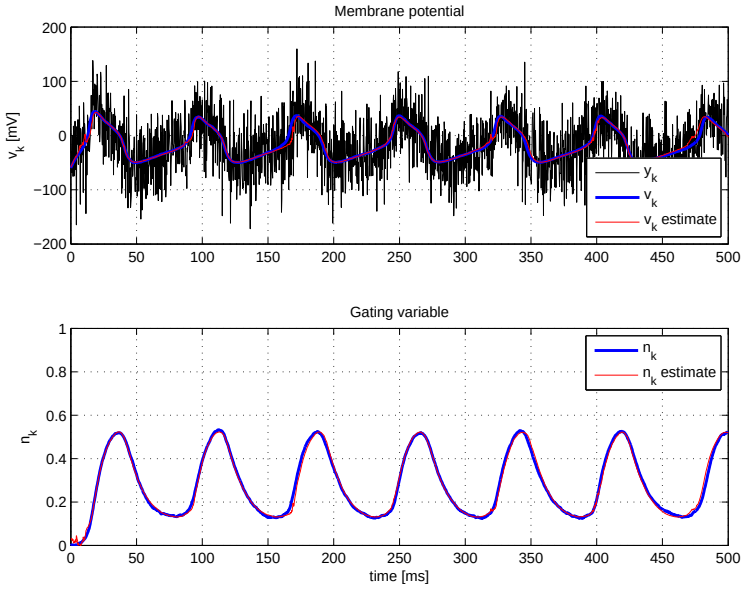


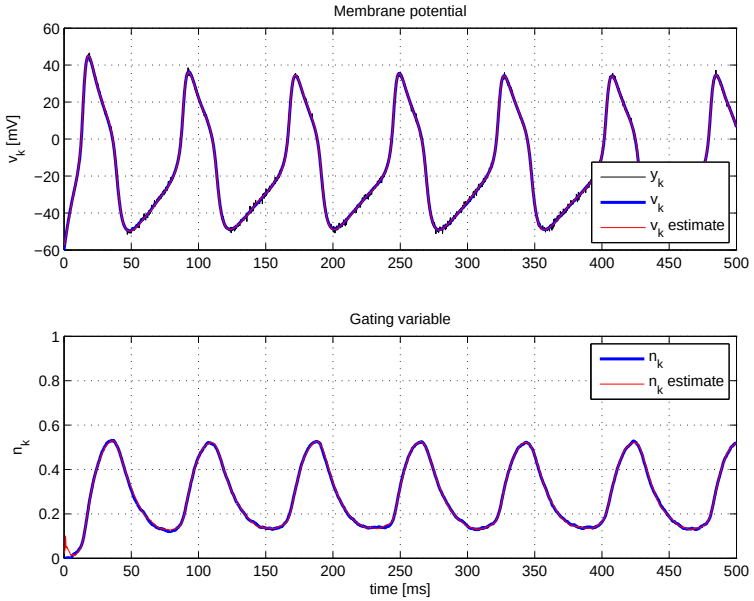
Figure 4.2: Evolution of the RMSE and the PCRB over time. Model inaccuracies where $\sigma_I = 0.1 \cdot I_o$ and $\sigma_{g_{\mathrm{L}}} = 0.1 \cdot \bar{g}_{\mathrm{L}}^{o}$.

Fig. 4.4 shows the results for a single realization when a number of parameters in the nominal model are unknown. We considered 1, 2, and 4 unknown parameters. Each of the plots include $M = 100$ iterations of the MCMC showing the evolution of the parameter estimation (top) and the superimposed recorded voltage in black and the filtered voltage trace in red (bottom). Model inaccuracies are of 1%, similarly as in Fig. 4.1. In these plots we have omitted the estimation of the gating variable for the sake of clarity. The true and initial values used in the experiments, as well as the initial covariance assumed by the method, can be consulted in Table 4.2. From the plots we can observe that the method performs reasonably well even in the case of estimating the model parameters at the same time it is filtering out the noise in the membrane voltage traces.

A biologically meaningful signal is the leakage current, I_L. In general, the leakage gathers those ionic channels that are not explicitly modeled and other non-modeled sources of activity. The parameters driving the leak current are $\bar{g}_L$ and E_{L}. Therefore, we tested and validated the proposed PMCMC in an experiment where the leak parameters are estimated at the same time the filtering solution is computed. Moreover, the statistics of the process noise are estimated as well, $\mathbf{\Sigma}_{x,k}$. In this case, we iterated the PMCMC method 1000 times and average the results over 100 Monte Carlo independent trials. The



(a) SNR = 0 dB



(b) SNR = 32 dB

Figure 4.3: A single realization of the particle filtering method for SNR = 0 dB (left) and SNR = 32 dB (right).

Parameter	True value	Initial value	Initial Covariance
$\bar{g}_{\mathrm{Ca}}$	4.4	8	1
$\bar{g}_{\mathrm{K}}$	8	5	1
σ_v	0.0307	0.05	0.01
σ_n	0.001	0.01	0.001
$\sigma_{y,k}$	1	10	0.5

Table 4.2: True value, initial value, and covariance of the parameters in Fig. 4.4.

results might be consulted in Fig. 4.5, where the RMSE performance of the PMCMC method is compared to the performance of the original PF with perfect knowledge of the model. Also, the performance curves for the estimation parameters are also shown for the sake of completeness.

It can be observed that the filtering performances with perfect knowledge of the model and with estimation of parameters by PMCMC are similar. Moreover, both approaches attain the theoretical lower bound of accuracy given by the PCRB.

In Fig. 4.6, validation results for the parameter estimation capabilities of the PMCMC are shown. Particularly, we plotted in Fig. 4.6(a) and 4.6(c) a number of independent realizations of the samples trajectories $\left\{\boldsymbol{\theta}^{(j)}\right\}_{j=1}^{M}$. We observe that all of them converge to the true values of the parameter. Recall that these true values are

$$\boldsymbol{\theta} = \begin{pmatrix} \bar{g}_L \\ E_{\mathrm{L}} \end{pmatrix} = \begin{pmatrix} 2 \\ -60 \end{pmatrix} . \tag{4.55}$$

In Fig. 4.6(b) and 4.6(d), the average of these realizations can be consulted, where the aforementioned convergence to the true parameter is highlighted.

4.5.3 Estimation of synaptic conductances

Finally, the algorithms were used to estimate jointly the intrinsic states of the neuron and the extrinsic inputs (i.e., the synaptic conductances). In this case, by simply putting together (2.8), (2.9), and (2.22), the continuous-time state-space model is defined by the

Parameter	True value	Initial value	Initial Covariance
τ_{E}	2.73	1.5	1
$g_{\mathrm{E},0}$	12.1	10	1
σ_{E}	12	25	5
τ_{I}	10.49	15	10
$g_{\mathrm{I},0}$	57.3	45	10
σ_{I}	26.4	35	5

Table 4.3: True value, initial value, and covariance of the parameters in Fig. 4.8.

following equations

$$C_m\dot{v} = -\bar{g}_{\mathrm{L}}(v-E_{\mathrm{L}}) - \bar{g}_{\mathrm{Ca}}m_\infty(v)(v-E_{\mathrm{Ca}}) - \bar{g}_{\mathrm{K}}n(v-E_{\mathrm{K}}) - I_{\mathrm{syn}} + I_{\mathrm{app}} \quad (4.56)$$

$$\dot{n} = \phi\frac{n_\infty(v)-n}{\tau_n(v)} \quad (4.57)$$

$$\dot{g}_{\mathrm{E}} = -\frac{1}{\tau_{\mathrm{E}}}(g_{\mathrm{E}}-g_{\mathrm{E},0}) + \sqrt{\frac{2\sigma_{\mathrm{E}}^2}{\tau_{\mathrm{E}}}}\chi \quad (4.58)$$

$$\dot{g}_{\mathrm{I}} = -\frac{1}{\tau_{\mathrm{I}}}(g_{\mathrm{I}}-g_{\mathrm{I},0}) + \sqrt{\frac{2\sigma_{\mathrm{I}}^2}{\tau_{\mathrm{I}}}}\chi\ , \quad (4.59)$$

where $I_{\mathrm{syn}} = g_{\mathrm{E}}(t)(v(t)-E_{\mathrm{E}}) + g_{\mathrm{I}}(t)(v(t)-E_{\mathrm{I}})$. Discretization is straightforward following Section 4.1, but continuous-time provides here a more compact expression of the system.

First, the method with perfect knowledge of the model (see Algorithm 4.1) was validated in Fig. 4.7. It can be observed in Fig. 4.7(a) that the intrinsic signals can be effectively recovered as before where synaptic inputs where not accounted for. The estimation of $g_{\mathrm{E}}(t)$ and $g_{\mathrm{I}}(t)$ is seen in Fig. 4.7(b). We see that the estimation of the the excitatory and inhibitory terms is quite accurate, and that the presence of spikes does not degrade the estimation capabilities of the method.

The PMCMC algorithm (see Algorithm 4.3) was tested in the same experiment. In this case, we assumed that the model parameters related to v_k and n_k where accurately estimated, for instance using the procedure described earlier in this chapter. Therefore, we focused on the estimation of those parameters that describe the OU process of each of the synaptic terms. Particularly, we considered the values in Table 4.3. The results can be consulted in Fig. 4.8 and compared to those in Fig. 4.7. We observe that little degradation with respect to the optimal case of perfectly knowing the model can be identified.

4.6 Summary

In this chapter we proposed a particle filtering method with optimal importance density that is able to sequentially infer the time-course of the membrane potential and the intrinsic activity of ionic channels from noisy observations of voltage traces. In addition, we tackle the problem of joint parameter estimation and state filtering by extending the design PF with an MCMC procedure. The overall method is iterative. The results show the validity of the approaches. The procedure can be applied to any neuron model. Forthcoming applications would be validating the method using real data recordings and adding synaptic terms to the neuron model and use our method to infer the synaptic conductances. The latter problem (addressed in next Chapter) is a challenging hot topic in the neuroscience literature, which is recently focusing on methods to extract the conductances from single-trace measurements, [Béd11], [Kob11]. We think that our PF method would give useful and interesting results to physiologists that aim at inferring brain's activation rules from neurons' activities.

Appendix 4.A PCRB in Morris-Lecar models

This Appendix is devoted to the derivation of the PCRB estimation bound for the Morris-Lecar model used in this Chapter. We follow the procedure learned in Chapter 3 for the case of *nonlinear states and linear measurements under additive Gaussian noise*.

Recall that the state-space we are dealing with is of the form

$$\begin{aligned} \boldsymbol{x}_k &= \mathbf{f}_{k-1}(\boldsymbol{x}_{k-1}) + \boldsymbol{\nu}_k \\ y_k &= \mathbf{h}\boldsymbol{x}_k + e_k \ , \end{aligned} \tag{4.60}$$

where $\mathbf{h} = (1, 0)$, $\boldsymbol{x}_k = (v_k, n_k)^\top$, and $\mathbf{f}_{k-1}(\boldsymbol{x}_{k-1})$ defined by (4.8) and (4.9). The noise terms are of the form

$$\nu_k \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}_{x,k}) \tag{4.61}$$

$$e_k \sim \mathcal{N}(0, \sigma^2_{y,k}) \ . \tag{4.62}$$

In this case, the PCRB can be computed recursively by virtue of Proposition 3.1 with

$$\begin{aligned} \mathbf{D}_k^{11} &= \mathbb{E}_{\boldsymbol{x}_k}\left\{\tilde{\mathbf{F}}_k^\top \boldsymbol{\Sigma}_{x,k}^{-1}\tilde{\mathbf{F}}_k\right\} \\ \mathbf{D}_k^{12} &= \mathbf{D}_k^{21} = -\mathbb{E}_{\boldsymbol{x}_k}\left\{\tilde{\mathbf{F}}_k^\top\right\}\boldsymbol{\Sigma}_{x,k}^{-1} \\ \mathbf{D}_k^{22} &= \boldsymbol{\Sigma}_{x,k}^{-1} + \mathbf{H}_{k+1}^\top \boldsymbol{\Sigma}_{y,k+1}^{-1}\mathbf{H}_{k+1} \ . \end{aligned} \tag{4.63}$$

where we $\mathbf{D}_k^{22}$ becomes deterministic, but the rest of terms involving expectations have to be computed by Monte Carlo approximations.

Since the state function is nonlinear, we use the Jacobian evaluated at the true value of $\boldsymbol{x}_k$ instead, that is

$$\tilde{\mathbf{F}}_k = \left[\nabla_{\boldsymbol{x}_k}\mathbf{f}_k^\top(\boldsymbol{x}_k)\right]^\top = \begin{pmatrix} \frac{\partial f_1}{\partial v_t} & \frac{\partial f_1}{\partial n_t} \\ \frac{\partial f_2}{\partial v_t} & \frac{\partial f_2}{\partial n_t} \end{pmatrix} \ , \tag{4.64}$$

where functions f_1 and f_2 are as in (4.8) and (4.9), respectively. Therefore, to evaluate the bound we need to compute the derivatives in the Jacobian. These are,

$$\frac{\partial f_1(\boldsymbol{x}_k)}{\partial v_k} = 1 - \frac{\mathrm{T}_s}{C_m}\left(\bar{g}_{\mathrm{L}} + \bar{g}_{\mathrm{K}} n_k + \bar{g}_{\mathrm{Ca}}\frac{\partial m_\infty(v_k)}{\partial v_k}v_k + \bar{g}_{\mathrm{Ca}} m_\infty(v_k)\right) \tag{4.65}$$

$$\frac{\partial f_1(\boldsymbol{x}_k)}{\partial n_k} = -\frac{\mathrm{T}_s}{C_m}\bar{g}_{\mathrm{K}}(v_k - E_{\mathrm{K}}) \tag{4.66}$$

$$\frac{\partial f_2(\boldsymbol{x}_k)}{\partial v_k} = \mathrm{T}_s\phi\frac{\frac{\partial n_\infty(v_k)}{\partial v_k}\tau_n(v_k) - (n_\infty(v_{k-1}) - n_{k-1})\frac{\partial \tau_n(v_k)}{\partial v_k}}{\tau_n^2(v_k)} \tag{4.67}$$

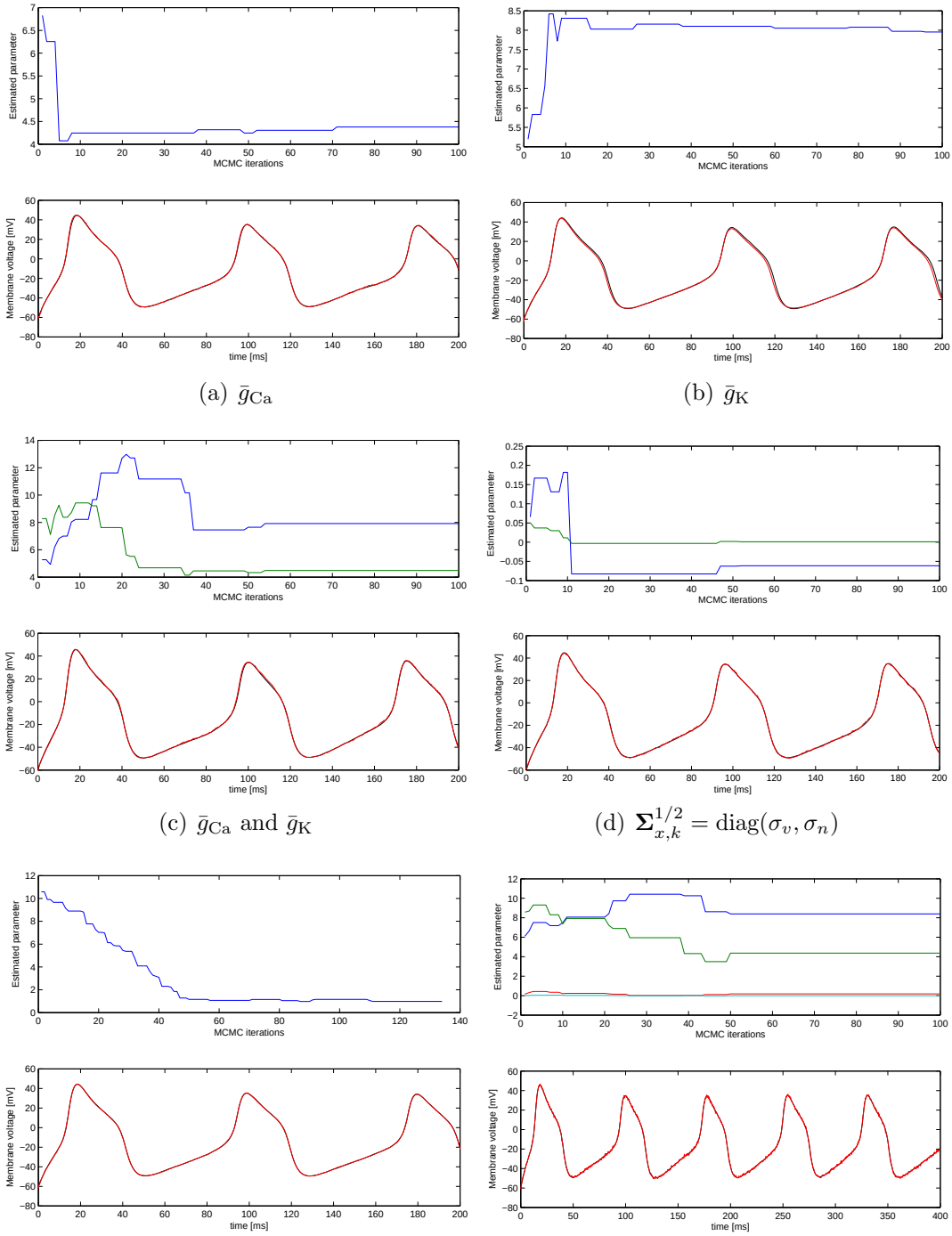
$$\frac{\partial f_2(\boldsymbol{x}_k)}{\partial n_k} = 1 - \frac{\mathrm{T}_s\phi}{\tau_n(v_k)} \tag{4.68}$$

with

$$\frac{\partial m_\infty(v_k)}{\partial v_k} = \frac{1}{2V_2}\mathrm{sech}^2\left(\frac{v_k - V_1}{V_2}\right) \tag{4.69}$$

$$\frac{\partial n_\infty(v_k)}{\partial v_k} = \frac{1}{2V_4}\mathrm{sech}^2\left(\frac{v_k - V_3}{V_4}\right) \tag{4.70}$$

$$\frac{\partial \tau_n(v_k)}{\partial v_k} = -\frac{1}{2V_4}\frac{\sinh\left(\frac{v_k - V_3}{2V_4}\right)}{\cosh^2\left(\frac{v_k - V_3}{2V_4}\right)} \,. \tag{4.71}$$

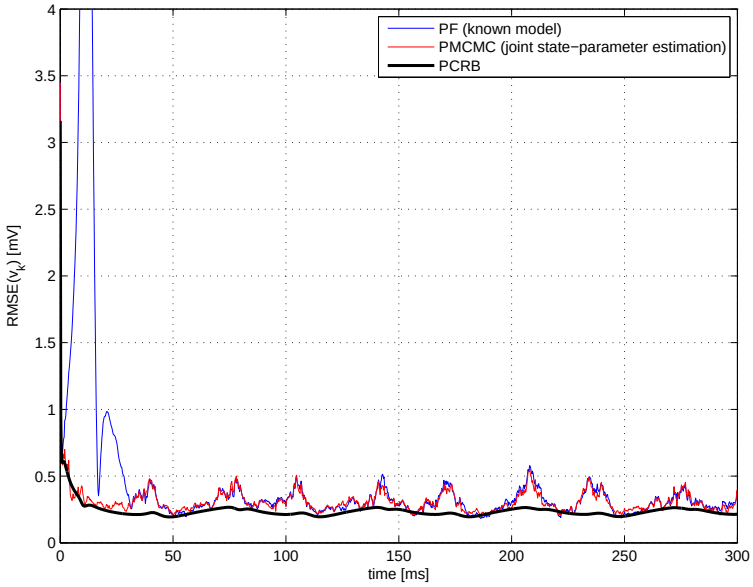


(a) $\bar{g}_{\mathrm{Ca}}$ (b) $\bar{g}_{\mathrm{K}}$

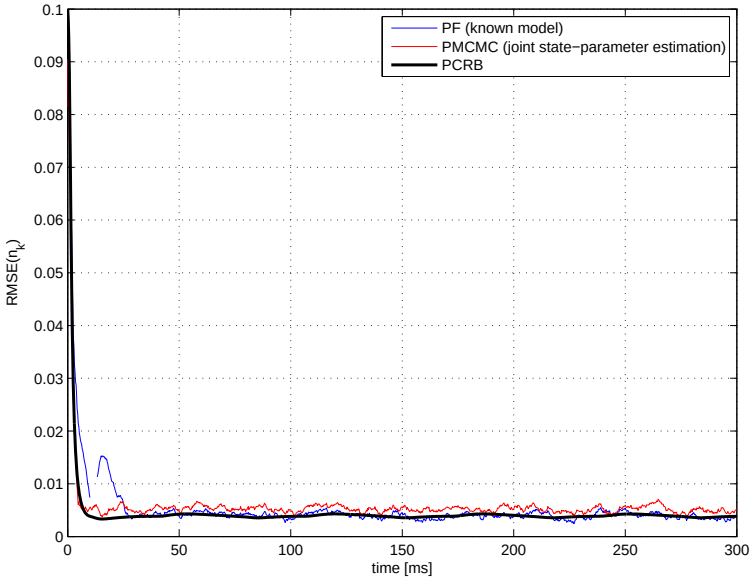
(c) $\bar{g}_{\mathrm{Ca}}$ and $\bar{g}_{\mathrm{K}}$ (d) $\mathbf{\Sigma}_{x,k}^{1/2} = \mathrm{diag}(\sigma_v, \sigma_n)$

(e) $\sigma_{y,k}$ (f) $\bar{g}_{\mathrm{Ca}}$, $\bar{g}_{\mathrm{K}}$, and $\mathbf{\Sigma}_{x,k}^{1/2}$

Figure 4.4: Realizations of the PMCMC algorithm for joint state-parameter estimation. Each plot corresponds to different unknown parameters, featuring the MCMC iterations (top) that converge to the true value of the parameter and the filtered voltage trace (bottom).

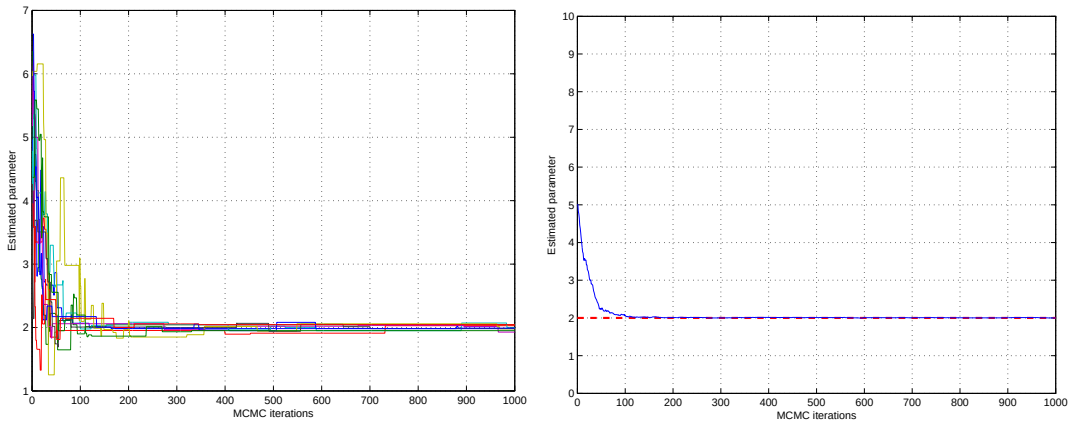


(a)



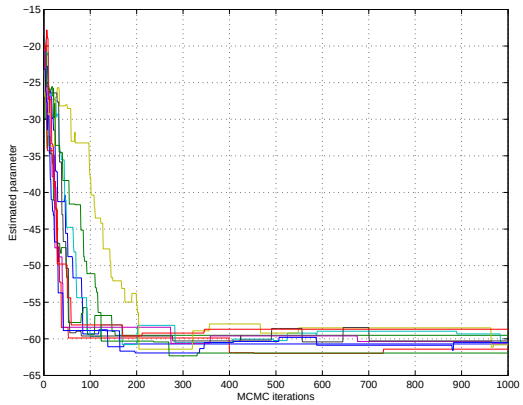
(b)

Figure 4.5: Evolution of RMSE(v_k) (top) and RMSE(n_k) (bottom) over time for the PMCMC method estimating the leakage parameters. Model inaccuracies where $\sigma_I = 0.1 \cdot I_o$ and $\sigma_{g_\mathrm{L}} = 0.1 \cdot \bar{g}_\mathrm{L}^o$.

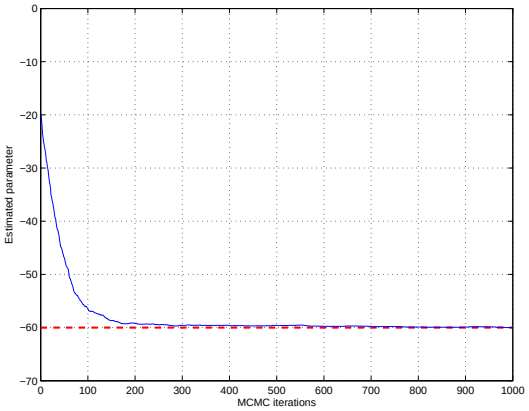


(a) $\bar{g}_{\mathrm{L}}$ estimates

(b) Mean $\bar{g}_{\mathrm{L}}$ estimate

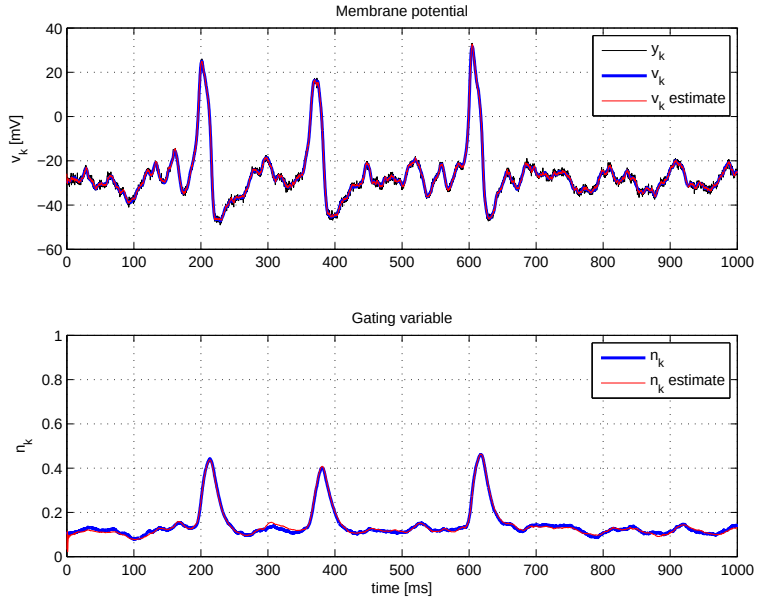


(c) E_{L} estimates

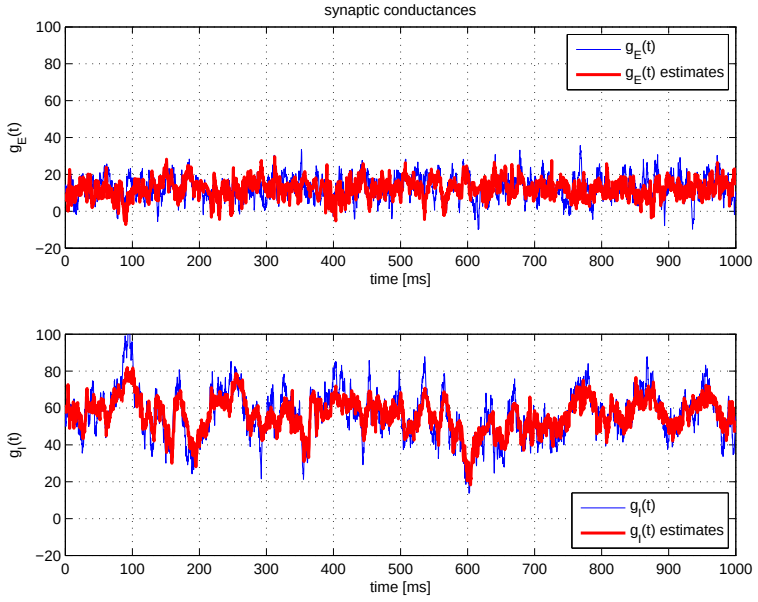


(d) Mean E_{L} estimate

Figure 4.6: Parameter estimation performance of the proposed PMCMC algorithm. Top plots show results for $\bar{g}_{\mathrm{L}} = 2$ estimation and bottom plots for $E_{\mathrm{L}} = -60$. Left plots show superimposed independent realizations and right plots the average estimate of the parameter.

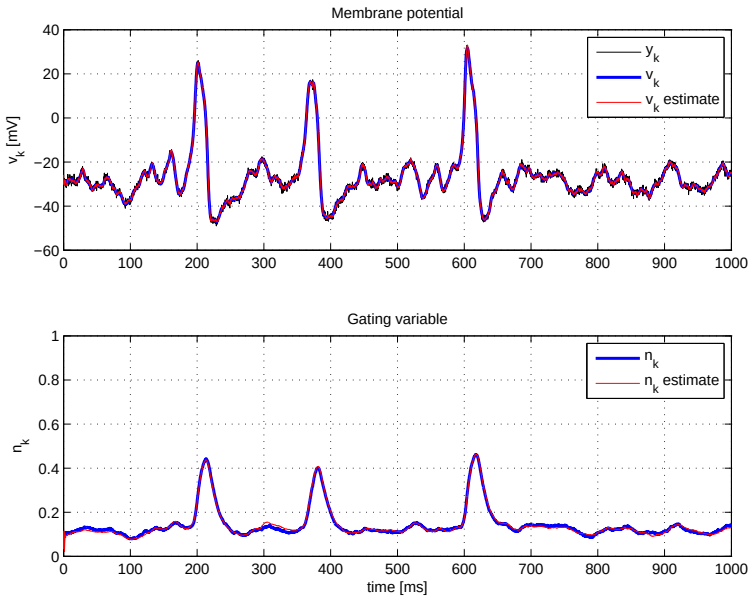


(a)

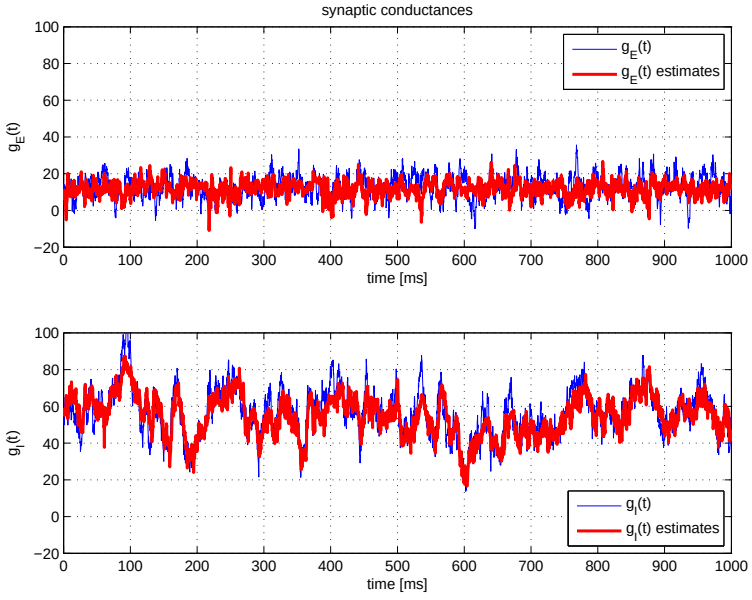


(b)

Figure 4.7: A single realization of the PF method with perfect model knowledge, estimating voltage and gating variables (top) and synaptic conductances in nS (bottom).



(a)



(b)

Figure 4.8: A single realization of the PMCMC method, estimating voltage and gating variables (top) and synaptic conductances in nS (bottom) as well as model parameters.

5

Conclusions and Outlook

THIS Thesis has dealt with the problem of estimating the dynamics of a neuron, its intrinsic signals, and the extrinsic contributions from neighboring neurons. The experimental setup was presented in Fig. 2.3. The **main results and conclusions** are herein itemized:

- We **reviewed the addressed neuroscience problem**, providing some basic facts for the non-specialist to follow the Thesis and understand the contribution.
- We provided a **tutorial review of Bayesian filtering**, which is the mathematical tool used along the dissertation. We identified the existing solutions to treat problems in state-space.
- The **experimental** problem was discussed, highlighting the **impairments** introduced by the sensing devices that provide the observations. Namely, we discussed the need to work in the discrete domain and the efficient treatment of noisy observations.
- We designed a method, based on particle filtering, that is able to **estimate the ionic channel activity and filter out the noise in voltage traces**. We have designed the particle filter such that it uses the optimal importance density to randomly generate the particles, contrary to other solutions in the literature. This

method is model-dependent and requires perfect knowledge of its parameters. The method attains the initial goals of operating in a single-trial basis, sequentially, **in the presence of spikes**, and efficiently.

- To accomplish the targeted **robustness** goal, we investigated a further improvement of the particle filtering algorithm were the parameters of the neural model are estimated from the single-trial voltage trace along with the time-evolving states of the neuron. We used the so-called **particle MCMC** algorithm, which we tested in the problem under study, obtaining promising results. The method attains the initial goals of operating in a single-trial basis, in the presence of spikes, robustly, and efficiently. The method is not purely sequential, as the MCMC step requires the successive processing of batches of observations, and thus there is a need for storing a set of data.

- We were able to show, by realistic computer simulation experiments, that the methods are able to recover the intrinsic activity of a neuron and the extrinsic inputs, i.e. the **synaptic conductances**.

- To benchmark the filtering solutions, we derived the PCRB for the system. This is of particular relevance since no other work in the literature has derived it, and thus compared their results to those provided by the fundamental estimation bounds. We **derived the PCRB** for the state-space system based on the Morris-Lecar neural model, but it could be straightforwardly derived for other neuron models.

- In summary, the reported work goes **beyond the state-of-the-art** in the estimation of neural activity, both providing novel filtering solutions and deriving the benchmark bounds.

Due to time limitations, there are a number of topics that we were not able to address or investigate further in this Thesis. Therefore, the following is a list of **future research topics** that are a result of the work reported in this dissertation:

- The filtering tools proposed in this Thesis should be further investigated in the context of **synaptic conductance** estimation. Here we only draw some preliminary results, but more exhaustive analysis are required.

- Use the methods with **real data** to show the benefits of the tools derived in this work.

- The parameter estimation procedure is not on-line. A further improvement could be to study **purely on-line methods** such as the on-line EM.

- From an algorithmic point of view, one may argue that under the Gaussian assumption one might use **solutions based on SPKF methods**. This would eventually produce savings in the computational complexity of the algorithm, which in the case of PF is rather high. Future work could be addressed to the comparison of such approaches to the ones based on particle filtering that are considered here.

- The inference methods herein proposed are model-dependent. Even the algorithm that is able to estimate the parameters of the model needs a model. Sometime it is not possible to identify which of the plethora of neural mathematical models to select (e.g., Morris-Lecar, HodgkinHuxley, Fitz-Nagumo,etc.). In the Bayesian paradigm it is natural to let the method know that we have **several options for the model** themselves. Therefore, future work could be in the incorporation of the so-called Interating Multiple Model structures in the proposed algorithms.

- We did some preliminary tests with the **smoothing** versions of the proposed filters. The results were promising and could be further investigated.

Bibliography

[And79] B. D. O. Anderson, and J. B. Moore, *Optimal Filtering*, Prentice–Hall, 1979.

[And04] C. Andrieu, Arnaud Doucet, S.S. Singh, and V.B. TADIC, "Particle methods for change detection, system identification, and control", *Proceedings of the IEEE*, vol. 92, no. 3, pp. 423–438, Mar 2004.

[And05] C. Andrieu, Arnaud Doucet, and Vladislav B. Tadic, "On-line parameter estimation in general state-space models", *Decision and Control, 2005 and 2005 European Control Conference. CDC-ECC '05. 44th IEEE Conference on*, pp. 332–337, Dec 2005.

[And10] C. Andrieu, A. Doucet, and R. Holenstein, "Particle Markov chain Monte Carlo methods", *Journal of the Royal Statistical Society Series B*, vol. 72, no. 3, pp. 269–342, 2010.

[Ara07] I. Arasaratnam, S. Haykin, and R. J. Elliot, "Discrete-time nonlinear filtering algorithms using Gauss-Hermite quadrature", *Proc. of the IEEE*, vol. 95, no. 5, pp. 953–977, 2007.

[Ara08] I. Arasaratnam, and S. Haykin, "Square–Root Quadrature Kalman Filtering", *IEEE Trans. Signal Processing*, vol. 56, no. 6, pp. 2589–2593, June 2008.

[Ara09] I. Arasaratnam, and S. Haykin, "Cubature Kalman Filters", *IEEE Trans. Automatic Control*, vol. 54, no. 6, pp. 1254–1269, June 2009.

[Aru02] S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A Tutorial on Particle Filters for Online Nonlinear/Non-Gaussian Bayesian Tracking", *IEEE Trans. Signal Processing*, vol. 50, no. 2, pp. 174–188, February 2002.

[Béd11] C. Bédard, S. Béhuret, C. Deleuze, T. Bal, and A. Destexhe, "Oversampling method to extract excitatory and inhibitory conductances from single-trial membrane potential recordings.", *Journal of neuroscience methods*, set. 2011.

[Bay63] T. Bayes, "An essay towards solving a problem in the doctrine of chances", *The Philosophical Transactions of the Royal Society of London*, vol. 53, pp. 370–418, 1763.

[Ber97] C. Berzuini, N. Best, W. Gilks, and C. Larizza, "Dynamic conditional independence models and Markov chain Monte Carlo methods", *Journal of American Stat. Assoc.*, vol. 92, pp. 1403–1412, 1997.

[Ber99] N. Bergman, *Recursive Bayesian Estimation: Navigation and Tracking Applications*, PhD Thesis, Linköpings Universitet, Sweden, 1999.

[Ber01] N. Bergman, A. Doucet, and N. Gordon, "Optimal estimation and Cramer-Rao bounds for partial non-Gaussian state space models", *Ann. Inst. Statist. Math.*, vol. 53, no. 1, pp. 97–112, 2001.

[Bob75] B. Z. Bobrovsky, and M. Zakai, "A lower bound on the estimation error for Markov processes", *IEEE Trans. Automat. Contr.*, vol. 20, no. 6, pp. 785–788, December 1975.

[Bol04a] M. Bolić, *Architectures for Efficient Implementation of Particle Filters*, PhD Thesis, Stony Brook University, New York, August 2004.

[Bol04b] M. Bolić, P. Djurić, and S. Hong, "Resampling algorithms for particle filters: A computational complexity perspective", *EURASIP Journal on Applied Signal Processing*, , no. 15, pp. 2267–2277, 2004.

[Bol05] M. Bolić, P. M. Djurić, and S. Hong, "Resampling Algorithms and Architectures for Distributed Particle Filters", *IEEE Trans. Signal Processing*, vol. 53, no. 7, pp. 2442–2450, July 2005.

[Bre12] Romain Brette, and Alain Destexhe, *Handbook of Neural Activity Measurement*, Cambridge University Press, 2012.

[Bro11] S. Brooks, A. Gelman, G. Jones, and X. Meng, "Handbook of Markov Chain Monte Carlo", 2011.

[Bug07] M. F. Bugallo, T. Lu, and P. M. Djurić, "Target Tracking by Multiple Particle Filtering", *Proc. of IEEE Aerospace Conference*, Big Sky, MT, USA, March 2007.

[Cev07] V. Cevher, R. Velmurugan, and J. H. McClellan, "Acoustic multitarget tracking using direction-of-arrival batches", *IEEE Trans. Signal Processing*, vol. 55, no. 6, pp. 2810–2825, June 2007.

[Che03] Z. Chen, "Bayesian filtering: From Kalman filters to particle filters, and beyond", Tech. rep., Adaptive Syst. Lab., McMaster University, Ontario, Canada, 2003.

[Clo09] P. Closas, *Bayesian Signal Processing Techniques for GNSS Receivers: from multipath mitigation to positioning*, PhD Thesis, Dept. of Signal Theory and Communications, Universitat Politècnica de Catalunya (UPC), Barcelona, Spain, June 2009.

[Clo12a] P. Closas, and M. F. Bugallo, "Improving Accuracy by Iterated Multiple Particle Filtering", *IEEE Signal Processing Lett.*, vol. 19, no. 8, pp. 531–534, August 2012.

[Clo12b] P. Closas, C. Fernandez-Prades, and J. Vila-Valls, "Multiple Quadrature Kalman Filtering", *Signal Processing, IEEE Transactions on*, vol. 60, no. 12, pp. 6125–6137, Dec 2012.

[Clo13a] P. Closas, and A. Guillamon, "Estimation of neural voltage traces and associated variables in uncertain models", *BMC Neuroscience*, vol. 14, no. 1, pp. 1151, July 2013.

[Clo13b] P. Closas, and A. Guillamon, "Sequential estimation of gating variables from voltage traces in single-neuron models by particle filtering", *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP 2013*, Vancouver, Canada, May 2013.

[Cri01] D. Crisan, *Sequential Monte Carlo methods in Practice*, Chap. Particle Filters - A theoretical perspective, pp. 17–41, Springer-Verlag, 2001.

[Cri02] D. Crisan, and A. Doucet, "A Survey of Convergence Results on Particle Filtering for Practitioners", *IEEE Trans. Signal Processing*, vol. 50, no. 3, pp. 736–746, March 2002.

[Dau03] F. Daum, and J. Huang, "Curse of dimensionality and particle filters", *Proc. of IEEE Aerospace Conference*, vol. 4, pp. 1979–1993, Big Sky, MT, USA, March 2003.

[Day05] Peter Dayan, and L. F. Abbott, *Theoretical Neuroscience: Computational and Mathematical Modeling of Neural Systems*, The MIT Press, 2005.

[Dju03] P. M. Djurić, J. H. Kotecha, J. Zhang, Y. Huang, T. Ghirmai, M. F. Bugallo, and J. Míguez, "Particle Filtering", *IEEE Signal Processing Mag.*, vol. 20, no. 5, pp. 19–38, September 2003.

[Dou98] A. Doucet, "On Sequential Simulation–Based Methods for Bayesian Filtering", Tech. Rep. CUED/F-INFENG/TR.310, University of Cambridge, 1998.

[Dou00] A. Doucet, S. J. Godsill, and C. Andrieu, "On sequential Monte Carlo sampling methods for Bayesian filtering", *Stat. Comput.*, vol. 3, pp. 197–208, 2000.

[Dou01a] A. Doucet, N. de Freitas, and N. Gordon (eds.), *Sequential Monte Carlo Methods in Practice*, Springer, 2001.

[Dou01b] A. Doucet, N. De Freitas, and N. Gordon (eds.), *Sequential Monte Carlo methods in practice*, Springer, 2001.

[Dou05] R. Douc, O. Cappé, and E. Moulines, "Comparison of resampling schemes for particle filtering", *Proc. of the 4th International Symposium on Image and Signal Processing and Analysis, ISPA'05*, pp. 64–69, Zagreb, Croatia, Sept. 2005.

[Erm10] B. Ermentrout, D. Terman, "Mathematical Foundations of Neuroscience", New York: Springer, 2010. ISBN 978-0-387-87708-2.

[Fit61] R. FitzHugh, "Impulses and physiological states in theoretical models of nerve membrane", *Biophys. J.*, vol. 1, pp. 445–466, 1961.

[Gew89] J. Geweke, "Bayesian inference in econometric models using Monte Carlo integration", *Econometrica*, vol. 24, pp. 1317–1399, 1989.

[Gil96] W.R. Gilks, S. Richardson, and D.J. Spiegelhalter (eds.), *Markov Chain Monte Carlo in Practice: Interdisciplinary Statistics*, CRC Interdisciplinary Statistics Series, Chapman & Hall, 1996.

[God01] S. Godsill, and T. Clapp, *Sequential Monte Carlo methods in Practice*, Chap. Improvement strategies for Monte Carlo particle filters, pp. 139–158, Springer-Verlag, 2001.

[Gol73] G. Golub, and V. Pereyra, "The differentiation of pseudo-inverses and nonlinear least squares problems whose variables separate", *SIAM J. Numer. Anal.*, vol. 10, pp. 413–432, 1973.

[Gol96] G. H. Golub, and C. F. van Loan, *Matrix Computations*, The John Hopkins University Press, $3^{\underline{\text{rd}}}$ ed., 1996.

[Gor93] N. Gordon, D. Salmond, and A. Smith, "A novel approach to nonlinear/non-Gaussian Bayesian state estimation", *IEE Proceedings on Radar and Signal Processing*, vol. 140, pp. 107–113, 1993.

[Gui06] Antoni Guillamon, David W. McLaughlin, and John Rinzel, "Estimation of synaptic conductances.", *Journal of physiology, Paris*, vol. 100, no. 1-3, pp. 31–42, 2006.

[Ham54] J. M. Hammersley, and K. W. Morton, "Poor man's Monte Carlo", *Journal of the Royal Statistical Society*, vol. 16, pp. 23–38, 1954.

[Hen08] G. Hendeby, *Performance and Implementation Aspects of Nonlinear Filtering*, PhD Thesis, Linköpings Universitet, Linköping, Sweden, 2008.

[Hig95] T. Higuchi, "Kitagawa monte carlo filter from the perspective of genetic algorithm", Research memorandum, The Institute of Statistical Mathematics, 1995.

[Hod52] A. L. Hodgkin, and A. F. Huxley, "The components of membrane conductance in the giant axon of Loligo", *J Physiol.*, vol. 116, no. 4, pp. 473–496, April 1952.

[Ito00] K. Ito, and K. Xiong, "Gaussian Filters for Nonlinear Filtering Problems", *IEEE Trans. Automat. Contr.*, vol. 45, no. 5, May 2000.

[Izh04] E.M. Izhikevich, "Which model to use for cortical spiking neurons?", *Neural Networks, IEEE Transactions on*, vol. 15, no. 5, pp. 1063–1070, Sept 2004.

[Izh06] E. Izhikevich, *Dynamical systems in neuroscience: the geometry of excitability and bursting*, MIT Press, Cambridge, MA, 2006.

[Jaz70] A. H. Jazwinski, *Stochastic Processes and Filtering Theory*, vol. 64 of *Mathematics in Science and Engineering*, Academic Press, 1970.

[Jul97] S. Julier, and J. K. Uhlmann, "A new extension of the Kalman filter to nonlinear systems", *Int. Symp. Aerospace/Defense Sensing, Simul. and Controls*, vol. 3, 1997.

[Jul00] S. Julier, J. Uhlmann, and H. F. Durrant-White, "A new method for nonlinear transformation of means and covariances in filters and estimators", *IEEE Trans. Automat. Contr.*, vol. 45, pp. 477–482, March 2000.

[Jul04] S. J. Julier, and J. K. Ulhmann, "Unscented filtering and nonlinear estimation", *Proc. of the IEEE*, vol. 92, no. 3, pp. 401–422, March 2004.

[Kal60] R. E. Kalman, "A new approach to linear filtering and prediction problems", *Transactions of the ASME-Journal of Basic Engineering*, vol. 82, no. Series D, pp. 35–45, 1960.

[Kar05] R. Karlsson, *Particle Filtering for Positioning and Tracking Applications*, PhD Thesis, Linköpings Universitet, Linköping, Sweden, March 2005.

[Kay93] S. M. Kay, *Fundamentals of Statistical Signal Processing. Estimation Theory*, Prentice Hall, 1993.

[Kee09] James P. Keener, and James Sneyd, *Mathematical physiology. I. , Cellular physiology*, Interdisciplinary applied mathematics, Springer, New York, London, 2009.

[Kit96] G. Kitakawa, "Monte Carlo filter and smoother for non-Gaussian non-linear state space models", *Journal of Computational and Graphical Statistics*, vol. 5, pp. 1–25, 1996.

[Kob11] R. Kobayashi, Y. Tsubo, P. Lansky, and S. Shinomoto, "Estimating time-varying input signals and ion channel states from a single voltage trace of a neuron", *Advances in Neural Information Processing Systems (NIPS)*, vol. 24, pp. 217–225, 2011.

[Kon94] A. Kong, J. S. Liu, and W. H. Wong, "Sequential imputations and Bayesian missing data problems", *Journal of the American Stat. Assoc.*, vol. 89, no. 425, pp. 278–288, 1994.

[Kot03a] M. Kotecha, and P. M. Djurić, "Gaussian Particle Filtering", *IEEE Trans. Signal Processing*, vol. 51, no. 10, pp. 2592–2601, October 2003.

[Kot03b] M. Kotecha, and P. M. Djurić, "Gaussian Sum Particle Filtering", *IEEE Trans. Signal Processing*, vol. 51, no. 10, pp. 2602–2610, October 2003.

[Lan13] M. Lankarany, W. P. Zhu, M. N. S. Swamy, and Taro Toyoizumi, "Inferring trial-to-trial excitatory and inhibitory synaptic inputs from membrane potential using Gaussian mixture Kalman filtering", *Frontiers in Computational Neuroscience*, vol. 7, 2013.

[Lan14] M. Lankarany, W.-P. Zhu, and M.N.S. Swamy, "Joint estimation of states and parameters of hodgkinhuxley neuronal model using kalman filtering", *Neurocomputing*, vol. 136, pp. 289 – 299, 2014.

[Liu95] J. S. Liu, and R. Chen, "Blind deconvolution via sequential imputations", *Journal of the American Stat. Assoc.*, vol. 90, pp. 567–576, 1995.

[Liu98] J. S. Liu, and R. Chen, "Sequential Monte Carlo methods for dynamical problems", *Journal of the American Stat. Assoc.*, vol. 93, pp. 1032–1044, 1998.

[Liu08] Jun S. Liu, *Monte Carlo Strategies in Scientific Computing*, Springer, jan. 2008.

[Míg04] J. Míguez, M. F. Bugallo, and P. M. Djurić, "A new class of particle filters for random dynamic systems with unknown statistics", *EURASIP J. Appl. Signal Processing*, vol. 2004, no. 15, pp. 2278–2294, 2004.

[McG00] S. McGinnity, and G. W. Irwin, "Multiple Model Bootstrap Filter for maneuvering target tracking", *IEEE Trans. Aerosp. Electron. Syst.*, vol. 36, pp. 1006–1012, 2000.

[McL00] D. McLaughlin, R. Shapley, M. Shelley, and D. J. Wielaard, "A neuronal network model of macaque primary visual cortex (V1): orientation selectivity and dynamics in the input layer 4Calpha.", *Proceedings of the National Academy of Sciences of the United States of America*, vol. 97, no. 14, pp. 8087–8092, jul. 2000.

[Mer00] R. Van der Merwe, A. Doucet, N. Freitas, and E. Wan, "The unscented particle filter", Tech. Rep. CUED/F-INFENG/TR 380, University of Cambridge, August 2000.

[Mer03] R. Van Der Merwe, and E. Wan, "Sigma-Point Kalman Filters for Probabilistic Inference in Dynamic State-Space Models", *Proceedings of the Workshop on Advances in Machine Learning*, Montreal, Canada, June 2003.

[Mor81] C. Morris, and H. Lecar, "Voltage Oscillations in the barnacle giant muscle fiber", *Biophys J.*, vol. 35, no. 1, pp. 193–213, July 1981.

[Mus00] C. Musso, N. Oudjane, and F. Le Gland, *Sequential Monte Carlo methods in Practice*, Chap. Improving Regularised Particle Filters, pp. 247–272, Springer-Verlag, 2000.

[Nag62] J. Nagumo, S. Arimoto, and S. Yoshizawa, "An active pulse transmission line simulating nerve axon", *Proceeding IRE*, vol. 50, pp. 2061–2070, 1962.

[Nor00] M. Norgaard, N. K. Poulsen, and O. Ravn, "New developments in state estimation of nonlinear systems", *Automatica*, vol. 36, pp. 1627–1638, 2000.

[Pan12] Liam Paninski, Michael Vidne, Brian DePasquale, and Daniel Gil Ferreira, "Inferring synaptic inputs given a noisy voltage trace via sequential Monte Carlo methods", *Journal of Computational Neuroscience*, vol. 33, no. 1, pp. 1–19, 2012.

[Pap01] A. Papoulis, and S. U. Pillai, *Probability, Random Variables and Stochastic Processes*, McGraw–Hill, New Delhi, India, $4^{\underline{\text{th}}}$ ed., 2001.

[Pit01] M. Pitt, and N. Shephard, *Sequential Monte Carlo methods in Practice*, Chap. Auxiliary variable based particle filters, pp. 273–294, Springer-Verlag, 2001.

[Piw04] M. Rudolph Z. Piwkowska, M. Badoual, T. Bal, and A. Destexhe, "A method to estimate synaptic conductances from membrane potential fluctuations", *J. Neurophysiol.*, no. 91, pp. 2884–2896, 2004.

[Pos08] Martin Pospischil, Maria Toledo-Rodriguez, Cyril Monier, Zuzanna Piwkowska, Thierry Bal, Yves Frégnac, Henry Markram, and Alain Destexhe, "Minimal hodgkin-huxley type models for different classes of cortical and thalamic neurons.", *Biological Cybernetics*, vol. 99, no. 4-5, pp. 427–441, 2008.

[Pot63] J. E. Potter, and R. G. Stern and, "Statistical filtering of space navigation measurements", *Proceedings of the AIAA Guidance, Navigation, and Control Conference*, Cambridge, MA, August 1963.

[Poy11] George Poyiadjis, Arnaud Doucet, and Sumeetpal S. Singh, "Particle approximations of the score and observed information matrix in state space models with application to parameter estimation", *Biometrika*, no. 98, pp. 65–80, 2011.

[Pun03] E. Punskaya, *Sequential Monte Carlo Methods for digital communications*, PhD Thesis, University of Cambridge, UK, July 2003.

[Rin98] J. R. Rinzel, and G. B. Ermentrout, "Analysis of neural excitability and oscillations", C. Koch, I. Segev (eds.), *Methods in Neural Modeling*, pp. 135–169, MIT Press., Cambridge, MA, 1998.

[Ris04] B. Ristic, S. Arulampalam, and N. Gordon, *Beyond the Kalman Filter: Particle Filters for tracking applications*, Artech House, Boston, 2004.

[Rub88] D. B. Rubin, J. M. Bernardo, M. H. De Groot, D. V. Lindley, and A. F. M. Smith, *Bayesian Statistics 3*, Oxford: University Press, 1988.

[Rud03] M. Rudolph, and A. Destexhe, "Characterization of subthreshold voltage fluctuations in neuronal membranes", *Neural Comput.*, vol. 15, no. 11, pp. 2577–2618, nov. 2003.

[Sär13] S. Särkkä, *Bayesian Filtering and Smoothing*, Cambridge University Press, New York, NY, USA, 2013.

[Sch05] T. Schön, F. Gustafsson, and P.J. Nordlund, "Marginalized particle filters for mixed linear/nonlinear state-space models", *IEEE Trans. Signal Processing*, vol. 53, no. 7, pp. 2279–2289, July 2005.

[Sim01] M. Simandl, J. Královec, and P. Tichavský, "Filtering, predictive, and smoothing Cramér-Rao bounds for discrete-time nonlinear dynamic systems", *Automatica*, vol. 37, pp. 1703–1716, 2001.

[Sor88] H. Sorenson, *Recursive estimation for nonlinear dynamic systems*, New York: Marcel Dekker, 1988.

[Spe02] "Special Issue on Monte Carlo methods for Statistical Signal Processing", *IEEE Trans. Signal Processing*, vol. 50, no. 2, February 2002.

[Str71] A. H. Stroud, *Approximate calculation of multiple integrals*, Prentice–Hall, Englewood Cliffs, NJ, 1971.

[Tao04] Louis Tao, Michael Shelley, David Mclaughlin, and Robert Shapley, "An egalitarian network model for the emergence of simple and complex cells in visual cortex.", *Proc. Natl. Acad. Sci. USA*, vol. 101, no. 1, pp. 366–371, jan. 2004.

[Tic98] P. Tichavský, C. .H. Muravchik, and A. Nehorai, "Posterior Cramér-Rao Bounds for Discrete-Time Nonlinear Filtering", *IEEE Trans. Signal Processing*, vol. 46, no. 5, pp. 1386–1396, May 1998.

[Tre68] Harry L. Van Trees, *Part I of Detection, Estimation and Modulation Theory*, Wiley Interscience, New York, 1968.

[Tre07] Harry L. Van Trees, and Kristine L. Bell (eds.), *Bayesian Bounds for Parameter Estimation and Nonlinear Filtering/Tracking*, Wiley Interscience, 2007.

[Van01] R. Van der Merwe, and E. A. Wan, "The square–root unscented Kalman filter for state and parameter estimation", *Proceedings of ICASSP'01*, vol. 6, pp. 3461–3464, Salt Lake City, UT, May 2001.

[Vih12] Matti Vihola, "Robust adaptive metropolis algorithm with coerced acceptance rate.", *Statistics and Computing*, vol. 22, no. 5, pp. 997–1008, 2012.

[Wan00] E. Wan, and R. Van der Merwe, "The unscented kalman filter for nonlinear estimation", *IEEE Symp. Adaptive Systems for Signal Proc., Comm. and Control*, pp. 153–158, Alberta, Canada, 2000.

[Wie01] D. J. Wielaard, M. Shelley, D. McLaughlin, and R. Shapley, "How Simple Cells Are Made in a Nonlinear Network Model of the Visual Cortex", *Journal of Neuroscience*, vol. 21, pp. 5203–5211, 2001.

Index