

DTU Fotonik
Department of Photonics Engineering

Machine Learning Methods for Laser Intensity and Frequency Noise Characterization

Giovanni Brajato

Supervisors

Darko Zibar - (dazi@fotonik.dtu.dk)

Jesper Mørk - (jesm@fotonik.dtu.dk)

Kongens Lyngby 2018



DTU Fotonik
Department of Photonics Engineering
Technical University of Denmark

Ørstedes Plads
Building 343
2800 Kongens Lyngby, Denmark
Phone +45 4525 6352
www.fotonik.dtu.dk

Contents

Contents	i
Acknowledgements	iii
Preface	v
Abstract	vii
1 Introduction	1
1.1 From electrons to photons	1
1.2 A machine learning approach	2
2 Laser Dynamics and model simulation	5
2.1 Introduction	5
2.2 Laser modelling	5
2.3 Numerical methods for SDEs	11
2.4 System simulation	14
3 Tracking in state space models	21
3.1 Introduction	21
3.2 From deterministic to probabilistic state-space models	21
3.3 Bayesian filtering equations	22
3.4 Kalman filtering theory	24
3.5 RIN and FN estimation	26
4 Parameter estimation in a Bayesian framework	35
4.1 Introduction	35
4.2 Energy function	36
4.3 Point estimate	37
4.4 Density estimate: Metropolis-Hastings sampling	38
4.5 Testing and validation	40
5 Experimental results	49
5.1 Introduction	49
5.2 Laser set-up and measurement acquisition	49

5.3	Methodology used and initialization	50
5.4	RIN, FN and parameter estimation	51
6	Conclusion	57
6.1	Future work	57
A	Appendix	59
	Bibliography	63

Acknowledgements

I dedicate this thesis to the people I love, who always have been supporting and believing me in what I do. My family, my friends near and far, my supervisors: thank you!

Preface

This MSc thesis was prepared at the department of Photonics Engineering at the Technical University of Denmark in fulfillment of the requirements for acquiring a MSc degree in Telecommunication Engineering. Part of the results achieved here have been submitted in a paper with title "Joint Learning of Laser Relative Intensity and Frequency Noise from Single Experiment and Single Detected Quadrature" and accepted for an oral presentation at the 44th European Conference on Optical Communication-ECOC, held in Rome, Italy, on September 23 - 27, 2018.

Abstract

The trend of next generation computing circuits involves the use of light to create so-called photonics computing systems, aimed to break the current limits relying on current silicon-based chips. Integrating optical component at nano-scale level involves careful design of microscopic coherent light sources: *nanolasers*. Model such components requires high characterization of the dynamics involved in the laser cavity, in particular its noise characterization. This is conventionally done by a direct analysis of the optical power spectrum. For nanolasers however, the limited output SNR limit the precision of such techniques.

By use of Bayesian machine learning techniques, this thesis wants to address an alternative way to characterize such light sources. The methods presented here involves Bayesian filtering techniques for joint estimation of amplitude and phase noise and Monte Carlo sampling algorithms to estimate the laser static parameters governing the cavity. This endorses a more precise depiction of a lasing source in presence of noisy detection, by defining it according to the static and dynamic parameters which describe the laser internal behavior.

All the proposed techniques are first tested and validated on a simulated environment and then applied to infer laser specifications from a mathematical model in experimental data obtained from conventional sources. The programming environment used is MATLAB®. The results obtained from experimental data are in-line with the expectations. This concludes that the framework appears promising in term of laser characterization and may open a new area of research in machine learning applied to laser inference.

CHAPTER 1

Introduction

1.1 From electrons to photons

Ever since the first world wide network was created, the diffusion of Internet and the ease of information access grew exponentially. The consequences from a massive and free flow of information exchanged around the globe conditioned the whole evolution of economy, politics and society. Following in parallel with the last industrial revolution, the necessity to shorten distances and move information in a fast and reliable way made possible to invest resources and efforts to develop new dedicated technologies [1]. The last century has seen the advance in the technology field of *Telecommunications*, one of the most mesmerizing aspect of the modern era. One of the factor that allowed this rapid expansion involves the improvement on the *way* information is transmitted. By far, to stand up against the massive mole of data flowing exchanged globally [2, 3], the most efficient method involves the use of electromagnetic waves (EM) as carriers of information. Thanks to optical fibers which allows to confine certain EM frequencies in a waveguide, the visible and near visible spectra are widely exploited for data transmission [4] and characterize the systems dealing with long-distance backbone links [5].

In the recent decades, light-based transmission systems have started to move towards smaller scales. Thanks to price drops due to large scale manufacturing, the technology cost of semiconductor-based components decreased significantly. Coupled with growing investments in research, this led also to a progressive miniaturization and large use of silicon-derived components in nowadays appliances, for example *system on chip* (SOC) used in modern microprocessors.

The next step of the current trend is to move from optical transmission to optical computing [6], to revolution the way digital logic is processed. Instead of exploiting transistor to manipulate electric voltages in a Boolean logic, optical circuits employs photons and non-linear optic effects to develop logic gates [7]. This aims to overcome the technical limits arising from miniaturization of logic processors. Efforts have been made made to create hybrid chips based on electronic and optical components [8, 9, 8]. However, one of the key challenge remains the implementation of light sources on a chip. Semiconductor lasers [10], widely used for a daily range of applications, are one of promising coherent signal sources that may be integrated in nanoscale photonic chips, under the name of nanolasers [11, 12].

Nanolasers are lasers operating at scales on the order of nanometers [13]. At

this resolution, the design of the internal cavity can be very sensitive to manufacture process defects. Therefore, it is desirable to accurately test and control the cavity characteristics. In order to achieve this, it is important to accurately define the laser dynamic and extract the numerical parameters involved in the mathematical model. Moreover, it is beneficial to quantify also the accuracy of such model and relative components.

Typical mathematical models of lasers and more in general, dynamical processes, involves the use of stochastic differential equations (SDE). Those models include randomness occurring in nature which affects the laser internal behaviour. As it will be clear in Chapter 2, the noisy fluctuation of the internal laser cavity affects the light that will be produced. It is important to characterize this noise floor because, in a photonic chip setting, discern a high state (1) from a low state (0) is crucial to avoid logic errors. If the noise generated from a source is not accurately modelled, it can mislead the logic computation interpreting 0 as 1 and vice versa.

The problem arises from the difficulty to characterize these small components, since the only information provided is contained in the emitted light. In conventional lasers it is possible to provide a characterization of the noise and the parameters governing the model [14, 15]. However, when nanolasers come to play, the ultra low output power limit the achievable signal to noise ration (SNR) thus by compromising the quality of estimate.

1.2 A machine learning approach

A huge contribution in the field of computer science in the last decade is give by machine learning (ML) [16]. Machine learning is a sub-field of artificial intelligence which constituting all the techniques aimed to infer knowledge about great moles of observed data. Such techniques are widely exploited nowadays in many daily-life problems, such as computer vision, pattern recognition, speech synthesis and processing and so on [17]. The Idea to applying machine learning to solve problems occurring in optical communication is fairly new in the scientific community [18, 19, 20, 21] and it has shown itself as a promising application. The branch of ML that can help to tackle the problem of laser noise characterization are Bayesian techniques for filtering and parameter estimation. Such methods involve the inference of static and dynamic parameters involved in a dynamical process (laser) based on observation of noisy data (emitted light). By filtering out the noise affecting the measures, the Bayesian approach improves the estimation of cavity noise in low SNR settings and computes a probability density function (PDF) of the laser parameters given the obtained measurements.

Bayesian filtering had already been experimented for tracking the phase noise of an optical channel [22] and a low output SNR laser [23]. In this thesis the filtering algorithm implements a full non-linear model for the laser dynamics which parameters can be inferred from the observed data. In particular, the described technique involves the joint characterization of the amplitude and phase noise of a laser using a Kalman-

based filtering technique and merging it together with a Metropolis-Hasting algorithm for static parameter estimation. On the hypothesis that a laser is correctly defined by its mathematical model, this technique can completely specify a laser from the emitted light. In addition, since it can be applied to any dynamical model, the general work-flow developed here can be theoretically adapted to infer parameters involving lasers described with different models.

This thesis is organized as follow: the first part, comprising Chapter 2, 3 and 4 present the theory and the methodology used; the second part, Chapter 5 and 6 illustrates the results obtained on experimental data and relative discussion. Chapter 2 illustrates the physical principles of a laser diode, as well as the mathematics involved to represent and solve numerically a system of stochastic differential equations. It concludes with a numerical simulation of a laser, used in the subsequent chapter to assess and validate the tracking and sampling algorithms. Chapter 3 introduces the concepts of probabilistic state-space modelling, and the necessary assumption to applying Kalman-based filters on the sequence of measurement obtained from the laser. Two Kalman-filter based algorithms are then applied to a numerical simulation to assess the goodness of each tracker in presence of noisy observation. Chapter 4 explain the concepts of parameter estimation and sampling, to describe then how to use them to infer the posterior distribution of the laser parameters. Also here the chapter concludes with test applied on simulated data. Results obtained on experimental data are shown in Chapter 5 and Chapter 6 yields the conclusion of this thesis work, what was achieved and possible future directions.

CHAPTER 2

Laser Dynamics and model simulation

2.1 Introduction

This chapter presents an overview of the working principle of a laser system, based on the rate-equation model. It then follows by presenting numerical methods to solve the differential equations involved in the process and use them to create a numerical simulation of a laser. It concludes by presenting some results on simulation, and shows the conventional way to estimate amplitude and phase noise from a series of detected measurements.

2.2 Laser modelling

LASER is an acronym who stands for *light amplification by stimulated emission of radiation*, coined by Gordon in [24]. Such a device is a light source capable of emitting spatially and temporally correlated photons, by mean of an optical gain medium in an optical resonant cavity.

In a diode laser, this coherent emission is achieved by interaction between carriers and photons present in the semiconductor resonant cavity [25]. A carrier is an electron-hole pair present in the semiconductor material. Normally, when no external stimuli are present, all the free electrons occupies the next available hole and no carriers are present. However, when an external source of energy is fed into the cavity, some bonds of electrons and their nuclei are broken creating new carriers. These free carriers can spontaneously recombine but a faction of them release the excess energy after recombination in the form of photons. Free photons in the cavity contribute to achieve *stimulated recombination*: an incident photon stimulate the recombination of a carrier and simultaneously generate a new photon, identical to the first one. This process is responsible to create the optical gain of a laser. This is the key to obtain coherent light, which propagates inside the cavity and eventually scatters out from it becoming available optical power.

2.2.1 The rate equations

The difference between a laser and a nanolaser is substantially on the cavity dimensions and the static parameter that regulates such emission. Therefore, a general mathematical model for a laser can be adapted, without loss of generalities, to smaller scales. From a systematic point of view, a laser can be seen as a system in which two populations, carriers and photons interact each other (see Figure 2.1). Dynam-

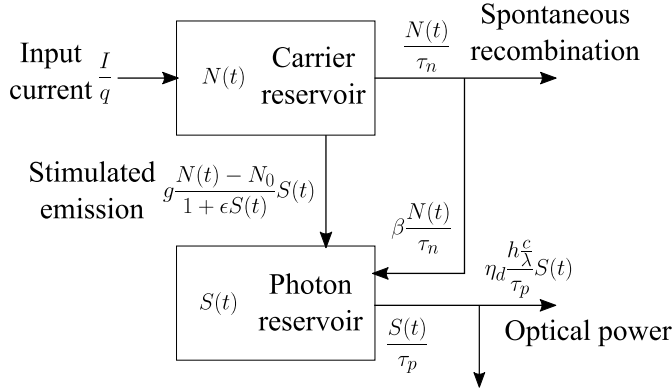


Figure 2.1: Basic schematic of the laser's dynamics.

ical systems like this are nicely described by a set of ordinary differential equations (ODE), which describe the time-behaviour of carriers $N(t)$ and photons $S(t)$

$$\frac{dN(t)}{dt} = \frac{I}{q} - \frac{N(t)}{\tau_n} - g \frac{N(t) - N_0}{1 + \epsilon S(t)} S(t) \quad (2.1)$$

$$\frac{dS(t)}{dt} = g \frac{N(t) - N_0}{1 + \epsilon S(t)} S(t) - \frac{S(t)}{\tau_p} + \frac{\beta N(t)}{\tau_n} \quad (2.2)$$

In (2.1), one can see that the number of carriers increases by supplying an external input I/q , where I is an external current fed into the cavity and q is the elementary charge. A spontaneous recombination occurs at an exponential decay rate τ_n . The stimulated emission process mentioned above transfer elements from the carrier reservoir to the photon reservoir, with a rate dependent on both reservoirs and some parameters such as the non-linear gain compression factor ϵ , the gain slope constant g and the carriers at transparency N_0 . The photon's population described in (2.2) grows thanks to the stimulated recombination and to a fraction β of the spontaneous recombination. The photon's losses in the cavity and their propagation outside are

modelled using an exponential decay term with constant τ_p . The effective output optical power that can be detected, usually by employing a *photodiode*, is

$$P(t) = \eta_d \frac{hc}{\lambda \tau_p} S(t) \quad (2.3)$$

where η_d is the differential quantum efficiency of the detector, h is the Planck's constant, c is the light speed in the medium and λ is the emitted wavelength.

With the basic schematic of Figure 2.1, the two reservoir reaches an equilibrium when $dN(t)/dt = dS(t)/dt = 0$. This is called *steady state* of the system, reached with a constant input current outputting a constant optical power P_{ss} . The steady state for (2.1) and (2.2) is uniquely determined from the static parameters of the system, and it can be calculated by solving by setting the time-derivatives to zero [14] as

$$\begin{aligned} N_{ss} &= \frac{\tau_n S_{ss}(I\epsilon + gN_0q) + I}{q S_{ss}(\epsilon + g\tau_n) + 1} \\ S_{ss} &= \frac{-b + \sqrt{b^2 - 4ac}}{2a} \end{aligned} \quad (2.4)$$

where

$$\begin{aligned} a &= q(\epsilon + g\tau_n) \\ b &= q + g\tau_p [(1 - \beta)N_0q - \tau_n I] - \beta\epsilon\tau_p I \\ c &= -\beta\tau_p I \end{aligned}$$

2.2.2 Langevin sources

In real lasers however, due to the nature of the random carrier and photon recombination intrinsically present inside the laser's cavity, at equilibrium the laser states oscillates around their steady state values. This causes also variation in the laser's optical phase, being such a variation time-dependant on the carrier number deviation [25].

To model such randomness in the model, it is useful to introduce a noise source associate which each state, called Langevin noise, leading to a more complete schematic shown in Figure 2.2.

From a mathematical point of view, including those Langevin random process in the dynamic model transform an ODE into a system of stochastic differential equations (SDE) [14] which describes the time evolution of carriers, photons and the

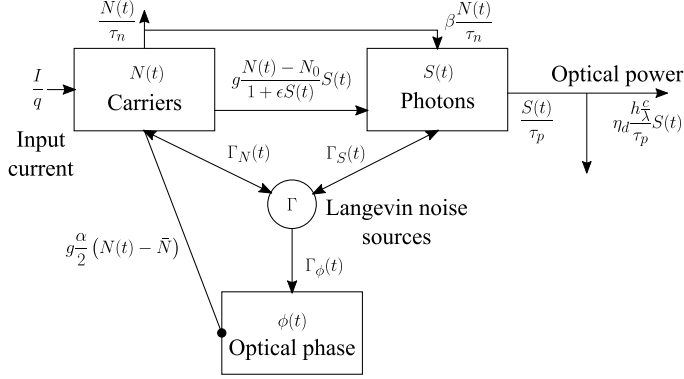


Figure 2.2: Complete schematic of the laser's dynamics.

variation of the optical phase $\phi(t)$

$$\frac{dN(t)}{dt} = \frac{I}{q} - \frac{N(t)}{\tau_n} - g \frac{N(t) - N_0}{1 + \epsilon S(t)} S(t) + \Gamma_N(t) \quad (2.5)$$

$$\frac{dS(t)}{dt} = g \frac{N(t) - N_0}{1 + \epsilon S(t)} S(t) - \frac{S(t)}{\tau_p} + \frac{\beta N(t)}{\tau_n} + \Gamma_S(t) \quad (2.6)$$

$$\frac{d\phi(t)}{dt} = \frac{\alpha}{2} g (N(t) - \bar{N}) + \Gamma_\phi(t) \quad (2.7)$$

(2.5) and (2.6) are an extension to (2.1) and (2.2) including the Langevin sources $\Gamma_N(t)$ and $\Gamma_S(t)$. (2.7) describes the time-variation in the optical phase according to the deviation of the carriers $N(t)$ from their time average $\bar{N} = \int_0^t N(\tau) d\tau / t$ which occurs in the steady state mode in presence of noise. α is a constant parameter called linewidth enhancement factor and $\Gamma_\phi(t)$ is the Langevin source associated with the phase. Since in this work a laser is analyzed during its steady state mode, \bar{N} is considered to be the steady state carrier number N_{ss} .

The Langevin sources are modelled as white noise with zero mean and power spectrum dependent on the instantaneous value of the system's states

$$\begin{aligned} \langle \Gamma_i(t) \rangle &= 0 \\ \langle \Gamma_i(t) \Gamma_{i'}(t') \rangle &= D_{ii'}(t) \delta(t - t') \end{aligned} \quad (2.8)$$

where $D_{ii'}$ is the correlation strength between the source i and the source i' , $\delta(\cdot)$ is the Kronecker delta and $\langle \cdot \rangle$ denotes a statistical average operation. In the laser

system, the autocorrelation strengths are defined as follows [14]

$$\begin{aligned} D_{NN}(t) &= \frac{2N(t)(1 + \beta S(t))}{\tau_n} \\ D_{SS}(t) &= \frac{2\beta N(t)S(t)}{\tau_n} \\ D_{\phi\phi}(t) &= \frac{\beta N(t)}{2\tau_n S(t)} \end{aligned} \quad (2.9)$$

and the crosscorrelation between the sources are

$$\begin{aligned} D_{SN}(t) &= -\frac{2\beta N(t)S(t)}{\tau_n} \\ D_{S\phi}(t) &= 0 \\ D_{N\phi}(t) &= 0 \end{aligned} \quad (2.10)$$

In other words, Langevin sources are modelled as additive white Gaussian noise to the system states which covariance matrix is time dependent on the state instantaneous values.

2.2.3 Measuring intensity and frequency noise

Because of the presence of noise sources during the steady-state operation, the output power magnitude and the optical phase are subject to random oscillation which can be quantified globally with two important measures for lasers: the relative intensity noise (RIN) and the frequency noise (FN).

The RIN quantifies the noise floor for the output power spectrum and it is calculated as below

$$\text{RIN}(f) = \frac{1}{\langle P \rangle^2} \left[\frac{1}{T} \left| \int_0^T \delta P(\tau) e^{-j2\pi f \tau} d\tau \right|^2 \right] \quad (2.11)$$

by taking the power spectrum of the instantaneous power fluctuations $\delta P(t) = P(t) - \langle P \rangle$ around the mean and dividing it by the squared mean of the power. Here the calculation are referred to a suited interval $[0, T]$ where laser states are assumed to be stationary.

The FN quantifies the noise affecting the optical phase. It is calculated as

$$\text{FN}(f) = \frac{1}{T} \left| \int_0^T \Delta\nu(\tau) e^{-j2\pi f \tau} d\tau \right|^2 \quad (2.12)$$

where $\Delta\nu(t) = (2\pi)^{-1} d\phi/dt$ is the fluctuation of the lasing frequency. In other words, RIN and FN are the power spectra of the power and frequency oscillations around the steady-state operation, caused by the presence of Langevin sources.

RIN and FN spectra are interesting parameters to compute for a laser, as discussed in Chapter 1. However, a measurement system cannot "read" directly access to the internal laser's states, therefore refereed as *hidden states*. All the information is accessible on the emitted light, or detected EM wave. As additional inherent challenge

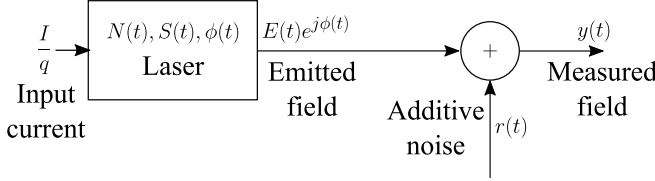


Figure 2.3: Basic schematic of a measurement system.

which occurs in a basic measurement system (Figure 2.3), the light detected from the laser differs from the emitted one because of the presence different noise sources between the laser and the detector. This is defined as *measurement noise* and is typically modelled as an additive i.i.d. Gaussian source with zero mean and σ_r^2 variance acting in the channel between source and detector. The measure becomes thus a function of the laser's states, which in phasor notation may be expressed as

$$y(t) := E(t)e^{j\phi(t)} + r(t) \quad (2.13)$$

The measurement equation (2.13) describes the measured field as a EM wave with amplitude $E(t)$ and phase $\phi(t)$, corrupted by the complex Gaussian noise $r(t)$. The amplitude of the electric field is linked to the power (2.3) by

$$E(t) = \sqrt{P(t)} \quad (2.14)$$

To simplify the dealing with complex number, using Euler's formula the measurement equation is typically split into real (*in-phase*) and imaginary (*quadrature*) part

$$\begin{aligned} y_i(t) &:= E(t) \cos(\phi(t)) + r_i(t) \\ y_q(t) &:= E(t) \sin(\phi(t)) + r_q(t) \end{aligned} \quad (2.15)$$

such that $y(t) = y_i(t) + jy_q(t)$. Without loss of generality, the two noise components are considered independent each other, both with the same variance

$$r_i(t), r_q(t) \sim \mathcal{N}(0, \sigma_r^2) \quad (2.16)$$

Since the measurements (2.13) are the only information directly available from the laser's dynamics, the naive way to estimate RIN and FN is to use the measurements directly. In the hypothesis of an high SNR,

$$\hat{P}(t) \approx |y(t)|^2 = [y_i(t)]^2 + [y_q(t)]^2 \quad (2.17)$$

$$\hat{\phi}(t) \approx \angle y(t) = \arctan \left(\frac{y_q(t)}{y_i(t)} \right) \quad (2.18)$$

(2.17) and (2.18) may be refereed as the *conventional* way to estimate power and phase, and subsequently use (2.11) - (2.12) to estimate RIN and FN. This approach is however inefficient when dealing with low SNR, as it will be seen in Section 2.4.

2.3 Numerical methods for SDEs

To simulate a laser system, it is necessary to integrate (2.5), (2.6) and (2.7). A stochastic system of differential equations differs from an ordinary system because of the presence of random processes that needs to be integrated. Recalling that each Langevin source i can be seen as a stochastic process whose instances are normally distributed with zero mean and $\sigma_{\Gamma_i}^2 = D_{ii}$ as variance,

$$\Gamma_i(t) \sim \mathcal{N}(0, D_{ii}(t)) \quad (2.19)$$

it is possible to decompose such sources as

$$\Gamma_i(t) = g_i(t) \cdot q_i(t) \text{ with } q_i(t) \sim \mathcal{N}(0, 1) \forall t \quad (2.20)$$

where $g_i(t) = \sqrt{D_{ii}(t)}$ is the *diffusion* term associated with the source i . Using the SDE terminology, (2.5) - (2.7) can therefore be written in the form of

$$d\mathbf{x}(t) = \mathbf{f}(\mathbf{x}(t))dt + [\mathbf{I}_3 \mathbf{g}(\mathbf{x}(t))] d\mathbf{q}(t) \quad (2.21)$$

where $\mathbf{x}(t) = [N(t), S(t), \phi(t)]^\top$ represent the system state, $\mathbf{f}(\cdot) : \mathbb{R}^3 \rightarrow \mathbb{R}^3$ incorporates the drift terms and $\mathbf{g}(\cdot) : \mathbb{R}^3 \rightarrow \mathbb{R}^3$ the diffusion terms. Both can be easily identified in (2.5) - (2.7) by setting

$$\mathbf{f} \begin{pmatrix} N(t) \\ S(t) \\ \phi(t) \end{pmatrix} := \begin{pmatrix} I/q - N(t)/\tau_n - g(N(t) - N_0)S(t)/(1 + \epsilon S(t)) \\ g(N(t) - N_0)S(t)/(1 + \epsilon S(t)) - S(t)/\tau_p + \beta N(t)/\tau_n \\ \alpha g(N(t) - N_{ss})/2 \end{pmatrix} \quad (2.22)$$

$$\mathbf{g} \begin{pmatrix} N(t) \\ S(t) \\ \phi(t) \end{pmatrix} := \begin{pmatrix} \sqrt{2N(t)(1 + \beta S(t))/\tau_n} \\ \sqrt{2\beta N(t)S(t)/\tau_n} \\ \sqrt{\beta N(t)/(2\tau_n S(t))} \end{pmatrix} \quad (2.23)$$

To be consistent with the multivariate product rules, \mathbf{I}_3 is a 3×3 identity matrix and $\mathbf{q}(t) \in \mathbb{R}^3$ is the stochastic vector

$$\mathbf{q}(t) := \begin{pmatrix} q_N(t) \\ q_S(t) \\ q_\phi(t) \end{pmatrix} \quad (2.24)$$

In this way the sources $q_i(t)$ can be identified as *Wiener processes* or *Brownian motions*, for which a consistent theory regarding their integration in SDE using Itô integrals has been developed in stochastic calculus [26, 27]. Having the system in the form

of (2.21) makes easier to apply numerical methods for integration, which will be discussed below.

Solving an SDE given the Wiener process $\mathbf{q}(t)$ and the starting point $\mathbf{x}(0)$ means to find a trajectory $\mathbf{x}(t)$ in the interval $[0, T]$ which satisfies (2.21). For simpler cases, analytic solutions do exist and can be computed in closed form, but for the rate equation model (2.5) - (2.7) it is necessary to use numerical approximations. Solving (2.21) using *numerical methods* requires to approximate the process $\mathbf{x}(t)$ with a discrete-time version \mathbf{x}_n calculated on successive $N_T + 1$ regularly spaced mesh points $\{t_n, n = 0 \dots N_T + 1 : t_0 < t_1 < t_2 < \dots < t_{N_T}\}$, using a mapping function $\Phi(\cdot)$

$$\mathbf{x}_{n+1} = \Phi(\mathbf{x}_n, \Delta t, \Delta \mathbf{q}_n) \quad (2.25)$$

which iteratively calculates the next point \mathbf{x}_{t+1} using the approximation of the previous one \mathbf{x}_t . $\Delta t := t_{n+1} - t_n \forall n \geq 0$ is a fixed timestep and $\Delta \mathbf{q}_n := \mathbf{q}(t_{n+1}) - \mathbf{q}(t_n)$ are independent Wiener increments. Moreover, it follows from the properties of a Wiener process that its increments are i.i.d normally distributed $\Delta \mathbf{q}_t \sim \mathcal{N}(0, \Delta t)$.

2.3.1 Euler-Maruyama method

One of the simplest to integrate (2.21) is to adopt a linear approximation of the integrand functions. By integrating between t_n and t_{n+1}

$$\mathbf{x}(t_{n+1}) = \mathbf{x}(t_n) + \int_{t_n}^{t_{n+1}} \mathbf{f}(\mathbf{x}(\tau)) d\tau + \int_{t_n}^{t_{n+1}} [\mathbf{I}_3 \mathbf{g}(\mathbf{x}(\tau))] d\mathbf{q}(\tau) \quad (2.26)$$

the *Riemann* integral on the left can be approximated, using the *trapezoid rule*

$$\int_{t_n}^{t_{n+1}} \mathbf{f}(\mathbf{x}(\tau)) d\tau \approx \mathbf{f}(\mathbf{x}(t_n))(t_{n+1} - t_n) \approx \mathbf{f}(\mathbf{x}_n) \Delta t \quad (2.27)$$

while the stochastic integral on the right, with a slight abuse of notation, yields to a similar result

$$\int_{t_n}^{t_{n+1}} [\mathbf{I}_3 \mathbf{g}(\mathbf{x}(\tau))] d\mathbf{q}(\tau) \approx [\mathbf{I}_3 \mathbf{g}(\mathbf{x}(t_n))] (\mathbf{q}(t_{n+1}) - \mathbf{q}(t_n)) \approx [\mathbf{I}_3 \mathbf{g}(\mathbf{x}_n)] \Delta \mathbf{q}_n \quad (2.28)$$

The last approximations in (2.27) - (2.28) compute the drift and diffusion processes in the previous approximated points. This is the key approach of the Euler-Maruyama (EM) method, described in Algorithm 1.

In a calculator, a numerical solution is simulated by generating normal distributed pseudo-random numbers for the stochastic part and propagate them, together with successive iterations of the states using (2.29). Example of solutions for the rate equation model can be seen in Section 2.4.

While EM method is simple and effective, it has to deal with the trade-off of the Δt choice. A small value reduces the error between the numerical solution and

Algorithm 1 EM - Euler-Maruyama numerical solver

Initialize. Select a starting point $\mathbf{x}_0 \in \mathbb{R}^d$ and a timestep Δ_t

for $n = 0$ to N_T **do**

 Compute the next mesh point approximation as

$$\mathbf{x}_{n+1} = \mathbf{x}_n + \mathbf{f}(\mathbf{x}_n)\Delta_t + [\mathbf{I}_d \mathbf{g}(\mathbf{x}_n)] \Delta \mathbf{q}_n \quad (2.29)$$

end for

return $\mathbf{x}_{0:N_T}$

the analytic one, but increases the number of calculations in the same time interval. On the other hand, a larger value will reduce the number of calculations at the price of having an higher approximation error [28]. Another problem that some non-linear SDE can suffer when solved using standard numerical methods is *stiffness* [29]. Stiff problems have the peculiarity that the convergence and stability of a numerical solution is not guaranteed for arbitrary choice of Δ_t . This is indeed the case for the rate equation model when using a larger timestep, which will be shown in Section 2.4. Therefore, whereas Δ_t cannot be freely chosen, a stabler numerical method is introduced in the next section.

2.3.2 S-ROCK methods

The Îto S-ROCK methods [30] are a family of methods derived from the Chebyshev methods [31], which use m_{sr} different stages to approximate the next point \mathbf{x}_{n+1} using Chebyshev polynomials $T_j(\cdot)$ and include a stochastic finishing procedure to incorporate the diffusion part into the solution of the SDE system. The method uses also η_{sr} as a damping parameter. Both of m_{sr} and η_{sr} modify the convergence region of a numerical method. For the rate-equations, default values of $m_{\text{sr}} = 10$ and $\eta_{\text{sr}} = 100$ were adopted prior verifying the convergence for a given timestep. No investigation have been made regarding the stability of a numerical solution and the role of the S-ROCK parameters, which may be left for future research. The method is presented in Algorithm 2.

This method requires much more function evaluation, which scales linearly with the number of stages. However, it guarantees a larger stability with larger Δ_t than EM, (see Figure 2.7). Therefore, in cases where the timestep can be freely be changed, EM methods is advisable. In other cases, S-ROCK will guarantee more stability at the price of a lower speed. Both Algorithms 1 and 2 are one-step methods, meaning that they approximate the solution recursively by employing the value previously calculated. This characteristic will be exploited in Chapter 3, where the Markovianity evolution of the states will be required to implement discrete-time state space models employed in Bayesian filtering algorithms.

Algorithm 2 S-ROCK - Ito S-ROCK method [30]

Initialize. Select a starting point $\mathbf{x}_0 \in \mathbb{R}^d$, a timestep Δ_t , a damping parameter η_{sr} and the number of Chebyshev stages m_{sr}

$$w_0 = 1 + \eta_{\text{sr}}/m_{\text{sr}}^2$$

$$w_1 = T_{m_{\text{sr}}}(w_0)/T'_{m_{\text{sr}}}(w_0)$$

for $n = 0$ to N_T **do**

Chebyshev deterministic stages

$$\mathbf{k}_0 = \mathbf{x}_n \tag{2.30}$$

$$\mathbf{k}_1 = \mathbf{x}_n + \Delta_t \frac{w_0}{w_1} \mathbf{f}(\mathbf{k}_0) \tag{2.31}$$

for $j = 2$ to m_{sr} **do**

$$\mathbf{k}_j = 2\Delta_t w_1 \frac{T_{j-1}(w_0)}{T_j(w_0)} \mathbf{f}(\mathbf{k}_{j-1}) + 2w_0 \frac{T_{j-1}(w_0)}{T_j(w_0)} \mathbf{k}_{j-1} - \frac{T_{j-2}(w_0)}{T_j(w_0)} \mathbf{k}_{j-2} \tag{2.32}$$

end for

$$\mathbf{k}_{m_{\text{sr}}-1}^* = \mathbf{k}_{m_{\text{sr}}-1} + \sqrt{\Delta_t} \mathbf{g}(\mathbf{k}_{m_{\text{sr}}-1}) \tag{2.33}$$

Finishing stochastic stage

$$\mathbf{x}_{n+1} = \mathbf{k}_{m_{\text{sr}}} + (\mathbf{I}_d \Delta \mathbf{q}_t) \mathbf{g}(\mathbf{k}_{m_{\text{sr}}-1}) + \frac{\mathbf{I}_d}{2\sqrt{\Delta_t}} (\mathbf{g}(\mathbf{k}_{m_{\text{sr}}-1}^*) - \mathbf{g}(\mathbf{k}_{m_{\text{sr}}-1})) (\Delta \mathbf{q}_n^2 - \Delta_t) \tag{2.34}$$

end for

return $\mathbf{x}_{0:N_T}$

2.4 System simulation

The simulation of the system is necessary not only to validate the proposed method on simulated data, but also to give useful tools when it is necessary to track the laser internal states (see Chapter 3). MATLAB® was the simulation environment used for testing and validating the methods.

The static parameters $\boldsymbol{\theta}$ used for simulating a laser are presented in Table 2.1. Because of the free choice of Δ_t in a simulated setting, the EM numerical method was used, since faster and stable with $\Delta_t = 1$ ps. Those parameters are commonly used during different simulation tests and they do not change unless explicitly denoted. For the sake of notation, let t be a discrete time index. Some attention may require to assure N_t and S_t are positive quantities and do not lead any singularity when propagated inside a square root. However if the numerical method is stable, by initialize the system in the steady state condition (2.4) positivity of the solution is

Algorithm 3 Laser simulation

Initialize. Select a starting point \mathbf{x}_0 , a timestep Δ_t and a vector of static parameter $\boldsymbol{\theta}$
for $t = 0$ to N_T **do**
 Generate the process noise

$$\Delta \mathbf{q}_n \sim \mathcal{N}(\mathbf{0}, \Delta_t \mathbf{I}_3) \quad (2.35)$$

Use $\Delta \mathbf{q}_n$ and \mathbf{x}_n to compute the next mesh point approximation using Algorithm 1 or Algorithm 2

Generate the measurement noise

$$\mathbf{r}_n \sim \mathcal{N}(\mathbf{0}, \sigma_r^2 \mathbf{I}_2) \quad (2.36)$$

Use \mathbf{r}_n and \mathbf{x}_n to compute the measurement using (2.37)
end for
return $\mathbf{x}_{0:N_T}$, $\mathbf{y}_{0:N_T}$

guaranteed. For simulating also a set of sampled measurements $\mathbf{y}_{0:N_T-1}$, a discrete-time version of (2.15) is employed, i.e.

$$\mathbf{y}_t = \sqrt{\eta_d \frac{hc}{\lambda \tau_p} S_t} \begin{bmatrix} \cos(\phi_t) \\ \sin(\phi_t) \end{bmatrix} + \begin{bmatrix} r_{i,t} \\ r_{q,t} \end{bmatrix} \quad (2.37)$$

For calculating (2.11) - (2.12) on discrete sequences, the tool used is the fast Fourier transform (FFT) ready available on MATLAB®. Given the discrete sequences P_t and ϕ_t of N_T points s.t. $\Delta_t N_T = T$

$$\text{RIN}_f = \frac{1}{\langle P \rangle^2 \Delta_t N_T} \left| \sum_{k=0}^{N_T-1} (P_k - \langle P \rangle) e^{-j\omega k / N_T} \Delta_t \right|^2 \quad (2.38)$$

$$\text{FN}_f = \frac{1}{\Delta_t N_T} \left| \sum_{k=0}^{N_T-1} \Delta \nu_k e^{-j\omega k / N_T} \Delta_t \right|^2 \quad (2.39)$$

where

$$\langle P \rangle = \frac{1}{N_T} \sum_{k=0}^{N_T-1} P_k \quad (2.40)$$

$$\Delta \nu_t = \frac{1}{2\pi} \frac{\phi_t - \phi_{t-1}}{\Delta_t} \quad (2.41)$$

The timestep used for the model determines the maximum resolution in frequency, while the number of samples used for the Fourier transform fix the resolution for each

Parameter	Name	Value
τ_n	Carrier lifetime	$0.33 \cdot 10^{-9}$
τ_p	Photon lifetime	$7.15 \cdot 10^{-12}$
g	Gain slope	$1.13 \cdot 10^4$
β	Spontaneous recombination fraction	$3.54 \cdot 10^{-5}$
ϵ	Non-linear gain compression	$4.58 \cdot 10^{-8}$
N_0	Carriers at transparency	$8.2 \cdot 10^6$
α	Linewidth enhancement factor	$4.55 \cdot 10^0$
I	Input current	$1.29 \cdot 10^{-2}$
η_d	Quantum efficiency	$2.1 \cdot 10^{-1}$
λ	Optical wavelength	$1.55 \cdot 10^{-6}$

Table 2.1: Common parameters used for a laser simulation.

bin. The considered interval for calculating RIN and FN has to be within the steady state regime.

The first test that is going to be presented is a simulation of the SDE trajectories for carriers and photons, shown in Figure 2.4. As it can be noticed, the time evolution goes toward a potential well, the steady state solution (N_{ss}, S_{ss}). However, because of the Langevin noises, the solution gets perturbed and both drift (2.22) and diffusion (2.23) terms force the trajectory to oscillates around the equilibrium. In Figure 2.5 is shown in details the sequence after reaching the equilibrium, highlighting even more the harmonic behaviour of the system. The optical phase, on the way it is defined, does not have a steady state therefore its trajectory deviates from ϕ_0 to increase in absolute value.

As mentioned in Section 2.3.1, the stiffness in the rate equation model arises using simple numerical methods such as EM. In Figure 2.6, it is possible to see how changes in Δ_t have severe consequences on the overall numerical solution. For larger values, the evolution do not oscillates anymore around the equilibrium and diverges. In Figure 2.7 is shown a comparison between the EM method and the S-ROCK method, comparing the evolution of N_t . While the former for larger timesteps starts to diverge from the equilibrium, the latter keeps the solution stable and stationary. The limit of both numerical methods were not tested exactly, but the S-ROCK methods works fine even in the order of nanoseconds (see Chapter 5).

The final figures shows the computation of RIN and FN on the simulated states. This will be the reference for the *true* spectra, being compared with the filtering estimate (Chapter 3) when is necessary to assess the quality of the tracking. In Figure 2.8 is shown the results of using (2.38) and (2.39) on the generated sequences $P_{0:N_T-1}$ and $\phi_{0:N_T-1}$ using a numerical solver. Since RIN/FN spectra would be computed over a sequence affected by random noise, it is usually necessary to average different experiments in order to find the envelope. This approach, although more correct can be slow when for each sequence a sampling and tracking algorithm are applied. Therefore, the all the computed/estimated spectra are lowpass filtered using a moving

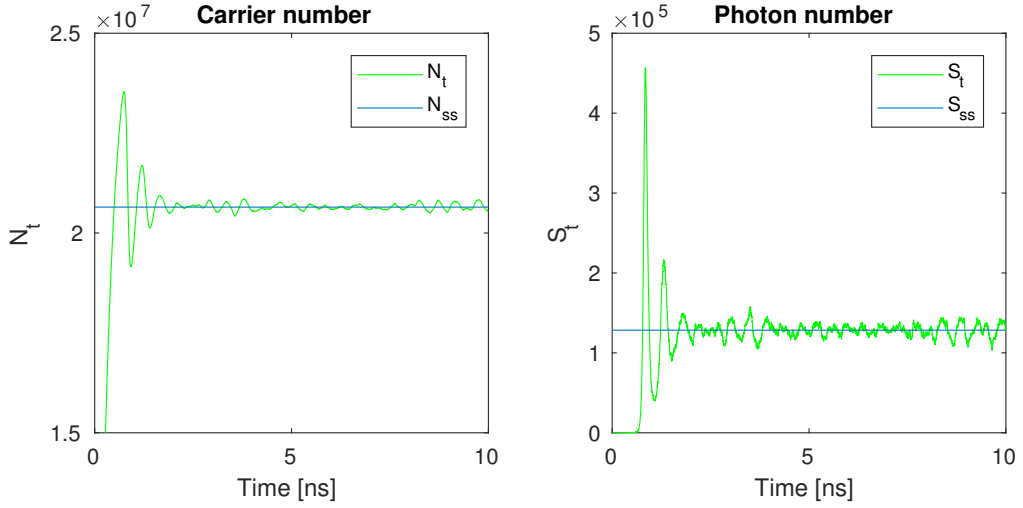


Figure 2.4: Simulation of the laser's states using MATLAB®. Both states start from zeros and converge toward the steady state solution.

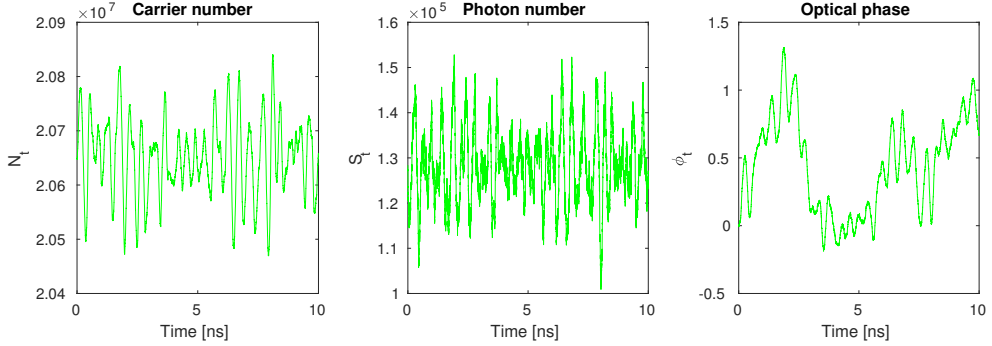


Figure 2.5: Simulation of the laser's states using MATLAB®.

average filter, leading to qualitatively good and faster results. To see how effective is the conventional way to estimate RIN and FN from the received measurements using (2.17) and (2.18), different conventional estimate for different level of measurement noise are plotted and compared with the true reference in Figure 2.9. It immediately appears that the method fails in presence of low SNR. Moreover, the effect of Gaussian noise in the channel becomes not white anymore when propagated through (2.17) - (2.18), explaining then why the shape of the conventional does not look flat.

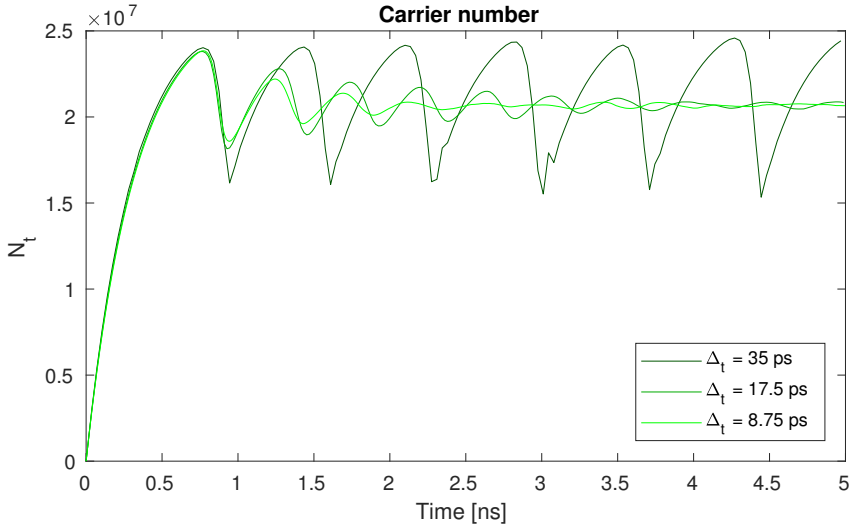


Figure 2.6: Simulation of N_t states using MATLAB®, using EM method with different Δ_t .

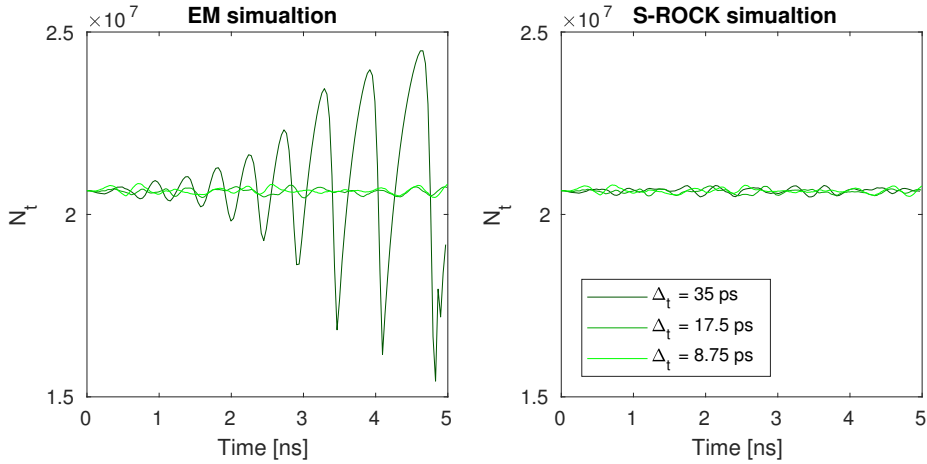


Figure 2.7: Side to side comparison between EM and S-ROCK numerical solvers during steady state regime.

It has been shown how to accurately simulate a laser model. This method will

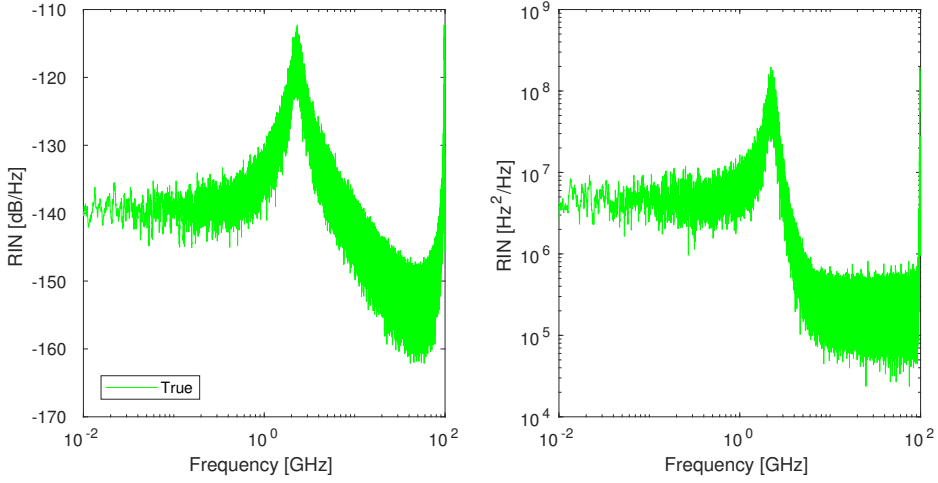


Figure 2.8: True RIN/FN spectra computed on simulated states.

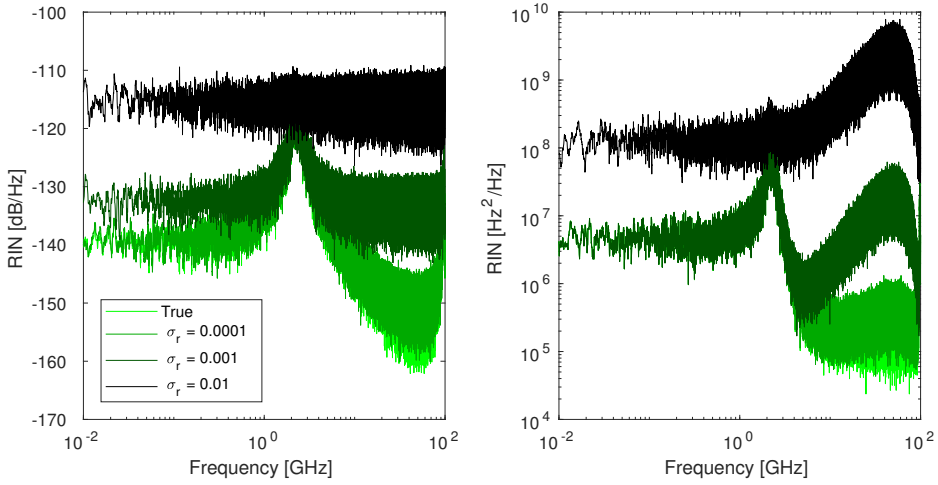


Figure 2.9: Comparison between the true RIN/FN spectra and the one extracted from the noisy measurements, with different noise levels.

be used not only to perform experiments on simulated data but also to implement discrete-time state-space models in the filtering algorithms. These filtering techniques

will be introduced in Chapter 3 and will help to filter out the measurement noise when dealing with low SNR settings, significantly outperform the conventional way.

CHAPTER 3

Tracking in state space models

3.1 Introduction

As discussed in Chapter 2, in presence of low SNR, using the measurements directly to extract optical power and phase does not give a good estimation of RIN and FN. The measurement noise is non-informative and un-desired and for this reason, it is necessary to filter it out from the observed data to achieve a better characterization of the spectra. A very useful framework to for noise filtering in dynamic models is identified in the *Bayesian filtering* theory [32, 33]. The key idea behind *filtering* is to estimate the state dynamic evolution based on the received noisy data, given some assumptions and knowledge of the model. The term *Bayesian* refers instead to the Bayesian approach to model the underlying stochastic phenomena and to compute the best state estimate.

3.2 From deterministic to probabilistic state-space models

To start with, it is necessary to set up a laser dynamic model into a *state-space model* (SSM) . Such model completely describe the system evolution from time $t \geq 0$ given the current state \mathbf{x}_t and the dynamic/measurement functions

$$\mathbf{x}_{t+1} = \mathbf{f}(\mathbf{x}_t, \mathbf{q}_t) \quad (3.1)$$

$$\mathbf{y}_t = \mathbf{h}(\mathbf{x}_t, \mathbf{r}_t) \quad (3.2)$$

where $\mathbf{f}(\cdot) : \mathbb{R}^{d_{\mathbf{x}}+d_{\mathbf{q}}} \rightarrow \mathbb{R}^{d_{\mathbf{x}}}$ is the deterministic process function, $\mathbf{h}(\cdot) : \mathbb{R}^{d_{\mathbf{x}}+d_{\mathbf{r}}} \rightarrow \mathbb{R}^{d_{\mathbf{y}}}$ is the deterministic measurement function, $\mathbf{x}_t \in \mathbb{R}^{d_{\mathbf{x}}}$ is the system hidden state, $\mathbf{y}_t \in \mathbb{R}^{d_{\mathbf{y}}}$ is the measurement available, $\mathbf{q}_t \in \mathbb{R}^{d_{\mathbf{q}}}$ is the process noise and $\mathbf{r}_t \in \mathbb{R}^{d_{\mathbf{r}}}$ is the measurement noise. All these vector values quantities are part of a time series dependent on the discrete temporal index t .

These equations are completely general and depends on the assumption of the underlying model. From the physics of the rate-equation, the system dynamics (3.1)

can be identified in (2.29) or (2.34) whether EM or S-ROCK method is preferred and (3.2) can be identified in (2.37). In alternative, a simpler linear model can be used instead with implications discussed in Section 3.5.2. On the form of (3.1) - (3.2), $\mathbf{f}(\cdot)$ and $\mathbf{h}(\cdot)$ may depend also on the current temporal index t , the current used Δ_t and/or the specific static parameters $\boldsymbol{\theta}$ used. For notation simplicity, additional dependency on extra variables will be denoted if necessary. In this chapter, the static parameters $\boldsymbol{\theta}$ are considered know a priori.

The Bayesian filtering theory assumes that the dynamical system in exams has to be a *probabilistic state-space model* PSSM, in which both the process and measurement models are described by conditional probability density distributions,

$$\mathbf{x}_{t+1} \sim p(\mathbf{x}_{t+1}|\mathbf{x}_t) \quad (3.3)$$

$$\mathbf{y}_t \sim p(\mathbf{y}_t|\mathbf{x}_t) \quad (3.4)$$

Where (3.3) describes the stochastic distribution of the system states and (3.4) the distribution of the measurements given the current state. This description suffices to define the entire model when two additional assumptions are considered. The first assumption is the Markovian property of the states

$$p(\mathbf{x}_{t+1}|\mathbf{x}_{0:t}, \mathbf{y}_{1:t}) = p(\mathbf{x}_{t+1}|\mathbf{x}_t) \quad (3.5)$$

meaning that the state dynamics depends only on the previous state. Now it is clear why using one-step methods presented in Chapter 2 preserve the Markovianity assumption of the model. The second basic assumption is the conditional independence of measurements

$$p(\mathbf{y}_t|\mathbf{x}_{0:t}, \mathbf{y}_{1:t-1}) = p(\mathbf{y}_t|\mathbf{x}_t) \quad (3.6)$$

meaning that the current measurement depends only on the current state. This implies that the measurements are correlated indirectly via the hidden states. These two assumptions, and the probabilistic model (3.3) - (3.4) are closely related to the theory developed for *Hidden Markov Models* HMMs [34]. However, the main difference is that HMMs consider a discretization of the state space, while here the dynamic states take values in a subset of $\mathbb{R}^{d_{\mathbf{x}}}$. The obtained PSSM can also be described using *graphical modelling theory* [16], for which an example is presented in Figure 3.1.

3.3 Bayesian filtering equations

Having stated the model (3.3) - (3.4) what follows is the Bayesian filtering algorithm [32], which provide the necessary equations to compute the *predicted distribution* $p(\mathbf{x}_t|\mathbf{y}_{1:t-1})$ and the *filtering distribution* $p(\mathbf{x}_t|\mathbf{y}_{1:t})$, in other words the probability distribution functions (PDF) of the dynamic states before and after observing a new measurement. Algorithm 4 basically iterates, as long new measurement from the sequence are provided, two basics steps. The prediction step (3.7) uses the previous

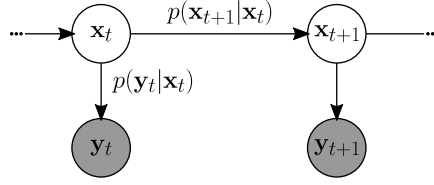


Figure 3.1: Graphical model representation of the PSSM. The hidden states are linked in a chain fashion, where the arrows indicate the conditional dependence relations. Shaded nodes are observed variables.

Algorithm 4 General Bayesian Filtering

Initialize. Assume a prior distribution on the states $p(\mathbf{x}_0)$, a probabilistic process(3.3) and measurement model (3.4).

for $t = 1$ to N_T **do**

Prediction step. Given the previous updated posterior $p(\mathbf{x}_{t-1}|\mathbf{y}_{1:t-1})$, compute the predictive distribution as

$$p(\mathbf{x}_t|\mathbf{y}_{1:t-1}) = \int p(\mathbf{x}_t|\mathbf{x}_{t-1})p(\mathbf{x}_{t-1}|\mathbf{y}_{1:t-1})d\mathbf{x}_{t-1} \quad (3.7)$$

Update step. Using the predicted distribution and the new measurement \mathbf{y}_t , compute the filtering distribution as

$$p(\mathbf{x}_t|\mathbf{y}_{1:t}) = \frac{p(\mathbf{y}_t|\mathbf{x}_t)p(\mathbf{x}_t|\mathbf{y}_{1:t-1})}{\int p(\mathbf{y}_t|\mathbf{x}_t)p(\mathbf{x}_t|\mathbf{y}_{1:t-1})d\mathbf{x}_t} \quad (3.8)$$

end for

return $p(\mathbf{x}_t|\mathbf{y}_{1:t})$ for $t = 1 \dots N_T$

filtered distribution plus the probabilistic dynamic model to predict the distribution of the next state. It then employs the Bayesian theorem (3.8) to update the knowledge on the current state given the new measurement $p(\mathbf{x}_t|\mathbf{y}_{1:t})$, by multiplying the likelihood $p(\mathbf{y}_t|\mathbf{x}_t)$ with the predictive distribution $p(\mathbf{x}_t|\mathbf{y}_{1:t-1})$ (acting like a prior) and normalize the result.

The advantage of the Bayesian approach is that it resolves the temporal sequence of hidden states not only by returning a point estimate, but the whole posterior PDF after the observations $p(\mathbf{x}_t|\mathbf{y}_{1:N_T})$. From this, a point estimate can be extracted, for example the mean

$$\mathbf{m}_t := \mathbb{E}[\mathbf{x}_t|\mathbf{y}_{1:t}] \quad (3.9)$$

or the uncertainty, in the form of covariance matrix

$$\mathbf{P}_t := \mathbb{E} [(\mathbf{x}_t - \mathbf{m}_t)(\mathbf{x}_t - \mathbf{m}_t)^\top | \mathbf{y}_{1:t}] \quad (3.10)$$

The problems in this approach involves not only to identify the PSSM given the SSM, but also to working out (3.7) - (3.8): usually a closed form solution does not exists, therefore numerical integration is required. However, numerical integration in high dimension is intractable. This issue is part of a common problem involving high-dimensional spaces, called *the curse of dimensionality* [35]. Integrals involved here are not restricted to a small intervals, but they span over all the state-space. Some methods deal with this by approximating integrals as a weighted sum of particles, i.e. *Particle filters* [36]. Drawback of these methods is the computational time spent to propagate the particles and estimate the probability distribution and re-sample on the new PDF [37]. Luckily, with some additional assumptions, the Bayesian filtering equations can be solved or approximated in close forms. The Kalman filtering theory provides useful tool to approximate in close form (3.7) - (3.8) and find the PSSMs starting from generic and non-linear SSM.

3.4 Kalman filtering theory

The elegant theory of Kalman filtering [32] solves the multidimensional integrals of (3.7) and (3.8) in close form on a special case. The Kalman filter belongs to the family of Gaussian filters [38], where non-linear filtering problems used Gaussian density approximations. On the hypothesis that the prior distribution of the states is Gaussian $p(\mathbf{x}_0) \sim \mathcal{N}(\mathbf{m}_0, \mathbf{P}_0)$, the dynamic and measurement models are linear and \mathbf{q}_t and \mathbf{r}_t are i.i.d Gaussian distributed, then the predictive and filtering distribution share also the same PDF,

$$p(\mathbf{x}_t | \mathbf{y}_{1:t-1}) \sim \mathcal{N}(\mathbf{m}_t^-, \mathbf{P}_t^-) \quad (3.11)$$

$$p(\mathbf{x}_t | \mathbf{y}_{1:t}) \sim \mathcal{N}(\mathbf{m}_t, \mathbf{P}_t) \quad (3.12)$$

This is a direct consequence of linear transformation and integration of Gaussian distributed random variables [39]. The Kalman filtering algorithm provides equations to compute the posterior mean and covariance, both before $(\mathbf{m}_t^-, \mathbf{P}_t^-)$ and after $(\mathbf{m}_t, \mathbf{P}_t)$ the update step. Least but not last, it identifies the PSSM in (3.11) and (3.12).

However, for the model in (3.1) - (3.2) this is not the case. The only ingredient missing is that \mathbf{f} and \mathbf{h} are not linear. Therefore, in the next sections are presented some extension of the usual Kalman filter which allows to exploit the same filtering structure by mean of some approximations.

3.4.1 Extended Kalman filter

When a Gaussian r.v. undergoes a non-linear transformation, the resulting distribution is not Gaussian anymore. The key idea for the extended Kalman filter (EKF) is

to use Taylor expansion to linearize the system which makes the transformed variable Gaussian as well. At each time instant t the system is linearized around the current mean estimate. For example, the approximation of \mathbf{f} around the current mean is the following

$$\mathbf{f}(\mathbf{x}_t, \mathbf{q}_t) \approx \mathbf{f}(\mathbf{m}_{\mathbf{x},t}, \mathbf{m}_{\mathbf{q},t}) + \mathbf{J}_{\mathbf{f}}(\mathbf{m}_{\mathbf{x},t}, \mathbf{m}_{\mathbf{q},t}) \cdot \begin{bmatrix} \mathbf{x}_t - \mathbf{m}_{\mathbf{x},t} \\ \mathbf{q}_t - \mathbf{m}_{\mathbf{q},t} \end{bmatrix} \quad (3.13)$$

Where $\mathbf{J}_{\mathbf{f}}$ is the Jacobian of \mathbf{f} . This can be further decomposed in two block matrices,

$$\mathbf{J}_{\mathbf{f}} = [\mathbf{J}_{\mathbf{f},\mathbf{x}}, \mathbf{J}_{\mathbf{f},\mathbf{q}}] \quad (3.14)$$

where $\mathbf{J}_{\mathbf{f},\mathbf{x}}$ and $\mathbf{J}_{\mathbf{f},\mathbf{q}}$ are the Jacobians w.r.t. \mathbf{x} and \mathbf{q} respectively. When linearizing the system, is possible to provide a close form solution for the Bayesian filtering equations, exposed in Algorithm 5.

Algorithm 5 EKF. Extended Kalman filter

Initialize. Assume an initial mean \mathbf{m}_0 , an initial covariance \mathbf{P}_0 , process (3.1) and measurement (3.2) models.

for $t = 1$ to N_T **do**

Prediction step. Compute the predicted posterior (3.11) by explicit calculating its mean and covariance

$$\mathbf{m}_t^- = \mathbf{f}(\mathbf{m}_{t-1}, \mathbf{0}) \quad (3.15)$$

$$\mathbf{P}_t^- = \mathbf{J}_{\mathbf{f},\mathbf{x}}(\mathbf{m}_{t-1}, \mathbf{0})\mathbf{P}_{t-1}\mathbf{J}_{\mathbf{f},\mathbf{x}}^\top(\mathbf{m}_{t-1}, \mathbf{0}) + \mathbf{J}_{\mathbf{f},\mathbf{q}}(\mathbf{m}_{t-1}, \mathbf{0})\mathbf{Q}\mathbf{J}_{\mathbf{f},\mathbf{q}}^\top(\mathbf{m}_{t-1}, \mathbf{0}) \quad (3.16)$$

Update step. Compute the filtered posterior (3.12) by explicit calculating its mean and covariance

$$\boldsymbol{\mu}_t = \mathbf{h}(\mathbf{m}_t^-, \mathbf{0}) \quad (3.17)$$

$$\mathbf{e}_t = \mathbf{y}_t - \boldsymbol{\mu}_t \quad (3.18)$$

$$\mathbf{S}_t = \mathbf{J}_{\mathbf{h},\mathbf{x}}(\mathbf{m}_t^-, \mathbf{0})\mathbf{P}_t^-\mathbf{J}_{\mathbf{h},\mathbf{x}}^\top(\mathbf{m}_t^-, \mathbf{0}) + \mathbf{J}_{\mathbf{h},\mathbf{r}}(\mathbf{m}_t^-, \mathbf{0})\mathbf{R}\mathbf{J}_{\mathbf{h},\mathbf{r}}^\top(\mathbf{m}_t^-, \mathbf{0}) \quad (3.19)$$

$$\mathbf{G}_t = \mathbf{P}_t^-\mathbf{J}_{\mathbf{h},\mathbf{r}}^\top(\mathbf{m}_t^-, \mathbf{0})\mathbf{S}_t^{-1} \quad (3.20)$$

$$\mathbf{m}_t = \mathbf{m}_t^- + \mathbf{G}_t\mathbf{e}_t \quad (3.21)$$

$$\mathbf{P}_t = \mathbf{P}_t^- - \mathbf{G}_t\mathbf{S}_t\mathbf{G}_t^\top \quad (3.22)$$

end for

return $(\mathbf{m}_t, \mathbf{P}_t)$ for $t = 1 \dots N_T$

The advantages of EKF are its simplicity and speed, compare to other filters which have to deal with non-linearities. In fact it reduces the Bayesian filtering steps using matrix multiplications, with the exception of the calculation of \mathbf{S}_t^{-1} . However, to linearize the system it requires the computation of derivatives, which restrict the

class of models that can be used in the filtering equations: they must be differentiable. In addition, a linear approximation sometimes does not represent a good approximation, such as in systems presenting high non-linearities. As long as (3.1) is differentiable, there are no limits on the Taylor expansion used in (3.13). As denoted in [32], to increase the degree of precision EKF can be generalized by expanding the function second-order polynomials. The drawback is the complexity of the algorithm increasing, and in general the gain in precision is not appreciable. In the next section, another derivative-free approach will be presented.

3.4.2 Unscented Kalman filter

The linearization approach of the system adopted in EKF makes possible to obtain the mean and covariance of the transformation in close form. Unscented Kalman filter (UKF), instead, selects a deterministic set of point called *sigma-points*, which are propagated through the non-linearity while mean and covariance are re-estimated after the transformation. This procedure is called *unscented transform* [40], and it may resemble the way particle filters are working. However the difference is that the sigma-set is chosen deterministically in order to match the first two moments of the initial distribution. In d dimension, those points corresponds the mean value plus the intersection of a d -hypersphere with the d axis, making in total $2d + 1$ points. This value is much less than the number of particles typically used in PF, making UKF more fast in propagating the points. Moreover, the transformed r.v. is approximated Gaussian with mean and covariance calculated from the transformed weighted points. Building a sigma-set is necessary to perform the prediction and update step of the Unscented Kalman filter. There are 3 free parameters, $\alpha_{\text{ukf}}, \beta_{\text{ukf}}$ and κ_{ukf} which regulate the scaling of the sigma-set. Here, without focusing too much on the tracking performance impact, some common values are adopted $\alpha_{\text{ukf}} = 10^{-3}, \beta_{\text{ukf}} = 2$ and $\kappa_{\text{ukf}} = 0$. Further details about different families of UKF are explained in [41]. The full UKF algorithm is presented in Algorithm 8, and Algorithm 7 is for building the sigma-set. Both of them are found in Appendix A.

Figure 3.2 shows why in some cases UKF is preferable over EKF. This is the case with high non-linearities, where the estimate of the covariance after a non-linear transform is much more accurate and closely match the covariance calculated in all propagated points (typical approach of a PF). The transformed covariance estimated with EKF fails to capture the structure of the transformed data, making it less accurate. However, when speed is more important, UKF is considerably slower having to propagate more points than EKF and to calculate the square root of a matrix for building the sigma-sets.

3.5 RIN and FN estimation

The advantage of using a filter for RIN/FN estimation is to filter out the noise from the measurements, to achieve an estimation closer to the ground truth. Given a set

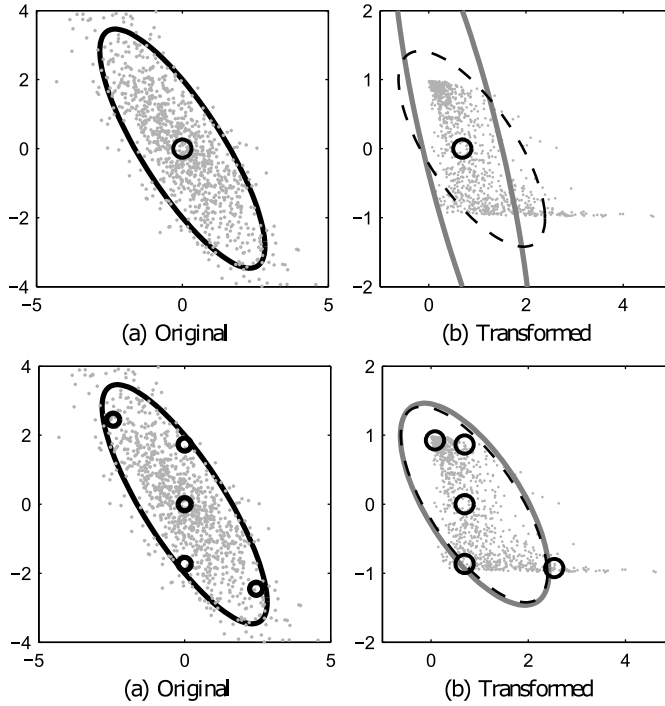


Figure 3.2: Advantages of the UKF (bottom row) over EKF (top row) [32]. The gray points are the particles before (left) and after (right) a non linear transform. The circles are the point propagated each method, while the ellipses are the covariances estimated for each method. The dashed ellipse is the covariance estimated on all particles.

of measurements from a laser $\mathbf{y}_{1:N_T}$ the UKF or EKF filter are able to return an estimate of the laser internal states, for example using the posterior mean $\mathbf{m}_{1:N_T}$.

To validate the tracking algorithms, it is necessary to simulate the laser model using Algorithm 3, apply a filter algorithm (EKF or UKF) and compute the RIN/FN spectra on the chosen point estimate for the dynamic states, the mean in this case.

3.5.1 Non-linear tracker

To include the full knowledge of the laser model in the tracker, is necessary to use (2.29) or (2.34) as dynamic equation. The measurement model is (3.2). A filtering

algorithm implementing the rate-equation model can be thereafter refereed as non-linear tracker (NL). By adopting the same timestep of Chapter 2, EM was chosen over SROCK because it is simple and it does not require multiple stages to approximate the next point. Depending on the noise level in the system and the choice of the starting point \mathbf{m}_0 , there may be a *lag* interval where the tracker slowly converges to the optimal estimate, reducing the error \mathbf{e}_t . However, when \mathbf{f} identifies the rate-equation model, \mathbf{m}_0 can be initialized with the steady states (2.4)

$$\mathbf{m}_0 = [N_{ss}, S_{ss}, 0]^\top \quad (3.23)$$

This method will reduce the lag significantly, on the hypothesis that the laser is in the steady-state regime and the static parameters are known. Regarding initialization of \mathbf{P}_0 , any diagonal positive definite matrix will work, usually setting $\mathbf{P}_0 = \mathbf{I}_3$ for simplicity.

Regarding the choice of the filter, in Figure 3.3 is possible to see that on the same setting, the mean estimate for the states of UKF and EKF almost overlap. This is an evidence of the fact that despite the high non-linearities in the rate-equation, with a small timestep a linear approximation carried out with EKF works fine.

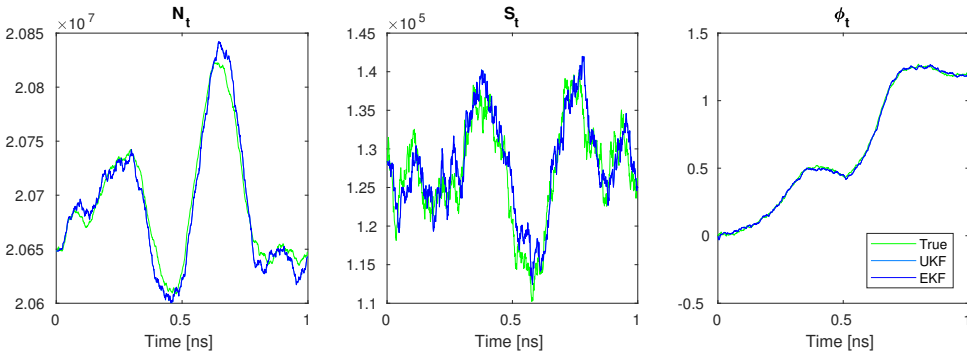


Figure 3.3: State tracking comparing EKF with UKF.

Since both filters have the same performance, the best filter would be the fastest one. The choice is of course directed toward EKF. In Figure 3.4 the performance of RIN/FN estimation is compared with the conventional method. This shows that with the same SNR ($\sigma_r = 10^{-3}$) the EKF is able to characterize the noise close to the ground truth. A performance of the method in presence of higher noise levels will be shown in the next section.

3.5.2 Linear tracker

An alternative way to perform filtering on the measurement of the laser is to track only the electric field and the laser's phase, using a *Markovian Random walk* to model

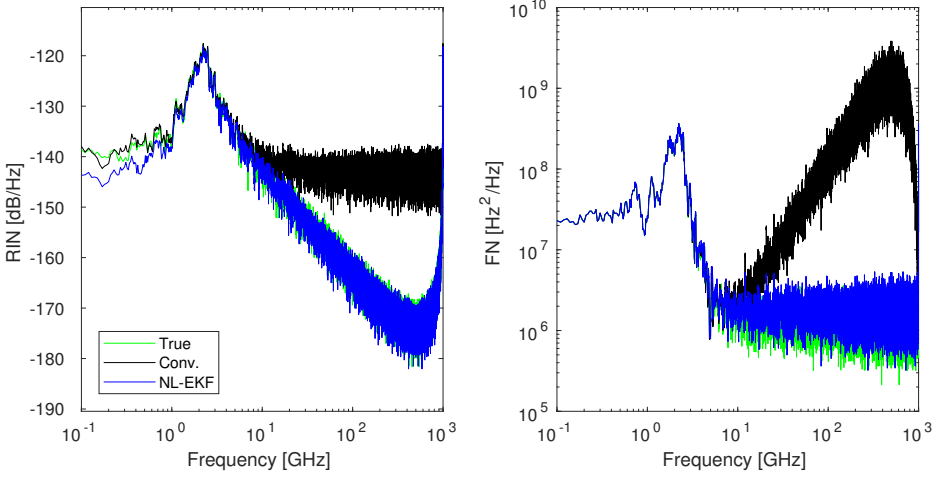


Figure 3.4: RIN/FN estimate comparing a non-linear EKF tracker with the conventional way.

the lasers' dynamics

$$\begin{aligned} E_{t+1} &= E_t + q_{E,t} \\ \phi_{t+1} &= \phi_t + \sigma_\phi q_{\phi,t} \end{aligned} \quad (3.24)$$

with $q_{E,t} \sim \mathcal{N}(0, \sigma_E^2)$ and $q_{\phi,t} \sim \mathcal{N}(0, \sigma_\phi^2)$. σ_E^2 and σ_ϕ^2 are chosen by estimating the noise variance of the processes $\{E_t - E_{t-1}\}$ and $\{\phi_t - \phi_{t-1}\}$ generated from the system simulation. This is called linear tracker (L) by referring to the state dynamics, despite the fact that the measurement equation is still non-linear, even if it simplifies in

$$\begin{aligned} y_{i,t} &= E_t \cos(\phi_t) + r_{i,t} \\ y_{q,t} &= E_t \sin(\phi_t) + r_{q,t} \end{aligned} \quad (3.25)$$

Motivations under the use of this model relies on its simplicity and on the approximation that in a steady state regime, in (2.29) the drift term can be neglected and the diffusion coefficients are approximately constant. Therefore, (2.29) can be approximated as a random walk. Using a linear tracker can be useful when the knowledge of the internal laser model is not know, therefore making it adaptive in different scenarios.

Despite this model behaves completely different from the rate equation, in Figure 3.5 is demonstrated the linear tracker has the same tracking performance of the non-linear tracker. However, the two tracker agree more on the estimated phase than

the amplitude. This behavior can be understood by considering that, the phase dynamics for the non-linear tracker, not being forced to oscillates around a steady state, behaves closer to a random walk model. Therefore, the linear-tracker approximation for the phase is more correct than the one for the amplitude. Is worth also noticing that this tracker works as long as the SNR is not too high. The reason is that it relies more on the update step than on the prediction step, being the mean prediction a simple copy of the previous point i.e. $\mathbf{m}_t^- = \mathbf{m}_{t-1}$.

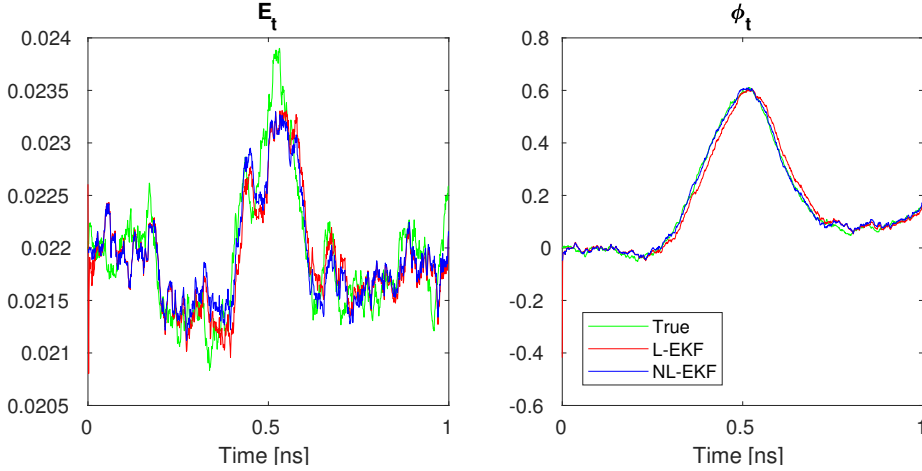


Figure 3.5: Amplitude and phase tracking comparing a linear (L-EKF) and non-linear tracker (NL-ELK).

3.5.3 Performance limits on noisy channels with one quadrature available

Different simulated experiments are performed to test the limit of each tracking model.

The aim is to verify where a tracker is able to produce a good estimate when there is a reduction of the available measurement set. This may occurs when only the in-phase measurement can be detected, precisely when

$$y_t = E_t \cos(\phi_t) + r_{i,t} \quad (3.26)$$

This scenario is compared to the one where both measurements are available. For each scenarios, different level of noise are added to the measurements in order to see how much measurement noise a tracking algorithm can sustain before the spectrum estimate completely separates from the ground truth. It is important to know that

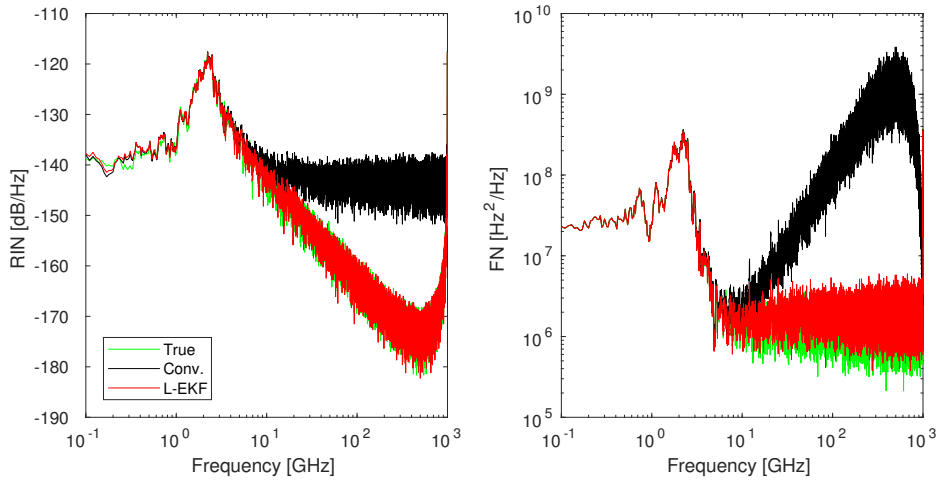


Figure 3.6: RIN/FN estimate comparing a linear EKF tracker with the conventional way.

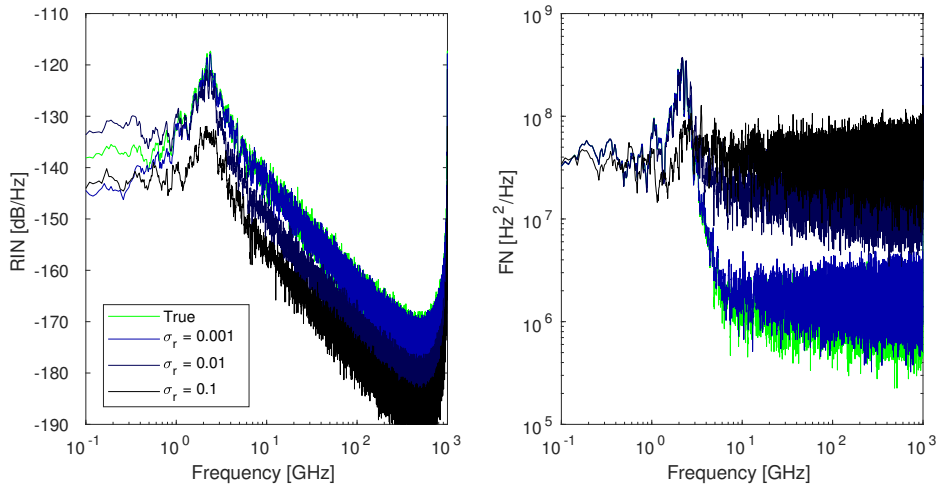


Figure 3.7: RIN/FN estimate using a NL-EKF with different levels of noise and both quadratures available.

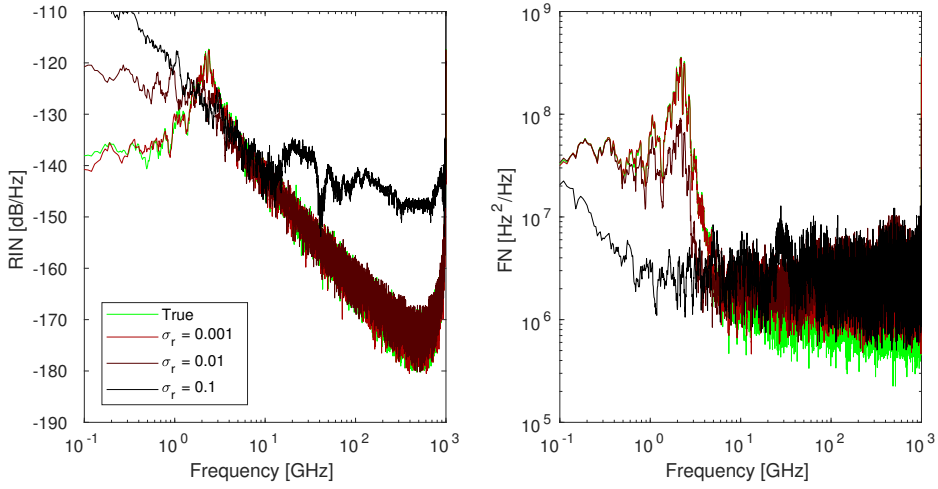


Figure 3.8: RIN/FN estimate using a L-EKF with different levels of noise and both quadratures available.

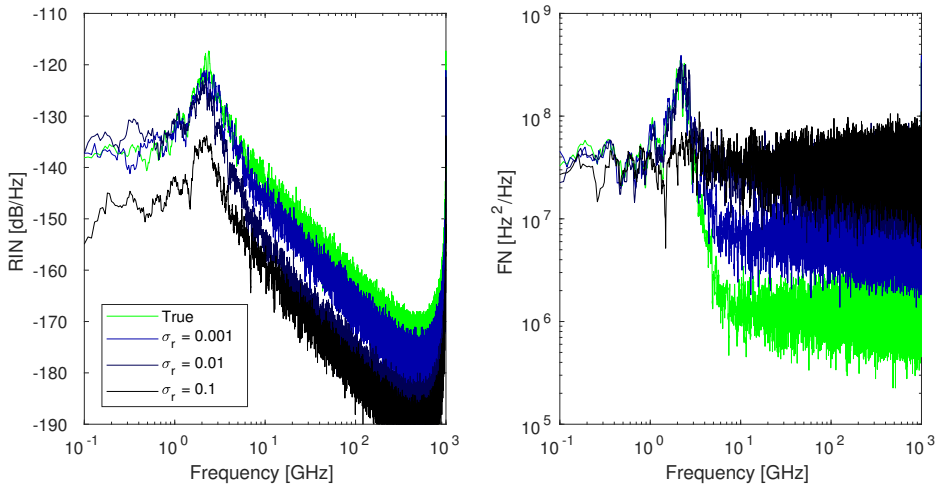


Figure 3.9: RIN/FN estimate using a NL-EKF with different levels of noise and one quadrature available.

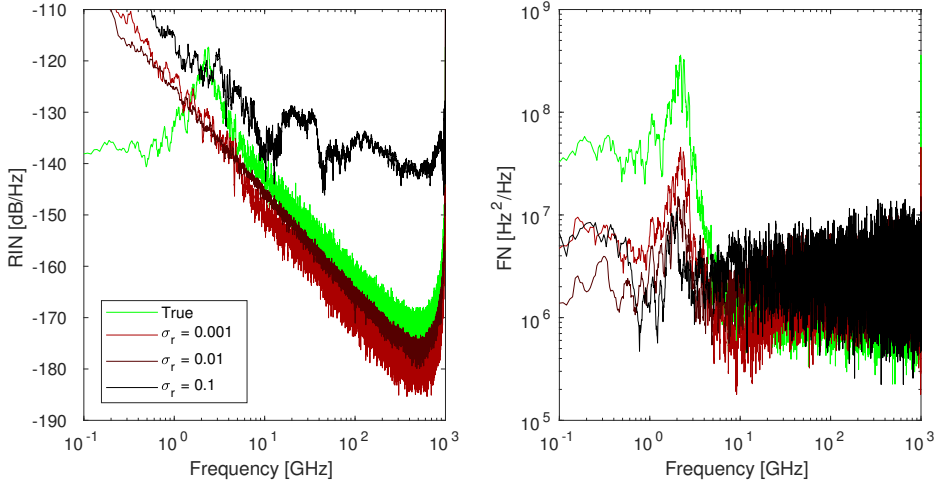


Figure 3.10: RIN/FN estimate using a L-EKF with different levels of noise and one quadrature available.

such limits are important because they will represent the best case for the tracking algorithm, i.e. when the model parameters are known a priori.

As expected, by comparing Figure 3.7 with Figure 3.9 and Figure 3.8 with Figure 3.10, the goodness of the estimate is better when the knowledge of both quadrature is available. Moreover, when the noise is increased, both NL and L tend to overestimate the phase noise but shows different behavior regarding the intensity noise. Again, this lack of agreement on the amplitude may have the same explanation given in the previous section, relying on the different nature of the dynamical model implemented.

Although a tracking algorithm performs significantly better in terms of RIN/FN characterization compared to a conventional estimate, it still cannot face high levels of measurement noise. An additional improvement to the quality of the estimate may be found in the next chapter, where the estimated static parameters can also be used for simulating the RIN/FN curves.

CHAPTER 4

Parameter estimation in a Bayesian framework

4.1 Introduction

When dealing with modelling laser systems, the static parameters θ plays an important role in characterizing such models. In fact, giving that a mathematical model formulation of a laser is correct, such parameters completely describe the dynamic evolution of a laser system, including the profiling RIN and FN curves. This can be useful, for example, in low SNR settings where tracking the system states has difficulties to reconstruct the original spectra due to measurement noise.

However, in many practical cases those parameters are not known exactly or are known within a range of uncertainty, by assuming a prior PDF $p(\theta)$. Therefore, a technique is desired to refine the knowledge over θ after observing new measurements acquired from the laser.

Static parameters, in contrast with the dynamical ones are constant quantities which does not change in time. A solution proposed in the *state augmentation* approach [32] consists to treat θ as additional dynamic hidden variable together with \mathbf{x} . The problem with this methods is that may lead to singularities hard to handle for non-linear filters.

In the Bayesian terminology, this translated to obtain the posterior distribution of the parameters given the laser measurements, i.e. $p(\theta|\mathbf{y}_{1:N_T})$. By including these new quantities in the PSSM [32]

$$\theta \sim p(\theta) \tag{4.1}$$

$$\mathbf{x}_0 \sim p(\mathbf{x}_0|\theta) \tag{4.2}$$

$$\mathbf{x}_{t+1} \sim p(\mathbf{x}_{t+1}|\mathbf{x}_t, \theta) \tag{4.3}$$

$$\mathbf{y}_t \sim p(\mathbf{y}_t|\mathbf{x}_t, \theta) \tag{4.4}$$

and the straightforward Bayesian way to compute the posterior of θ consists in

marginalizing out the joint conditional of static and dynamic parameters

$$p(\boldsymbol{\theta}|\mathbf{y}_{1:N_T}) = \int p(\mathbf{x}_{0:N_T}, \boldsymbol{\theta}|\mathbf{y}_{1:N_T}) d\mathbf{x}_{0:N_T} \quad (4.5)$$

where the integrand is identified by solving (4.1) - (4.4)

$$\begin{aligned} p(\mathbf{x}_{0:N_T}, \boldsymbol{\theta}|\mathbf{y}_{1:N_T}) &= \frac{p(\mathbf{y}_{1:N_T}|\mathbf{x}_{0:N_T}, \boldsymbol{\theta})p(\mathbf{x}_{0:N_T}|\boldsymbol{\theta})p(\boldsymbol{\theta})}{p(\mathbf{y}_{1:N_T})} \\ &= \frac{\left[\prod_{k=1}^{N_T} p(\mathbf{y}_k|\mathbf{x}_k, \boldsymbol{\theta}) \right] \left[p(\mathbf{x}_0) \prod_{k=1}^{N_T} p(\mathbf{x}_k|\mathbf{x}_{k-1}, \boldsymbol{\theta}) \right] p(\boldsymbol{\theta})}{p(\mathbf{y}_{1:N_T})} \end{aligned} \quad (4.6)$$

Computing (4.5) becomes challenging because of the high dimensionality and the complexity proportional on the amount of obtained measures. To overcome this, the posterior over $\boldsymbol{\theta}$ may be approximated without taking into account the state distribution [32], i.e.

$$p(\boldsymbol{\theta}|\mathbf{y}_{1:N_T}) = \frac{p(\mathbf{y}_{1:N_T}|\boldsymbol{\theta})p(\boldsymbol{\theta})}{\int p(\mathbf{y}_{1:N_T}|\boldsymbol{\theta})p(\boldsymbol{\theta})d\boldsymbol{\theta}} \quad (4.7)$$

Where $p(\mathbf{y}_{0:N_T-1}|\boldsymbol{\theta})$ is the likelihood of the measurements given the current parameters. In the hypothesis that the measurements are conditionally independent, this likelihood factorizes further

$$p(\mathbf{y}_{1:N_T}|\boldsymbol{\theta}) = \prod_{t=1}^{N_T} p(\mathbf{y}_t|\mathbf{y}_{1:t-1}, \boldsymbol{\theta}) \quad (4.8)$$

Thankfully the Bayesian filtering theory already provide an efficient way to estimate $p(\mathbf{y}_t|\mathbf{y}_{0:t-1}, \boldsymbol{\theta})$. In fact this can be compute by marginalizing over $d\mathbf{x}_t$

$$p(\mathbf{y}_t|\mathbf{y}_{1:t-1}, \boldsymbol{\theta}) = \int p(\mathbf{y}_t|\mathbf{x}_t, \boldsymbol{\theta})p(\mathbf{x}_t|\mathbf{y}_{1:t-1}, \boldsymbol{\theta})d\mathbf{x}_t \quad (4.9)$$

where the probabilistic measurement model $p(\mathbf{y}_t|\mathbf{x}_t, \boldsymbol{\theta})$ can be identified from basic assumption on the system and the prediction distribution $(\mathbf{x}_t|\mathbf{y}_{1:t-1}, \boldsymbol{\theta})$ is can be computed from a Bayesian filtering algorithm.

Once obtained the posterior probabilities of the parameters available, it is possible then to chose a point $\hat{\boldsymbol{\theta}}$ coherently with $p(\boldsymbol{\theta}|\mathbf{y}_{1:N_T})$ to perform a tracking of the system states for computing a filtered RIN/FN or use such estimate to simulate a laser and then generate the simulated spectra.

4.2 Energy function

When dealing with probabilities, due to the limited numerical precision on representing the real interval $[0, 1]$, a much more convenient way is to evaluate (4.7) in the

log-domain. In addition, the normalization constant at the denominator of (4.7) does not depend on $\boldsymbol{\theta}$, therefore it can be excluded and

$$\log p(\boldsymbol{\theta}|\mathbf{y}_{1:N_T}) \propto \varphi_{N_T}(\boldsymbol{\theta}) := \sum_{t=1}^{N_T} \log p(\mathbf{y}_t|\boldsymbol{\theta}) + \log p(\boldsymbol{\theta}) \quad (4.10)$$

The last term of (4.10) is called *energy function* [32] and it has the interesting property that it can be estimated with the following recursion

$$\begin{aligned} \varphi_0(\boldsymbol{\theta}) &= -\log p(\boldsymbol{\theta}) \\ \varphi_t(\boldsymbol{\theta}) &= \varphi_{t-1}(\boldsymbol{\theta}) + \log p(\mathbf{y}_t|\boldsymbol{\theta}) \text{ for } t = 1, \dots, N_T \end{aligned} \quad (4.11)$$

Because $p(\boldsymbol{\theta}|\mathbf{y}_{1:N_T}) \propto \exp \varphi_{N_T}(\boldsymbol{\theta})$, the energy function is of fundamental importance when the estimation of the static parameters is required. From the recursion (4.11), the key to compute $\varphi_{N_T}(\boldsymbol{\theta})$ is to obtain $p(\mathbf{y}_t|\boldsymbol{\theta})$, referred also as *prediction error probability*. In fact, (4.9) can be carried out easily in close form when dealing with linear Kalman filter. For the filters used in this work, like EKF and UKF, (4.9) can still be approximated in closed form. In fact, in any Gaussian filter the prediction error follows the distribution

$$p(\mathbf{y}_t|\boldsymbol{\theta}) \sim \mathcal{N}(\boldsymbol{\mu}_t(\boldsymbol{\theta}), \mathbf{S}_t(\boldsymbol{\theta})) \quad (4.12)$$

where $\boldsymbol{\mu}_t(\boldsymbol{\theta})$ is the predicted measurement from the predicted state \mathbf{m}_t^- and $\mathbf{S}_t(\boldsymbol{\theta})$ is the relative uncertainty associated. In both EKF and UKF, $\boldsymbol{\mu}_t$ and \mathbf{S}_t are already estimated during the prediction step. Finally, the approximation for the energy recursion is

$$\varphi_t(\boldsymbol{\theta}) \approx \varphi_{t-1}(\boldsymbol{\theta}) - \frac{1}{2} \log |2\pi \mathbf{S}_t(\boldsymbol{\theta})| - \frac{1}{2} \mathbf{e}_t^\top(\boldsymbol{\theta}) \mathbf{S}_t^{-1}(\boldsymbol{\theta}) \mathbf{e}_t(\boldsymbol{\theta}) \quad (4.13)$$

where $\mathbf{e}_t(\boldsymbol{\theta}) = \mathbf{y}_t - \boldsymbol{\mu}_t(\boldsymbol{\theta})$ is the prediction error, already defined in Chapter 3.

4.3 Point estimate

One way to select a proper parameter is to minimize the error probability, or equivalently, minimize (4.13). This correspond to finding the *maximum a posteriori* MAP point for $\boldsymbol{\theta}$

$$\hat{\boldsymbol{\theta}}^{\text{MAP}} = \arg \min_{\boldsymbol{\theta}} [\log p(\boldsymbol{\theta}|\mathbf{y}_{1:N_T})] \approx \arg \min_{\boldsymbol{\theta}} [\varphi_{N_T}(\boldsymbol{\theta})] \quad (4.14)$$

If the prior probability on the parameters $p(\boldsymbol{\theta})$ is uniform distributed in the parameter space, such prior does not depend on $\boldsymbol{\theta}$ anymore therefore it is independent on the minimization of (4.14). This express the maximum prior uncertainty on $\boldsymbol{\theta}$ and the same minimization carried on (4.13) becomes the *maximum likelihood* point estimate

$$\hat{\boldsymbol{\theta}}^{\text{ML}} = \arg \min_{\boldsymbol{\theta}} \left[\prod_{t=1}^{N_T} p(\mathbf{y}_t|\boldsymbol{\theta}) \right] \approx \arg \min_{\boldsymbol{\theta}} [\varphi_{N_T}(\boldsymbol{\theta})] \text{ with } \varphi_0(\boldsymbol{\theta}) = 0 \quad (4.15)$$

It is interesting to note that such point estimates that involves the minimization of the energy function try to reduce the prediction error \mathbf{e}_t but take also account of the uncertainty of the current estimate and the received measurement, encoded in $\mathbf{S}_t(\boldsymbol{\theta})$. Neglecting such uncertainty may lead to overfit the estimate to the measurement noise, thus approaches involving *least square* point estimate

$$\hat{\boldsymbol{\theta}}^{\text{LS}} = \arg \min_{\boldsymbol{\theta}} \left[\sum_{t=1}^{N_T} \mathbf{e}_t^\top(\boldsymbol{\theta}) \mathbf{e}_t(\boldsymbol{\theta}) \right] \quad (4.16)$$

are better avoided in presence of high measurement noise.

Minimizing directly $\varphi(\boldsymbol{\theta})$ may be challenging. Unfortunately it is not an analytic function, meaning that there is not an explicitly dependence on $\boldsymbol{\theta}$. Classical methods from calculus analysis fail because differentiation is not possible. According to the literature, the function in exam is a *black box* function. There is not a simple way to describe the relationship between input $\boldsymbol{\theta}$ and output $\varphi_{N_T}(\boldsymbol{\theta})$. It is only possible to compute the output $\varphi_{N_T}(\boldsymbol{\theta})$ of the filtering algorithm for every given input $\boldsymbol{\theta}$. In the literature several *derivative-free optimization* algorithms have been developed [42, 43] aimed to find a local minimum for these functions. However, most of the methods are based on heuristic approaches, does not guarantee to converge to a global minimizer and they produce a single point-estimate as output. In fact, the overall behavior of $\varphi(\boldsymbol{\theta})$ is not known and it may have multiple local minimizers, depending on the parameter space, the model used and the set of measurements.

4.4 Density estimate: Metropolis-Hastings sampling

Computing the overall form of the energy function may be desirable to obtain a better characterization of the posterior distribution of $\boldsymbol{\theta}$. Naive methods can compute $\varphi(\cdot)$ over a sufficient number of equally spaced lattice points, but this approach fails when dealing with high dimensions, in which the number of lattice points increase exponentially with d [35]. One useful method to build $p(\boldsymbol{\theta}|\mathbf{y}_{1:N_T})$ is to use *sampling algorithms*. By sampling from this distribution and then analyze the generated particles, with enough points it is possible to estimate the mean, mode, median and all the moments that can be estimated from a collection of samples. Because the posterior is not directly available, Metropolis-Hasting algorithm (MH) [32] uses the energy function to generate samples in a Markov-Chain fashion. MH uses a proposal distribution to generate new samples $\boldsymbol{\theta}^*$ and accept or reject them basing on $\varphi_T(\boldsymbol{\theta}^*)$, in the specific case by evaluating the goodness of the tracking on the given measurements. When a new particle is accepted, the proposal distribution changes according to the new accepted sample. In this way, the generated samples follow a Markov chain fashion. It is guaranteed, when the number of samples $N_{\text{MH}} \rightarrow +\infty$ that the stationary distribution of the MC converges to the full posterior distribution. A very simple and basic MH algorithm is presented in Algorithm 6.

Algorithm 6 MH. Metropolis Hasting

Initialize. Define the number of total samples, the proposal distribution $q(\cdot)$ and the starting point $\boldsymbol{\theta}^{(0)}$
for $i = 1$ to N_{MH} **do**
 Sample a candidate point

$$\boldsymbol{\theta}^* \sim q(\boldsymbol{\theta}^* | \boldsymbol{\theta}^{(i-1)}) \quad (4.17)$$

Calculate the acceptance probability ξ_i

$$\xi_i := \min \left\{ 1, \frac{p(\boldsymbol{\theta}^* | \mathbf{y}_{0:T})}{p(\boldsymbol{\theta}^{(i-1)} | \mathbf{y}_{0:T})} \frac{q(\boldsymbol{\theta}^{(i-1)} | \boldsymbol{\theta}^*)}{q(\boldsymbol{\theta}^* | \boldsymbol{\theta}^{(i-1)})} \right\} \quad (4.18)$$

Generate $u \sim \mathcal{U}[0, 1]$ and set the next point

$$\boldsymbol{\theta}^{(i)} = \begin{cases} \boldsymbol{\theta}^* & \text{if } u \leq \xi_i \\ \boldsymbol{\theta}^{(i-1)} & \text{otherwise} \end{cases} \quad (4.19)$$

end for

return $\{\boldsymbol{\theta}^{(i)}\}_{i=1}^{N_{\text{MH}}}$

In practice, basing on the choice of the proposal distribution, (4.18) can be further simplified. For example in a random walk Metropolis-Hastings, the proposal is a Gaussian distribution centred in the previous point and with a defined covariance $\boldsymbol{\Sigma}^{(i)}$, there

$$q(\boldsymbol{\theta}^* | \boldsymbol{\theta}^{(i-1)}) := \mathcal{N}(\boldsymbol{\theta}^{(i-1)}, \boldsymbol{\Sigma}^{(i)}) \quad (4.20)$$

this makes ξ_i independent on the proposal value, being the latter symmetric. Hence, (4.18) further simplifies in

$$\xi_i := \min \left\{ 1, \exp \left(\varphi_T(\boldsymbol{\theta}^{(i-1)}) - \varphi_T(\boldsymbol{\theta}^*) \right) \right\} \quad (4.21)$$

However, despite the convergence is proven in the long run, sometimes it can be very difficult to navigate into the parameter space to characterize the posterior over the parameters. This is a problem especially when dealing with high dimensions, when the form of the proposal distribution has high dissimilarity with the posterior and when $\varphi(\boldsymbol{\theta})$ presents different local minima. When this happens, the sampling algorithm may get stuck in a single point and fails to explore the $\boldsymbol{\theta}$ space. A good feedback on the goodness of sampling can be given by the fraction of accepted samples over the total generated $\bar{\xi}$.

To overcome these difficulties, modification of the original MH algorithm have been proposed to include component-wise sampling [44] and dynamic adaptive of the algorithm [45]. Despite those approach are heuristic, they sometimes improve the characterization of the parameter space. Algorithm 9, presented in Appendix A,

shows a modified version of MH, called Random Walk Adaptive Metropolis Hastings (RWAMH) which can help to navigate high dimensional spaces. It is a personal reinterpretation of a random walk MH by adapting the covariance based on the previous generated samples. Every U_{MH} iterations $\Sigma^{(i)}$ is being re-estimated on the previous H_{MH} generated samples and eventually re-scaled by a factor s_{MH} if the average acceptance rate does not satisfy a minimum requirement ξ_d . The rescaling factor scales $\Sigma^{(i)}$ up if the acceptance rate is too high or down if it is too low. The constant c_{MH} controls the weight given to the previous samples and ϵ_{MH} controls the additive term for rescaling in case the covariance re-estimation is singular.

When the sampling algorithm has finished, typically a part of the samples is discarded due to the *burn-in period* that an MH-based algorithm needs for converging to the true posterior distribution [46]. For this reason an initial fraction of the particles, here $1/3$, is discarded due to the high dependence on the original starting value. The obtained particles can be arranged in an histogram, for which an empirical PDF can be estimated. Other methods, not presented here, may use a *kernel density estimation* [47] to obtain a continuous PDF from discrete sample distribution. Giving the collection of samples, a good point estimate can be represented by the mean value

$$\mathbf{m}_\theta = \frac{1}{N_{\text{MH}} - N_{\text{burn-in}}} \sum_{i=N_{\text{burn-in}}+1}^{N_{\text{MH}}} \theta^{(i)} \quad (4.22)$$

This is in fact the point that will be used for tracking a sequence of measurements after sampling.

4.5 Testing and validation

When comes to estimate the parameters for a sequence of measurements, it is necessary to define the starting point and the initial covariance value, in addition to other secondary parameters for Algorithm 9. In addition, some parameters may have some restriction in their domain, meaning that $\theta \in \Theta \subseteq \mathbb{R}^d$. For example, some parameters may be restricted to be positive, therefore lying in the range $[0, +\infty]$. In absence of a prior distribution $p(\theta)$, the space exploration of a sampling algorithm has to be enforced to be contained in Θ . By mean of a truncated Gaussian proposal [48], it is possible to sample by remaining inside the allowed space. Luckily this distribution still preserves the symmetry, therefore for RWAMH (4.21) will remain the same. Because during this section of sampling the true parameters are known a priori, $\theta^{(0)}$ is selected by sampling around a uniform distribution of range u_{MH} around the true value. $\Sigma^{(0)}$ is diagonal whose terms are $c_{\text{MH}} \cdot |\theta^{(0)}|$. This allows to take into account the different magnitude of the model parameters involved in the sampling algorithm. True model parameters corresponds to the same used in Chapter 3. The noise variance this time was set to $\sigma_r = 0.1$, a high value that in the previous chapter failed to provide a good RIN/FN characterization from the measurements. The

parameters used for the Metropolis algorithm are determined heuristically based on different tests. In this specific case, those are presented in Table 4.1.

RWAMH parameters	
Parameter	Value
N_{MH}	10^3
U_{MH}	10^2
c_{MH}	50
ϵ_{MH}	$1 \cdot 10^{-5}$
s_{MH}	10^2
ξ_d	0.2
u_{MH}	0.5

Table 4.1: Parameters used for sampling.

The two main trackers introduced in Chapter 3 are tested in a sampling setting. This means that there will be two different parameter spaces, depending on the model used, which can be seen in Tables 4.2 and 4.3. As did in Chapter 3, for each tracker the performance of sampling is tested when one or both quadratures are available. The posterior distribution of the particles will be presented as an histogram of samples, and the selected estimate will be the mean of all the samples after the burn-in period (4.22). Each histogram represents an empirical marginal probability distribution for the correspondent parameter.

Using different level of measurement noise and different length of the measure set was tested but not presented here. In general however, it was observed that with longer observed sequences and low levels of measurement noise, the histogram distribution of samples tends to peak around a local minimum. Vice versa, when the noise is high and less measures are available, the distribution spreads more. This has also consequences on the resulting acceptance rate, which drops down for high-peaked posterior distributions. It is also important to remark that using less measurements reduces the computational time but gives a poor characterization, while more measurements increases the computational time. This is important because, for every new sample generated by a Metropolis algorithm, the whole sequence needs to be filtered again.

4.5.1 Linear-tracker

Sampling in \mathbb{R}^3 makes the posterior characterization of the parameters much easier. In particular, the true value is approached by the mean estimate on the particles, making it an easy minimum to reach. Given the fact that the space dimensionality is low, a nice characterization of the posterior distribution is achieved using one (Figure 4.1) or both (Figure 4.2) quadratures available.

However by overlapping the histograms in Figure 4.4 it is evident that there exists multiple minima closer to the real parameter value. Those minimizers are close to each

Linear tracker		
Parameter	Theoretical range	Initial value
σ_E	$[0, +\infty]$	$\sim \mathcal{U}[(1 - u_{\text{MH}})\sigma_E, (1 + u_{\text{MH}})\sigma_E]$
σ_ϕ	$[0, +\infty]$	$\sim \mathcal{U}[(1 - u_{\text{MH}})\sigma_\phi, (1 + u_{\text{MH}})\sigma_\phi]$
σ_r	$[0, +\infty]$	$\sim \mathcal{U}[(1 - u_{\text{MH}})\sigma_r, (1 + u_{\text{MH}})\sigma_r]$

Table 4.2: Parameters sampled for L tracker.

other so that the difference compared with a true estimate (presented in Chapter 3) is hardly noticeable. However this situation becomes more frequent when the number of parameter used grows, as it will be seen by testing the non-linear tracker.

The drawback of this sampling setting is that, despite the parameters are correctly estimated, the quality of the spectra obtained in Figure 4.3 cannot be improved due to the noise limit seen in Chapter 3. In fact the estimated parameters are not particularly useful for identifying a laser model.

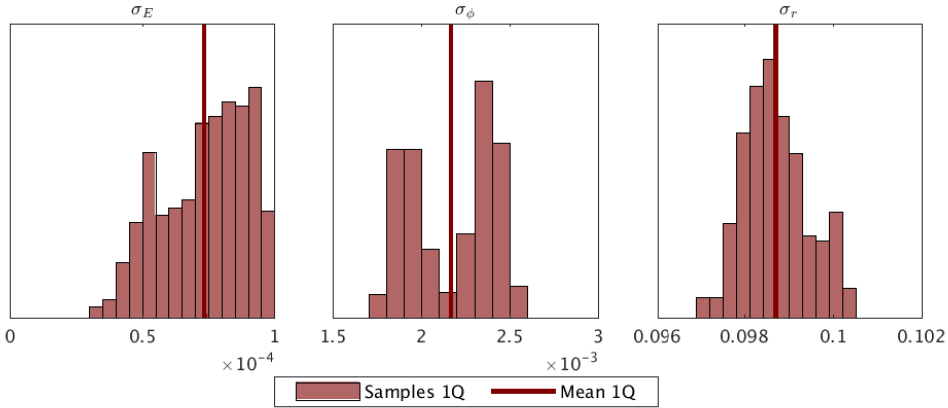


Figure 4.1: Samples of L-tracker using 1 quadrature.

4.5.2 Non-linear tracker

For the non linear-tracker, the same framework is adopted as the one in Section 4.5.1. Regarding parameter estimation, both η_d and λ are coupled together because it is impossible to discern them individually given their fraction appearing in the measurement equation.

Thanks to the RWAMH adaptivity, the histogram distribution in Figures 4.5 and 4.6 provide a nice characterization of the true posterior around the mode. Un-

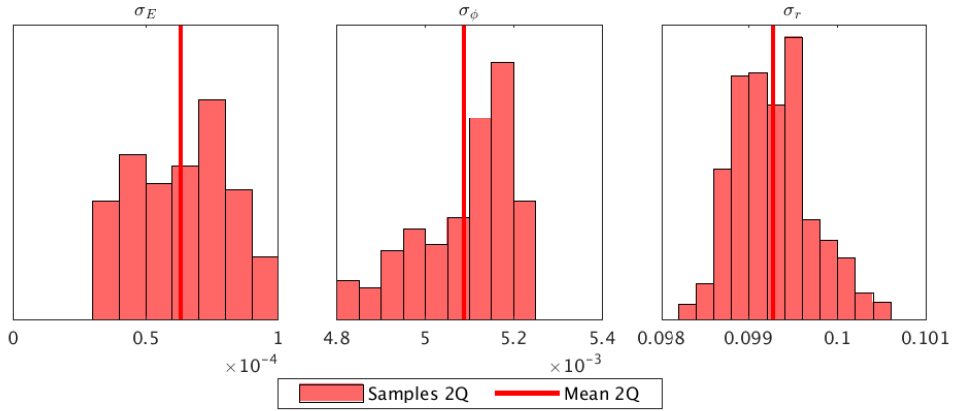


Figure 4.2: Samples of L-tracker using 2 quadratures.

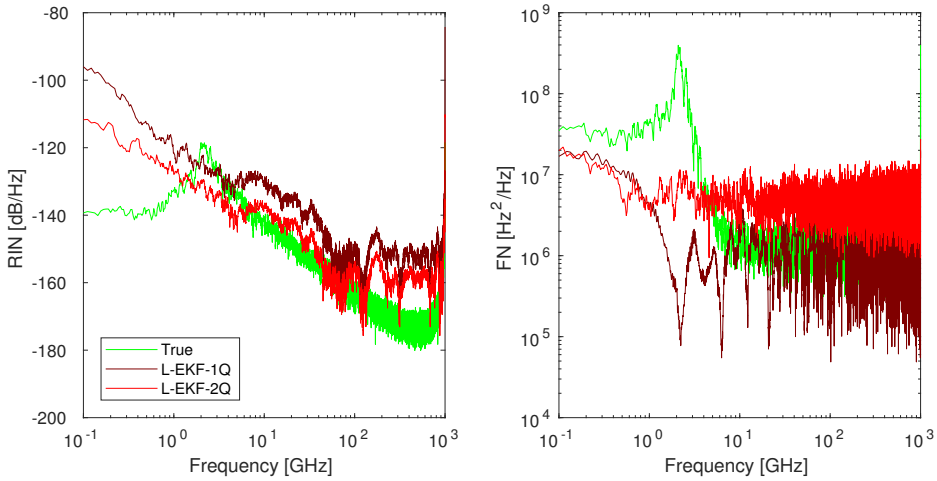


Figure 4.3: RIN/FN estimation of a L-tracker using 1 or 2 quadratures available.

fortunately, the behavior of the energy function in a bigger sampling space is not as good as expected. The sampling algorithm get sometimes stuck in a local minimum. This happens when using one or two quadratures. Figure 4.8 shows that for some parameters, the mean estimate coincides with the true value while for other it does

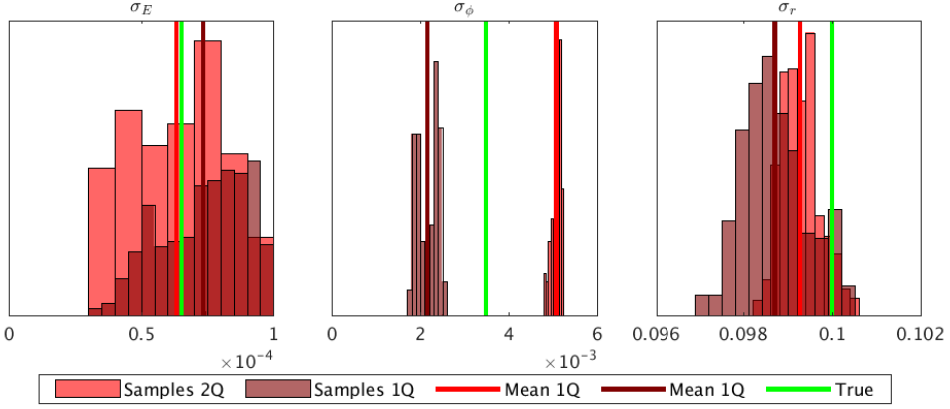


Figure 4.4: Overlap histograms of samples of L-tracker using 1 or 2 quadratures.

Non-Linear tracker		
Parameter	Theoretical range	Initial value
τ_n	$[0, +\infty]$	$\sim \mathcal{U}[(1 - u_{\text{MH}})\tau_n, (1 + u_{\text{MH}})\tau_n]$
τ_p	$[0, +\infty]$	$\sim \mathcal{U}[(1 - u_{\text{MH}})\tau_p, (1 + u_{\text{MH}})\tau_p]$
g	$[0, +\infty]$	$\sim \mathcal{U}[(1 - u_{\text{MH}})g, (1 + u_{\text{MH}})g]$
β	$[0, +\infty]$	$\sim \mathcal{U}[(1 - u_{\text{MH}})\beta, (1 + u_{\text{MH}})\beta]$
ϵ	$[0, +\infty]$	$\sim \mathcal{U}[(1 - u_{\text{MH}})\epsilon, (1 + u_{\text{MH}})\epsilon]$
N_0	$[0, +\infty]$	$\sim \mathcal{U}[(1 - u_{\text{MH}})N_0, (1 + u_{\text{MH}})N_0]$
α	$[0, +\infty]$	$\sim \mathcal{U}[(1 - u_{\text{MH}})\alpha, (1 + u_{\text{MH}})\alpha]$
I	$[0, +\infty]$	$\sim \mathcal{U}[(1 - u_{\text{MH}})I, (1 + u_{\text{MH}})I]$
η_d/λ	$[0, +\infty]$	$\sim \mathcal{U}[(1 - u_{\text{MH}})\eta_d/\lambda, (1 + u_{\text{MH}})\eta_d/\lambda]$
σ_r	$[0, +\infty]$	$\sim \mathcal{U}[(1 - u_{\text{MH}})\sigma_r, (1 + u_{\text{MH}})\sigma_r]$

Table 4.3: Parameters sampled for NL tracker.

not. If the true posterior distribution is multimodal, then is difficult for the proposed sampling algorithms to characterize all possible modes. Therefore, the mode found is probably the closest one to the original starting value. Perhaps with additional knowledge on the parameters range or a non-uniform prior distribution, many modes can be actually discarded. Because of the difficulty to move in such sampling space, the original parameters cannot be reconstructed exactly. However, because the energy function converges to a minimum in any case, different sets of parameters achieve the same good fit on the given measurements. In fact, the filtered states from the mean estimate in Figure 4.7 gives an estimation comparable to the one achieved with

true parameters in Chapter 3. So in general, parameter estimation cannot clean more noise in the measurements than a filtering algorithm. But because the extracted parameters can be used to simulate a laser, in Figure 4.7 is presented also the simulated spectrum. This latter one attain a closer characterization of the ground truth, making the parameter estimation techniques useful also to characterizing RIN and FN spectra of a laser.

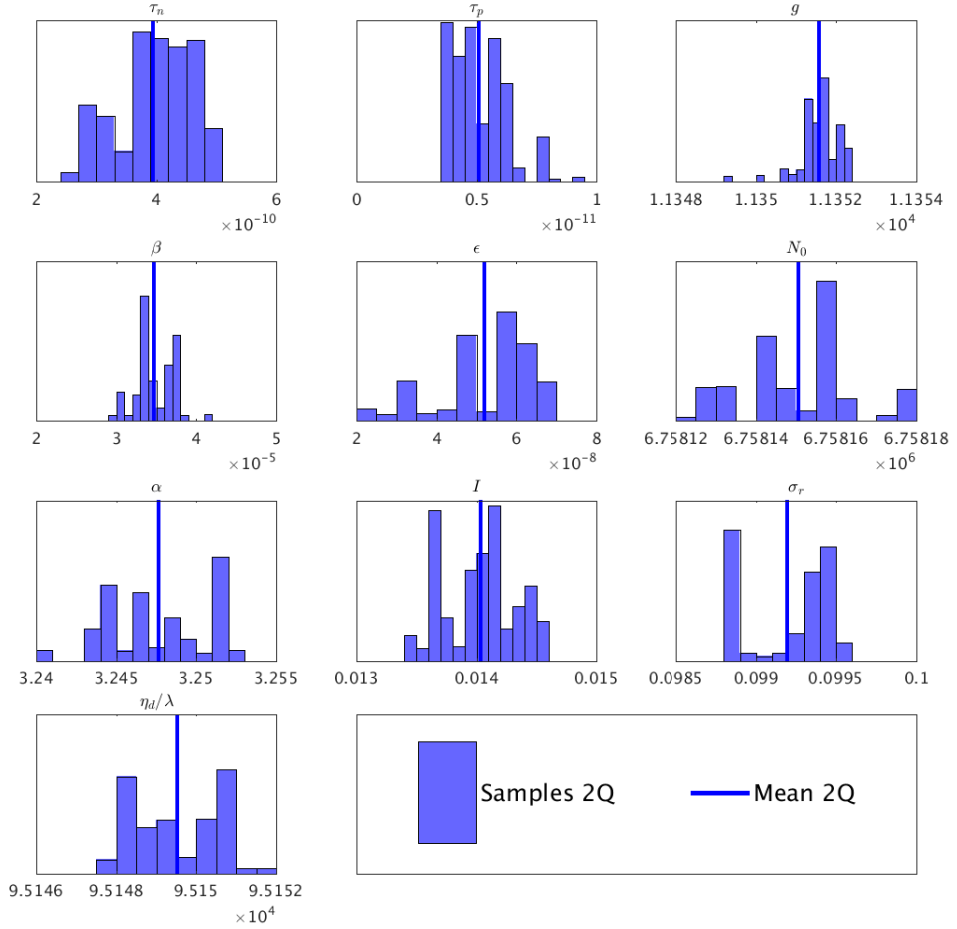


Figure 4.5: Samples of NL-tracker using 2 quadratures.

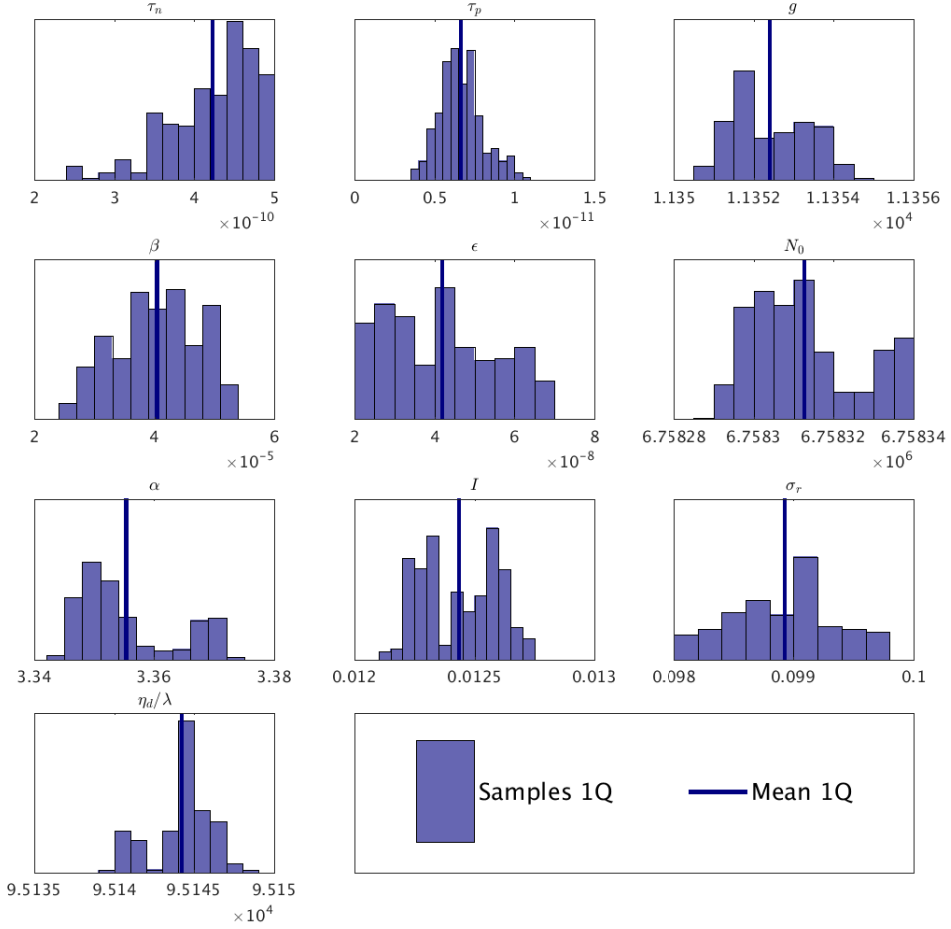


Figure 4.6: Samples of NL-tracker using 1 quadrature.

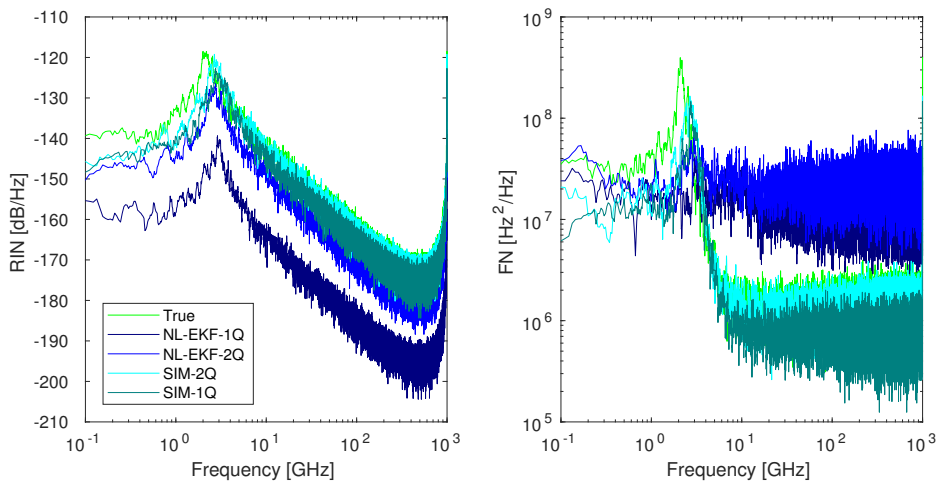


Figure 4.7: RIN/FN estimation of a NL-tracker using 1 or 2 quadratures available.

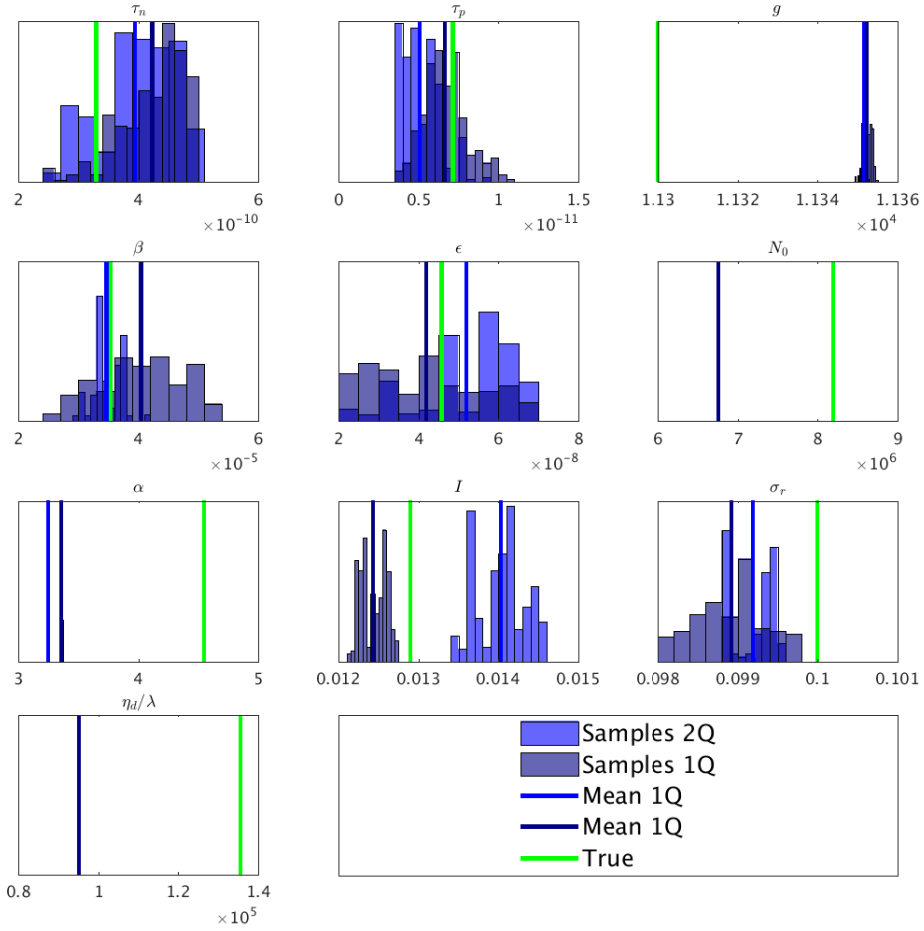


Figure 4.8: Overlap histograms of samples of NL-tracker using 1 or 2 quadratures.

CHAPTER 5

Experimental results

5.1 Introduction

Being the methods exposed in Chapters 2, 3 and 4 validated under a system simulation using MATLAB®, the whole set-up can be tested using experimental data. The aim is, given the digital data acquired from the output light power of a conventional diode laser, estimate the RIN, FN and, where using the non-linear tracking model, provide a characterization on the static parameters figuring in the rate-equation. The work-flow is basically the follow:

- Record the measurements and digitized them, at a given sampling frequency Δ_f ;
- Set-up a sampling algorithm to obtain the empirical posterior distribution of the parameter space;
- Use a point estimate for θ obtained from the histogram of samples and perform a tracking of the measurement set;
- Where possible, simulate a sequence of states from the extracted parameters;
- Estimate RIN and FN using tracking and simulation algorithms;

Each point will be treated in detail in the next section.

5.2 Laser set-up and measurement acquisition

What is represented in Figure 5.1 is a typical laser-detector scheme commonly used nowadays: a *digital coherent detection* scheme [49]. The emitted light from the laser is combined with another source called *local oscillator* (LO). The frequency of the resulting signal correspond to the difference $\Delta\omega$ of the single sources. This implies that the detected phase has a linear shift, which has to be compensated by including in the measurement equation a new parameter $\Delta\omega$

$$\mathbf{y}_t = E_t \begin{bmatrix} \cos(\Delta\omega t \Delta_t + \phi_t) \\ \sin(\Delta\omega t \Delta_t + \phi_t) \end{bmatrix} + \mathbf{r}_t \quad (5.1)$$

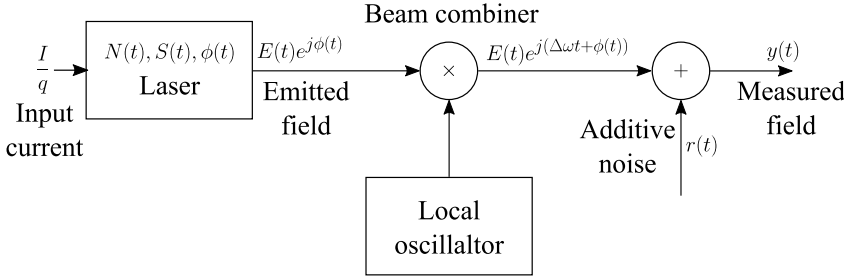


Figure 5.1: Schematic of a coherent detection system..

For the experimental setup, a single detected quadrature is acquired, having the following one-dimensional measurement equation

$$y_t = E_t \cos(\Delta\omega t \Delta_t + \phi_t) + r_t^i \quad (5.2)$$

The data collected from a laser is stored in a .mat file for MATLAB®, containing a vector measurements in double precision numerical format. When using the measurements to calculate the conventional spectrum estimate, $\Delta\omega$ was removed from the phase extracted by least square fitting a linear polynomial and removing the linear term from $\hat{\phi}$.

5.3 Methodology used and initialization

Two different measurement sets have been acquired from two different laser, the former with $\Delta_t = 25$ ps and the latter with $\Delta_t = 80$ ns. These huge time-step are not suitable for using a non-linear tracker with EM method. Therefore, S-ROCK approximation was necessary. For the tracking method and energy function evaluations, the EKF is preferred as faster than UKF. In Chapter 3 it was seen that EKF can be adapted both in a linear and non-linear dynamical model. Combining S-ROCK with EKF requires to calculate the Jacobian matrices on the S-ROCK approximation. The algorithm is similar to Algorithm 2, and works by propagating the Jacobian matrices during the Chebyshev stages. An alternative way for using EM with longer sampling times requires to work with a fraction $\Delta'_t = \Delta_t/k$ of the original sampling time. This is calculated such that it makes the numerical method stable. Then, in a tracking algorithm, the update step is executed every k iterations and for the rest $k - 1$ prediction are performed. The result then is downsampled to the original timestep. This method has been tested but it has not been presented here because of its poor efficiency. For a given sequence length, S-ROCK based tracker can process k times

many more measurement samples than a "delayed" tracker implementing EM. For the sampling algorithm, RWAMH is employed. The initial starting point $\theta^{(0)}$ is the same used for simulating the model in Chapter 2. For the linear tracker instead, $\theta^{(0)}$ is estimated from the acquired measurement, in a similar way as done in Chapter 4. The other parameters are substantially the same and described in Chapter 4 as well. The frequency shift $\Delta\omega$ is added to the list of sampled parameters. The value calculated in the previous section for it can be of service for initialize it. The choice of the Sampling and tracking method is based on the trade-off between the overall speed and the quality of results. Quality in this case, is assessed on the overall form of the estimated RIN/FN and on the samples obtained. This because when it comes to analyze experimental data there is not a ground truth to compare with the obtained spectra. The set of measurement used for sampling is normally a fraction of the total sequence used for the final tracking step. Here, from 10^6 total measurements available, the first 10^5 were used for sampling and the remaining for tracking.

5.4 RIN, FN and parameter estimation

Each EKF tracking method (linear and non-linear) is coupled with RWAMH to sample the parameters from the acquired measurements and computed the RIN/FN spectra on the mean estimate. Because such estimate for the NL-EKF tracker can also be used for simulating a laser system, a simulation of the RIN and FN is also presented, aside with the conventional spectra carried directly on the measurements.

For both lasers the characterization of the static parameters, presented in Figures 5.2 and 5.3 for the first laser and Figures 5.5 and 5.6 for the second one, looks compliant with the expectations carried on during the testing and validation session (Chapter 4). Both linear and non-linear tracker agree on the common parameters $\Delta\omega$ and σ_r , thus a good index of a successful sampling phase.

Results of tracking phase and amplitude and relative noise extraction can be seen in Figure 5.4 for the first laser and in Figure 5.7 for the second one. For the FN, the difference between the two tracking methods lies on the capability to filter out the measurement noise. Both follows the conventional estimate at lower frequency and subsequently lower down to clean more high frequency noise. The RIN estimation follows another trend, depending on the chosen tracker. The non-linear one, despite having the same shape of the linear counterpart, it actually underestimate the optical power leaving with a much lower RIN curve. This anomaly on the two trackers behavior has already been exposed in Chapter 3, and it is believed that the differences inside the tracking model itself are causing high dissimilarities on the spectra estimate. As foreseen during the testing phase, the trackers agree more on the phase rather than the amplitude noise. In fact, each sampling-tracking model tries its best to find sample by minimizing the energy function, which has a common structure for both. Therefore, the difference necessary relies on the way each tracker is able to interpret the received measurements. The simulated spectrum, presented aside the filtered ones, have been included for a comparison. However, its behavior does not

quite match either the conventional either the linear/non-linear RIN/NF estimates. The reason may be found on the fact that, as argued in Chapter 4, there are possibly many different local minima for the energy function - and therefore many different possible modes for the posterior distribution of the parameters. Just by giving a grasp look at the sampled particles, in more than one histograms it is possible to see a clue that multiple modes may exists for $p(\theta|\mathbf{y}_{1:N_T})$.

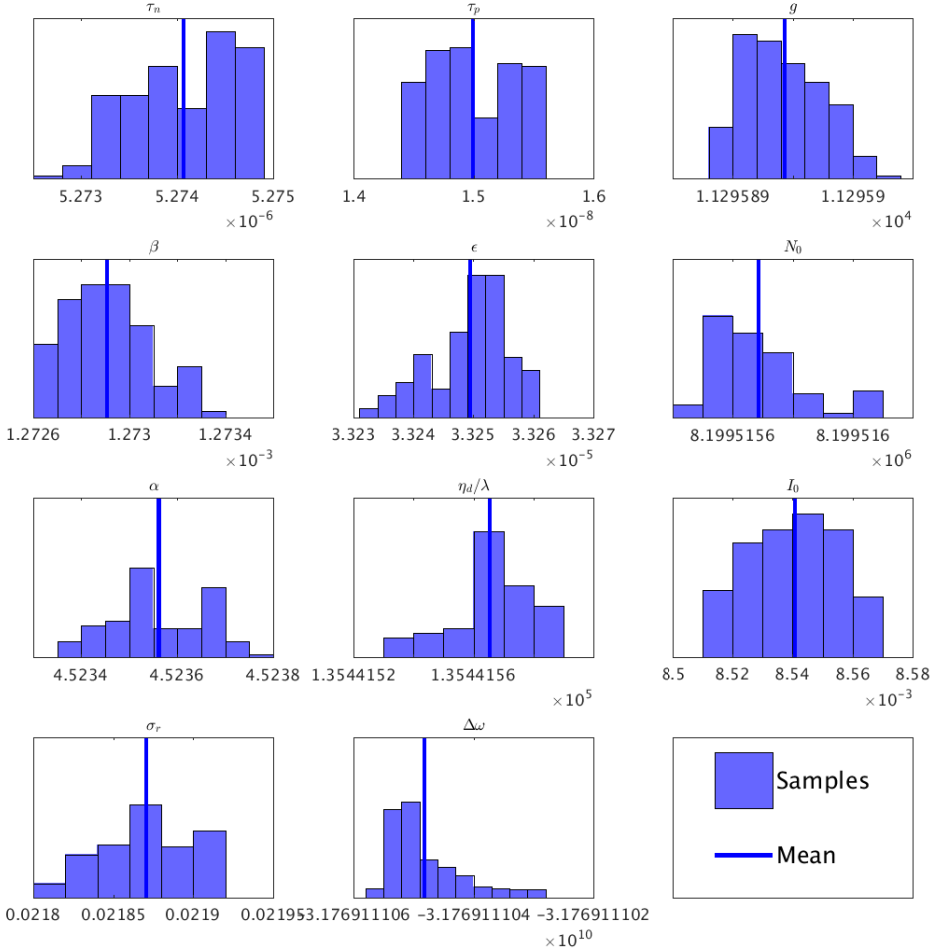


Figure 5.2: Samples of parameters on the first laser obtained with a non-linear tracker.

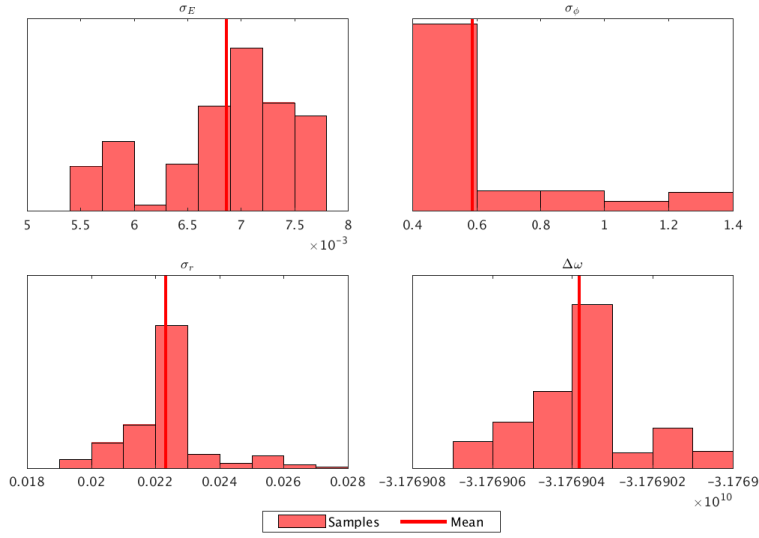


Figure 5.3: Samples of parameters on the first laser obtained with a linear tracker.

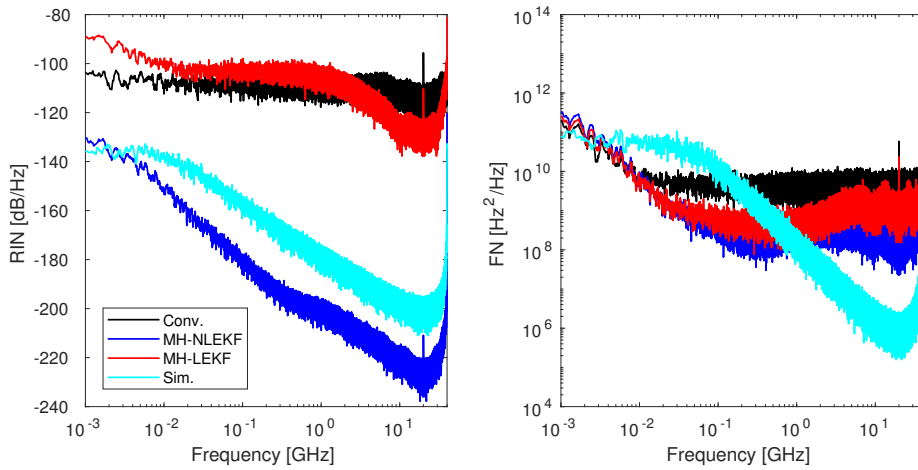


Figure 5.4: RIN/FN estimation on the first laser with the mean estimate from two different sampling methods.

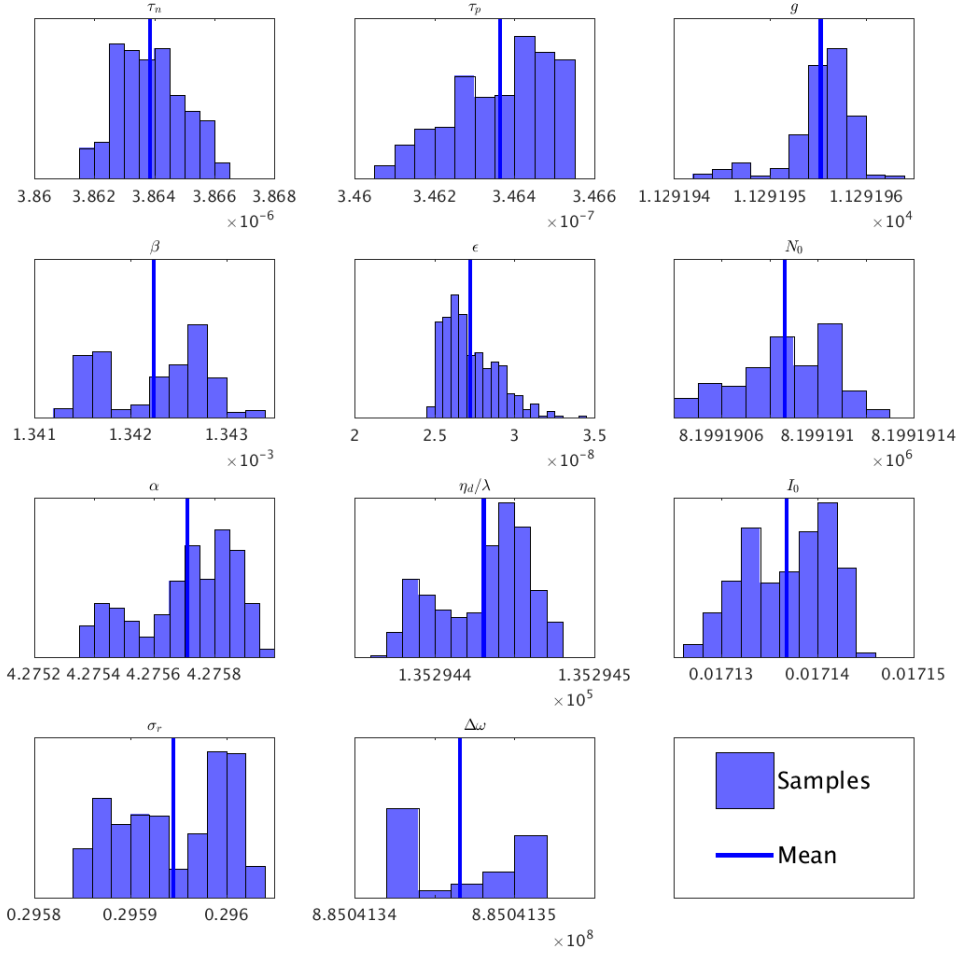


Figure 5.5: Samples of parameters on the second laser obtained with a non-linear tracker.

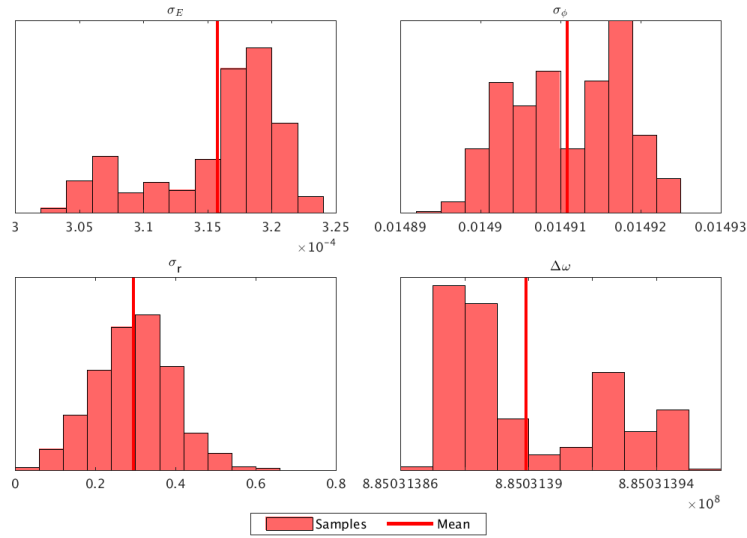


Figure 5.6: Samples of parameters on the second laser obtained with a linear tracker.

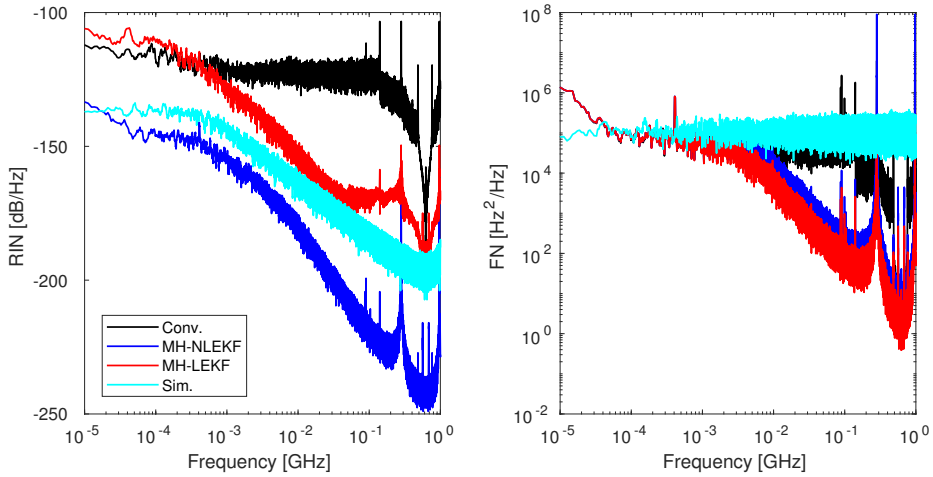


Figure 5.7: RIN/FN estimation on the second laser with the mean estimate from two different sampling methods.

CHAPTER 6

Conclusion

This thesis has shown the possibility of applying machine learning methods for lasers and nanolasers characteristic inference. Such methods, Kalman-based tracking and Metropolis-hasting based sampling required necessary attention to be implemented in this specific case. In Chapter 2 a new numerical solver S-ROCK has been introduced to solve the problem involving stiffness for simulating a rate-equation model described in the same chapter. This allowed the possibility to use a non-linear tracker (Chapter 3) on experimental data, where the sampling time has to be set according to the acquisition sampling frequency. EKF was validated as faster than UKF and accurate enough to be used as a tracking algorithm. In Chapter 4, a customized sampling method (RWAMH) was proposed to overcome some technical difficulties involving the classic Metropolis algorithm. This method allowed a good characterization of the parameters posterior distribution in terms of sampled particles, which can be used to extract a point estimate or to build continuous marginal densities over the parameters. Two basic tracking models (linear and non-linear) have been exposed and analyzed in a tracking (Chapter 3) and in a sampling (Chapter 4) setting. Both have been tested on collected experimental data (Chapter 5), showing promising results in terms of laser characterization. A full RIN/FN characterization for each laser tested has been achieved, coupled together with the relative model parameters uncertainty. More accuracy can be achieved by including any possible prior knowledge on both static and dynamic parameters. These results are the starting point to develop more accurate techniques aimed to improve the quality of the laser extracted characteristics. In particular, giving that this framework can be extended to other rate-equation based models, can be of use to validate such mathematical descriptions as well.

6.1 Future work

Possible extensions of the work done may involve the test and development of new models, both for the rate-equation and for the measurement function. For example, assuming Gaussian noise on the channel can sometimes be restrictive and not quite be representative of the reality. Additional models for the dynamics [50, 51]

Another key point would be to compare the characterization obtained from several different models to learn more on the true characteristics. Here, the linear and non-linear tracker had in common the measurement noise, the frequency shift and the optical phase. More models can have more common parameters, allowing a cross-

check to understand which tracker can be trusted and which not. This may include the possibility to directly interact during the sampling phase, letting different models interact with the cost or energy function used in MH-based algorithms.

On the hypothesis that the posterior on the parameters contains multiple modes, it is necessary to modifying the sampling algorithms to correctly characterize all possible modes [52, 53].

APPENDIX A

Appendix

Algorithm 7 Build a sigma set

Initialize. Assume a mean $\mathbf{m} \in \mathbb{R}^d$ and a covariance $\mathbf{P} \in \mathbb{R}^{d \times d}$ of the distribution, plus some scaling parameters $\alpha_{\text{ukf}}, \beta_{\text{ukf}}$ and κ_{ukf}

Calculate scaling

$$\lambda_{\text{ukf}} = \alpha_{\text{ukf}}^2(d + \kappa_{\text{ukf}}) - d \quad (\text{A.1})$$

Calculate weights

$$w_0^{(m)} = \lambda_{\text{ukf}} / (d + \lambda_{\text{ukf}}) \quad (\text{A.2})$$

$$w_0^{(c)} = \lambda_{\text{ukf}} / (d + \lambda_{\text{ukf}}) + (1 - \alpha_{\text{ukf}}^2 + \beta_{\text{ukf}}) \quad (\text{A.3})$$

for $i = 1$ to $2d$ **do**

$$w_i^{(m)} = w_i^{(c)} = 1 / (2d + 2\lambda_{\text{ukf}}) \quad (\text{A.4})$$

end for

Form the set

$$\mathbf{x}^{(0)} = \mathbf{m} \quad (\text{A.5})$$

for $i = 1$ to d **do**

$$\mathbf{x}^{(i)} = \mathbf{m} + \sqrt{d + \lambda_{\text{ukf}}} \left[\sqrt{\mathbf{P}} \right]_{:,i} \quad (\text{A.6})$$

end for

for $i = d + 1$ to $2d$ **do**

$$\mathbf{x}^{(i+n)} = \mathbf{m} - \sqrt{d + \lambda_{\text{ukf}}} \left[\sqrt{\mathbf{P}} \right]_{:,i} \quad (\text{A.7})$$

end for

return $\left(\left\{ \mathbf{x}^{(i)} \right\}_{i=0}^{2d}, \left\{ w_i^{(m)} \right\}_{i=0}^{2d}, \left\{ w_i^{(c)} \right\}_{i=0}^{2d} \right)$

Algorithm 8 UKF Unscented Kalman Filter

Initialize. Assume a prior distribution $p(\mathbf{x}_0) \sim \mathcal{N}(\mathbf{m}_0, \mathbf{P}_0)$ and the parameters for the sigma set $\alpha_{\text{ukf}}, \beta_{\text{ukf}}$ and κ_{ukf}

for $t = 0$ to $N_T - 1$ **do**

Prediction step. Using Algorithm 7, generate $\left\{ \tilde{\mathbf{x}}_{t-1}^{(i)}, w_i^{(m)'}, w_i^{(c)'} \right\}_{i=0}^{2d_{\mathbf{x}+\mathbf{q}}}$ using

$$\tilde{\mathbf{m}}_{t-1} = (\mathbf{m}_{t-1}, \mathbf{0}), \tilde{\mathbf{P}}_{t-1} = \begin{bmatrix} \mathbf{P}_{t-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{Q}_{t-1} \end{bmatrix}, d_{\mathbf{x}+\mathbf{q}} = d_{\mathbf{x}} + d_{\mathbf{q}}, \lambda'_{\text{ukf}}$$

for $i = 0$ to $2d_{\mathbf{x}+\mathbf{q}}$ **do**

Propagate

$$\hat{\mathbf{x}}_{t-1}^{(i)} = \mathbf{f} \left(\tilde{\mathbf{x}}_{t-1}^{(i)} \right) \quad (\text{A.8})$$

end for

Estimate mean and covariance of the transformation

$$\mathbf{m}_t^- = \sum_{i=0}^{2d_{\mathbf{x}+\mathbf{q}}} w_i^{(m)'} \hat{\mathbf{x}}_{t-1}^{(i)} \quad (\text{A.9})$$

$$\mathbf{P}_t^- = \sum_{i=0}^{2d_{\mathbf{x}+\mathbf{q}}} w_i^{(c)'} \left(\hat{\mathbf{x}}_{t-1}^{(i)} - \mathbf{m}_t^- \right) \left(\hat{\mathbf{x}}_{t-1}^{(i)} - \mathbf{m}_t^- \right)^\top \quad (\text{A.10})$$

Update step. Using Algorithm 7 generate $\left\{ \tilde{\mathbf{x}}_t^{-(i)}, w_i^{(m)''}, w_i^{(c)''} \right\}_{i=0}^{2d_{\mathbf{x}+\mathbf{q}}}$ using

$$\tilde{\mathbf{m}}_t^- = (\mathbf{m}_t^-, \mathbf{0}), \tilde{\mathbf{P}}_t^- = \begin{bmatrix} \mathbf{P}_t^- & \mathbf{0} \\ \mathbf{0} & \mathbf{R}_t \end{bmatrix}, d_{\mathbf{x}+\mathbf{r}} = d_{\mathbf{x}} + d_{\mathbf{r}}, \lambda''_{\text{ukf}}.$$

for $i = 0$ to $2d_{\mathbf{x}+\mathbf{r}}$ **do**

Propagate

$$\hat{\mathbf{y}}_t^{(i)} = \mathbf{h} \left(\tilde{\mathbf{x}}_t^{-(i)} \right) \quad (\text{A.11})$$

end for

Estimate mean, covariance and cross-covariance of the transformation

$$\boldsymbol{\mu}_t = \sum_{i=0}^{2d_{\mathbf{x}+\mathbf{r}}} w_i^{(m)''} \hat{\mathbf{y}}_t^{(i)} \quad (\text{A.12})$$

$$\mathbf{S}_t = \sum_{i=0}^{2d_{\mathbf{x}+\mathbf{r}}} w_i^{(c)''} \left(\hat{\mathbf{y}}_t^{(i)} - \boldsymbol{\mu}_t \right) \left(\hat{\mathbf{y}}_t^{(i)} - \boldsymbol{\mu}_t \right)^\top \quad (\text{A.13})$$

$$\mathbf{C}_t = \sum_{i=0}^{2d_{\mathbf{x}+\mathbf{r}}} w_i^{(c)''} \left(\tilde{\mathbf{x}}_t^{-(i), \mathbf{x}} - \mathbf{m}_t^- \right) \left(\hat{\mathbf{y}}_t^{(i)} - \boldsymbol{\mu}_t \right)^\top \quad (\text{A.14})$$

$$\mathbf{G}_t = \mathbf{C}_t \mathbf{S}_t^{-1} \quad (\text{A.15})$$

$$\mathbf{e}_t = (\mathbf{y}_t - \boldsymbol{\mu}_t) \quad (\text{A.16})$$

$$\mathbf{m}_t = \mathbf{m}_t^- + \mathbf{G}_t \mathbf{e}_t \quad (\text{A.17})$$

$$\mathbf{P}_t = \mathbf{P}_t^- - \mathbf{G}_t \mathbf{S}_t \mathbf{G}_t^\top \quad (\text{A.18})$$

end for

return $(\mathbf{m}_t, \mathbf{P}_t)$ for $t = 0 \dots N_T - 1$

Algorithm 9 RWAMH. Random Walk Adaptive Metropolis Hasting

Initialize. Initialize the parameters described in Table 4.1, the initial covariance $\Sigma^{(0)}$ and the starting point $\theta^{(0)}$

for $i = 1$ to N_{MH} **do**

 Sample a candidate point

$$\theta^* \sim \mathcal{N}(\theta^{(i-1)}, \Sigma^{(i-1)}) \quad (\text{A.19})$$

 Calculate the acceptance probability ξ_i

$$\xi_i := \min \left\{ 1, \exp \left(\varphi_T(\theta^{(i-1)}) - \varphi_T(\theta^*) \right) \right\} \quad (\text{A.20})$$

 Generate $u \sim \mathcal{U}[0, 1]$ and set the next point

$$\theta^{(i)} = \begin{cases} \theta^* & \text{if } u \leq \xi_i \\ \theta^{(i-1)} & \text{otherwise} \end{cases} \quad (\text{A.21})$$

 Update the average acceptance rate

$$N_{\text{ACC}} = N_{\text{ACC}} + \mathbb{1}[u \leq \xi_i] \quad (\text{A.22})$$

$$\bar{\xi} = N_{\text{ACC}}/i \quad (\text{A.23})$$

if $i \bmod U_{\text{MH}} = 0$ **then**

$$\Sigma_i = c_{\text{MH}} \cdot \text{COV}(\theta^{(i-H_{\text{MH}}+1:i)}) + \epsilon_{\text{MH}} \cdot \Sigma^{(0)} \quad (\text{A.24})$$

if $\bar{\xi} \geq \xi_d$ **then**

$$\Sigma^{(i)} = s_{\text{MH}} \Sigma^{(i)}$$

else

$$\Sigma^{(i)} = \frac{1}{s_{\text{MH}}} \Sigma^{(i)}$$

end if

else

$$\Sigma^{(i)} = \Sigma^{(i-1)}$$

end if

end for

return $\{\theta^{(i)}\}_{i=1}^{N_{\text{MH}}}$

Bibliography

- [1] Anton A. Huurdeman. *The Worldwide History of Telecommunication*. John Wiley & Sons, July 2003.
- [2] R. W. Tkach. “Scaling optical communications for the next decade and beyond”. In: *Bell Labs Technical Journal* 14.4 (Winter 2010), pages 3–9. ISSN: 1089-7089. DOI: 10.1002/bltj.20400.
- [3] D Law et al. *IEEE 802.3 Industry Connections Ethernet Bandwidth Assessment. IEEE 802.3 BWA Ad Hoc Report*. 2012.
- [4] Peter J Winzer. “Making spatial multiplexing a reality”. In: *Nature Photonics* 8.5 (2014), page 345.
- [5] Rod C Alferness, Herwig Kogelnik, and Thomas H Wood. “The evolution of optical systems: Optics everywhere”. In: *Bell Labs Technical Journal* 5.1 (2000), pages 188–202.
- [6] A. Shacham, K. Bergman, and L. P. Carloni. “Photonic Networks-on-Chip for Future Generations of Chip Multiprocessors”. In: *IEEE Transactions on Computers* 57.9 (September 2008), pages 1246–1260. ISSN: 0018-9340. DOI: 10.1109/TC.2008.78.
- [7] Sing H. Lee Jürgen Jahns. *Optical Computing Hardware*. Academic Press, 2014.
- [8] Amir H Atabaki et al. “Integrating photonics with silicon nanoelectronics for the next generation of systems on a chip”. In: *Nature* 556.7701 (2018), page 349.
- [9] Chen Sun et al. “Single-chip microprocessor that communicates directly using light”. In: *Nature* 528.7583 (2015), page 534.
- [10] Claus F. Klingshirn. *Semiconductor Optics*. Springer, 2012.
- [11] Qing Gu and Yeshaiahu Fainman. *Semiconductor Nanolasers*. Cambridge University Press, 2017. DOI: 10.1017/9781316275122.
- [12] C. Z. Ning. “Nanolasers, high speed modulation, and energy efficiency”. In: *2015 IEEE Summer Topicals Meeting Series (SUM)*. July 2015, pages 148–148. DOI: 10.1109/PHOSST.2015.7248239.
- [13] Weng W Chow, Frank Jahnke, and Christopher Gies. “Emission properties of nanolasers during the transition to lasing”. In: *Light: Science & Applications* 3.8 (2014), e201.

- [14] M. Wicks I. Fatadin D. Ives. "Numerical simulation of intensity and phase noise from extracted parameters for CW DFB lasers". In: *IEEE Journal of Quantum Electronics* 42.9 (September 2006), pages 934–941. ISSN: 0018-9197. DOI: 10.1109/JQE.2006.880117.
- [15] A. Villafranca et al. "Characterization of the Main Semiconductor Laser Static and Dynamic Working Parameters From CW Optical Spectrum Measurements". In: *IEEE Journal of Quantum Electronics* 43.2 (February 2007), pages 116–122. ISSN: 0018-9197. DOI: 10.1109/JQE.2006.886813.
- [16] Christopher M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.
- [17] Kajaree Das and Rabi Narayan Behera. "A survey on machine learning: concept, algorithms and applications". In: *International Journal of Innovative Research in Computer and Communication Engineering* 5.2 (2017), pages 1301–1309.
- [18] Darko Zibar and Christian G. Schäffer. "Machine learning concepts in coherent optical communication systems". In: *Proceedings of Signal Processing in Photonic Communications*. Optical Society of America (OSA), 2014. ISBN: 9781557527370.
- [19] Francesco Musumeci et al. "A Survey on Application of Machine Learning Techniques in Optical Networks". In: (March 2018).
- [20] Darko Zibar et al. "Machine learning techniques in optical communication". In: *Journal of Lightwave Technology* 34.6 (2016), pages 1442–1452. ISSN: 0733-8724. DOI: 10.1109/JLT.2015.2508502.
- [21] Darko Zibar, Henk Wymeersch, and Ilya Lyubomirsky. "Machine learning under the spotlight". In: *Nature Photonics* 11.12 (2017), pages 749–751. ISSN: 1749-4885. DOI: 10.1038/s41566-017-0058-3.
- [22] D. Zibar et al. "Bayesian filtering for phase noise characterization and carrier synchronization of up to 192 Gb/s PDM 64-QAM". In: *2014 The European Conference on Optical Communication (ECOC)*. September 2014, pages 1–3. DOI: 10.1109/ECOC.2014.6963844.
- [23] Darko Zibar Molly Piels Idelfonso Tafur Monroy. "Laser characterization with advanced digital signal processing". In: volume 9388. 2015, pages 9388 - 9388 - 8. DOI: 10.1117/12.2078335. URL: <https://doi.org/10.1117/12.2078335>.
- [24] R. Gordon Gould. "The LASER, Light Amplification by Stimulated Emission of Radiation". In: *The Ann Arbor Conference on Optical Pumping, the University of Michigan, 15 June through 18 June 1959*. Ann Arbor.
- [25] Milan L. Mašanović Larry A. Coldren Scott W. Corzine. *Diode Lasers and Photonic Integrated Circuits, 2nd Edition*. Wiley Online Library, March 2012. ISBN: 978-0-470-48412-8.
- [26] Fima C. Klebaner. *Introduction to Stochastic Calculus with Applications, 3rd ed.* Imperial College Press, March 2012.

- [27] Richard Durrett. *Stochastic Calculus: A Practical Introduction*. Edited by CRC Press. March 2018.
- [28] David Stewart Kendall E. Atkinson Weimin Han. *Numerical Solution of Ordinary Differential Equations*. John Wiley & Sons, Inc., October 2011. DOI: 10.1002/9781118164495.
- [29] K. Burrage, P. M. Burrage, and T. Tian. “Numerical methods for strong solutions of stochastic differential equations: an overview”. In: *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences* 460.2041 (2004), pages 373–402. ISSN: 1364-5021. DOI: 10.1098/rspa.2003.1247. eprint: <http://rspa.royalsocietypublishing.org/content/460/2041/373.full.pdf>. URL: <http://rspa.royalsocietypublishing.org/content/460/2041/373>.
- [30] A. Abdule and T. Li. “S-ROCK methods for stiff Ito SDEs”. In: *Commun. Math. Sci.* 6.4 (December 2008), pages 845–868. URL: <https://projecteuclid.org/443/euclid.cms/1229619673>.
- [31] Gerhard Wanner Ernst Hairer. *Solving Ordinary Differential Equations II*. Springer, 1997. DOI: 10.1007/978-3-642-05221-7.
- [32] Simo Särkkä. *Bayesian Filtering and Smoothing*. Cambridge University Press, September 2013.
- [33] Zhe Chen et al. “Bayesian filtering: From Kalman filters to particle filters, and beyond”. In: *Statistics* 182.1 (2003), pages 1–69.
- [34] Tobias Ryden Olivier Cappé Eric Moulines. *Inference in Hidden Markov Models*. Springer, April 2006.
- [35] Eamonn Keogh and Abdullah Mueen. “Curse of Dimensionality”. In: *Encyclopedia of Machine Learning and Data Mining*. Edited by Claude Sammut and Geoffrey I. Webb. Boston, MA: Springer US, 2017, pages 314–315. ISBN: 978-1-4899-7687-1. DOI: 10.1007/978-1-4899-7687-1_192. URL: https://doi.org/10.1007/978-1-4899-7687-1_192.
- [36] F. Gustafsson et al. “Particle filters for positioning, navigation, and tracking”. In: *IEEE Transactions on Signal Processing* 50.2 (February 2002), pages 425–437. ISSN: 1053-587X. DOI: 10.1109/78.978396.
- [37] Arnaud Doucet, Nando de Freitas, and Neil Gordon. “An Introduction to Sequential Monte Carlo Methods”. In: *Sequential Monte Carlo Methods in Practice*. Edited by Arnaud Doucet, Nando de Freitas, and Neil Gordon. New York, NY: Springer New York, 2001, pages 3–14. ISBN: 978-1-4757-3437-9. DOI: 10.1007/978-1-4757-3437-9_1. URL: https://doi.org/10.1007/978-1-4757-3437-9_1.
- [38] Kazufumi Ito. “Gaussian filter for nonlinear filtering problems”. In: *Decision and Control, 2000. Proceedings of the 39th IEEE Conference on*. Volume 2. IEEE. 2000, pages 1218–1223.

- [39] John H. Drew, Diane L. Evans, Andrew G. Glen, Lawrence M. Leemis. *Computational Probability*. Springer, Cham, 2008. DOI: 10.1007/978-3-319-43323-3.
- [40] S. J. Julier. “The scaled unscented transformation”. In: *Proceedings of the 2002 American Control Conference (IEEE Cat. No. CH37301)*. Volume 6. May 2002, 4555–4559 vol.6. DOI: 10.1109/ACC.2002.1025369.
- [41] H. M. T. Menegaz et al. “A Systematization of the Unscented Kalman Filter Theory”. In: *IEEE Transactions on Automatic Control* 60.10 (October 2015), pages 2583–2598. ISSN: 0018-9286. DOI: 10.1109/TAC.2015.2404511.
- [42] Andrew R Conn, Katya Scheinberg, and Luis N Vicente. *Introduction to derivative-free optimization*. Volume 8. Siam, 2009.
- [43] Luis Miguel Rios and Nikolaos V Sahinidis. “Derivative-free optimization: a review of algorithms and comparison of software implementations”. In: *Journal of Global Optimization* 56.3 (2013), pages 1247–1293.
- [44] Konstantin M Zuev and Lambros S Katafygiotis. “Modified Metropolis–Hastings algorithm with delayed rejection”. In: *Probabilistic Engineering Mechanics* 26.3 (2011), pages 405–412.
- [45] Heikki Haario, Eero Saksman, and Johanna Tamminen. “Componentwise adaptation for high dimensional MCMC”. In: *Computational Statistics* 20.2 (2005), pages 265–273.
- [46] Sean P Meyn and Richard L Tweedie. *Markov chains and stochastic stability*. Springer Science & Business Media, 2012.
- [47] Simon J Sheather and Michael C Jones. “A reliable data-based bandwidth selection method for kernel density estimation”. In: *Journal of the Royal Statistical Society. Series B (Methodological)* (1991), pages 683–690.
- [48] Nicolas Chopin. “Fast simulation of truncated Gaussian distributions”. In: *Statistics and Computing* 21.2 (2011), pages 275–288.
- [49] Ezra Ip et al. “Coherent detection in optical fiber systems”. In: *Opt. Express* 16.2 (January 2008), pages 753–791. DOI: 10.1364/OE.16.000753. URL: <http://www.opticsexpress.org/abstract.cfm?URI=oe-16-2-753>.
- [50] Jesper Mørk and G. L. Lippi. “Rate equation description of quantum noise in nanolasers with few emitters”. In: *Applied Physics Letters* 112.14 (2018). ISSN: 0003-6951. DOI: 10.1063/1.5022958.
- [51] A. Moelbjerg et al. “Dynamical Properties of Nanolasers Based on Few Discrete Emitters”. In: *IEEE Journal of Quantum Electronics* 49.11 (November 2013), pages 945–954. ISSN: 0018-9197. DOI: 10.1109/JQE.2013.2282464.
- [52] Radford M Neal. “Sampling from multimodal distributions using tempered transitions”. In: *Statistics and computing* 6.4 (1996), pages 353–366.

- [53] Farhan Feroz and MP Hobson. “Multimodal nested sampling: an efficient and robust alternative to Markov Chain Monte Carlo methods for astronomical data analyses”. In: *Monthly Notices of the Royal Astronomical Society* 384.2 (2008), pages 449–463.

