DUANE PETROVICH

SEQUENTIAL MONTE CARLO METHODS FOR PERSONAL POSITIONING

Master of Science Thesis

Subject approved by the Department Council on 7.12.2005
Examiners: Prof. Ioan Tabus
            Prof. Robert Piché
            Lic.Tech. Kari Heine
Preface

This thesis was written while working in the Personal Positioning Algorithms Research Group at Tampere University of Technology and was funded by Nokia Corporation. I would like to express my gratitude to prof. Robert Piché for keeping me employed and to prof. Ioan Tabus for participating in the examination of this thesis. Sincere thanks goes to Lic.Tech. Kari Heine for patience with all of my “quick” questions that went on for hours. Additional thanks goes to M.Sc. Simo Ali-Löytty, M.Sc. Niilo Sirola, and Henri Pesonen.

Tampere, 22th May 2006

Duane Petrovich
## Contents

Abstract v  
Symbols vi  
Abbreviations and Acronyms xi  

1 Introduction 1  
1.1 Organization .................................................. 2  

2 Bayesian Estimation 3  
2.1 Historical Notes .............................................. 4  
2.2 Mathematical Background ................................... 5  
2.3 Bayesian Inference ............................................ 7  
2.4 Probabilistic Models for Dynamic Systems ................. 8  
2.5 Recursive Bayesian Filtering ................................ 9  
2.6 Bayesian Filtering Algorithms .............................. 10  
2.6.1 Kalman Filter .............................................. 11  
2.6.2 Extended Kalman Filter ................................. 12  

3 SMC Approximation of Bayesian Filter 14  
3.1 Monte Carlo Background ..................................... 15  
3.1.1 Classical Monte Carlo .................................... 15  
3.1.2 Importance Sampling .................................... 16  
3.2 Particle Filtering ............................................ 19  
3.2.1 Sequential Importance Sampling ....................... 19  
3.2.2 Sequential Importance Sampling/Resampling ....... 23  
3.2.3 Examples of Importance Distributions ............... 25  
3.3 Auxiliary Particle Filters ................................... 26  
3.3.1 Empirical Mixtures and Auxiliary Variables .......... 27  
3.3.2 Examples of Importance Distributions for APF .... 31  
3.4 Resampling .................................................. 34  
3.4.1 Algorithms ................................................ 35  
3.4.2 Resampling Schedule .................................... 38
4 SMC for Personal Positioning
  4.1 Literature Review of Related SMC Applications .................................. 41
  4.2 Personal Positioning Background and Principles .................................... 41
    4.2.1 Measurements ................................................................................. 42
    4.2.2 Static Case ...................................................................................... 43
    4.2.3 Dynamic Case .................................................................................. 44
  4.3 Simulation Set 1: Testbench ...................................................................... 45
    4.3.1 Description ....................................................................................... 45
    4.3.2 Multiple Trajectories: EKF vs PF at Different Sample Sizes ............... 47
    4.3.3 Comparing PFs with Different Importance Distributions ................. 51
  4.4 Simulation Set 2: Comparing PF Posterior Approximations ....................... 54
    4.4.1 Comparing Distributions with $\chi^2$-tests ..................................... 54
    4.4.2 Probability Binning .......................................................................... 55
    4.4.3 Application to Linear and Gaussian Filtering Scenarios ................... 56
    4.4.4 Application to General Filtering Scenarios ...................................... 63

5 Conclusions
Abstract

The location of a mobile device can be estimated using signals from satellite and cellular network based systems. Positioning technologies have numerous applications and consequently, are an active area of research. Bayesian estimation is an attractive approach, but the models of measurements typically encountered in positioning result in an intractable solution. In addition, analytic approximations of the models, e.g. linearizations, can often be inadequate and lead to poor results. An alternative approach is numerical approximation.

In this thesis, we present a survey of sequential Monte Carlo (SMC) methods and illustrate their use in positioning. We include examples of well-known importance distributions, along with a brief discussion of resampling techniques. For completeness, this thesis also provides a review of the basics of Bayesian estimation and Monte Carlo based methods.

We have implemented and tested SMC methods in a number of simulations intended to resemble positioning environments. The improvement of SMC over an extended Kalman filter is most apparent when using only base station measurements and is not significant when using only satellite measurements. With fewer measurements, a larger sample size may be required for reliable estimates; for this reason, incorporating as much information as possible, e.g. base station sector and maximum range, is advised. Using our motion and measurement models, more computationally expensive importance distributions do not give significant improvements. Finally, the use of a multivariate binning technique is considered as a feasible alternative criterion to compare the performance of SMC.
Symbols

General

\( T \)  
Transpose of a vector or matrix

\( -1 \)  
Inverse of a matrix

\( \setminus \)  
Set difference

\( \subseteq \)  
Improper subset

\( \{ \ldots \} \)  
Unordered set

\( (\ldots) \)  
Ordered set

\( [\cdot] \)  
Floor function, i.e. rounding to nearest integer towards zero

\( \|\cdot\| \)  
Euclidean norm of a vector

\( 0 \)  
Vector of 0’s

\( A, B, C, \ldots \)  
Arbitrary sets

\( a, b, c, \ldots \)  
Arbitrary elements of a set

\( (a,b,c,\ldots) \)  
The ordered set \((a,b,\ldots,c)\)

\( a := b \)  
\( a \) is equal to \( b \) by definition

\( a \propto b \)  
\( a \) is proportional to \( b \)

\( a \in A, a \notin A \)  
\( a \) is in set \( A \), \( a \) is not in set \( A \)

\( A \times B \)  
Cartesian product of the sets \( A \) and \( B \)

\( B(\mathbb{R}^n) \)  
Borel algebra of subsets of \( \mathbb{R}^n \)

\( f : A \to B \)  
Function that maps elements from set \( A \) to set \( B \)

\( \mathbb{N} \)  
Set of all natural numbers including 0

\( \mathcal{P} \)  
Power set

\( \mathbb{R} \)  
Set of all real numbers

\( \mathbb{R}^n \)  
Set of all \( n \)-dimensional real numbers

\( \mathbb{R}^{m \times n} \)  
The set of all \( m \) by \( n \) matrices of real numbers
## Probability Theory

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>( (\Omega, \mathcal{F}, P) )</td>
<td>Probability space</td>
</tr>
<tr>
<td>( (\Omega, \mathcal{F}) )</td>
<td>Measurable space</td>
</tr>
<tr>
<td>( \chi^2, \chi^2_a )</td>
<td>Chi-squared distribution, chi-squared distribution with ( a ) degrees of freedom</td>
</tr>
<tr>
<td>( E[\mathbf{x}] )</td>
<td>Expectation of random variable ( \mathbf{x} )</td>
</tr>
<tr>
<td>( E_p[\mathbf{x}] )</td>
<td>Expectation of random variable ( \mathbf{x} ) having density ( p )</td>
</tr>
<tr>
<td>( \mathcal{F} )</td>
<td>Collection of events</td>
</tr>
<tr>
<td>( \mu, C )</td>
<td>Mean vector and covariance matrix</td>
</tr>
<tr>
<td>( \mathcal{N}(\mu, C) )</td>
<td>Normal distribution with mean ( \mu ) and covariance ( C )</td>
</tr>
<tr>
<td>( \nu(\mu, C) )</td>
<td>Normal density with mean ( \mu ) and covariance ( C )</td>
</tr>
<tr>
<td>( n_x, n_y )</td>
<td>Dimension of ( x ) and ( y )</td>
</tr>
<tr>
<td>( \Omega, \omega )</td>
<td>Sample space, sample point</td>
</tr>
<tr>
<td>( P(A) )</td>
<td>Probability of event ( A \in \mathcal{F} )</td>
</tr>
<tr>
<td>( P_\mathbf{x} )</td>
<td>Probability distribution of random variable ( \mathbf{x} )</td>
</tr>
<tr>
<td>( P_\mathbf{x} \mid \mathbf{y} )</td>
<td>Probability distribution of random variable ( \mathbf{x} ) given ( \mathbf{y} = y )</td>
</tr>
<tr>
<td>( p_\mathbf{x} )</td>
<td>Probability density of random variable ( \mathbf{x} )</td>
</tr>
<tr>
<td>( p_\mathbf{x} \mid \mathbf{y} )</td>
<td>Conditional probability density of random variable ( \mathbf{x} ) given ( \mathbf{y} )</td>
</tr>
<tr>
<td>( \psi )</td>
<td>Test statistic</td>
</tr>
<tr>
<td>( \mathcal{U}_{[a,b]} )</td>
<td>Uniform distribution over interval ([a, b])</td>
</tr>
<tr>
<td>( \mathbf{x} \sim P )</td>
<td>Random variable ( \mathbf{x} ) is distributed according to ( P )</td>
</tr>
<tr>
<td>( \mathbf{x} \sim \mathbf{p} )</td>
<td>( \mathbf{x} ) is a realization of random variable ( \mathbf{x} ) with distribution ( \mathbf{p} )</td>
</tr>
<tr>
<td>( \mathbf{V}[\mathbf{x}] )</td>
<td>Variance of random variable ( \mathbf{x} )</td>
</tr>
<tr>
<td>( \mathbf{V}_p[\mathbf{x}] )</td>
<td>Variance of random variable ( \mathbf{x} ) having density ( p )</td>
</tr>
</tbody>
</table>
Bayesian Estimation

\[ f \] Signal model function, \( f : \mathbb{R}^{n_x} \rightarrow \mathbb{R}^{n_x} \)

\[ F \] Matrix defining the linear signal model function, \( F \in \mathbb{R}^{n_x \times n_x} \)

\[ \hat{F} \] Jacobian matrix of signal model function \( f \)

\[ \gamma \] Power spectral density of signal noise

\[ h \] Observation model function, \( h : \mathbb{R}^{n_x} \rightarrow \mathbb{R}^{n_y} \)

\[ H \] Matrix defining the linear observation model function, \( H \in \mathbb{R}^{n_y \times n_x} \)

\[ \hat{H} \] Jacobian matrix of observation model function \( h \)

\[ K \] Kalman gain

\[ \pi_{a:b|c} \] Conditional distribution of the signal process \( x_{a:b} \) given realization of the observation process \( y_{1:c} = y_{1:c} \)

\[ \pi_{b|c} \] Marginal conditional distribution of the signal process \( x_b \) given realization of the observation process \( y_{1:c} = y_{1:c} \)

\[ \Sigma_v, \Sigma_w \] Covariance matrix of a signal noise and observation noise process

\[ \sigma^2 \] Variance (scalar) of observation noise

\[ V \] Signal noise process

\[ W \] Observation noise process

\[ X \] Signal process

\[ Y \] Observation process
Monte Carlo

$\beta^i$ The $j^{th}$ unnormalized first-stage weight
$\tilde{\beta}^i$ The $j^{th}$ normalized first-stage weight
$\delta_x$ Delta-Dirac unit mass located at $x$
$N$ The number of samples
$N_{\text{eff}}$ Effective sample size
$\tilde{N}_{\text{eff}}$ Approximation of effective sample size
$M$ The number of importance samples
$\hat{\mu}, \hat{C}$ The realized SMC estimate of mean $\mu$ and covariance $C$, respectively
$\tilde{\pi}_{a,b|c}^N$ The discrete weighted approximation of $\pi_{a,b|c}$, based on $N$ samples
$P, p$ Arbitrary target distribution and density, respectively
$P^N$ The discrete equally-weighted approximation of distribution $P$, based on $N$ samples
$\tilde{P}^N$ The discrete weighted approximation of distribution $P$, based on $N$ samples
$\tilde{p}^i$ The $j^{th}$ linearized model
$\tilde{P}, \tilde{p}$ Empirical approximation of distribution $P$ and density $p$, respectively
$Q, q$ Arbitrary importance distribution and density, respectively
$Q^j$ The $j^{th}$ distribution component of mixture importance distribution
$w(\cdot)$ Unnormalized importance weight function
$\tilde{w}^i$ Normalized importance weight of $\tilde{x}^i$
$\tilde{x}^i$ The $i^{th}$ importance sample of $x$
$x^i$ The $i^{th}$ unweighted sample of $x$
## Positioning

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$b, \dot{b}$</td>
<td>Bias of pseudorange, delta pseudorange measurement</td>
</tr>
<tr>
<td>$\Delta t_k$</td>
<td>Time difference, $t_k - t_{k-1}$</td>
</tr>
<tr>
<td>$r^b$</td>
<td>Position of a base station</td>
</tr>
<tr>
<td>$r^s, u^s$</td>
<td>Position, velocity of a satellite</td>
</tr>
<tr>
<td>$r^{si}, u^{si}$</td>
<td>Position, velocity of the $i^{th}$ satellite</td>
</tr>
<tr>
<td>$r, u$</td>
<td>User position, velocity</td>
</tr>
<tr>
<td>$V(\cdot)$</td>
<td>Loss Function</td>
</tr>
</tbody>
</table>
## Abbreviations and Acronyms

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AOA</td>
<td>Angle of Arrival</td>
</tr>
<tr>
<td>APF</td>
<td>Auxiliary Particle Filter</td>
</tr>
<tr>
<td>BS</td>
<td>Base Station</td>
</tr>
<tr>
<td>CDF</td>
<td>Cumulative Distribution Function</td>
</tr>
<tr>
<td>CEP</td>
<td>Circular Error Probability</td>
</tr>
<tr>
<td>EKF</td>
<td>Extended Kalman Filter</td>
</tr>
<tr>
<td>ENU</td>
<td>East-North-Up</td>
</tr>
<tr>
<td>GPS</td>
<td>Global Positioning System</td>
</tr>
<tr>
<td>IID</td>
<td>Independent and Identically Distributed</td>
</tr>
<tr>
<td>IS</td>
<td>Importance Sampling</td>
</tr>
<tr>
<td>MC</td>
<td>Monte Carlo</td>
</tr>
<tr>
<td>MCMC</td>
<td>Markov Chain Monte Carlo</td>
</tr>
<tr>
<td>MS</td>
<td>Mobile Station</td>
</tr>
<tr>
<td>MVU</td>
<td>Minimum Variance Unbiased (estimator)</td>
</tr>
<tr>
<td>PF</td>
<td>Particle Filter</td>
</tr>
<tr>
<td>RMSE</td>
<td>Root Mean Square Error</td>
</tr>
<tr>
<td>RSS</td>
<td>Received Signal Strength</td>
</tr>
<tr>
<td>RTD</td>
<td>Round Trip Delay</td>
</tr>
<tr>
<td>SIR</td>
<td>Sampling Importance Resampling</td>
</tr>
<tr>
<td>SIS</td>
<td>Sequential Importance Sampling</td>
</tr>
<tr>
<td>SISR</td>
<td>Sequential Importance Sampling/Resampling</td>
</tr>
<tr>
<td>SMC</td>
<td>Sequential Monte Carlo</td>
</tr>
<tr>
<td>SV</td>
<td>Space Vehicle</td>
</tr>
<tr>
<td>TOA</td>
<td>Time of Arrival</td>
</tr>
</tbody>
</table>
Chapter 1

Introduction

The problem of determining one’s location from measurements is an age-old problem that is still a relevant research topic today. Satellite-based systems, e.g. Global Positioning System (GPS), can be effective in providing accurate position estimates, but have difficulty in urban areas or indoors due to attenuated signals. Local wireless networks, e.g. cellular networks, offer positioning capabilities, but typically with less accuracy than that of satellite-based systems. Different systems have their strengths and weaknesses, and a given system is generally not suitable for all possible scenarios. Positioning methods that are able to combine measurements from multiple sources, e.g. both satellite systems and cellular networks, are therefore advantageous.

The Bayesian formulation of the positioning problem conveniently allows measurements from different sources to be incorporated in the estimation of position coordinates. This approach inherently involves treating uncertainty as probability distributions. Theoretically, we have an optimal probability distribution that describes everything we know about the position coordinates having received the measurements. Unfortunately, the models of the measurements commonly used in positioning result in an intractable optimal solution.

One approach is to use analytic approximations of the true models so that the Bayesian solution becomes tractable. However, this can perform quite poorly if the approximations are inadequate. Another approach is to solve the original, intractable problem by numerical approximation. Monte Carlo based methods, also known as sampling based methods, are computational methods that approximate distributions with empirically simulated samples. Gelfand and Smith [1990] have bluntly stated that “…the attraction of the sampling-based methods is their computational simplicity and ease of implementation for users with computing resources but without numerical analytic expertise.”
CHAPTER 1. INTRODUCTION

Sequential Monte Carlo methods, also known as particle filters, are a class of techniques to approximate the optimal probability distribution recursively over time. A practical Monte Carlo algorithm for filtering was introduced about a decade ago and since then, the methods have been the focus of a great deal of research. In this work, we review some of the theory behind sequential Monte Carlo methods and illustrate their use in positioning.

1.1 Organization

This work is organized as follows. Bayesian estimation is discussed in Chapter 2. We first introduce the ideas for the stationary case, and then extend the discussion to dynamic systems. We describe two Bayesian algorithms, one optimal and one suboptimal, that will be used later in the examples and simulations. Throughout this work, we have avoided discussion of the construction of the model and have focused on the estimation task given the models.

Sequential Monte Carlo is introduced in Chapter 3 and we present the methods in two separate formulations. The choice of importance distribution is emphasized as a design parameter and we give a number of examples. We conclude this chapter with a brief introduction to some resampling techniques.

Having covered the basics of Bayesian estimation and the sequential Monte Carlo approximation of the Bayesian filter, we apply the concepts to the positioning problem in Chapter 4. We present empirical results illustrating filter performance in different positioning environments and also comparing different importance distributions. In addition, we apply a multivariate binning technique to the comparison of sequential Monte Carlo methods. We give some concluding remarks in Chapter 5.
Chapter 2

Bayesian Estimation

Our topic of interest is estimating the state of a system from noisy observations. Kay [1993] identifies two approaches to estimation: classical and Bayesian. In classical estimation, the state is assumed to be a deterministic but unknown constant. One is then typically interested in the estimator that is unbiased with minimum variance—the MVU estimator. On the other hand, Bayesian estimation assumes the unknown state is a random variable with a particular realization that we are to estimate. It earns the name Bayesian because the implementation is based on Bayes’ theorem. One motivation for the Bayesian approach is the convenient use of prior knowledge, which is not straightforward to incorporate into classical estimation. Another motivation is that in situations when the MVU estimator cannot be found, the Bayesian approach provides a framework to find an estimator that performs well on the average [Kay 1993, pg. 309].

In this work, we will focus exclusively on Bayesian estimation. We start in Section 2.2 by briefly covering the necessary mathematical concepts and notation. In Section 2.3, we give a general introduction of the Bayesian framework describing the basic principles for a static system. We work our way towards applying this methodology to dynamic systems by first, introducing the needed probabilistic models in Section 2.4 and then presenting the Bayesian recursion for dynamic systems in Section 2.5. Finally, in Section 2.6, we present some well-known algorithms for implementing the Bayesian solution in practice.
2.1 Historical Notes

The origins of modern estimation theory are usually traced back to the invention of least squares. Around 200 years ago, Legendre and Gauss independently invented and applied the method to determining the orbits of celestial bodies. In Legendre’s approach, there was no probabilistic interpretation and the proposal was just a convenient method that minimized a function of errors. It is generally accepted that Gauss and Laplace are responsible for the development of the theory and a probabilistic interpretation. [Stigler 1981], [Merriman 1877]

During the late 1930s and early 40s, Kolmogorov and Wiener independently proposed theory for the linear prediction of processes with statistical properties that do not change in time, i.e. stationary processes. Wiener also worked on filtering problems and, using frequency domain techniques, was able to formulate the optimum linear estimate of a stationary process corrupted by additive noise. It was not until 1960, motivated by the coming space age and by the inadequacy of the Wiener filter to deal with nonstationary problems, that the linear, nonstationary filtering problem was solved by Kalman. [Haykin 1991], [Jazwinski 1970]

The history of nonlinear filtering is not as successful and is still a field of active research. During the 60s and into the 70s, nonlinear filters relying on analytical approximations began to appear—most notably, the extended Kalman filter and Gaussian sum filter, e.g. Alspach and Sorenson [1972]. In the early 1980s, Beneš published the optimal solution for a special class of nonlinear filtering problems—a type of saturating process, see e.g. Farina et al. [2002]. This was later generalized by Daum but there still is no optimal solution for all nonlinear problems. [Daum 2005], [Anderson and Moore 1979]

Today, the state of the art in nonlinear filters includes a number of computational methods, e.g. Monte Carlo based methods and numerical solutions for partial differential equations, as well as new variations of the Kalman filter, e.g. the unscented Kalman filter [Daum 2005].
2.2 Mathematical Background

In this work, an underlying probability space \((\Omega, \mathcal{F}, P)\) is assumed, where the sample space \(\Omega\) defines all possible outcomes, the \(\sigma\)-algebra \(\mathcal{F}\) on \(\Omega\) is the collection of events of interest, and the probability measure \(P\) on the measurable space \((\Omega, \mathcal{F})\) assigns a probability to each of these events. Let \(\mathcal{B}(\mathbb{R}^n)\) denote the collection of Borel sets of \(\mathbb{R}^n\).

**Definition 2.1.** Let \(x\) be a random variable defined on some probability space \((\Omega, \mathcal{F}, P)\) such that \(x : \Omega \rightarrow \mathbb{R}^n\). The induced probability measure \(P_x\) is referred to as the **probability distribution** of \(x\) and is defined as

\[
P_x(A) := P(x \in A), \quad A \in \mathcal{B}(\mathbb{R}^n),
\]

where we use the common shorthand notation

\[
P(x \in A) := P(\{\omega \in \Omega \mid x(\omega) \in A\}).
\]

Similarly, the **conditional probability distribution** \(P_{x|y}\) of the random variable \(x\) given the event \(y = y\) is defined such that

\[
P_{x|y}(A) := P(x \in A \mid y = y), \quad A \in \mathcal{B}(\mathbb{R}^n).
\]

Regarding notation, random variables will be denoted in bold font, i.e. \(\mathbf{x}\), while realizations of random variables will be in plain font, i.e. \(x\), and there is no explicit notation for vectors. Note that both of the probability measures of Definition 2.1 are deterministic. We will avoid discussion of the random measures generated by conditioning on a \(\sigma\)-algebra, see e.g. Crisan [2001], as it is outside the scope of this work.

In this work, a probability density \(p_x\) of a distribution \(P_x\) and a conditional probability density \(p_{x|y}(\cdot|y)\) of a conditional distribution \(P_{x|y}\) are assumed to exist without explicit notation and satisfy the following

\[
P_x(A) = \int_A p_x(x) \, dx
\]

\[
P_{x|y}(A) = \int_A p_{x|y}(x|y) \, dx,
\]

where again \(A \in \mathcal{B}(\mathbb{R}^n)\). Subscripts on the density may be omitted if the random variable in question is obvious, e.g. \(p(x|y)\) instead of \(p_{x|y}(x|y)\).

The Lebesgue integral, see e.g. Gariepy and Ziemer [1995, pg. 133-134], provides a generalization of the Riemann integral that will be useful for our purposes. Expectation of some function \(f\) of a random variable is defined using the (Lebesgue) integral with respect to a probability measure \(P\), i.e.

\[
E[ f(x) ] := \int f(x) \, P(dx), \quad (2.1)
\]
where the function \( f \) is assumed to be \( P \)-integrable.

The usefulness of the Lebesgue integral in (2.1) can be illustrated as follows. If the probability measure \( P \) has a density \( p \) (with respect to the Lebesgue measure), then (2.1) is taken to be

\[
E[f(x)] = \int f(x) \, p(x) \, dx.
\]

If the probability measure \( P \) is a discrete function taking values \( p^i \) at the points \( x^i \) for \( i = 1, \ldots, N \), then (2.1) is taken to be

\[
E[f(x)] = \sum_{i=1}^{N} f(x^i) \, p^i.
\]

In this work, Lebesgue integration is used so that expectations of continuous and discrete random variables are under the same framework.

Maybeck [1979, pg. 78-80] derives various forms of a conditional probability density, alternatively known as Bayes’ rule. The result is derived as a limit because conditioning on an event with zero probability, e.g. \( y = y \) when \( y \) is continuous, can be problematic.

**Definition 2.2.** Various forms of Bayes’ rule for (continuous or discrete) random variables are given by

\[
p(x|y) = \frac{p(x,y)}{p(y)} = \frac{p(y|x) \, p(x)}{p(y)},
\]

where \( p(y) \) is a nonzero normalization constant given by

\[
p(y) = \int p(x,y) \, dx = \int p(y|x) \, p(x) \, dx.
\]

We will often need to discuss a time-sequence of random variables, and so we take the following definition from Maybeck [1979, pg. 133].

**Definition 2.3.** Let \( \Omega \) be a sample space and \( T \subseteq \mathbb{R} \). Then a stochastic or random process \( X \) is a real function defined on the product space \( \Omega \times T \) such that for any \( t \in T \), \( X(\cdot, t) =: x_t \) is a random variable.

The stochastic process may also be denoted by \( X = (x_t)_{t \in T} \). If \( T \) is discrete, as it will always be in this work, then we can refer to the stochastic process as a stochastic sequence and explicitly denote this by using a time index \( k \). A process that we will encounter quite frequently is the Markov process.
Definition 2.4. A stochastic process \( X = (x_k)_{k \in \mathbb{N}} \) is a Markov process (of order one) if for all \( k \in \mathbb{N}\setminus\{0\} \),

\[
p(x_k | x_0, x_1, \ldots, x_{k-1}) = p(x_k | x_{k-1}).
\]

Such Markov processes are completely defined by the initial distribution \( P_{x_0} \) and the conditional distributions \( P_{x_k|x_{k-1}} \), also called transition kernels [Robert and Casella 1999, pg. 141].

We do not explicitly define the distribution of a finite stochastic sequence because it can be considered as the distribution of a random variable, i.e. a stacked vector of all the individual random variables in time.

### 2.3 Bayesian Inference

We often have a quantity that we are interested in estimating and want to make some sort of inference on this quantity based on observed data. **Bayesian inference** refers to the use of probability statements to combine prior knowledge with the information gained from observing data. This inherently involves treating uncertainty and belief as probability distributions—a fundamental characteristic of the Bayesian philosophy.

Let us denote our quantity of interest \( x \) and the observed data \( y \), which are considered to be realizations of the random variables \( x \) and \( y \) respectively. The term “hidden” is often used when referring to \( x \) to emphasize that we infer information only through measurements, i.e. \( x \) itself is unobserved. We start with a priori knowledge of our quantity of interest. This represents what we know about \( x \) before any data is received and is given in terms of a probability distribution with density \( p(x) \). Next, we need a model that relates the observed data to the hidden quantity. The observations are assumed to be from some distribution that depends on our quantity; if the observations were independent of our quantity then there would be no inference possible. Let \( p(y|x) \) denote both the conditional density of the observations and the likelihood function, depending on the argument\(^1\).

Using Bayes’ rule for random variables, we have

\[
p(x|y) \propto p(y|x) p(x).
\]

\(^1\)Note that \( p_{y|x}(y|x) \) is a function of both \( x \) and \( y \). If we fix \( x \) and treat \( p_{y|x} \) as a function of \( y \), it is a density. But, if we fix \( y \) and treat \( p_{y|x} \) as a function of \( x \), then it is called the likelihood function and does not necessarily have the properties of a density, e.g. integrating to unity [Lee 1997, pg. 34].
The conditional distribution having conditional density \( p(x|y) \) represents our a posteriori knowledge in that it embodies all available statistical information and is the complete solution to the problem based on the initial assumptions [Lee 1997]. We will use the term prior to refer to both the distribution and density of the a priori knowledge, and similarly use the term posterior to refer to both the distribution and density of the a posteriori knowledge.

We should emphasize that, in practice, the motivating factor for a Bayesian approach is typically the various estimates that can be obtained from the posterior, i.e. not the posterior distribution itself. In Bayesian estimation, we use the posterior distribution to form estimates, e.g. by taking expectations with respect to the posterior distribution or by maximizing the posterior density. It is for these reasons that we are interested in determining the posterior distribution.

### 2.4 Probabilistic Models for Dynamic Systems

Consider now a time-sequence of values we are interested in estimating \((x_k)_{k=0}^{\infty}\) and a corresponding time-sequence of measurements \((y_k)_{k=1}^{\infty}\). Recall that in Bayesian estimation, the unknown value we want to estimate, as well as the observations, are considered to be realizations of random variables. To apply the Bayesian framework to this estimation, we must define these random variables in terms of probabilistic models. This section formulates the time-sequence estimation task in such a framework so that we can proceed to present the Bayesian solution.

We define two stochastic processes \( X = (x_k)_{k=0}^{\infty} \) and \( Y = (y_k)_{k=1}^{\infty} \), where \( x_k \in \mathbb{R}^n_x \) and \( y_k \in \mathbb{R}^n_y \). The process \( X \) is called the unobserved (hidden) signal process and is modelled as a (first-order) Markov process defined by an initial distribution \( P_{x_0} \) and transition kernel \( P_{x_k|x_{k-1}} \), i.e.

\[
x_0 \sim P_{x_0}, \quad x_k \mid x_{0:k-1}, y_{1:k-1} \sim P_{x_k|x_{k-1}}.
\]  \hspace{1cm} (2.3)

The process \( Y \) is called the observation process and is modelled as

\[
y_k \mid x_{0:k}, y_{1:k-1} \sim P_{y_k|x_k}.
\]  \hspace{1cm} (2.4)

We will refer to (2.3) as the signal model, as it describes how the signal evolves with time, and (2.4) as the observation model, as it defines how the observations are related to the signal.


2.5 Recursive Bayesian Filtering

We will often use the term filtering to refer to a specific type of estimation task when discussing time-sequence processing; in particular, to distinguish it from smoothing and prediction. Recall that we have a time-sequence of values we are interested in estimating \( x_k \) \( k=0 \) and a corresponding time-sequence of measurements \( y_k \) \( k=1 \). In this context, filtering refers to the estimation of \( x_k \) using all the information received up to time \( k \), i.e. \( \{y_1, y_2, ..., y_k\} \). In smoothing, we use measurements received after \( k \) and in prediction, we use only measurements available before \( k \). [Anderson and Moore 1979, pg. 10-11]

Using the Bayesian framework, we will form the probability distribution of the signal process conditioned on the realization of the observation process. Let the conditional distribution of the signal be denoted as

\[
\pi_{x|y} := P_{x_{a:b}|y_{a:c}}.
\]

At any time \( k \), the density of the joint posterior distribution \( \pi_{0:k|k} \) is given by Bayes’ rule as

\[
p(x_{0:k}|y_{1:k}) = \frac{p(y_{1:k}|x_{0:k})p(x_{0:k})}{p(y_{1:k})},
\]

where the denominator is a normalizing constant independent of \( x_{0:k} \) given by

\[
p(y_{1:k}) = \int p(y_{1:k}|x_{0:k})p(x_{0:k}) \, dx_{0:k} = \int \ldots \int p(y_{1:k}|x_{0:k})p(x_{0:k}) \, dx_0 \, dx_1 \ldots dx_k.
\]

However, it is more useful in practice to have the density of the marginal or filtering distribution \( \pi_{k|k} \) and form a recursion that obtains this solution from the previous solution \( \pi_{k-1|k-1} \) in two stages: prediction and update.

Suppose at time \( k-1 \), the density for the posterior \( \pi_{k-1|k-1} \) is available; this embodies everything we know about \( x_{k-1} \), i.e. the state of the signal at time \( k-1 \), having received all the measurements \( y_1, \ldots, y_{k-1} \). The prediction stage consists of applying the signal model (2.3) to this density to describe what we know about \( x_k \), i.e. the state of the signal at time \( k \), before receiving any data from time \( k \). The prediction distribution \( \pi_{k|k} \) embodies this information and has a density given via [Jazwinski 1970, pg. 174]

\[
p(x_k|y_{1:k-1}) = \int p(x_k, x_{k-1}|y_{1:k-1}) \, dx_{k-1} = \int p(x_k|x_{k-1}, y_{1:k-1})p(x_{k-1}|y_{1:k-1}) \, dx_{k-1} = \int p(x_k|x_{k-1})p(x_{k-1}|y_{1:k-1}) \, dx_{k-1}.
\]
The Markov assumption of the signal was used in the third equality. Equation (2.6) is often referred to as the Chapman-Kolmogorov equation.

When the measurement at time $k$ becomes available, the update stage then uses Bayes’ rule with the measurement model (2.4) to obtain the density of the posterior $p_{k|k}$ given by [Jazwinski 1970, pg. 174]

$$p(x_k|y_{1:k}) = \frac{p(y_k|x_k) p(x_k|y_{1:k-1})}{p(y_k|y_{1:k-1})},$$

where the denominator is a normalizing constant independent of $x_k$ given by

$$p(y_k|y_{1:k-1}) = \int p(y_k|x_k) p(x_k|y_{1:k-1}) \, dx_k.$$

Common estimates used in Bayesian filtering include the conditional mean and covariance, which are expectations with respect to the probability measure $p_{k|k}$ and are given respectively by

$$\mu_k = \mathbb{E}[x_k | y_{1:k}] = \int x_k p(x_k|y_{1:k}) \, dx_k$$
$$C_k = \mathbb{E}[(x_k - \mu_k)(x_k - \mu_k)^T | y_{1:k}] = \int (x_k - \mu_k)(x_k - \mu_k)^T p(x_k|y_{1:k}) \, dx_k.$$

### 2.6 Bayesian Filtering Algorithms

We shift from our general discussion of the models in probabilistic terms to introduce an example stochastic system. These system equations will aid in presenting the algorithms. Let

$$x_k = f_{k-1}(x_{k-1}) + v_{k-1}$$
$$y_k = h_k(x_k) + w_k,$$

where $V = (v_k)_{k=0}^\infty$ and $W = (w_k)_{k=1}^\infty$ are the signal noise and observation noise processes, respectively. The subscript on the functions $f_{k-1}$ and $h_k$ indicates a possible dependence on time. To emphasize the relation of the system equations to the probabilistic models of Section 2.4, we can write the following

$$p(x_k|x_{k-1}) = p_{v_{k-1}}(x_k - f_{k-1}(x_{k-1}))$$
$$p(y_k|x_k) = p_{w_k}(y_k - h_k(x_k)),$$

where $p_{v_{k-1}}$ is the density of $v_{k-1}$ and $p_{w_k}$ is the density of $w_k$. 
By imposing certain restrictions on the models, the number of parameters needed by a filter to characterize the exact posterior distribution can be finite. Such cases are rare but are of interest because we can derive algorithms to compute the optimal solution. Optimal algorithms include the Kalman filter for linear-Gaussian cases, grid-based methods when the state-space is discrete and finite, and the Benes and Daum filters for a special class of nonlinear filtering problems [Ristic et al. 2004]. The Kalman filter is described in Section 2.6.1.

In general, however, there is no finite set of parameters to describe the posterior. Approximations are then typically introduced. Suboptimal algorithms include the extended Kalman filter, approximate grid-based methods, Gaussian sums, and particle filters [Ristic et al. 2004]. The extended Kalman filter is described in Section 2.6.2 and the remainder of this work is devoted to particle filters.

2.6.1 Kalman Filter

Suppose the posterior distribution at a given time $k-1$ is Gaussian and therefore, is completely characterized by a mean vector $\mu_{k-1}$ and covariance matrix $C_{k-1}$. By restricting the filtering problem to be linear and Gaussian, then both the prediction distribution $\pi_{k|k-1}$ and posterior distribution $\pi_{k|k}$ of the next time step will be also be Gaussian [Ristic et al. 2004, pg. 7-8]. A linear filtering problem is one in which the functions $f_k$ and $h_k$ of the system equations are linear functions of the state of the signal, while a Gaussian filtering problem is one in which the noise processes $V$ and $W$ in the system equations are Gaussian distributed. We will further assume that the noise processes are zero-mean and independent in time and of each other. For filtering problems that are linear and Gaussian, the system equations can be written as

$$x_k = F_{k-1}x_{k-1} + v_k$$
$$y_k = H_kx_k + w_k,$$

where $v_{k-1} \sim \mathcal{N}(0, \Sigma_{v_{k-1}})$, $w_k \sim \mathcal{N}(0, \Sigma_{w_k})$, and $F_{k-1}$ and $H_k$ are matrices, possibly time-dependent.

The Kalman filter algorithm [Kalman 1960] recursively computes the parameters of the Gaussian prediction and posterior distributions of the signal. It is assumed that we know the matrices $F_k$ and $H_k$ and the noise covariances $\Sigma_{v_k}$ and $\Sigma_{w_k}$, have received the observation $y_k$, and have the previous solution $(\mu_{k-1}, C_{k-1})$. In this case,$$
\pi_{k|k-1} = \mathcal{N}(\mu_{k}^-, C_{k}^-), \quad \pi_{k|k} = \mathcal{N}(\mu_{k}, C_{k}),$$
where [Ristic et al. 2004, pg. 8]

\[
\begin{align*}
\mu_k^- &= F_{k-1} \mu_{k-1} \\
C_k^- &= \Sigma_{v_{k-1}} + F_{k-1} C_{k-1} F_{k-1}^T \\
\mu_k &= \mu_k^- + K_k (y_k - H_k \mu_k^-) \\
C_k &= C_k^- - K_k H_k C_k^- \\
K_k &= C_k^- H_k^T \left( H_k C_k^- H_k^T + \Sigma_w \right)^{-1}.
\end{align*}
\]

(2.11)

\[2.6.2\] Extended Kalman Filter

When the functions \( f_{k-1} \) and \( h_k \) of the system equations are nonlinear functions of the state of the signal, then this is a **nonlinear filtering** problem and the previous algorithm is no longer applicable. In this case, one approach is to approximate the functions by a truncated Taylor expansion evaluated at the current estimate of the signal. This method is known as the extended Kalman filter (EKF) and will be briefly described here.

The nonlinear functions can be approximated as

\[
\begin{align*}
f_{k-1}(x_{k-1}) &\approx f_{k-1}(\mu_{k-1}) + \tilde{F}_{k-1}(x_{k-1} - \mu_{k-1}) \\
h_k(x_k) &\approx h_k(\mu_k^-) + \tilde{H}_k(x_k - \mu_k^-),
\end{align*}
\]

where \( \tilde{F}_{k-1} \) denotes the Jacobian of \( f_{k-1}(x_{k-1}) \) evaluated at \( \mu_{k-1} \) and \( \tilde{H}_k \) denotes the Jacobian of \( h_k(x_k) \) evaluated at \( \mu_k^- \). Higher order terms of the Taylor expansion can be included at additional computational cost. Substituting these approximations of the nonlinear functions into the system equations (2.9) and (2.10), then we have new system equations given by

\[
\begin{align*}
x_k &= \tilde{F}_{k-1} x_{k-1} + v_{k-1} + (f_{k-1}(\mu_{k-1}) - \tilde{F}_{k-1} \mu_{k-1}) \\
y_k &= \tilde{H}_k x_k + w_k + (h_k(\mu_k^-) - \tilde{H}_k \mu_k^-).
\end{align*}
\]

The Kalman filter from Section 2.6.1 can now be applied to this approximate model as it is now linear and still assumed Gaussian. The algorithm equations of (2.11) are now [Ristic et al. 2004, pg. 20]

\[
\begin{align*}
\mu_k^- &= f_{k-1}(\mu_{k-1}) \\
C_k^- &= \Sigma_{v_{k-1}} + \tilde{F}_{k-1} C_{k-1} \tilde{F}_{k-1}^T \\
\mu_k &= \mu_k^- + K_k (y_k - H_k \mu_k^-) \\
C_k &= C_k^- - K_k H_k C_k^- \\
K_k &= C_k^- \tilde{H}_k^T \left( \tilde{H}_k C_k^- \tilde{H}_k^T + \Sigma_w \right)^{-1}.
\end{align*}
\]

(2.12)
There are, in fact, a number of variations of the EKF although we have presented only the most basic algorithm here, see e.g. Daum [2005] for a brief overview. It is important to keep in mind that these EKF methods are approximations and have no optimality properties [Kay 1993, pg. 451].
Chapter 3

Sequential Monte Carlo Approximation of Bayesian Filter

Sequential Monte Carlo (SMC) methods, also known as particle filters (PFs), implement the recursive Bayesian filter with Monte Carlo (MC) simulation. The posterior distribution is approximated by a simulated set of samples with appropriate weights. Rather than solving an approximate problem exactly, as with the EKF, SMC methods solve an exact problem by numerical approximation. This is most attractive in nonlinear and non-Gaussian situations where the integrals of (2.6) and (2.7) are not tractable.

There are different formulations of the SMC methods and in this work, we have taken the approach to present two separate formulations. The first is the probably the most common perspective, which is sequential importance sampling followed by a resampling step, see e.g. Liu and Chen [1998]; Doucet et al. [2000]; Ristic et al. [2004]. The second approach is the mixture formulation of Pitt and Shephard [1999], also known as the auxiliary particle filter. Some authors have described the first formulation from within this mixture framework, see e.g. Carpenter et al. [1999], but we keep them separate here to facilitate comparing this work to the majority of existing literature. We conclude this chapter with a brief discussion of resampling techniques.

In Liu et al. [2001], it is mentioned that the sequential importance sampling method dates back to the 50s with applications in statistical physics. Doucet [1998] notes that there was work during the 1960s and 70s in automatic control that used SMC methods. It is the work of Gordon et al. [1993] that is generally credited for demonstrating the use of resampling in the SMC context, which, in turn, began the recent surge of interest in SMC methods.
3.1 Monte Carlo Background

This section introduces two MC-based methods, classical Monte Carlo and importance sampling. The MC methods we will be concerned with approximate a distribution by a finite collection of realizations of random variables, called samples or particles. The following definition will be needed.

Definition 3.1. The delta-Dirac unit mass located at \( x \) is a function denoted by \( \delta_x \) such that

\[
\delta_x(A) = \begin{cases} 
1 & \text{if } x \in A \\
0 & \text{if } x \notin A,
\end{cases}
\]

where \( A \in \mathcal{B}(\mathbb{R}^n) \) if \( x \in \mathbb{R}^n \) and \( A \in \mathcal{P}(\{x_1, \ldots, x_N\}) \) if \( x \in \{x_1, \ldots, x_N\} \), with \( \mathcal{P} \) denoting the power set.

The process of obtaining realizations of a random variable \( x \) with a given distribution will be called simulating from the distribution or according to the distribution. It has also been described in the literature as generating or drawing samples. In general, it is not always feasible to simulate samples according to a given distribution. See e.g. Devroye [1986] for discussions on random number generation.

3.1.1 Classical Monte Carlo

Let \( \{x^i\}_{i=1}^N \) be a realization of \( \{x^i\}_{i=1}^N \), which is a set of independent and identically distributed (IID) random variables with some distribution \( P \). Then an unweighted discrete approximation of \( P \) is given by

\[
\hat{P} := \frac{1}{N} \sum_{i=1}^N \delta_{x^i}.
\]

Integrals of the form

\[
I = \int f(x) p(x) \, dx,
\]

for some \( P \)-integrable function \( f \), can be approximated by the classical Monte Carlo approximation

\[
I \approx \int f(x) P^N(dx) = \frac{1}{N} \sum_{i=1}^N f(x^i).
\]

The justification for the approximation in (3.2) relies on the strong law of large numbers, see e.g. Shiryayev [1984, pg. 366]. This then implies that the empirical distribution \( P^N \) approximates the true distribution \( P \). Note that the use of the Lebesgue integral in (3.2) is important for our formulation of the approximations.
3.1.2 Importance Sampling

An alternative method of approximating $P$ relies on the use of importance sampling (IS). The idea relies on writing (3.1) as

$$\int f(x)p(x)\,dx = \int f(x)\frac{p(x)}{q(x)}q(x)\,dx,$$

where $q$ is a density of probability distribution $Q$. Notice that the left-hand side is an expectation with respect to the probability measure $P$, i.e. $E_p[f(x)]$, and the right-hand side is an expectation with respect to the probability measure $Q$, i.e. $E_q[f(x)p(x)/q(x)]$. If $\{\tilde{x}^i\}_{i=1}^N$ is a realization of a set of IID random variables with distribution $Q$, then provided that the support of $q$ includes the support of $p$, approximations of integrals of the form in (3.1) can be given by the importance sampling approximation

$$I \approx \int f(x)p(x)Q^N(dx) = \frac{1}{N} \sum_{i=1}^N f(\tilde{x}^i)\frac{p(\tilde{x}^i)}{q(\tilde{x}^i)}.$$ (3.3)

The justification for this approximation is for the same reasons as the classical Monte Carlo approximation. Importance sampling is often introduced as a solution to the scenario when one cannot sample directly from $P$ but can sample from a “close” distribution $Q$, known as the importance or proposal distribution, and evaluate the ratio $p(x)/q(x)$.

If the densities $p$ and $q$ are only known up to proportionality, then we can use a similar, but different approximation of (3.1). Assume that we can evaluate $w(x) \propto p(x)/q(x)$, called the unnormalized importance weight. If $\{\tilde{x}^i\}_{i=1}^N$ is a realization of a set of IID random variables with distribution $Q$, then

$$I = \frac{\int f(x)w(x)q(x)\,dx}{\int w(x)q(x)\,dx} \approx \frac{1}{N} \sum_{i=1}^N f(\tilde{x}^i)\frac{w(\tilde{x}^i)}{\sum_{j=1}^N w(\tilde{x}^j)},$$ (3.4)

again, provided that the support of $q$ includes the support of $p$. Note that the same random samples are used in the numerator and denominator for the approximation in (3.4).

For our purposes, it will be convenient to formulate the approximation of (3.4) in terms of a probability measure that approximates $P$. A weighted discrete approximation of $P$ is given by

$$P \approx \tilde{P}^N := \sum_{i=1}^N \tilde{w}^i \delta_{\tilde{x}^i},$$ (3.5)

where

$$w(\tilde{x}^i) \propto p(\tilde{x}^i)/q(\tilde{x}^i), \quad \tilde{w}^i := \frac{w(\tilde{x}^i)}{\sum_{j=1}^N w(\tilde{x}^j)}.$$ (3.6)
The approximation of (3.4) can be equivalently written as

$$I \approx \int f(x) \tilde{P}^N(dx) = \sum_{i=1}^{N} \tilde{w}^i f(\tilde{x}^i).$$

(3.7)

The $\tilde{w}^i$ terms are the normalized importance weights, and the division by the sum of weights in (3.6) will be referred to as normalization. Note that we may omit the “normalized” or “unnormalized” term when discussing the importance weights if it is clear from notation which we are referring to.

In this work, we will only consider importance sampling for scenarios where $p$ and $q$ are known up to proportionality. The requirement mentioned concerning the support of the importance density $q$ including the support of the true density $p$ will be assumed and not explicitly repeated for the rest of this work. Alternative methods for approximating a distribution that cannot be sampled from directly include rejection sampling and Markov Chain Monte Carlo (MCMC), see e.g. Robert and Casella [1999]. These methods will not be discussed in this work.

**Sampling Importance Resampling (SIR)**

An algorithm for generating equally-weighted samples approximately distributed according to a distribution $P$ using an importance distribution $Q$ is known as sampling importance resampling (SIR) [Rubin 1987]. It consists of first drawing $M$ proposal points from the importance distribution and calculating the importance weights. The points and weights define a weighted discrete approximation, see (3.5), from which we can sample $N$ points from. The effectiveness of the SIR algorithm depends on $M$ being sufficiently large which, in turn, depends on how close $P$ and $Q$ are. As $M/N \to \infty$, the $N$ samples will be IID from $P$ [Rubin 1987].

Sampling from a discrete distribution will be referred to as resampling. There are various methods for resampling and we have devoted Section 3.4 to describing these techniques. For now, we will refer to this resampling step in our algorithms with a generic function “RESAMPLE”, which takes a set of samples and weights as input, i.e. a weighted discrete distribution, and returns the resampled points and, optionally, the new weights of the resampled points. The SIR algorithm is given in Algorithm 1.
Algorithm 1 Sampling Importance Resampling

\begin{verbatim}
% Importance Sampling
For i = 1 : M, sample \( \tilde{x}^i \sim Q \\
For i = 1 : M, evaluate unnormalized importance weights:
\[ w(\tilde{x}^i) \propto p(\tilde{x}^i)/q(\tilde{x}^i) \]
For i = 1 : M, normalize importance weights:
\[ \tilde{w}^i = w(\tilde{x}^i) / \sum_{j=1}^{N} w(\tilde{x}^j) \]

% Resampling
\{x^i\}_{i=1}^{N} = RESAMPLE( \{\tilde{x}^i, \tilde{w}^i\}_{i=1}^{M} )
\end{verbatim}
3.2 Particle Filtering

Returning now to the filtering scenario, we aim to construct an approximation to the joint posterior distribution \( \pi_{0:k|k} \) and the marginal distribution \( \pi_{k|k} \). We now apply IS to Bayesian filtering. The general idea is still to sample from some importance distribution and compute importance weights, which forms a weighted discrete approximation of a probability distribution. The difference here is that we are now sampling realizations from a stochastic process and the probability distribution that we are interested in approximating is the Bayesian posterior distribution.

3.2.1 Sequential Importance Sampling

In this section, we describe the application of IS to the Bayesian filtering context. We will also be interested in formulating this as a recursive algorithm. The recursive formulation is important for online applications, where storage capacity may be limited and fast estimates are required. However, Daum [2005] has suggested that with the cheap, fast memory and powerful processors available today, these reasons for preferring a recursive algorithm are often not an issue for many applications.

Bayesian Filtering via IS

Recall the shorthand notation \( \pi_{0:k|k} \) for the joint posterior distribution that was defined as

\[
\pi_{a:b|c} := P_{x_{a:b}|y_{1:c}}.
\]

Since, in general, it is not feasible to sample from \( \pi_{0:k|k} \), we adopt an IS approach. Let \( Q_{x_{0:k}|y_{1:k}} \) be a different distribution that is “close” to \( \pi_{0:k|k} \) that we can sample from, i.e. an importance distribution, and let \( \{\tilde{x}^i_{0:k}\}_{i=1}^N \) be a realization of a set of IID random trajectories with distribution \( Q_{x_{0:k}|y_{1:k}} \). If we can evaluate the densities \( p(x_{0:k}|y_{1:k}) \) and \( q(x_{0:k}|y_{1:k}) \) up to proportionality, then we have a weighted discrete approximation to the joint posterior similar to (3.5) given by

\[
\pi_{0:k|k} \approx \tilde{\pi}^N_{0:k|k} := \sum_{i=1}^N \tilde{w}^i_k \delta_{\tilde{x}^i_{0:k}},
\]

where the importance weights are computed similar to (3.6) as

\[
w(\tilde{x}^i_{0:k}) \propto \frac{p(\tilde{x}^i_{0:k}|y_{1:k})}{q(\tilde{x}^i_{0:k}|y_{1:k})}, \quad \tilde{w}^i = \frac{w(\tilde{x}^i_{0:k})}{\sum_{j=1}^N w(\tilde{x}^j_{0:k})}.
\]
This method, however, is not recursive. If we computed an IS approximation for the distribution at time \( k - 1 \), then there is no mechanism to use this solution to compute an IS approximation for the distribution at time \( k \).

**Bayesian Filtering via SIS**

We aim to be able to compute an approximation of \( \pi_{0:k|k} \) recursively, i.e. from our approximation of \( \pi_{0:k-1|k-1} \). To do this, we must derive a recursive importance weight update.

We write the density of the importance distribution as

\[
q(x_{0:k}|y_{1:k}) = q(x_k|x_{0:k-1}, y_{1:k}) q(x_{0:k-1}|y_{1:k-1}). \tag{3.10}
\]

The idea is that, having samples that are realizations from \( Q_{x_{0:k-1}|y_{1:k-1}} \), we can obtain samples from \( Q_{x_{0:k}|y_{1:k}} \) just by augmenting each existing \( \tilde{x}_{0:k-1}^i \) with a sample \( \tilde{x}_k^i \sim Q_{x_k|\tilde{x}_{0:k-1}^i, y_{1:k}} \), i.e. \( \tilde{x}_{0:k}^i = (\tilde{x}_{0:k-1}^i, \tilde{x}_k^i) \).

The importance weight update can be computed recursively as follows. From the factorization of the posterior density given by

\[
p(x_{0:k}|y_{1:k}) = \frac{p(y_k, x_{0:k}|y_{1:k-1})}{p(y_k|y_{1:k-1})} \tag{3.11}
\]

\[
\propto p(y_k|x_{0:k}, y_{1:k-1}) p(x_{0:k}|y_{1:k-1}) = p(y_k|x_{0:k}, y_{1:k-1}) p(x_k|x_{0:k-1}, y_{1:k-1}) p(x_{0:k-1}|y_{1:k-1})
\]

and the factorization of the importance density in (3.10), the normalized importance weights satisfy the following

\[
\tilde{w}_k^i \propto \frac{p(\tilde{x}_{0:k}^i|y_{1:k})}{q(\tilde{x}_{0:k}^i|y_{1:k})} \propto \frac{p(y_k|\tilde{x}_{0:k}^i, y_{1:k-1}) p(\tilde{x}_k^i|\tilde{x}_{0:k-1}^i, y_{1:k-1})}{q(\tilde{x}_k^i|\tilde{x}_{0:k-1}^i, y_{1:k})} \frac{p(\tilde{x}_{0:k}^i|y_{1:k-1})}{q(\tilde{x}_{0:k}^i|y_{1:k-1})} \tilde{w}_{k-1}^i. \tag{3.12}
\]

The \( p(y_k|y_{1:k-1}) \) term in (3.11) is a constant that often cannot be expressed in closed-form. However, it is independent of \( i \) and can be left out of the weight update since we need to evaluate the weights only up to proportionality.

The form of (3.12) is very general. We will restrict our attention to importance distributions of the form

\[q(x_k|x_{0:k-1}, y_{1:k}) = q(x_k|x_{k-1}, y_k)\]
and models of the form in Section 2.4. Imposing these restrictions, the weight update simplifies to

$$
\tilde{w}_k^i \propto \frac{p(y_k | \tilde{x}_k^i) p(\tilde{x}_k^i | \tilde{x}_{k-1}^i)}{q(\tilde{x}_k^i | \tilde{x}_{k-1}^i, y_k)} \tilde{w}_{k-1}^i.
$$  \hspace{1cm} (3.13)

Having sampled from an importance distribution and computed the importance weights, the weighted discrete approximation to the joint posterior can be formed exactly as in (3.8). The difference here is that the sampling and weight update are done recursively over time, hence the name sequential importance sampling (SIS).

Notice that we can work with the marginal distribution \( \pi_{k|k} \) rather than the joint posterior \( \pi_{0:k|k} \). In practice, this means that instead of storing the full trajectory of each particle, only the particles from the previous time step need to be stored due to our Markov assumption of the signal. We should keep in mind that when drawing samples of \( x_k \), we are, in fact, drawing samples of \( x_{0:k} \) and just discarding \( x_{0:k-1} \).

The SIS algorithm is given in Algorithm 2. Note that we have simplified the initialization for the algorithm and assumed that we can sample from \( P_{x_0} \).
Algorithm 2 Sequential Importance Sampling

\[
\begin{align*}
\text{\% Initialize, } k = 0 \\
\text{For } i = 1 : N, \text{ sample } \tilde{x}_0^i \sim P_{\pi_0} \\
\text{For } i = 1 : N, \text{ set } \tilde{w}_0^i = 1/N \\
\text{For } k = 1, 2, \ldots \\
\hspace{1em} \text{For } i = 1 : N, \text{ sample } \tilde{x}_k^i \sim Q_{\pi_k|\tilde{x}_{k-1}^i}, y_k \\
\hspace{1em} \text{For } i = 1 : N, \text{ evaluate unnormalized importance weights:} \\
\hspace{2em} w(\tilde{x}_0^i) \propto \frac{p(y_k|\tilde{x}_k^i) p(\tilde{x}_k^i|\tilde{x}_{k-1}^i)}{q(\tilde{x}_k^i|\tilde{x}_{k-1}^i), y_k} \tilde{w}_{k-1}^i \\
\hspace{1em} \text{For } i = 1 : N, \text{ normalize importance weights:} \\
\hspace{2em} \tilde{w}_k^i = w(\tilde{x}_0^i) / \sum_{j=1}^N w(\tilde{x}_0^j)
\end{align*}
\]
3.2.2 Sequential Importance Sampling/Resampling

Unfortunately, the SIS algorithm presented so far often performs quite poorly in practice. It can happen that after only a few iterations of the algorithm, all but one of the normalized importance weights are close to zero. This results in a large computation effort being devoted to updating trajectories with little contribution to the final estimate. Consequently, the algorithm will fail to adequately represent the posterior distribution. Such a scenario has been described as a degeneracy of the algorithm. [Doucet et al. 2000, pg. 199].

In Gordon et al. [1993], a resampling step is introduced so that at every time step, $N$ samples are drawn from the distribution of $N$ weighted samples. This effectively duplicates trajectories with large weight and removes particles with negligible weight. This approach is generally considered to be what established SMC as a practical filtering method. Roughly speaking, the resampling step “rejuvenates” the sampler in the hope that it will improve results for future states, although it does not improve the results for the current state [Liu and Chen 1998, pg. 9]. This implies that any estimates formed from the sample set should be computed before resampling.

We now give a general algorithm for the SIS/Resampling (SISR) filter in Algorithm 3. Note that when resampling, this algorithm is just the SIR algorithm with $M = N$ to fit the filtering context.

Having a recursive algorithm that approximates the posterior distribution, we can turn our attention to the estimates. Since expectations with respect to a weighted discrete probability measure are conveniently evaluated using summations, see (3.7), approximations of the conditional mean (2.7) and conditional covariance (2.8) are given respectively by

$$\mu_k \approx \hat{\mu}_k := \sum_{i=1}^{N} \tilde{w}_k^i \tilde{x}_k^i,$$

$$C_k \approx \hat{C}_k := \sum_{i=1}^{N} \tilde{w}_k^i (\tilde{x}_k^i - \hat{\mu}_k)(\tilde{x}_k^i - \hat{\mu}_k)^T.$$
Algorithm 3 Sequential Importance Sampling/Resampling

% Initialize, $k = 0$
For $i = 1 : N$, sample $\tilde{x}_0^i \sim P_{x_0}$
For $i = 1 : N$, set $\tilde{w}_0^i = 1/N$

For $k = 1, 2, \ldots$

% Importance Sampling
For $i = 1 : N$, sample $\tilde{x}_k^i \sim Q_{x_k|x_{k-1}, y_k}$

For $i = 1 : N$, evaluate unnormalized importance weights:

$$w(\tilde{x}_{0,k}^i) \propto \frac{p(y_k|x_k^i) p(x_k^i|x_{k-1}^i)}{q(x_k^i|x_{k-1}^i, y_k)} \tilde{w}_{k-1}^i$$

For $i = 1 : N$, normalize importance weights:

$$\tilde{w}_k^i = w(\tilde{x}_{0,k}^i) / \sum_{j=1}^N w(\tilde{x}_{0,k}^j)$$

% Resampling
$[\{\tilde{x}_k^i, \tilde{w}_k^i\}_{i=1}^N] = \text{RESAMPLE}(\{\tilde{x}_k^i, \tilde{w}_k^i\}_{i=1}^N)$
### 3.2.3 Examples of Importance Distributions

When approximating some probability distribution $P$ using importance sampling, the choice of importance distribution $Q$ is arbitrary provided that the support of $q$ includes the support of $p$. However, some choices will perform better than others and some choices will require more computational complexity than others. In this section, we review a few well-known choices as examples.

#### Example 1: Prior Importance Distribution

Probably the most common choice of importance distribution is to use the prior, i.e.

$$Q_{x_k|x_{k-1}} = P_{x_k|x_{k-1}}.$$  

The normalized importance weights are given from (3.13) as

$$\tilde{w}_k^i \propto \frac{p(y_k|x_k^i)}{q(x_k^i|x_{k-1}^i, y_k)} \tilde{w}_{k-1}^i$$  

$$= \frac{p(y_k|x_k^i)}{p(x_k^i|x_{k-1}^i)} \tilde{w}_{k-1}^i$$  

$$= p(y_k|x_k^i) \tilde{w}_{k-1}^i.$$  

This choice of importance distribution simplifies the algorithm: all that is needed is to be able to simulate samples from $P_{x_k|x_{k-1}}$ and to evaluate the likelihood $p(y_k|x_k^i)$. However, exploring the state space without using the present measurement $y_k$, i.e. having the importance distribution independent of $y_k$, can be inefficient if the prior is not “close” to the posterior. It should be noted that this was the choice of importance distribution in the seminal paper by Gordon et al. [1993].

#### Example 2: Extended Kalman Importance Distribution

Another intuitive choice for the importance distribution results from linearizing the model locally as with the EKF, see e.g. van der Merwe et al. [2000] and Ristic et al. [2004]. In this case, each particle has an importance distribution that is a Gaussian distribution with mean and covariance given by the EKF posterior. We can then write the $i^{th}$ particle’s importance distribution as

$$Q_{x_k|x_{k-1}} = \mathcal{N}(\mu_k^i, C_k^i).$$
CHAPTER 3. SMC APPROXIMATION OF BAYESIAN FILTER

where the EKF update is given from (2.12) as

\[ \begin{align*}
\mu_k^{(i)} &= f_k^{-1}(\tilde{x}_k^{i-1}) \\
C_k^{(i)} &= \Sigma_{w_{k-1}} + \tilde{F}_k^{i}C_{k-1}^{i}\tilde{F}_k^{iT} \\
\mu_k^i &= \mu_k^{(i)} + K_k^i(y_k - h_k(\mu_k^{(i)})) \\
C_k^i &= C_k^{(i)} - K_k^i\tilde{H}_k^iC_k^{(i)} \\
K_k^i &= C_k^{(i)}(\tilde{H}_k^i)^T[\tilde{H}_k^iC_k^{(i)}(\tilde{H}_k^i)^T + \Sigma_{w_k}]^{-1}.
\end{align*} \]

This means that if there are \( N \) particles, then the algorithm involves \( N \) extended Kalman filter updates. Notice that this requires that the error covariance matrix for each particle, i.e. \( C_i^k \), be saved and propagated at every time step, as well as resampled when resampling is performed.

The normalized importance weights are given from (3.13) as

\[ \tilde{w}_k^i \propto \frac{p(y_k|\tilde{x}_k^i)p(\tilde{x}_k^i|\tilde{x}_{k-1}^i)}{q(\tilde{x}_k^i|\tilde{x}_{k-1}^i,y_k)} \tilde{w}_{k-1}^i = \frac{p(y_k|\tilde{x}_k^i)p(\tilde{x}_k^i|\tilde{x}_{k-1}^i)}{\nu(\tilde{x}_k^i;\mu_k^i,C_k^i)} \tilde{w}_{k-1}^i, \]

where \( \nu(\cdot;\mu,C) \) is the density of a Gaussian distribution with mean \( \mu \) and covariance \( C \).

The improvement from using this choice of importance distribution will depend on whether the extended Kalman posterior is significantly different from the prior and whether the inaccuracies introduced by the linearization are significant. The computational aspect of performing a large number of EKF updates can also be an issue. In fact, other Kalman related filters could be used in place of the basic EKF that we have described. For example, the unscented Kalman filter was used in the proposed particle filter of van der Merwe et al. [2000].

3.3 Auxiliary Particle Filters

Pitt and Shephard [1999] present a related filter which relies on a slightly different formulation of particle filtering. A mixture distribution, i.e. a distribution that is a finite weighted sum of distributions, is introduced as the target distribution along with a trick to keep the computation manageable. We have included this formulation because there is more flexibility in choosing importance distributions. It is various auxiliary particle filters that will be tested in the simulations.
3.3.1 Empirical Mixtures and Auxiliary Variables

In the SISR framework previously presented, at each time step, we have \( N \) different prediction and posterior distributions—one for each particle. In this alternative framework, at each time step, we approximate the true prediction and posterior each by a single \( N \) component mixture. The empirical prediction distribution and density are given respectively by

\[
\hat{P}_{x_k|y_{1:k-1}} = \sum_{j=1}^{N} \tilde{w}_k^j P_{x_k|x_k}^{j}
\]

and the empirical filtering distribution and density are given respectively by

\[
\hat{p}(x_k|y_{1:k-1}) = \sum_{j=1}^{N} \tilde{w}_k^j p(x_k|x_k) \]

and

\[
\hat{p}(x_k|y_{1:k-1}) \propto p(y_k|x_k) \hat{p}(x_k|y_{1:k-1}).
\]

where \( A \in \mathcal{B}(\mathbb{R}^{n_x}) \).

Consider what happens when attempting importance sampling using these mixture approximations. In the general case, evaluating the importance weight for each sample will require evaluating \( N \) densities because the prior is a mixture with \( N \) components, i.e.

\[
w(x_k) \propto \frac{\hat{p}(x_k|y_{1:k})}{q(x_k|y_{1:k})} \propto p(y_k|x_k) \sum_{j=1}^{N} \tilde{w}_k^j p(x_k|x_k^{j}).
\]

As an example, drawing \( M \) samples would require \( M \times N \) evaluations. This is considered to be too computationally expensive to be a feasible algorithm since \( M \) and \( N \) are typically large. Note that this problem does not arise when sampling from the empirical prediction distribution, i.e. \( q(x_k|y_{1:k}) = \sum_{j=1}^{N} \tilde{w}_k^j p(x_k|x_k^{j}) \), as the evaluations of the empirical prior density will cancel out of (3.19).

Pitt and Shephard [1999] propose to perform the filtering in a higher dimension. By introducing a discrete auxiliary variable \( j \), which refers to the index of a mixture component, the mixture is essentially broken up over this new dimension, see Figure 3.1. The joint density, i.e. the density of the joint distribution of state and auxiliary variable, is defined as [Pitt and Shephard 1999, pg. 592]

\[
\hat{p}(x_k, j|y_{1:k}) \propto \tilde{w}_k^j p(y_k|x_k) p(x_k|x_k^{j}) \quad \text{for } j = 1, \ldots, N.
\]
The unnormalized importance weight can then be conveniently evaluated in this joint space as

\[ w(x_k, j) \propto \frac{\hat{p}(x_k, j|y_1:k)}{q(x_k, j|y_1:k)} \propto \frac{\tilde{w}_k^j p(y_k|x_k) p(x_k|x_{k-1}^j)}{q(j|y_1:k) q(x_k|j, y_1:k)}, \quad (3.20) \]

where the importance distribution is also a mixture distribution with \( N \) components.

For notational convenience, we will write the \( j \)th mixture weight of the importance distribution as \( \beta_k^j \propto q(j|y_1:k) \), and the \( j \)th mixture component of the importance distribution as \( Q_k^j := Q_{x_k|j, y_1:k} \) with density \( q(x_k) := q(x_k|j, y_1:k) \), where the conditioning on the realized observation process is understood. The \( \beta_k^j \) terms may be referred to as the (unnormalized) **first-stage weights** and the \( w_k^j := w(x_k^j, j) \) terms, which are the unnormalized importance weights of the \( i \)th importance sample \((\tilde{x}_k^i, j^i)\), may be referred to as the (unnormalized) **second-stage weights** [Pitt and Shephard 1999].

In practice, we can sample from the joint space by first, simulating an index of the mixture with probability proportional to \( \beta_k^j \) and then simulating a sample from the distribution given that index, i.e. \( \tilde{x}_k \sim Q_k^j \). The simulated indices are discarded afterwards as they are only to assist the simulation. The choice of first-stage weights \( \beta_k^j \) is arbitrary as long as the normalized first-stage weights sum to one and the conditioning is on the measurements received up to time \( k \), i.e. the choice of \( \beta_k^j \) cannot depend on a measurement received after time \( k \). Different choices of importance distribution mixture weights and components will be given in Section 3.3.2.
We now present an algorithm for the **auxiliary particle filter (APF)** in Algorithm 4. Notice that the indices are simulated from a weighted discrete distribution, and so we have used our generic function “RESAMPLE” in the algorithm to emphasize this. It has been suggested to sample $M = N$ from the importance distribution and skip the second “extra” resampling step altogether, see e.g. Carpenter et al. [1999] and Ristic et al. [2004, pg. 50-51]. We will use this special case frequently in the simulations.

It should be pointed out that even in the situation where we can sample from the empirical filtering distribution (3.17), this will not produce IID samples from the true posterior due to the finite mixture approximation [Pitt and Shephard 1999, pg. 593]. We will refer to this point later.
Algorithm 4 Auxiliary Particle Filter

% Initialize, \( k = 0 \)
\For {i = 1 : N}, sample \( \tilde{x}_0^i \sim P_{x_0} \)
\For {i = 1 : N}, set \( \tilde{w}_0^i = 1/N \)

\For {k = 1, 2, \ldots}

% First-stage weights
\For {i = 1 : N}, assign unnormalized 1st stage weights \( \beta_k^i \)
\For {i = 1 : N}, normalize 1st stage weights:
\[
\tilde{\beta}_k^i = \frac{\beta_k^i}{\sum_{i=1}^N \beta_k^i}
\]

% Importance Sampling
Sample indices \( \{ j^i \}_{i=1}^M = \text{RESAMPLE}(\{i, \tilde{\beta}_k^i\}_{i=1}^N) \)
\For {i = 1 : M}, sample \( \tilde{x}_k^i \sim Q_k^i \)
\For {i = 1 : M}, evaluate unnormalized 2nd stage weights:
\[
\tilde{w}_k^i \propto \frac{\tilde{p}(\tilde{x}_k^i, j^i | y_{1:k})}{q(\tilde{x}_k^i, j^i | y_{1:k})}
\]
\For {i = 1 : M}, normalize 2nd stage weights:
\[
\tilde{w}_k^i = \frac{\tilde{w}_k^i}{\sum_{i=1}^N \tilde{w}_k^i}
\]

% Resampling
\[
[\{\tilde{x}_k^i, \tilde{w}_k^i\}_{i=1}^N] = \text{RESAMPLE}(\{\tilde{x}_k^i, \tilde{w}_k^i\}_{i=1}^M)
\]
3.3.2 Examples of Importance Distributions for APF

We will now present four examples that illustrate the choice of mixture weights $\beta_k^j$ and components $Q_k^j$ for the importance distribution. We then derive the form for the importance weights $w(x_k, j)$ by simply substituting our choice of $\beta_k^j$ and $q^j(x_k)$ into (3.20). The second-stage weights $w_k^i$ can be evaluated by substituting the importance sample $(x_k^i, j^i)$ into the function $w(x_k, j)$.

Example 1 is the basic example that samples from the empirical prior. Example 2 shows a different choice of weights, while Example 3 shows a different choice of components. Example 4 illustrates an example using different weights and components. These four examples will be used in the simulations and we have used the naming convention of SIR1, SIR2, SIR3, and SIR4 so that we can easily refer to them.

Example 1: Prior Components with Previous Weights (SIR1)

The common first example is to sample from the empirical prediction distribution, i.e. $Q_{x_k|y_{1:k}} = \sum_{j=1}^N \tilde{w}_{k-1}^j P_{x_k|x_{k-1}^j}$. From this form of the importance distribution, we see that the first-stage weights are proportional to the second-stage weights of the last time step, i.e. $\beta_k^j \propto \tilde{w}_{k-1}^j$, and the mixture components are equal to the conditional prediction distributions, i.e. $Q_k^j = P_{x_k|x_{k-1}^j}$. Notice that neither the weights or the components in this choice of importance distribution use the latest measurement.

We can derive the importance weights as

$$ w(x_k, j) \propto \frac{\tilde{w}_{k-1}^j p(y_k|x_k) p(x_k|x_{k-1}^j)}{\beta_k^j q^j(x_k)} \propto \frac{\tilde{w}_{k-1}^j p(y_k|x_k) p(x_k|x_{k-1}^j)}{\tilde{w}_{k-1}^j p(x_k|x_{k-1}^j)} = p(y_k|x_k). $$

This means that the $i^{th}$ importance sample $(\tilde{x}_k^i, j^i)$ will be assigned the second-stage weight $w_k^i \propto p(y_k|\tilde{x}_k^i)$. For the remaining examples, we only give the form of the importance weight function $w(x_k, j)$ and do not explicitly write the second-stage weight $w_k^i$ to keep notation simple.
Example 2: Prior Components with Alternative Weights (SIR2)

Consider the following example to illustrate a convenient choice of first-stage weights. Pitt and Shephard [1999] propose using an importance distribution that treats the likelihood \( p(y_k | x_k) \) of the \( j \)th mixture component as the constant \( p(y_k | \xi_k^j) \), where \( \xi_k^j \) is the mean, the mode, a draw, or some other likely value associated with the density of \( x_k | \tilde{x}_{k-1}^j \). The empirical filtering distribution (3.17) is approximated using the importance distribution

\[
Q_{x_k | y_{1:k}}(A) \propto \sum_{j=1}^{N} \tilde{w}_{k-1}^j p(y_k | \xi_k^j) \int_A p(x_k | \tilde{x}_{k-1}^j) \, dx_k
\]

\[
= \sum_{j=1}^{N} \tilde{w}_{k-1}^j p(y_k | \xi_k^j) P_{x_k | \tilde{x}_{k-1}^j}(A) \quad A \in \mathcal{B}(\mathbb{R}^n).
\]

This means that our importance mixture consists of the mixture components given by the transition distributions \( Q_k^j = P_{x_k | \tilde{x}_{k-1}^j} \) again, but now with the first-stage weights \( \beta_k^j \propto \tilde{w}_{k-1}^j p(y_k | \xi_k^j) \). The intention is that we will simulate more samples from those transition distributions that are associated with large likelihood and hopefully, this will lead to a more accurate particle representation of the target distribution.

The importance weights can be derived as

\[
w(x_k, j) \propto \frac{\tilde{w}_{k-1}^j p(y_k | x_k) p(x_k | \tilde{x}_{k-1}^j)}{\beta_k^j q^j(x_k)}
\]

\[
\propto \frac{\tilde{w}_{k-1}^j p(y_k | x_k) p(x_k | \tilde{x}_{k-1}^j)}{\tilde{w}_{k-1}^j p(y_k | \xi_k^j) p(x_k | \tilde{x}_{k-1}^j)}
\]

\[
= \frac{p(y_k | x_k)}{p(y_k | \xi_k^j)}.
\]

In our simulations, SIR2 will always use the mean of the density of \( x_k | \tilde{x}_{k-1}^j \) for \( \xi_k^j \).

Example 3: Kalman Components with Previous Weights (SIR3)

Using the prior distributions for the mixture components of the importance distribution, i.e. \( Q_k^j = P_{x_k | \tilde{x}_{k-1}^j} \), is often a convenient choice because it can be computationally cheap to simulate samples from them and the second-stage weights

\[1\]This example has been described in the literature as the auxiliary particle filter, see e.g. Ristic et al. [2004, pg. 49-52], whereas we present it as an example of an auxiliary particle filter, see Pitt and Shephard [1999].
that follow are relatively easy to compute. However, other mixture components can be used to design better approximations of the empirical filtering distribution.

The posterior distributions of the Kalman filter can also be used as mixture components. For nonlinear filtering scenarios, a local linearization of the models, as in the EKF, can be used. This means that $Q^j_k = \mathcal{N}(\mu^j_k, C^j_k)$, where

\begin{align}
\mu^j_k &= f_{k-1} (\tilde{x}^j_{k-1}) \\
C^j_k &= \Sigma_{\nu_k-1} \\
\mu^j_k &= \mu^j_k + K^j_k \left( y_k - h_k(\mu^j_k) \right) \\
C^j_k &= C^j_k - K^j_k \hat{H}^j_k C^j_k \\
K^j_k &= C^j_k \hat{H}^j_k^T \left[ \hat{H}^j_k C^j_k \hat{H}^j_k^T + \Sigma_{\nu_k} \right]^{-1}
\end{align}

and $\hat{H}^j_k$ is the Jacobian matrix of $h_k$ evaluated at $f_{k-1}(\tilde{x}^j_{k-1})$. Comparing this with the extended Kalman importance distribution in the SISR framework, note that here there is no error covariance propagated from the previous time step, i.e., compare (3.21b) with (3.14b).

If we choose $Q_{\pi_k|y_{1:t}} = \sum_{j=1}^{N} \tilde{w}^j_{k-1} \mathcal{N}(\mu^j_k, C^j_k)$, we have an importance distribution defined by the mixture weights $\beta^j_k \propto \tilde{w}^j_{k-1}$ and the mixture components $Q^j_k = \mathcal{N}(\mu^j_k, C^j_k)$. The importance weights are then given by

\begin{align}
w(x_k, j) &\propto \frac{\tilde{w}^j_{k-1} p(y_k|x_k) p(x_k|\tilde{x}^j_{k-1})}{\beta^j_k q^j(x_k)} \\
&\propto \frac{\tilde{w}^j_{k-1} p(y_k|x_k) p(x_k|\tilde{x}^j_{k-1})}{\tilde{w}^j_{k-1} \nu(x_k; \mu^j_k, C^j_k)} \\
&= \frac{p(y_k|x_k) p(x_k|\tilde{x}^j_{k-1})}{\nu(x_k; \mu^j_k, C^j_k)}.
\end{align}

**Example 4: Kalman Components with Alternative Weights (SIR4)**

Note that while Example 2 introduced a convenient choice for alternative mixture weights, the choice of mixture components did not use the latest measurement. Example 3, on the other hand, introduced a convenient choice for the mixture components, while the mixture weights did not use the latest measurement. In Heine [2005], an example is given that uses both these design parameters. The Kalman posteriors are again used for the mixture components of the importance distribution, but now with alternative weights.

The EKF uses approximations of the true models so that the approximate models are linear and Gaussian. We write the $j^{th}$ component’s linearized model as $\tilde{p}^j$ and
substitute these approximations into the empirical filtering distribution (3.17) to derive an importance distribution. Define the constants $c^j_k$ as

$$c^j_k := \int_{\mathbb{R}^n} \bar{p}^j(y_k|x_k) \bar{p}^j(x_k|\tilde{x}^j_{k-1}) \, dx_k.$$ 

Then we have an importance distribution as

$$Q_{x_k|y_{1:k}}(A) \propto \sum_{j=1}^{N} \bar{w}^j_{k-1} \int_A \bar{p}^j(y_k|x_k) \bar{p}^j(x_k|\tilde{x}^j_{k-1}) \, dx_k$$

$$= \sum_{j=1}^{N} \bar{w}^j_{k-1} c^j_k \frac{\int_A \bar{p}^j(y_k|x_k) \bar{p}^j(x_k|\tilde{x}^j_{k-1}) \, dx_k}{\int_{\mathbb{R}^n} \bar{p}^j(y_k|x_k) \bar{p}^j(x_k|\tilde{x}^j_{k-1}) \, dx_k}$$

$$= \sum_{j=1}^{N} \bar{w}^j_{k-1} c^j_k \mathcal{N}(A; \mu^j_k, C^j_k), \quad A \in \mathcal{B}(\mathbb{R}^n),$$

where $\mu^j_k$ and $C^j_k$ again are the $j$th extended Kalman posterior mean and covariance given by (3.21). In this case, the constants $c^j_k$ have a closed-form solution given by [Heine 2005]

$$c^j_k = \nu \left( y_k, h_k (f_{k-1}(\tilde{x}^j_{k-1})), \hat{H}^j_k \Sigma_{w_{k-1}} (\hat{H}^j_k)^T + \Sigma_{w_k} \right).$$

The importance distribution is defined by the first-stage weights, which are $\beta^j_k \propto \bar{w}^j_{k-1} c^j_k$, and the distribution components, which are the extended Kalman posteriors $Q^j_k = \mathcal{N}(\mu^j_k, C^j_k)$. The importance weights are then given by

$$w(x_k, j) \propto \frac{\bar{w}^j_{k-1} p(y_k|x_k) p(x_k|\tilde{x}^j_{k-1})}{\beta^j_k q^j(x_k)}$$

$$\propto \frac{\bar{w}^j_{k-1} p(y_k|x_k) p(x_k|\tilde{x}^j_{k-1})}{\bar{w}^j_{k-1} c^j_k \nu(x_k; \mu^j_k, C^j_k)}$$

$$= \frac{p(y_k|x_k) p(x_k|\tilde{x}^j_{k-1})}{c^j_k \nu(x_k; \mu^j_k, C^j_k)}.$$

### 3.4 Resampling

The two main design choices in SMC are the choice of the importance distribution and the type of resampling. Various importance distributions were given in the examples of Section 3.2.3 and Section 3.3.2, so we now turn our attention to the resampling step.

One way of thinking about the resampling step is that the objective is to approximate the weighted discrete approximation $\tilde{P}^M$ with an equally-weighted discrete...
approximation $P^N$. Often, $M$ and $N$ are chosen to be equal, but we keep the notation separate. The resampling algorithms we will focus on will generate a set of equally-weighted samples $\{x^i\}_{i=1}^N$ from the weighted discrete distribution defined by $\{\tilde{x}^i, \tilde{w}^i\}_{i=1}^M$. Various methods for resampling will be briefly reviewed in Section 3.4.1.

Although referred to in Section 3.2.2 as the key to successfully applying MC methods to the sequential context, resampling does introduce other problems. The resampling step introduces repeated points into the sample set, which can lead to a lack of particle “diversity”. This has been referred to as sample impoverishment [Ristic et al. 2004, pg. 43]. For this reason, different ideas have been proposed concerning the resampling schedule, i.e. performing the resampling step only on some chosen time steps. This will be discussed in Section 3.4.2.

### 3.4.1 Algorithms

This section will discuss several different algorithms and heuristics that have been suggested for resampling in the particle filter. These methods are introduced with the use of the cumulative distribution function (CDF). We write the CDF of the indices of the discrete distribution as

$$F_\tilde{w}(j) := \sum_{i=1}^j \tilde{w}^i, \quad \text{for } j = 1, \ldots, M$$

and the inverse CDF as

$$F^{-1}_\tilde{w}(u) := j, \quad \text{for } q_{j-1} < u \leq q_j,$$

where $q_j = \sum_{i=1}^j \tilde{w}^i$ and $q_0 = 0$.

#### Multinomial Resampling

Consider the simulation of an IID sample from a weighted discrete distribution, i.e. the realization of $\{x^i\}_{i=1}^N$ where $x^i \sim \tilde{P}^M = \sum_{j=1}^M \tilde{w}^j \delta_{\tilde{x}^j}$ for $i = 1, \ldots, N$. This is an experiment of $N$ independent trials with $M$ possible outcomes $\{\tilde{x}^j\}_{j=1}^M$, where the probability of outcome $\tilde{x}^j$ is equal to $\tilde{w}^j$ for $j = 1, \ldots, M$. Then, the number of times that outcome $\tilde{x}^j$ occurred is a random variable $N^j$ and the joint distribution of $(N^1, \ldots, N^M)$ is said to be a multinomial distribution. Therefore, generating samples is this manner is called multinomial resampling.

The classic implementation of this resampling algorithm is with repeated use of the inversion method [Devroye 1986, pg. 28]. The algorithm is as follows. Sample $\{u^i\}_{i=1}^N \sim U_{(0,1]}$. Set $j^i = F^{-1}_\tilde{w}(u^i)$ and $x^i = \tilde{x}^{j^i}$ for $i = 1, \ldots, N$. See
Figure 3.2 for an illustration. Efficient algorithms for implementing this can be found in, e.g., Carpenter et al. [1999] and Doucet [1998].

**Stratified/Systematic/Deterministic Resampling**

Stratified random sampling is commonly used in survey sampling [Cochran 1963]. The idea is to divide a large population into a number of smaller disjoint sub-populations called **strata**. If there is little variation within each strata, then a small sample of a strata will result in a precise estimate. These precise estimates can then be combined into an estimate of the whole population.

Kitagawa [1996] describes a **stratified resampling** procedure for particle filters that partitions the $[0, 1]$ interval, i.e. the range of $F_{\tilde{w}}$, into $N$ intervals of equal width and samples one uniform random number from each interval. The intention is not to draw more than one realization from a group of particles with total weight of $1/N$. Note that in survey sampling, it is the population that is divided into strata, whereas with particle filters, it is the $[0, 1]$ interval that is divided into strata.

Similarly, we could link all $N$ uniform random numbers by simulating only one random number $u^1$ from $\mathcal{U}_{[0, 1/N]}$ and setting

$$u^i = (i - 1)/N + u^1, \quad \text{for } i = 2, \ldots, N.$$ 

One can find different names for this method in the literature, but in this work, it is referred to as **systematic resampling**, see e.g. Douc et al. [2005] and Cochran [1963].

Taking this idea to the extreme, we could replace the one random number of systematic resampling with a deterministically chosen number. This effectively removes all randomness from the algorithm, so the method is appropriately referred to as **deterministic resampling** [Kitagawa 1996].

These three methods are illustrated in Figure 3.3. It has been noted that in stratified, systematic, and deterministic resampling, the algorithm is affected by the order of the samples [Kitagawa 1996; Douc et al. 2005].

**Residual Resampling**

This method is mentioned in Liu and Chen [1998] as being computationally efficient and having smaller MC variance. The algorithm is as follows. First, take $[N\tilde{w}^i]$ copies of each $\tilde{x}^i$ for $i = 1, \ldots, M$. The remaining samples needing to be
Figure 3.2: Illustration of multinomial resampling with inversion method. There are 5 samples ($\tilde{x}^1, \tilde{x}^2, \tilde{x}^3, \tilde{x}^4, \tilde{x}^5$) with respective weights (0.2, 0.3, 0.1, 0.3, 0.1). After forming the CDF of the indices, we sample $u \sim U[0,1]$, which is say 0.68. We then determine $F^{-1}_w(u)$, which is index $j = 4$ in this case. The fourth sample is then selected as the sampled point $x$.

Figure 3.3: Illustration of different sampling methods. Interval is partitioned into $N = 7$ strata of equal width and we need one sample from each strata. Stratified sampling draws a random sample from each strata. Systematic sampling draws only one random number, which determines the remaining samples. Deterministic sampling does not draw any random samples, where we have chosen to align the samples at the center of each strata.
drawn are equal to \( N - \sum_{i=1}^{M} \lfloor N \hat{w}^i \rfloor \), which are then obtained by multinomial sampling from \( \{ \hat{x}^i \}_{i=1}^{M} \) with weights proportional to \( N \hat{w}^i - \lfloor N \hat{w}^i \rfloor \) for \( i = 1, \ldots, M \).

### 3.4.2 Resampling Schedule

As the resampling is an additional step that is introducing additional approximations into the algorithm, it has been observed that resampling at every time step is neither necessary or efficient. For example, Liu and Chen [1998] mention two such schedules for resampling: deterministic and dynamic. With a **deterministic schedule**, resampling is performed at fixed intervals while, with a **dynamic schedule**, one determines whether to resample or not as the algorithm runs. This is usually done by computing some value that is meant to be an indicator of degeneracy and then checking if it is below a predefined threshold.

#### Effective Sample Size

An indicator of degeneracy was proposed in Kong et al. [1994]. Consider random MC estimates of the integral of a function \( f \). We denote \( \hat{I}^N(f) \) as the classical MC approximation, assuming it is possible to sample from the distribution of interest, and \( \tilde{I}^N(f) \) as the approximation using importance sampling of the form in (3.4). The ratio of the variances of these two estimates, i.e. \( \frac{\mathbb{V}_q[\hat{I}^N(f)]}{\mathbb{V}_p[I^N(f)]} \), is suggested to measure the relative efficiency of the importance sampling. The authors then give an approximation of this ratio as

\[
\frac{\mathbb{V}_q[\tilde{I}^N(f)]}{\mathbb{V}_p[I^N(f)]} \approx 1 + \mathbb{V}_q[p/q].
\]  

(3.22)

Notice that this approximation conveniently no longer depends on the function \( f \) being integrated.

The approximation of (3.22) is then used to define the **effective sample size** as

\[
N_{\text{eff}} := \frac{N}{1 + \mathbb{V}_q[p/q]} = \frac{N}{\bar{E}_q[(p/q)^2]}.
\]

As this cannot be computed exactly, this is further approximated as

\[
\hat{N}_{\text{eff}} := \frac{1}{\sum_{i=1}^{N} (\hat{w}^i)^2},
\]

where the \( \hat{w}^i \) are the normalized importance weights. In practice, this can be computed every time step and when it falls below some threshold, resampling is performed.
Low values of $\hat{N}_{\text{eff}}$ can be useful to indicate inaccurate estimates, but it should be noted that estimates can be extremely inaccurate with no indication from this quantity. Such a scenario could occur, for example, if an important mode of $P$ is missed when sampling from $Q$ [Neal 1998, pg. 8].
Chapter 4

SMC for Personal Positioning

**Personal positioning** refers to calculating the position coordinates of a mobile station (MS) using measurements. Much of the motivation for the positioning of mobile devices stems from a Federal Communications Commission order that required accurate location information for emergency 911 calls made from mobile phones. The requirements are given in terms of circular error probability (CEP) so that $\text{CEP}x = y_m$ means that at least $x\%$ of the reported location solutions are within a $y$ meter radius of the true location. The mandate specifies that for mobile-based positioning, $\text{CEP}_{67} = 50\text{m}$ and $\text{CEP}_{95} = 150\text{m}$. Other applications, such as guidance and navigation, yellow page services, and location-sensitive billing, are also driving forces for the technology. [Gustafsson and Gunnarsson 2005]

SMC methods have been proposed for positioning and other related applications, such as navigation and tracking. In Section 4.1, we give a literature review. We then present the general concepts of positioning in Section 4.2. Our emphasis is on the mathematical models to describe the motion of the MS and the measurements from satellite and cellular network based systems. The Bayesian framework will be essential in our formulation of the positioning problem so that the SMC methods are applicable. In Section 4.3 and Section 4.4, we give two separate sets of simulations using SMC.
CHAPTER 4. SMC FOR PERSONAL POSITIONING

4.1 Literature Review of Related SMC Applications

SMC has been successfully applied to many related problems, and here we note some contributions. This literature review is by no means exhaustive, but is meant to indicate the application-oriented material that was consulted for this work.

The Department of Electrical Engineering at Linköping University has been quite prolific in publishing SMC applications for positioning, navigation, and tracking. Illustrations of SMC in aircraft navigation can be found in Bergman [1999], Nordlund and Gustafsson [2002], Schön et al. [2003], and Nordlund and Gustafsson [2001]. Gustafsson and Gunnarsson [2003] present a MC technique for static positioning from time difference measurements and mention that it can be easily extended to dynamic scenarios. Nordlund et al. [2002] illustrate SMC for positioning in a wireless network using distance and power measurements, where velocity is assumed known (e.g. in a car-mounted application). Several of these applications are also included in the survey, Gustafsson et al. [2002].

Morelande et al. [2003] and Moreland and Challa [2005] present a study of particle filters in target tracking although there is focus on the data association problem of tracking in clutter. Ristic et al. [2004] is one of the few books on SMC currently available and it has a tutorial with a large section of the book devoted to tracking applications. Similar work can be found in the robot localization applications of Fox et al. [2001].

4.2 Personal Positioning Background and Principles

In this section, we describe the ideas of personal positioning and presents a Bayesian approach to the problem. To start, recall that we have a signal process \( X = (x_k)_{k=0}^\infty \), which represents what we intend to estimate. For our purposes, we define \( x_k \), i.e. state of the signal at time \( k \), to consist of a MS position and velocity component \( [r_k, u_k]^T \) per coordinate and will either have 2 or 3 coordinates, i.e. \( x_k \in \mathbb{R}^4 \) or \( x_k \in \mathbb{R}^6 \) respectively.

In Section 4.2.1, we briefly discuss the various signals and measurement models commonly used in positioning. In Section 4.2.2, we present the positioning problem for the stationary case and in Section 4.2.3, for the dynamic case. The dynamic case formulation defines the system equations, (2.9) and (2.10), which connects the positioning problem to the Bayesian filtering.
4.2.1 Measurements

Positioning relies on measurements that depend on the state of the MS. Recall the general form of the measurements used in the system equations (2.10), i.e. \( y_k = h_k(x_k) + w_k \). We will be interested in formulating our measurements for positioning in this form, where now \( x_k \) is the MS state.

There are a variety of signals in cellular networks that can be used for positioning, see e.g. Syrjärinne [2001]. For example, Time of Arrival (TOA) and Round Trip Delay (RTD) positioning use propagation times of signals, Received Signal Strength (RSS) positioning uses measured signal strengths, and Angle of Arrival (AOA) positioning uses directions of signal arrival. The signal travel time used in TOA is proportional to the distance between the MS and base station (BS), so we model these measurements as range measurements, i.e.

\[
h_k(x_k) = \| r^b_k - r_k \|
\]

where \( r^b_k \) is the known position of a base station. In addition, the coverage area of a base station, known as a cell, is normally divided into three sectors so that if the current sector is known, it can be used to assist positioning, see e.g. Sun et al. [2005]. If available, base station information about the maximum range of detection can also be incorporated. We may refer to base station sector and maximum range information as “measurements”.

Satellite based networks have their own signals and measurements, see e.g. Kaplan [1996]. Measuring signal travel times typically requires a synchronized network. However, in the GPS, the MS clock is not synchronized to the clocks of the satellites, which results in an added bias term for the range measurements. These biased range measurements will be called pseudorange measurements, where the observation function can be modelled as [Parkinson and Spilker 1996, pg. 410]

\[
h_k(x_k) = \| r^s_k - r_k \| + b_k
\]

where \( r^s_k \) is the known position of a satellite (SV, i.e. space vehicle) at time \( k \) and \( b_k \) is the unknown, time-dependent bias term. In addition, delta pseudorange measurements are available, which rely on signal Doppler shifts produced by the satellite and user motion. The observation function is modelled as [Parkinson and Spilker 1996, pg. 411]

\[
h_k(x_k) = \frac{( r^s_k - r_k )^T}{ \| r^s_k - r_k \| } ( u^s_k - u_k ) + \dot{b}_k,
\]

where \( r^s_k \) and \( u^s_k \) are the position and velocity of a satellite at time \( k \) respectively, and \( \dot{b}_k \) is an unknown time-dependent bias term.

Since, at a given time step \( k \), the \( b_k \) bias terms are the same for all pseudorange measurements and the \( \dot{b}_k \) bias terms are the same for all delta pseudorange measurements, we can use differenced measurements to subtract out the bias, see
e.g. Ali-Löytty et al. [2005]. This means that for two satellites $s_0$ and $s_1$ that can be seen at a time $k$, the **differenced pseudorange measurement** uses the model

$$h_k(x_k) = \|r^s_{k1} - r_k\| - \|r^s_{k0} - r_k\|$$

and the **differenced delta pseudorange measurement** uses the model

$$h_k(x_k) = \frac{(r^s_{k1} - r_k)^T}{\|r^s_{k1} - r_k\|} (u^s_{k1} - u_k) - \frac{(r^s_{k0} - r_k)^T}{\|r^s_{k0} - r_k\|} (u^s_{k0} - u_k).$$

Using differenced measurements requires that we have at least two measurements. It should be noted that a vector of differenced measurements can have correlated errors.

We will frequently refer to the measurements of this section in our simulations. Note that the Euclidean norm $\|\cdot\|$ is a nonlinear operation so that these observation functions $h_k$ are nonlinear functions of $x_k$.

### 4.2.2 Static Case

The basic approach to positioning solves the problem at each time step separately using only measurements from the given time step—hence, we say it is a **static** or **stationary problem**. The measurements that are used are typically only functions of the MS position and an error term, i.e. $y_k = h_k(r_k) + w_k$. The static positioning problem is then to find the position that minimizes some loss function $V(r_k)$. Multiple measurements are generally needed to prevent underdetermined cases with multiple solutions.

Several common choices for loss functions are given in Gustafsson and Gunnarsson [2005], which we list in Table 4.1. There are special cases where the nonlinear least squares problem can be reformulated as a linear least squares problem so that there exists a closed-form solution, see e.g. Smith and Abel [1987]. In general, however, the solution of these problems requires an iterative numerical search algorithm, such as a steepest descent or Gauss-Newton method.

<table>
<thead>
<tr>
<th>Loss Functions $V(r_k)$</th>
<th>Differential Expression</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nonlinear Least Squares</td>
<td>$(y_k - h_k(r_k))^T (y_k - h_k(r_k))$</td>
</tr>
<tr>
<td>Weighted Nonlinear Least Squares</td>
<td>$(y_k - h_k(r_k))^T \Sigma_{w_k}^{-1} (y_k - h_k(r_k))$</td>
</tr>
<tr>
<td>Maximum Likelihood</td>
<td>$-\log p_{w_k} (y_k - h_k(r_k))$</td>
</tr>
</tbody>
</table>
4.2.3 Dynamic Case

Positioning can also be formulated as a dynamic problem, where, assuming some motion model, we can use the solution of a given time step to infer knowledge about the solution of the next time step. Incorporating this additional information should improve the estimates. The dynamic positioning approach does not require multiple measurements at each time step as the static positioning approach does; this could be an important consideration in situations where measurements are scarce.

There are a wide variety of mathematical models that describe the evolution of an object’s kinematic state with respect to time, see e.g. Li and Jilkov [2003]. Recall the general form of the signal model used in the system equations (2.9), i.e. \( x_k = f_{k-1}(x_{k-1}) + v_{k-1} \), where now the state of the signal is the position and velocity of the MS. Note that the choice of motion model defines the signal model.

We will refer to the following model in the simulations of this work. The evolution of the state in continuous time can be described by a white noise acceleration model. The discretized state equation for each coordinate is written as

\[
\begin{align*}
x_k &= F_{k-1}x_{k-1} + v_{k-1}, \\
F_{k-1} &= \begin{bmatrix} 1 & \Delta t_k \\ 0 & 1 \end{bmatrix},
\end{align*}
\]

where \( \Delta t_k := t_k - t_{k-1} \). The signal noise is independent in time and given such that \( v_{k-1} \sim \mathcal{N}(0, \Sigma_{v_{k-1}}) \) with

\[
\Sigma_{v_{k-1}} = \begin{bmatrix} \frac{1}{2} \Delta t_k^3 & \frac{1}{2} \Delta t_k^2 \\ \frac{1}{2} \Delta t_k^2 & \Delta t_k \end{bmatrix} \gamma,
\]

where \( \gamma \) is a design parameter. If \( \gamma \) is chosen such that changes in the velocity are small compared to the actual velocity, then this is called the nearly constant-velocity model. [Bar-Shalom and Li 1998]

Note that having defined the observation models and signal models for the state equations of (2.9) and (2.10), then the Bayesian recursions give the optimal solution in the form of a probability distribution, i.e. the posterior. Although the white noise acceleration model defines a linear and Gaussian signal model, the observation models are nonlinear, which prevents a tractable form of the posterior. Extended Kalman filtering is a natural approach and is commonly applied to positioning using satellite measurements [Kaplan 1996]. However, the nonlinearity from terrestrial-based measurements can be more severe in which case the linear approximations made by the EKF are inadequate, see e.g. Ali-Löytty et al. [2005].

The nonlinear filtering formulation of positioning is ideally suited for SMC. Of course, the computational and memory requirements for SMC are quite large so
that it might not be currently a feasible solution to implement on a consumer’s mobile phone. Nonetheless, it is worthwhile to consider the pros and cons of such an approach, as it will most likely be feasible eventually. At the very least, an SMC filter could be used as a reference to evaluate the quality of other Bayesian filters.

4.3 Simulation Set 1: Testbench

We have implemented the particle filter for a positioning testbench—a software package that generates positioning-style measurements and computes a number of filter evaluation statistics\(^1\). We first present a brief description of the testbench, followed by two separate simulations using it. In the first set of simulations, given in Section 4.3.2, we illustrate how the PF performs in different positioning scenarios. In the next set of simulations, given in Section 4.3.3, we consider how the choice of importance distribution affects the performance of the filter.

4.3.1 Description

This section describes the models that were used in the simulations. Note that because we are using simulated data, the correct motion and measurement models are available for the filters. In any application, these models would have to be estimated and there would most likely be modelling errors, which then complicates the comparison of different filters. We have not considered modelling errors in this work.

State and Dynamics

All of the simulations of this section used a position-velocity model with three-dimensional position and velocity in a local East-North-Up (ENU) reference frame, i.e.

\[
x_k = [r_e^k r_n^k r_u^k u_e^k u_n^k u_u^k]^T.
\]

The state’s initial distribution \(P_{x_0}\) is Gaussian and known. No model is given to describe how the the bias terms \((b_k, \_b_k)\) in the pseudorange and delta pseudorange measurements evolve in time, so we do not attempt to estimate them; we use differenced measurements to eliminate the bias terms.

\(^1\)The positioning testbench software, including data generation and filter evaluation statistics, was designed and implemented by Niilo Sirola, of the Personal Positioning Algorithms Research Group at Tampere University of Technology.
All simulated trajectories are 120 seconds long with time steps of one second. The signal model function is linear, time-invariant, and defined by the matrix

\[
F = \begin{bmatrix}
  1 & 0 & 0 & 1 & 0 & 0 \\
  0 & 1 & 0 & 0 & 1 & 0 \\
  0 & 0 & 1 & 0 & 0 & 1 \\
  0 & 0 & 0 & 1 & 0 & 0 \\
  0 & 0 & 0 & 0 & 1 & 0 \\
  0 & 0 & 0 & 0 & 0 & 0.9
\end{bmatrix}.
\]

The signal noise is independent in time and is modelled as Gaussian with zero mean and covariance given by (4.2) with \( \gamma \) equal to \( 2 \text{ m}^2/\text{s}^3 \) for north and east directions and \( 0.1 \text{ m}^2/\text{s}^3 \) for the up direction. Note that this resembles the white noise acceleration model mentioned in Section 4.2.3 but the 0.9 term in the \( F \) matrix is a heuristic to restrict motion in the vertical direction. We use this heuristic of the testbench without justification, but note that the data is simulated so for our purposes, it is important only that the filters use the same models that generated the data.

**Measurements**

The testbench simulates the measurements mentioned in Section 4.2.1: range, pseudorange, delta pseudorange, sector, and maximum range. The range, pseudorange, and delta pseudorange measurements have an additive error term, which is independent for each measurement and is independent in time. The errors are modelled as Gaussian with zero-mean and a known scalar variance \( \sigma^2 \), i.e. \( w_k \sim \mathcal{N}(0, \sigma^2) \). Positions and velocities of satellites are based on real data, measured on the Tampere University of Technology campus on June 18, 2003. The base station positions are simulated and are placed on a hexagonal grid on the local East-North plane.

There are five different positioning scenarios that were explored with the testbench, see Table 4.2 for their general characteristics. Note that at a given time step, a base station may report a range measurement, information concerning the

<table>
<thead>
<tr>
<th>Case</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>BS</td>
<td>low</td>
<td>high</td>
<td>med</td>
<td>med</td>
<td>none</td>
</tr>
<tr>
<td>Sectors</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>yes</td>
<td>no</td>
</tr>
<tr>
<td>SV</td>
<td>none</td>
<td>none</td>
<td>low</td>
<td>med</td>
<td>high</td>
</tr>
</tbody>
</table>
maximum range of detection, and possibly sector information, while a satellite may report a pseudorange and a delta pseudorange measurement.

### 4.3.2 Multiple Trajectories: EKF vs PF at Different Sample Sizes

From each positioning environment given in Table 4.2, we simulate 100 different trajectory realizations, each with a corresponding measurement realization, and run the PF once on each set of measurements. Evaluation statistics are computed over each separate positioning environment so that we can get a general idea of how the PF behaves in the different environments. See Table 4.3 for details on the generated measurements.

This set of simulations was intended to give a “system level” perspective of the PF performance where the details of individual trajectories are intentionally left unanalyzed. Later simulations will look at the PF performance in closer detail.

#### Evaluation Criteria

We will now present a brief description of the evaluation criteria used in this set of simulations. All of the computed statistics use only the East-North position dimensions. The Euclidean distance between the filter mean estimates and the true track is calculated at every time step and the average of these errors is referred to as “Mean error [m]”. The “95% error [m]” statistic shows the distance that 95% of the errors were within, while “Inside 50m [%]” and “Inside 150m [%]” gives

<table>
<thead>
<tr>
<th>Case</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td># of BS</td>
<td>0–3</td>
<td>0–5</td>
<td>0–4</td>
<td>0–4</td>
<td>0</td>
</tr>
<tr>
<td>% of time with # BS &gt; 1,2</td>
<td>27.1, 2</td>
<td>67.7, 21.3</td>
<td>40.1, 3.8</td>
<td>44.7, 7.4</td>
<td>–</td>
</tr>
<tr>
<td>(\sigma_{\text{range}}) [m]</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>–</td>
</tr>
<tr>
<td># of sectors</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>3</td>
<td>–</td>
</tr>
<tr>
<td>Cell radius [km]</td>
<td>1.5</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>–</td>
</tr>
<tr>
<td># of SV</td>
<td>0</td>
<td>0</td>
<td>0–4</td>
<td>0–4</td>
<td>1–6</td>
</tr>
<tr>
<td>% of time with # SV &gt; 1,2</td>
<td>–</td>
<td>–</td>
<td>55.2, 23.4</td>
<td>69.8, 41.9</td>
<td>94.3, 65.2</td>
</tr>
<tr>
<td>(\sigma_{\text{pseudorange}}) [m]</td>
<td>–</td>
<td>–</td>
<td>10–30</td>
<td>8–30</td>
<td>8–20</td>
</tr>
<tr>
<td>(\sigma_{\text{delta pseudorange}}) [m/s]</td>
<td>–</td>
<td>–</td>
<td>1–4</td>
<td>1–3</td>
<td>1–2</td>
</tr>
</tbody>
</table>

Table 4.3: Testbench Cases
the percentage of time steps that the filter estimates were within the specified range. The “Yield [%]” statistic gives the percentage of trajectories that the filter was able to complete successfully. The PF might not be able to complete a trajectory successfully if all the samples are in an insignificant region of the state-space and the importance weights sum to zero; in this case, the weights cannot be normalized due to the division by zero and the filter cannot continue.

Tested Filters

Only one choice of importance distribution for the PF is considered here, along with an EKF. As we have claimed that the approximations made by the EKF may be inadequate in a number of cases, it is worthwhile to include EKF results for comparison. The EKF does not use the base station sector or maximum range information since they are not linear and Gaussian measurements and it is not straightforward to incorporate the information in this filter. On the other hand, the PF simply incorporates this information with the weight updates, i.e. if a given sample is outside of the reported sector or beyond the maximum range of detection, then it is assigned zero weight.

The PF samples from the empirical prediction distribution, i.e. we use the SIR1 filter. Systematic sampling is used to draw $M = N$ samples from the importance distribution at every step, and the final resampling step is skipped. We have run the PF using a number of different sample sizes: $10^2$, $10^3$, $10^4$, $10^5$, $10^6$, $2 \cdot 10^6$. The largest sample sizes was around where the computational complexity began to be a burden.

The intention of this set of simulations is as follows. The quality of the PF approximations of the posterior should improve with larger sample sizes and should degrade with smaller sample sizes. Correspondingly, we expect the quality of the filter estimates to improve with larger sample sizes and degrade with smaller sample sizes. If an increase of sample size does not change the filter results, then we can be confident that we have a sufficient number of samples to adequately represent the posterior distributions. On the other hand, if the filter results continue to change as we increase the sample size, then we cannot be sure if the results would continue to change at even larger sample sizes. The main idea of this simulation is to repeat this analysis for the different positioning scenarios and determine which environments present more difficulty for the PF.

Results

The simulation results for each positioning environment are presented separately in Table 4.4. First, notice that there is not much difference between the filters
considering the “Inside 50m [%]” and “Inside 150m [%]” criteria. Although it is perhaps interesting to compare these numbers for the different test cases, it does not emphasize the difference between the filters for a given test case. The “Yield [%]” results were included to indicate the few times that the small sample PFs could not continue. The comments of this section will focus on the results for the two remaining criteria, “Mean error [m]” and “95% error [m]”.

Let us first compare the PF at different sample sizes. In general, the PF results seem to be converging to specific values as the sample size increases. It is reassuring that the results of all test cases do not change significantly after, say $10^4$ samples. For a personal positioning application, this difference would most likely be irrelevant and the larger number of samples would not be necessary. On the other hand, from the point-of-view of approximating the optimal filter, it is disturbing that for some cases, the “95% error [m]” results continue to change as the sample size increases. For Cases 2 and 4, the results stabilize relatively quickly ($10^4$ samples), while Case 1 needed much more samples ($10^6$ samples). The results for Cases 3 and 5 do not indicate that there are sufficient samples for this behavior.

Care should be taken when interpreting the results from the PFs with smaller sample size; if increasing the sample size changes the results, then repeating the experiment with the same sample size could also change the result. As an example, different realizations of the “95% error [m]” result for the Case 2 data at a sample size of $10^3$ gave 185, 185, 175, 190, and 182 m. This randomness explains the nonintuitive scenario when there are smaller errors for smaller sample sizes which, of course, is not indicative of better performance.

The fact that the PF seems to require a large number of samples for Case 5 is peculiar since this case uses only satellite measurements, which are more accurate than the base station data. This oddity was due to trajectories in which there were no more than two satellites seen at each step. Such a scenario can result in narrow distributions that can grow quite large, which then require a large number of samples for a good approximation. This will be looked at in the next section.

The scenarios when the PF outperformed the EKF are not surprising: the difference is most visible in the noisy and highly nonlinear scenarios of Case 1 and 3, with no apparent difference for the satellite-only measurements of Case 5.

To make conclusions about the PF performance in specific scenarios, we can say the following. With fewer measurements, the PF behaves more unpredictably and could require a large number of samples (see Cases 1 and 3). Also, the use of maximum range and sector information from base stations could be useful for the PF (compare Cases 4 and 5). Finally, the more nonlinear and noisy scenarios should show more of an improvement of the PF over the EKF.
Table 4.4: Simulation Results for Cases 1-5

<table>
<thead>
<tr>
<th>Case 1</th>
<th>PF</th>
<th>EKF</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sample size</td>
<td>$10^2$</td>
<td>$10^3$</td>
</tr>
<tr>
<td>Mean error [m]</td>
<td>231</td>
<td>217</td>
</tr>
<tr>
<td>95% error [m]</td>
<td>837</td>
<td>801</td>
</tr>
<tr>
<td>Inside 50m [%]</td>
<td>21.8</td>
<td>25.6</td>
</tr>
<tr>
<td>Inside 150m [%]</td>
<td>55.2</td>
<td>59.2</td>
</tr>
<tr>
<td>Yield [%]</td>
<td>88</td>
<td>99</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Case 2</th>
<th>PF</th>
<th>EKF</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sample size</td>
<td>$10^2$</td>
<td>$10^3$</td>
</tr>
<tr>
<td>Mean error [m]</td>
<td>75.7</td>
<td>70.3</td>
</tr>
<tr>
<td>95% error [m]</td>
<td>200</td>
<td>191</td>
</tr>
<tr>
<td>Inside 50m [%]</td>
<td>43.1</td>
<td>46.2</td>
</tr>
<tr>
<td>Inside 150m [%]</td>
<td>90.3</td>
<td>91.7</td>
</tr>
<tr>
<td>Yield [%]</td>
<td>90</td>
<td>100</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Case 3</th>
<th>PF</th>
<th>EKF</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sample size</td>
<td>$10^2$</td>
<td>$10^3$</td>
</tr>
<tr>
<td>Mean error [m]</td>
<td>113</td>
<td>97.4</td>
</tr>
<tr>
<td>95% error [m]</td>
<td>437</td>
<td>389</td>
</tr>
<tr>
<td>Inside 50m [%]</td>
<td>52.7</td>
<td>59.3</td>
</tr>
<tr>
<td>Inside 150m [%]</td>
<td>83.7</td>
<td>87.8</td>
</tr>
<tr>
<td>Yield [%]</td>
<td>96</td>
<td>100</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Case 4</th>
<th>PF</th>
<th>EKF</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sample size</td>
<td>$10^2$</td>
<td>$10^3$</td>
</tr>
<tr>
<td>Mean error [m]</td>
<td>55.7</td>
<td>43.6</td>
</tr>
<tr>
<td>95% error [m]</td>
<td>138</td>
<td>128</td>
</tr>
<tr>
<td>Inside 50m [%]</td>
<td>70.9</td>
<td>75.7</td>
</tr>
<tr>
<td>Inside 150m [%]</td>
<td>95.6</td>
<td>96</td>
</tr>
<tr>
<td>Yield [%]</td>
<td>87</td>
<td>98</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Case 5</th>
<th>PF</th>
<th>EKF</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sample size</td>
<td>$10^2$</td>
<td>$10^3$</td>
</tr>
<tr>
<td>Mean error [m]</td>
<td>61.8</td>
<td>60.6</td>
</tr>
<tr>
<td>95% error [m]</td>
<td>260</td>
<td>270</td>
</tr>
<tr>
<td>Inside 50m [%]</td>
<td>75.6</td>
<td>80.2</td>
</tr>
<tr>
<td>Inside 150m [%]</td>
<td>91.5</td>
<td>91.7</td>
</tr>
<tr>
<td>Yield [%]</td>
<td>98</td>
<td>100</td>
</tr>
</tbody>
</table>
4.3.3 Comparing PFs with Different Importance Distributions

As the PF relies on random simulations, the estimates conditioned on a realization of the observation process are random. This is in contrast with, e.g. an EKF, which forms an estimate that is deterministic conditioned on the realized measurements. PF performance can be evaluated by filtering one realization of the observation process multiple times and analyzing how the estimates vary over different realizations of the algorithm, i.e. the MC variation of the filter. Better PFs vary less. In this set of simulations, we will be interested in the MC variation of the PF mean estimates, and more specifically, comparing this MC variation from PFs using different importance distributions.

One trajectory and a Case 5 set of measurements was generated so that there are only satellites measurements, i.e. pseudoranges and delta pseudoranges. In Figure 4.1(a), the trajectory realization is plotted in the East-North plane along with the PF estimates from one realization of the algorithm. This set of measurements was a rare case in that there are no more than two satellites seen at each time step; the filters then have no more than one differenced pseudorange measurement and one differenced delta pseudorange measurement at each time step. This causes the significant region of the likelihood function to be quite narrow, which in effect, causes the significant region of the posterior distribution to be narrow also and to “stretch out” as time progresses from the uncertainty in the dynamic model.

This set of measurements was chosen for two reasons. First, we wanted to illustrate the scenario that can occur when the Bayesian mean does not resemble the true track. Notice how the true track veers to a northern direction while the filter mean estimates continue eastward. Intuitively, one might want to compare the filter estimates with the true track and claim that the “better” filter is the one giving estimates closer to the true track. We must keep in mind that the “better” Bayesian filter is the one that gives a closer approximation of the true Bayesian solution, and is not necessarily the filter that has an estimate closer to the true track. Cases such as this complicate the comparison of different Bayesian filters because we do not know the true Bayesian solution so we must evaluate how close the filters approximate a value we do not know.

The second reason that this set of measurements was chosen was to illustrate the MC variation of the PF estimates. A PF was run 500 times on this set of measurements and we have plotted the covariances of the filter mean estimates as $3\sigma$ ellipses in Figure 4.1(b), which is a close-up of the subregion box in Figure 4.1(a). These ellipses characterize the MC variation that more elaborate PFs should reduce. The ellipses are quite narrow so that it is convenient to only look at the ellipse major-axis vertices; lines through every time step’s ellipse vertices
are drawn in Figure 4.1(c).

We should point out that in computing the covariance of the mean estimate, we are using the ensemble average (the average over the algorithm realizations) in place of the unknown true Bayesian mean. Many simulations in the literature have used the true track in place of the true Bayesian mean, which of course, would not be appropriate here. Similar results were obtained by using an estimate of the true Bayesian mean from a large sample SIR1 PF, e.g. at $10^6$ and $10^7$.

The different importance distributions tested include the four examples from Section 3.3.2, i.e. SIR1, SIR2, SIR3, and SIR4. All the SMC methods used systematic resampling to simulate $M = N = 10^4$ samples from the importance distribution at every step, and skipped the final resampling step. The effectiveness of the different importance distributions is illustrated in Figure 4.1(d). Each successive filter shows an improvement over the previous, i.e. SIR2 was better than SIR1, SIR3 was better than SIR2, etc. These improvements can be compared with an SIR1 PF with larger sample sizes in Figure 4.1(e).

The improvement of the different importance distributions is slight. One motivation for the construction of an importance distribution based on local linearizations is for very accurate sensors, i.e. peaked likelihoods, see e.g. van der Merwe et al. [2000, pg. 2]. Considering the types of measurements available in our scenarios, the satellite measurements are the more accurate ones and the base station measurements are noisier. We should expect even less of an improvement with only base station data. Peaked likelihoods could also result from many measurements received at a given time step, but we are less concerned with such situations because a static positioning solution could then be appropriate, see Section 4.2.2. Another motivation for the linearizations is for data that undergoes sudden changes, see e.g. van der Merwe et al. [2000, pg. 2]; this subject was not considered in this work.
Figure 4.1: (a) PF estimates from one realization of the algorithm. (b) Subregion of (a) showing a number of mean realizations and covariance of mean estimates. (c) Envelope characterizing the covariance ellipses. (d) Comparison of different importance distributions at a sample size of $10^4$. (e) Comparison of SIR1 at different sample sizes. All covariance ellipses are $3\sigma$ contours and are plotted at 30 second intervals.
4.4 Simulation Set 2: Comparing PF Posterior Approximations

In the literature, many PF simulations focus on the MC variation of the mean estimates, i.e. the randomness introduced by the MC algorithm that can be observed from the empirical mean estimates. In this section, we develop a method to compare PF performance that uses the distribution approximation itself rather than a single estimate formed from it.

A distribution analysis could be more useful than an estimate analysis. As examples, two filters could give similar mean estimates although one of them has a distribution “closer” to the true distribution. Also, in a bimodal case, the mean could be between the two modes in a region of the state-space where there is little probability of the target being located; in a case such as this, the mean is less interesting to analyze. We will discuss some of the background related to comparing distributions and then apply these methods to the comparison of the particle filters.

4.4.1 Comparing Distributions with $\chi^2$-tests

The comparison of distributions from empirical data is an established branch of statistics with applications in numerous fields. These techniques are primarily hypothesis tests. This involves computing a test statistic which, under some assumed hypothesis, should have a certain distribution. If the computed statistic falls outside a given confidence region of the assumed distribution, the hypothesis is rejected and is not rejected otherwise. [Bar-Shalom and Li 1998]

The chi-squared ($\chi^2$) test is one such procedure for distribution comparisons, see e.g. Cochran [1952] or Moore [1986] for an overview. When comparing empirical data with a known distribution, the idea relies on grouping the $n$ observed data samples into $B$ mutually exclusive bins and forming a hypothesis that states the probability $p_j$ that an observed data point will fall into the $j^{th}$ bin, for $j = 1, \ldots, B$. To check this hypothesis, the following quantity is used [Cochran 1952]

$$
\psi = \sum_{j=1}^{B} \frac{(o_j - e_j)^2}{e_j},
$$

where $o_j$ is the observed number of samples in the $j^{th}$ bin and $e_j$ is the expected number of samples for the $j^{th}$ bin, i.e. $e_j = n p_j$. Pearson showed in 1900 that if the hypothesis is true and $n$ is large, then the distribution of the test statistic $\psi$ is approximately $\chi^2$ with $B - 1$ degrees of freedom ($\chi^2_{B-1}$) [Cochran 1952].
The idea can be extended from comparing empirical data with a known distribution to comparing two sets of empirical data with each other. In this case, the hypothesis is that the two samples are from a common distribution. The form of the computed test statistic is [Baggerly 2001]

\[ \psi = \sum_{i=1}^{2} \sum_{j=1}^{B} \frac{(o_{ij} - e_{ij})^2}{e_{ij}}, \]  

where \( o_{ij} \) is the observed count of the \( j^{th} \) bin of the \( i^{th} \) sample set and \( e_{ij} \) is the expected count of the \( j^{th} \) bin of the \( i^{th} \) sample set given by

\[ e_{ij} = n_i \frac{(o_{1j} + o_{2j})}{(n_1 + n_2)}, \]

where \( n_i \) is the number of samples in set \( i \).

A convenient aspect of the \( \chi^2 \)-test is that it has a natural extension to multivariate scenarios. The difficult part is the construction of the bins. The simplest approach would divide the state-space into a number of equally sized bins, but this can be computationally difficult for large dimensions, i.e. a large number of bins, most of which are empty. Choosing bins of unequal size that depend on the data is a solution. This results in smaller sized bins, hence more resolution, in regions where more data is expected and larger bins, hence coarse resolution, where less data is expected. This is known as adaptive or data-dependent binning and has been discussed in the literature, see e.g. Baggerly [2001, pg. 142] for a brief overview.

### 4.4.2 Probability Binning

An interesting application of these ideas in the field of cytometry was given by Roederer et al. [2001]. Test samples, i.e. sets of multidimensional data, are to be ranked according to their similarity to a control sample, which is a sample of data chosen to represent some known behavior. A multivariate data-dependent binning technique was proposed that adaptively constructed bins according to the control sample, followed by the use of a test statistic to quantify the difference between the test and control sample. Baggerly [2001] provides a more theoretical discussion with the recommendation to use the standard two-sample \( \chi^2 \)-test statistic (4.3).

The algorithm for constructing the bins, called probability binning, is quite simple and is as follows. The variance of the control sample along each of the \( d \) dimensions is computed and the dimension with the largest variance is chosen to be divided. The sample median value of the chosen dimension is then chosen as the point at which to partition the state-space in two. This is then repeated for
each partitioned subspace, continuing until the desired number of bins has been reached. The result is a set of $d$-dimensional hyper-rectangular bins with sides parallel to the coordinate axes and each bin containing roughly the same number of control samples, see Figure 4.2. Assuming then, that a test sample is from the same distribution as the control sample, each bin will roughly have the same expected frequency.

Example

This simple example illustrates the idea of probability binning and comparing distributions using the two-sample $\chi^2$-test statistic. We use $10^6$ samples from a standard Gaussian distribution in $\mathbb{R}^4$ as the control sample to partition the state-space into $B = 64$ bins. Test samples, each of $10^4$ samples, are then generated from the same distribution, binned, and scored. Since all the test samples are from the same distribution as the control sample, we expect the realized test scores to follow the theoretical $\chi^2_{B-1}$ distribution quite well. This can be verified in Figure 4.3, where we plot the histogram of the realized test scores along with the theoretical density.

4.4.3 Application to Linear and Gaussian Filtering Scenarios

In this section, we apply the probability binning to a simple filtering scenario. Assume for a moment, that we are able to generate IID samples from the true marginal posterior $\pi_{k|k}$. We can then apply the procedure in Section 4.4.2 at each time step so that the state-space $\mathbb{R}^{n_x}$ is partitioned into bins using a sample from the true marginal posterior. It might be reasonable then to assume that the quality of a PF approximation of $\pi_{k|k}$ can be assessed using the two-sample $\chi^2$-test, where the hypothesis is that the samples from the true marginal posterior and the samples from the PF approximation are from the same distribution.

Roughly speaking, we might expect better PFs to give better test scores, where “better test scores” refers to realizations of a random variable with distribution closer to the assumed $\chi^2$ distribution. We check this empirically by repeating the simulation 1000 times and comparing the mean and variance of the realized test scores to that from the assumed $\chi^2$ distribution. For a $\chi^2_{B-1}$ distribution, the mean is $B - 1$ and the variance is $2 \cdot (B - 1)$.

To verify that the test score analysis is meaningful, we will compare our conclusions to those made from a root mean square error (RMSE) analysis of the
Figure 4.2: Probability binning in $\mathbb{R}^2$ with $B = 32$

Figure 4.3: Simple test with standard Gaussian distribution in $\mathbb{R}^4$ with $B = 64$. (Scaled) Histogram of empirically realized test scores ($10^4$ realizations). Density of $\chi^2_{B-1}$ distribution is shown in red. All test samples used $10^4$ samples and the control sample used $10^6$ samples.
mean estimates. The RMSE is given as

$$\text{RMSE}_k = \sqrt{E\left[ \left\| \hat{\mu}_k - \mu_k \right\|^2 \right]},$$

(4.4)

where $\hat{\mu}_k$ is the random posterior mean estimate and $\mu_k$ is the true posterior mean. The expectation in (4.4) is approximated by repeating the simulation 500 times and using the sample mean.

The binning of the PF samples is done as follows. At each time step, we simulate $M = N = 10^4$ samples using multinomial sampling from the importance distribution, evaluate weights, and skip the final resampling step. The binning at each time step uses the weighted samples. We draw $10^4$ samples using systematic resampling from this weighted discrete distribution, bin the equally-weighted samples, compute the test score, and discard these resampled points. If the weights are equal, which is the case for SIR4 in the linear and Gaussian scenario, then the binning can proceed without resampling.

**Simulation**

For this simulation, the state is in $\mathbb{R}^4$ with two position coordinates and two velocity coordinates, i.e.

$$x_k = [r^n_k \ r^n_k \ u^n_k \ u^n_k]^T.$$ 

We simulate a 200 second trajectory from a white noise acceleration model with a constant time step of 1 second and $\gamma = 3 \text{ m}^2/\text{s}^3$. The state’s initial distribution $P_{x_0}$ is Gaussian and known. One measurement is produced every time step and is modelled as

$$y_k = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} x_k + w_k, \quad w_k \sim \mathcal{N}\left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} \sigma^2 & 0 \\ 0 & \sigma^2 \end{bmatrix}\right).$$

As this filtering problem is linear and Gaussian, the marginal posterior distribution at each time step is Gaussian and can be computed with the Kalman filter. In this case, we can then draw samples directly from the marginal posterior to use as a control sample to partition the state-space. A particle filter can then be run on the data, where at every time step, we bin the samples and compute test scores. Of course, this is just for illustrative purposes because the Kalman filter solves the problem exactly and there is no reason to resort to particle filtering.

We again test the four PF examples of Section 3.3.2, i.e. SIR1, SIR2, SIR3, and SIR4. Note that this is a special case for SIR3 and SIR4 since there is no need for linearization. All filters sampled $M = N = 10^4$ samples from the importance distribution and skipped the final resampling step. For each filter,
four separate simulations were run, each using a different variance $\sigma^2$ for the noise in the measurement model: $10^4$, 25, 1, 0.1 m$^2$.

The mean and variance of the test scores at a single time step are given in Table 4.5. It is interesting to note that the test scores rarely resemble the $\chi^2$ theoretical distribution. In fact, this occurs only with the SIR4 filter in the $\sigma^2 = 1$ and 0.1 m$^2$ cases. However, there appears to be different behavior for the different filters. For cases with larger measurement noise variances ($\sigma^2 = 10^4$ and 25 m$^2$), the two filters using alternative weights in the importance distribution (SIR2 and SIR4) have smaller means and variances. For cases with smaller measurement noise variances ($\sigma^2 = 1$ and 0.1 m$^2$), the two filters using alternative components in the importance distribution (SIR3 and SIR4) show smaller means and variances, while the SIR2 results behave erratically. It might be reasonable then to conclude that the filters having test scores closer to the theoretical distribution $\chi^2_{B-1}$, i.e. having smaller test score means and variances, are working better.

Similar results, concerning the score mean, can be seen in Figure 4.4, where the score mean is plotted over the whole simulation. Note that all the results are rather similar with large $\sigma^2$, and the differences become more apparent as $\sigma^2$ decreases. Also note that in Figure 4.4(a) and Figure 4.4(b) there is near identical results for the SIR1 and SIR3 filters and for the SIR2 and SIR4 filter. In Figure 4.4(d), the SIR2 score means are quite large and are outside of the plotted region. The intention of these plots was not to display the actual score means, but instead to show relative performance of the different filters.

Table 4.5: Mean and variance of two-sample $\chi^2$-test statistic for linear-Gaussian scenario at time step $k = 180$ with number of bins $B = 64$.

<table>
<thead>
<tr>
<th>$\sigma^2$</th>
<th>$\chi^2$</th>
<th>SIR1</th>
<th>SIR2</th>
<th>SIR3</th>
<th>SIR4</th>
</tr>
</thead>
<tbody>
<tr>
<td>$10000$ m$^2$</td>
<td>Mean</td>
<td>63</td>
<td>781</td>
<td>611</td>
<td>784</td>
</tr>
<tr>
<td></td>
<td>Variance</td>
<td>126</td>
<td>$4.02 \cdot 10^4$</td>
<td>$2.69 \cdot 10^4$</td>
<td>$4.40 \cdot 10^4$</td>
</tr>
<tr>
<td>$25$ m$^2$</td>
<td>Mean</td>
<td>63</td>
<td>245</td>
<td>119</td>
<td>237</td>
</tr>
<tr>
<td></td>
<td>Variance</td>
<td>126</td>
<td>2823</td>
<td>586</td>
<td>2847</td>
</tr>
<tr>
<td>$1$ m$^2$</td>
<td>Mean</td>
<td>63</td>
<td>276</td>
<td>730</td>
<td>152</td>
</tr>
<tr>
<td></td>
<td>Variance</td>
<td>126</td>
<td>2561</td>
<td>$5.32 \cdot 10^6$</td>
<td>705</td>
</tr>
<tr>
<td>$0.1$ m$^2$</td>
<td>Mean</td>
<td>63</td>
<td>929</td>
<td>$2.65 \cdot 10^4$</td>
<td>112</td>
</tr>
<tr>
<td></td>
<td>Variance</td>
<td>126</td>
<td>$2.94 \cdot 10^4$</td>
<td>$2.30 \cdot 10^9$</td>
<td>394</td>
</tr>
</tbody>
</table>
We now consider an analysis of the RMSE of the mean estimates. The RMSE values are divided by the SIR1 RMSE values and plotted in Figure 4.5. These relative RMSE values show how the MC variation of mean estimates for the different filters compare to SIR1: values lower than one indicate improved performance compared to SIR1 (less MC variation) while values above one indicate worse performance (more MC variation). With the largest noise variance ($\sigma^2 = 10^4 \text{ m}^2$), all the filters seem to have similar MC variance. With $\sigma^2 = 25 \text{ m}^2$, the SIR2 and SIR4 filters clearly have less MC variation, while for cases with smaller measurement noise variance ($\sigma^2 = 1 \text{ and } 0.1 \text{ m}^2$), the SIR3 and SIR4 filters have less MC variance. Also, the SIR2 behaves erratically at the smaller measurement noise variance.

In summary, similar conclusions about the relative performance of the different PFs can be found using alternative criteria, i.e. the distribution analysis and the RMSE analysis of the mean estimates. The usefulness of this distribution analysis is limited because we required an analytic form of the marginal posterior. In Section 4.4.4, we extend the analysis to the general filtering scenario.
Figure 4.4: Mean of two-sample $\chi^2$-test statistic for linear-Gaussian scenario
Figure 4.5: Relative RMSE with respect to SIR1 for linear-Gaussian scenario
4.4.4 Application to General Filtering Scenarios

We would like to apply this distribution analysis to PFs in the general filtering scenario where we do not have an analytic form for the posterior. The difficulty is then to determine a control sample that can be used to partition the state-space. Ideally, we would have some algorithm that was known to produce IID samples from the marginal posterior that we could use to produce the control sample. In the absence of such an ideal algorithm, we have chosen one PF algorithm as the reference and proceed with the analysis.

The partitioning of the state-space using a reference PF is done as follows. We have used the SIR3 importance distribution and, at each time step, draw $M = 10^7$ samples from the importance distribution, evaluate weights, and then resample $N = 10^4$ samples from this weighted distribution. The importance sampling uses deterministic sampling ($10^3$ samples from each mixture component) and the resampling uses systematic sampling. The binning at each time step uses the $M$ weighted samples. We draw $10^6$ samples from this weighted discrete distribution, perform the probability binning on these equally-weighted samples, and discard these resampled points. It does seem perhaps more elegant to instead use the $N$ equally-weighted samples to form the bins, but using a small number of points in the control sample (relative to the number of points of the test samples) was noticed to affect the test score distribution, see Figure 4.6.

Our choice of using SIR3 instead of SIR4 for the reference PF deserves discussion and our reasoning was as follows. A mixture distribution covers the same region of the state-space regardless of what the mixture weights are. Better weights of the mixture may result in a more efficient sampler, i.e. a smaller sample size $N$ is needed for a good approximation; however, a large sample $M$ from the unweighted mixture, where $M$ is much larger than $N$, is inefficient but should give approximations that are just as good, if not better. In addition, the alternative weights in SIR2 caused erratic behavior for the small measurement noise scenarios, so it was decided to be cautious and not use any alternative weights in the reference PF.

We repeat the experiments of Section 4.4.3 but now use a PF approximation of the posterior as the control sample. It should be mentioned that the simulation scenario of this section is the same simulation scenario as that in Heine [2005], i.e. the same signal and measurement models and the same realized measurement and signal process. We have reproduced the relative RMSE plots of that publication here for convenience. The contribution of this work is the application of the probability binning method to the comparison of the PFs.
Figure 4.6: Same example as in Section 4.4.2, but now using different number of samples $N$ for the control sample. (Scaled) Histogram of empirically realized test scores ($10^4$ realizations). Density of $\chi^2_{B-1}$ distribution is shown in red. All test samples used $10^4$ samples.
CHAPTER 4. SMC FOR PERSONAL POSITIONING

Simulation

We use the same signal model and realized signal process as the simulation in Section 4.4.3, but now use range measurements. It is the nonlinearity of the range measurements that is the distinguishing feature of this set of simulations and that prevents the posterior distribution from having an analytical form.

The range measurements are of the form

\[ y_k = \| r^b - r_k \| + w_k, \quad w_k \sim \mathcal{N}(0, \sigma^2), \]

where \( r^b \) is the known position of a base station. Three base stations are used and, at each time step, one base station produces a range measurement. In the simulation of the observation process, the probability of a base station producing a measurement is inversely proportional to the squared distance to the target.

Again, for each filter, four separate simulations were run, each using a different variance \( \sigma^2 \) for the noise in the measurement model: \( 10^4 \), 25, 1, 0.1 m². We repeat the experiments of Section 4.4.3 and present results along with relative RMSE plots. The true posterior mean \( \mu_k \) in the RMSE analysis, see (4.4), is approximated using the same reference PF that was used for creating the bins.

We first consider the \( \chi^2 \) results. Due to weights summing to zero in the PF algorithm on the \( \sigma^2 = 0.1 \) m² case, the SIR1 results used only 995 realizations and the SIR2 results used only 711 realizations for that case. Comparing Table 4.5 and Table 4.6, it is disturbing that the mean and variance values of the test scores in the nonlinear scenario are larger than those in the linear scenario. This could be due to the poor approximation of the true posterior, i.e. the reference PF. If we compare the relative performance of the different nonlinear filters, i.e. which filters have the smaller test score means and variances, then the conclusions are familiar: SIR3 and SIR4 have smaller means and variances for cases with smaller \( \sigma^2 \), SIR2 and SIR4 have smaller means and variances for cases with larger \( \sigma^2 \), and SIR2 is unpredictable for cases with smaller \( \sigma^2 \).

Similar results, concerning the score mean, can be seen in Figure 4.7, where the score mean is plotted over the whole simulation. Again, note that in Figure 4.7(a) and Figure 4.7(b) there is near identical results for the the SIR1 and SIR3 filters and for the SIR2 and SIR4 filter and in Figure 4.7(d), the SIR2 results are outside of the plotted region.

We now consider the RMSE results, which are given in Figure 4.8. Due to weights summing to zero in the PF algorithm on the \( \sigma^2 = 0.1 \) m² case, SIR1 is averaged over only 998 realizations and SIR2 is averaged over only 365 realizations for that case. Comparing these results to those from the linear scenario in Figure 4.5, the results are similar although the RMSE estimates are noisier, which again could be due to the poor approximation of the true posterior. The RMSE results of
the nonlinear scenario illustrate that all the filters have similar MC variation for \( \sigma^2 = 10^4 \text{ m}^2 \), SIR2 and SIR4 have less MC variation for \( \sigma^2 = 25 \text{ m}^2 \), SIR3 and SIR4 have less MC variation for \( \sigma^2 = 1 \) and 0.1 \( \text{ m}^2 \), and SIR2 is unpredictable for \( \sigma^2 = 0.1 \text{ m}^2 \).

To summarize, our conclusions from the \( \chi^2 \)-test are similar to those from the RMSE analysis. The empirical comparison of different PFs using \( \chi^2 \) techniques seems to be feasible, even in scenarios where the state-space partitioning relies on a PF.

Table 4.6: Mean and variance of two-sample \( \chi^2 \)-test statistic for nonlinear-Gaussian scenario at time step \( k = 180 \) with number of bins \( B = 64 \).

<table>
<thead>
<tr>
<th>( \sigma^2 ) (m(^2))</th>
<th>( \chi^2 )</th>
<th>SIR1</th>
<th>SIR2</th>
<th>SIR3</th>
<th>SIR4</th>
</tr>
</thead>
<tbody>
<tr>
<td>( 10^4 )</td>
<td>Mean</td>
<td>63</td>
<td>1000</td>
<td>890</td>
<td>1004</td>
</tr>
<tr>
<td></td>
<td>Variance</td>
<td>126</td>
<td>9.46 ( \times 10^4 )</td>
<td>7.14 ( \times 10^4 )</td>
<td>8.95 ( \times 10^4 )</td>
</tr>
<tr>
<td>( 25 )</td>
<td>Mean</td>
<td>63</td>
<td>349</td>
<td>259</td>
<td>346</td>
</tr>
<tr>
<td></td>
<td>Variance</td>
<td>126</td>
<td>9.09 ( \times 10^3 )</td>
<td>4.44 ( \times 10^3 )</td>
<td>1.02 ( \times 10^4 )</td>
</tr>
<tr>
<td>( 1 )</td>
<td>Mean</td>
<td>63</td>
<td>456</td>
<td>825</td>
<td>347</td>
</tr>
<tr>
<td></td>
<td>Variance</td>
<td>126</td>
<td>1.52 ( \times 10^4 )</td>
<td>1.63 ( \times 10^6 )</td>
<td>9.73 ( \times 10^3 )</td>
</tr>
<tr>
<td>( 0.1 )</td>
<td>Mean</td>
<td>63</td>
<td>1050</td>
<td>2.87 ( \times 10^4 )</td>
<td>406</td>
</tr>
<tr>
<td></td>
<td>Variance</td>
<td>126</td>
<td>1.07 ( \times 10^5 )</td>
<td>1.76 ( \times 10^9 )</td>
<td>1.25 ( \times 10^4 )</td>
</tr>
</tbody>
</table>
Figure 4.7: Mean of two-sample $\chi^2$-test statistic for nonlinear-Gaussian scenario
Figure 4.8: Relative RMSE with respect to SIR1 for nonlinear-Gaussian scenario
Chapter 5

Conclusions

In this work, we have reviewed the theory of Bayesian estimation and SMC methods and have illustrated their application to positioning. We have included a number of simulations and now review our main conclusions. We also give suggestions for future work.

In Section 4.3.2, we looked at the PF performance in a number of different positioning environments. We saw that the improvement of the PF over the EKF was most apparent in the scenarios with base station only measurements, and was not noticeable with the satellite only measurements. Sample sizes of about $N = 10^4$ seemed to be sufficient for reasonable filtering estimates in most of our test cases. In addition, there were cases to illustrate that we should be cautious to claim that the PF at large sample sizes, e.g. $N = 10^6$, is an adequate approximation of the optimal filter.

In fact, the estimates from smaller sample sizes $N = 10^3$ and $10^2$ may even be sufficient for some personal positioning applications. The significant problem with our implementation of the small sample PF was the numerical problem of division by zero. Some simple ad-hoc procedure to recover from these scenarios could be useful in practice.

In Section 4.3.3, we tested four different importance distributions and compared their reduction of the MC variation of the mean estimates. The improvements were slight so that, in similar positioning environments, it hardly seems worthwhile to use the more computationally expensive importance distributions. Other PF methods, such as the regularized particle filter, see e.g. Ristic et al. [2004], might be useful although they were not covered in this work.

In Section 4.4, we applied a multivariate binning technique from Roederer et al. [2001] to the comparison of PFs. The conclusions resulting from the proposed test are similar to the conclusions from an RMSE analysis of the mean estimates.
This was shown for a linear and Gaussian filtering scenario, where we have an analytical form of the marginal posterior, and also for a nonlinear and Gaussian filtering scenario, where we estimated the optimal solution with a PF.

We have not offered a detailed discussion of the practical, implementation aspects of such a test. It should be mentioned that our implementation of the test, i.e. the construction of the bins and the actual binning of the samples, used data structures similar to Kd-trees, see e.g. de Berg et al. [2000], and was computationally feasible for the cases that we considered.

The literature on the $\chi^2$-test is vast and admittedly, our treatment of the test has been brief. In this section, we point out some questionable aspects of our use of this test for comparing PFs. First, we should question the use of the $\chi^2$-test itself. We are testing whether the two samples are from the same distribution, although it was already noted in Section 3.3.1 that the methods will not produce IID samples from the true posterior due to the finite mixture approximation. In spite of this, we have still considered the test scores as a way to empirically quantify the difference between distributions.

Second, we should question how the $\chi^2$-test was actually performed. The number of bins and the construction of the bins in $\chi^2$-tests is not always straightforward and is therefore debatable. The resampling that is carried out before binning the PF samples is also questionable; due to the “extra” resampling, our test is then comparing an approximation of the PF approximation, which complicates the analysis. Finally, our use of a large sample PF to approximate the optimal posterior, as well as our choice of importance distribution for this PF, is not properly justified; other methods such as rejection sampling or MCMC might result in a better approximation of the posterior.

The intention of the distribution comparison was to devise better ways of comparing PFs and possibly other Bayesian filters. The binning procedure that was described is, of course, limited to sample-based methods. Other possible methods could include integrating the posterior distribution over a finite number of regions and then using some distance, e.g. Kullback Leibler divergence or total variation distance [Gibbs and Su 2002], on the resulting finite state-space to quantify the distance between the distributions. However, the choice of distance is then quite arbitrary and there is little reason to prefer one distance over another. This seems to be an interesting area for future work.
Bibliography


