Tracking Filters for Radar Systems

by

Wing Ip Tam

A thesis submitted in conformity with the requirements for the degree of Master of Applied Science
Graduate Department of Electrical and Computer Engineering
University of Toronto

©Copyright by Wing Ip Tam 1997
The author has granted a non-exclusive licence allowing the National Library of Canada to reproduce, loan, distribute or sell copies of this thesis in microform, paper or electronic formats.

The author retains ownership of the copyright in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author’s permission.

L’auteur a accordé une licence non exclusive permettant à la Bibliothèque nationale du Canada de reproduire, prêter, distribuer ou vendre des copies de cette thèse sous la forme de microfiche/film, de reproduction sur papier ou sur format électronique.

L’auteur conserve la propriété du droit d’auteur qui protège cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

0-612-28884-6
Abstract

In this paper we discuss the problem of target tracking in Cartesian coordinates with polar measurements and propose two efficient tracking algorithms. The first algorithm uses the multidimensional Gauss-Hermite quadrature to evaluate the optimal estimate as a weighted sum of functional values. To reduce the computational requirements of this quadrature technique we have suggested several ways to reduce the dimension, the number and the order of the quadratures required for a given accuracy. The second tracking algorithm is based on the Gaussian sum filter. To alleviate the computational burden associated with the Gaussian sum filter, we have found two efficient and systematic ways to approximate a non-Gaussian and measurement-dependent function by a weighted sum of Gaussian density function and we have derived the formula for updating the parameters involved in the bank of Kalman-type filters and we also have proposed new techniques to control the number of terms in the Gaussian mixture at each iteration. Simulation results show that these two proposed methods are more accurate than the classical method, such as the extended Kalman filter (EKF).
Acknowledgements

I am especially grateful for the extensive and generous assistance of my superior, Professor Dimitrios Hatzinakos, who spent time with me each week to guide me through the thesis and made numerous helpful suggestions to me.

I wish to thank Kostas Plataniotis for his thorough reviews and perceptive comments which greatly improve the quality of my thesis work. I also wish to thank Professor Pasupathy and Professor Kwong for their consultations.

I thank my friends William Ma, Erwin Pang, Jacky Yan, Richard Lee, Wilson Wong and Ryan Lee and his family for all the fun time.

Lastly, much credit must be given to the members of my family, who have supported me and have borne with me during this period.
# Contents

1 Introduction 1

2 Problem Definition and Literature Survey 4

2.1 Basic Model for Radar Systems 4

2.1.1 Target Dynamics 5

2.1.2 Radar Measurement 7

2.2 The Bayesian Approach 8

2.3 Sub-Optimal Nonlinear Filters for Radar Tracking 12

3 An Efficient Radar Tracking Algorithm Using Multidimensional Gaussian-Hermite Quadratures 18

3.1 Introduction 18

3.2 Gauss-Hermite Quadrature 19

3.3 Basic Principles of the Proposed Filter 21

3.4 Complexity Analysis 24

3.5 Filter Structure 25

3.6 Simulation Results 26

4 Adaptive Gaussian Sum Algorithm for Radar Tracking 29

4.1 Introduction 29

4.2 Theoretical Foundations 30

4.3 Basic Principles of the Proposed Filter 30

4.3.1 Approximation of Densities 31
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.3.2</td>
<td>Bank of Kalman Filters</td>
<td>40</td>
</tr>
<tr>
<td>4.3.3</td>
<td>Growing Memory Problem</td>
<td>41</td>
</tr>
<tr>
<td>4.4</td>
<td>Filter Structure</td>
<td>45</td>
</tr>
<tr>
<td>4.5</td>
<td>Simulation Results</td>
<td>46</td>
</tr>
<tr>
<td>5</td>
<td>Conclusion</td>
<td>54</td>
</tr>
<tr>
<td>A</td>
<td>Discretization of the Continuous-Time Equation of Motion</td>
<td>57</td>
</tr>
<tr>
<td>B</td>
<td>Track Initialization</td>
<td>59</td>
</tr>
<tr>
<td>C</td>
<td>Detail Derivations of the Curve-fitting Approach</td>
<td>61</td>
</tr>
<tr>
<td>D</td>
<td>Detail Derivation of the Transformation Approach</td>
<td>63</td>
</tr>
<tr>
<td>E</td>
<td>Reproducing Property of Gaussian Densities</td>
<td>65</td>
</tr>
<tr>
<td>F</td>
<td>Bank of Kalman Filters</td>
<td>68</td>
</tr>
</tbody>
</table>
List of Figures

2.1 Flow Diagram for the Bayesian Estimator ........................................... 9

3.1 With the factorization the order of the quadrature is reduced for a
given accuracy. ......................................................... 23

3.2 Flow Diagram of the Proposed Filter ............................................. 26

3.3 Comparison of the position errors .................................................. 28

3.4 Comparison of the velocity errors .................................................. 28

4.1 Plots for the curve-fitting approach ................................................. 33

4.2 Basic Principles of the Proposed Gaussian mixture Approximation
Method ................................................................. 34

4.3 Fundamentals of the proposed Gaussian sum approximation method .... 37

4.4 A. The function $p(z_n|x_n)$ can be obtained through a nonlinear transfor-
mation from the density $p(w_n)$. B. Similarly, some initial parameters
from the statistics of the noise $w_n$ can be transformed into a Gaussian
sum approximation for the function $p(z_n|x_n)$ ............................... 38

4.5 Plots of the original function $p(z_n|x_n)$ and the approximation .......... 39

4.6 Mixture before Reduction (50 components) ...................................... 44

4.7 Mixture after Reduction (10 components) ........................................ 44

4.8 Flow Diagram of the Proposed Gaussian Sum Filter .......................... 45

4.9 Comparison of the position errors for measurement noise level 1 ......... 50

4.10 Comparison of the velocity errors for measurement noise level 1 ........ 50

4.11 Comparison of the position errors for measurement noise level 2 ........ 51

4.12 Comparison of the velocity errors for measurement noise level 2 ........ 51
<table>
<thead>
<tr>
<th>Section</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.13 Superposition of 100 realizations of experiment based on Extended Kalman Filter</td>
</tr>
<tr>
<td>4.14 Superposition of 100 realizations of experiment based on Extended Kalman Filter</td>
</tr>
<tr>
<td>4.15 Comparison of the position errors for 3-D tracking scenario</td>
</tr>
<tr>
<td>4.16 Comparison of the velocity errors for 3-D tracking scenario</td>
</tr>
<tr>
<td>C.1 Weighted Gaussian Function</td>
</tr>
<tr>
<td>C.2 Affine Transformation of the ellipsoidal base</td>
</tr>
</tbody>
</table>
List of Tables

4.1 Comparison of the accuracy and the efficiency of the proposed Gaussian sum approximation method and the classical method. Note that the classical method is based on the Marquardt algorithm and the number of Gaussian terms $N$ is fixed to be 5 for both methods. 36

4.2 Comparison of the accuracy and the efficiency of the proposed Gaussian sum approximation method and the classical method. Note that the classical method is based on the Marquardt algorithm and the number of Gaussian terms $N$ is fixed to be 5 for both methods. 40
Chapter 1

Introduction

For nearly three decades the target tracking-trajectory estimation problem has been a fruitful applications area for state estimation. It has found wide applications in both military and commercial areas [1, 2] such as inertial navigation, guidance & control; global positioning system (GPS); differential global positioning system (DGPS); wide area augmentation system (WAAS); inertial navigation system (INS); missiles guidance system; satellite orbit determination; maritime surveillance; air traffic control; freeway traffic system; fire control system; automobile navigation system; fleet management; underwater target tracking system. Many problems have been solved, yet new and diversified applications still challenge engineers.

This paper addresses the problem of target tracking in Cartesian coordinates with polar measurements. In tracking applications the target motion is usually best modeled in a simple fashion using Cartesian coordinates. Unfortunately, in most radar systems the target position measurements are provided in polar coordinates (range and azimuth) with respect to the sensor location. Tracking in Cartesian coordinates using polar measurements is a problem of nonlinear estimation. A rigorous treatment of the nonlinear estimation problem requires the use of stochastic integrals and stochastic differential equations [3]. In this paper, we will adapt the formal manipulation of the white noise process and omit the rigorous derivations using Ito calculus. The nonlinear estimation problem is very challenging because the distribution of the
state is generally non-Gaussian. Without the Gaussian property, efficient computation of the conditional mean is very difficult because the optimal (conditional mean) nonlinear estimator cannot be realized with a finite-dimensional implementation; consequently all practical nonlinear filters are suboptimal. These suboptimal nonlinear filters can be divided into two categories in general: the most popular technique uses a Taylor-series expansion to approximate the nonlinear system model, e.g. the extended Kalman filter (EKF); the other approximates the conditional probability density function in such a fashion that makes the computation of the conditional mean efficient, e.g. the Gaussian sum filter. Both approaches yield computationally feasible algorithms; however, each has shortcoming that must be addressed. The first approach is efficient but not accurate and often results in filter divergence whereas the second approach yields very accurate result but the complexity is so high that precludes practical real-time applications.

To improve the performance of the existing approaches, we propose two suboptimal tracking algorithms which are believed to be more accurate and efficient than the most widely used Taylor series methods, e.g. the EKF. The first method uses the multidimensional Gauss-Hermite quadrature to evaluate the optimal (conditional mean) estimate of the target states directly from the Bayesian equations. This quadrature technique replaces the integrals in the Bayesian equations with a weighted sum of samples of the integrand and this approximation can be very accurate if the integrand is smooth. However, this method has a major drawback, namely its computational complexity because of the dimension of the problem and the number and the order of the quadratures involved the number of samples required are usually too large to be handled in real-time. In this paper we have suggested several ways to reduce the computational requirements of this quadrature technique by reducing the dimension, the number and the order of the quadratures required for a given accuracy. The second tracking algorithm is based on the Gaussian sum filter. The Gaussian sum approximation method can approximate any probability density function as closely as desired [4]. Moreover, the optimal estimate (conditional mean) can be computed in a
simple manner as a weighted sum of the estimates from a bank of Kalman-type filters operating in parallel. However, this method has a very high complexity since the number of Gaussian terms required in the approximation increases exponentially in time. To alleviate the computational burden associated with the Gaussian sum filter, we have found two efficient and systematic ways to approximate a non-Gaussian and measurement-dependent function by a weighted sum of Gaussian density function and we have derived the formula for updating the parameters involved in the bank of Kalman-type filters and we also have suggested several ways to control the number of terms in the Gaussian mixture at each iteration. Simulation results show that these two proposed methods are more accurate than the classical method, such as the EKF.

The rest of the paper is organized as follows: In chapter 2, mathematical models for target tracking applications and the existing tracking algorithms are discussed. In chapter 3, the tracking algorithm based on the multidimensional Gauss-Hermite quadrature is introduced. Motivation and implementation issues for this technique are discussed and simulation results are presented also in this chapter. In chapter 4, the tracking algorithm based on the Gaussian sum filter is discussed. Modified Gaussian sum approximation method is analyzed and several mixture reduction algorithms are presented in this chapter. Simulation results of this technique and the performance of this proposed tracking filter is described. Finally, chapter 5 summarizes our conclusions and future works.
Chapter 2

Problem Definition and Literature Survey

2.1 Basic Model for Radar Systems

The tracking problem is a state estimation problem, i.e. assuming the state of a target evolves in continuous time according to the equation

\[ \dot{x}(t) = f(x(t)) + v(t) \]  (2.1)

The state vector \( x(t) \) usually contains target position, velocity and sometimes acceleration as state variables and its corresponding discrete measurement vector is given by

\[ z_n = h(x_n) + w_n \]  (2.2)

where \( v(t) \) and \( w_n \) are system and measurement noise processes respectively and they are assumed to be zero mean white noise processes. The covariance of the system noise, \( Q \), is selected to compensate for modelling errors (discrepancies between the model and the actual process). The covariance of the measurement noise process sequence, \( R_n \), should also be chosen to represent all possible excursions such as measurement biases, false measurements, etc.
2.1.1 Target Dynamics

Constant Velocity (CV) Model For some targets such as airplanes, a constant velocity (CV) model is sufficient, i.e. the state vector contains six variables

\[
\begin{align*}
x(t) &= \begin{bmatrix} x_1(t), x_2(t), x_3(t), x_4(t), x_5(t), x_6(t) \end{bmatrix} \\
&= \begin{bmatrix} x(t), \dot{x}(t), y(t), \dot{y}(t), z(t), \dot{z}(t) \end{bmatrix}^T
\end{align*}
\]

(2.3)

where \((x(t), y(t), z(t))\) are coordinates of a Cartesian coordinates system used to describe the target dynamics. The state equations are

\[
\begin{align*}
\dot{x}_i(t) &= x_{i+1}(t) \\
\dot{x}_{i+1}(t) &= v_i(t)
\end{align*}
\]

(2.4)

where \(v_i(t)\) is the system noise terms used to characterize modelling errors.

Constant Accelerating (CA) Model If the target being tracked is maneuvering (accelerating), a constant accelerating (CA) model is sometimes used, i.e.,

\[
\begin{align*}
x(t) &= \begin{bmatrix} x_1(t), x_2(t), x_3(t), x_4(t), x_5(t), x_6(t), x_7(t), x_8(t), x_9(t) \end{bmatrix} \\
&= \begin{bmatrix} x(t), \dot{x}(t), \ddot{x}(t), y(t), \dot{y}(t), \ddot{y}(t), z(t), \dot{z}(t), \ddot{z}(t) \end{bmatrix}^T
\end{align*}
\]

(2.5)

The state equations can be written accordingly, i.e.,

\[
\begin{align*}
\dot{x}_i(t) &= x_{i+1}(t) \\
\dot{x}_{i+1}(t) &= x_{i+2}(t) \\
\dot{x}_{i+2}(t) &= v_i(t)
\end{align*}
\]

(2.6)

These two state equations are also referred to as the first- and second-order polynomial dynamics respectively.

Target with Sudden Maneuvers Targets with sudden maneuvers can be modelled as systems with abrupt changes [5]. We can modify equation (2.1) to become two sets of equations, one representing the premaneuver dynamics and the other incorporating the maneuver feature

\[
\dot{x}(t) = f(x(t), x_m(t)) + v(t)
\]

(2.7)
where $x_m(t)$ is the vector representing maneuvering force and satisfies

$$
\begin{cases}
  x_m(t) = 0 & \text{for } t < t_m \\
  \dot{x}_m(t) = f_m(x_m(t)) + v_m(t) & \text{for } t \geq t_m
\end{cases}
$$

(2.8)

where $t_m$ is the time the maneuver begins. $f_m(\cdot)$ is the maneuvering dynamics, and $v_m(t)$ is the system noise for the maneuvering dynamics $f_m(\cdot)$. For targets with sudden maneuvers, $t_m$ is unknown, $f_m(\cdot)$ and $v_m(t)$ may be unknown or partially known.

The target maneuver $x_m(t)$ is correlated in time; namely, if a target is accelerating at time $t$, it is likely to be accelerating at time $t + \tau$ for sufficiently small $\tau$. For example, a lazy turn will often give rise to correlated acceleration inputs for up to one minute, evasive maneuvers will provide correlated acceleration inputs for periods between ten and thirty seconds, and atmospheric turbulence may provide correlated acceleration inputs for one or two seconds. A typical representative model often used in airplane tracking is

$$
\dot{x}_m(t) = -\alpha x_m(t) - v_\alpha(t)
$$

(2.9)

where $\alpha$ is the correlation constant and $v_\alpha(t)$ is a noise process. Methods for selecting values for $\alpha$ and statistics for $v_\alpha(t)$ can be found in [6].

**Discrete Time Equations of Motion** The above target equations of motion must be first discretized [7] into appropriate discrete time equations in order for the application of suitable digital filters [6]. In this paper, we assume the motion of the target (such as aircrafts, and missiles) generally follows a straight-line constant-velocity (CV) trajectory. Turns, evasive maneuvers, and acceleration due to turbulence of the surrounding environment may be modeled as perturbations upon the constant-velocity trajectory. Assume the target is moving in a two-dimensional plane, i.e. the state vector $x_n$ contains four variables

$$
x_n = [x_n, u_{x,n}, y_n, u_{y,n}]^T
$$

(2.10)

1The discretization of continuous-time system is presented in Appendix A
where \((x_n, y_n)\) are the target positions and \((v_{x,n}, v_{y,n})\) are the target velocities at time step \(n\). The discrete time target equations of motions are given as

\[
\begin{bmatrix}
  x_{n+1} \\
v_{x,n+1} \\
y_{n+1} \\
v_{y,n+1}
\end{bmatrix} =
\begin{bmatrix}
  1 & \Delta T & 0 & 0 \\
  0 & 1 & 0 & 0 \\
  0 & 0 & 1 & \Delta T \\
  0 & 0 & 0 & 1
\end{bmatrix}
\begin{bmatrix}
  x_n \\
v_{x,n} \\
y_n \\
v_{y,n}
\end{bmatrix} +
\begin{bmatrix}
  \Delta T^2/2 & 0 \\
  \Delta T & 0 \\
  0 & \Delta T^2/2 \\
  0 & \Delta T
\end{bmatrix}
\begin{bmatrix}
a_{x,n} \\
a_{y,n}
\end{bmatrix}
\] (2.11)

where the target accelerations \(a_{x,n}\) and \(a_{y,n}\) are modeled as stationary Gaussian distributed random processes with zero mean and variance \(\sigma_a^2\). It is assumed that the accelerations in \(x\) and \(y\) directions are mutually statistically independent. \(\Delta T\) is the radar scan interval.

In matrix notation

\[
x_{n+1} = Fx_n + Gv_n
\] (2.12)

where \(v_n\) is the noise process sequence \([a_{x,n} a_{y,n}]^T\) with zero mean and covariance matrix \(Q_n\) (which equals \(\sigma_a^2 I\)). \(F\) is called the transition matrix and \(G\) the noise gain matrix.

### 2.1.2 Radar Measurement

When a radar is used to measure the position of a target moving in the two-dimensional space, the measurements are reported in range and bearing which are in polar coordinates. The polar coordinate measurement of the target position related to the Cartesian coordinate target state is denoted as

\[
\begin{bmatrix}
r_n \\
\theta_n
\end{bmatrix} =
\begin{bmatrix}
\sqrt{x_n^2 + y_n^2} \\
\tan^{-1}\frac{y_n}{x_n}
\end{bmatrix} +
\begin{bmatrix}
w_{r,n} \\
w_{\theta,n}
\end{bmatrix}
\] (2.13)

where \(r_n\) and \(\theta_n\) are the measured range and bearing of the moving target and \(w_{r,n}\) and \(w_{\theta,n}\) are the measurement noise of the range and the bearing; they are assumed to be zero mean white Gaussian processes with variances \(\sigma_r^2\) and \(\sigma_\theta^2\) respectively. Moreover, the range and the bearing of the measurement are assumed to be uncorrelated.
In matrix notation:

\[ z_n = h(x_n) + w_n \]  \hspace{1cm} (2.14)

where \( z_n \) is the vector of polar coordinates measurement \([r_n, \theta_n]^T\). \( h(\cdot) \) is the Cartesian-to-polar coordinate transformation. \( w_n \) is the observation noise process \([w_{r,n}, w_{\theta,n}]^T\) which is assumed to be zero-mean white Gaussian noise process with covariance matrix \( R_n \) (which equals \( \begin{bmatrix} \sigma_r^2 & 0 \\ 0 & \sigma_{\theta}^2 \end{bmatrix} \)).

The basic problem is to estimate the current state \( x_n \) based on the sequence of observations \( Z^n = \{z_i\}_{i=1}^n \), the statistics of the noise input and the measurement and system models. Generally, one choose the optimal estimate by extremizing some performance criterion such as the mean-square error. Regardless of the performance criterion, given the a posteriori density function \( p(x_n|Z^n) \) any type of estimate can be determined. Thus, the estimation problem can be approached as the problem of determining the a posteriori density. This approach is generally referred to as the Bayesian approach.

2.2 The Bayesian Approach

In this paper, we will choose the optimal estimate based on the minimum mean squared error (MMSE) criterion. In this case, the optimal estimate \( \hat{x}_{\text{opt},n|n} \) at time \( n \) is just the mean of the state density function conditioned on the measurement history \( Z^n \).

\[ \hat{x}_{\text{opt},n|n} = E[x_n|Z^n] = \int x_n p(x_n|Z^n) dx_n \]  \hspace{1cm} (2.15)

This requires the conditional density \( p(x_n|Z^n) \) be known at each iteration. From the state equation 2.12, the state \( x_{n+1} \) depends only on the previous state \( x_n \) and a white noise sequence \( v_n \); it is called an incompletely observed Markov process [8], which inherits the important property of the Markov process, namely

\[ p(x_{n+1}|x_n, \ldots, x_0) = p(x_{n+1}|x_n) \]  \hspace{1cm} (2.16)
Similarly, because the measurement $z_n$ depends on the state $x_n$ and a white observation noise sequence $w_n$,

$$p(z_n | x_n, Z^{n-1}) = p(z_n | x_n)$$  \hspace{1cm} (2.17)

Utilizing the above property the \textit{a posteriori} density $p(x_n | Z^n)$ can be determined recursively by the following Bayesian equations in two stages: prediction and update [9].

![Flow Diagram for the Bayesian Estimator](image)

**Figure 2.1: Flow Diagram for the Bayesian Estimator**

**Prediction:** Suppose that the conditional density $p(x_{n-1} | Z^{n-1})$ at time $n - 1$ is known. Then using the state equation (2.12) it is possible to obtain the conditional density of the state $p(x_n | Z^{n-1})$ at time $n$

$$p(x_n | Z^{n-1}) = \int p(x_n | x_{n-1})p(x_{n-1} | Z^{n-1})dx_{n-1}$$  \hspace{1cm} (2.18)

This equation is called Chapman-Kolmogorov equation.

The probabilistic model of the state evolution, $p(x_n | x_{n+1})$ can be defined by the state equation and the known statistics of $v_{n-1}$

$$p(x_n | x_{n-1}) = \int p(x_n | x_{n-1}, v_{n-1})p(v_{n-1} | x_{n-1})dv_{n-1}$$  \hspace{1cm} (2.19)
By assumption \( p(v_{n-1}|x_{n-1}) = p(v_{n-1}) \) we have

\[
p(x_n|x_{n+1}) = \int \delta(x_n - f(x_{n-1}) - v_{n-1}) p(v_{n-1}) dv_{n-1}
\]

\[
= p_n(x_n - f(x_{n-1}))
\]  

(2.20)

where \( \delta(\cdot) \) is the Dirac delta function. The delta function is used because if \( x_{n-1} \) and \( v_{n-1} \) are known, then \( x_n \) can be obtained deterministically. Since the noise sequence \( v_n \) is assumed to be an zero-mean white Gaussian random process with covariance matrix \( Q_n \) and the system dynamics \( f(\cdot) \) is linear; therefore the density \( p(x_n|x_{n-1}) \) is also white Gaussian of the form

\[
p(x_n|x_{n-1}) = \frac{1}{2\pi|Q_n|^{1/2}} e^{-\frac{1}{2}(x_n - Fx_{n-1})^T Q_n^{-1} (x_n - Fx_{n-1})}
\]

(2.21)

Update: The conditional density \( p(x_n|Z^n) \) can be written using Bayes' formula

\[
p(x_n|Z^n) = \frac{p(z_n|x_n)p(x_n|Z^{n-1})}{p(z_n|Z^{n-1})}
\]

(2.22)

where the normalizing constant is given by

\[
p(z_n|Z^{n-1}) = \int p(z_n|x_n)p(x_n|Z^{n-1}) dx_n
\]

(2.23)

The conditional density of \( z_n \) given \( x_n \), \( p(z_n|x_n) \) is defined by the measurement model (2.14) and the known statistics of \( w_n \)

\[
p(z_n|x_n) = \int \delta(z_n - h_n(x_n) - w_n) p(w_n) dw_n
\]

\[
= p_w(z_n - h(x_n))
\]

(2.24)

The observation noise sequence \( w_n \) is white Gaussian and the measurement equation is nonlinear; This density becomes

\[
p(z_n|x_n) = \frac{1}{2\pi|R_n|^{1/2}} e^{-\frac{1}{2}(z_n - h(x_n))^T R_n^{-1} (z_n - h(x_n))}
\]

(2.25)

The initial conditional density \( p(x_0|Z^0) \) is given by

\[
p(x_0|Z^0) = p(x_0|z_0) = \frac{p(x_0)p(z_0|x_0)}{p(z_0)}
\]

(2.26)
where $p(x_0)$ is usually assumed to be white Gaussian.

This optimal (conditional mean) nonlinear filter cannot be realized with a finite-dimensional implementation. It is because the distribution of the state is generally non-Gaussian; thus, the integration indicated in equations (2.18) and (2.23) is generally impossible to accomplish in closed form except when the measurement equation is linear (i.e. the measurement function $h(x_n)$ is linear) and the statistics of the initial state and the noise sequences are all white Gaussian. In that case, the equations (2.18) and (2.22) can be evaluated in closed form and the conditional density $p(x_n|Z^n)$ is reduced to a Gaussian density function. The propagation of its mean and covariance matrix is collectively known as the Kalman filter equations.

**The Kalman Filter** Assume the measurement equation is linear

$$z_n = H_n \cdot x_n + w_n \tag{2.27}$$

where $H_n$ is the measurement matrix and the noise process $w_n$ is modeled as a zero-mean white Gaussian random process with covariance matrix $R_n$.

The linear-Gaussian assumption of the model leads to the preservation of the Gaussian property of the conditional density $p(x_n|Z^n)$. Thus, they are completely characterized by their means and their covariance matrices, which can be obtained recursively in two stages: prediction and update [10].

**Prediction**

$$\hat{x}_{n|n-1} = F_{n-1} \cdot \hat{x}_{n-1|n-1} \tag{2.28}$$

$$P_{n|n-1} = F_{n-1} P_{n-1|n-1} F_{n-1}^T + G_{n-1} Q_{n-1} G_{n-1}^T \tag{2.29}$$

**Update**

---

2The track is initialized from the first two measurements using the algorithm presented in Appendix B
\[ \dot{x}_{n|n} = \dot{x}_{n|n-1} + K_n(z_n - H_n \dot{x}_{n|n-1}) \]  
(2.30)

\[ P_{n|n} = P_{n|n-1} - K_n H_n P_{n|n-1} \]  
(2.31)

where

\[ K_n = P_{n|n-1} H_n^T [H_n P_{n|n-1} H_n^T + R_n]^{-1} \]  
(2.32)

is called the Kalman gain.

Most of the sub-optimal nonlinear filters are based on the linear Kalman filter equations by transforming the nonlinear measurement equation into a linear equation with white additive Gaussian noise, i.e. forcing the requirements of the Kalman filter equations satisfied. The next section presents several sub-optimal filters used in target tracking in Cartesian coordinates with noisy polar measurements.

### 2.3 Sub-Optimal Nonlinear Filters for Radar Tracking

The most widely used tracking filter is the extended Kalman filter (EKF) \[10, 8\] which employs the first-order Taylor series approximation to adapt the linear Kalman filter to the nonlinear system described by equations (2.12) and (2.14). Since the state is in Cartesian coordinates and the measurements are in polar coordinates. Therefore, there is a nonlinear measurement function \( h(x_n) \) and linearization of the measurement equation is required for the state and covariance update. The prediction stage of the EKF proceeds as in equations (2.28) and (2.29) but the update stage is described as follows:

\[ \dot{x}_{n|n} = \dot{x}_{n|n-1} + K_n(z_n - h(\dot{x}_{n|n-1})) \]  
(2.33)

\[ P_{n|n} = P_{n|n-1} - K_n J_h(\dot{x}_{n|n-1}) P_{n|n-1} \]  
(2.34)

where

\[ K_n = P_{n|n-1} J_h(\dot{x}_{n|n-1})^T [J_h(\dot{x}_{n|n-1}) P_{n|n-1} J_h(\dot{x}_{n|n-1})^T + R_n]^{-1} \]  
(2.35)
is the Kalman gain and
\[
J_h(\mathbf{x}_{n|n-1}) = \left[ \frac{\partial \mathbf{h}(\mathbf{x}_n)}{\partial \mathbf{x}_n} \right]_{\mathbf{x}_n=\mathbf{x}_{n|n-1}} \tag{2.36}
\]
is the Jacobian of the nonlinear function \(\mathbf{h}(\mathbf{x}_n)\). Its accuracy however depends heavily on the stability of the Jacobian matrix. In practice, due to the modelling error the Jacobian matrix is often numerically unstable resulting in filter divergence. This Taylor series expansion method can be extended further to include higher order terms in the expansion for the nonlinear function \(\mathbf{h}(\mathbf{x}_n)\) to seek better approximation, such as the second-order filter [9]. It is however questionable whether higher-order approximations would improve performance in cases where extended Kalman filter (EKF) diverges [9]. The crucial defect in the approximations is that replace global properties of a function by its local properties (its derivatives). Clearly, if the true state lies outside the region in which the nonlinear system is accurately represented by its Taylor series, not only will the first-order approximation be inadequate but so will any higher-order approximation.

There exists similar approaches to the EKF which utilizes mixed coordinate filters, i.e. the state is in Cartesian coordinates and the measurements are in polar coordinate. They all share the same filter structure; they are only different in the way their filter gains \(K_n\) are computed. One technique is the iterated extended Kalman filter [10]. It improves the accuracy of the extended Kalman filter (EKF) by repeatedly updating the estimate \(\mathbf{x}_{n|n}\), and the kalman gain \(K_n\) based on first-order Taylor series expansion about the most recent estimate. Denote the \(i^{th}\) estimate of \(\mathbf{x}_{n|n}\) by \(\hat{\mathbf{x}}_{n|i}^{(i)}\), \(i = 0, 1, \ldots, N\) and \(\hat{\mathbf{x}}_{n|n,0} = \hat{\mathbf{x}}_{n|n-1}\). The following equations summarize the iterated extended Kalman filter (IKF) measurement update:

\[
\hat{\mathbf{x}}_{n|i} = \mathbf{x}_{n|i-1} + \mathbf{K}_{n|i}(\mathbf{z}_n - \mathbf{h}(\hat{\mathbf{x}}_{n|i-1}) - J_h(\hat{\mathbf{x}}_{n|i})(\hat{\mathbf{x}}_{n|i-1} - \hat{\mathbf{x}}_{n|i-1})) \tag{2.37}
\]
\[
\mathbf{K}_{n|i} = \mathbf{P}_{n|i-1}J_h(\hat{\mathbf{x}}_{n|i-1})^T[J_h(\hat{\mathbf{x}}_{n|i-1})\mathbf{P}_{n|i-1}J_h(\hat{\mathbf{x}}_{n|i-1})^T + \mathbf{R}_n]^{-1} \tag{2.38}
\]
\[
\hat{\mathbf{x}}_{n|n} = \hat{\mathbf{x}}_{n|i} \tag{2.39}
\]
\[
\mathbf{P}_{n|n} = \mathbf{P}_{n|i} = (\mathbf{I} - \mathbf{K}_{n|i}\mathbf{J}_h(\hat{\mathbf{x}}_{n|i}))\mathbf{P}_{n|i-1} \tag{2.40}
\]

13
Similar to the extended Kalman filter (EKF). Its accuracy also depends heavily on the stability of the Jacobian matrix. If the Jacobian matrix is numerically unstable, it may result in slowed or even non-convergence of the estimates. A new nonlinear iterated filter based on the Julier et al.'s discrete approximation and the Levenberg-Marquardt algorithm [11] has been shown to be more accurate and robust than the EKF and the IKF. There is also a "quasi-extended" Kalman filter [12], which shows improvements when tracking maneuvering targets at close range. There is another technique, modified gain extended Kalman filter (MGEKF) [13], has been shown to guarantee stability and exponential convergence for a special class of nonlinearities.

Another technique uses the statistical linearization as opposed to the Taylor series expansion [10] to approximate the nonlinear function \( h(x_n) \) such that the mean square error between the linear approximation and the nonlinear function is minimized. The computational requirements of this method is greater than that of the extended Kalman filter; however, the performance advantages offered by this method may make the additional computation worthwhile. There exists similar techniques which are based on various series expansions, such as Edgeworth and Gram-Charlier expansion. In some cases, because of the computational constraints, it may be impractical to compute the filter gain in real time. Under such conditions one must use either a set of precomputed filter gains or a constant gain filter. One popular constant gain filter is \( \alpha - \beta - \gamma \) filter (or \( \alpha - \beta \) filter when using a CV model) [14, 15].

The extended Kalman filter (EKF) uses the polar coordinates measurement. There is an alternative approach which transforms the polar coordinates measurements to the Cartesian coordinate systems. Then the conventional Kalman filter is applied. This approach is generally called the converted measurement Kalman filter (CMKF). With the converted measurement Kalman filter [16], the polar coordinate measurement \( z_n^p \) is first converted to the Cartesian coordinate measurements \( z_n^c \) using the inverse transformation \( h^{-1}(z_n^p) \). The original noise process \( w_n \) acting on the converted measurement \( z_n^c \) no longer behaves rigorously as an additive term, but in some com-
licated fashion. However, at least when the covariance of the noise $w_n$ is small, the new Cartesian coordinate measurement equation can be written as follows:

$$z_n^c = D x_n + \tilde{w}_n$$  \hspace{1cm} (2.41)

where $D = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$ and $\tilde{w}_n$ is approximated as a white Gaussian noise process on the converted measurement $z_n^c$ with zero mean and covariance matrix $M_n$

$$M_n = E[\tilde{w}_n \tilde{w}_n^T] = J_{h^{-1}}(\hat{z}_{n|n-1}) R_n J_{h^{-1}}(\hat{z}_{n|n-1})^T$$  \hspace{1cm} (2.42)

where

$$J_{h^{-1}}(\hat{z}_{n|n-1}) = \frac{\partial h^{-1}(z_n^p)}{\partial z_n^p} \bigg|_{z_n^p = \hat{z}_{n|n-1}}$$  \hspace{1cm} (2.43)

As a result, the new measurement equation in Cartesian coordinates becomes linear and the noise process is Gaussian, the standard Kalman filter can be applied to the problem with the measurement equation defined in Equation (2.41) and the statistics of the measurement noise given as in Equation (2.42). This method however is an acceptable approximation only for moderate cross-range errors (cross-range error is defined as the product of the range and the bearing angle). An improved method using debiased converted measurements [17] which accounts for the sensor inaccuracies over all practical geometries and accuracies and provides consistent estimates (i.e. compatible with the filter calculated covariances) for all practical situations. There is another technique [2] which preprocesses the transformed measurements to reduce measurement uncertainty and increase state estimation accuracy by explicitly employing the knowledge available about target motion and trajectory.

So far we have discussed various algorithms based on first-order approximation techniques. For the rest of the section, we will present several well-known nonlinear algorithms based on global approximation techniques. These nonlinear algorithms are not commonly used for real-time applications, because the computation time and data storage requirements are too excessive. One global approach is to approximate the
densities directly in such a manner that makes the integrations involved in Bayesian equations (2.22) and (2.18) as trackable as possible. The simplest method is the point-mass method [18], where the densities are approximated by point masses located on a rectangular grid. As a result, the integrations in the Bayesian equations can be accomplished numerically as a discrete nonlinear convolution which requires \(O(N^2)\) operations to compute, where \(N\) is the number of grid points. An improved method approximates the densities with a piecewise constant function [19] and this approximation is slightly more sophisticated than the point mass method, but it converts the Bayesian equations to a discrete linear convolution which requires only \(O(N \log N)\) operations. There is another technique which uses a Gaussian-sum approximation for the density [4]. In this case, the Bayesian equations become a bank of Kalman filter equations. However, the number of components in the approximating sum grows exponentially with time. More sophisticated interpolation schemes such as different spline approximations [20, 21, 22, 23] which require fewer grid points for a given accuracy are also studied. There exists alternative technique, called the bootstrap filter [24], which represents and recursively propagates the required density function as a set of random samples, rather than a function over state space. As the number of samples becomes very large, they effectively provide an exact, equivalent, representation of the required density function. However, the number of samples required to give a satisfactory representation of the densities for the filter operation is very difficult to determine. Another global approach is to approximate the integrations involved in the Bayesian equations directly. All such methods amount to replacing the integral with a weighted sum of samples of the integrand based on some particular quadrature formula [25, 26, 27]. The basic issue again is trading off computational complexity against the number of grid points required to achieve a desired accuracy.

Fuzzy logic and neural network have been considered extensively for target tracking recently [28, 29]. An fuzzy-Kalman estimator [30] which uses a fuzzy system to estimate the current target maneuvering acceleration and a Kalman filter to estimate the target position is studied. The resulting estimator would keep the features of the
Kalman filter but with improved accuracy if good estimates of target maneuvering acceleration can be produced. Similarly, neurofuzzy estimators [31] are used to initialize the Kalman filter and extended Kalman filter. It is shown that filters which have been initialized with neurofuzzy estimates converge faster and give improved performance. Fuzzy logic is also introduced into a conventional $\alpha - \beta$ constant gain filter [32] in an attempt to improve its tracking ability for maneuvering targets.
Chapter 3

An Efficient Radar Tracking Algorithm Using Multidimensional Gauss-Hermite Quadratures

3.1 Introduction

In this chapter, a new tracking algorithm based on multidimensional Gauss-Hermite quadrature is presented. This quadrature technique evaluates the optimal (conditional mean) estimate by replacing the integrals involved in the Bayesian equations with a weighted sum of samples of the integrand and this approximation can be very accurate if the integrand is smooth. However, this technique requires excessive computational time and data storage, because of the dimension, the number and the order of the quadrature involved the number of samples required are usually too large to be handled in real-time. In this chapter, we suggest several ways to reduce the computational requirements of this quadrature technique by reducing the dimension, the number and the order of the quadratures required for a given accuracy.
3.2 Gauss-Hermite Quadrature

Based upon the orthogonality property of the Hermite polynomials, one can develop the Gauss-Hermite quadrature [33].

$$\int_{-\infty}^{\infty} e^{-u^2} f(u) du = \sum_{i=1}^{k} B_i f(u_i)$$  \hspace{1cm} (3.1)

where $B_i$ are the weights and $u_i$ are the roots of the Hermite polynomial of degree $k$ and they are chosen so that the sum of the $k$ appropriately weighted functional values yields the integral exactly when $f(u)$ is a polynomial of degree $k$ or less.

The above one-dimensional quadrature formula can be easily extended to the more general multi-dimensional quadrature formula [34]. Consider the evaluation of the following multidimensional integral

$$I = \int_{\mathbb{R}^n} f(r) e^{-(r-\hat{r})^T P^{-1}(r-\hat{r})} dr$$  \hspace{1cm} (3.2)

where $r$ and $\hat{r}$ are $n$-dimensional vectors and $P$ is a $n \times n$ nonsingular matrix. If $P$ is a positive diagonal matrix, then the multidimensional quadrature formulae by Stroud [34] may be applied, which is obtained by successively applying the one dimensional Gauss-Hermite formula. If $P$ is not diagonal, an affine transformation of variables can make $P$ diagonal. One approach is to find the eigenvector decomposition of the matrix $P$:

$$P = VDV^T$$  \hspace{1cm} (3.3)

where $D$ is a diagonal matrix of eigenvalues and $V$ is a full matrix whose columns are the corresponding eigenvectors. The expression

$$u = \frac{V}{\sqrt{D}}(r - \hat{r})$$  \hspace{1cm} (3.4)

achieves the diagonalization. A simpler approach is to find the square-root of $P$, $W$ using the Cholesky algorithm [35] such that

$$P = WW^T$$  \hspace{1cm} (3.5)
and the expression

\[ u = W^{-1}(r - \hat{r}) \]  \hspace{1cm} (3.6)

also achieves diagonalization. Then the multidimensional integral in equation (3.2) can be evaluated numerically by applying the one dimensional Gauss-Hermite quadrature formula to one variable at a time and it becomes

\[ I = \sum_{i_1=1}^{k} \sum_{i_N=1}^{k} B_{i_1, \ldots, i_N} f(r_{i_1, \ldots, i_N}) \]  \hspace{1cm} (3.7)

where

\[ B_{i_1, \ldots, i_N} = |P|^{1/2} B_{i_1} \cdots B_{i_N} \]  \hspace{1cm} (3.8)

\[ r_{i_1, \ldots, i_N} = Wu_{i_1, \ldots, i_N} + \hat{r} \]  \hspace{1cm} (3.9)

\[ u_{i_1, \ldots, i_N} = [u_{i_1}, \ldots, u_{i_N}]^T \]  \hspace{1cm} (3.10)

where \( r_{i_1, \ldots, i_N} \) are the \( N \)-dimensional grid points and \( B_{i_1, \ldots, i_N} \) are the corresponding weights. \( B_{i_m}(m = 1, \ldots, N) \) and \( u_{i_m}(m = 1, \ldots, N) \) are the weights and the grid point for the one dimensional Gauss-Hermite quadrature from equation (3.1). The grid \( r \) is called a floating grid and it is centered at the current mean \( \hat{r} \). The square-root approach has the advantage over the eigenvector approach in that the computation of the square root of a matrix is faster than computation of the eigenvectors of a matrix.

The quadrature formula in equation (3.7) involves the summation of \( k^N \) weighted functional values, where \( N \) is the dimension of the quadrature and \( k \) is the order of the quadrature. For high dimensional problems it is desirable to keep the order of quadrature small in order to reduce the complexity of the quadrature. However, the result will not be accurate if the function \( f(\cdot) \) cannot be well approximated by a low order algebraic expression. In this case, one may factor \( f(\cdot) \) into two components: one is nearly algebraic and the other being Gaussian. The Gaussian component can then be combined with the original Gaussian weighting function to obtain a new Gaussian weighting function.
3.3 Basic Principles of the Proposed Filter

The central issue here is to compute the optimal (conditional mean) estimate based on the Bayesian equations accurately and efficiently using the multidimensional Gauss-Hermite quadrature technique. To reduce the number of quadratures required, analytic results are applied to the prediction stage (time update), and the application of quadrature techniques is restricted only to the update stage (measurement update), because the equation of target motion (2.12) is linear and the measurement equation (2.14) is nonlinear. Assume at time step n the mean $\hat{x}_{n-1|n-1}$ and the covariance matrix $P_{n-1|n-1}$ of the conditional density $p(x_{n-1}|Z^{n-1})$ is known. Then we approximate the distribution of the state prediction $p(x_n|Z^{n-1})$ as a Gaussian distribution with mean $\hat{x}_{n|n-1}$ and covariance matrix $P_{n|n-1}$.

$$\hat{x}_{n|n-1} = F \hat{x}_{n-1|n-1}$$
$$P_{n|n-1} = FP_{n-1|n-1}F^T + GQ_nG^T$$

This algorithm rests on the assumption that the state prediction density is Gaussian at each step. This assumption is closely satisfied in many situations of practical interest, in particular if the state noise $v_n$ is Gaussian [36].

Then the conditional mean (optimal estimate) $\hat{x}_{n|n}$ and the conditional covariance $P_{n|n}$ can be determined from the Bayesian equations as follows:

$$\hat{x}_{n|n} = \int x_n p(x_n|Z^n) dx_n$$
$$= \frac{\int x_n p(z_n|x_n)p(x_n|Z^{n-1}) dx_n}{\int p(z_n|x_n)p(x_n|Z^{n-1}) dx_n}$$

$$P_{n|n} = \int x_n^T x_n p(x_n|Z^n) dx_n - \hat{x}_{n|n}^T \hat{x}_{n|n}$$
$$= \frac{\int x_n^T x_n p(z_n|x_n)p(x_n|Z^{n-1}) dx_n}{\int p(z_n|x_n)p(x_n|Z^{n-1}) dx_n} - \hat{x}_{n|n}^T \hat{x}_{n|n}$$

The above integrations generally cannot be accomplished in closed form because the densities of the state are non-Gaussian. To make the calculations of the conditional mean and the conditional covariance possible and efficient, we replace the integrals involved in the Bayesian equations with a weighted sum of samples of the integrand.
based on the Gauss-Hermite quadrature formula. This requires the evaluation of the integration of the form

$$I = \int A(x_n)p(z_n|x_n)p(x_n|\mathcal{Z})dx_n$$

$$= \int C_1A(x_n)e^{-\frac{1}{2}(z_n-h(x_n))^TR_n^{-1}(z_n-h(x_n))}$$

$$\times e^{-\frac{1}{2}(x_n-x_{n|n-1})^TP_{n|n-1}^{-1}(x_n-x_{n|n-1})}dx_n$$  \hspace{1cm} (3.15)

where $C_1 = \frac{1}{2\pi|R_n|^\frac{1}{2}} \frac{1}{2\pi|P_{n|n-1}|^\frac{1}{2}}$ and $A(x_n)$ can be 1, $x_n$ or $x_n^T x_n$. To apply the multidimensional Gauss-Hermite quadrature formula, one may use the expression

$$F_1(x_n) = e^{-\frac{1}{2}(x_n-x_{n|n-1})^TP_{n|n-1}^{-1}(x_n-x_{n|n-1})}$$

as the Gaussian weighting function, but the remaining expression

$$A(x_n)F_2(x_n) = A(x_n)e^{-\frac{1}{2}(z_n-h(x_n))^TR_n^{-1}(z_n-h(x_n))}$$

is not algebraic. To determine the value of the integral in equation (3.15) accurately with $F_1(x_n)$ as the Gaussian weighting function will require a high-order quadrature formula. However, if the expression $F_2(x_n)$ is factored into two expressions: one is Gaussian, $F_2^1(x_n)$; the other is nearly algebraic within the desired region, $F_2^2(x_n)$ and a new weighting function determined by $F_1(x_n) \cdot F_2^1(x_n)$ is used, then the same accuracy can be obtained with a lower order integration formula (see Figure 3.1).

To determine $F_2^1(x_n)$ and $F_2^2(x_n)$, we use the linear approximation to the function $h(x_n)$.

$$\tilde{h}(x_n) = Hx_n + H_0$$  \hspace{1cm} (3.16)

where the matrices $H$ and $H_0$ can be determined either by first-order Taylor series expansion or the statistical linearization. The error of this linear approximation is then given by

$$e(x_n) = \tilde{h}(x_n) - h(x_n)$$  \hspace{1cm} (3.17)
Figure 3.1: With the factorization the order of the quadrature is reduced for a given accuracy. For example, in A, if $F_2(x)$ is the weighting function, the number of grid points required to cover the integrand $F_1(x)$ completely will be 7; however, if the integrand $F_1(x)$ is factorized and the new weighting function is $F_2(x) \times F_1^1(x)$, then the number of grid points required is reduced to 4 as shown in B.

The expression $F_2(x_n)$ can be rewritten as

$$F_2(x_n) = e^{-\frac{1}{2}(\bar{x}_n - \bar{\bar{h}}(x_n) + \varepsilon(x_n))^T R_{\varepsilon}^{-1}(\bar{x}_n - \bar{\bar{h}}(x_n) + \varepsilon(x_n))}$$

$$= F_2^1(x_n) F_2^2(x_n)$$

where

$$F_2^1(x_n) = e^{-\frac{1}{2}(\bar{x}_n - \bar{\bar{h}}(x_n))^T R_{\varepsilon}^{-1}(\bar{x}_n - \bar{\bar{h}}(x_n))}$$

$$F_2^2(x_n) = e^{(\bar{x}_n - \bar{\bar{h}}(x_n))^T R_{\varepsilon}^{-1} \varepsilon(x_n) - \frac{1}{2} \varepsilon(x_n)^T R_{\varepsilon}^{-1} \varepsilon(x_n)}$$

As a result, the new weighting function determined by $F(x_n) = F_1(x_n) \cdot F_2^1(x_n)$ becomes

$$F(x_n) = C_2 e^{-\frac{1}{2}(\bar{x}_n - \tilde{x}_{n|n})^T \tilde{P}_{n|n}^{-1}(\bar{x}_n - \tilde{x}_{n|n})}$$

where

$$\tilde{x}_{n|n} = \hat{x}_{n|n-1} + K_n (x_n - \hat{h}(x_n))$$

$$\tilde{P}_{n|n} = P_{n|n-1} - K_n H P_{n|n-1}$$

$$K_n = P_{n+1|n} H^T (P_{n|n-1} H^T + R_n)^{-1}$$
These equations are exactly the linearized Kalman filter equations and the constant term $C_2$ is not necessary to evaluate as it will be canceled later.

Applying the multidimensional Gauss-Hermite quadrature formula to the integral in Equation (3.15) with $F(x_n)$ in Equation (3.20) as the new weighting function and $A(x_n)F^2_2(x_n)$ in Equation (3.19) as the new integrand, the integral in Equation (3.15) becomes

$$I = C_1 C_2 \sum_{i_1=1}^{k} \sum_{i_N=1}^{k} B_{i_1,...,i_N} A(x_{n,i_1,...,i_N})F^2_2(x_{n,i_1,...,i_N}),$$  \hspace{1cm} (3.24)

where

$$W_n W_n^T = \tilde{P}_{n\mid n},$$  \hspace{1cm} (3.25)
$$x_{n,i_1,...,i_N} = W_n u_{i_1,...,i_N} + \tilde{x}_{n\mid n},$$  \hspace{1cm} (3.26)
$$u_{i_1,...,i_N} = [u_{i_1}, ..., u_{i_N}]^T,$$  \hspace{1cm} (3.27)
$$B_{i_1,...,i_N} = |W_n| B_{i_1} \cdots B_{i_N}$$  \hspace{1cm} (3.28)

where $W_n$ is the square root of $P_{n\mid n}$ from the Cholesky algorithm; $x_{n,i_1,...,i_N}$ is a $N$-dimensional grid point which is constructed from $N$ one-dimensional grid points $u_{ij} (j=1, \ldots, N)$ and $B_{i_1,...,i_N}$ is the corresponding weight which is just the product of $N$ one-dimensional weights $B_{ij} (j=1, \ldots, N)$. Utilizing the results from Equation (3.24), the conditional mean (optimal estimate) $\tilde{x}_{n\mid n}$ and the conditional covariance $P_{n\mid n}$ becomes

$$\tilde{x}_{n\mid n} = \frac{\sum_{i_1=1}^{k} \sum_{i_N=1}^{k} x_{n,i_1,...,i_N} B_{i_1,...,i_N} F^2_2(x_{n,i_1,...,i_N})}{\sum_{i_1=1}^{k} \sum_{i_N=1}^{k} B_{i_1,...,i_N} F^2_2(x_{n,i_1,...,i_N})}$$  \hspace{1cm} (3.29)
$$P_{n\mid n} = \frac{\sum_{i_1=1}^{k} \sum_{i_N=1}^{k} x_{n,i_1,...,i_N}^T x_{n,i_1,...,i_N} B_{i_1,...,i_N} F^2_2(x_{n,i_1,...,i_N})}{\sum_{i_1=1}^{k} \sum_{i_N=1}^{k} B_{i_1,...,i_N} F^2_2(x_{n,i_1,...,i_N})} - \tilde{x}_{n\mid n} \tilde{x}_{n\mid n}^T$$  \hspace{1cm} (3.30)

### 3.4 Complexity Analysis

This tracking algorithm basically consists of two parts: the first part is based on the linearized Kalman filter to obtain a rough estimate of the conditional mean and
the conditional covariance; the second part is based on the multidimensional Gauss-Hermite quadrature formula to compensate for any error introduced from the linear approximation.

The evaluation of the initial grid points \( u_{i_1, \ldots, i_N} \) and its corresponding weights \( B_{i_1, \ldots, i_N} \) in equation (3.27) and (3.28) requires the most computations; they are determined from a set of \( N \) one-dimensional grid points \( u_{ij} \) and weights \( B_{ij} \). Fortunately, these initial \( N \)-dimensional grid points and the weights can be computed off-line. Assume the dimension of the quadrature is \( N \) and the order of the quadrature is \( k \). The computational requirements at each iteration are as follows:

1. the calculation of the square root \( W_n \) of \( P_{n|n} \) from the Cholesky algorithm (3.25) requires \( O(N^2) \) operations;

2. the affine transformation (3.26) requires \( N^2 \) scalar multiplications and \( N^2 \) scalar additions for each grid points \( u_{i_1, \ldots, i_N} \).

3. the calculation of the conditional mean \( \hat{x}_{n|n} \) and the conditional covariance \( P_{n|n} \) requires \( k^N \) evaluations of \( F_2^2(\cdot) \), \( (N^2 + N + 1) \cdot k^N \) scalar multiplications and \( (N^2 + N + 1) \cdot (k - 1)^N \) scalar additions.

The whole algorithm requires excessive amount of additions and multiplications and data storage.

### 3.5 Filter Structure

The block diagram of the proposed filter is presented in Figure 3.2. It consists of the following three stages:

**Stage 0: Previous estimates:**

Assume at step \( n \) the previous mean and covariance estimates \( \hat{x}_n \) and \( P_{n|n} \) are known.
Stage 1: Linearized Kalman filter:

The new mean $\hat{x}_{n|n}$ and the new covariance $\hat{P}_{n|n}$ are first estimated by the linearized Kalman filter equations.

Stage 3: Error Compensation:

To compensate for the errors introduced in the linear approximation in Stage 2 multidimensional Gauss-Hermite quadrature is used to evaluate the conditional mean $\hat{x}_n$ and the conditional covariance $P_{n|n}$.

### 3.6 Simulation Results

To compare the performance of our proposed filter with that of currently popular approximate filters a two-dimensional target tracking application described by the system equation (2.12) and the measurement equation (2.14) with the following parameters is simulated.

$$x_{n+1} = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix} x_n + \begin{bmatrix} 1/2 \\ 0 \\ 0 \\ 1/2 \end{bmatrix} w_n$$  \hspace{1cm} (3.31)

$$z_n = \begin{bmatrix} \sqrt{x_n^2 + y_n^2} \\ \tan^{-1} y_n/x_n \end{bmatrix} + v_n$$  \hspace{1cm} (3.32)

$$Q = \begin{bmatrix} 0.0001 & 0 \\ 0 & 0.0001 \end{bmatrix} \quad R = \begin{bmatrix} 100 & 0 \\ 0 & 0.01 \end{bmatrix}$$


The following algorithms are used in the simulation: the extended Kalman filter (EKF), the converted measurement Kalman filter (CMKF) and proposed quadrature-based filter. All these filters require an initial filtered estimate $\hat{x}_{0\mid 0}$ and an initial error covariance $P_{0\mid 0}$. They are estimated from the first two measurements using the algorithm presented in Appendix B. The dimensions and the order of the quadrature we used is 4 and 16 respectively, i.e. the proposed filter has a computational complexity of order $16^4$. The results presented in Figures 3.3 and 3.4 are based on 100 measurements averaged over 500 independent realizations of the experiment with the sampling interval of one second and the initial state $x_0$ is chosen randomly for each realizations of the experiment.

The proposed filter is compared with the well-known classical filters, e.g. the EKF and the converted measurement Kalman filter (CMKF). The position errors and the velocity errors for each filters are shown in Figs. 3.3 and 3.4 where the error is defined as the root mean square of the difference between the actual value and the estimated value. Our proposed method converges faster and yields results of smaller error than the EKF and the converted measurement Kalman filter (CMKF) does whereas the EKF on average diverges due to the instability of the Jacobian matrix. Although this quadrature-based filter does yield more accurate estimates, the computations and the data storage it required are still too excessive. To reduce this computational burden, we must investigate some other means to represent the densities such that the optimal estimate can be computed in a simple and efficient fashion.
Figure 3.3: Comparison of the position errors

Figure 3.4: Comparison of the velocity errors
4.1 Introduction

Recall that it is generally impossible to accomplish the integration involved in the Bayesian equations in closed form when the measurement equation is nonlinear. This problem leads to either the numerical evaluation of the conditional densities or the investigation of density approximation for which the required integration can be accomplished in a straightforward manner. In this chapter, the Gaussian sum approximation method is used to approximate the conditional densities, such as $p(x_n|Z^n)$ such that the integration indicated in equations (2.18) and (2.22) can be accomplished in closed form, i.e. the optimal estimate can be computed analytically. The Gaussian sum approximation method can approximate any probability density function as closely as desired. Moreover, it is comprised of a bank of Kalman-type filters operating in parallel with each individual estimate possessing its own corresponding weighting term; hence, the optimal estimate can be computed efficiently as a weighted sum of the estimates from each Kalman-type filters.
4.2 Theoretical Foundations

Lemma: Given any probability density function \( p(x) \), it is possible to approximate it as closely as desired in the Hilbert space \( L_1(R^n) \) by a linear combination of Gaussian densities.

\[
p_A(x) = \sum_{i=1}^{N} \alpha_i \mathcal{N}(x - \mathbf{m}_i, \mathbf{B}_i) \quad (4.1)
\]

where

\[
\mathcal{N}(x - \mathbf{m}_i, \mathbf{B}_i) = \frac{1}{\sqrt{(2\pi)^n|\mathbf{B}_i|}} \exp \left[ \frac{1}{2} (x - \mathbf{m}_i)^T \mathbf{B}_i^{-1} (x - \mathbf{m}_i) \right] \quad (4.2)
\]

and the weighting factors \( \alpha_i \) have the constraints:

\[
\sum_{i=1}^{N} \alpha_i = 1 \quad \alpha_i \geq 0 \quad (4.3)
\]

and \( \mathbf{m}_i \) is a mean vector and \( \mathbf{B}_i \) is a positive definite covariance matrix. For a proof of this result, see Sorenson and Alspach [4].

These properties insure that the approximation is always positive and integrates to unity. This approximation is thus a true probability density function in its own right and when this representation is used the Bayesian recursion relations can be solved exactly in closed form.

4.3 Basic Principles of the Proposed Filter

To evaluate the state prediction density \( p(x_n|Z^{n-1}) \) efficiently from equation (2.18), we will assume the conditional density \( p(x_{n-1}|Z^{n-1}) \) to be Gaussian with mean \( \hat{x}_{n-1|n-1} \) and covariance matrix \( P_{n-1|n-1} \). Based on this assumption, the equation (2.18) can be evaluated in closed form and the resulting state prediction density \( p(x_n|Z^{n-1}) \) is a Gaussian density function with mean \( \hat{x}_{n|n-1} \) and covariance matrix \( P_{n|n-1} \).

\[
\hat{x}_{n|n-1} = F \hat{x}_{n-1|n-1} \quad (4.4)
\]

\[
P_{n|n-1} = FP_{n-1|n-1}F^T + GQ_nG^T \quad (4.5)
\]
Even though the density \( p(z_n|x_n) \) is a Gaussian density function with mean \( h(x_n) \) and covariance matrix \( R_n \), the integration involved in equation (2.22) cannot be accomplished in closed form, because the integration involved is done with respect to \( x_n \), not \( z_n \). The density \( p(z_n|x_n) \) when expressed in terms of \( x_n \) is a non-Gaussian function. The word function is used instead of density function because the density \( p(z_n|x_n) \) when expressed in terms of \( x_n \), denoted as \( p(z_n|x_n) \), has all the properties of a density function, except it doesn't have an area of unity, i.e. \( \int_{-\infty}^{\infty} p(z_n|x_n) dx_n \neq 1 \). Since the function \( p(z_n|x_n) \) is non-Gaussian, the integration involved in equation (2.22) cannot be accomplished analytically. The Gaussian sum approximation method is used to approximate this function such that the integration involved in the Bayesian equations can be accomplished in a simple manner, and the optimal state estimate \( \hat{x}_{opt,n|n} \) can be computed analytically and efficiently as a weighted sum of the estimates from a bank of Kalman-type filters.

4.3.1 Approximation of Densities

Since the function \( p(z_n|x_n) \) has all the properties of a density function, except it doesn't have an area of unity, the Gaussian sum approximation may be modified such that it can be used to approximate this function with a weighted sum of Gaussian density function, where the sum of weights is equal to the area of the function, instead of unity.

\[
p(z_n|x_n) \approx \sum_{i=1}^{N} \alpha_{n,i} \mathcal{N}(Dx_n - m_{n,i}, B_{n,i})
\]

(4.6)

where \( \sum_{i=1}^{N} \alpha_{n,i} = \int_{-\infty}^{\infty} p(z_n|x_n) dx_n \) and \( D = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \). It is however very difficult to choose the "best" parameters \( \alpha_{n,i}, m_i \) and \( B_i \) efficiently because the function \( p(z_n|x_n) \) depends on the current value of the measurement \( z_n \).

Classical Optimization Approach

This approach selects the parameters \( B_{n,i}, m_i \) and \( \alpha_{n,i} \) so that the \( L^k \) norm between the actual function \( p(z_n|x_n) \) and the Gaussian sum approximation \( p_{GS}(x_n) \) is
minimized \([4]\).

\[
\|p(z_n | x_n) - p_{GS}(x_n)\|^k = \int_{-\infty}^{\infty} \left| p(z_n | x_n) - \sum_{i=1}^{N} \alpha_{n,i} \cdot N(Dx_n - m_{n,i}, B_{n,i}) \right|^k dx_n
\]

As the number of terms \(N\) increases and as the covariance \(B_{n,i}\) decreases to zero, the \(L^k\) norm between the actual density and the approximation must vanish. However, for finite \(N\) and nonzero covariance matrix, it is reasonable to attempt to minimize the \(L^k\) norm such as this. In doing this, the stochastic estimation problem becomes a deterministic optimization problem.

To actually carry out this optimization procedure, we must obtain a set of equally spaced samples from the function \(p(z_n | x_n)\) and from the samples we determine the "best" sum of Gaussians [37]. Thus, the problem becomes to choose the parameters \(B_{n,i}, m_{n,i}\) and \(\alpha_{n,i}\) such that

\[
\sum_{j=1}^{K} \left| p(z_n | x_{n,j}) - \sum_{i=1}^{N} \alpha_{n,i} N(Dx_{n,j} - m_{n,i}, B_{n,i}) \right|^k < \epsilon
\]

where \(\{x_{n,j} : j = 1, \ldots, K\}\) is the set of uniformly spaced points and \(\epsilon\) is the prescribed accuracy. Note that in this approximation, the number of Gaussians used, \(N\), also has to be determined, and it is desirable to find the smallest \(N\) possible.

Assume the dimension of problem is \(m\) and the \(L^2\) norm is used; then a Gaussian term determined by parameters \(\hat{B}_{n,i}, \hat{m}_{n,i}\) and \(\hat{\alpha}_{n,i}\) has \((m + 1)(m + 2)/2\) unknowns. For a known number of Gaussian terms \(N\), this optimization problem is equivalent to that of solving a system of \(N(m + 1)(m + 2)/2\) nonlinear equations by least-squares method using \(K \gg N(m + 1)(m + 2)/2\) data points. To solve a system of nonlinear equations, we can use the Marquardt algorithm [38]. In this algorithm, starting from an initial set of parameters, the parameters are iteratively modified until a local minimum is reached in the sum-of-squared error between the data and the approximating function.
This approach is not very suitable for the radar tracking applications, because the function \( p(z_n|x_n) \) we want to approximate depends on the current value of the measurement \( z_n \). Thus, the optimization procedure indicated in equation (4.7) must be accomplished on-line for each new measurement, which is impossible because the dimensions of the problem and the number of parameters involved are too large to be handled in real-time. Consequently we propose two efficient and yet accurate methods to approximate the function \( p(z_n|x_n) \).

**Curve-fitting Approach**

**Fundamentals**

Our method utilizes the special symmetric geometry of the function \( p(z_n|x_n) \) to select the parameters \( \alpha_{n,i}, m_{n,i} \) and \( B_{n,i} \) to obtain the near-optimal Gaussian sum approximation for the function \( p(z_n|x_n) \). The function \( p(z_n|x_n) \) has a heavy-tailed structure and its projection onto the \( x_n \)-plane has a crescent shape. If we can fit this crescent shape with some ellipses geometrically, then we can represent the function \( p(z_n|x_n) \) as a sum of Gaussian mixture obtained from those ellipses, because the projection of a Gaussian function onto the \( x_n \)-space is an ellipse (see Figure 4.1).

![Figure 4.1](image_url)

Figure 4.1: (a) Plots of the function \( p(z_n|x_n) \) and (b) its projection
Figure 4.2: Basic Principles of the Proposed Gaussian mixture Approximation Method

Algorithm

Step 0: Define the contour of the projection as shown in Figure 4.1b. Here we assume the region outside three standard derivations of the density $p(z_n|x_n)$ has negligible probability; i.e.

$$\text{Contour } C : (z_n - h(x_n))^T R^{-1} (z_n - h(x_n)) = 9$$

This contour when plotted in terms of $z_n$ is an ellipse, but when plotted in terms of $x_n$ is the shape shown in Figure 4.1b. Define Pt. 1 be the measurement $z_n$; Line 1 be the line joining Pt. 1 and the origin; Line 2 be the line perpendicular to Line 1 and passing through Pt. 1.
Step 1: Denote the four interceptions of Contour C, Line 1 and Line 2 (see Figure 4.2a) as Pt. 1a-d. Then the first Gaussian term is defined to have a ellipsoidal base bounded by Pt. 1a-d and a height equal to the value of the function \( p(z_n|x_n) \) evaluated at Pt. 1. Or the first Gaussian term can be defined by the following weight \( \alpha_1 \), mean \( \mathbf{m}_1 \) and covariance matrix \( \mathbf{B}_1 \): 

\[
\alpha_1 = 2\pi|\mathbf{B}_1|^{1/2} \bar{\alpha}_1 \\
\mathbf{m}_1 = \begin{bmatrix} z_1 \\ y_1 \end{bmatrix} \\
\mathbf{B}_1 = k_1 \begin{bmatrix}
(\sigma_{1b} \sin(\theta_1))^2 + (\sigma_{1a} \cos(\theta_1))^2 & (\sigma_{1a}^2 - \sigma_{1b}^2) \sin(\theta_1) \cos(\theta_1) \\
(\sigma_{1a}^2 - \sigma_{1b}^2) \sin(\theta_1) \cos(\theta_1) & (\sigma_{1b} \cos(\theta_1))^2 + (\sigma_{1a} \sin(\theta_1))^2
\end{bmatrix}
\]

where

\[
\bar{\alpha}_1 = p(z_n|\mathbf{D}x_n = \mathbf{m}_1) \\
k_1 = 9/(9 + 2\ln(2\pi|\mathbf{R}_n|^{1/2} \bar{\alpha}_1)) \\
\sigma_{1a} = |\text{Pt. 1} - \text{Pt. 1a}|/3 \\
\sigma_{1b} = |\text{Pt. 1} - \text{Pt. 1b}|/3
\]

Step 2: Define Arc A be the arc \( r = r_n \); Line 3 be the line joining Pt. 1a and the origin; Line 4 be the line perpendicular to Line 3 and passing through Pt. 2, where Pt. 2 is defined as the interception of Line 3 and Arc A. Repeat Step 1 with Line 1 replaced by Line 3, Line 2 by Line 4 and Pt. 1 by Pt. 2 and we can obtain the second Gaussian term (see Figure 4.2b). The rest Gaussian terms are obtained in a similar manner. This algorithm is terminated either when the number of terms exceeds a certain prescribed value or when Pt. 2 is close to the tips of Contour C (see Figure 4.2d).

Numerical Examples

We compare the accuracy and the efficiency of the proposed approximation method with the classical optimization method in approximating the function \( p(z_n|x_n) \) with a sum of Gaussian terms. We assume the noise \( w_n \) is a white Gaussian process sequence with zero mean and covariance matrix \( \mathbf{R}_n = \begin{bmatrix} \sigma_x^2 & 0 \\ 0 & \sigma_\theta^2 \end{bmatrix} \). We further assume

---

1The detail derivation of this curve-fitting approach is presented in Appendix C
the measurement \( z_n \) is equal to \[
\begin{bmatrix}
\sigma_r = 10 \text{km} \\
\sigma_\theta = \pi/4 \text{rad}
\end{bmatrix}
\]. The comparison is based on three different scenarios: in Case 1 \( \sigma_r = 10 \text{m} \) and \( \sigma_\theta = 0.1 \text{rad} \); in Case 2 \( \sigma_r = 20 \text{m} \) and \( \sigma_\theta = 0.2 \text{rad} \); in Case 3 \( \sigma_r = 50 \text{m} \) and \( \sigma_\theta = 0.25 \text{rad} \). The simulation results are summarized in Table 4.1. They show that the proposed Gaussian sum approximation method is very efficient but yet yields accurate results comparable to the classical optimization method.

### Transformation Approach

#### Fundamentals

Recall that the function \( p(z_n|x_n) \) can be defined by measurement equation (2.14) and the known statistics of \( w_n \)

\[
p(z_n|x_n) = \int p(z_n|x_n,w_n)p(w_n|x_n)dw_n \\
= \int \delta(z_n - h(x_n) - w_n)p_w(w_n)dw_n \\
= p_w(z_n - h(x_n))
\]  

(4.16)
where \( \delta(\cdot) \) is the Dirac delta function. The delta function is used because if \( x_n \) and \( w_n \) are known, then \( x_n \) can be obtained deterministically. Thus, the function \( p(z_n|x_n) \) can be obtained by applying the transformation \( w_n = z_n - h(x_n) \) to the density function \( p_w(w_n) \) (see Figures 4.4A and 4.3A). Utilizing this observation, we select some initial parameters \( \tilde{\alpha}_{n,i}, \tilde{m}_{n,i} \) and \( \tilde{B}_{n,i} \) from the known statistics of the noise \( w_n \), and transform these parameters from the \( w_n \)-space to the \( x_n \)-space based on the transformation \( w_n = z_n - h(x_n) \) and finally collect them as a Gaussian sum approximation for the function \( p(z_n|x_n) \) (see Figures 4.4B and 4.3B). This approach is more efficient than the classical approach because most of the computations is involved in the initialization which can be done off-line; the only computation required at each iteration is for updating the initial parameters.

Figure 4.3: Fundamentals of the proposed Gaussian sum approximation method

**Algorithm**

**Step 0:** For initialization, select the parameters \( \tilde{\alpha}_{n,i}, \tilde{m}_{n,i} \) and \( \tilde{B}_{n,i} \) for a prescribed value of \( N \) such that the following sum-of-squared error is minimized.

\[
\sum_{j=1}^{K} \left| p_w(w_{n,j}) - \sum_{i=1}^{N} \tilde{\alpha}_{n,i} N(w_{n,j} - \tilde{m}_{n,i}, \tilde{B}_{n,i}) \right|^2 < \epsilon
\]

(4.17)

where \( \{w_{n,j}: j = 1, \ldots, K\} \) is the set of uniformly spaced points distributed through the region containing non-negligible probability and \( \epsilon \) is the prescribed accuracy.
Figure 4.4: A. The function \( p(z_n|x_n) \) can be obtained through a nonlinear transformation from the density \( p(w_n) \). B. Similarly, some initial parameters from the statistics of the noise \( w_n \) can be transformed into a Gaussian sum approximation for the function \( p(z_n|x_n) \).

Step 1: For each new measurement \( z_n \), update the new parameters \( \alpha_{n,i}, m_{n,i} \) and \( B_{n,i} \), such that

\[
p(z_n|x_n) \approx \sum_{i=1}^{N} \alpha_{n,i} \mathcal{N}(m_{n,i} - Dx_n, B_{n,i})
\]

where

\[
m_{n,i} = h^{-1}(\tilde{m}_{n,i})
\]

\[
\tilde{m}_{n,i} = z_n - \tilde{m}_{n,i}
\]

\(^2\text{The detail derivation of this transformation approach is presented in Appendix D}\)
Numerical Examples

We compare the accuracy and the efficiency of the proposed approximation method with the classical curve-fitting method in approximating the function $p(z, \mathbf{x})$ with a sum of Gaussian terms. We assume the noise $w_n$ is a white Gaussian process sequence with zero mean and covariance matrix $R_n = \begin{bmatrix} \sigma^2 & 0 \\ 0 & \sigma_\theta^2 \end{bmatrix}$ where $\sigma_r = 10m$ and $\sigma_\theta = 0.1rad$. We further assume the current measurement $z_n = \begin{bmatrix} r_n \\ \theta_n \end{bmatrix}$ is known where $r_n = 1km$ and $\theta_n = \pi/4rad$.

The simulation results are summarized in Figure 4.5 and Table 4.2. They show that the proposed Gaussian sum approximation method is very efficient but yet yields accurate results comparable to the classical optimization method.
Table 4.2: Comparison of the accuracy and the efficiency of the proposed Gaussian sum approximation method and the classical method. Note that the classical method is based on the Marquardt algorithm and the number of Gaussian terms \( N \) is fixed to be 5 for both methods.

### 4.3.2 Bank of Kalman Filters

Either using the curve-fitting approach or using the transformation approach, we will end up with a Gaussian sum approximation for the density \( p(z_n|x_n) \). If we substitute this approximation into equation (2.22) and complete the squares \(^3\), the \( a \text{ posteriori} \) density \( p(x_n|Z^n) \) becomes

\[
p(x_n|Z^n) = \frac{p(x_n|Z^{n-1})p(z_n|x_n)}{\int p(x_n|Z^{n-1})p(z_n|x_n)dx_n} = \frac{1}{\int N(x_n - \hat{x}_{n|n-1}, P_{n|n-1})} \sum_{i=1}^{N} \alpha_{n,i} N(Dx_n - m_{n,i}, B_{n,i}) \frac{\sum_{i=1}^{N} \gamma_{n,i} N(x_n - \hat{x}_{n|n,i}, P_{n|n,i})}{\sum_{i=1}^{N} \gamma_{n,i}} = \sum_{i=1}^{N} \gamma_{n,i} N(x_n - \hat{x}_{n|n,i}, P_{n|n,i})
\]

where

\[
P_{n|n,i} = (D^T B_{n,i}^{-1} D + P_{n|n-1}^{-1})^{-1}
\]

\[
\hat{x}_{n|n,i} = P_{n|n,i} (D^T B_{n,i}^{-1} m_{n,i} + P_{n|n-1}^{-1} \hat{x}_{n|n-1})
\]

\[
\gamma_{n,i} = \frac{1}{\sum_{i=1}^{N} \gamma_{n,i}}
\]

\[^3\text{The detail derivation is presented in Appendix E}\]
4.3.3 Growing Memory Problem

Inherent in the Gaussian sum algorithm is a serious memory growing problem that causes the number of terms in the Gaussian sum to increase exponentially at each iteration. In this section we will present two approaches to alleviate this problem.

Equivalent Gaussian Approach

For simple applications the Gaussian sum approximation of the density \( p(x_n|Z^n) \) is collapsed into one equivalent Gaussian term [39]. Consequently, the total number of Gaussian terms is fixed to be \( N \). The mean and the covariance matrix of the equivalent Gaussian term, denoted \( \tilde{x}_{n|n} \) and \( P_{n|n} \) respectively, are given as follows:

\[
\tilde{x}_{n|n} = \sum_{i=1}^{N} \gamma_{n,i} \hat{x}_{n|n,i} \tag{4.28}
\]

\[
P_{n|n} = \sum_{i=1}^{N} \gamma_{n,i} \left[ P_{n|n,i} - (x_{n|n,i} - \tilde{x}_{n|n})(x_{n|n,i} - \tilde{x}_{n|n})^T \right] \tag{4.29}
\]

This method is simple and elegant and it usually yields very accurate estimates, because if the a posteriori density \( p(x_n|Z^n) \) can be approximated very closely by a sum of Gaussian density functions, then its mean can be estimated very accurately as a weighted sum of the means of those Gaussian density functions. Furthermore, to control the number of terms in the approximation we collapse the Gaussian mixture into one equivalent Gaussian term at the end of each iteration. However, by doing so we actually collapse the non-Gaussian nature of the a posteriori density \( p(x_n|Z^n) \) into a Gaussian density function, which of course can seriously degrade the performance of the filter. Hence, it will be desirable to reduce the number of terms in the approximation but at the same time retain the general structure of the distribution of the a posteriori density \( p(x_n|Z^n) \). This approach is called the mixture reduction approach.

Mixture Reduction Approach

For complex applications the number of Gaussian terms in the approximation is reduced to some specified limit \( M \) for \( M < N \). As a result, the total number of Gaussian
terms is fixed to be \( N \cdot M \). This approach results in another Gaussian sum approximation with lesser terms without modifying the structure of the distribution beyond some acceptable limit and it should preserve the mean and the covariance matrix of the original Gaussian sum. Suppose that the number of original Gaussian sum approximation is reduced by merging several terms together. If \( c \) is the set of Gaussian terms to be merged, then in order to preserve the mean and covariance matrix of the original Gaussian sum, the weight \( \gamma_{n,c} \), the mean \( \hat{x}_{n|n,c} \) and the covariance matrix \( P_{n|n,c} \) should be chosen \([40]\) as follows:

\[
\gamma_{n,c} = \sum_{i \in c} \gamma_{n,i} \tag{4.30}
\]

\[
\hat{x}_{n|n,c} = \frac{1}{\gamma_{n,c}} \sum_{i \in c} \gamma_{n,i} \hat{x}_{n|n,i} \tag{4.31}
\]

\[
P_{n|n,c} = \frac{1}{\gamma_{n,c}} \sum_{i \in c} \gamma_{n,i} (P_{n|n,i} + \hat{x}_{n|n,i} \hat{x}_{n|n,i}^T) - \hat{x}_{n|n,c} \hat{x}_{n|n,c}^T \tag{4.32}
\]

Ideally the partition of the Gaussian terms for merging should be such that some cost function is minimized. However, to reduce the Gaussian sum from \( N \) terms to \( M \) terms this could involve the evaluation of the criterion for every possible partition to identify the minimum. Such a procedure would be far too time-consuming; so a suboptimal approach has been adapted, in which a pair of Gaussian terms are merged at every iteration of the algorithm to minimize some chosen criterion.

**The Joining Algorithm**

This algorithm uses the Mahalanobis distance \([40]\) as the cost function

\[
d_{ij}^2 = \frac{\gamma_i \gamma_j}{\gamma_i + \gamma_j} (\hat{x}_i - \hat{x}_j)^T (P_i + P_j)^{-1} (\hat{x}_i - \hat{x}_j) \tag{4.33}
\]

where \( \gamma_i, \hat{x}_i \), and \( P_i \) are the weight, the mean and the covariance matrix of the \( i \)th Gaussian term; \( \gamma_j, \hat{x}_j \), and \( P_j \) are the weight, the mean and the covariance matrice of the \( j \)th Gaussian term. At each iteration, the two Gaussian terms which is closest in the sense of the Mahalanobis distance are combined to form a new Gaussian defined

---

\( ^4 \)When this mixture reduction approach is used, the Gaussian sum filter equations have to be modified. The modification is presented in Appendix F
by equations (4.30-4.32). To implement this algorithm a symmetric matrix containing the distance between every pair of Gaussian terms in the original Gaussian sum is evaluated. This merging should proceed until the minimum distance exceeds some specified threshold. At each iteration, the smallest element of the matrix is found and the corresponding pair of components are merged. This algorithm has a complexity of order $N^2$ where $N$ is the number of components in the original Gaussian sum. It can further simplified by merging the terms with the smallest and the second smallest weights together at each iteration until the number of remaining terms has been reduced below the specified value $M$. This method works because the terms with the smallest and the second smallest weights carry the least important information; thus, merging them will not modify the original mixture very much. This simplified joining algorithm has a complexity of order $N$.

### The Clustering Algorithm

This algorithm is based on the $K$-means clustering algorithm [41] which partitions the original Gaussian mixture into $M$ clusters such that the Mahalanobis distance between the Gaussian terms and each cluster center is minimized. Each cluster is then approximated by a single Gaussian defined by equations (4.30-4.32).

#### Modified $K$-means Clustering Algorithm

- Choose randomly $M$ of the Gaussian terms as the centers of the cluster $c_1, \ldots, c_M$.
- While there are no changes in the position of the cluster centers.
  - For each cluster
    * Use the cluster centers to classify the Gaussian terms into clusters by evaluating a $N \times M$ matrix $D$ with the entry $d_{ij}$ defined as the Mahalanobis distance between the $i$th Gaussian term and the $j$th cluster center $c_j$. To classify the $i$th Gaussian term into clusters for instance, we must find the smallest element on the $i$th row of the matrix $D$ and
the column that element belongs to corresponds to the cluster center that Gaussian term should go.

* Update the new cluster centers by collapsing each cluster into one single Gaussian using equations (4.30-4.32).

- end For

• end While

One drawback of this algorithm is that the results produced depend on the initial values for the cluster centers, and it frequently happens that suboptimal partitions are found.

Numerical Examples
To demonstrate this clustering algorithm we generate a Gaussian mixture with 50 randomly chosen components and then apply the algorithm to reduce its number of components to 10. Figure 4.6 shows the original Gaussian mixture and Figure 4.7 shows the Gaussian mixture after applying the clustering algorithm.

Figure 4.6: Mixture before Reduction (50 components)  Figure 4.7: Mixture after Reduction (10 components)
4.4 Filter Structure

The block diagram of the proposed Gaussian sum filter is presented in Figure 4.8. It consists of four stages:

Stage 0: Previous estimate
Assume at time $n$ the Gaussian mixture approximation of the conditional density $p(x_{n-1}|Z^{n-1})$ is known.

Stage 1: Gaussian Sum Approximation
The density $p(z_n|x_n)$ is approximated systematically by a weighted sum of Gaussian terms either by the curve-fitting approach or the transformation approach.

Stage 2: Bank of Kalman Filters
The Gaussian mixtures from Stage 0 and Stage 1 are passed to a bank of $N \times M$ Kalman filters which evaluate the new Gaussian mixture for the density $p(x_n|Z^n)$.
Stage 4: Mixture Reduction

To control the number of Gaussian terms, the density \( p(x_n|Z^n) \) is either collapsed into one equivalent Gaussian term or reduced to \( M \) Gaussian terms by the joining algorithm or the clustering algorithm.

4.5 Simulation Results

To compare the performance of our proposed filter with that of currently popular approximate filters two different scenarios are considered.

Scenario 1:

A two-dimensional target tracking application described by the system equation (2.12) and the measurement equation (2.14) with the following parameters and two different noise levels is simulated.

\[
x_{n+1} = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix} x_n + \begin{bmatrix} 1/2 & 0 \\ 1 & 0 \\ 0 & 1/2 \\ 0 & 1 \end{bmatrix} w_n
\]

\[
z_n = \begin{bmatrix} \sqrt{x_n^2 + y_n^2} \\ \tan^{-1} \frac{y_n}{x_n} \end{bmatrix} + v_n
\]

\[
Q = \begin{bmatrix} 0.001 & 0 \\ 0 & 0.0001 \end{bmatrix}
\]

\[
R = \begin{cases} (1) \begin{bmatrix} 100 & 0 \\ 0 & 0.01 \end{bmatrix}, & (2) \begin{bmatrix} 400 & 0 \\ 0 & 0.04 \end{bmatrix} \end{cases}
\]

The following algorithms are used in the simulation: the extended Kalman filter (EKF), the converted measurement Kalman filter (CMKF) and the proposed adaptive Gaussian sum filter (AGSF) which uses the Transformation approach in the Gaussian sum approximation and the Equivalent Gaussian approach to control the number of terms in the approximations. All these filters require an initial filtered
estimate $\mathbf{x}_{0|0}$ and an initial error covariance $\mathbf{P}_{0|0}$. They are estimated from the first two measurements using the algorithm presented in Appendix B. The number of the Gaussian terms we used in the approximation is 9 and they are collapsed into one equivalent Gaussian at the end of each iteration. The results presented in Figures 4.9-4.12 are based on 100 measurements averaged over 500 independent realizations of the experiment with the sampling interval of one second and two different measurement noise levels and the initial state $\mathbf{x}_0$ is chosen randomly for each realizations of the experiment.

The proposed filter is compared with the well-known classical filters, e.g. the extended Kalman filter (EKF) and the converted measurement Kalman filter (CMKF). The position errors and the velocity errors for each filters for each measurement noise levels are shown in Figs. 4.9, 4.10, 4.11 and 4.12 where the error is defined as the root mean square of the difference between the actual value and the estimated value. Our proposed method converges faster and yields estimates of smaller error than the EKF and the converted measurement Kalman filter (CMKF) does in all cases. Note that the extended Kalman filter (EKF) diverges in only some of the cases due to the instability of the Jacobian matrix as shown in figure 4.13 and 4.14.
Scenario 2:

A three-dimensional target tracking application described by the system equation (2.12) and the measurement equation (2.14) with the following parameters is simulated.

\[
x_{n+1} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} x_n + \begin{bmatrix} 1/2 & 0 & 0 \\ 0 & 1/2 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1/2 \\ 0 & 0 & 1 \end{bmatrix} w_n
\]

\[
z_n = \begin{bmatrix} \sqrt{x_n^2 + y_n^2 + z_n^2} \\ \tan^{-1} \frac{y_n}{x_n} \\ \cos^{-1} \frac{z_n}{\sqrt{x_n^2 + y_n^2 + z_n^2}} \end{bmatrix} + v_n
\]

\[
Q = \begin{bmatrix} 0.001 & 0 \\ 0 & 0.001 \end{bmatrix}
\]

\[
R = \begin{bmatrix} 100 & 0 & 0 \\ 0 & 0.01 & 0 \\ 0 & 0 & 0.01 \end{bmatrix}
\]

The following algorithms are used in the simulation: the extended Kalman filter (EKF), the converted measurement Kalman filter (CMKF) and the proposed adaptive Gaussian sum filter (AGSF) which uses the Transformation approach in the Gaussian sum approximation and the Mixture Reduction approach to control the number of terms in the approximations. All these filters require an initial filtered estimate \(\hat{x}_{0|0}\) and an initial error covariance \(P_{0|0}\). They are estimated from the first two measurements using the algorithm presented in Appendix B. The number of the Gaussian terms we used in the approximation is 49 and they are reduced to 25 equivalent Gaussian terms at the end of each iteration. The results presented in Figures 4.15 and 4.16 are based on 100 measurements averaged over 10 independent realizations of the experiment with the sampling interval of one second and the initial state \(x_0\) is
chosen randomly for each realizations of the experiment.

The proposed filter is compared with the well-known classical filters, e.g. the extended Kalman filter (EKF) and the converted measurement Kalman filter (CMKF). The position errors and the velocity errors for each filters for each measurement noise levels are shown in Figs. 4.15 and 4.16 where the error is defined as the root mean square of the difference between the actual value and the estimated value. Our proposed method clearly outperforms the others by converging faster and yielding estimates of smaller error. Note that both the extended Kalman filter (EKF) and the converted measurement Kalman filter (CMKF) diverges due to the highly nonlinear nature of the problem whereas our proposed adaptive Gaussian sum filter (AGSF) converges because it is able to retain the non-Gaussian nature of the distribution of the a posteriori density $p(x_n|Z^n)$ at each iteration.
Figure 4.9: Comparison of the position errors for measurement noise level 1

Figure 4.10: Comparison of the velocity errors for measurement noise level 1
Figure 4.11: Comparison of the position errors for measurement noise level 2

Figure 4.12: Comparison of the velocity errors for measurement noise level 2
Figure 4.13: Superposition of 100 realizations of experiment based on Extended Kalman Filter.

Figure 4.14: Superposition of 100 realizations of experiment based on Extended Kalman Filter.
Figure 4.15: Comparison of the position errors for 3-D tracking scenario

Figure 4.16: Comparison of the velocity errors for 3-D tracking scenario
Chapter 5

Conclusion

In this paper we have discussed the problem of target tracking in Cartesian coordinates with polar measurements. Two suboptimal tracking algorithms which is believed to be more accurate and efficient than the most widely used Taylor series methods, e.g. the EKF are introduced. The first method uses the multidimensional Gauss-Hermite quadrature to evaluate the optimal (conditional mean) estimate of the target states directly from the Bayesian equations. This quadrature technique replaces the integrals in the Bayesian equations with a weighted sum of samples of the integrand and this approximation can be very accurate if the integrand is smooth. However, this method has a major drawback, namely its computational complexity because of the dimension of the problem and the number and the order of the quadratures the number of samples involved are usually too large to be handled in real-time.

In this paper we have suggested several ways to reduce the computational requirements of this quadrature technique by reducing the dimension, the number and the order of the quadratures required for a given accuracy. To reduce the number of quadrature required, analytic results are applied to the prediction stage and the application of quadrature techniques is restricted to the update stage because only the measurement equation is nonlinear. To reduce the order of quadrature, the integrand is factored into two components: one is Gaussian and the other is nearly algebraic within the desired region. In this case, the Gaussian component can be combined the original Gaussian weighting function to obtain a new Gaussian weighting function.
Simulation results show this quadrature approach is more accurate than the classical methods, such as the EKF and the converted measurement Kalman filter (CMKF), but it still has a high computational complexity.

The second tracking algorithm is based on the Gaussian sum filter. The Gaussian sum approximation method yields very accurate results because it can approximate any probability density function as closely as desired; moreover, the optimal estimate (conditional mean) can be computed in a simple manner as a weighted sum of the estimates from a bank of Kalman-type filters operating in parallel. However, this method has a very high complexity since the number of Gaussian terms required in the approximation increases exponentially in time. To alleviate the computational burden associated with the Gaussian sum filter, we have found two efficient and systematic ways to approximate a non-Gaussian and measurement-dependent function by a weighted sum of Gaussian density functions. The first approach utilizes the special symmetric geometry of the function to select the number and the parameters of Gaussians in the mixture. The second approach selects some initial parameters from the known statistics of the noise and transforms these parameters based on the measurement equation to obtain the Gaussian sum approximation. We have derived the formula for updating the weights and the parameters involved in the bank of Kalman-type filters and we also have suggested several ways to control the number of terms in the Gaussian mixture at each iteration. The first approach is to collapse the mixture into one equivalent Gaussian and the second approach is to reduce the number of Gaussians to some specified limit using the proposed joining algorithm or the k-means clustering algorithm. Simulation results show that this Gaussian sum filter is more accurate than the classical methods, such as the EKF and the converted measurement Kalman filter (CMKF), and yet as efficient as they are.
Future Works

First of all, there are still some problems remains unsolved and requires more attentions:

- What is the optimal number of Gaussian terms we should use in the approximation?

- What is the number of terms we should keep in the mixture reduction process?

So far these numbers are pre-determined and therefore it will be desirable to have certain means to determine them analytically. However in the long run we should investigate

- A new adaptive and intelligent Gaussian sum approximation method for a general non-Gaussian and non-stationary distribution;

- A new adaptive and intelligent mixture reduction method which reduces the number of Gaussian terms but retains the general structure of the distribution.

We should also apply this filter to other applications such as the underwater target tracking [42].
Appendix A

Discretization of the Continuous-Time Equation of Motion

Assume in a single physical dimension, the target equations of motions may be represented by

\[ \dot{x}(t) = Fx(t) + Gv(t) \]  \hspace{1cm} (A.1)

where \( x(t) = [x(t), \dot{x}(t), \ddot{x}(t)]^T \) and \( v(t) \) is a white noise driving function with variance \( 2\alpha \sigma_m^2 \).

\[
F = \begin{bmatrix}
0 & 1 & 0 \\
0 & 0 & 1 \\
0 & 0 & \alpha \\
\end{bmatrix} \quad G = \begin{bmatrix}
0 \\
0 \\
1 \\
\end{bmatrix}
\]

The sensors have a constant scan interval sampling target position every \( \Delta T \) seconds, i.e. the measurements are taken in discrete time. The appropriate discrete time target equations of motions are given by

\[ x_{n+1} = \Phi(\Delta T, \alpha)x_n + u_n \]  \hspace{1cm} (A.2)
where \( \Phi(\Delta T, \alpha) \) is the target state transition matrix and \( u_n \) is the inhomogenous driving input. Since

\[
x(t + \Delta T) = e^{F\Delta T}x(t) + \int_t^{t+\tau} e^{F(\Delta T-\tau)}Gv(\tau)d\tau
\]  
(A.3)

it follows that

\[
\Phi(\Delta T, \alpha) = e^{F\Delta T}
\]  
(A.4)

\[
u_n = \int_t^{t+\tau} e^{F(\Delta T-\tau)}Gv(\tau)d\tau
\]  
(A.5)

It can be verified that

\[
\Phi(\Delta T, \alpha) = \begin{bmatrix}
1 & \Delta T & (-1 + \alpha\Delta T + e^{-\alpha\Delta T})/\alpha^2 \\
0 & 1 & (1 - e^{-\alpha\Delta T})/\alpha \\
0 & 0 & e^{-\alpha\Delta T}
\end{bmatrix}
\]  
(A.6)

When \( \alpha\Delta T \) is small, \( \Phi(\Delta T, \alpha) \) reduces to the Newtonian matrix

\[
\Phi = \begin{bmatrix}
1 & \Delta T & \Delta T^2/2 \\
0 & 1 & \Delta T \\
0 & 0 & 1
\end{bmatrix}
\]  
(A.7)

The noise sequence \( u_n \) satisfies

\[
u_n = \int_{n\Delta T}^{(n+1)\Delta T} \begin{bmatrix}
1/\alpha^2\{-1 + \alpha((n + 1)\Delta T - \tau) + \exp[-\alpha((n + 1)\Delta T - \tau)]\} \\
1/\alpha^2\{1 - \exp[-\alpha((n + 1)\Delta T - \tau)]\} \\
\exp[-\alpha((n + 1)\Delta T - \tau)]
\end{bmatrix} v(\tau)d\tau
\]  
(A.8)

Since \( v(t) \) is white noise, \( E[u_n u_{n+i}] = 0 \) for \( i \neq 0 \) so that \( u_n \) is a discrete time white noise sequence.
Appendix B

Track Initialization

Assume the initial density \( p(x_0) \) is white Gaussian. Its mean \( \hat{x}_0 \) and its covariance matrix \( P_0 \) can be estimated from the first two measurements \( z_{-1} \) and \( z_0 \).

Recall that the polar measurement \( z_n^p \) at time \( n \) is related to the Cartesian target state as follows:

\[
z_n^p = \begin{bmatrix} r_n \\ \theta_n \end{bmatrix} = \begin{bmatrix} \sqrt{x_n^2 + y_n^2} \\ \tan^{-1}(y_n/x_n) \end{bmatrix} + \begin{bmatrix} w_{r,n} \\ w_{\theta,n} \end{bmatrix} \tag{B.1}
\]

This polar coordinate measurement \( z_n^p \) is first converted to the Cartesian coordinate measurements \( z_n^c \) using the inverse transformation \( h^{-1}(z_n^p) \)

\[
z_n^c = \begin{bmatrix} z_{x,n} \\ z_{y,n} \end{bmatrix} = \begin{bmatrix} x_n \\ y_n \end{bmatrix} + \begin{bmatrix} \bar{w}_{x,n} \\ \bar{w}_{y,n} \end{bmatrix} \tag{B.2}
\]

The noise \( \bar{w}_n = [\bar{w}_{x,n}, \bar{w}_{y,n}]^T \) on the converted measurement \( z_n^c \) can be assumed to be a zero-mean white Gaussian noise process with covariance matrix \( M_n \) defined by equation (2.42).

Then the initial mean \( \hat{x}_0 \) is defined as

\[
\hat{x}_0 = \begin{bmatrix} E[x_0] \\ E[w_{x,0}] \\ E[y_0] \\ E[w_{y,0}] \end{bmatrix} = \begin{bmatrix} \begin{bmatrix} z_{x,0} \\ (\hat{x}_{x,0} - x_{x,-1})/\Delta T \end{bmatrix} \\ \begin{bmatrix} z_{y,0} \\ (\hat{x}_{y,0} - x_{y,-1})/\Delta T \end{bmatrix} \end{bmatrix} \tag{B.3}
\]
where

\[
\begin{align*}
E[x_0] &= E[x_{z,0} - w_{z,0}] = z_{x,0} \\
E[u_{z,0}] &= E[(x_{z,0} - x_{z,-1})/\Delta T] = (z_{x,0} - z_{x,-1})/\Delta T
\end{align*}
\]

Similarly, the initial covariance matrix \( P_0 \) is defined as

\[
P_0 = \begin{bmatrix}
M_0(1,1) & M_0(1,1)/\Delta T & M_0(1,2) & M_0(1,2)/\Delta T \\
M_0(1,1)/\Delta T & (M_0(1,1) + M_{-1}(1,1))/\Delta T^2 & M_0(1,2)/\Delta T & (M_0(1,2) + M_{-1}(1,2))/\Delta T^2 \\
M_0(2,1) & M_0(2,1)/\Delta T & M_0(2,2) & M_0(2,2)/\Delta T \\
M_0(2,1)/\Delta T & (M_0(2,1) + M_{-1}(2,1))/\Delta T & M_0(2,2)/\Delta T & (M_0(2,2) + M_{-1}(2,2))/\Delta T^2
\end{bmatrix}
\]

(E.4)

where

\[
\begin{align*}
E[(x_0 - E[x_0])^2] &= E[(x_0 - z_{x,0})^2] = E[\tilde{w}_{x,0}^2] = M_0(1,1) \\
E[(x_0 - E[x_0])(v_{z,0} - E[v_{z,0}]) &= E[\tilde{w}_{x,0}(\tilde{w}_{x,0} - w_{x,-1})/\Delta T] = M_0(1,1)/\Delta T \\
E[(v_{z,0} - E[v_{z,0}])^2] &= E[(\tilde{w}_{x,0} - w_{x,-1})^2/\Delta T^2] = (M_0(1,1) + M_{-1}(1,1))/\Delta T^2
\end{align*}
\]

and the matrices \( M_{-1} \) and \( M_0 \) are evaluated from equation (2.42) based on the first two measurements \( z_{-1} \) and \( z_0 \).
Appendix C

Detail Derivations of the Curve-fitting Approach

The objective here is to determine a weighted Gaussian density function such as it has the ellipsoidal base defined by Pt. 1a-d as shown in figure C.1 and the weights equal to the value of the function $p(z_n|x_n)$ at Pt. 1, where Pt. 1 is just the mean of the first Gaussian term.

![Weighted Gaussian Function](image)

Figure C.1: Weighted Gaussian Function

The ellipsoidal base can be determined by rotating the ellipse defined by the lengths $\sigma_1a$ and $\sigma_1b$ in Figure C.2A through the angle $\theta_1$; as a result the equivalent covariance matrix can be determined by applying an rotation affine transformation to
the covariance matrix \( \begin{bmatrix} \sigma_{1a}^2 & 0 \\ 0 & \sigma_{1b}^2 \end{bmatrix} \) as follows:

\[
B_1 = k_1 \begin{bmatrix}
\sigma_{1a}^2 \sin(\theta_1)^2 + (\sigma_{1a} \cos(\theta_1))^2 & (\sigma_{1a}^2 - \sigma_{1b}^2) \sin(\theta_1) \cos(\theta_1) \\
(\sigma_{1a}^2 - \sigma_{1b}^2) \sin(\theta_1) \cos(\theta_1) & (\sigma_{1b} \cos(\theta_1))^2 + (\sigma_{1a} \sin(\theta_1))^2
\end{bmatrix}
\] (C.1)

where

\[
s_{1a} = |\text{Pt. 1} - \text{Pt. 1a}|/3
\] (C.2)

\[
s_{1b} = |\text{Pt. 1} - \text{Pt. 1b}|/3
\] (C.3)

\[
\theta_1 = \tan^{-1}(y_1/x_1)
\] (C.4)

---

**Figure C.2: Affine Transformation of the ellipsoidal base**

The value of \( k_1 \) and the weight \( \alpha_1 \) are determined as follows: At Pt. 1,

\[
\alpha_1 = p(x_n | D_x_n - m_1) = \frac{\alpha_1}{2\pi|B_1|^{1/2}}
\]

\[
\alpha_1 = 2\pi\alpha_1 |B_1|^{1/2}
\] (C.5)

At three standard derivations level,

\[
\frac{1}{2\pi|R_n|^{1/2}} e^{-\frac{1}{2}} = \frac{\alpha_1}{2\pi|B_1|^{1/2}} e^{-\frac{1}{2}}
\]

\[
k_1 = \frac{9}{9 - 2 \ln(\frac{|B_1|^{1/2}}{\alpha_1|R_n|^{1/2}})}
\] (C.6)

Substitute equation (C.6) into equation (C.5), the value of \( k_1 \) becomes

\[
k_1 = \frac{9}{9 + 2 \ln(2\pi\alpha_1|R_n|^{1/2})}
\] (C.7)
Appendix D

Detail Derivation of the Transformation Approach

The function \( p(z_n | x_n) \) can be rewritten as

\[
p(z_n | x_n) = p_w(z_n - h(x_n))
\]
\[
\approx \sum_{i=1}^{N} \tilde{c}_{n,i} N(z_n - h(x_n) - \tilde{x}_{n,i}, \tilde{B}_{n,i})
\]
\[
\approx \sum_{i=1}^{N} \tilde{c}_{n,i} N(\tilde{m}_{n,i} - h(x_n), \tilde{B}_{n,i})
\]

where \( \tilde{m}_{n,i} = z_n - \tilde{m}_{n,i} \) and \( \tilde{c}_{n,i} \), \( \tilde{m}_{n,i} \) and \( \tilde{B}_{n,i} \) are the initial parameters determined from the known statistics of the noise \( w_i \) from equation (4.17) and

\[
N(\tilde{m}_{n,i} - h(x_n), \tilde{B}_{n,i}) = \frac{1}{2\pi |\tilde{B}_{n,i}|^{1/2}} e^{-\frac{1}{2}(m_{n,i} - h(x_n))^T \tilde{B}_{n,i}^{-1}(m_{n,i} - h(x_n))}
\]

is a non-Gaussian function of \( x_n \).

Denote \( F_i(x_n) \) as the exponential component of the function \( N(\tilde{m}_{n,i} - h(x_n), \tilde{B}_{n,i}) \):

\[
F_i(x_n) = -\frac{1}{2}(\tilde{m}_{n,i} - h(x_n))^T \tilde{B}_{n,i}^{-1}(\tilde{m}_{n,i} - h(x_n))
\]

Let consider the Taylor series expansion of the function \( F_i(x_n) \) about \( m_n \):

\[
F_i(x_n) = F(m_{n,i}) - J_{F_i}(m_{n,i})(Dx_n - m_{n,i})
\]
\[
+ \frac{1}{2} (Dx_n - m_{n,i}) H_{F_i}(m_{n,i})(Dx_n - m_{n,i})^T + H.O.T.
\]

(D.4)
where $J_{F_i}(m_{n,i})$ and $H_{F_i}(m_{n,i})$ are the Jacobian and the Hessian of the function $F_i(x_n)$ respectively.

\[
J_{F_i}(m_{n,i}) = \left. \frac{\partial F_i(x_n)}{\partial x_n} \right|_{x_n = m_{n,i}} = \frac{1}{2} J_h(m_{n,i})^T \tilde{B}_{n,i}^{-1} (m_{n,i} - h(m_{n,i})) \tag{D.5}
\]

\[
H_{F_i}(m_{n,i}) = \left. \frac{\partial^2 F_i(x_n)}{\partial x_n \partial x_n^T} \right|_{x_n = m_{n,i}} = -[H_h(m_{n,i})^T \tilde{B}_{n,i}^{-1} (m_{n,i} - h(m_{n,i})) + J_h(m_{n,i})^T \tilde{B}_{n,i}^{-1} J_h(m_{n,i})] \tag{D.6}
\]

where $J_h(x_n)$ and $H_h(m_{n,i})$ are the Jacobian and the Hessian of the function $h(x_n)$ respectively. If $m_{n,i}$ is chosen such that $m_{n,i} = h^{-1}(\bar{m}_{n,i})$\footnote{Here, we assume the function is invertible; however, if the inverse does not exist, then we must choose $m_{n,i}$ to be the most likely solution given $m_{n,i} = h(\bar{m}_{n,i})$}, the constant term $F(m_{n,i})$ becomes zero and the first order term $J_{F_i}(m_{n,i})$ also becomes zero; only the second order term and the higher order terms remain. Thus, the function $\mathcal{N}(\bar{m}_{n,i} - h(x_n), \tilde{B}_{n,i})$ can be approximated as a Gaussian function by taking only the second order term and ignoring the remaining higher order terms.

\[
\mathcal{N}(\bar{m}_{n,i} - h(x_n), \tilde{B}_{n,i}) \approx \frac{1}{2\pi |\tilde{B}_{n,i}|^{1/2}} e^{-\frac{1}{2} (Dx_n - m_{n,i})^T \tilde{B}_{n,i}^{-1} (Dx_n - m_{n,i})} = \beta_{n,i} \mathcal{N}(Dx_n - m_{n,i}, B_{n,i}) \tag{D.7}
\]

where

\[
m_{n,i} = h^{-1}(\bar{m}_{n,i}) \tag{D.8}
\]

\[
B_{n,i} = [J_h(m_{n,i})^T \tilde{B}_{n,i}^{-1} J_h(m_{n,i})]^{-1} \tag{D.9}
\]

\[
\beta_{n,i} = |J_h(m_{n,i})| \tag{D.10}
\]

Finally, substitute equation (D.7) into equation (D.1), the density $p(z_n|x_n)$ becomes

\[
p(z_n|x_n) \approx \sum_{i=1}^N \alpha_{n,i} \mathcal{N}(m_{n,i} - Dx_n, B_{n,i}) \tag{D.11}
\]

where $\alpha_{n,i} = \beta_{n,i} \bar{\alpha}_{n,i}$. 

64
Appendix E

Reproducing Property of Gaussian Densities

First we make some definitions:

\[ \begin{align*}
    a & \text{ be an } r \times 1 \text{ matrix (column vector)}, \\
    A & \text{ be an } r \times r \text{ symmetric, positive definite matrix}, \\
    b & \text{ be an } s \times 1 \text{ matrix (column vector)}, \\
    B & \text{ be an } s \times s \text{ symmetric, positive definite matrix}, \\
    D & \text{ be an } r \times s \text{ matrix}, \\
    x & \text{ be an } s \times 1 \text{ random matrix (column vector)}. \\
\end{align*} \]

Define

\[ C = (D^T A^{-1} D + B^{-1})^{-1} \]  
\[ = B - BD^T (A + DBD^T)^{-1} DB \]

(E.1)  
(E.2)

and

\[ c = C(D^T A^{-1} a + B^{-1} b) \]  
\[ = b + CD^T A^{-1} (a - Db) \]

(E.3)  
(E.4)

Also define \( N_r(x - a, A) \) be a \( r \)-dimensional Gaussian density function with mean \( a \)
and covariance matrix $A$.

$$
N_r(x - a, A) = \frac{1}{(2\pi)^{r/2}|A|^{1/2} e^{-\frac{1}{2}(x-a)^T A^{-1} (x-a)}}
$$

**Theorem**  Let variables $a$, $A$, $r$, $b$, $B$, $s$, $c$, $C$ and $D$ be defined as above. Then

$$
N_r(Dx - a, A)N_s(x - b, B) = N_s(x - c, C)N_r(Db - a, A + DBD^T)
$$

**Proof**  The proof is straightforward, though somewhat tedious, and involves completing the square and applying the matrix inversion lemma.

$$
N_r(Dx - a, A)N_s(x - b, B)
= \frac{1}{(2\pi)^{r/2}|A|^{1/2} (2\pi)^{s/2}|B|^{1/2} e^{-\frac{1}{2}[Dx-a]^T A^{-1} (Dx-a)+[x-b]^T B^{-1} (x-b)]}
= \frac{1}{(2\pi)^{r/2}|A|^{1/2} (2\pi)^{s/2}|B|^{1/2} e^{-\frac{1}{2} x^T (D^T A^{-1} D+B^{-1}) x-(aA^{-1}D+bB^{-1}) x-x^T (x^T D^T A^{-1} +B^{-1} b) + (a^T A^{-1} a+b^T B^{-1} b)}}
= \frac{1}{(2\pi)^{r/2}|A|^{1/2} (2\pi)^{s/2}|B|^{1/2} e^{-\frac{1}{2} [(x-(D^T A^{-1} D+B^{-1})^{-1}(D^T A^{-1} a+bB^{-1} b))^T (D^T A^{-1} D+B^{-1})^{-1} x-(D^T A^{-1} D+B^{-1})^{-1}(D^T A^{-1} a+bB^{-1} b)]}}
$$

From the definition of $c$ and $C$ from equations (E.3) and (E.1), the above equation becomes

$$
N_r(Dx - a, A)N_s(x - b, B)
= \frac{1}{(2\pi)^{r/2}|A|^{1/2} (2\pi)^{s/2}|B|^{1/2} e^{-\frac{1}{2} [(x-c)^T C^{-1} (x-c) +c^T A^{-1} a+b^T B^{-1} b]}}
= \frac{1}{(2\pi)^{r/2}|C|^{1/2} e^{-\frac{1}{2} [(x-c)^T C^{-1} (x-c)]}} \frac{1}{(2\pi)^{s/2}|A|^{1/2} |B|^{1/2} e^{-\frac{1}{2} [a^T A^{-1} a+b^T B^{-1} b -c^T C^{-1} c]}}
$$

Knowing the relations that

$$
\frac{|C|^{1/2}}{|A|^{1/2}|B|^{1/2}} = \frac{1}{(DBD^T + A)^{1/2}}
$$

$$
a^T A^{-1} a + b^T B^{-1} b - c^T C^{-1} c = (Db - a)^T (A + DBD^T)^{-1} (Db - a)
$$

(E.7)  (E.8)
the above equation can be simplified as follows:

\[ \mathcal{N}_r(Dx - a, A)\mathcal{N}_r(x - b, B) = \mathcal{N}_r(x - c, C)\mathcal{N}_r(Db - a, A + DBD^T) \]

Note that this theorem can be used to shift the dependence on \( x \) from a pair of Gaussians to a single Gaussian. From the theorem the product of two Gaussians is Gaussian whose normalizing constant is the reciprocal of another Gaussian independent of \( x \).
Appendix F

Bank of Kalman Filters

When the mixture reduction algorithm is used instead of the equivalent Gaussian algorithm, the resulting conditional density \( p(x_n|Z^n) \) will be expressed as a weighted sum of \( M \) Gaussian terms

\[
p(x_n|Z^n) = \sum_{i=1}^{M} \gamma_{n,i} \mathcal{N}(x_n - \hat{x}_{n|n,i}, P_{n|n,i}) \tag{F.1}
\]

Using this result the state prediction density \( p(x_{n+1}|Z^n) \) can be evaluated analytically from equation (2.18).

\[
p(x_{n+1}|Z^n) = \int p(x_n|Z^n) p(x_{n+1}|x) dx_n
\]

\[
= \int \sum_{i=1}^{M} \gamma_{n,i} \mathcal{N}(x_n - \hat{x}_{n|n,i}, P_{n|n,i}) \mathcal{N}(x_{n+1} - Fx_n, GQ_n G^T) dx_n
\]

\[
= \sum_{i=1}^{M} \gamma_{n,i} \mathcal{N}(x_{n+1} - \hat{x}_{n+1|n,i}, P_{n+1|n,i}) \tag{F.2}
\]

where

\[
\hat{x}_{n+1|n,i} = F\hat{x}_{n|i} \tag{F.3}
\]

\[
P_{n+1|n,i} = FP_{n|i}F^T + GQ_n G^T \tag{F.4}
\]

Utilizing the proposed Gaussian sum approximation, we can approximate the density \( p(z_{n+1}|x_{n+1}) \) as the following weighted sum of Gaussian terms

\[
p(z_{n+1}|x_{n+1}) = \sum_{i=1}^{N} \alpha_{n+1,i} \mathcal{N}(Dx_{n+1} - m_{n+1,i}, B_{n+1,i}) \tag{F.5}
\]
Substitute this result into equation (2.22) the conditional density \( p(x_{n+1}|Z^{n+1}) \) becomes

\[
p(x_{n+1}|Z^{n+1}) = \frac{p(z_{n+1}|x_{n+1})p(x_{n+1}|Z^n)}{\int p(z_{n+1}|x_{n+1})p(x_{n+1}|Z^n)dx_{n+1}}
\]

\[
= \frac{\sum_{j=1}^N \alpha_{n+1,j} N(Dx_{n+1} - m_{n+1,j}, B_{n+1,j}) \sum_{i=1}^M \gamma_{n,i} N(x_{n+1} - \hat{x}_{n+1|n+i}, P_{n+1|n+i})}{\int \sum_{j=1}^N \alpha_{n+1,j} N(Dx_{n+1} - m_{n+1,j}, B_{n+1,j}) \sum_{i=1}^M \gamma_{n,i} N(x_{n+1} - \hat{x}_{n+1|n+i}, P_{n+1|n+i})dx_{n+1}}
\]

\[
= \sum_{i=1}^M \sum_{j=1}^N \gamma_{n+1,i,j} N(x_{n+1} - \hat{x}_{n+1|n+1,i,j}, P_{n+1|n+1,i,j})/ \sum_{i=1}^M \sum_{j=1}^N \beta_{n+1,i,j}
\]

\[
= \sum_{i=1}^M \sum_{j=1}^N \gamma_{n+1,i,j} N(x_{n+1} - \hat{x}_{n+1|n+1,i,j}, P_{n+1|n+1,i,j})
\]  \hspace{1cm} (F.6)

where footnotesize

\[
P_{n+1|n+1,i,j} = (D^T B_{n+1,j}^{-1} D + P_{n+1|n,i}^{-1})^{-1}
\]  \hspace{1cm} (F.7)

\[
\hat{x}_{n+1|n+1,i,j} = P_{n+1|n+1,i,j} (D^T B_{n+1,j}^{-1} m_{n+1,j} + P_{n+1|n,i}^{-1} \hat{x}_{n+1|n,i})
\]  \hspace{1cm} (F.8)

\[
\beta_{n+1,i,j} = \alpha_{n+1,i} \left| P_{n+1|n+1,i,j} \right|^{1/2} \left| P_{n+1|n+i} \right|^{1/2} e^{-\frac{1}{2} \left[ \hat{x}_{n+1|n,i}^T P_{n+1|n,i}^{-1} \hat{x}_{n+1|n,i} + m_{n+1,j}^T B_{n+1,j}^{-1} m_{n+1,j} \right.}
\]

\[
- \hat{x}_{n+1|n+1,i,j}^T P_{n+1|n+1,i,j}^{-1} \hat{x}_{n+1|n+1,i,j} \right]^{1/2}
\]  \hspace{1cm} (F.9)

\[
\gamma_{n+1,i,j} = \beta_{n+1,i,j} / \sum_{j=1}^N \sum_{i=1}^M \beta_{n+1,i,j}
\]  \hspace{1cm} (F.10)

Finally, the optimal estimate \( \hat{x}_{n+1|n+1} \) at time step \( n+1 \) becomes

\[
\hat{x}_{n+1|n+1} = \sum_{i=1}^M \sum_{j=1}^N \gamma_{n+1,i,j} \hat{x}_{n+1|n+1,i,j}
\]  \hspace{1cm} (F.11)
Bibliography


TEST TARGET (QA-3)