Bayesian Filtering and Smoothing for Tracking in High Noise and Clutter Environments

by

Yongho Seo

A Thesis Presented in Partial Fulfillment
of the Requirement for the Degree
Master of Science

Approved April 2022 by the
Graduate Supervisory Committee:

Antonia Papandreaou-Suppappola, Chair
Daniel W. Bliss
Chaitali Chakrabarti
Bahman Moraffah

ARIZONA STATE UNIVERSITY

May 2022

ABSTRACT

Object tracking refers to the problem of estimating a moving object's time-varying parameters that are indirectly observed in measurements at each time step. Increased noise and clutter in the measurements reduce estimation accuracy as they increase the uncertainty of tracking in the field of view. Whereas tracking is performed using a Bayesian filter, a Bayesian smoother can be utilized to refine parameter state estimations that occurred before the current time. In practice, smoothing can be widely used to improve state estimation or correct data association errors, and it can lead to significantly better estimation performance as it reduces the impact of noise and clutter.

In this work, a single object tracking method is proposed based on integrating Kalman filtering and smoothing with thresholding to remove unreliable measurements. As the new method is effective when the noise and clutter in the measurements are high, the main goal is to find these measurements using a moving average filter and a thresholding method to improve estimation. Thus, the proposed method is designed to reduce estimation errors that result from measurements corrupted with high noise and clutter.

Simulations are provided to demonstrate the improved performance of the new method when compared to smoothing without thresholding. The root-mean-square error in estimating the object state parameters is shown to be especially reduced under high noise conditions.

To my beloved Mom, Wife, Son, Baby Girl and in memory of my Father

ACKNOWLEDGEMENTS

I appreciate my advisor, Professor Antonia Papandreou-Suppappola from the bottom of my heart. Her advise, inspiration and unchanging support always encourage me to keep going through even under any obstacle. She always gave me a strong belief and it makes me believe in myself whenever I doubt myself. I could proceed to the next step under her sincere and sharp advice.

I am also grateful to my committee, Professor Daniel Bliss, Professor Chaitali Chakrabarti, and Dr. Bahman Moraffah for the precious advice. It gave me many different perspectives that I had not considered, and could keep going my idea in diverse ways. I am also grateful to SPAS group at Arizona State University.

I also would like to thank the Republic of Korea Navy for supporting and giving me and my family a precious opportunity to study and live abroad.

TABLE OF CONTENTS

Page

CHAPTER Page

LIST OF TABLES

Table Page

LIST OF FIGURES

Chapter 1

INTRODUCTION

1.1 Motivation

In applications such as tracking, monitoring and surveillance, the aim is to accurately infer the trajectory of a moving object using sensor observations [1, 2]. For example, in radar tracking, the radar measurements collected at the current time step is processed to estimate the current position and velocity of a moving target [2, 3]. The accuracy of a tracking algorithm depends on multiple factors, including the assumed kinetic model, relevance of the measurements, prior knowledge of operational and environmental conditions during tracking.

Tracking algorithms use Bayesian filtering methods to estimate the time-varying state of a system [4, 5]. These methods use Bayesian statistics to estimate the probability density function (PDF) of the unknown state, together with sequential sensor measurements and kinematics models to provide a physics-based description of the system process. The Kalman filter (KF) is an example of Bayesian filtering that provides a recursive solution for linear systems under Gaussian noise assumptions [6, 7]. Modified versions of the KF for use with nonlinear systems include the extended KF (EKF) which is based on linearization approaches [8, 9] and the unscented KF (UKF) which is based on approximate filtering approaches [9–11]. For highly nonlinear or non-Gaussian systems, sequential Monte Carlo methods, such as particle filtering, provide higher tracking accuracy than the modified KFs approaches [12–14].

In realistic tracking scenarios, Bayesian smoothing may be required in addition to Bayesian filtering to further improve tracking estimation. This is especially the

case when tracking under high noise or heavy clutter conditions. Whereas Bayesian filtering uses prior estimates to predict the unknown state before updating it with incoming measurements, Bayesian smoothing makes use of posterior information. In particular, smoothing makes predictions by estimating posterior information for the current time step using measurements obtained after the time of interest [15–23]. Using both Kalman filtering and smoothing can results in overall better state estimation, depending on the tracking scenario [4, 18, 21, 22, 24]. The two-filter smoothing approach integrates the estimates obtained from forward filtering and backward smoothing [21, 25, 26]. Kalman smoothing (KS), that is applicable to linear Gaussian systems, performs a separate backward smoothing after the forward filtering from the KF [16, 27]. The forward-backward pass algorithm used by the KS uses the Rauch-Tung-Striebel (RTS) algorithm, which is an efficient algorithm for fixed interval smoothing [17]. Extensions to nonlinear systems include extended Kalman smoothing (EKS) and unscented Kalman smoothing (UKS) [8, 18, 22, 24]. Note that smoothing has also been applied to particle filtering for nonlinear and non-Gaussian systems [28, 29].

Many applications which are not required a real-time process, in practice, are used to improve their performance with additional processing after collecting all measurements since the estimation accuracy can be further improved using all measurements than using only real-time measurements. For instance, when monitoring vital signs in healthcare, a large probability of false alarm may only require additional testing whereas a high probability of miss could lead to long term complications. Another example is the use of ground penetrating radar in forensic investigations [30]. A high probability of false alarm may result in further testing in order to locate a gravesite whereas a high probability of miss may lead to unsolved murder cases. Our proposed work was originated from this perspective, and designed to improve the tracking

accuracy by integrating the Bayesian filtering and smoothing method.

1.2 Proposed Work

In this thesis, we consider a nonlinear state space system formulation for tracking a moving object using sensor measurements that are corrupted by high noise and clutter. We propose a tracking approach that, in addition to Bayesian filtering and smoothing, uses a thresholding to increase state estimation accuracy. In particular, after the unknown state is sequentially estimated over all time steps using the UKF, we use a process that is aimed to eliminate measurements that may have resulted from false alarms. Such measurements could have been accepted as true detections during matched filtering, even though they did not contain any information on the object [31]. The method compares the estimated state at each time step to a neighborhood of estimated states, both from previous and future time steps, using a distance-based metric. If the metric threshold is exceeded at a particular time step, then the measurement-updated state is replaced with the predicted value of the UKF process. After thresholding, we use the UKS to further improve the state estimation accuracy.

We use simulations to compare the estimation mean-square error before and after the proposed thresholding process for different tracking scenarios with high noise and/or high clutter. We also studied the effect of varying the parameters that affect the performance of the proposed method, such as the metric threshold and the size of the neighborhood. Our simulations demonstrate that tracking improvement is achieved in high noise tracking scenarios.

1.3 Thesis Organization

The remainder of the thesis is organized as follows. In Section 1.3, we provide a list of mathematical symbols and acronyms that are used throughout the thesis. In Chapter

2, we provide background information on sampling methods, Bayesian inference and estimation, Bayesian filtering and Bayesian smoothing. In Chapter 3, formulate our tracking problem and describe our new proposed approach. In Chapter 4, provide simulations to demonstrate the performance of our proposed method. We compare the performance between the original and the proposed method and discuss the results. In Chapter 5, we provide concluding remarks and discuss future directions.

1.4 Thesis Acronyms and Notation

In this section, we provide a list of the acronyms and a list of the mathematical symbol notation that we used in the thesis. Note that, throughout the thesis, we use lower case letters to denote scalars (e.g., x, t, α), boldface lower case letters to denote vectors (e.g., $\mathbf{x}$, $\boldsymbol{\phi}$), boldface upper case letters to denote matrices (e.g., $\mathbf{X}$), upper case calligraphy letters to denote sets (e.g., $\mathcal{N}$), and blackboard-bold form to denote spaces (e.g., $\mathbb{H}$).

List of Mathematical Symbols

x, t, α, β	Scalars
$\mathbf{x}, \mathbf{t}, \boldsymbol{\alpha}, \boldsymbol{\beta}$	Vectors
$\mathbf{X}, \mathbf{Y}, \mathbf{F}, \mathbf{H}$	Matrices
$\mathcal{X}, \mathcal{Y}, \mathcal{F}, \mathcal{H}$	Sets
$\mathbb{X}, \mathbb{Y}, \mathbb{F}, \mathbb{H}$	Spaces
$\mathbf{A}^\top$	Transpose of matrix
$\mathbf{A}^{-1}$	Inverse of matrix
$\mathbf{A}^{-\top}$	Inverse of transpose of matrix
$\lvert a \rvert$	absolute value of scalar a
$\lvert \mathbf{A} \rvert$	Determinant of matrix $\mathbf{A}$
$\sqrt{\mathbf{P}}$	Matrix such that $\mathbf{P} = \sqrt{\mathbf{P}}\sqrt{\mathbf{P}}^\top$
$E[\mathbf{x}]$	Expectation of the random variable $\mathbf{x}$
$E[\mathbf{x}\lvert\mathbf{z}]$	Conditional expectation of the random variable $\mathbf{x}$ given $\mathbf{z}$
$\mathrm{Cov}(\mathbf{x})$	Covariance of the random variable $\mathbf{x}$, $E[(\mathbf{x} - E[\mathbf{x}])(\mathbf{x} - E[\mathbf{x}])^\top]$

$p(\mathbf{x})$	Probability density function (PDF) of continuous (or discrete) random variable $\mathbf{x}$
$p(\mathbf{x}\|\mathbf{z})$	Conditional probability density function $\mathbf{x}$ given $\mathbf{z}$
$\mathcal{N}(\cdot)$	Gaussian distribution(i.e., normal distribution)
$\mathcal{U}(\cdot)$	Uniform distribution
$a \triangleq b$	a is defined to be equal to b
$a \approx b$	a is approximately equal to b
$a \propto b$	a is proportional to b
$\mathbf{x}_{1:K}$	Set or sequence of the vectors $\{\mathbf{x}_1, \ldots, \mathbf{x}_K\}$
κ	Parameter of the unscented transform
λ	Parameter of the unscented transform
$g(\cdot)$	Dynamic transition function in a state space model
$h(\cdot)$	Measurement model function in a state space model
$\mathbf{I}$	Identity matrix
$\mathbf{J}(\cdot)$	Jacobian matrix
$\mathbf{K}$	Kalman gain matrix
$\mathbf{P}$	Covariance of the Gaussian distribution
$\mathbf{Pois}(\lambda)$	Covariance of the Gaussian distribution
$\mathbf{Q}$	Covariance of the process noise
$\mathbf{R}$	Covariance of the measurement noise
$\mathbb{R}^n$	n-dimensional space of real numbers
$\mathbf{S}$	Innovation covariance of a Kalman filter
W_i	ith weight in sigma-point approximation

$\mathbf{x}$	Random variable or state
$\mathcal{X}^{(\cdot)}$	Sigma point of $\mathbf{x}$
$\mathbf{z}$	Random variable or measurement
$\mathcal{Z}^{(\cdot)}$	Sigma point of $\mathbf{y}$
Z	Normalization constant

List of Acronyms

EKF	Extended Kalman filter
EM	Expectation-maximization
ERTSS	Extended Rauch-Tung-Striebel smoothing
FIR	Finite impulse response
FISST	Finite set statistics
FOV	Field of view
HMM	Hidden Markov model
HNHC	High Noise and High Clutter
HNLC	High Noise and Low Clutter
KF	Kalman filter
LNHC	Low Noise and High Clutter
MA	Moving average
MAP	Maximum a posteriori
MC	Monte Carlo
MCMC	Markov chain Monte Carlo
MH	Metropolis-Hastings
ML	Maximum likelihood
PF	Particle filter
PPP	Poisson point process
PS	Particle smoothing
RFS	Rapid finite set
RMSE	Root mean squared error
RTSS	Rauch-Tung-Striebel smoothing
SIR	Sequential importance resampling
SIR-PS	Sequential importance resampling particle smoothing

SIS	Sequential importance sampling
SMC	Sequential Monte Carlo
T-UKF	Unscented Kalman filter after thresholding process
T-UKS	Unscented Kalman smoothing after thresholding process
UKF	Unscented Kalman filter
UKS	Unscented Kalman smoothing
URTSS	Unscented Rauch-Tung-Striebel smoothing
UT	Unscented transform

Chapter 2

BAYESIAN PROCESSING

2.1 Monte Carlo Sampling Methods

Monte Carlo (MC) methods are used to approximate a solution to a problem by randomly drawing samples [32]. MC integration methods are often used to estimate the average value of a continuous function within an interval. Following the law of large numbers in probability theory, the integral of a function over a given interval can be approximated by taking the sample mean of independent uniform samples of the function variable. A common use of the MC method is to estimate the value of π, as demonstrated in Figure 2.1. The value was estimated by drawing a circle to fully occupy a square, uniformly selecting points in the square, and counting the number of points inside the circle. This corresponds to counting the points located within a radius of the circle from the origin.

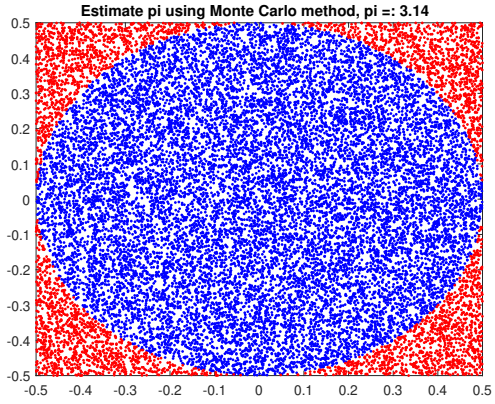


Figure 2.1: Estimation of π using 20,000 Monte Carlo points.

MC sampling methods form a class of algorithms that can be used to randomly sample a probability distribution [33–37]. Direct sampling methods can be used to

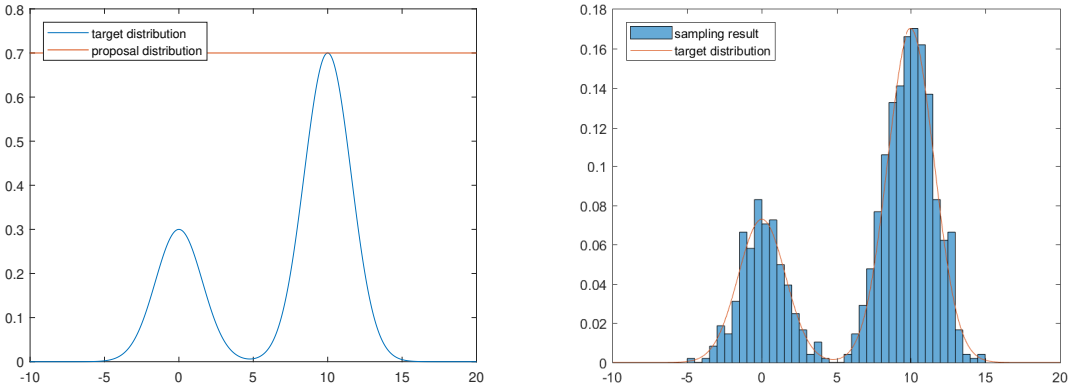
obtain samples from a probability density function (PDF) that can be written in closed form, such as Gaussian or uniform PDFs. In most cases, however, samples must be realized using simulation-based methods to approximate unknown PDFs in practical problems. Bayesian processing methods are often used to estimate the posterior PDF, which is the conditional PDF of unknown state parameters given sensor measurements. This PDF can be estimated by generating samples from other PDFs, such as the prior PDF and the likelihood function.

2.1.1 Rejection Sampling Method

The rejection sampling method is an example of an MC sampling. It can be used to generate samples from a random variable Y, with PDF $p_Y(y) = p(y)$, using a proposal distribution $q(y)$. Even though the PDF of Y cannot be written in closed form, the rejection sampling method assumes that a constant number of samples M, y_m, $m = 1, \ldots, M$, can be generated from the proposal $q(y)$ provided that M satisfies

$$p(y) \leq M\, q(y), \quad \forall y\,.$$

The rejection sampler draws sample y from the proposal $q(y)$ and then sample u such that $u < z$, with $z = p(y)/(M\, q(y))$. Then, the generated two samples (y, z) are uniformly distributed over $p(y)$. Finally, u is rejected if $u > z$ and accepted otherwise [38]. These steps show that, even though we cannot sample directly from $p(y)$, samples of Y can be drawn from the function $M\, q(y)$. The process is repeated until sampled pairs approximately follow the desired distribution$p(y)$. Thus, samples for Y are generated from the proposal distribution $q(y)$.



(a) Target and proposal distribution (b) Rejection sampling results

Figure 2.2: Target Distribution and Rejection Sampling.

2.1.2 Markov Chain Monte Carlo Sampling Method

Markov chain Monte Carlo (MCMC) methods sample from a probability distribution by constructing a Markov chain whose equilibrium distribution approaches the unknown distribution [32, 38–40]. A Markov chain is a discrete-time random process that model a future event whose behavior only depends on the current event, and not on past events [41]. The corresponding continuous-time process is called Markov process. A first-order Markov chain of a discrete-time random process y_k satisfies the memoryless property. Specifically, given random samples $y_1, \ldots, y_{k-1}$, the conditional PDF of y_k satisfies

$$p(y_k \mid y_1, \ldots, y_{k-1}) = p(y_k \mid y_{k-1}) \,.$$

The MCMC method allows sampling from a large class of distributions and can overcome the limitations of rejection sampling and importance sampling in high dimensional spaces.

The Metropolis-Hastings (MH) method is the most popular MCMC method [42]. The MH computational steps for approximating a target distribution $p(y_k)$ from a

Algorithm 1 Metropolis-Hastings (MH) Algorithm

Input: Target distribution $p(y_k)$, proposal distribution $q(x_k)$, initial sample x_0

for $k = 0$ to K **do**

Draw $y_k \sim q(\cdot \mid x_k)$

Compute $\alpha = \dfrac{p(y_k)\, q(x_k \mid y_k)}{p(x_k)\, q(x_k \mid y_k)}$

if $\alpha > 1$ **then**

$x_{k+1} = x_k$

else if $\alpha \leq 1$ **then**

Draw $\epsilon \sim \mathcal{U}(0, 1)$, where $\mathcal{U}(a, b)$ is a uniform distribution between a and b

if $\epsilon \leq \alpha$ **then**

$x_{k+1} = y_k$

else

$x_{k+1} = x_k$

end if

end if

end for

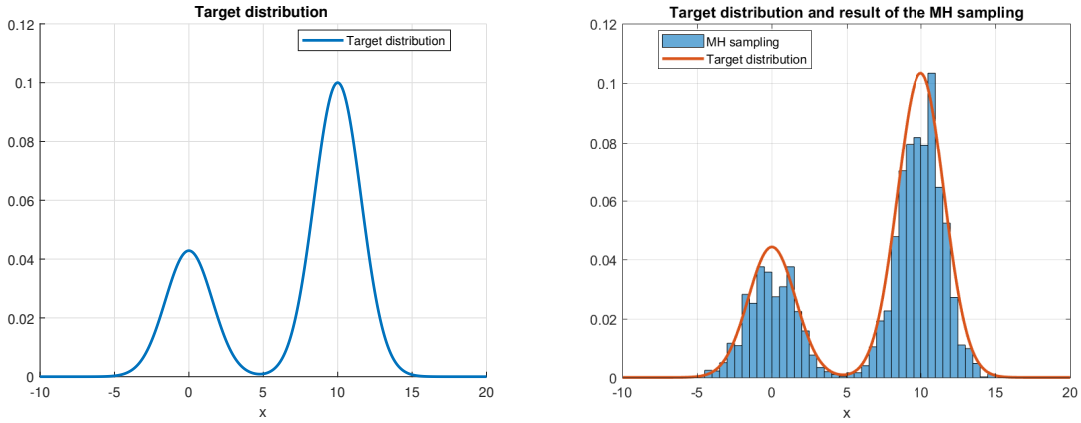
Output: MH samples $x_{1:K}$

proposal distribution $q(x_k)$ are provided in Algorithm 1. In Figure 2.3, we provide a simulation to demonstrate that the large number of 5,000 iterations closely approximate the target distribution.

2.2 Bayesian Inference for Dynamic Systems

2.2.1 Bayesian Estimation

Bayesian inference is a methodology based on Bayes' theorem that can be used to update the probability of a hypothesis as new information becomes available [43]. It is



(a) Target distribution

(b) Estimated distribution

Figure 2.3: MH Sampling to Estimate a Target Distribution Using 5,000 Iterations.

often combined with MCMC sampling to estimate the distribution of hidden parameters given some measurements. Especially for parameters that do not vary with time, maximum likelihood estimation (MLE) is used to estimate a parameter θ given the set of measurements $\mathbf{z}_{1:k} = \{\mathbf{z}_1, \dots \mathbf{z}_k\}$, up to time step k. This involves maximizing the likelihood function $p(\mathbf{z}_{1:k}; \theta)$ over all possible θ values [44]. Bayesian estimation can also be used, assuming some prior knowledge on the parameter θ. Specifically, using Bayes' theorem, the posterior PDF of parameter θ given the measurements can be given by

$$p(\theta \mid \mathbf{z}_{1:k}) = \frac{p(\mathbf{z}_{1:k} \mid \theta)\, p(\theta)}{p(\mathbf{z}_{1:k})} = \frac{p(\mathbf{z}_{1:k} \mid \theta)\, p(\theta)}{\int p(\mathbf{z}_{1:k} \mid \theta)\, p(\theta)\, d\theta} \tag{2.1}$$

where $p(\mathbf{z}_{1:k} \mid \theta)$ is the conditional likelihood PDF and $p(\theta)$ is the assumed known prior PDF of the parameter. This method, referred to as maximum a posteriori estimation (MAP), provides an estimate of θ by maximizing the posterior PDF in Equation (2.1) over all possible θ values. If the measurements are independent, then

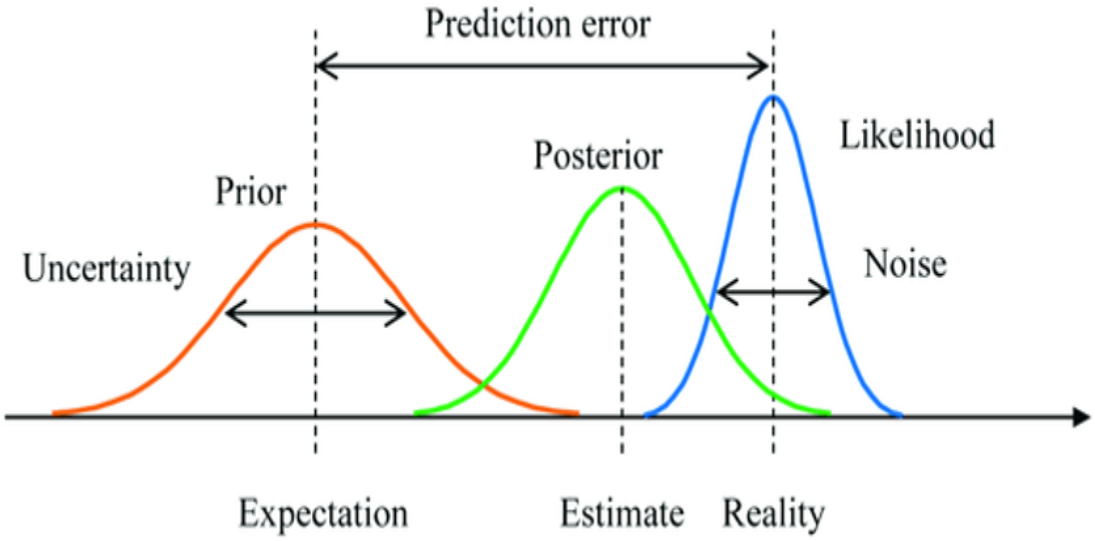


Figure 2.4: Bayesian Estimation.

the posterior PDF can also be written as

$$p(\theta \mid \mathbf{z}_{1:k}) \propto p(\theta) \prod_{l=1}^{k} p(\mathbf{z}_l \mid \theta)$$

As depicted in Figure 2.4, Bayesian estimation involves obtaining an estimate of some unknown parameter using the prior PDF and the likelihood function. If the prior PDF is not available, the estimate of the parameter can be first predicted using some available knowledge. Then estimate is updated using the available measurements. shows the Bayesian inference from the prior, likelihood and posterior distribution. Specifically, the posterior PDF is computed by updating the prior belief using the likelihood. Note that the wider the spread or variance of the prior PDF, the higher the uncertainty and the harder it is to correctly make a prediction.

2.2.2 Dynamic State Space Representation

In a dynamic system, as unknown parameters vary with time, they have to be estimated at each time step. If the dynamic system can be represented in a state space

formulation, then Bayesian recursion can be used to estimate the unknown state parameter. The estimate is obtained recursively, at each time step, using a prediction step and an update step with the measurements provided.

We consider a dynamic system with an unknown state parameter vector $\mathbf{x}_k$ and measurement vector $\mathbf{z}_k$ at each time step k, $k = 1, \ldots, K$. The state space representation of the system is given by

$$\mathbf{x}_k = g(\mathbf{x}_{k-1}) + \mathbf{q}_{k-1} \tag{2.2}$$

$$\mathbf{z}_k = h(\mathbf{x}_k) + \mathbf{r}_k \tag{2.3}$$

The state transition equation in (2.2) originates from some known physics-based model that relates the state at time step $k-1$ to the state $\mathbf{x}_k$ at time step k. This model is described using the function $g(\cdot)$, and the random process $\mathbf{q}_k$ is used to account for possible modeling errors. Equation (2.3) is the measurement equation that provides the relationship between the state $\mathbf{x}_k$ and the measurement $\mathbf{z}_k$ at time step k. The random process $\mathbf{r}_k$ is measurement noise at time step k. It is often assumed that the state $\mathbf{x}_k$ only depends on the previous state, following the first order Markov process assumption. Note that state and measurement vectors can have different dimensionality. Using Bayesian estimation, the posterior PDF of all the states, given all the measurements can be given by

$$p(\mathbf{x}_{0:k} \mid \mathbf{z}_{1:k}) \propto p(\mathbf{z}_k \mid \mathbf{x}_k)\, p(\mathbf{x}_k \mid \mathbf{x}_{k-1})\, p(\mathbf{x}_{0:k-1} \mid \mathbf{z}_{1:k-1}) \tag{2.4}$$

where $\mathbf{x}_{0:k} = \{\mathbf{x}_0,\ \mathbf{x}_1,\ \ldots,\ \mathbf{x}_k\}$. Equation (2.4) shows that the posterior density can be derived recursively assuming knowledge of the initial prior PDF $p(\mathbf{x}_0)$.

2.3 Bayesian Filtering

Bayesian filtering involves the estimation of the unknown state $\mathbf{x}_k$ using the estimated posterior PDF $p(\mathbf{x}_k \mid \mathbf{z}_k)$ [7, 45]. The posterior is estimated recursively at each time

step k. The recursion has two main steps. The prediction step uses the transition model in (2.2) to obtain an estimate of $\mathbf{x}_k$ from the prior PDF $p(\mathbf{x}_k \mid \mathbf{x}_{k-1})$. The update step uses the likelihood function $p(\mathbf{z}_k \mid \mathbf{x}_k)$ in Equation in (2.3) to improve the estimate.

Bayesian filtering is based on the Markovian assumption that the state $\mathbf{x}_k$ at time step k depends only on previous state $\mathbf{x}_{k-1}$. Specifically, the current state $\mathbf{x}_k$ is conditionally independent of earlier states given previous state $\mathbf{x}_{k-1}$,

$$p(\mathbf{x}_k \mid \mathbf{x}_{0:k-1}) = p(\mathbf{x}_k \mid \mathbf{x}_{k-1})$$

The following two assumptions are also made. It is assumed that the current state $\mathbf{x}_k$ is conditionally independent of past measurements $\mathbf{z}_{1:k-1}$ given the past state $\mathbf{x}_{k-1}$,

$$p(\mathbf{x}_k \mid \mathbf{x}_{k-1}, \mathbf{z}_{1:k-1}) = p(\mathbf{x}_k \mid \mathbf{x}_{k-1}) \tag{2.5}$$

It is also assumed that the current measurement $\mathbf{z}_k$ is conditionally independent of past measurements, $\mathbf{z}_{1:k-1}$, given the current state $\mathbf{x}_k$

$$p(\mathbf{z}_k \mid \mathbf{x}_k, \mathbf{z}_{1:k-1}) = p(\mathbf{z}_k \mid \mathbf{x}_k) \tag{2.6}$$

The recursive implementation of Bayesian filtering starts with some initial PDF $p(\mathbf{x}_0)$. After time step $k-1$, we assume that the posterior PDF $p(\mathbf{x}_{k-1} \mid \mathbf{z}_{1:k-1})$ was recursively obtained. At time step k, the system model is used to predict forward from $p(\mathbf{x}_{k-1} \mid \mathbf{z}_{1:k-1})$ to $p(\mathbf{x}_k \mid \mathbf{z}_{1:k-1})$ using the Chapman-Kolmogorov equation [7],

$$\begin{aligned} p\big(\mathbf{x}_k \mid \mathbf{z}_{1:k-1}\big) &= \int p\big(\mathbf{x}_k \mid \mathbf{x}_{k-1}, \mathbf{z}_{1:k-1}\big)\, p\big(\mathbf{x}_{k-1} \mid \mathbf{z}_{1:k-1}\big)\, d\,\mathbf{x}_{k-1} \\ &= \int p\big(\mathbf{x}_k \mid \mathbf{x}_{k-1}\big)\, p\big(\mathbf{x}_{k-1} \mid \mathbf{z}_{1:k-1}\big)\, d\,\mathbf{x}_{k-1} \end{aligned} \tag{2.7}$$

where we used $p\big(\mathbf{x}_k \mid \mathbf{x}_{k-1}, \mathbf{z}_{1:k-1}\big) = p\big(\mathbf{x}_k \mid \mathbf{x}_{k-1}\big)$ from Equation (2.5). When the

new measurement $\mathbf{z}_k$ is available, then the estimate is updated as

$$p(\mathbf{x}_k \mid \mathbf{z}_{1:k}) = \frac{p(\mathbf{z}_k \mid \mathbf{x}_k, \mathbf{z}_{1:k-1})\, p(\mathbf{x}_k \mid \mathbf{z}_{1:k-1})}{p(\mathbf{z}_k \mid \mathbf{z}_{1:k-1})}$$

$$= \frac{p(\mathbf{z}_k \mid \mathbf{x}_k)\, p(\mathbf{x}_k \mid \mathbf{z}_{1:k-1})}{p(\mathbf{z}_k \mid \mathbf{z}_{1:k-1})}$$

where we used $p(\mathbf{z}_k \mid \mathbf{x}_k, \mathbf{z}_{1:k-1}) = p(\mathbf{z}_k \mid \mathbf{x}_k)$ from Equation (2.6), and

$$p(\mathbf{z}_k \mid \mathbf{z}_{1:k-1}) = \int p(\mathbf{z}_k \mid \mathbf{x}_k)\, p\big(\mathbf{x}_k \mid \mathbf{z}_{1:k-1}\big) d\mathbf{x}_k \,.$$

2.4 Kalman Filtering

2.4.1 Linear Dynamic System Representation

For many dynamic systems, the prediction step of Bayesian filtering in Equation (2.7) is difficult to compute. For systems that have linear prior and measurement relations and assume Gaussian random processes, the equation can be iteratively solved in closed form using the Kalman filter (KF) [4, 5, 7]. The KF has been widely used as an optimal solution in many tracking applications, though it is limited by the linearity and Gaussian assumptions.

For the KF, the state space representation in Equations (2.2) and (2.3) can be written using linear functions or in matrix form. Specifically,

$$\mathbf{x}_k = \mathbf{F}\,\mathbf{x}_{k-1} + \mathbf{q}_{k-1} \tag{2.8}$$

$$\mathbf{z}_k = \mathbf{H}\,\mathbf{x}_k + \mathbf{r}_k \tag{2.9}$$

If $\mathbf{x}_k$ is the $N \times 1$ state vector and $\mathbf{z}_k$ is the $M \times 1$ measurement vector, then $\mathbf{F}$ is the $N \times N$ state transition matrix and $\mathbf{H}$ is the $M \times N$ measurement matrix. For the KF, the random processes $\mathbf{q}_k$ and $\mathbf{r}_k$ are both assumed zero-mean Gaussian with

corresponding covariance matrices $E[\mathbf{q}_k\,\mathbf{q}_k^\top] = \mathbf{Q}_k$ and $E[\mathbf{r}_k\,\mathbf{r}_k^\top] = \mathbf{R}_k$, respectively. Here, $E[\cdot]$ denotes statistical expectation and $\mathbf{q}_k^\top$ is the vector transpose of $\mathbf{q}_k$.

2.4.2 Derivation of Recursive Kalman Filter

The KF can be implemented recursively in closed form as derived next in detail. We first assume that the state estimate $\hat{\mathbf{x}}_{k-1} = \hat{\mathbf{x}}_{k-1|k-1}$ was recursively obtained at time step $(k-1)$. Using this estimate, we can compute the $N \times N$ error covariance matrix as the correlation matrix $\mathbf{P}_{k-1|k-1} = E[\mathbf{e}_{k-1|k-1}\mathbf{e}_{k-1|k-1}^\top]$, where $\mathbf{e}_{k-1|k-1} = \mathbf{x}_{k-1} - \hat{\mathbf{x}}_{k-1|k-1}$ is the state parameter error at time step $(k-1)$. Note that $\mathbf{x}_{k-1}$ is assumed to be the true state parameter.

At time step k, we first use Equation (2.8) to predict a new estimate given by

$$\hat{\mathbf{x}}_{k|k-1} = \mathbf{F}\,\hat{\mathbf{x}}_{k-1|k-1}\,. \tag{2.10}$$

The predicted error covariance matrix

$$\mathbf{P}_{k|k-1} = E\left[(\mathbf{x}_k - \hat{\mathbf{x}}_{k|k-1})\,(\mathbf{x}_k - \hat{\mathbf{x}}_{k|k-1})^\top\right]$$

can be obtained using $\mathbf{x}_k = \mathbf{F}\mathbf{x}_{k-1} + \mathbf{q}_{k-1}$ in Equation (2.8) and $\hat{\mathbf{x}}_{k|k-1} = \mathbf{F}\,\hat{\mathbf{x}}_{k-1|k-1}$ in Equation (2.10) to obtain

$$\begin{aligned}
\mathbf{P}_{k|k-1} &= E[(\mathbf{x}_k - \hat{\mathbf{x}}_{k|k-1})\,(\mathbf{x}_k - \hat{\mathbf{x}}_{k|k-1})^\top] \\
&= E\left[\left(\mathbf{F}\mathbf{x}_{k-1} + \mathbf{q}_{k-1} - \mathbf{F}\,\hat{\mathbf{x}}_{k-1|k-1}\right)\left(\mathbf{F}\mathbf{x}_{k-1} + \mathbf{q}_{k-1} - \mathbf{F}\,\hat{\mathbf{x}}_{k-1|k-1}\right)^\top\right] \\
&= E\left[\left(\mathbf{F}(\mathbf{x}_{k-1} - \hat{\mathbf{x}}_{k-1|k-1}) + \mathbf{q}_{k-1}\right)\left(\mathbf{F}(\mathbf{x}_{k-1} - \hat{\mathbf{x}}_{k-1|k-1}) + \mathbf{q}_{k-1}\right)^\top\right] \\
&= \mathbf{F}\,\mathbf{P}_{k-1|k-1}\,\mathbf{F}^\top + \mathbf{Q}_{k-1}
\end{aligned} \tag{2.11}$$

Using the measurement $\mathbf{z}_k$ at time step k, the updated state estimate $\hat{\mathbf{x}}_{k|k}$ can be viewed as the predicted estimate $\hat{\mathbf{x}}_{k|k-1}$ in Equation (2.10) with an update term extracted from $\mathbf{z}_k$. Specifically,

$$\hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k \left(\mathbf{z}_k - \mathbf{H}\,\hat{\mathbf{x}}_{k|k-1}\right) \tag{2.12}$$

This update term is obtained from the measurement equation in (2.9) as the residual measurement (or innovation)

$$\mathbf{b}_{k|k-1} = \mathbf{z}_k - \mathbf{H}_k\,\hat{\mathbf{x}}_{k|k-1} \tag{2.13}$$

The update term is weighted by some factor $\mathbf{K}_k$, which is derived next as the Kalman gain. The error covariance at time step k is given by

$$\mathbf{P}_{k|k} = E[\mathbf{e}_{k|k}\,\mathbf{e}_{k|k}^\top] = E[(\mathbf{x}_k - \hat{\mathbf{x}}_{k|k})\,(\mathbf{x}_k - \hat{\mathbf{x}}_{k|k})^\top] \tag{2.14}$$

If we replace $\hat{\mathbf{x}}_{k|k}$ from Equation (2.12) in the covariance matrix, then

$$\begin{aligned}\mathbf{P}_{k|k} = E\Big[\ &\left[(\mathbf{I} - \mathbf{K}_k\,\mathbf{H})(\mathbf{x}_k - \hat{\mathbf{x}}_{k|k-1}) - \mathbf{K}_k\,\mathbf{r}_{k|k-1}\right] \\ &\left[(\mathbf{I} - \mathbf{K}_k\,\mathbf{H})(\mathbf{x}_k - \hat{\mathbf{x}}_{k|k-1}) - \mathbf{K}_k\,\mathbf{r}_{k|k-1}\right]^\top\Big]\end{aligned} \tag{2.15}$$

where $\mathbf{I}$ is the identify matrix. Using $E[\mathbf{r}_{k|k-1}\,\mathbf{r}_{k|k-1}^\top] = \mathbf{R}_k$ and expectation properties, Equation (2.15) can be written as

$$\begin{aligned}\mathbf{P}_{k|k} &= (\mathbf{I} - \mathbf{K}_k\,\mathbf{H})\ E\left[(\mathbf{x}_k - \hat{\mathbf{x}}_{k|k-1})\,(\mathbf{x}_k - \hat{\mathbf{x}}_{k|k-1})^\top\right]\ (\mathbf{I} - \mathbf{K}_k\,\mathbf{H})^\top + \mathbf{K}_k\,\mathbf{R}_k\,\mathbf{K}_k^\top \\ &= (\mathbf{I} - \mathbf{K}_k\,\mathbf{H})\ \mathbf{P}_{k|k-1}\ (\mathbf{I} - \mathbf{K}_k\,\mathbf{H})^\top + \mathbf{K}_k\,\mathbf{R}_k\,\mathbf{K}_k^\top \\ &= \mathbf{P}_{k|k-1} - \mathbf{K}_k\,\mathbf{H}\,\mathbf{P}_{k|k-1} - \mathbf{P}_{k|k-1}\,\mathbf{H}^\top\,\mathbf{K}_k^\top + \mathbf{K}_k\,\mathbf{H}\,\mathbf{P}_{k|k-1}\,\mathbf{H}^\top\mathbf{K}_k^\top + \mathbf{K}_k\,\mathbf{R}_k\,\mathbf{K}_k^\top \\ &= \mathbf{P}_{k|k-1} - 2\,\mathbf{K}_k\,\mathbf{H}\,\mathbf{P}_{k|k-1} + \mathbf{K}_k\,(\mathbf{H}\,\mathbf{P}_{k|k-1}\,\mathbf{H}^\top + \mathbf{R}_k)\,\mathbf{K}_k^\top\end{aligned} \tag{2.16}$$

where $\mathbf{P}_{k|k-1}$ is given in (2.11) and $\mathbf{K}_k\,\mathbf{H}\,\mathbf{P}_{k|k-1} = \mathbf{P}_{k|k-1}^\top\,\mathbf{H}^\top\,\mathbf{K}_k^\top = \mathbf{P}_{k|k-1}\,\mathbf{H}^\top\,\mathbf{K}_k^\top$, using properties of a covariance matrix.

To optimize the estimation accuracy, we want to find the Kalman gain $\mathbf{K}_k$ that minimizes the mean-squared error, MSE_k. We can obtain the MSE as the trace $T[\mathbf{P}_{k|k}]$ of the error covariance matrix in Equation (2.16). Thus,

$$\begin{aligned}\text{MSE}_k &= T[\mathbf{P}_{k|k}] \\ &= T[\mathbf{P}_{k|k-1}] - 2\,T[\mathbf{K}_k\,\mathbf{H}\,\mathbf{P}_{k|k-1}] + T\left[\mathbf{K}_k\,(\mathbf{H}\,\mathbf{P}_{k|k-1}\,\mathbf{H}^\top + \mathbf{R}_k)\,\mathbf{K}_k^\top\right] \quad (2.17)\end{aligned}$$

To minimize the MSE, we take the derivative of Equation (2.17) with respect to $\mathbf{K}_k$, set the derivative to zero and solve for $\mathbf{K}_k$. Considering the property of first order derivatives of the trace of a matrix

$$\frac{d}{d\mathbf{A}}\,T[\mathbf{A}\mathbf{B}] = \mathbf{B}^\top$$

for matrices $\mathbf{A}$ and $\mathbf{B}$, we can obtain

$$\frac{d}{d\,\mathbf{K}_k}\,T[\mathbf{P}_{k|k}] = -2\,\mathbf{P}_{k|k-1}^\top\,\mathbf{H}^\top + 2\,\mathbf{K}_k(\mathbf{H}\,\mathbf{P}_{k|k-1}\,\mathbf{H}^\top + \mathbf{R}_k) = 0$$

The solution of the Kalman gain is thus given by

$$\mathbf{K}_k = \mathbf{P}_{k|k-1}\,\mathbf{H}^\top\,(\mathbf{H}\,\mathbf{P}_{k|k-1}\,\mathbf{H}^\top + \mathbf{R}_k)^{-1} \quad (2.18)$$

If we define the innovation covariance as

$$\mathbf{S}_k = \mathbf{H}\,\mathbf{P}_{k|k-1}\,\mathbf{H}^\top + \mathbf{R}_k \quad (2.19)$$

then Equation (2.18) becomes

$$\mathbf{K}_k = \mathbf{P}_{k|k-1}\,\mathbf{H}^\top\,\mathbf{S}_k^{-1} \quad (2.20)$$

If we substitute $\mathbf{K}_k$ from Equation (2.18) in Equation (2.16), then the error covariance

Algorithm 2 Kalman Filtering (KF)

Input: Initial parameter state $\hat{\mathbf{x}}_{0|0}$, sequential measurements $\mathbf{z}_k$, $k = 1, \ldots, K$ state space model in Equations (2.8) and (2.9)

for $k = 1$ to K **do**

Use Equation (2.10) to predict estimate $\hat{\mathbf{x}}_{k|k-1}$

Use Equation (2.11) to predict error covariance matrix $\mathbf{P}_{k|k-1}$

Obtain innovation $(\mathbf{z}_k - \mathbf{H}_k\,\hat{\mathbf{x}}_{k|k-1})$ in Equation (2.13)

Use Equation (2.19) to compute innovation covariance matrix $\mathbf{S}_k$

Use Equation (2.20) to compute Kalman gain $\mathbf{K}_k$

Use Equation (2.12) to update state estimate $\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_{k|k}$

Use Equation (2.21) to update error covariance $\mathbf{P}_k = \mathbf{P}_{k|k}$

end for

Output: State estimate $\hat{\mathbf{x}}_k$ and error covariance matrix $\mathbf{P}_k$

matrix can be written as

$$\begin{aligned}\mathbf{P}_{k|k} &= \mathbf{P}_{k|k-1} - 2\,\mathbf{K}_k\,\mathbf{H}\,\mathbf{P}_{k|k-1} + \mathbf{K}_k\,\mathbf{S}_k\,\mathbf{K}^\top \\ &= \mathbf{P}_{k|k-1} - \mathbf{P}_{k|k-1}\,\mathbf{H}^\top\,\mathbf{S}_k^{-1}\,\mathbf{H}\,\mathbf{P}_{k|k-1} \\ &= \mathbf{P}_{k|k-1} - \mathbf{K}_k\,\mathbf{H}\,\mathbf{P}_{k|k-1} \\ &= (\mathbf{I} - \mathbf{K}_k\,\mathbf{H})\,\mathbf{P}_{k|k-1} \qquad (2.21)\end{aligned}$$

In summary, the KF recursion steps are summarized in Algorithm 2 and also depicted in Figure 2.5.

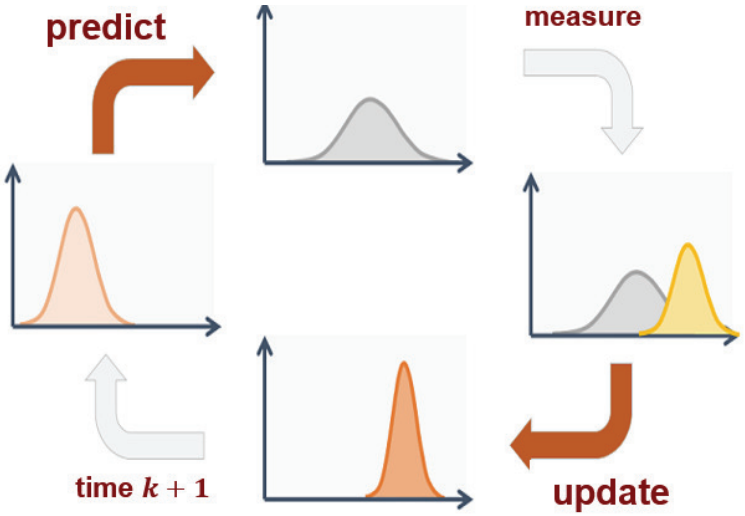


Figure 2.5: Depiction of Recursive Kalman Filtering.

2.5 Extended Kalman Filtering

The extended Kalman Filter (EKF) offers an alternative to the KF when the dynamic system is nonlinear, which is often the case in many applications. We consider the state space representation in Equations (2.2) and (2.3),

$$\mathbf{x}_k = g(\mathbf{x}_{k-1}) + \mathbf{q}_{k-1}$$

$$\mathbf{z}_k = h(\mathbf{x}_k) + \mathbf{r}_k$$

where we assume that $g(\mathbf{x}_k)$ and $h(\mathbf{x}_k)$ are nonlinear functions. The EKF uses the Taylor series expansion and Jacobian matrix to linearize the functions around the Gaussian's mean of the estimated state. The Jacobian matrix that is a matrix of its first-order partial derivatives of a multivariate function was introduced to describe the best linearized approximation of the change of $\mathbf{f}$ around $\mathbf{x}$ [46, 47]. Consider a function , where $\mathbf{f} : \mathbb{R}^n \to \mathbb{R}^m$ can be described the Jacobian as follows.

$$\mathbf{J_f} = \begin{bmatrix} \frac{\partial \mathbf{f}}{\partial x_1} & \cdots & \frac{\partial \mathbf{f}}{\partial x_n} \end{bmatrix} = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_m}{\partial x_1} & \cdots & \frac{\partial f_m}{\partial x_n} \end{bmatrix}$$

In particular, using Taylor series expansion, we expand the two functions and only keep the first two terms to obtain

$$g(\mathbf{x}_{k-1}) = g(\hat{\mathbf{x}}_{k-1|k-1}) + \mathbf{J}_g(\hat{\mathbf{x}}_{k-1|k-1})\,(\mathbf{x}_{k-1} - \hat{\mathbf{x}}_{k-1|k-1})$$

$$h(\mathbf{x}_k) = h(\hat{\mathbf{x}}_{k|k-1}) + \mathbf{J}_h(\hat{\mathbf{x}}_{k|k-1})\,(\mathbf{x}_k - \hat{\mathbf{x}}_{k|k-1})$$

It is assumed that the contribution of higher orders in the expansions is negligible. After linearization of the functions, we can proceed with the KF steps. The updated estimate and error covariance matrix are given by

$$\hat{\mathbf{x}}_{k|k-1} \approx g(\hat{\mathbf{x}}_{k-1|k-1})$$

$$\mathbf{P}_{k|k-1} = \mathbf{J}_g(\hat{\mathbf{x}}_{k|k-1})\,\mathbf{P}_{k-1|k-1}\,\mathbf{J}_g^\top(\hat{\mathbf{x}}_{k|k-1}) + \mathbf{Q}_{k-1},$$

the update step is also derived as follows:

$$\hat{\mathbf{x}}_{k|k} \approx \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k\left(\mathbf{z}_k - h(\hat{\mathbf{x}}_{k|k-1})\right)$$

$$\mathbf{K}_k = \mathbf{P}_{k|k-1}\,\mathbf{J}_h^\top(\hat{\mathbf{x}}_{k|k-1})\Big(\mathbf{J}_h(\hat{\mathbf{x}}_{k|k-1})\,\mathbf{P}_{k|k-1}\mathbf{J}_h^\top(\hat{\mathbf{x}}_{k|k-1}) + \mathbf{R}_k\Big)^{-1}$$

$$\mathbf{P}_{k|k} = \big(\mathbf{I} - \mathbf{K}_k\,\mathbf{J}_h(\hat{\mathbf{x}}_{k|k-1})\big)\,\mathbf{P}_{k|k-1}.$$

The derivation of the EKF is similar to that of the KF; it can be found in [48].

2.6 Unscented Kalman Filtering

2.6.1 Unscented Transform

The unscented transform (UT) is a method used to estimate the distribution of a random vector that undergoes a nonlinear transformation. Instead of applying

the nonlinear transformation to the random vector, it uses the nonlinear function to approximate the distribution of the transformed random vector [4, 27, 49]. In particular, the UT can be used to estimate the joint PDF of random vectors $\mathbf{x}$ and $\mathbf{z}$, where $\mathbf{x}$ has a Gaussian distribution with mean $\hat{\mathbf{x}}$ and covariance matrix $\mathbf{P}$, and $\mathbf{z} = g(\mathbf{x})$ is related to $\mathbf{x}$ via the nonlinear function $g(\cdot)$. Specifically,

$$\mathbf{z} = g(\mathbf{x}), \quad \text{for} \quad \mathbf{x} \sim \mathcal{N}(\hat{\mathbf{x}}, \mathbf{P})$$

The UT directly approximates the mean and covariance of the target distribution using a finite number of sigma points to capture the mean and covariance of the original distribution [4]. Assuming $2\,n + 1$ sigma points $\mathcal{X}^{(i)}$, $i = 0, 1, \ldots 2\,n + 1$, for random vector $\mathbf{x}$, then each sigma point is propagated through the nonlinear function obtain the corresponding sigma points for $\mathbf{z}$ [4]. In particular,

$$\mathcal{Z}^{(i)} = g(\mathcal{X}^{(i)}), \quad i = 0, \ldots, 2\,n,$$

The UT selects sigma points $\mathcal{X}$ differently from Monte Carlo estimation. The points are obtained as

$$\begin{aligned}
\mathcal{X}^{(0)} &= \hat{\mathbf{x}} \\
\mathcal{X}^{(i)} &= \hat{\mathbf{x}} + \sqrt{n + \lambda}\, \left[\sqrt{\mathbf{P}}\right]_i \\
\mathcal{X}^{(i+n)} &= \hat{\mathbf{x}} - \sqrt{n + \lambda}\, \left[\sqrt{\mathbf{P}}\right]_i, \quad i = 1, \ldots, n
\end{aligned}$$

where $\hat{\mathbf{x}}$ and $\mathbf{P}$ are the mean and covariance of $\mathbf{x} \sim \mathcal{N}(\hat{\mathbf{x}}, \mathbf{P})$, the matrix square root is such that $\sqrt{\mathbf{P}}\sqrt{\mathbf{P}^\top} = \mathbf{P}$ [10], and $[\mathbf{P}]_i$ denotes the ith column of matrix $\mathbf{P}$. The UT two UT parameters, α and κ are used to determine the scaling parameter λ as

$$\lambda = \alpha^2(n + \kappa) - n$$

Note that parameter α determines the spread of the sigma points around $\hat{\mathbf{x}}$ and is usually set to a small positive value, whereas parameter κ is a secondary scaling parameter which is usually set to 0 [4].

The mean and covariance of $\mathbf{z}$ are then approximated using the corresponding weighted mean and covariance of the posterior sigma points [10]. Specifically,

$$
\begin{aligned}
E[g(\mathbf{x})] &\simeq \hat{\mathbf{z}} = \sum_{i=0}^{2n} w_m^{(i)}\, \mathcal{Z}^{(i)} \\
\mathrm{Cov}(g(\mathbf{x})) &\simeq \mathbf{C}_{\mathcal{Z}} = \sum_{i=0}^{2\,n} w_c^{(i)} \Big(\mathcal{Z}^{(i)} - \hat{\mathbf{z}})(\mathcal{Z}^{(i)} - \hat{\mathbf{z}} \Big)^{\top}
\end{aligned}
$$

where the constant weights $w_m^{(i)}$ and $w_c^{(i)}$ are obtained as [10]

$$
\begin{aligned}
w_m^{(0)} &= \frac{\lambda}{n+\lambda} \\
w_c^{(0)} &= \frac{\lambda}{n+\lambda} + (1 - \alpha^2 + \beta) \\
w_m^{(i)} &= \frac{1}{2(n+\lambda)}, \quad i = 1,\ \ldots,\ 2\,n, \\
w_c^{(i)} &= \frac{1}{2(n+\lambda)}, \quad i = 1,\ \ldots,\ 2\,n,
\end{aligned}
\tag{2.22}
$$

and β is used to incorporate prior knowledge of the distribution of $\mathbf{x}$ [10]. The estimated cross-covariance of $\mathbf{x}$ and $\mathbf{z}$ is approximated as

$$
\mathbf{C}_{\mathcal{X},\mathcal{Z}} = \sum_{i=0}^{2\,n} w_c^{(i)} \Big(\mathcal{X}^{(i)} - \hat{\mathbf{x}})(\mathcal{Z}^{(i)} - \hat{\mathbf{z}} \Big)^{\top}
$$

The joint PDF of $\mathbf{x}$ and $\mathbf{z}$ can be approximated using concatenated pairs of sigma points $\mathcal{X}$ and $\mathcal{Z}$. Specifically,

$$
[\mathbf{x}\ \mathbf{z}]^{\top} \sim \mathcal{N}\Big([\hat{\mathbf{x}}\ \hat{\mathbf{z}}]^{\top}, \mathbf{C}_{\mathbf{x},\,\mathbf{z}} \Big)
$$

where

$$
\mathbf{C}_{\mathbf{x},\,\mathbf{z}} = \begin{bmatrix} \mathbf{P} & \mathbf{C}_{\mathcal{X},\,\mathcal{Z}} \\ \mathbf{C}_{\mathcal{X},\,\mathcal{Z}}^{\top} & \mathbf{C}_{\mathcal{Z}} \end{bmatrix}
$$

The UT method is summarized in in Algorithm 3 [4, 10].

Algorithm 3 Unscented Transform (UT)

Input: Mean and covariance at time k, $\hat{\mathbf{x}}_k$, $\mathbf{P}_k$, nonlinear function $g(\cdot)$

Form a set of $2n+1$ sigma points:

$$\begin{aligned}
\mathcal{X}_k^{(0)} &= \hat{\mathbf{x}}_k \\
\mathcal{X}_k^{(i)} &= \hat{\mathbf{x}}_k + \sqrt{n+\lambda}[\sqrt{\mathbf{P}}_k]_i, \\
\mathcal{X}_k^{(i+n)} &= \hat{\mathbf{x}}_k - \sqrt{n+\lambda}[\sqrt{\mathbf{P}}_k]_i, \quad i = 1, \ldots, n.
\end{aligned}$$

Assign weights that correspond to each sigma point:

$$\begin{aligned}
w_m^{(0)} &= \tfrac{\lambda}{n+\lambda}, \qquad w_c^{(0)} = \tfrac{\lambda}{n+\lambda} + (1 - \alpha^2 + \beta) \\
w_m^{(i)} &= \tfrac{\lambda}{2(n+\lambda)}, \quad w_c^{(i)} = \tfrac{\lambda}{2(n+\lambda)}, \qquad i = 1, \ldots, 2n
\end{aligned}$$

Transform the sigma points through the nonlinear function $\mathbf{g}(\cdot)$:

$$\mathcal{Z}_k^{(i)} = \mathbf{g}(\mathcal{X}_k^{(i)}), \quad i = 1, \ldots, n.$$

Compute the transformed mean and covariance from the sigma points:

$$\begin{aligned}
\hat{\mathbf{z}}_k &= \sum_{i=0}^{2n} w_m^{(i)} \mathcal{Z}_k^{(i)}, \\
\mathbf{C}_{\mathcal{Z}_k} &= \sum_{i=0}^{2n} w_c^{(i)} (\mathcal{Z}_k^{(i)} - \hat{\mathbf{z}}_k)(\mathcal{Z}_k^{(i)} - \hat{\mathbf{z}}_k)^\top \\
\mathbf{C}_{\mathcal{X}_k, \mathcal{Z}_k} &= \sum_{i=0}^{2n} w_c^{(i)} (\mathcal{X}_k^{(i)} - \hat{\mathbf{x}}_k)(\mathcal{Z}_k^{(i)} - \hat{\mathbf{x}}_k)^\top.
\end{aligned}$$

Output: The predicted mean, covariance and cross-covariance $\hat{\mathbf{z}}_k, \mathbf{C}_{\mathcal{Z}_k}, \mathbf{C}_{\mathcal{X}_k, \mathcal{Z}_k}$.

2.6.2 Unscented Kalman Filter Algorithm

The EKF performance is adequate for dynamic systems that are almost linear, but the performance deteriorates with the degree of nonlinearity. As an alternative to the EKF for nonlinear systems, the UT in Section 2.6.1 is applied to the KF, resulting in the unscented Kalman filter (UKF) [4, 49, 50]. The UKF assumes that the state parameter $\mathbf{x}$ is Gaussian with a known mean and covariance. The main algorithm is summarized next.

At time step $k-1$, we have the estimated state $\hat{\mathbf{x}}_{k-1|k-1}$ and its corresponding

error covariance matrix $\mathbf{P}_{k-1|k-1}$ from the previous iteration. During the prediction step, sigma points are first obtained using $\hat{\mathbf{x}}_{k-1|k-1}$ and $\mathbf{P}_{k-1|k-1}$ as

$$\mathcal{X}^{(0)}_{k-1|k-1} = \hat{\mathbf{x}}_{k-1|k-1}$$

$$\mathcal{X}^{(i)}_{k-1|k-1} = \hat{\mathbf{x}}_{k-1|k-1} + \sqrt{n+\lambda}\,\left[\sqrt{\mathbf{P}_{k-1|k-1}}\,\right]_i$$

$$\mathcal{X}^{(i+n)}_{k-1|k-1} = \hat{\mathbf{x}}_{k-1|k-1} - \sqrt{n+\lambda}\,\left[\sqrt{\mathbf{P}_{k-1|k-1}}\,\right]_i, \quad i = 1, \ldots, n$$

The sigma points are then propagated using the nonlinear transition function $g(\cdot)$ as

$$\mathcal{X}^{(i)}_{k|k-1} = g(\mathcal{X}^{(i)}_{k-1|k-1}), \quad i = 0, \ldots, 2\,n.$$

The predicted mean and covariance of the state can then be computed using

$$\hat{\mathbf{x}}_{k|k-1} = \sum_{i=0}^{2\,n} w_m^{(i)}\,\mathcal{X}^{(i)}_{k|k-1}$$

$$\mathbf{P}_{k|k-1} = \sum_{i=0}^{2\,n} w_c^{(i)}\left(\mathcal{X}^{(i)}_{k|k-1} - \hat{\mathbf{x}}_{k|k-1}\right)\left(\mathcal{X}^{(i)}_{k|k-1} - \hat{\mathbf{x}}_{k|k-1}\right)^{\top} + \mathbf{Q}_{k-1}$$

where the weights are defined in Equation (2.22).

For the innovation step, as in Equation (2.13) for the KF, the sigma points are first updated

$$\mathcal{X}^{(0)}_{k-1|k} = \hat{\mathbf{x}}_{k-1|k}$$

$$\mathcal{X}^{(i)}_{k-1|k} = \hat{\mathbf{x}}_{k-1|k} + \sqrt{n+\lambda}\,\left[\sqrt{\mathbf{P}_{k-1|k}}\,\right]_i$$

$$\mathcal{X}^{(i+n)}_{k-1|k} = \hat{\mathbf{x}}_{k-1|k} - \sqrt{n+\lambda}\,\left[\sqrt{\mathbf{P}_{k-1|k}}\,\right]_i, \quad i = 1, \ldots, n$$

and then propagate through the nonlinear measurement equation $h(\cdot)$ as

$$\mathcal{Z}^{(i)}_{k|k-1} = h(\mathcal{X}^{(i)}_{k|k-1}), \quad i = 0, \ldots,\ 2\,n$$

The corresponding mean and covariance for $\mathcal{Z}_{k|k-1}^{(i)}$ are found as

$$\hat{\mathbf{z}}_{k|k-1} = \sum_{i=0}^{2n} w_m^{(i)}\, \mathcal{Z}_{k|k-1}^{(i)}$$

$$\mathbf{C}_{\mathcal{Z}_{k|k-1}} = \sum_{i=0}^{2\,n} w_c^{(i)}\, \left(\mathcal{Z}_{k|k-1}^{(i)} - \hat{\mathbf{z}}_{k|k-1}\right) \left(\mathcal{Z}_{k|k-1}^{(i)} - \hat{\mathbf{z}}_{k|k-1}\right)^{\top} + \mathbf{R}_k$$

$$\mathbf{C}_{\mathcal{X}_{k|k-1},\,\mathcal{Z}_{k|k-1}} = \sum_{i=0}^{2\,n} w_c^{(i)} \left(\mathcal{X}_{k|k-1}^{(i)} - \hat{\mathbf{x}}_{k|k-1}\right) \left(\mathcal{Z}_{k|k-1}^{(i)} - \hat{\mathbf{z}}_{k|k-1}\right)^{\top},$$

where $\hat{\mathbf{z}}_{k|k-1}$ is the predicted mean, $\mathbf{C}_{\mathcal{Z}_{k|k-1}}$ is the predicted covariance of the measurement, and $\mathbf{C}_{\mathcal{X}_{k|k-1},\,\mathcal{Z}_{k|k-1}}$ is the cross-covariance of the state and the measurement [4]. Using the measurement $\mathbf{z}_k$ at time step k, we compute the innovation term

$$\mathbf{b}_{k|k-1} = \mathbf{z}_k - \mathbf{H}\,\hat{\mathbf{x}}_{k|k-1} = \mathbf{z}_k - \hat{\mathbf{z}}_{k|k-1}$$

and obtain the Kalman gain as

$$\mathbf{K}_k = \mathbf{C}_{\mathcal{X}_{k|k-1},\,\mathcal{Z}_{k|k-1}} \left(\mathbf{C}_{\mathcal{Z}_{k|k-1}}\right)^{-1}.$$

Using these two terms, we can compute the updated state estimate as

$$\hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k\, \left(\mathbf{z}_k - \mathbf{H}\,\hat{\mathbf{x}}_{k|k-1}\right)$$

and the updated error covariance as

$$\mathbf{P}_{k|k} = \mathbf{P}_{k|k-1} - \mathbf{K}_k\, \mathbf{C}_{\mathcal{Z}_{k|k-1}}\, \mathbf{K}_k^{\top}.$$

The steps of the UKF method are provided in Algorithm 4, 3 [4, 10]. Note that the UT algorithm parameters α, which determines the spread of the sigma points around the mean, β, which incorporates prior information, and λ, a scaling (spreading) parameter, are selected based on the application.

Algorithm 4 Unscented Kalman Filter (UKF) Algorithm

Input: Mean and covariance $\hat{\mathbf{x}}_0$, $\mathbf{P}_0$, measurements $\mathbf{z}_{1:K}$, dynamic system equations, covariance of process and measurement noise $\mathbf{Q}_k$ and $\mathbf{R}_k$

for $k = 1$ to K **do**

Prediction Step:

$$\{\hat{\mathbf{x}}_{k|k-1}\mathbf{P}_{k|k-1}\} = \texttt{UT}\Big(\hat{\mathbf{x}}_{k-1}, \mathbf{P}_{k-1}, g(\cdot), \mathbf{Q}_{k-1}, n, \lambda\Big) \text{ from Algorithm 3}$$

$$\hat{\mathbf{x}}_{k|k-1} = \sum_{i=0}^{2n} w_m^{(i)}\, \mathcal{X}_{k|k-1}^{(i)}$$

$$\mathbf{P}_{k|k-1} = \sum_{i=0}^{2n} w_c^{(i)} (\mathcal{X}_{k|k-1}^{(i)} - \hat{\mathbf{x}}_{k|k-1})\, (\mathcal{X}_{k|k-1}^{(i)} - \hat{\mathbf{x}}_{k|k-1})^\top + \mathbf{Q}_{k-1}$$

Update Step:

$$\{\hat{\mathbf{x}}_{k|k}, \mathbf{P}_{k|k}, \mathbf{K}_k\} = \texttt{UT}\Big(\hat{\mathbf{x}}_{k|k-1}, \mathbf{z}_k, \mathbf{P}_{k|k-1}, h(\cdot), \mathbf{R}_k, n, \lambda\Big)$$

$$\hat{\mathbf{z}}_{k|k-1} = \sum_{i=0}^{2n} w_m^{(i)}\, \mathcal{Z}_{k|k-1}^{(i)}$$

$$\mathbf{C}_{\mathcal{Z}_k} = \sum_{i=0}^{2n} w_c^{(i)}\, (\mathcal{Z}_{k|k-1}^{(i)} - \hat{\mathbf{z}}_{k|k-1})\, (\mathcal{Z}_{k|k-1}^{(i)} - \hat{\mathbf{z}}_{k|k-1})^\top + \mathbf{R}_k$$

$$\mathbf{C}_{\mathcal{X}_k,\mathcal{Z}_k} = \sum_{i=0}^{2n} w_c^{(i)} (\mathcal{X}_{k|k-1}^{(i)} - \hat{\mathbf{x}}_{k|k-1})(\mathcal{Z}_{k|k-1}^{(i)} - \hat{\mathbf{z}}_{k|k-1})^\top$$

$$\mathbf{K}_k = \mathbf{C}_{\mathcal{X}_k,\mathcal{Z}_k} \mathbf{C}_{\mathcal{Z}_k}^{-1}$$

$$\hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k(\mathbf{z}_k - \hat{\mathbf{z}}_{k|k-1})$$

$$\mathbf{P}_{k|k} = \mathbf{P}_{k|k-1} - \mathbf{K}_k\, \mathbf{C}_{\mathcal{Z}_k}\, \mathbf{K}_k^\top$$

end for

Output: Filtered Kalman gain, mean and covariance $\{\hat{\mathbf{x}}_{k|k}, \mathbf{P}_{k|k}, \mathbf{K}_k\}_{k=1:K}$

2.7 Particle Filtering

The particle filter (PF) is based on the idea of approximating the PDF of the observations $\mathbf{z}_k$ given the state $\mathbf{x}_k$ using sequential Monte Carlo (SMC) method. If we can not solve the integrals required for a Bayesian recursive filter analytically, we represent the posterior probability composed of a set with randomly chosen weighted samples. In the previous description, a randomly chosen method follows the MC method. An increasing number of samples makes the prediction converge to the true PDF.

2.7.1 Sequential Importance Sampling

The sequential importance sampling (SIS) is a basic framework for most PF based algorithms. Usually we can not draw samples $\mathbf{x}_k^{(i)}$ from the target distribution $p(\cdot)$ directly. Assume we sample directly from a different importance function $q(\cdot)$. Our approximation is still correct if

$$w_{k-1}^{(i)} \propto \frac{p(\mathbf{x}_{0:k-1}^{(i)} \mid \mathbf{z}_{1:k-1})}{q(\mathbf{x}_{0:k-1}^{(i)} \mid \mathbf{z}_{1:k-1})}.$$

We can take the advantage of it by choosing $q(\cdot)$ we want. If the importance function is chosen to factorize such that

$$q(\mathbf{x}_{0:k} \mid \mathbf{z}_{1:k}) = q(\mathbf{x}_k \mid \mathbf{x}_{0:k-1}, \mathbf{z}_{1:k}) q(\mathbf{x}_{0:k-1} \mid \mathbf{z}_{1:k-1}),$$

then we can augment old particles $\mathbf{x}_{0:k-1}^i$ by $\mathbf{x}_k \sim q(\mathbf{x}_k \mid \mathbf{x}_{0:k-1}, \mathbf{z}_{1:k})$ to get new particles $\mathbf{x}_{0:k}^i$. We have the case of state space modeling where the weights are updated as follows:

$$w_k^{(i)} \propto \frac{p(\mathbf{z}_k \mid \mathbf{x}_k^{(i)}) p(\mathbf{x}_k^{(i)} \mid \mathbf{x}_{k-1}^{(i)})}{q(\mathbf{x}_k^{(i)} \mid \mathbf{x}_{0:k-1}^{(i)}, \mathbf{z}_{1:k})} w_{k-1}^{(i)}. \tag{2.23}$$

Furthermore, if $q(\mathbf{x}_k \mid \mathbf{x}_{0:k-1}, \mathbf{z}_{1:k}) = q(\mathbf{x}_k \mid \mathbf{x}_{k-1}, \mathbf{z}_{1:k})$ which means the importance function is only dependent on the last state and observations, then we do not need

to preserve trajectories $\mathbf{x}_{0:k-1}^{(i)}$ and observations $\mathbf{z}_{1:k-1}$.

$$p(\mathbf{x}_k \mid \mathbf{z}_{1:k}) \approx \sum_{i=i}^{N_s} w_k^{(i)} \delta(\mathbf{x}_k - \mathbf{x}_k^{(i)})$$

The SIS algorithm is provided in Algorithm 5 [4].

Algorithm 5 Sequential Importance Sampling (SIS) Algorithm

Input: Weighted set of particles $\{\mathbf{x}_{k-1}^{(i)}, w_{k-1}^{(i)}\}_{i=1}^{N_s}$, measurements $\mathbf{z}_k$, target and importance distributions $p(\cdot)$, $q(\cdot)$

Draw N_s samples from the importance distributions

$\mathbf{x}_0^{(i)} \sim p(\mathbf{x}_0)$, and set $w_0^{(i)} = 1/N \quad i = 1, \ldots, N.$

for $k = 1$ to K **do**

Draw $\mathbf{x}_k^{(i)} \sim q(\mathbf{x}_k \mid \mathbf{x}_{0:k-1}^{(i)}, \mathbf{z}_{1:k}), \quad i = 1, \ldots, N.$

Update weights according to Equation (2.23) and normalize the weights.

Compute estimate State $\hat{\mathbf{x}}_k = \sum_{i=1}^{N_s} w_k^{(i)} \mathbf{x}_k^{(i)}$

end for

Output: an updated weighted set of particles $\{\mathbf{x}_k^{(i)}, w_k^{(i)}\}_{i=1}^{N_s}$

Therefore, the SIS is a set of recursively propagated weighted points through sequentially received measurements [12]. However, the SIS has a degeneracy problem in that most particles have negligible weight, which means the weight is concentrated on a few particles only after a few iterations. The amount of degeneracy can be estimated based on the variance of weights [51]. The variance of the weights $\left\{w_k^{(i)}\right\}_{i=1}^{N_s}$ will increase with time k. The degeneracy problem can be solved with a method known as resampling, which is outlined next part.

Resampling

Resampling method can be reduced the effects of degeneracy. The basic concept of resampling is to get rid of particles that have small weights and to concentrate on particles with large weights [12]. In other word, the idea of the resampling procedure is to remove particles with very small weights and duplicate particles with large weights [4]. The resampling steps generate a new set of particles by drawing new N_s samples from the discrete distribution that each weight $w_k^{(i)}$ as the probability of obtaining the sample index. Next, we replace the old sample set with the newly received samples with all constant weight, $w_k^{(i)} = 1/N_s$ [4]. Even though the resampling method can reduce the effects of the degeneracy problem, there are still a number of potential problems with the resampling method. First, it limits the opportunity to parallelize because all particles must be combined [12]. Next, a problem is called sample impoverishment or weight degeneracy. This can happen if most of the weights are placed on a single particle and leads to a loss of diversity among the particles. Another problem is the propagated states are drawn from the prior distribution, $p(\mathbf{x}_k \mid \mathbf{x}_{k-1})$, without accounting for the next observation, z_k [13].

Sequential Importance Resampling

The SIS algorithm with a resampling step leads to sequential importance resampling (SIR) [26, 34, 52]. The SIR algorithm is usually referred to as the PF. The SIR works the same as the SIS method except that the resampling process happens at the time step if it is actually needed. The SIR method can solve the degeneracy problems by introducing a resampling step to eliminate samples with low importance ratios and multiply samples with high importance ratios. The SIS algorithm is described in Algorithm 6 [4]. In Algorithm 6, N_{eff} is the "effective" number of particles and we

resample if the variance of the particle weights is less then N_{eff} [53]. [4] introduces an example, where $N_{\text{eff}} = N/10$, to perform a resampling algorithm.

2.8 Bayesian Smoothing

Bayesian filtering methods, as discussed in Section 2.3, use measurements as they are sequentially observed to estimate the state parameter $\mathbf{x}_k$ at each time step k, $k = 1, \ldots, K$. Bayesian smoothing in dynamic systems, on the other hand, is a post processing method, as it assumes knowledge of all measurements $\mathbf{z}_{1:K}$ [4, 16–18, 20–22, 54, 55]. In particular, smoothing is used to improve the accuracy of the estimated state parameter $\mathbf{x}_k$ by approximating the posterior PDF using measurements observed after k [20]. It is considered an a posteriori estimation that uses measurements after time step k to improve the estimate at time k, assuming the estimates are not needed in real time.

A commonly used smoothing approach is the Rauch-Tung-Striebel (RTS) forward-backward algorithm [17]. It involves a first estimation pass that moves forward in time to obtain estimates $\mathbf{x}_{k+1|k}$ and $\mathbf{x}_{k+1|k+1}$, for $k = 1, \ldots, K$. It is then followed by a second estimation pass that goes backward in time to compute $\mathbf{x}_{k|K}$ for $k = 1, \ldots, K$.

Algorithm 6 Sequential Importance Resampling (SIR)

Input: Weighted set of particles $\{\mathbf{x}_{k-1}^{(i)}, w_{k-1}^{(i)}\}_{i=1}^{N_s}$, measurements $\mathbf{z}_k$, target and importance distribution $p(\cdot)$, $q(\cdot)$

$\{\mathbf{x}_k, w_k^{(j)}\}_{j=1}^{N_s} = \texttt{SIS}(\{\mathbf{x}_{k-1}^{(i)}, w_{k-1}^{(i)}\}_{i=1}^{N_s}, \mathbf{z}_k)$

if $\frac{1}{\sum_{i=1}^{N}(w_k^{(i)})^2} < N_{\text{eff}}$ **then**

$\texttt{resample}(\{\mathbf{x}_k^{(i)}, w_k^{(i)}\}_{i=1}^{N_s})$

end if

Output: a resampled weighted set of particles $\{\mathbf{x}_k^{(i)}, w_k^{(i)}\}_{i=1}^{N_s}$

Using the dynamic system representation in Equations (2.2) and (2.3), the smoothing PDF is given by

$$p(\mathbf{x}_k \mid \mathbf{z}_{1:K}) = p(\mathbf{x}_k \mid \mathbf{z}_{1:k}) \int \frac{p(\mathbf{x}_{k+1} \mid \mathbf{x}_k)\, p(\mathbf{x}_{k+1} \mid \mathbf{z}_{1:K})}{p(\mathbf{x}_{k+1} \mid \mathbf{z}_{1:k})}\, d\mathbf{x}_{k+1} \tag{2.24}$$

where $p(\mathbf{x}_k \mid \mathbf{z}_{1:k})$ is the estimated posterior PDF from filtering. The smoothing PDF can be computed as the marginal of the conditional joint PDF [56]

$$\begin{aligned} p(\mathbf{x}_k \mid \mathbf{z}_{1:K}) &= \int p(\mathbf{x}_k, \mathbf{x}_{k+1} \mid \mathbf{z}_{1:K})\, d\mathbf{x}_{k+1} \\ &= \int p(\mathbf{x}_k \mid \mathbf{x}_{k+1}, \mathbf{z}_{1:K})\; p(\mathbf{x}_{k+1} \mid \mathbf{z}_{1:K})\, d\mathbf{x}_{k+1} \end{aligned}$$

where, from the Markov assumption, $p(\mathbf{x}_k \mid \mathbf{x}_{k+1}, \mathbf{z}_{1:K}) = p(\mathbf{x}_k \mid \mathbf{x}_{k+1}, \mathbf{z}_{1:k})$ and

$$p(\mathbf{x}_k \mid \mathbf{x}_{k+1}, \mathbf{z}_{1:k}) = \frac{p(\mathbf{x}_k, \mathbf{x}_{k+1} \mid \mathbf{z}_{1:k})}{p(\mathbf{x}_{k+1} \mid \mathbf{z}_{1:k})} \tag{2.25}$$

Using filtering steps, we can obtain the numerator of Equation (2.25) as

$$p(\mathbf{x}_k, \mathbf{x}_{k+1} \mid \mathbf{z}_{1:k}) = p(\mathbf{x}_{k+1} \mid \mathbf{x}_k)\, p(\mathbf{x}_k \mid \mathbf{z}_{1:k})$$

and the denominator of Equation (2.25) as

$$p(\mathbf{x}_{k+1} \mid \mathbf{z}_{1:k}) = \int p(\mathbf{x}_{k+1} \mid \mathbf{x}_k)\, p(\mathbf{x}_k \mid \mathbf{z}_{1:k})\, d\mathbf{x}_k\,.$$

2.9 Kalman Smoothing

Similar to the KF in Section 2.4, a Kalman smoother (KS) algorithm exists in closed form for linear system systems with Gaussian PDFs. In particular, $p(\mathbf{x}_k \mid \mathbf{z}_{1:K})$ can be obtained as a Gaussian PDF with mean $\hat{\mathbf{x}}_k^{(s)}$ and covariance matrix $\mathbf{P}_k^{(s)}$; here, the superscript $^{(s)}$ is used to denote the result of smoothing. The backward recursion equations for the KS are provided in Algorithm 7 [4, 23]. Note that, due to the

backwards computation, the initial values of the recursive algorithm are $\hat{\mathbf{x}}_K^{(s)} = \hat{\mathbf{x}}_K$ and $\mathbf{P}_K^{(s)} = \mathbf{P}_K$.

Algorithm 7 Kalman Smoothing (KS)

Input: Filtered mean and covariance $\hat{\mathbf{x}}_k$, $\mathbf{P}_k$, $k = 1 \ldots, K$, dynamic model $\mathbf{F}$, covariance $\mathbf{Q}_k$

Initialize: $\hat{\mathbf{x}}_K^{(s)} = \hat{\mathbf{x}}_K$ and $\mathbf{P}_K^{(s)} = \mathbf{P}_K$

for $k = K - 1$ to 1 **do**

$\hat{\mathbf{x}}_{k+1|k} = \mathbf{F}_k \hat{\mathbf{x}}_{k|k}$

$\mathbf{P}_{k+1|k} = \mathbf{F}_k \mathbf{P}_{k|k} \mathbf{F}_k^\top + \mathbf{Q}_k$

$\mathbf{K}_k = \mathbf{P}_{k|k} \mathbf{F}_k^\top [\mathbf{P}_{k+1|k}]^{-1}$

$\hat{\mathbf{x}}_k^{(s)} = \hat{\mathbf{x}}_{k|k} + \mathbf{K}_k (\hat{\mathbf{x}}_{k+1}^{(s)} - \hat{\mathbf{x}}_{k+1|k})$

$\mathbf{P}_k^{(s)} = \mathbf{P}_{k|k} + \mathbf{K}_k (\mathbf{P}_{k+1}^{(s)} - \mathbf{P}_{k+1|k}) \mathbf{K}_k^\top$

end for

Output: Estimation smoothing mean $\hat{\mathbf{x}}_k^{(s)}$ and covariance $\mathbf{P}_k^{(s)}$, $k = 1, \ldots, K$

2.10 Extended Kalman Smoothing

Extended KS (EKS) based on an analogous approximation to the EKF is used to form Gaussian approximations by making nonlinear state-space models linearize. The EKS can be obtained from the KS equations by replacing the prediction equations with first-order approximations [4]. The difference between the EKS and the KS is the same as the difference between the EKF and KF, which is the state transition matrix is replaced with Jacobian [55]. Like the KS, the EKS also form a Gaussian approximation to a smoothing distribution $p(\mathbf{x}_k \mid \mathbf{z}_{1:K}) \simeq \mathcal{N}(\mathbf{x}_k \mid \hat{\mathbf{x}}_k^{(s)}, \mathbf{P}_k^{(s)})$.

2.11 Unscented Kalman Smoothing

Unscented KS (UKS) is a Gaussian approximation based smoothing where the non-linearity is approximated using the UT [4]. T he UKS is derived as follows. First, the sigma points are formed for the random variable $(\mathbf{x}_k, \mathbf{q}_k)$ as follows:

$$\begin{aligned}
\mathcal{X}_k^{(0)} &= \hat{\mathbf{x}}_k \\
\mathcal{X}_k^{(i)} &= \hat{\mathbf{x}}_k + \sqrt{n+\lambda}\big[\sqrt{\mathbf{P}}_k\big]_i, \\
\mathcal{X}_k^{(i+n)} &= \hat{\mathbf{x}}_k - \sqrt{n+\lambda}\big[\sqrt{\mathbf{P}}_k\big]_i, \quad i = 1, \ldots, n.
\end{aligned}$$

Then, the sigma points are propagated through the dynamic model as follows:

$$\mathcal{X}_{k+1|k}^{(i)} = g(\mathcal{X}_{k|k}^{(i)}), \quad i = 0, \ldots, 2n.$$

Then, we can compute the predicted mean $\hat{\mathbf{x}}_{k+1|k}$ by adding the whole weighted sigma points, the predicted covariance $\mathbf{P}_{k+1|k}$, and the cross-covariance $\mathbf{C}_{\mathcal{X}_{k+1}}$ as follows [4]:

$$\begin{aligned}
\hat{\mathbf{x}}_{k+1|k} &= \sum_{i=0}^{2n} w_m^{(i)} \mathcal{X}_{k+1|k}^{(i)}, \\
\mathbf{P}_{k+1|k} &= \sum_{i=0}^{2n} w_c^{(i)} (\mathcal{X}_{k+1|k}^{(i)} - \hat{\mathbf{x}}_{k+1|k})(\mathcal{X}_{k+1|k}^{(i)} - \hat{\mathbf{x}}_{k+1|k})^\top \\
\mathbf{C}_{\mathcal{X}_{k+1}} &= \sum_{i=0}^{2n} w_c^{(i)} (\mathcal{X}_{k|k}^{(i)} - \hat{\mathbf{x}}_{k|k}) \left(\mathcal{X}_{k+1|k}^{(i)} - \hat{\mathbf{x}}_{k+1|k}\right)^\top,
\end{aligned}$$

We can then compute the smoothing gain $\mathbf{K}_k$, the estimation smoothing mean $\hat{\mathbf{x}}_k^{(s)}$, and the covariance $\mathbf{P}_k^{(s)}$ as follows [4]:

$$\begin{aligned}
\mathbf{K}_k &= \mathbf{C}_{\mathcal{X}_{k+1}} [\mathbf{P}_{k+1|k}]^{-1}, \\
\hat{\mathbf{x}}_k^{(s)} &= \hat{\mathbf{x}}_{k|k} + \mathbf{K}_k [\hat{\mathbf{x}}_{k+1}^{(s)} - \hat{\mathbf{x}}_{k+1|k}], \\
\mathbf{P}_k^{(s)} &= \mathbf{P}_{k|k} + \mathbf{K}_k [\mathbf{P}_{k+1}^{(s)} - \mathbf{P}_{k+1|k}] \mathbf{K}_k^\top.
\end{aligned}$$

The UKS algorithm is provided in Algorithm 8 [23].

Algorithm 8 Unscented Kalman Smoothing (UKS) Algorithm

Input: Filtered mean $\hat{\mathbf{x}}_k$ and covariance $\mathbf{P}_k$, $k = 1, \ldots, K$, measurements $\mathbf{z}_{1:K}$
dynamic model $g(\cdot)$, covariance of process noise $\mathbf{Q}_k$

Initialize: $\hat{\mathbf{x}}_K^{(s)} = \hat{\mathbf{x}}_K$ and $\mathbf{P}_K^{(s)} = \mathbf{P}_K$

for $k = K - 1$ to 1 **do**

$$\{\hat{\mathbf{x}}_{k+1|k}, \mathbf{P}_{k+1|k}\} = \mathtt{UT}\left(\hat{\mathbf{x}}_{k|k}, \mathbf{P}_{k|k}, g(\cdot), \mathbf{Q}_{k-1}, n, \lambda\right) \text{ from Algorithm 3}$$

$$\hat{\mathbf{x}}_{k+1|k} = \sum_{i=0}^{2n} w_m^{(i)} \mathcal{X}_{k+1|k}^{(i)}$$

$$\mathbf{P}_{k+1|k} = \sum_{i=0}^{2n} w_c^{(i)} (\mathcal{X}_{k+1|k}^{(i)} - \hat{\mathbf{x}}_{k+1|k}) \left(\mathcal{X}_{k+1|k}^{(i)} - \hat{\mathbf{x}}_{k+1|k}\right)^\top + \mathbf{Q}_{k-1}$$

$$\mathbf{C}_{\mathcal{X}_{k+1}} = \sum_{i=0}^{2n} w_c^{(i)} (\mathcal{X}_{k|k}^{(i)} - \hat{\mathbf{x}}_{k|k})(\mathcal{X}_{k+1|k}^{(i)} - \hat{\mathbf{x}}_{k+1|k})^\top$$

$$\mathbf{G}_k = \mathbf{C}_{\mathcal{X}_{k+1}} [\mathbf{P}_{k+1|k}]^{-1}$$

$$\hat{\mathbf{x}}_k^{(s)} = \hat{\mathbf{x}}_{k|k} + \mathbf{G}_k (\hat{\mathbf{x}}_{k+1}^{(s)} - \hat{\mathbf{x}}_{k+1|k})$$

$$\mathbf{P}_k^{(s)} = \mathbf{P}_{k|k} + \mathbf{G}_k (\mathbf{P}_{k+1}^{(s)} - \mathbf{P}_{k+1|k}) \mathbf{G}_k^\top$$

end for

Output: The smoothing mean, covariance and gain $\{\hat{\mathbf{x}}_k^{(s)}, \mathbf{P}_k^{(s)}, \mathbf{G}_k\}_{k=1:K}$

Chapter 3

PROPOSED TRACKING METHOD

3.1 Detection Formulation

In our problem formulation, sensor observations are first processed to determine the presence of an object before estimating the object's parameters as they change with time. As a fixed time step, and for a fixed probability of false alarm, we want to determine whether the received noisy observations $y[n],\ n=0,\ldots,N-1$, include information on the object of interest. We formulate a binary detection hypothesis as

$$\mathcal{H}_0:\ y[n]=v[n],\quad n=0,1,\ldots,N-1$$

$$\mathcal{H}_1:\ y[n]=s[n;\boldsymbol{\theta}]+v[n],\quad n=0,1,\ldots,N-1$$

where $s[n]$, $n=0,1,\ldots,N-1$, is the transmit signal, and $\boldsymbol{\theta}$ is a vector that consists of unknown changes to the transmit signal due to transmission in the medium. For example, for radar detection, $\boldsymbol{\theta}$ may include amplitude fading, time delay and Doppler shift. Under hypothesis $\mathcal{H}_0$, the received signal is assumed to consist only of white Gaussian noise samples $v[n]$ and under hypothesis $\mathcal{H}_1$, the received signal consists of both the transmit signal and noise. Using the Neyman-Pearson detector, we want to find a detection statistic that maximizes the probability of detection for a given probability of alarm $\mathrm{P_{FA}}$ [31]. Once the presence of the transmit signal is detected, the received signal can be used to estimate the unknown parameter vector $\boldsymbol{\theta}$. This vector is then processed to provide a measurement vector $\mathbf{z}$ that can be used in tracking. For a radar tracking application, for example, $\mathbf{z}$ can include the distance between the radar receiver and the object and the changein the object's velocity.

3.2 Tracking Formulation

For the tracking problem formulation, multiple signal transmissions are considered at different time steps k. Assuming the signal is detected at time step k, $k = 1, \ldots, K$, then the resulting measurement vector $\mathbf{z}_k$ is used in the tracking problem formulation to estimate the unknown state parameter vector $\mathbf{x}_k$ of the moving object. As discussed in Chapter 2, the the dynamic tracking system can be modeled using the state-space representation

$$\mathbf{x}_k = g(\mathbf{x}_{k-1}) + \mathbf{q}_{k-1} \tag{3.1}$$

$$\mathbf{z}_k = h(\mathbf{x}_k) + \mathbf{r}_k \,. \tag{3.2}$$

Here, $\mathbf{r}_k$ is the measurement noise vector consisting of zero-mean white Gaussian samples that are independent and identically distributed with known variance. The random vector $\mathbf{q}_k$ is used to account for modeling errors in the state transition equation. The function $g(\mathbf{x}_k)$ models the transition of the unknown state vector between time steps and $h(\mathbf{x}_k)$ provides the relationship between the measurement and unknown state. The unknown state parameter is obtained by estimating the state posterior probability density function (PDF) $p(\mathbf{x}_k \mid \mathbf{z}_k)$. This can be achieved recursively using Bayesian filtering, as discussed in Chapter 2. At each time step k, first the object state $\mathbf{x}_k$ is predicted using the prior PDF $p(\mathbf{x}_k \mid \mathbf{x}_{k-1})$ in Equation (3.1). Second, the likelihood PDF $p(\mathbf{z}_k \mid \mathbf{x}_k)$ in Equation (3.2) is used to update the object state $\mathbf{x}_k$. Assuming that the probabilistic models for $\mathbf{q}_k$ in Equation (3.1) and $\mathbf{r}_k$ and Equation (3.2) are known, the posterior PDF can be estimated recursively. Different approaches for Bayesian filtering were discussed Chapter 2, depending on the characteristics of the dynamic system. Such methods include the Kalman filter (KF), extended KF (EKF), unscented KF (UKF) and sequential Monte Carlo methods such as particle

filtering [12, 34].

3.3 Tracking Scenario

We consider a tracking scenario under low signal-to-noise-ratio (SNR) conditions and high clutter. We assume that the motion model in Equation (3.1) is matched to the scenario, and thus the modeling error process $\mathbf{q}_k$ has low variance. As a result, any large errors in the estimated state parameters after Bayesian filtering may be attributed to high noise observations that are incorrectly processed due to, for example, a high probability of false alarm. The selection of an acceptable probability of false alarm depends on a few factors in a given radar tracking application. For example, when monitoring vital signs in healthcare, a large probability of false alarm may only require additional testing whereas a high probability of miss could lead to long term complications. Another example is the use of ground penetrating radar in forensic investigations [30]. a high probability of false alarm may result in further testing in order to locate a gravesite whereas a high probability of miss may lead to unsolved murder cases. Note that, in both of these radar tracking examples, the estimated parameters are not required in real time. As a result, the estimation accuracy can be further improved with additional processing after collecting all measurements. Scenarios that allow for high probability of false alarm can thus benefit from further processing to improve estimation performance.

Our proposed method is designed to improve the tracking accuracy of an estimated at a time step k by making use of all available measurements available after time k. In particular, the new method integrates both Bayesian filtering and Bayesian smoothing with a thresholding technique. The thresholding helps to identify and remove high-noise measurements from the estimation process.

3.4 Integration of thresholding With Bayesian Filtering and Smoothing

3.4.1 Integrated Tracking Algorithm

Our proposed tracking method uses both Bayesian filtering and Bayesian smoothing, as discussed in Sections 2.3 and 2.8, respectively. To allow for nonlinear dynamic systems, we select the unscented Kalman filter (UKF) and the unscented Kalman smoother (UKS), as they were shown to perform better than their extended Kalman counterparts/ The UKF and UKS methods are described in Sections 2.6 and 2.11, respectively.

Using the system representation in Equations (3.1) and (3.2), we first use the UKF to sequentially estimate the state parameter $\mathbf{x}_k$, for all time steps, $k = 1, \ldots, K$. We then use a distance-based metric to compare the estimated state at each time step to a neighborhood of estimated states, both from previous and future time steps. Thresholding is then applied to determine the validity of the estimate. This process is aimed to identify large deviations in the estimate at time k from its nearest neighbors. When such a deviation occurs, we decide to replace that estimate with its predicted value from the UKF. In essence, the process is aimed to eliminate measurements that may have resulted from false alarms and accepted as true detections during detection. The thresholding process is described next.

For each UKF estimated state $\mathbf{x}_k$, we define an L size neighborhoods of $\mathbf{x}_k$ as the set of $2\,L + 1$ estimated states

$$\mathcal{X}_k = \{\mathbf{x}_{k-L}, \mathbf{x}_{k-(L-1)}, \ldots, \mathbf{x}_{k+(L-1)}, \mathbf{x}_{k+L}\}\,,\ k = L+1, \ldots, K-L \tag{3.3}$$

The estimates in this L-neighborhood are used to compute the moving average of the

differences of consecutive estimated states. In particular, at times step k, we compute

$$D_{k,L} = \frac{1}{2\,L} \sum_{m=-L}^{L-1} |\mathbf{x}_{k+m+1} - \mathbf{x}_{k+m}| = \frac{1}{2\,L} \sum_{m=-L}^{L-1} d_{k+m+1,k+m} \tag{3.4}$$

where the difference between two consecutive estimated states is given by

$$d_{k+m+1,k+m} = |\mathbf{x}_{k+m+1} - \mathbf{x}_{k+m}|$$

We then compute the decision metric criterion

$$\mathcal{D}_{k,L} = \frac{D_{k,L}}{D_{k,1}} = \frac{1}{L} \frac{\sum_{m=-L}^{L-1} |\mathbf{x}_{k+m+1} - \mathbf{x}_{k+m}|}{\Big(|\mathbf{x}_k - \mathbf{x}_{k-1}| + |\mathbf{x}_{k+1} - \mathbf{x}_k|\Big)} \tag{3.5}$$

From the decision metric criterion in Equation (3.5), a high denominator would indicate a large estimation error in $\mathbf{x}_k$ due to the large difference between $\mathbf{x}_k$ and its two immediate neighbors, $\mathbf{x}_{k-1}$ and $\mathbf{x}_{k-1}$. The increased denominator value thus implies that the UKF estimate $\mathbf{x}_k$ was updated using an unacceptable measurement $\mathbf{z}_k$, either due to high noise or high clutter. If we select a threshold α such that $0 < \alpha < 1$, then, if $\mathcal{D}_{k,L} < \alpha$, the UKF estimate should be replaced with its predicted value $g(\mathbf{x}_{k-1})$ from the state transition equation in (3.1). As a result, thresholding yields a new estimate $\tilde{\mathbf{x}}_k$ accoriding to

$$\tilde{\mathbf{x}}_k = \begin{cases} g(\mathbf{x}_{k-1}, & \mathcal{D}_{k,L} < \alpha \\ \mathbf{x}_k, & \text{otherwise} \end{cases} \quad k = L+1, \ldots, K-L \tag{3.6}$$

As we cannot form L size neighborhoods for the UKF state estimates $\mathbf{x}_k$ when $k < L+1$ and $k > K-L$, then $\tilde{\mathbf{x}}_k = \mathbf{x}_k$.

Note that we refer to this new method of integrating UKF with thresholding as T-UKF. The overall steps of the method are provided in Algorithm 9. To further increase tracking performance, we also consider the T-UKS method, where the estimates of the T-UKF are further smoothed using the UKS.

Algorithm 9 Thresholding of UKF State Parameter Estimates

Input: UKF estimated state $\mathbf{x}_k$, $k = 1, \ldots, K$, transition function $g(\cdot)$

threshold α, neighborhood size L

for $k < L + 1$ or $k < K - L$ **do**

Set $\tilde{\mathbf{x}}_k = \mathbf{x}_k$

end for

for $k = L + 1$ to $K - L$ **do**

Obtain L neighborhood of $\mathbf{x}_k$ in Equation (3.3)

Compute $D_{k,L}$ and $D_{k,1}$ using Equation (3.4)

Compute $\mathcal{D}_{k,L}$ using Equation (3.5)

Use decision criterion in in Equation (3.6) to select new state estimate $\tilde{\mathbf{x}}_k$

end for

Output: Thresholded state estimate $\tilde{\mathbf{x}}_{1:K}$

Algorithm 10 Smoothing of the Thresholded UKF State Parameter Estimates

Input: T-UKF estimated state parameters $\tilde{\mathbf{x}}_k$, $k = 1, \ldots, K$ from Algorithm 9

Use UKS in Algorithm 8

Output: State estimates after filtering, thresholding and smoothing

3.5 Thresholding Demonstration With Varying Parameters

The tracking performance of the proposed T-UKF and T-UKS methods highly depend on two parameters. The parameters are the neighborhood size L and the decision threshold α in Equation (3.6). If the neighborhood size is too small, we do not expect to see an increase in performance; on the other hand, if L is too large, valid measurements could be removed. A similar behavior in performance can be observed with the threshold α. In the next chapter, we fix one parameter at a time and vary the other one in order to select an adequate pair to increase performance for a particular

tracking scenario.

In order to demonstrate the effect of the thresholding process in Equation (3.6), we simulate a simple random process

$$x_k = x_{k-1} + w_k, \quad w \sim \mathcal{N}(0, 100^2),$$

where x_k is time-varying variable and w is observation noise that zero-mean white Gaussian noise with high variance 10,000. Figure 3.1a shows a simulation of x_k, for $k = 1, \ldots, 100$, with the observation noise added x_k values marked. We set the initial state $x_0 = 0$. We apply the thresholding algorithm following the steps in Algorithm 9. For this demonstration, we selected $L = 6$ and $\alpha = 0.6$. The estimated x_k values after thresholding are shown superimposed on the true values in Figure 3.1b. As indicated in the figure, five values of the estimated x_k were thresholded and moved closer to the true values. These values occurred at $k = 4, 20, 45, 56, 65$. The thresholding process assumes that the estimates at these times came from measurements that should not have resulted in valid detection.

In subsequent simulations, we varied the values of L and α. In Figure 3.2, the threshold value is fixed to $\alpha = 0.6$ and L was varied as $L = 3, 4, 5$. When the neighborhood size is $L = 4$, 7 points were thresholded, whereas 6 points were thresholded for $L = 5$ and $L = 6$. The overall comparison of the three L values is shown in the bottom right figure. As it can be seen, for this example, the larger L value resulted in better estimation performance. This is also shown in Table 3.1, where estimation root mean-squared error (RMSE) results are shown after 10,000 Monte Carlo simulations. Since the state without the observation noise is supposed to be 0, and we want to know how the thresholding process can decrease the effect of the observation noise w, the RMSE is computed as $\sqrt{\frac{1}{100}\left(\sum_{k=1}^{100}(x_k - 0)^2\right)}$. The table shows the lowest RMSE is when $L = 4$ when the threshold was fixed to $\alpha = 0.5$, and when $\alpha = 0.6$ when the

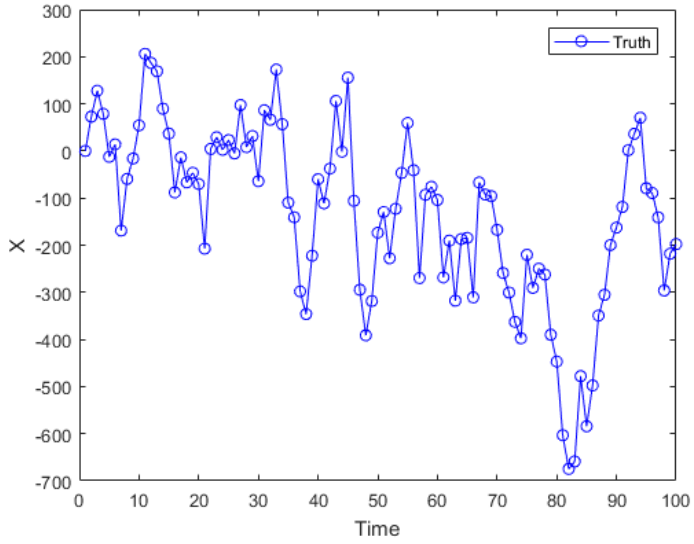
L is fixed to $L = 4$.

In Figure 3.3, the neighborhood size is fixed to $L = 6$ and and the threshold varied $\alpha = 0.5, 0.6, 0.7$. The number of corresponding points thresholded is shown on the plots as 4, 9 and 14. Both Table 3.1 and the bottom right figure show that in this case, the higher $\alpha = 0.7$ resulted in better performance from $L = 6$. It is important to note that the overall performance is as summarized in Table 3.1 as 10,000 Monte Carlo runs were used. The demonstrations in Figures 3.2 and 3.3 are only from one run.

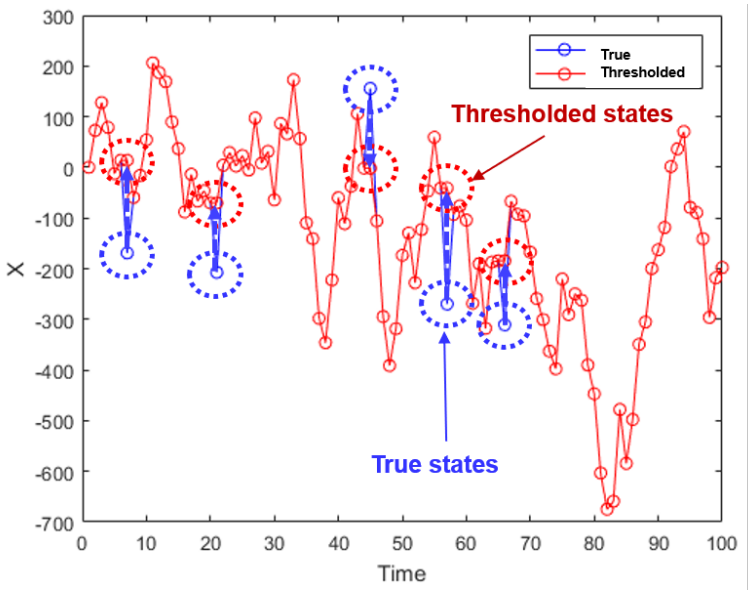
As future work, we will develop methods to adaptively select the best algorithm parameters based on the particular tracking scenario.

	$L = 4$	$L = 5$	$L = 6$
RMSE in samples for $\alpha = 0.6$	580	583	590
	$\alpha = 0.5$	$\alpha = 0.6$	$\alpha = 0.7$
RMSE in samples for $L = 6$	545	586	600

Table 3.1: RMSE in Samples For Varying L and α Values



(a) True Values of x_k as a Function of Time Step k.



(b) Estimated Values of x_k Superimposed with True Values. The Five Circled Points Underwent Thresholding to Improve Estimation Performance.

Figure 3.1: Result of True and Thresholded States

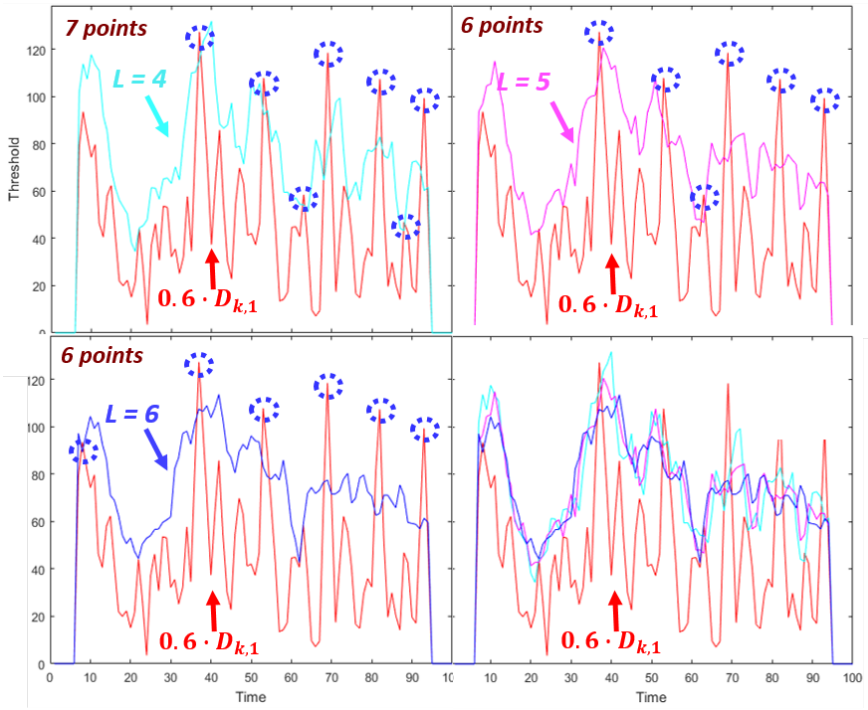


Figure 3.2: Thresholding for Fixed $\alpha = 0.6$ and Varying $L = 3, 4, 5$.

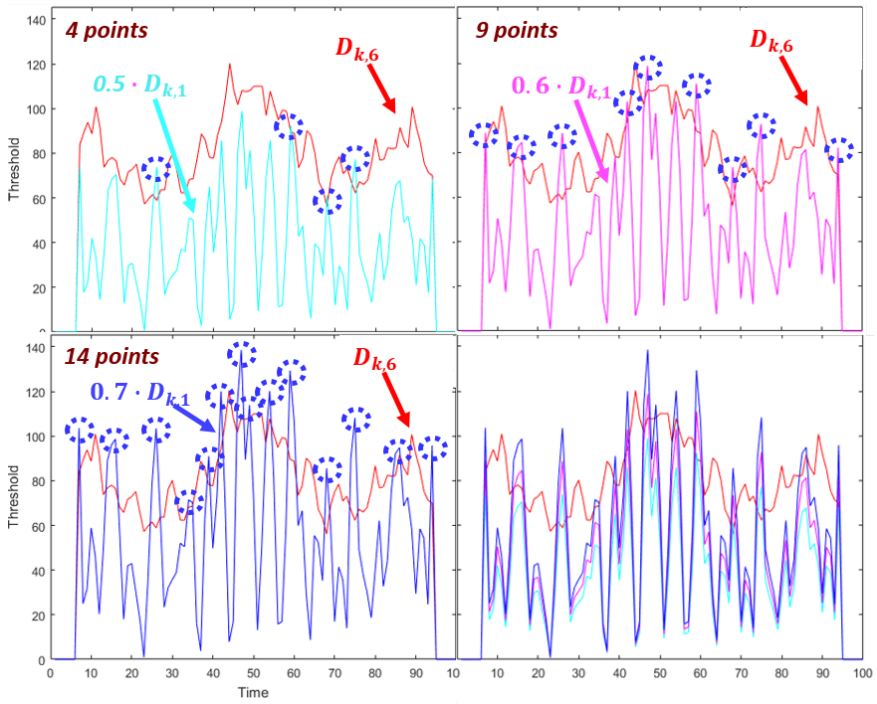


Figure 3.3: Thresholding for Varying $\alpha = 0.5, 0.6, 0.7$ and Fixed $L = 6$.

Chapter 4

SIMULATIONS AND DISCUSSION

4.1 Simulation Parameters

In this chapter, we simulate various tracking scenarios to estimate the position of an object using radar measurements of range and bearing. We use the simulations to compare the performance of our proposed thresholding-based method with methods that only rely on either Bayesian filtering or on both Bayesian filtering and smoothing.

We assume a dynamic system with two-dimensional (2-D) turning motion. The transition equation is given by

$$\mathbf{x}_k = [x_k \;\; y_k \;\; \dot{x}_k \;\; \dot{y}_k \;\; \omega_k]^\top = \mathbf{F}(\omega_k)\,\mathbf{x}_{k-1} + \mathbf{q}_{k-1}$$

where (x_k, y_k) and $(\dot{x}_k, \dot{y}_k)$ are 2-D Cartesian coordinates of position and velocity, respectively, ω_k is the turn rate in radians per second. We selected a simplified transition matrix

$$\mathbf{F}(\omega_k) = \begin{bmatrix} 1 & 0 & \Delta t & 0 & 0 \\ 0 & 1 & 0 & \Delta t & 0 \\ 0 & 0 & \cos(\omega_k\,\Delta t) & -\sin(\omega\,\Delta t) & 0 \\ 0 & 0 & \sin(\omega_k\,\Delta t) & \cos(\omega_k\,\Delta t) & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

where Δt is the time between any two steps. The initial state was $[x_0 \;\; y_0 \;\; \dot{x}_0 \;\; \dot{y}_0 \;\; \omega_0] = [0 \;\; 0 \;\; 4 \;\; 10 \;\; 0.02]$. The modeling error covariance $\mathbf{Q}_k = \mathbf{Q} = \sigma^2\mathbf{I}$, where $\mathbf{I}$ is the 5×5 identity matrix and with negligible σ^2. The tracking field-of-view (FOV) consists of an 1,600 m by 800 m rectangular area. We simulate $K = 100$ time steps with $\Delta t = 1$

s. The measurement equation is given by

$$\mathbf{z}_k = \begin{cases} h(\mathbf{x}_k) + \mathbf{r}_k \\ \\ \mathbf{z}_k^{(c)} + \mathbf{r}_k \end{cases} \tag{4.1}$$

where

$$h(\mathbf{x}_k) = [\phi_k \;\; r_k]^\top = \left[\arctan\left(y_k/x_k\right) \;\; \sqrt{x_k^2 + y_k^2}\right]^\top$$

with bearing angle ϕ_k in in radians and range r_k meters. The measurement noise covariance is given by

$$\mathbf{R}_k = \rho_k \begin{bmatrix} 1 & 0 \\ \\ 0 & 5 \end{bmatrix}$$

with time-varying noise intensity ρ_k that varies between two values with equal probability. For high noise (HN), it varies between 4 and 8; for low noise (LN), it varies between 2 and 4. In Equation (4.1), $\mathbf{z}_k^{(c)}$ is clutter that is modeled as a Poisson point process with rate parameter λ false alarms per scan of uniform clutter in the FOV. We use $\lambda = 10$ for low clutter (LC) and $\lambda = 50$ for high clutter (HC). Unless otherwise stated, we obtain estimation root mean-squared error (RMSE) performance by running 10,000 Monte Carlo (MC) runs. The detection probability is 0.98. For all simulations, we assume that all measurements are due to valid detections; some of these detections however, are a result of a high probability of false alarm. We also assume that the object is always present in the FOV. The actual 2-D Cartesian coordinates (x_k, y_k) of the object is shown in Figure 4.1.

We compute performance using estimation root mean-squared error (RMSE) for the number of runs for which thresholding has occurred. This is because the UKF estimate only changes if $\mathcal{D}_{k,L} < \alpha$ in Equation (3.6). As a result, we do not include the runs when the UKF estimate remains the same.

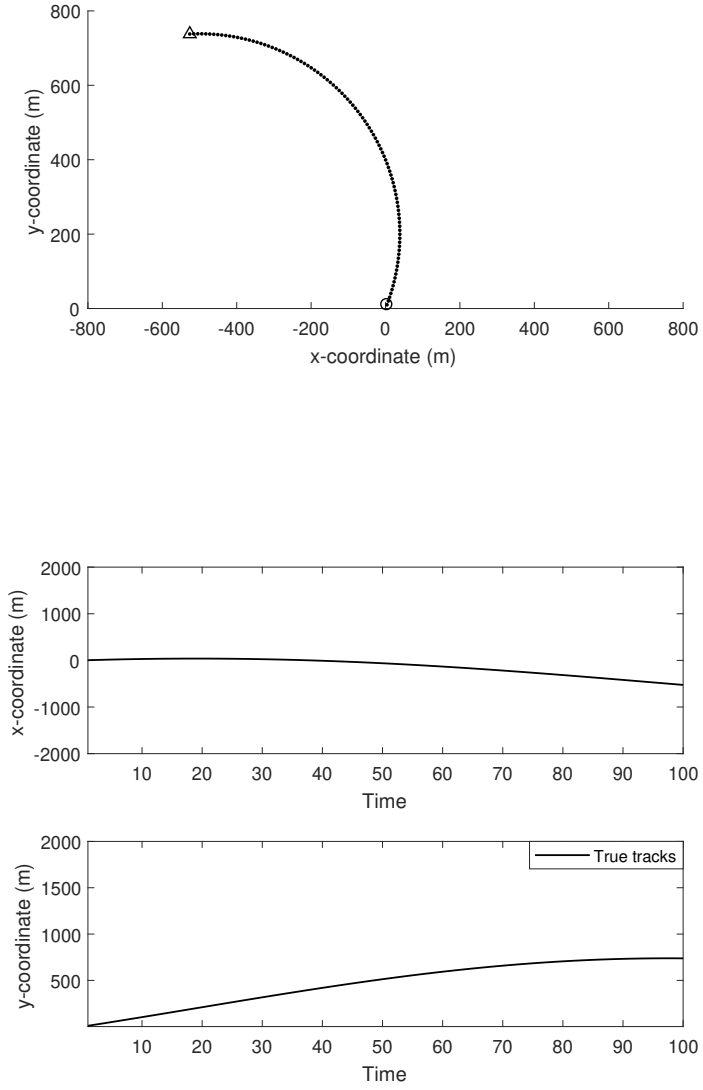


Figure 4.1: Actual 2-D Cartesian Coordinates of Moving Object.

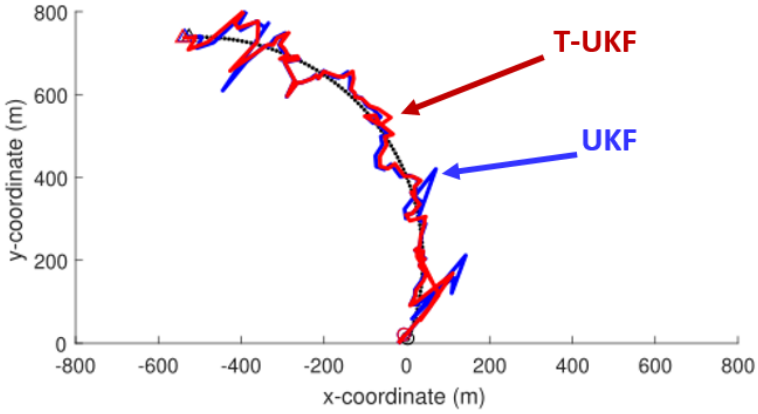
4.2 UKF and T-UKF Estimation Methods

The T-UKF estimation method integrates UKF with thresholding in Algorithm 9. The comparison of the UKF results and the proposed T-UKF method is given in

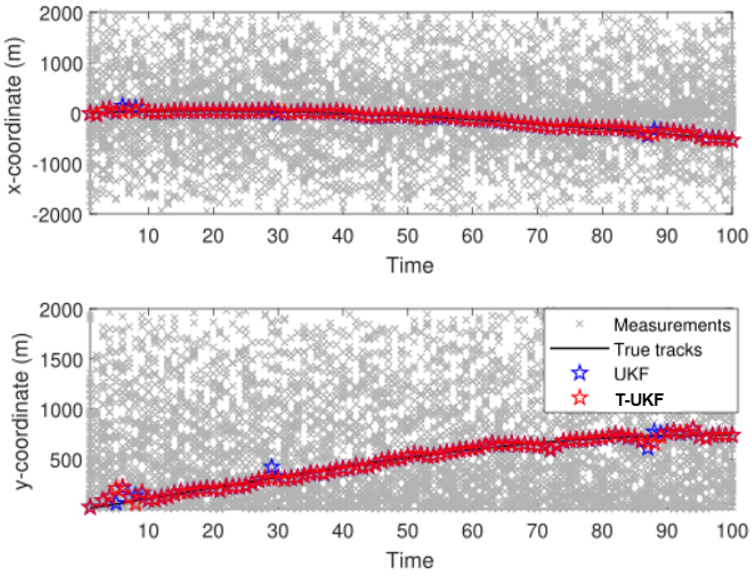
Figure 4.2. In the figure, UKF is shown in blue and T-UKF is shown in red. High noise measurements can be seen at time step 25 and 39; they can also be seen in the bottom graph of Figure 4.2(b) with blue stars. These errors in the UKF are thresholded using the T-UKF.

4.2.1 UKS and T-UKS Estimation Methods

In this process, we estimate each state given all the measurements we have obtained using the Bayesian smoothing method. Especially, in this nonlinear scenario, we use the UKS which is the closed form smoothing solution to get the result. The only difference between UKS and T-UKS process is using the T-UKF states instead of the states of the UKF. Figure 4.3 shows the comparison between the results of the UKS and T-UKS. If the result of the T-UKF is better than the UKF, The T-UKS is better than the UKS probabilistically. The result of the T-UKS restrains the distortion of the trajectory from the thresholding process. The accuracy of the result does not always show the proposed method better, this is the reason to find proper threshold parameters.

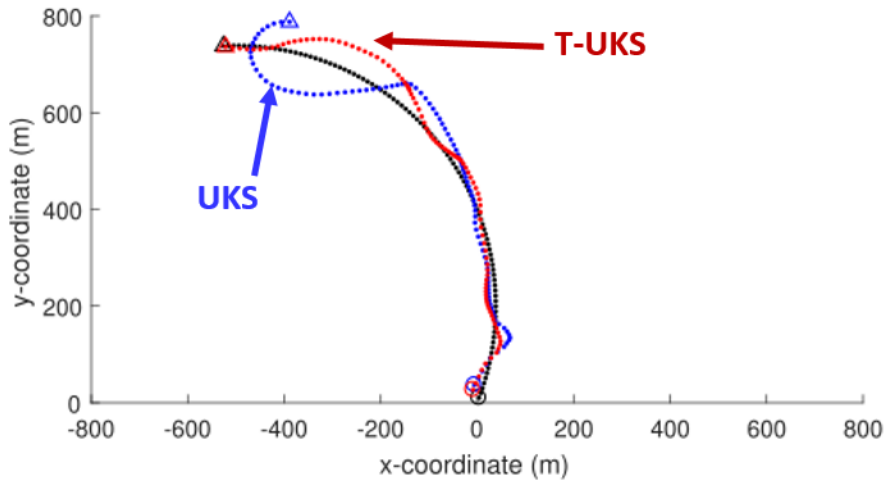


(a) Estimated (x_k, y_k) Using UKF and T-UKF.

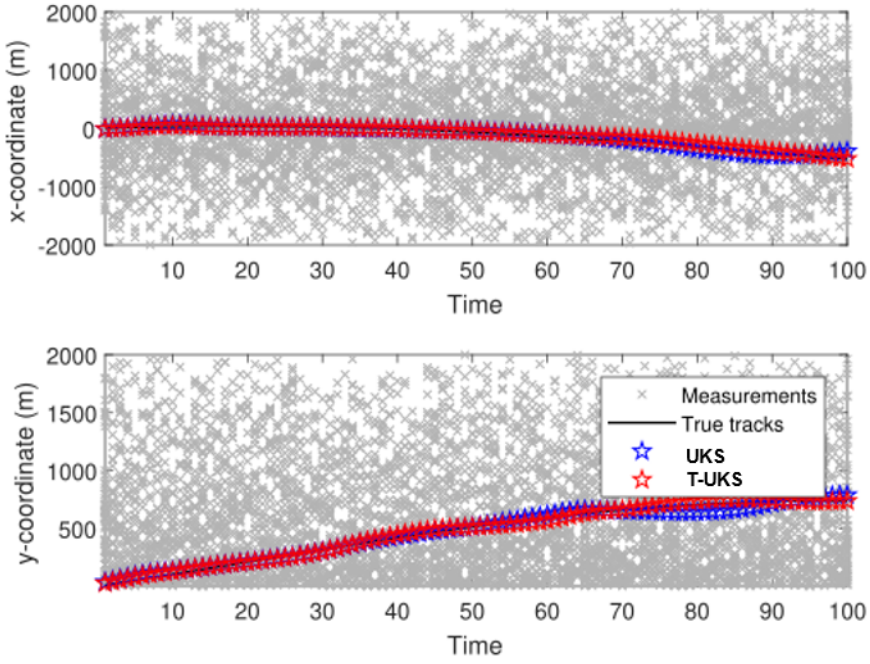


(b) UKF and T-UKF Estimates, True Values and Measurements of x_k and y_k.

Figure 4.2: Comparison of UKF and T-UKF Performance.



(a) Estimated (x_k, y_k) Using UKS and T-UKS



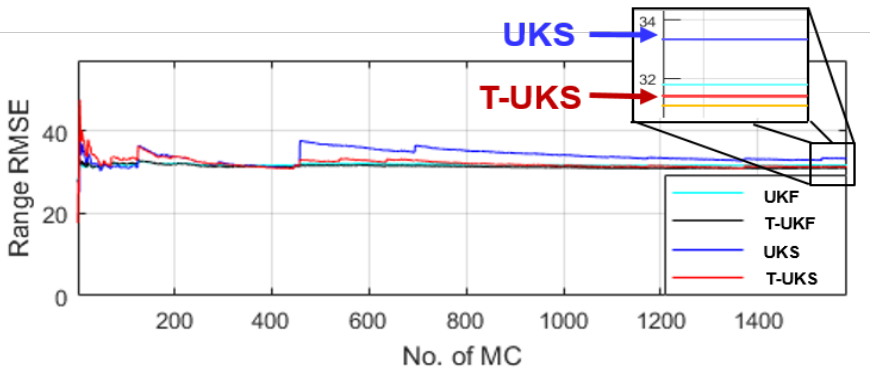
(b) UKS and T-UKS Estimates, True Values and Measurements of x_k and y_k.

Figure 4.3: Comparison of UKS and T-UKS Performance.

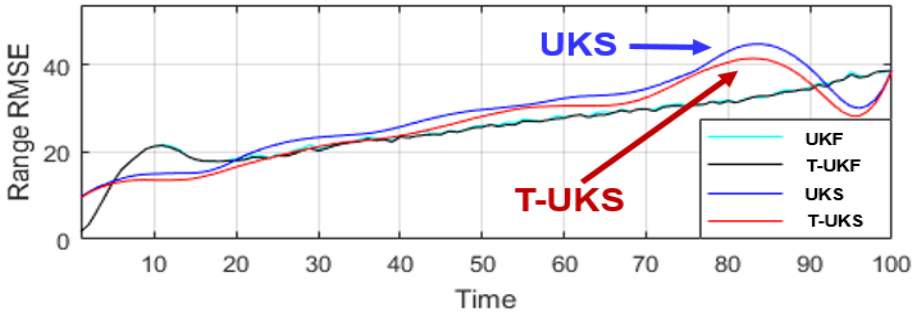
4.3 Simulation Results

We simulated 10,000 MCs for each scenario of high noise and high clutter (HNHC), high noise and low clutter (HNLC), low noise and high clutter (LNHC). Each simulation derives two figures; first, the RMSE according to the number of MCs, second, the RMSE of 10,000 MCs according to 1 to 100 time steps. The RMSE is used to compute the distance between a set of true states and state estimations and has been widely adopted to estimate an average of the squares of errors of the object tracking. Figure 4.4, 4.5, 4.6 show the 10,000 MCs results of nonlinear single object filtering and smoothing for each environment. From the simulation, we expect that the proposed method performs better than the original method probabilistically, especially when it comes to the case of high noise. Since the proposed method does not perform better always, however, if we divide the total MCs into two cases, which are the cases the proposed method works or not respectively, we know that the only cases of the threshold process works show the difference compared to the original method. In the cases that it does not work, in other words, the proposed method decides that if there is no state estimation with high noise to eliminate, it does not show the different results between the two methods. The important thing is that the performance of the proposed method depends on the proper threshold parameter we choose because too low threshold makes too many measurements missed detections, and too high threshold makes no difference in between. The simulation result shows that the proposed method either makes the result the same as the original method or performs better by neglecting high noise measurement.

Figure 4.4a shows the T-UKS performs better than the UKS under HNHC. In addition, with the parameter we set, around thresholding process operates around 1,580 times when we simulate 10,000 MCs. In other words, the thresholding process



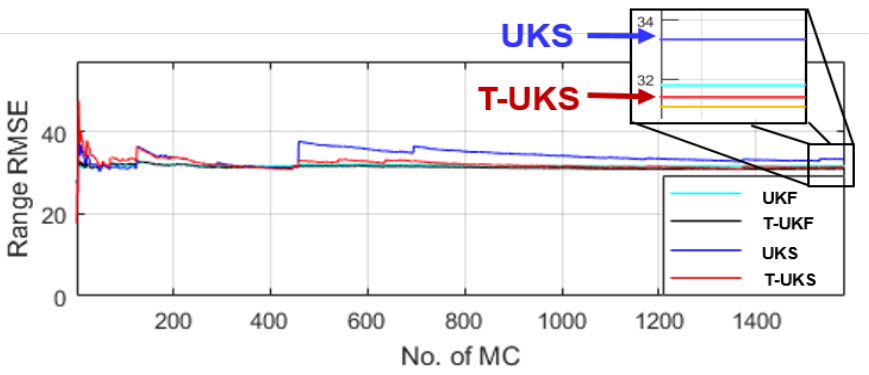
(a) The RMSE According to the Number of the MCs under HNHC



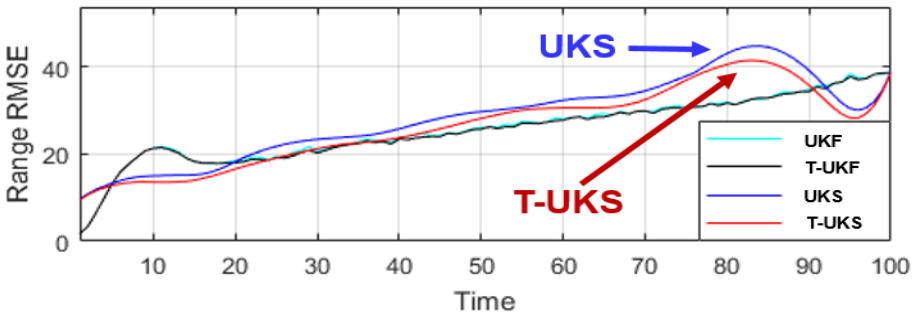
(b) RMSE According to the Entire Time Step under HNHC

Figure 4.4: RMSE under High Noise and High Clutter

does not find and work for 8,420 MCs.



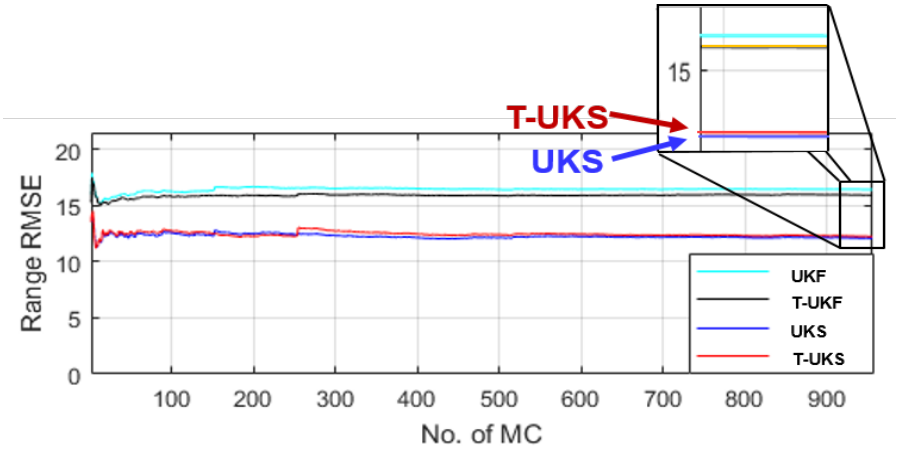
(a) The RMSE According to the Number of the MCs under HNLC



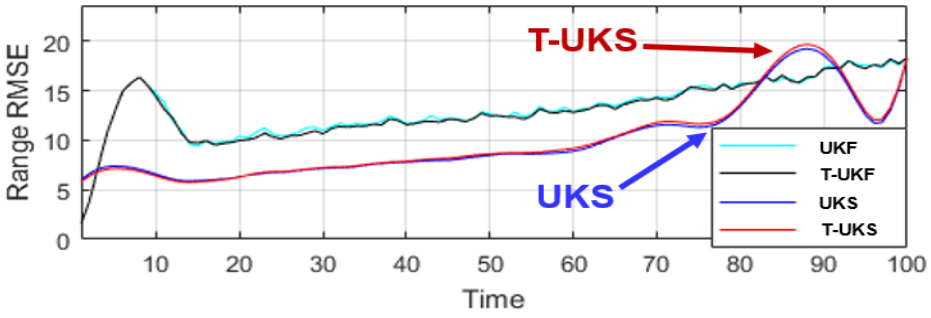
(b) The RMSE According to the Entire Time Step under HNLC

Figure 4.5: RMSE under High Noise and Low Clutter

Figure 4.5a shows the T-UKS performs better than the UKS under HNLC, like the case of HNHC. In addition, with the parameter we set, around thresholding process operates around 942 times when we simulate 10,000 MCs. In other words, the thresholding process does not find and work for 9058 MCs. Also, regardless of the number of clutter we set, the proposed and conventional methods perform well.



(a) RMSE vs Number of MC Runs Under LNHC



(b) RMSE as a Function of Time Step Under LNHC

Figure 4.6: RMSE for Low Noise and High Clutter

Figure 4.6a shows the performance of the UKS and T-UKS is almost similar since the UKS, which has low uncertainty from low noise, estimates the states accurate enough and has no significant difference compared to the T-UKS. Thus, we find out the thresholding process does not derive considerable results for the scenario with low noise, whether it works or not. To summarize the simulation, the proposed method performs better when noise is high and similar when low than the conventional method.

In the simulation, we set relatively high noise covariance to the HNLC and HNLC cases.

	HNHC	HNLC	LNHC
UKF	31.77	30.55	16.43
T-UKF	31.06	29.80	15.93
UKS	33.31	34.07	12.09
T-UKS	31.34	31.64	12.25

Table 4.1: The RMSE of Three Scenario

	HNHC	HNLC	LNHC
Thresholding Operated Cases	15.8% (1,580/10,000)	9.4% (942/10,000)	9.6% (956/10,000)
T-UKF Outperformed Cases	71.5% (1,130/1,580)	71.8% (676/942)	79.2% (757/956)
T-UKS Outperformed Cases	45.6% (720/1,580)	45.3% (427/942)	48.6% (465/956)

Table 4.2: Number of Outperformed MCs

Table 4.1 shows the RMSE of 10,000 MCs of the UKF, T-UKF, UKS, T-UKS of each scenario. Table 4.2 shows how many times that thresholding process works among the entire MCs per scenario. Also, Table 4.2 shows how many times the T-UKF and T-UKS perform better than the UKF and UKS. In Table 4.2, we notice that the proposed method does not always perform better due to the uncertainty of the true trajectory, noise, clutter and so on. Finding a probable threshold is important because the threshold we set is not optimal. In other words, parameters are all varied depending on the characteristics of the object so that the performance can be better depending on the proper parameters. If the thresholding process does not operate, we only can compare the result of the UKF and UKS. The RMSE of the UKS performs better than the UKF, unlike the result of Table 4.2, and it means when the

thresholding stage does not find any suspicious high-noise contained measurement, the smoothing usually performs better than the filter. Thus, finding a fitted threshold to each scenario increases MC cases that show better performance of the T-UKF and T-UKS and avoids the case of not having high noise.

4.3.1 Simulation for Cross-Validation

As we discussed, the process parameters are selected heuristic-based so that the performance is changed based on the parameter selection. Thus, we can find parameters that perform better by simulating them many times with varied parameters. Since we have two parameters in the thresholding process, α, L, we can find the better parameters by keeping α fixed, and varying L. Next, do the same simulation with varied α while keeping the parameter L of what we find from the previous validation. A recursively performed cross-validation process can find the optimal parameters for the scenario, and it results in the optimal parameters based on varied tracking scenarios.

With the same tracking environment that we simulated, we try to find better parameters by simulating cross-validation. First, we find the parameter L, and α respectively. In this work, we focus on the RMSE of the T-UKS since we want to enhance its performance compared to the result of the UKS.

Figure 4.7 shows the result of finding the best parameter among $L = 4$, 5, 6 with the parameter α fixed as 0.4. Through the cross-validation simulation, we found out that the results of the thresholding process of both filtering and smoothing are better when the parameter $L = 6$. Then, we proceed the varied α using the computed parameter L.

Figure 4.8 shows the result of finding the best parameter among $\alpha = 0.36$, 40, 44 with the parameter L fixed as 6 as what we found in the previous simulation.

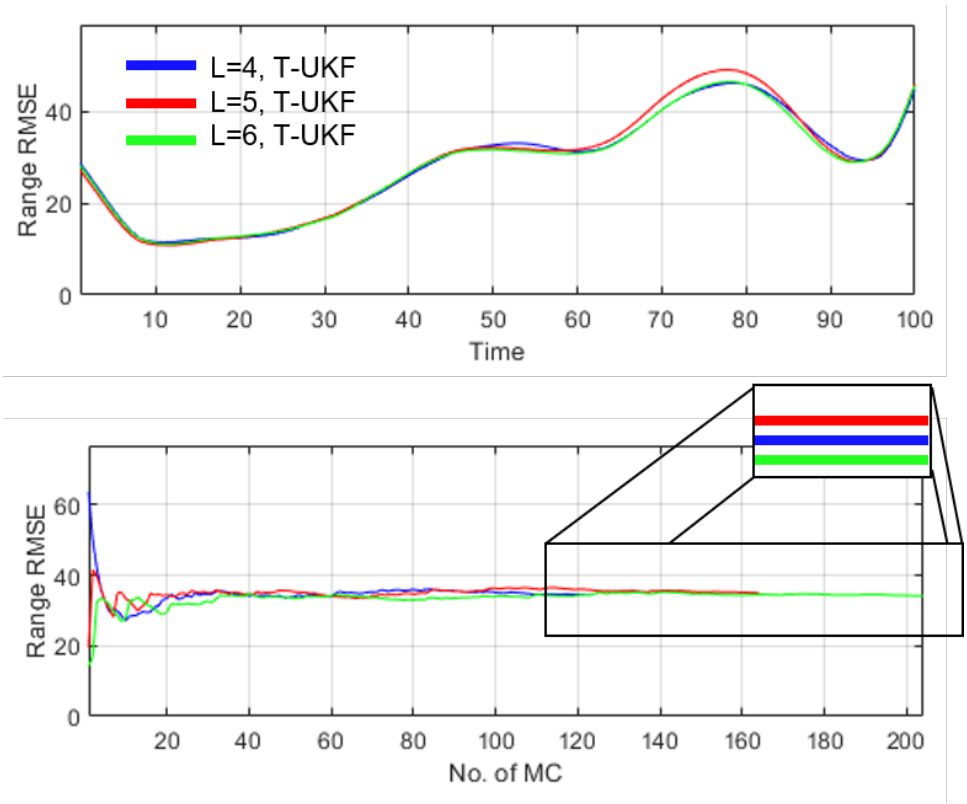


Figure 4.7: RMSE for Varied $L = 4, 5, 6$

Table 4.3 shows the RMSE of varied parameters respectively while we do the cross-validation. We can simulate this process recursively to find the optimal parameters and later set the proper parameters from the optimal parameters that we trained.

RMSE in samples for $\alpha = 0.4$	$L = 4$	$L = 5$	$L = 6$
UKS	35.2	35.2	**34.4**
T-UKS	34.5	35.0	**34.2**
RMSE in samples for $L = 6$	$\alpha = 0.36$	$\alpha = 0.40$	$\alpha = 0.44$
UKS	**33.3**	34.4	37.1
T-UKS	**32.9**	34.3	37.0

Table 4.3: RMSE in Samples For Varying L and α Values

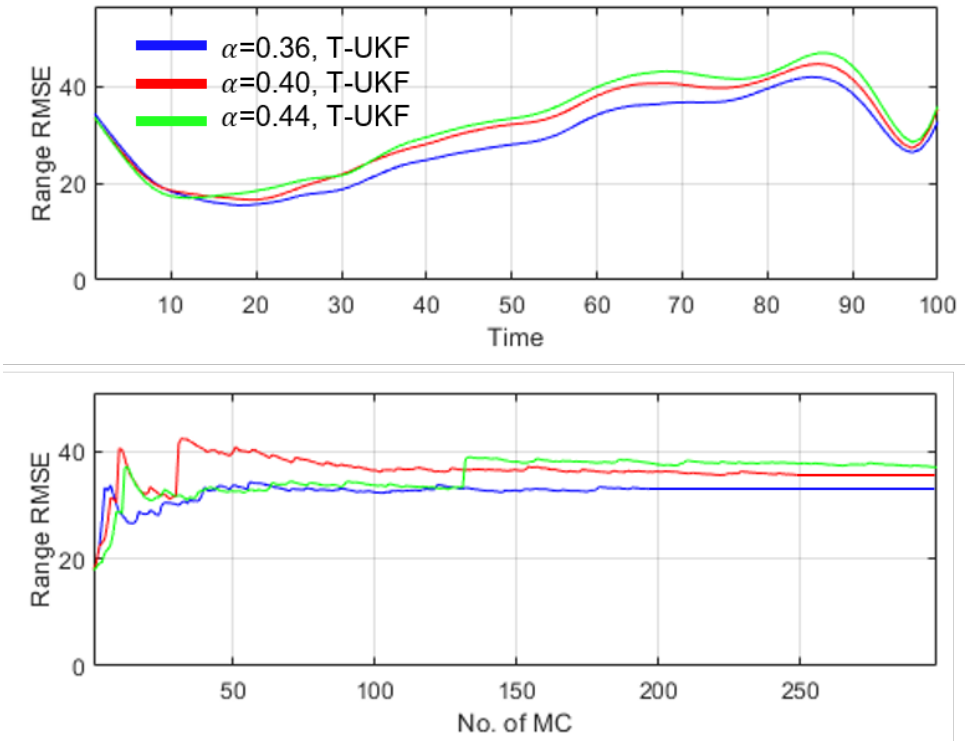


Figure 4.8: RMSE for Varied $L = 4, 5, 6$

Chapter 5

CONCLUSIONS

5.1 Conclusion

In this thesis, we considerd a nonlinear state space system formulation for tracking a moving object using sensor measurements with high noise and clutter due to high probability of false alarm conditions. We proposed a new tracking method that integrates Bayesian filtering and smoothing with a thresholding process to increase estimation accuracy for off line processing. In particular, after the unknown state is sequentially estimated over all time steps using the unscented Kalman filter (UKF), we use a process that is aimed to eliminate measurements that may have resulted from false alarms. The method compares the estimated state at each time step to a neighborhood of estimated states, both from previous and future time steps, using a distance-based metric. If the metric threshold is exceeded at a particular time step, then the measurement-updated state is replaced with the predicted value of the UKF process. After thresholding, we use unscented Kalman smoothing (UKS) to further improve the state estimation accuracy. Using simulations, we demonstrated that the proposed method works well in high noise conditions. The thresholding process performed well in finding and removing corrupted measurements due to high noise. On the other hand, the method does not improve performance under high clutter and low noise conditions.

5.2 Future Work

5.2.1 High-noise Attenuated Multi-object Smoothing

Many methods of multi-object filtering and smoothing have been studied and suggested for almost a century, and it is still a state-of-art topic in the field of signal processing in order to utilize it in practice, since not only considering the probable parameters such as survival, death, birth, detection, and a miss detection but managing and restraining uncertainty is a key factor in evaluating whether the method performs well.

The multi-object smoothing has been suggested relatively currently due to the low feasibility in practice. Recently, the generalized labeled multi-Bernoulli tracker with partial smoothing [23] was suggested to filter and smooth the multi-object case. With each labeled target, the tracker[23] smoothes each object partially and combines them. Based on the partial smoothing, applying the state estimations after a thresholding process can make the result better, as we suggest in this thesis, and we can extend it to multi-object tracking that uses the methods such as the cardinalized probability hypothesis density, generalized labeled multi-Bernoulli filter. Multi-object implies varied characteristics for each object, and we should find the proper threshold algorithm for each object. For instance, we expect an optimization algorithm to find the threshold for each object. Also, the object that is generated by clutter could be eliminated by a proper object lifetime threshold.

REFERENCES

[1] A. Yilmaz, O. Javed, and M. Shah, "Object tracking: A survey," *ACM Computing Surveys*, vol. 38, no. 4, pp. 13–es, 2006.

[2] S. Challa, M. R. Morelande, D. Mušicki, and R. J. Evans, *Fundamentals of Object Tracking*. Cambridge University Press, 2011.

[3] Y. Bar-Shalom, X. R. Li, and T. Kirubarajan, *Estimation with Applications to Tracking and Navigation*. John Wiley, 2001.

[4] S. Särkkä, *Bayesian Filtering and Smoothing*. Cambridge University Press, 2013.

[5] Z. Chen, "Bayesian filtering: From Kalman filters to particle filters, and beyond," *Statistics*, vol. 182, no. 1, pp. 1–69, 2003.

[6] H. W. H. W. Sorenson, ed., *Kalman Filtering: Theory and Application*. IEEE Press, 1985.

[7] R. E. Kalman, "A new approach to linear filtering and prediction problems," *Journal of Basic Engineering*, vol. 82, no. 1, pp. 35–45, 1960.

[8] B. M. Yu, K. V. Shenoy, and M. Sahani, "Derivation of extended Kalman filtering and smoothing equations," 2004. Online available at: `https://users.ece.cmu.edu/~byronyu/papers/derive_eks.pdf`.

[9] S. J. Julier, J. K. Uhlmann, and H. F. Durrant-Whyte, "A new approach for filtering nonlinear systems," in *American Control Conference*, vol. 3, pp. 1628–1632, 1995.

[10] E. A. Wan, R. Van Der Merwe, and S. Haykin, "The unscented Kalman filter," *Kalman Filtering and Neural Networks*, vol. 5, no. 2007, pp. 221–280, 2001.

[11] S. J. Julier and J. K. Uhlmann, "Unscented filtering and nonlinear estimation," *Proceedings of the IEEE*, vol. 92, no. 3, pp. 401–422, 2004.

[12] M. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking," *IEEE Transactions on Signal Processing*, vol. 50, no. 2, pp. 174–188, 2002.

[13] M. Johannes and N. Polson, "Particle filtering," in *Handbook of Financial Time Series*, pp. 1015–1029, Springer, 2009.

[14] B. Ristic, S. Arulampalam, and N. Gordon, *Beyond the Kalman Filter: Particle Filters for Tracking Applications*. Artech House Publishers, 2004.

[15] N. Wiener, *Extrapolation, Interpolation, and Smoothing of Stationary Time Series*. Wiley, New York, 1949.

[16] H. Rauch, "Solutions to the linear smoothing problem," *IEEE Transactions on Automatic Control*, vol. 8, no. 4, pp. 371–372, 1963.

[17] H. E. Rauch, F. Tung, and C. T. Striebel, "Maximum likelihood estimates of linear dynamic systems," *AIAA Journal*, vol. 3, no. 8, pp. 1445–1450, 1965.

[18] C. T. Leondes, J. B. Peller, and E. B. Stear, "Nonlinear smoothing theory," *IEEE Transactions on Systems Science and Cybernetics*, vol. 6, no. 1, pp. 63–71, 1970.

[19] H. E. Doran, "Constraining Kalman filter and smoothing estimates to satisfy time-varying restrictions," *The Review of Economics and Statistics*, vol. 74, pp. 568–572, 1992.

[20] S. Särkkä, "Unscented Rauch–Tung–Striebel smoother," *IEEE Transactions on Automatic Control*, vol. 53, no. 3, pp. 845–849, 2008.

[21] M. Briers, A. Doucet, and S. Maskell, "Smoothing algorithms for state-space models," *Annals of the Institute of Statistical Mathematics*, vol. 62, 2010.

[22] B.-N. Vo, B.-T. Vo, and R. P. Mahler, "Closed-form solutions to forward–backward smoothing," *IEEE Transactions on Signal Processing*, vol. 60, no. 1, pp. 2–17, 2011.

[23] T. T. D. Nguyen and D. Y. Kim, "GLMB tracker with partial smoothing," *Sensors*, vol. 19, no. 20, p. 4419, 2019.

[24] M. S. Lowe, *Extended and Unscented Kalman Smoothing for Re-linearization of Nonlinear Problems with Applications*. PhD thesis, Worcester Polytechnic Institute, 2015.

[25] Y. Bresler, "Two-filter formula for discrete-time non-linear Bayesian smoothing," *International Journal of Control*, vol. 43, pp. 629–641, 1986.

[26] G. Kitagawa, "Monte Carlo filter and smoother for non-Gaussian nonlinear state space models," *Journal of Computational and Graphical Statistics*, vol. 5, no. 1, pp. 1–25, 1996.

[27] S. Särkkä and J. Hartikainen, "Sigma point methods in optimal smoothing of non-linear stochastic state space models," in *IEEE International Workshop on Machine Learning for Signal Processing*, 2010.

[28] A. Doucet and A. M. Johansen, "A tutorial on particle filtering and smoothing: fifteen years later," 2011. Oxford Handbook of Nonlinear Filtering.

[29] M. Klass, M. Briers, N. de Freitas, A. Doucet, S. Maskell, and D. Lang, "Fast particle smoothing: If I had a million particles," in *International Conference on Machine Learning*, 2006.

[30] P. M. Marone and R. M. Di Maggio, "Forensic geophysics: ground penetrating radar (GPR) techniques and missing persons investigations," *Forensic Science Research*, vol. 4, pp. 337–340, 2019.

[31] S. M. Kay, *Fundamentals of Statistical Signal Processing Volume II: Detection Theory*. Prentice Hall PTR, 1998.

[32] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller, "Equation of state calculations by fast computing machines," *The Journal of Chemical Physics*, vol. 21, no. 6, pp. 1087–1092, 1953.

[33] J. E. Gentile, *Random Number Generation and Monte Carlo Methods*. Springer Verlag, 1998.

[34] A. Doucet, N. de Freitas, and N. Gordon, eds., *Sequential Monte Carlo Methods in Practice*. Springer-Verlag, 2001.

[35] P. M. Djurić, Y. Huang, and T. Ghirmai, "Perfect sampling: A review and applications to signal processing," *IEEE Transactions on Signal Processing*, vol. 50, 2002.

[36] C. P. Robert and G. Casella, *Monte Carlo Statistical Methods*. Springer-Verlag, 2 ed., 2004.

[37] J. V. Candy, *Bayesian Signal Processing: Classical, Modern, and Particle Filtering Methods*. Wiley, 2009.

[38] C. M. Bishop and N. M. Nasrabadi, *Pattern Recognition and Machine Learning*, vol. 4. Springer, 2006.

[39] W. K. Hastings, "Monte Carlo sampling methods using Markov chains and their applications," *Biometrika*, vol. 57, pp. 97–109, 1970.

[40] W. R. Gilks, S. Richardson, and D. Spiegelhalter, *Markov Chain Monte Carlo in Practice*. Chapman & Hall/CRC Interdisciplinary Statistics, 1995.

[41] P. A. Gagniuc, *Markov Chains: from Theory to Implementation and Experimentation*. John Wiley & Sons, 2017.

[42] C. P. Robert and G. Casella, *Metropolis–Hastings Algorithms*, ch. 6. Springer, New York, 2010.

[43] J. M. Bernardo and A. F. Smith, *Bayesian Theory*, vol. 405. John Wiley & Sons, 2009.

[44] S. M. Kay, *Fundamentals of Statistical Signal Processing Volume I: Estimation Theory*. Prentice Hall PTR, 1993.

[45] Y. Bar-Shalom and T. E. Fortmann, *Tracking and Data Association*. Academic Press, 1988.

[46] D. Arrowsmith and C. Place, "The linearization theorem," in *Dynamical Systems: Differential Equations, Maps, and Chaotic Behaviour*, pp. 77–81, Chapman & Hall, 1992.

[47] M. W. Hirsch and S. Smale, "Differential equations, dynamical," *Systems and Linear*, 1974.

[48] A. G. Terejanu, "Extended Kalman filter tutorial." `https://www.cse.sc.edu/~terejanu/files/tutorialEKF.pdf`, 2008. [Online; accessed 28-April-2022].

[49] S. J. Julier and J. K. Uhlmann, "New extension of the Kalman filter to nonlinear systems," in *Signal Processing, Sensor Fusion, and Target Recognition VI*, vol. 3068, pp. 182–193, 1997.

[50] H. M. T. Menega, Y. I. J. ao, G. A. Borges, and A. N. Vargas, "A systematization of the unscented Kalman filter theory," *IEEE Transactions on Automatic Control*, vol. 60, p. 2583–2598, 2015.

[51] J. S. Liu and R. Chen, "Sequential Monte Carlo methods for dynamic systems," *Journal of the American Statistical Association*, vol. 93, no. 443, pp. 1032–1044, 1998.

[52] N. J. Gordon, D. J. Salmond, and A. F. Smith, "Novel approach to nonlinear/non-Gaussian Bayesian state estimation," in *IEE Proceedings F-radar and Signal Processing*, vol. 140, pp. 107–113, 1993.

[53] J. S. Liu and R. Chen, "Blind deconvolution via sequential imputations," *Journal of the American Statistical Association*, vol. 90, no. 430, pp. 567–576, 1995.

[54] D. Simon, *Optimal State Estimation.* Wiley, 2006.

[55] J. Hartikainen, A. Solin, and S. Särkkä, "Optimal filtering with Kalman filters and smoothers, a manual for MATLAB toolbox EKF/UKF." `https://www.researchgate.net/publication/228683456`, 2011. [Online; accessed 28-April-2022].

[56] G. Kitagawa, "Non-Gaussian state—space modeling of nonstationary time series," *Journal of the American Statistical Association*, vol. 82, no. 400, pp. 1032–1041, 1987.