# VEHICLE TRACKING USING ULTRA-WIDEBAND RADAR

A Thesis
Presented to
The Academic Faculty

by

Andrew Leonard

In Partial Fulfillment
of the Requirements for the Degree
Master of Science in the
Woodruff School of Mechanical Engineering

Georgia Institute of Technology
December 2016

# VEHICLE TRACKING USING ULTRA-WIDEBAND RADAR

Approved by:

Dr. Jonathan Rogers, Advisor
Woodruff School of Mechanical Engineering
*Georgia Institute of Technology*

Dr. Mark Costello
The Daniel Guggenheim School of
Aerospace Engineering
*Georgia Institute of Technology*

Dr. Oliver Sawodny
Insitut für System Dynamik
*Universität Stuttgart*

Dr. Cristina Tarin
Insitut für System Dynamik
*Universität Stuttgart*

Date Approved: August 18, 2016

*To my parents.*

# ACKNOWLEDGEMENTS

As an inaugural member of the Joint Degree program between the Georgia Institute of Technology and the University of Stuttgart this thesis was made possible by the hard-work, support, and mentor-ship of numerous people across two continents. The last two years spent between Atlanta and Stuttgart have been truly memorable.

First and foremost, I would like to thank Dr. Neitzel, from Georgia Tech, and Dr. Sawodny, from the University of Stuttgart, for their hard-work creating and overseeing the program. From the first meeting in March of 2014 through the final presentation in August 2016, it has been a truly rewarding experience.

I owe sincere gratitude to my advisor, Dr. Jonathan Rogers, for his mentor-ship while at Georgia Tech. His advice and guidance helped ignite my research career, providing the foundation on which the work of this thesis was built. Thanks is also due to my lab-mate, Jonathan, who was always available for questions, discussions, and–most importantly–coffee.

At the University of Stuttgart I had the pleasure of being a member of the Institute of System Dynamics. Two members deserve special recognition for their support and guidance over the course of my one year tenure: Frank Bender, who helped tremendously during my transition to Stuttgart, and Florian Morlock, who provided constructive advice, insights, and corrections to my thesis and presentation.

The work in this thesis was performed during my time as a *Werkstudent* at Daimler AG in Sindelfingen, Germany. Most importantly I would like to thank my supervisor, Anton Feldmann, whose insights and experience were key to the project's success. Thanks must also be given to my other colleagues, Herr Walter, Paul, and Marius, who made my time at Daimler even more unforgettable.

# TABLE OF CONTENTS

# LIST OF TABLES

# LIST OF FIGURES

# SUMMARY

The goal of this thesis is to facilitate the detection and tracking of vehicles using an ultra-wideband radar sensor. Prior research of this topic is strongly focused on Collision Avoidance Systems (CASs) for use in specific situations such as highway driving. The target application of this thesis is one of lower speeds but greater uncertainity in the tracked-vehicles's entry and exit points. Therefore, different assumptions and considerations must be acknowledged. The focus of this thesis is on tracking vehicles, with no considerations given to possible collisions.

This thesis first investigates two widely-used nonlinear estimation techniques, the Extended Kalman Filter and the Particle Filter. It is found that unrestrictive framework of the Particle Filter is better suited for tracking problem in this thesis. With the target application in mind, a Particle Filter is developed to first detect new vehicles and then track them during a state-refinement stage. Pruning and gating techniques are developed to leverage the cluttered radar-sensor data in the measurement update procedures. Considering the quality and separation of the provided radar data, the data association step is simplified allowing mutiple-vehicle tracking. To extend the tracking system to be used on a moving platform, mappings are developed to allow the transformation of data from a moving radar frame-of-reference to an inertial frame, and vice versa. The mappings require knowledge of the moving platform's state, an Extended Kalman Filter is formulated to estimate the state from limited, noisy onboard-sensor data.

The tracking system is tested for both single and multi-vehicle tracking using real-world data collected by a short-range, ultra-wideband radar sensor. The results show that the developed system sufficiently tracks each vehicle in an uncontrolled and cluttered environment. Real-world data collected from a test vehicle shows that the developed Extended Kalman Filter provides a suitable state estimate from the limited available sensors. Further research is suggested to rigorously test the system and extend its capabilities.

# CHAPTER I

# INTRODUCTION

The task of tracking an object can be forumalated into a straightforward question: *using previous observations and an understanding of the object in question, can we predict where it will be in the future?*

In the context of the question posed above; to predict the future trajectory of an object, a suitable model must be defined and the current state of the object calculated. The available observations of the object are typically noisy and of a lower dimension than the state vector itself, so the state must be estimated. In essence, the task of tracking is analogous to state estimation performed remotely.

The chapter proceeds as follows. The next section provides a brief motivation to the work in the thesis and the main objectives. In section 1.2 the scope of the work, including the environment considered, data available, and limitations, is described. In section 1.3 a system to accomplish the objectives, the contribution of this thesis, is proposed. Next, a brief review of the state-of-the-art is provided. Finally, section 1.5 outlines the remaining chapters of this thesis.

## *1.1 Motivation and Objectives*

The ability to track an object is of great benefit to a vehicle safety system. If the future trajectory of an object can be predicted, conclusions can be drawn on whether a collision is imminent. This knowledge provides a base to determining whether corrective measures are neccessary. The specific application pursued in this thesis is a driver-less system in a controlled environment. While tracking is a natural and intuitive task for a human, systematic approaches are challenging and involved, and thus an active area of research.

In this thesis such an approach is developed to be applied in a vehicle safety system. The developed approach should provide a mechanism to derive useful object-state information, needed for trajectory generation, from otherwise inconsequential sensor data. As such, the objectives of the work are as follows.

First, the tracker must be able to handle suitable state dynamics, observation, and uncertainity models. The state dynamics model serves to predict the object's future state; thus, to accurately describe the object, the tracker should be able to leverage linear and nonlinear models. The observation model is the mechanism to map the state vector into a space whose axes are the available measurements. Again, to accurately define this mapping the tracker should accept linear and nonlinear models. The uncertainity model represents our trust, or distrust, in the previous two models through process noise and oberservation noise, respectively. The tracker should accept noise sampled from arbitrary densities.

Second, the tracker should expect to run continuously and in real-time, with objects entering and exiting the system's environment. Thus, a mechanism for track creation and deletion must be provided. In this sense, the tracker is responsible for first detecting the object before tracking it.

Third, the tracker should possess the ability to track multiple objects. Given a sufficient field of view one can expect multiple objects to be present at one time. The system should concurrently handle detection and tracking with a method seamlessly switch when a vehicle has been detected.

Finally, the system should be extendable to a moving platform; that is, the observations made from a moving ego-vehicle, which we will refer to as the radar-vehicle. The system must be able to transform the observations from the vehicle reference frame to an inertial frame where the objects of interest reside; this requires knowledge of the radar-vehicle's state. Elements from the first objective can be used to perform this local state estimation.

## *1.2 Scope of Work*

To further develop the foundation of this thesis, the scope of the work must be considered. This includes a description of the environment considered, available sensors, and other limitations.

The application that the system is targeted towards is for use on the grounds of a manufacturing plant. To observe the environment an Ultra-Wideband (UWB) radar sensor is employed. The sensor, an SRR-208 produced by Continental Engineering, provides range, azimuth, Radar Cross Section (RCS), and relative velocity data on upto 128 objects. The data, transmitted via a CAN bus, is updated every 66 milliseconds[1].

While people, bikes, and vehicles can all be found traversing the area, and thus could be considered as objects, the work in this thesis focuses on tracking vehicles. This is due in part to limitations of the radar sensor, which has shown difficulty in detecting slow, non-metallic objects.

From the radar-vehicle, individual wheel speeds, steering angle, and location data is available. The location data is provided by an auxillary Real Time Kinetic(RTK) GPS, rather than the standard system available in the vehicle. The RTK system provides accuracies in the centimeter range[42]. The GPS data is updated every 20 milliseconds while the vehicle data is updated every 5 milliseconds. In the targeted application, the radar-vehicle will travel slowly (<12 km/h) and without rapid maneuvers.

To test the ability to estimate the ego-vehicle's state collected data is used. To test the tracking system simulated and collected data is used. In both cases the ego-vehicle is stationary. The method to extend the system to a moving vehicle, linking the two endeavours, is discussed but left open for a future research endeavour.

## *1.3 Method*

The approach taken to develop the tracking system is as follows. To track other vehicles within the radar-vehicle's environment a particle filter (PF) is employed. The framework provided by the Monte Carlo based filter allows for the use of a nonlinear vehicle model, nonlinear observation model, and arbitrary noise distributions. The ability to use non-Gaussian noise in the observation model is critical. Additionally, the particle filter framework provides a mechanism, with some limitations, to detect when a new vehicle is being tracked.

To estimate the state of the radar-vehicle an Extended Kalman Filter (EKF) has been chosen. The EKF allows for the use of nonlinear vehicle and observation models, but typically considers the process and measurement noise to be Gaussian. This assumption fits well with the available measurements and desired output. To transform the radar data to the inertial frame, position, orientation, velocity, and yaw rate must be estimated.

The methods listed above both fall under the Bayesian estimation designation, which will be developed in Chapter 2. Both frameworks will be developed such that a clear connection and comparision can be made between the two allowing for a clear choice for the proposed application.

## *1.4 State of the Art*

The area of state estimation, especially in the context of vehicle tracking, is currently a very heavily-researched field. The ability to detect and track vehicles is of great interest in Driver Assistance Systems (DAS). The detection and tracking methods employed by these systems vary with intent, application, and available sensors. A comprehensive review of current methodologies and sensors is provided by Mukhtar et al.[34]. Common sensors used in such systems are Radar and Laser or LIDAR based, which are considered *active*, and vision based, which are considered *passive.*

Vision-based systems are commonly deployed due to their low cost and detection quality but suffer from issues of complexity and sensitivity to external factors. The complexity in vision-based methods is rooted in the need to detect features and vehicles from the image data to obtain location measurements[41]. Alternatively, Radar-based systems provide location, and sometimes velocity, measurements directly. The advantages of both sensors can be leveraged by fusion algorithms, such as the one developed by Feng et al. in [28].

Using a radar sensor, the task of tracking is commonly handled by a framework such as the Kalman Filter (KF), the Extended Kalman Filter (EKF), or the Particle Filter (PF). Floudas et al. compare several variations of these frameworks in [15]. The variations include different methods to linearize the–typically nonlinear–measurement space for KF implementations and different resampling/linearization methods for the PF implementations.

The particle filter implementations for vehicle tracking have been investigated by a number of researchers, for example [19, 29]. Considering the broader cateogory of object tracking, the particle filter framework has been used in applications such as air traffic control[30] and ship tracking[9].

When transitioning to tracking multiple vehicles, or objects, especially in noisy or cluttered environments, emphasis is placed on the data association step [18, 36, 23, 32]. While this is branch of research unto itself, the focus of this thesis is tracking and estimation.

Much of the state-of-the-art is targeted towards specific applications and environments different than the one considered in this thesis. Additionally, many of the developed systems possess a level of complexity and sophistication beyond the scope of this thesis. A system targeted towards the application specified above is developed from the ground up, beginning with an investigation of state estimation techniques.

## *1.5 Outline of Work*

The thesis proceeds as follows. The first chapter provides a background on state estimation with a focus on Kalman and particle filtering techniques. Chapter II develops the system using elements from Chapter I and addressing application-specific objectives. Chapter III presents the data and discusses the filters' effectiveness. Finally, a Conclusion provides a review of the overall system and future areas of research and development.

# CHAPTER II

# STATE ESTIMATION METHODS

This chapter presents and develops available state estimation methods in an effort to determine which could be used in the tracking and radar-vehicle localization problems. In the context of the described problems, we are attempting to infer information about the current vehicle state, $s_t$, from the set of previous measurements, $Y = \{y_i\}_{i=0}^{n}$, which are provided by either the radar or onboard vehicle sensors.

This chapter is structured as follows. First, a description of the system and observation models that are considered is provided. The uncertainity of the measurements and models is then viewed within a Bayesian framework to provide systematic estimation methods. Next, the Kalman Filter is developed for both linear and nonlinear systems. Finally, the nonlinear estimation problem is approached using sequential Monte Carlo methods, known as a particle filter.

## *2.1 System and Observation Models*

The words *state* and *model* have already been used quite often in this thesis and yet an explaination of what is intended has not been offered, until now. Considering a state-space representation of the system, the *system* model describes the dynamics of the system, or how we expect the vehicle to move over time. The *observation* model is a mapping of the state-space to the domain of the measurements. In this thesis, both linear and nonlinear models are considered but limited to the discrete-time case; the models are given in Table 1. For methods on determining discrete-time systems from continuous time representations see [20].

In the linear case of the system model, $F_k$ is the state transition matrix and $B_k$ the input matrix. For the nonlinear case, $f(s_k, u_k)$ is an arbitrary function of the state

Table 1: System and Observation model types considered

| | Linear | Nonlinear |
|---|---|---|
| System Model | $s_{k+1} = F_k s_k + B_k u_k + w_k$ | $s_{k+1} = f(s_k, u_k) + w_k$ |
| Observation Model | $z_k = H_k s_k + v_k$ | $z_k = h(s_k) + v_k$ |

vector and input vector, respectively. In the linear case of the observation model, $H_k$ is the measurement model matrix, while in the nonlinear case $h$ is an arbitrary function of $s$. It is important to note that the variable $z$ is not just a selection of elements of $s$ but rather the result from any mapping between the two spaces. In the linear sense this amounts to any linear combination of the state vector elements.

Included in the system and observation models is the additive process and oberservation noise terms $w_k$ and $v_k$, respectively. The process noise, $w \in \mathcal{R}^n$, serves to accumulate the errors from the processes not included in the model; it is essentially a measure of trust in the system model. The measurement noise, $v \in \mathcal{R}^m$, serves to represent the inherent uncertainity involved in the measurement process; this is the measure of trust in the sensors. These two entities are what we have named the uncertainty model.

## 2.2 Bayesian Framework

To adequately represent the uncertainity about the true state of our system, a stochastic approach is taken. That is, the elements of the finite-length vehicle state vector, $s$, are each considered random variables. A joint probability density function (pdf) is used to describe the relative likelihood that a realization of the state is in fact the true state. This joint pdf yields an estimate of the vehicle state can be obtained using various estimators.

This estimation will take part within a Bayesian framework, which is based upon the work of Reverend Thomas Bayes[33]. Bayes Theorem, which famously reads as

$$p(A \mid B) = \frac{p(B \mid A)\, p(A)}{p(B)}, \tag{1}$$

The quantity $p(A \mid B)$ is known as the *posterior* density and is the probability of event $A$ occuring given that event $B$ has already occured. Conversely, $p(B \mid A)$ is the probability of event $B$ occuring given event $A$. The quantities $p(A)$ and $p(B)$ are the probabilities of events $A$ and $B$ ocurring without regard to the other, respectively.

In the context of state estimation, Bayes theorem provides a mechanism to systematically update the vehicle state pdf using prior knowledge given by a model and measurements. Given the set of states $s_{0:k} = \{s_0, \ldots, s_k\}$ and set of measurements $y_{1:k} = \{y_1, \ldots, y_k\}$, the posterior density at any time $k$ can be written as,

$$p(s_{0:k} \mid y_{1:k}) = \frac{p(y_{1:k} \mid s_{0:k}) p(s_{0:k})}{\int p(y_{1:k} \mid s_{0:k}) p(s_{0:k}) ds_{0:k}} \tag{2}$$

The posterior, $p(s_{0:k} \mid y_{1:k})$, can be interpreted as a measure of the relative likelihood that the discrete-time trajectory, $s_{0:k}$, is the true vehicle trajectory given the observed measurements, $y_{1:k}$.

Inherent in the state-space formulation specified in section 2.1 is that the systems are Markov chains. As such, when predicting $s_{k+1}$ the previous state $s_k$ provides the same information as the set of all previous states $s_{0:k}$. The expression in equation (2) can then be rewritten in a recursive fashion as,

$$p(s_{0:k+1} \mid y_{1:k+1}) = \frac{p(y_{k+1} \mid s_{k+1}) p(s_{k+1 \mid k})}{p(y_{k+1} \mid y_{1:k})} \tag{3}$$

Following the same principle as above, the marginal density of $s_k$ from the above posterior can also be expressed in a recursive manner. When separated into *prediction* and *update* steps it can be written as

$$\textit{predict}: \quad p(s_k \mid y_{1:k-1}) = \int p(s_k \mid s_{k-1}) p(s_{k-1} \mid y_{1:k-1}) ds_{k-1} \tag{4}$$

$$\textit{update}: \qquad p(s_k \mid y_{1:k}) = \frac{p(y_k \mid s_k)\, p(s_k \mid y_{1:k-1})}{\int p(y_k \mid s_k)\, p(s_k \mid y_{1:k-1}) ds_k} \tag{5}$$

The marginal density of the posterior describes the uncertainity in the vehicle state, rather than trajectory, at time $k$ given all previous measurements and is the density from which the state estimate will be calculated. For simplicity the marginal designation is dropped when describing the density for the remainder of this thesis. The result of equation (4), $p(s_k \,|\, y_{1:k-1})$, will be called the *prior* density and the result of equation (5), $p(s_k \,|\, y_{1:k})$, will be called the *posterior* density.

For a deeper formulation of the Bayes filtering problem interested readers can consult [13].

## *2.3 The Kalman Filter*

The Kalman Filter is aptly named after R.E. Kalman, the author of the 1960 paper: *A New Approach to Linear Filtering and Prediciton Problems*[26]. In his paper, Kalman provides an optimal solution to the linear quadratic estimation (LQE) problem.

As specified in the Bayesian framework, the vehicle state, $s$, is a random vector described by the joint pdf, $p(s)$. Within the Kalman Filter the posterior of the pdf is updated recursively, conditioned on the current measurement and prior distribution. While only the current measurement is explcitly stated during the update procedure, because of the recursive nature of the algorithm the posterior density contains information from all previous measurements.

The Kalman Filter is developed below, first in its original, linear form and then in the nonlinear form, which is known as the Extended Kalman Filter(EKF).

### 2.3.1 Linear Kalman Filter

To begin, it is assumed the system and measurement models are linear, as described in Table 1:

$$s_{k+1} = F_k s_k + B_k u_k + w_k \tag{6}$$

$$z_k = H_k s_k + v_k. \tag{7}$$

Additionally, it is to be assumed that the process and measurement noises are multivariate zero-mean Gaussian white noise vectors with covariance matrices $Q_k$ and $R_k$, respectively. That is, $w \sim \mathcal{N}(0, Q_k)$ and $v \sim \mathcal{N}(0, R_k)$, where

$$Q_k = E[w_k^\intercal w_k] \tag{8}$$

$$R_k = E[\, v_k^\intercal \, v_k]. \tag{9}$$

Now, as specified in the Bayesian framework, the vehicle state, $s$, is a random vector described by the joint pdf, $p(s)$. Within the Kalman Filter the joint pdf is characterized by a state estimate, $\hat{s}$, and error covariance matrix $P$. The error covariance matrix is a measure of uncertainity in the state estimate. This amounts to,

$$e_k = s_k - \hat{s}_k \tag{10}$$

$$P_k = E[\, e_k e_k^\intercal \,]. \tag{11}$$

To develop the algorithm, consider that at time $k$ the state estimate, $\hat{s}_{k|k}$, and error covariance matrix, $P_{k|k}$, are known and contain the information provided by all previous measurements. The quantities can then be propagated forward in time according to equations (12) and (13). This is the prediction step, where the resulting estimate and covariance matrix describe the prior density.

$$\hat{s}_{k+1|k} = F_k \hat{s}_{k|k} + B_k u_k \tag{12}$$

$$P_{k+1|k} = F_k P_{k|k} F_k + Q_k. \tag{13}$$

When a new measurement is available the posterior density is calculated, conditioned on the new measurement and the prior density; this is the update step. The corresponding posterior estimate and error covariance matrix are determined using equations (15) and (16), respectively,

$$y_k = z_k - H_k \hat{x}_{k|k-1} \tag{14}$$

$$\hat{s}_{k|k} = \hat{s}_{k|k-1} + K_k y_k \tag{15}$$

$$P_{k|k} = (I - K_k H_k) P_{k|k-1}. \tag{16}$$

Equation (14) is known as the *innovation*, which is a measure of the error between the measurement, $z_k$, and the state estimate mapped into the measurement-space. This measure is weighted by the Kalman gain, $K_k$, which is the foundation of the Kalman Filter.

The Kalman gain represents the *optimal* linear filter gain; that is, it returns the Minimum Mean-Squared Error(MMSE) estimate of the posterior distribution. Recalling the error function from equation (10), the MMSE estimator can be written as

$$\begin{aligned} \hat{s}_k^{MMSE} &= \arg\min_{\hat{s}_k} \; E[\, \|e_k\|^2] \\ &= \arg\min_{\hat{s}_k} \; E[(s_k - \hat{s}_k)(s_k - \hat{s}_k)^\intercal] \end{aligned} \tag{17}$$

The argument of equation (17), the Mean-Squared Error(MSE), is also equivalent to the trace of the error covariance matrix, $\mathrm{Tr}\,[P_k]$. Substituting in the equations for the posterior estimate, measurement model, and prior error covariance, the posterior error covariance can now be written as

$$P_{k|k} = P_{k|k-1} - K_k H_k P_{k|k-1} - P_{k|k-1} H^\intercal K_k^\intercal + K_k (H_k P_{k|k-1} H_k^\intercal + R_k) K_k^\intercal \tag{18}$$

To determine the optimal gain matrix the trace of equation (18) can be differentiated with respect to $K_k$ and set equal to zero,

$$\frac{\partial \, \mathrm{Tr}[P_{k|k}]}{\partial K_k} = -2(H_k P_{k|k-1})^\intercal + 2K_k (H_k P_{k|k-1} H_k^\intercal + R_k) = 0 \tag{19}$$

Solving for $K_k$ yeilds

$$K_k = P_{k|k-1} H_k^\intercal (H_k P_{k|k-1} H_k^\intercal + R_k)^{-1}. \tag{20}$$

At this point, all of the necessary steps of the Kalman Filter have been developed. Table 2 summarizes the key Kalman Filter quantities derived above.

Table 2: Recursive steps for linear Kalman Filter

| **Linear Kalman Filter Algorithm** | | |
|---|---|---|
| 0 | Initialize: | $k = 0,\ Given:\ \hat{s}_{0\,|\,0},\ P_{0\,|\,0}$ |
| | **Predict** (*prior*) | |
| 1 | State estimate: | $\hat{s}_{k+1\,|\,k} = F_k \hat{s}_{k|k} + B_k u_k$ |
| 2 | Error covariance: | $P_{k+1\,|\,k} = F_k P_{k|k} F_k + Q_k$ |
| | **Update** (*posterior*) | |
| 3 | Measurment: | $z_k, \quad set\ k = k + 1$ |
| 4 | Innovation: | $y_k = z_k - H_k \hat{s}_{k\,|\,k-1}$ |
| 5 | Kalman Gain: | $K_k = P_{k\,|\,k-1} H_k^\intercal (H_k P_{k\,|\,k-1} H_k^\intercal + R_k)^{-1}$ |
| 6 | State estimate: | $\hat{s}_{k\,|\,k} = \hat{s}_{k\,|\,k-1} + K_k y_k$ |
| 7 | Error covariance: | $P_{k\,|\,k} = (I - K_k H_k) P_{k\,|\,k-1}$ |
| 8 | Return: | $goto\ step\ 1$ |

### 2.3.2 Extended Kalman Filter

While the Kalman Filter was developed for estimation of linear systems, most real-life situations of interest are not linear. The Extended Kalman Filter(EKF) was developed to do exactly as the name implies; *extend* the applicability of the Kalman Filter to nonlinear systems[31]. The filter is developed below to provide a conceptual understanding, interested readers may turn to [16] for a deeper analysis.

Given the nonlinear models from Table 1,

$$\hat{s}_{k+1} = f(\hat{s}_k, u_{k+1}) + B_k u_{k+1} + w_k \tag{21}$$

$$z_k = h(\hat{s}_k) + v_k \tag{22}$$

an approximation of the true state and measurement vectors can be expressed by a linearized system about the current state estimate using a first-order Taylor series expansion. The approximated quantities are,

$$s_{k+1} \approx f(\hat{s}_k) + F_k(s_k - \hat{s}_k) \tag{23}$$

$$z_k \approx h(\hat{s}_k) + H_k(s_k - \hat{s}_k) \tag{24}$$

where $F_k$ and $H_k$ are the Jacobians of the system and measurement models, respectively, evaluated at the current state estimate.

$$F_k = \left.\frac{\partial f}{\partial s}\right|_{s=\hat{s}_k} \tag{25}$$

$$H_k = \left.\frac{\partial h}{\partial s}\right|_{s=\hat{s}_k} \tag{26}$$

The standard Kalman Filter can now be applied to the linearized system, a summary of the steps is available in Table 3. It should be noted that the Extended Kalman Filter algorithm applies the nonlinear transformation to the state estimate during the prediction and innovation steps. Since the linearized transformation is used to update the corresponding error covariance matrix, there is an inherent deviation from the true posterior and therefore the EKF is *not optimal*. When the system is linear, the first-order approximation is equivalent and there is no deviation from the true distribution and the filter collapses back down to the standard Kalman Filter.

Table 3: Recursive steps for the Extended Kalman Filter

| **Extended Kalman Filter Algorithm** | | |
|---|---|---|
| 0 | Initialize: | $k = 0,\ Given:\ \hat{x}_{0\mid 0},\ P_{0\mid 0}$ |
| | **Predict** (*prior*) | |
| 1 | Linearize $f$: | $F_k = \frac{\partial f}{\partial x}\Big\rvert_{x=\hat{x}_{k\mid k}}$ |
| 2 | State estimate: | $\hat{x}_{k+1\mid k} = f(\hat{x}_{k\mid k}, u_{k+1})$ |
| 3 | Error covariance: | $P_{k+1\mid k} = F_k P_{k\mid k} F_k + Q_k$ |
| | **Update** (*posterior*) | |
| 4 | Measurment: | $z_k, \quad set\ k = k+1$ |
| 5 | Linearize $h$: | $H_k = \frac{\partial h}{\partial x}\Big\rvert_{x=\hat{x}_{k+1\mid k}}$ |
| 6 | Innovation: | $y_k = z_k - h(\hat{x}_{k\mid k-1})$ |
| 7 | Kalman Gain: | $K_k = P_{k\mid k-1} H_k^{\intercal} (H_k P_{k\mid k-1} H_k^{\intercal} + R_k)^{-1}$ |
| 8 | State estimate: | $\hat{x}_{k\mid k} = \hat{x}_{k\mid k-1} + K_k y_k$ |
| 9 | Error covariance: | $P_{k\mid k} = (I - K_k H_k) P_{k\mid k-1}$ |
| 10 | Return: | $goto\ step\ 1$ |

The Extended Kalman filter, while loosening the requirement that the system and observation models are linear, is still restricted in the variety of situations to which it can be applied. Strong nonlinearities in the system model can cause the filter to diverge or converge to an incorrect solution[25]. The filter is also still restricted by the assumption of Gaussian noise.

## *2.4 The Particle Filter*

The particle filter (PF), also known as the *bootstrap filter*, is part of a simulation-based class of filters employing Sequential Monte Carlo (SMC) methods. The filters use Monte Carlo (MC) integration to solve the high dimensional integrals found equations (4) and (5). The advantage of SMC methods is the absence of restrictions on the model and noise types. That is, the framework can handle nonlinear–including highly nonlinear–models and any arbitrary noise form. The price of this unrestricted framework, however, is in the cost of the required computational power.

In the remainder of this section the particle filter is developed such that its applicability to the tracking problem can be analyzed. For an indepth formulation of SMC methods, including the particle filter, interested readers are directed to [7] and [14].

### 2.4.1 Sequential Monte Carlo (SMC) methods

**Importance Sampling**

As stated above, the basis of SMC is the use of MC integration. MC integration is useful when solving high-dimensional, definite integrals; such as when calculating an estimate of a distribution. Rather than discretizing the space over which the integral is taken, $\Omega$, MC integration evaluates the integrand at random points. If the random points $\bar{x} = \{x_1, \ldots, x_N \in \Omega\}$ are drawn from the distribution $p$, the definite integral can be approximated as

$$\int_{\Omega} h(\bar{x})p(\bar{x})dx_1, \ldots, dx_n \approx \frac{1}{N}\sum_{i=1}^{N} h(x_i) \tag{27}$$

In many situations, however, drawing samples from the distribution $p(x)$ is intractable or simply not possible because the distribution is not known in closed form. Instead, samples can be drawn from a proposal distribution, $\pi(x)$. This process is known as *importance sampling*,

$$\int_{\Omega} h(x)p(x)dx_1, \ldots, dx_n \approx \frac{1}{N}\sum_{i=1}^{N} \frac{h(\bar{x}_i)}{\pi(\bar{x}_i)} \tag{28}$$

In the law of large numbers, the approximation tends towards the exact integral as $N$ approaches inifinty,

$$\frac{1}{N}\sum_{i=1}^{N} \frac{h(\bar{x}_i)}{\pi(\bar{x}_i)} \underset{N\to\infty}{\Rightarrow} \int_{\Omega} h(x)p(x)dx_{1:n} \tag{29}$$

In the context of the vehicle-state estimation problem, importance sampling MC integration can be applied to the posterior distribution to obtain an estimate according to some function $f_k$,

$$I(f_k(s_k)) = E_{p(s_k \,|\, y_{1:k})}[f_k(s_k)] \triangleq \int f_k(s_k)p(s_k \,|\, y_{1:k})ds_k \tag{30}$$

Sampling from the exact posterior distribution $p(s_k \,|\, y_{1:k})$ is not possible though, so a proposal distribution, $\pi(s_k \,|\, x_{0:k-1}, y_{1:k})$ must be used. Following Doucet in [13], the integral can be written as,

$$I(f_k(s_k)) = \int f_k(s_k)\frac{p(s_k \,|\, y_{1:k})}{\pi(s_k \,|\, s_{0:k-1}, y_{1:k})}\pi(s_k \,|\, s_{0:k-1}, y_{1:k})ds_k \tag{31}$$

$$q(s_k) = \frac{p(s_k \,|\, y_{1:k})}{\pi(s_k \,|\, s_{0:k-1}, y_{1:k})} \tag{32}$$

where $q(s_k)$ is known as the importance weight. The proposal distribution is left–for now–as a general distribution of $s_k$ given all previous states and measurements. The choice of what specific distribution to use will be discussed later in section 2.4.2.

Assuming that $N$ independent and identically distributed (i.i.d) vehicle-state samples, or particles, $\{s_k^{(i)}\}_{i=1}^{N}$ can now be drawn from the proposal distribution, $\pi(s_k \,|\, s_{0:k-1}, y_{1:k})$, the MC estimate of the integral is,

$$\hat{I}(f_k(s_k)) = \frac{\frac{1}{N}\sum_{i=1}^{N} f_k(s_k^{(i)})\, q(s_k^{(i)})}{\frac{1}{N}\sum_{i=0}^{N} q(s_k^{(i)})} = \sum_{i=0}^{N} f_k(s_{0:k}^{(i)})\tilde{q}_k^{(i)} \tag{33}$$

$$\tilde{q}_k^{(i)} = \frac{q(s_k^{(i)})}{\sum_{j=1}^{N} q(s_k^{(j)})} \tag{34}$$

where $\tilde{q}_k^{(i)}$ is the normalized importance weight. The specifics of this normalization can be found in Schön[40]. Equations (33) and (34) highlight the importance weights' role as a *relative* measure of importance of each particle to the estimate of the distribution.

**Sequential Importance Sampling (SIS)**

The importance weights in their current form must be recomputed with each new measurement and therefore are not ideal for recursive estimation. Following Schön[40], equation (34) can be rewritten in a recursive fashion such that the updated value is a function of the previous state, previous weight, and current measurement.

In order to express the integral in a recursive form it must first be assumed that the proposal distribution be of the form

$$\pi(s_k \mid s_{0:k-1}, y_{1:k}) = \pi(s_0) \prod_{i=1}^{k} \pi(s_i \mid s_{0:i}, y_{1:i}). \tag{35}$$

This assumption allows the importance weights in equation (34) to be rewritten as

$$\tilde{q}_k^{(i)} \propto \tilde{q}_{k-1}^{(i)} \frac{p(y_k \mid s_k^{(i)})\, p(s_k^{(i)} \mid s_{k-1}^{(i)})}{\pi(s_k^{(i)} \mid s_{0:k-1}^{(i)}, y_{1:k})} \tag{36}$$

which is proportional up to a normalizing constant. Therefore, after each update the weights need to be normalized.

The SIS algorithm has a well documented problem known as *degeneracy*[27]. It can be shown that the variance of the importance weights is an increasing function of $k$. Thus, after a number of iterations–increasing $k$–all of the importance weight will be held by a single particle[2]. This phenomenom seriously degrades the estimate and is a waste of computational resources.

### 2.4.2 Sequential Importance Resampling (SIR)

In 1993 Gordon et al. proposed the origninal bootstrap filter in the paper: *Novel approach to nonlinear/no-Gaussian Bayesian state estimation*[17]. The bootstrap filter, which in this thesis is to be considered synonomous to the particle filter, introduces a resampling step to effectively eliminate particles with low importance weights, mitigating the variance issues of the SIS algorithm.

To resample, a new set of particles is drawn with replacement from the current set according to

$$\Pr(s_{k,new}^{(i)} = s_{k,curr}^{(i)}) = \tilde{q}_k^{(i)}, \quad i = 1, \ldots, N \tag{37}$$

with each particle's importance weight set to $\frac{1}{N}$. The resampling occurs after the measurement update, which updates the importance weights based on the most recent measurement. Thus, the resampling process can be seen as a measure to remove unlikely particles effectively concentrating computational resources on the most probable particles.

The original particle (bootstrap) filter proposed by Gordon performed the resampling step every iteration. This in itself is a computationally expensive process, so for efficiency it should only be performed when deemed necesary.

It is beneficial to introduce a metric to measure the variance of the importance weights. One such measure is the effective sample size, which is defined by Bergman[6] as

$$N_{eff} \triangleq \frac{N}{1 + E_{\pi(\cdot \mid y_{1:k})}[q_k^2]}. \tag{38}$$

This value cannot be computed exactly, but can be estimated by

$$\hat{N}_{eff} = \frac{1}{\sum_{i=1}^{N} (\tilde{q}_k^{(i)})^2}. \tag{39}$$

When all of the particles are uniformly weighted, $\hat{N}_{eff}$ is equal to $N$. Conversely, when all of the normalized importance weight is held by a single particle, $\hat{N}_{eff}$ is equal to one. Thus, a relative scale of the variance in the samples' importance weights is established. Resampling can be deemed necessary when $\hat{N}_{eff}$ is below a set threshold, $N_{th}$.

To complete the recursive algorithm a satifactory proposal density must be identified. Recalling equations (31) and (32), the proposal density $\pi(s_k \mid s_{0:k-1}, y_{1:k})$ is left as an arbitrary quantity given that it fits the form defined in equation (35). From equation (31) it can be seen that the optimal density would indeed be $p(s_k|y_{1:k})$, which is equivalent to $p(s_k \,|\, s_{k-1}, y_{1:k})$ given the Markovian property of the models. Sampling from this density, if possible, is difficult and in most cases involves additional assumptions. Interested readers may consult Arulampalam et. al[2] for a more indepth analysis.

Gordon proposed that the prior could be used as an approximation of the optimal proposal density:

$$\pi(s_k \,|\, s_{0:k-1}, y_{1:k}) = p(s_k \,|\, s_{k-1}). \tag{40}$$

This proposition, when subsituted into the recursive importance weight expression (equation (36)), yields

$$\tilde{q}_k^{(i)} \propto \tilde{q}_{k-1}^{(i)} \frac{p(y_k \,|\, s_k^{(i)})\, p(s_k^{(i)} \,|\, s_{k-1}^{(i)})}{\pi(s_k^{(i)} \,|\, s_{0:k-1}^{(i)}, y_{1:k})} = \tilde{q}_{k-1}^{(i)} p(y_k \,|\, s_k^{(i)}). \tag{41}$$

This choice is intuitive and easy to implement given the recursive framework. From the set of particles $s_{k-1} = \{s_{k-1}^{(i)}\}_{i=1}^{N}$ from the previous recursion, or initialization, the prior can be found by propagating each particle through the system model

$$s_k = f_{k-1}(x_{k-1}^{(i)}, u_k) + w_{k-1}^{(i)}, \quad i = 1, \ldots, N. \tag{42}$$

At this point the necessary steps in the particle filter algorithm have been developed. A summary of the algorithm can be found in Table 4.

Table 4: Recursive steps for the Particle Filter (SIR)

| | **Particle Filter (SIR) Algorithm** | |
|---|---|---|
| | **Initialize** | |
| | Given: | $k = 0,\ p(x_0)$ |
| 0 | Draw N samples: | $(x_0^{(i)}, q_0^{(i)}) \sim p(x_0),\ \text{\textit{for}}\ i = 1, \ldots, N$ |
| | **Propagate Posterior** | |
| 1 | Update Particles: | $x_{k+1\,\|\,k}^{(i)} = f(x_{k\,\|\,k}^{(i)}, u_{k+1}) + w_k^{(i)},\ \text{\textit{for}}\ i = 1, \ldots, N$ |
| | **Measurement Update** | |
| 2 | Set time: | $k = k + 1$ |
| 3 | Apply measurement: | $\tilde{q}_k^{(i)} \propto \tilde{q}_{k-1}^{(i)}\, p(z_k \mid x_{k\,\|\,k-1}^{(i)}),\ \text{\textit{for}}\ i = 1, \ldots, N$ |
| 4 | Normalize weights: | $\tilde{q}_k^{(i)} = \frac{\tilde{q}_k^{(i)}}{\sum_{j=1}^{N} \tilde{q}_k^{(j)}},\ \text{\textit{for}}\ i = 1, \ldots, N$ |
| 5 | Calculate $\hat{N}_{eff}$: | $\hat{N}_{eff} = \frac{1}{\sum_{i=1}^{N} (\tilde{q}_k^{(i)})^2}$ |
| | **Resample** | |
| 6 | Check $\hat{N}_{eff}$: | $if\, \hat{N}_{eff} < N_{th},\ continue.\ Else, goto\, step\ 1$ |
| 7 | Sample with replacement: | $x_k^{(i)} \in \{x_{k\,\|\,k-1}\}\ s.t\ \Pr(x_k^{(i)} = x) = \tilde{q}_k^{(j)},\ \text{\textit{for}}\ i = 1, \ldots, N$ |
| 8 | Reset weights: | $q_k^{(i)} = \frac{1}{N},\ \text{\textit{for}}\ i = 1, \ldots, N$ |
| 10 | Return: | $goto\ \ step\ 1$ |

# CHAPTER III

# SYSTEM DEVELOPMENT

The state estimation techniques of the previous chapter provide a foundation from which a vehicle tracker can be developed. The tracker leverages the unrestrictive particle filter framework and provides a mechanism for both the detection of new vehicles and refinement of tracked vehicles. The proceeding section covers the topics of models used, initial sampling and resampling, detection of new vehicles, and release of tracked vehicles. Next, methods to handle the radar sensor data are investigated, this includes pruning and gating for mitigating the effects of clutter. At this point a full particle filter is developed. Considering the objectives of this thesis, an algorithm is developed to allow for the simultaneous tracking of multiple vehicles. Finally, in section 3.4, modifications are proposed to allow the entire system to operate on a moving radar-vehicle. This section includes estimating the radar-vehicle's state via an Extended Kalman Filter as well as needed transformations between the inertial and moving frames of reference.

## *3.1 Vehicle Tracking*

To solve the vehicle tracking problem a particle filter is implemented. As seen in the previous chapter, the particle filter provides a much less restrictive framework with which to develop the system. Most importantly it allows for the use of a nonlinear vehicle model, which is discussed in section 3.1.1, and a nonlinear observation model, discussed in section 3.1.3. Additionally, the noises used may be non-Gaussian. This is especially important for the observation model.

### 3.1.1 Vehicle Model

The choice of a suitable model is of great importance when working on an estimation problem and is typically a tradeoff between simplicity and accuracy.

For the tracking problem presented here a nonlinear, kinematic vehicle model with state elements $s = [x, y, \theta, V, \omega]^\intercal$ is chosen such that,

$$s_{k+1} = f_k(s_k, u_k, T) + w_k \tag{43}$$

$$f_k(s_k, T) = \begin{bmatrix} x_k + (\frac{2V_k}{\omega_k})\sin(\frac{\omega_k T}{2})\cos(\theta_k + \frac{\omega_k T}{2}) \\ y_k + (\frac{2V_k}{\omega_k})\sin(\frac{\omega_k T}{2})\sin(\theta_k + \frac{\omega_k T}{2}) \\ \theta_k + \omega_k T \\ V_k \\ \omega_k \end{bmatrix}. \tag{44}$$

In the model above, $x$, $y$, and $\theta$ represent the location of a characteristic point of the vehicle and the orientation, which can be considered the direction perpendicular to a fixed axle, respectively. This is a common model for wheeled robots, with $V_k$ and $\omega_k$ typically representing the inputs to the system. Here, however, they are included as state elements, allowing them to be estimated.

### 3.1.2 Initial Sampling

As stated in the particle filter algorithm, found in table 4, it is assumed that the initial posterior, $p(s_0)$, is known and can be sampled from. At $k = 0$ it can be assumed that very little is known about the vehicle state and it would be beneficial to represent this fact in $p(s_0)$. Under the Kalman Filter framework this would be very difficult to do given the assumption of Gaussian noise about an initial estimate. Within the particle filter framework an arbitrary density can be assumed, which, for example, can be a mixture of Gaussian and uniform densities.

To begin, it is intuitive to state that the posterior is only nonzero within a subset of the state-space, the area of concern, and that is the area from which the initial samples should be drawn. For the $x$ and $y$ dimensions, this subset is the field of view (FOV) of the radar sensor. The radar FOV can be defined as

$$FOV = \left[ (x,y) : \substack{R_{min} \leq \sqrt{(x^2+y^2)} \leq R_{max} \\ \Psi_{min} \leq \arctan(\frac{y}{x}) \leq \Psi_{max}} \right] \forall x > 0,\ y \in \mathbb{R} \tag{45}$$

where $[R_{min}, R_{max}]$ is the minimum and maximum range measurement and $[\Psi_{min}, \Psi_{max}]$ is the minimum and maximum azimuth measurement. Similarily, the heading, $\theta$, and velocity, $V$, can be constrained such that,

$$-\pi \ \leq \theta \leq \ \pi \tag{46}$$

$$V_{min} \ \leq V \leq V_{max} \tag{47}$$

where $V_{min}$ and $V_{max}$ are reasonably chosen limits.

To represent the uncertainity in $p(s_0)$ it is beneficial to sample this space uniformly. To do so, a Halton sequence, which is a deterministic, low-discrepancy sequence[35], is transformed to fill the spaces defined above.

While the turning rate, $\omega$, is also unknown, it is reasonable to assume that for most tracked vehicles this value is zero. Therefore, for the intialization of the particle filter, the $\omega$ values should be sampled from a zero-mean, Normal distribution with a specified variance.

The result of this initial sampling procedure can be seen in figure 1. Even though not all of the state elements were sampled from uniform distributions, every particle is given an initial importance weight of $\frac{1}{N}$.

### 3.1.3 Observation Model

The observation model is the mechanism to relate the radar measurements to the current set of particles. As stated in the Introduction, the radar sensor produces measurements that include range($r$) and azimuth angle($\psi$). The nonlinear function

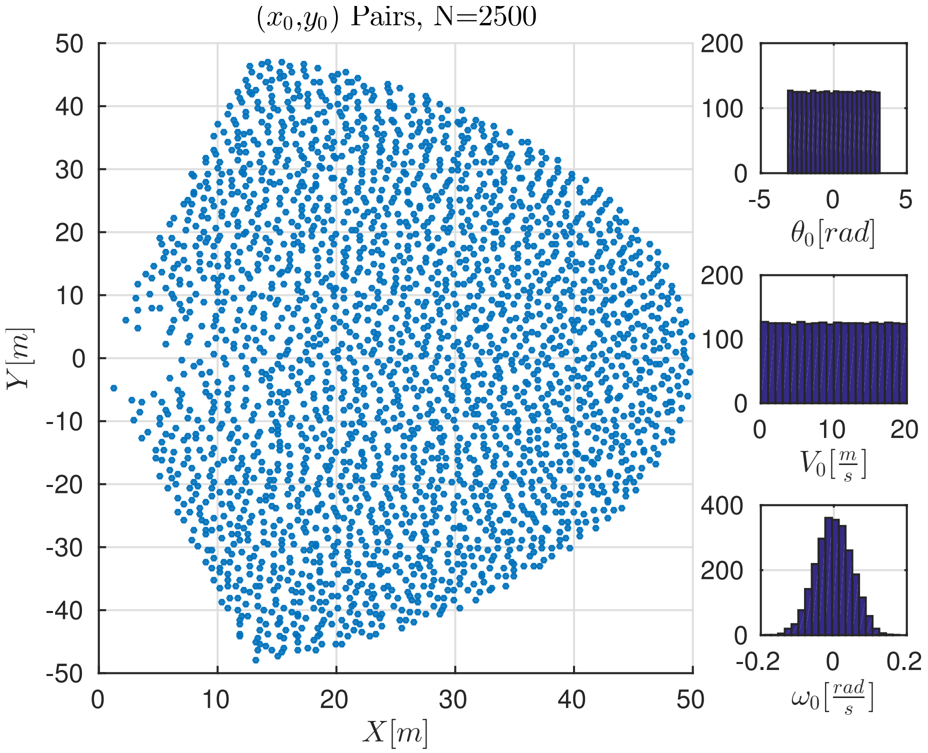


Figure 1: Initial particles sampled from $p(s_0)$.

$h(s)$ maps the state-space to the measurement-space by

$$\begin{bmatrix} r \\ \psi \end{bmatrix} = h(s) = \begin{bmatrix} \sqrt{x^2+y^2} \\ \arctan(\frac{y}{x}) \end{bmatrix}. \tag{48}$$

Recalling from section 2.4.2, the likelihood function $p(y_k \mid s_{k\,|\,k-1})$ is needed to update the importance weights, $\tilde{q}_k^{(i)}$, based on

$$\tilde{q}_k^{(i)} \propto \tilde{q}_{k-1}^{(i)} p(y_k \mid s_{k\,|\,k-1}^{(i)}). \tag{49}$$

where $y_k$ is the new measurement and $s_{k\,|\,k-1}^{(i)}$ is particle state vector predicted from the last measurement update.

When considering additive measurement noise, as is done in the defined models, the likelihood function is

$$p(y_k \mid s_{k\,|\,k-1}^{(i)}) = p_{v_t}(y_k - h(s_k^{(i)})), \tag{50}$$

where $p_{v_k}(\cdot)$ is the measurement noise density[40]. Within the particle filter framework this density can be arbitrary.

Since the noise density can be arbitrary, it can be easily extended to include multiple measurements; for example, if a vehicle has more than one data point associated with it. Assuming that the additive noise of each data point is Gaussian, the noise density can be represented by a mixed Gaussian distribution,

$$
\begin{aligned}
X^{(m)} &= y_k^{(m)} - h(s_k^{(i)}) \\
p(y_k \,|\, s_{k\,|\,k-1}^{(i)}) &\propto \sum_{m=1}^{M} \exp(-\frac{1}{2} X^{(m)\intercal}\, \Sigma^{-1}\, X^{(m)})
\end{aligned}
\tag{51}
$$

where $M$ is the number of measurements being mixed, $y_k^{(m)}$ is the individual measurement, $\Sigma$ is the measurement covariance matrix, and the individual Gaussians have been given equal weights. Since the importance weights are normalized after every update the standard Gaussian constant has been dropped.

### 3.1.4 Resampling

As mentioned previously in section 2.4.2, resampling is the step that makes the particle filter practical. Without this step, all of the importance weight will be held by a few–or single–particles after a handful of iterations. This produces a poor estimate of the posterior and is a waste of computational resources. The task of the resampling step is to choose the $N$ new particles, with replacement, from the set of current particles according to their normalized importance weight.

There exist a number of methods that have been proposed to solve the resampling problem. These include Multinomial Resampling, Residual Resampling, Stratified Resampling, and Systematic Resampling; all of which have been investigated by Douc et al.[12]. Additionally, the computational complexity and performance of the same algorithms have been compared by Hol et al.[21]. Accordingly, systematic resampling has been chosen due to its simplicity and performance.

Systematic resampling works by choosing the particle that corresponds to the particle set's cumulative distribution function (CDF) at systematically determined points. The CDF is the cumulative sum of the particles,

$$F(n) = \sum_{i=1}^{n} \tilde{q}_k^{(i)} \tag{52}$$

where $F(0) = 0$ and $F(N) = 1$. The selection points are determined such that,

$$U^{(i)} = \frac{(i-1)}{N} + U_0 \ \ \mathit{for}\ i = 1, \ldots, N \tag{53}$$

where $U_0$ is a single random number drawn from a uniform distribution with support $(0, \frac{1}{N}]$. Thus, the particle chosen at each selection point, $s_k^{(n)}$, is the one that satifies,

$$F(n-1) \leq U^{(i)} < F(n). \tag{54}$$

From equation (54) it can be seen that the greater the particle's importance weight the greater chance it has to be chosen once or multiple times.

It is important to realize that while the ordering of the particles defines the shape of the CDF, it has no effect on the resampling algorithm. Additionally, if the particles have uniform importance weights, every particle will be selected exactly once, regardless of $U_0$. From a logical standpoint, this is ideal since the discrepenancy between the particles' weights is already at a minimum.

The algorithm can be visualized in Figure 2, where the blue curve represents the CDF of 10 particles and the red circles represent the selection points. This example includes 10 particles with normalized weights chosen uniformly.

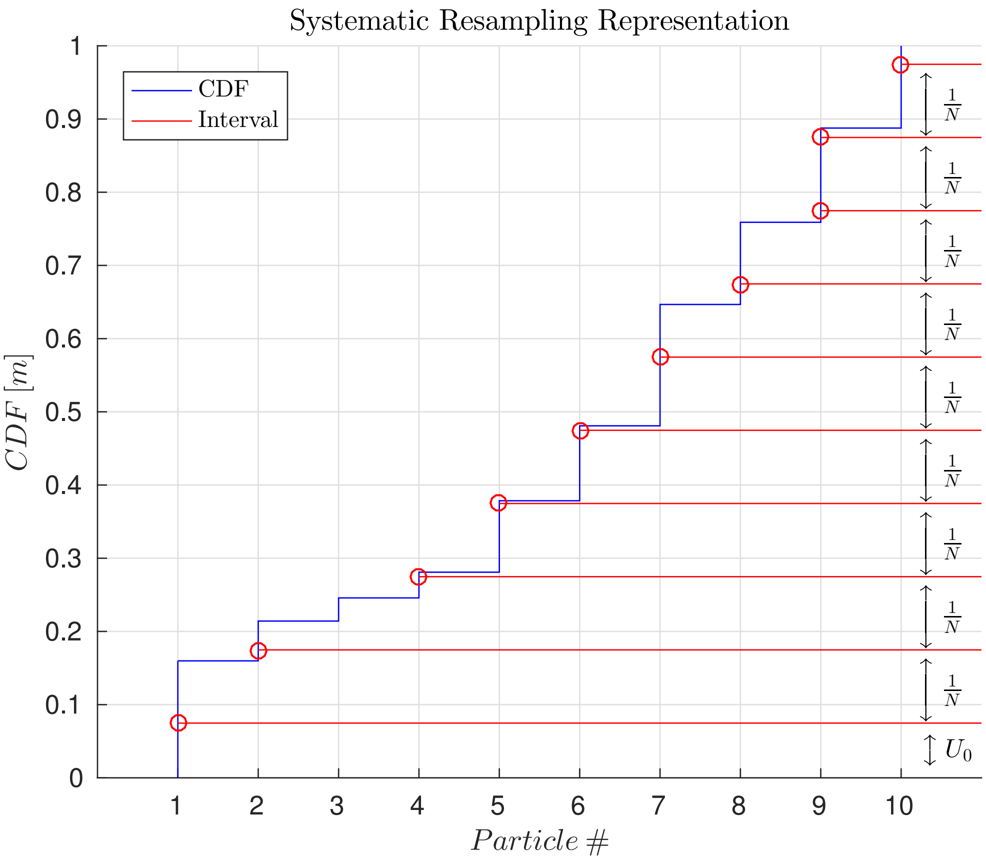


Figure 2: Systematic resampling.

### 3.1.5 Detection

A major goal of the vehicle tracking system is the ability to determine when the particle filter has detected a vehicle. This decision is different than the filter converging. Considering the information available to the system, $N$ particles with state vectors, $s = [x, y, \theta, V, \omega]$, and normalized importance weights, the particle filter is considered to have detected a vehicle when the variance of the particle headings is less than a set threshold.

It is important to note that the heading, $\theta$, is a circular quantity; that is, it is constrained to $(-\pi, \pi]$. Therefore the normal calculation of variance, $\text{Var}(x) = E[(x - \mu)^2]$, will not work. Rather, the following expression is used, which is a slight modification of the descriptive statistics defined in [22],

$$
\begin{aligned}
C_w &= \sum_{i=1}^{N} \tilde{q}_k^{(i)} \cos \theta_k^{(i)} \\
S_w &= \sum_{i=1}^{N} \tilde{q}_k^{(i)} \sin \theta_k^{(i)} \\
\mathrm{Var}(\theta) &= 1 - \sqrt{C_w^2 + S_w^2} \qquad (55)
\end{aligned}
$$

where $\tilde{q}_k$ are the normalized importance weights and $C_w$ and $S_w$ are weighted averages of the sine and cosine of the particles' headings, respectively. Under this formulation:

$$0 \leq \mathrm{Var}(\theta) \leq 1 \qquad (56)$$

Thus, this is only a relative measurement where $\mathrm{Var}(\theta) = 1$ describes high variance and $\mathrm{Var}(\theta) = 0$ occurs when all particles have the same heading.

With the metric defined by equation 55 the detection of a vehicle can be defined as,

$$\mathrm{Var}(\theta) \leq \sigma_{th} \qquad (57)$$

where $\sigma_{th}$ is the threshold value. As the heading variance falls below the threshold it is said that the particle filter has moved from the detection stage to the refinement stage.

### 3.1.6 Track Deletion/Detector Reset

As the system is targeted to run recursively, a mechanism is needed to systematically reset the tracker when deemed necessary. This is to ensure that the system runs both effectively and efficiently.

During the detection stage this amounts to ensuring the area of concern, the radars field of view, is fully populated by particles. When the particles have dispersed outside of the FOV, such that the covariance of the $xy$-dimensions is above a set threshold, the particle set is reinitialized to the set established during the initial sampling, section 3.1.2, including the uniform particle weights.

Similarly, during the refinement stage it is possible that a tracked vehicle has left the radars field of view or is no longer recognized, possibly because it is blocked by another object or vehicle. By recalling equation (45), the binary assessment of whether a tracked vehicle is out-of-bounds (outside the FOV), $OOB$, can be written as

$$OOB^{(j)} = \begin{cases} 0 & \left(\hat{x}^{(j)}, \hat{y}^{(j)}\right) \in FOV \\ 1 & \left(\hat{x}^{(j)}, \hat{y}^{(j)}\right) \notin FOV \end{cases}, \quad \mathit{for}\ j = 1, \ldots, J, \tag{58}$$

where J is the number of tracked vehicles at a given instance. If true, the vehicle is released and no longer tracked.

Additionally, a vehicle is considered lostno longer recognizedwhen the variance of its particles in the $xy$-dimensions is above a set threshold. The rise in the disparity among the particles occurs when the set is not continually updated by the radar data. Therefore, the threshold allows for vehicles with more refined estimates to survive during occlusion.

## *3.2 Radar Data*

As stated in the Introduction, the radar sensor provides information on numerous objects each update cycle. The majority of these data points do not originate from a vehicle and can be considered clutter, even if they do originate from a real object. The exact definition of clutter varies between the detection and refinement stages of the particle filter and are explained below.

### 3.2.1 Pruning for Detection

During the detection stage of the particle filter clutter can be considered any data points that fall below a relative velocity magnitude threshold. Since it is assumed that the radar-vehicle is stationary, this has the effect of removing all stationary objects from the field of view. A binary determination for each data point can be made using

$$stationary^{(m)} = \begin{cases} 0 & \left|v_{rel}^{(m)}\right| > v_{rel,th} \\ 1 & \left|v_{rel}^{(m)}\right| \leq v_{rel,th} \end{cases}, \quad \textit{for } m = 1, \ldots, M, \tag{59}$$

where $M$ is the total number of data points.

It is possible that this simple filter could also remove data points originating from vehicles driving in a constant radius circle around the radar-vehicle, but this case is unlikely. In the case that a vehicle does have zero relative velocity with respect to the radar-vehicle, a collision is not possible and thus the situation can be effectively ignored.

### 3.2.2 Gating for State Refinement

To determine whether a data point originated from a tracked vehicle a validation gate is used. The gate is a binary decision based on the Mahalanobis distance[5] calculated from the current state-estimate to the data point in question. To calculate such a distance, the state estimate and data point must be mapped into a space with common dimensions. The nonlinear mappings are,

$$h^{y}_{gate}(y) = \begin{bmatrix} x \\ y \\ v_{rel} \end{bmatrix} = \begin{bmatrix} R\cos\psi \\ R\sin\psi \\ v_{rel} \end{bmatrix}, \quad h^{s}_{gate}(s) = \begin{bmatrix} x \\ y \\ v_{rel} \end{bmatrix} = \begin{bmatrix} x \\ y \\ V\arctan\left(\frac{y}{x}\right) \end{bmatrix}. \tag{60}$$

The Mahalanobis distances between current state estimate and each data point, $D^{(m)}$, can then be calculated as

$$X_m = h^{y}_{gate}\left(y^{(m)}\right) - h^{s}_{gate}(\hat{s}) \tag{61}$$

$$D^{(m)} = \sqrt{X_m^{\intercal}\, \Sigma_{gate}^{-1}\, X_m}, \quad \textit{for } m = 1, \ldots, M \tag{62}$$

where $\Sigma_{gate}$ is the covariance matrix associated with the gate and $M$ is the total number of measurements. The quantity $D$ is a nondimensionalized measure of length and therefore not affected by differences in scale between the dimensions.

Since the covariance matrix is assumed to be both positive definite and symmetric, it can be decomposed into its eigenvalues and eigenvectors:

$$\Sigma_{gate} = S\Lambda S^{\intercal} \tag{63}$$

$$S(s) = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{64}$$

$$\Lambda = \text{diag}(\sigma_l,\ \sigma_w,\ \sigma_v) \tag{65}$$

where the columns of $S(s)$ are the eigenvectors and span the common space. In this case, the first eigenvector is aligned with the vehicle heading in the $xy$-plane, the second eigenvector is perpedicular to the first, and the third spans the $v_{rel}$ dimension.

This allows for the eigenvalues to be representative length scales in each dimension. Considering that the vehicle length is greater than its width, it would make sense that the gate is longer in the heading direction. Thus, it is chosen that $\sigma_l > \sigma_w$.

The binary determination of whether a data point originated from a tracked vehicle can be made using

$$vehicle^{(m)} = \begin{cases} 0 & D^{(m)} > \gamma \\ 1 & D^{(m)} \leq \gamma \end{cases} \tag{66}$$

where $\gamma$ is determined from the inverse $\chi^2$ cumulative distribution of $k$ degrees of freedom at a confidence interval, $\alpha$, which is normally 0.05 or 0.01[10]. In equation 66, there are $k = 3$ degrees of freedom.

## 3.3 Multi-Vehicle Tracking

As stated in the Introduction, an objective of this thesis is to develop a mechanism within the system to allow for the tracking of multiple vehicles. In a rigorous sense, this is a very complicated and computationally intensive task which is not well-suited for the target application. In the following section the algorithm to track multiple vehicles is proposed, which, under certain assumptions and considerations, substantially simplifies the data association step. The proceeding section, 3.3.2, explores the assumptions further.

### 3.3.1 Detached Operation

The particle filter developed above is tasked with tracking a single vehicle through its lifecyle: detection, continued refinement, and release. Since the particle filter is completely self contained, tracking multiple vehicles amounts to using a collection of individual particle filters.

At all times there is one particle filter acting as a detector. It is tasked with detecting new vehicles entering the field of view of the radar. Once a vehicle is detected, a procedure described in section 3.1.5, the particle filter is detached to run independently. While running indepentdently, the particle filter continues to refine its tracked vehicle's state using the subset of the radar-sensor data originating from the specific vehicle. When the tracked vehicle exits the field of view, or the signal is lost, the particle filter will remove itself from the collection. At the time of detection/detachment a new particle is initialized and set to run as a detector. This will happen for every new vehicle entering the radars field of view. An overview of the algorithm is provided in Table 5.

Table 5: Recursive steps for the multiple-vehicle tracking algorithm.

| **Multi-Vehicle Tracking Algorithm** | | |
|---|---|---|
| 0 | Initialize: | Detection PF: $\mathrm{PF}^{\mathrm{det}}$ |
| | | Tracking PF: $\mathrm{PF}^{\mathrm{trk}} = \left[\mathrm{PF}^1, \ldots, \mathrm{PF}^N\right], N = 0$ |
| 1 | Radar measurement: | $Z_k$ |
| | **Update tracking PFs** | |
| 2 | For each PF: | $\mathrm{PF}^i$, $i = 1, ..., N$ |
| 3 | | → perform measurement update. |
| 4 | | → *if*: out-of-bounds, loss of recognition, *then*: |
| 5 | | release $\mathrm{PF}^i$ from $\mathrm{PF}^{\mathrm{trk}}$, $N = N - 1$. |
| | **Update detector PF** | |
| 6 | Update $Z_k$: | remove data points used by $\mathrm{PF}^{\mathrm{trk}}$. |
| 7 | Measurement update: | perform measurement update with $Z_k$ |
| 8 | Check detection: | *if*: vehicle is detected, *then*: |
| 9 | | add $\mathrm{PF}^{\mathrm{det}}$ to $\mathrm{PF}^{\mathrm{trk}}$, $N = N + 1$. |
| 10 | | initialize a new detector, $\mathrm{PF}^{\mathrm{det}}$. |
| 11 | Return: | goto step 1 |

It is important that the detector does not include data points related to currently-tracked vehicles in its measurement update step. Including such points would result in the same vehicle potentially being tracked by multiple particle filters; a waste of resources and provides an incorrect representation of the environment. After it has been determined which data points belong to which tracked-vehicle, the data association process discussed in the next section, the remaining data points are passed to the detector.

### 3.3.2 Data Association

The complexity of the multi-target tracking problem stems from the need to associate sensor data points with their origin and, potentially, originless points as clutter. There exist a number of well researched methods to perform such a task, a table comparing the algorithms can be found in [11]. The more popular and widely-used algorithms include: Global Nearest Neighbor Standard Filter (GNNSF), Joint Probabalistic Data Association Filter (JPDAF), and Multiple Hypothesis Tracker (MHT), among others. These methods are not investigated in this thesis, but interested readers can refer to [38], [4], and [39] for an in-depth analysis of the algorithms.

Data association algorithms work–in a high-level sense–to minimize, or remove, the influence of data points not originating from the tracked object from being included in the update procedure and, therefore, influencing the tracked objects state estimate. This is a complex and computationally intensive task that grows as more data points and tracked objects are considered.

The methods are typically implemented in situations where it is particularly difficult to not only separate relevant data points from clutter and false alarms but also assign the relevant data points to their correct object tracks. This is common problem when tracking commercial aircraft[3]. The sensor used in this thesis, a short-range UWB radar, measures relative velocity in addition to position. This provides another dimension along which the relevant vehicles may be separated from not only clutter but also each other. As seen in Figure 3, data from moving vehicles are well separated from clutter.

Data association is also especially important in situations where measurements are infrequent such that updating with an incorrect measurement could lead to large errors over time. Alternatively, since the radar sensor used in this thesis provides updates every 66 milliseconds, on the rare occasion that an incorrect data point is used in the update procedure of a tracked vehicle the influence would be minimal.

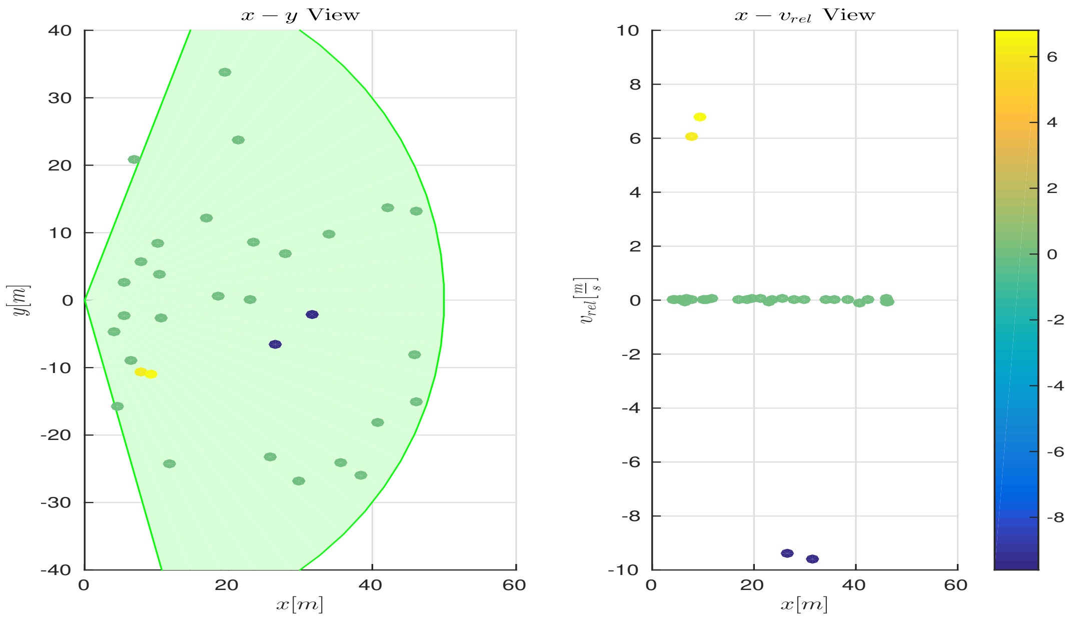


Figure 3: Left) the radar data points displayed in the $x - y$ plane with relative velocity identified by color. Right) the radar data displayed in the $x - v_{rel}$ plane to show separation from clutter. The set of yellow points and set of blue points each originated from different vehicles.

With these considerations and assumptions, no rigorous data association algorithm is explicitly included in the system developed within this thesis. The data association step is handled, instead, by the gating procedure. Essentially, if a data point is within the validation gate, which is dependent on both position and velocity, it is assumed to have originated from the tracked vehicle and therefore included in the measurement update. The assumptions fit within the scope of this thesis and justified by experimental data.

## *3.4 Extension to Moving Vehicle*

To extend the system developed above it is important that the state of the radar-vehicle is well known such that the set of particles, which exist in a global, inertial frame of reference, can be transformed into the radars frame of reference for use in the measurement update and gating procedures. This, once again, poses the problem of estimating the radar-vehicle state via a set of limited measurements; this is accomplished through the use of an Extended Kalman Filter.

### 3.4.1 Extended Kalman Filter

Unlike with the tracking system, the restrictions imposed by the EKF framework do not present issues when estimating the radar-vehicle state. It is reasonable to assume that an initial estimate of the vehicle state is available and the process and measurement noises are Gaussian. Given that the EKF is computationally less intensive and easier to implement than a particle filter, it is well suited for this specific estimation problem.

The state elements of interest include position and orientation terms, $x$, $y$, and $\theta$, and velocity terms, $V$ and $\omega$. Available measurements include GPS location, $(x, y)$, individual wheel angular velocity, $\Omega^{(1,2,3,4)}$, and steering angle, $\alpha$. Assuming a no-slip condition, which is safe at the low speeds of the target application [24], virtual measurements for vehicle velocity, $V_{virt}$, and turn-rate, $\omega_{virt}$, can be calculated as

$$V^{virt} = \frac{\Omega^{(3)} R_3 + \Omega^{(4)} R_4}{2} \tag{67}$$

$$\omega^{virt} = \frac{\Omega^{(4)} R_4 - \Omega^{(3)} R_3}{W} \tag{68}$$

where $R_{3,4}$ are the radii of the left and right rear wheels, respectively, and $W$ is the vehicle width. This approach is similar to that of Özkan et al.[37], who use a marginalized particle filter to perform the estimation under looser assumptions, and Carlson et al.[8], who include a gyroscope in the available sensors.

The accuracy of the two measurements is dependent on the knowledge of the true radii of the two rear wheels. Assuming a nominal measurement, $R_{nom}$, the true radii can be expressed as

$$R_3 = R_{nom} + \delta^{(3)} \tag{69}$$

$$R_4 = R_{nom} + \delta^{(4)}, \tag{70}$$

which can be substituted into equations (67) and (68) to produce

$$V = V^{virt} - \frac{\Omega^{(3)}\delta^{(3)}}{2} - \frac{\Omega^{(4)}\delta^{(4)}}{2} \tag{71}$$

$$\omega = \omega^{virt} + \frac{\Omega^{(3)}\delta^{(3)}}{W} - \frac{\Omega^{(4)}\delta^{(4)}}{W}. \tag{72}$$

The wheel radius errors can be estimated alongside the vehicle pose elements by augmenting the state vector such that $s = [x, y, \theta, \delta^{(3)}, \delta^{(4)}]$. Assuming a no-slip condition between the tires and road surface, a nonlinear, kinematic system model can be used:

$$\begin{bmatrix} x_{k+1} \\ y_{k+1} \\ \theta_{k+1} \\ \omega_{k+1} \\ \delta^{(3)}_{k+1} \\ \delta^{(4)}_{k+1} \end{bmatrix} = f_k(s_k, u_k, T) = \begin{bmatrix} x_k + \left(V_k^{virt} - \frac{\Omega_k^3\delta_k^3}{2} - \frac{\Omega_k^4\delta_k^4}{2}\right)\cos(\theta_k)T \\ y_k + \left(V_k^{virt} - \frac{\Omega_k^3\delta_k^3}{2} - \frac{\Omega_k^4\delta_k^4}{2}\right)\sin(\theta_k)T \\ \theta_k + \omega_k T \\ \left(V_k^{virt} - \frac{\Omega_k^3\delta_k^3}{2} - \frac{\Omega_k^4\delta_k^4}{2}\right)\frac{\tan(\alpha_k)}{L} \\ \delta_k^{(3)} \\ \delta_k^{(4)} \end{bmatrix}, \tag{73}$$

where the input vector at time $k$, $u_k = [V_k^{virt}, \Omega_k^3, \Omega_k^4, \alpha_k]$, includes the virtual velocity measurement, rear-left wheel angular velocity, rear-right wheel angular velocity, and steering angle, respectively. The Jacobian of the nonlinear function $f_k$ is

$$\frac{\partial F_k}{\partial s} = \begin{bmatrix} 1 & 0 & -\left(V_k - \frac{\Omega_k^3\delta_k^3}{2} - \frac{\Omega_k^4\delta_k^4}{2}\right)\sin(\theta_k)T & 0 & -\frac{\Omega_k^3}{2}\cos(\theta)T & -\frac{\Omega_k^4}{2}\cos(\theta)T \\ 0 & 1 & \left(V_k - \frac{\Omega_k^3\delta_k^3}{2} - \frac{\Omega_k^4\delta_k^4}{2}\right)\cos(\theta_k)T & 0 & -\frac{\Omega_k^3}{2}\sin(\theta)T & -\frac{\Omega_k^4}{2}\sin(\theta)T \\ 0 & 0 & 1 & T & 0 & 0 \\ 0 & 0 & 0 & 0 & -\frac{\Omega^3\tan(\alpha)}{2L}T & -\frac{\Omega^4\tan(\alpha)}{2L}T \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}.$$

To update the estimate the measurement vector $z_k = [x, y, \omega]^\intercal$ is used, where $x$ and $y$ are provided by the GPS and $\omega$ is the virtual measurement. Since the GPS data and onboard vehicle data are updated at different rates, two measurement models are needed; one for when GPS data is available, $h_k^{GPS}$, and one for when it is not, $h_k^{NOGPS}$. The two models are:

$$h_k^{GPS} = \begin{bmatrix} x_k \\ y_k \\ \omega \end{bmatrix}, \quad h_k^{NOGPS} = \begin{bmatrix} 0 \\ 0 \\ \omega \end{bmatrix},$$

with corresponding Jacobians,

$$\frac{\partial h^{(GPS)}}{\partial s} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}$$

$$\frac{\partial h^{(NOGPS)}}{\partial s} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}.$$

Given the quantities above the EKF will provide an estimate of the vehicle state, $\hat{s}$, from which the system output vector can be calculated:

$$\hat{y}_{out} = \begin{bmatrix} \hat{x}_k \\ \hat{y}_k \\ \hat{\theta}_k \\ \hat{\omega}_k \\ \hat{V}_k \end{bmatrix} = \begin{bmatrix} \hat{x}_k \\ \hat{y}_k \\ \hat{\theta}_k \\ \hat{\omega} \\ V^{virt} - \frac{\Omega^{(3)}\hat{\delta}^{(3)}}{2} - \frac{\Omega^{(4)}\hat{\delta}^{(4)}}{2} \end{bmatrix}. \qquad (74)$$

### 3.4.2 Coordinate Space Transformations

The tracker developed earlier in this chapter assumed that the radar-vehicle was stationary and thus the two reference frames, seen in Figure 4, could be arbitrarily aligned. With an estimate of the full vehicle-state provided by the EKF, the current set of particles can be transformed from the global frame of reference to the radar frame of reference, allowing the same update procedure to be used, via

$$x_r = \hat{x} + l_r \cos\hat{\theta} \tag{75}$$

$$y_r = \hat{y} + l_r \sin\hat{\theta} \tag{76}$$

$$\begin{bmatrix} R^{(i)} \\ \psi^{(i)} \end{bmatrix} = \begin{bmatrix} \sqrt{\left(x^{(i)} - x_r\right)^2 + \left(y^{(i)} - y_r\right)^2} \\ \arctan\left(\frac{y^{(i)} - y_r}{x^{(i)} - x_r}\right) - \hat{\theta} \end{bmatrix} \tag{77}$$

where $l_r$ is the known length from the rear axle of the radar-vehicle to the radar sensor, $(x_r, y_r)$ is the location of the radar sensor in the inertial frame, and $\hat{x}$, $\hat{y}$, and $\hat{\theta}$ are from the EKF output.

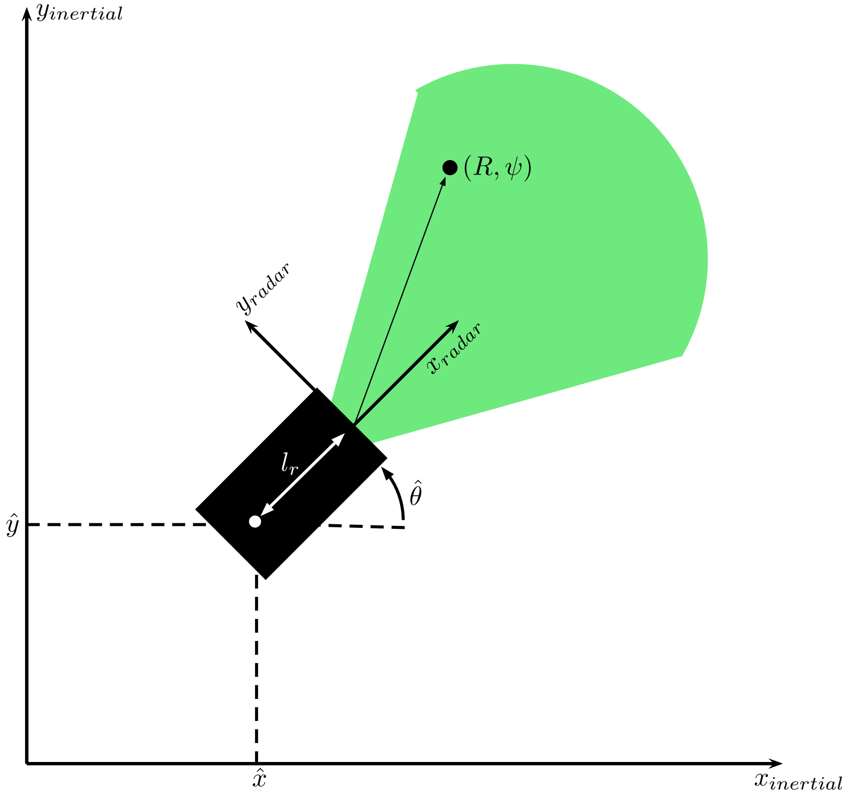


Figure 4: Illustration of the two frames of reference. The radar-vehicle frame of reference extends out from the radar-sensor which is located $l_r$ in front of the rear axle. It is assumed the normal vector of the face of the sensor is aligned with the vehicle heading at all times.

Additionally, the nonlinear mappings for use in the gating step can be extended to a moving radar-vehicle. Using the EKF output the velocity of the radar sensor can be computed as

$$V_r = \sqrt{\hat{V}^2 + (\hat{\omega} l_r)^2} \tag{78}$$

$$\theta_r = \arctan\left(\frac{\hat{V}\sin\hat{\theta} + \hat{\omega} l_r \cos\hat{\theta}_r}{\hat{V}\cos\hat{\theta} + \hat{\omega} l_r \sin\hat{\theta}_r}\right), \tag{79}$$

where $V_r$ and $\theta_r$ are the magnitude and direction of the radar sensor in the inertial frame, respectively.

Using equations (78) and (79) the relative velocity vector between the radar sensor and individual particle can then be written as

$$\begin{bmatrix} v_{rel,x}^{(i)} \\ v_{rel,y}^{(i)} \end{bmatrix} = \begin{bmatrix} V^{(i)}\cos\psi^{(i)} - V_r\cos\theta_r \\ V^{(i)}\sin\psi^{(i)} - V_r\sin\theta_r \end{bmatrix}. \tag{80}$$

The relative velocity in the radial direction of a particle is calculated using a dot product of the vector found in equation (80) and the directional vector from the radar sensor to the particle in question:

$$v_{rel,R} = \frac{\begin{bmatrix} v_{rel,x}^{(i)} & v_{rel,y}^{(i)} \end{bmatrix} \cdot \begin{bmatrix} x^{(i)} - x_r \\ y^{(i)} - y_r \end{bmatrix}}{\sqrt{\left(x^{(i)} - x_r\right)^2 + \left(y^{(i)} - y_r\right)^2}}. \tag{81}$$

The nonlinear mappings from the radar measurement-space and vehicle state-space into the common gate-space can now be defined as

$$h_{gate}^{y}(s,y) = \begin{bmatrix} x \\ y \\ v_{rel} \end{bmatrix} = \begin{bmatrix} R\cos\left(\psi + \hat{\theta}\right) + x_r \\ R\sin\left(\psi + \hat{\theta}\right) + y_r \\ v_{rel} \end{bmatrix} \tag{82}$$

$$h_{gate}^{s}(s,y) = \begin{bmatrix} x \\ y \\ v_{rel} \end{bmatrix} = \begin{bmatrix} \hat{x} \\ \hat{y} \\ v_{rel,R} \end{bmatrix} \tag{83}$$

where $x_r$ and $y_r$ are from equations (75) and (76), respectively and $(R, \psi, v_{rel})$ is the radar data point.

### 3.4.3 Tracker Modifications

In addition to the transformations described above, certain aspects of the original tracker must be modified to effectively work on a moving radar-vehicle. These modifications ensure that the system is efficient and effective while operating recursively.

During the detection stage it is important that the radar's field of view is sufficiently populated with particles. The area from which particles are initially sampled, the area of concern, must be expanded such that the FOV is always populated by particles. It is compuationally intractable to consider the entire environment during the initial sampling procedure. Rather, it is more feasible to consider a local area dependent on the radar-vehicle's current state. A metric can be developed to measure how populated the FOV is and, when needed, reinitialize the detector particle filter.

The pruning method to identify data points originating from stationary objects, described in scetion 3.2.1, must also be modified to be extended to a moving radar-vehicle. By first assuming that the data point in question, $\left(R^{(m)}, \psi^{(m)}, v_{rel}^{(m)}\right)$, is from a stationary object in the inertial frame, its radial relative-velocity to the radar sensor can be computed using

$$v_{rel,stat}^{(m)} = V_r \cos\left(\psi^{(m)} + th\hat{e}ta - \theta_r\right), \tag{84}$$

where $V_r$ and $\theta_r$ are from equations (78) and (79), respectively, and $\hat{\theta}$ is the estimated vehicle heading. Equation (59) can now be modified to include the influence of a moving radar-vehicle:

$$stationary^{(m)} = \begin{cases} 0 & \left|v_{rel}^{(m)} - v_{rel,stat}^{(m)}\right| > v_{rel,th} \\ 1 & \left|v_{rel}^{(m)} - v_{rel,stat}^{(m)}\right| \leq v_{rel,th} \end{cases}, \quad \mathit{for}\ m = 1, \ldots, M. \tag{85}$$

Given perfect information, the expression $\left|v_{rel}^{(m)} - v_{rel,stat}^{(m)}\right|$ should equate to zero for a stationary object.

By extending the system to a moving radar-vehicle, other vehicles are no longer the only safety concern. Previously, stationary objects were effectively ignored since a collision with them was not possible. Now, however, a collision is possible and therefore these objects must be considered as obstacles and handled accordingly.

# CHAPTER IV

# RESULTS

In this chapter the vehicle-tracking and radar-vehicle state-estimation subsystems developed previously are tested against the objectives stated in the Introduction. For this purpose, real-life (experimental) data, collected and recorded from the radar and onboard vehicle sensors specified in the Introduction, is used.

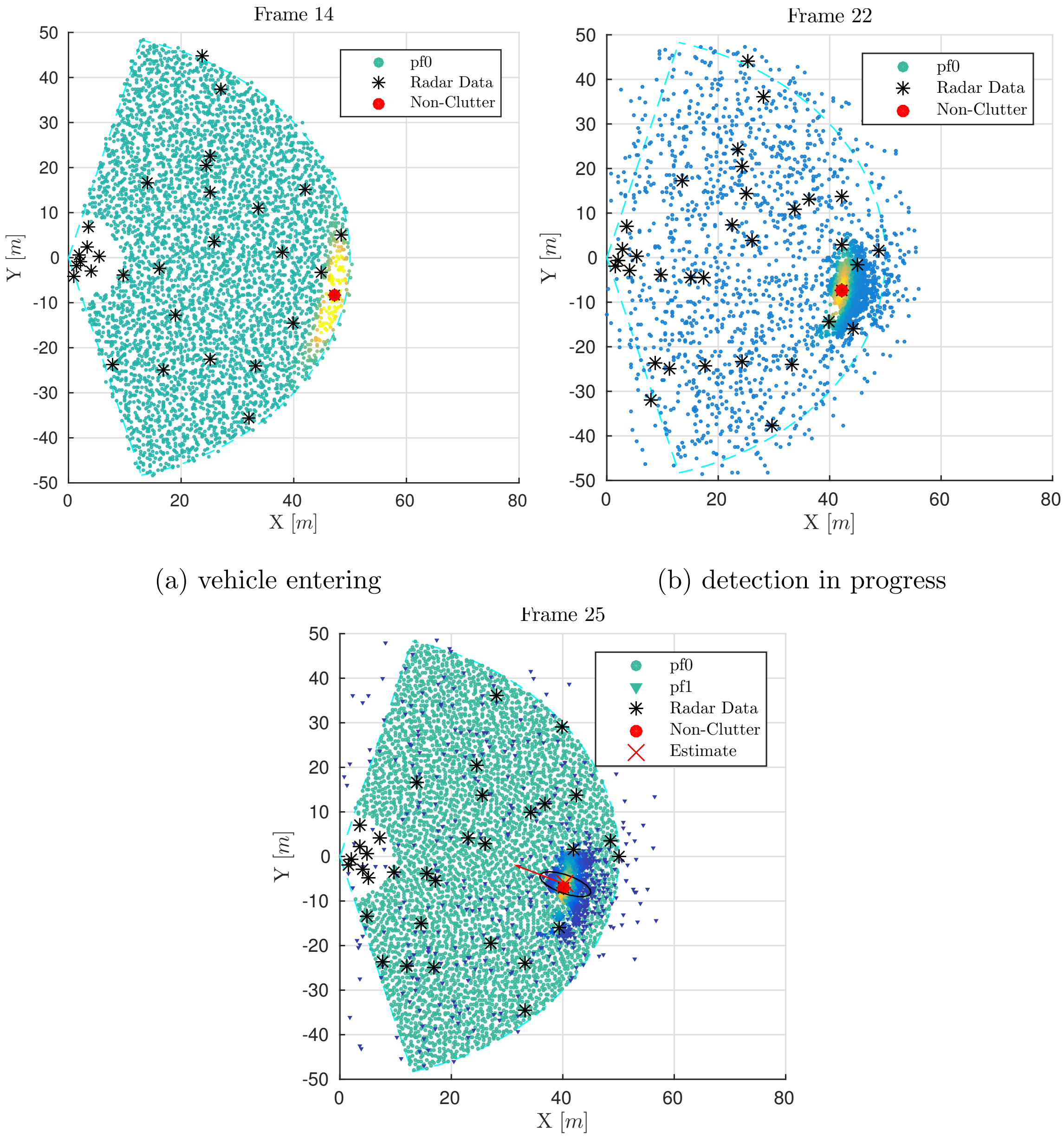
## *4.1 Vehicle Tracker*

The vehicle tracking system is tested below using experimental data collected using the ultra-wideband radar sensor presented in the Introduction. First, the system is employed to track a single vehicle moving through the radar's field of view (FOV) allowing the detection, refinement, and release stages to be viewed. Next, the capability to track multiple vehicles is tested. In both cases, the environment from which the data are collected is uncontrolled and cluttered. As stated in section 1.2, Scope of Work, the tracker is tested on a stationary radar-vehicle only.

### 4.1.1 Detection, Refinement, and Release

The detection mechanism of the system is tested using experimental data. The test case involves a vehicle entering the radar's FOV directly ahead of the sensor with a slight offset to the right. The vehicle enters and begins to turn such that it crosses the sensor's centerline. The detection procedure can be seen in Figure 5, where 5000 particles are used.

The top-left plot of Figure 5 shows the environment when the vehicle first enters the FOV. The red circle represents the radar data point that originated from the vehicle; its influence on the particle's importance weights can be seen. The black markers are the other radar data points, these points have been deemed clutter and thus ignored by the detection mechanism. The top-right plot of Figure 5 is the environment after several update cycles.



(a) vehicle entering

(b) detection in progress

(c) vehicle detected

Figure 5: Progression of detection mechanism. (a) is the first frame in which the vehicle is within the FOV. (b) the detector is converging on the vehicle's state. (c) a new vehicle has been detected.

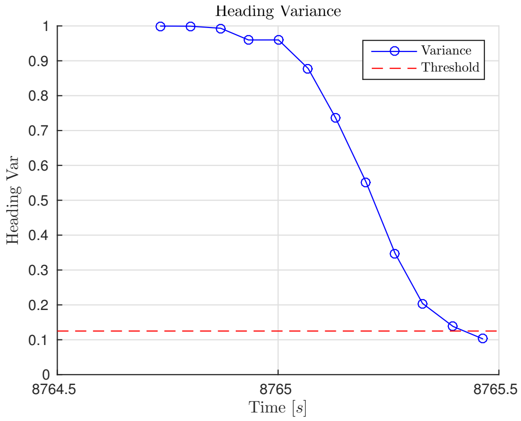


Figure 6: Heading variance during detection stage.

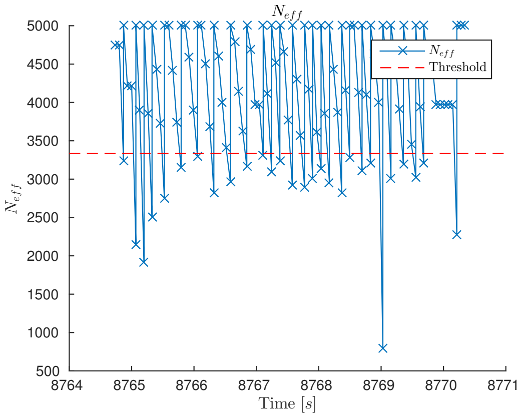


Figure 7: Effective sample size to signal resampling.

The bottom plot of Figure 5 shows the environment when it has been determined that a vehicle has been detected. The estimated location and heading of this vehicle is represented by the red 'X' and red arrow, respectively. Within the environment exist two sets of particles; one for the newly spawned vehicle tracker, *pf1*, and another for the detection mechanism, *pf0*.

The detection, and subsequent detachment of the tracker, is based on the variance in the headings of the individual particles. The time series of this value over the time spanned by the frames discussed above can be seen in Figure 6.

The same test-case can be used to demonstrate the refinement and release stages of the tracker. As shown in Figure 8, the tracker continues to update the tracked-vehicle's state estimate after detection. During this stage only data points falling within the tracker's validation gate are used in the update procedure. In Frames 55 and 68, two radar data points fall within the specific vehicle's validation gate.

During both the detection and refinement stages the particle filter is resampled according to section 3.1.4. Figure 7 shows the time series of the effective sample size, $N_{eff}$. It can be seen that by using this metric, along with a threshold, resampling is not required after every measurement update, saving computational resources.

While the *pf1* particle filter actively tracks the detected vehicle, the *pf0* particle filter continues to work as the detector. Given that the only non-clutter points originate from the tracked-vehicle, and therefore within the validation gate, the detector is passed no relevant data and stands idle for the duration of the refinement stage.

It can be seen in Frame 99 of Figure 8 that the tracked vehicle is on the verge of the FOV. After the vehicle exits the radar's FOV the tracker automatically releases itself, freeing up valuable computational resources. This occurs in Frame 100, which is not shown. This ends the specific particle filter's lifecycle.

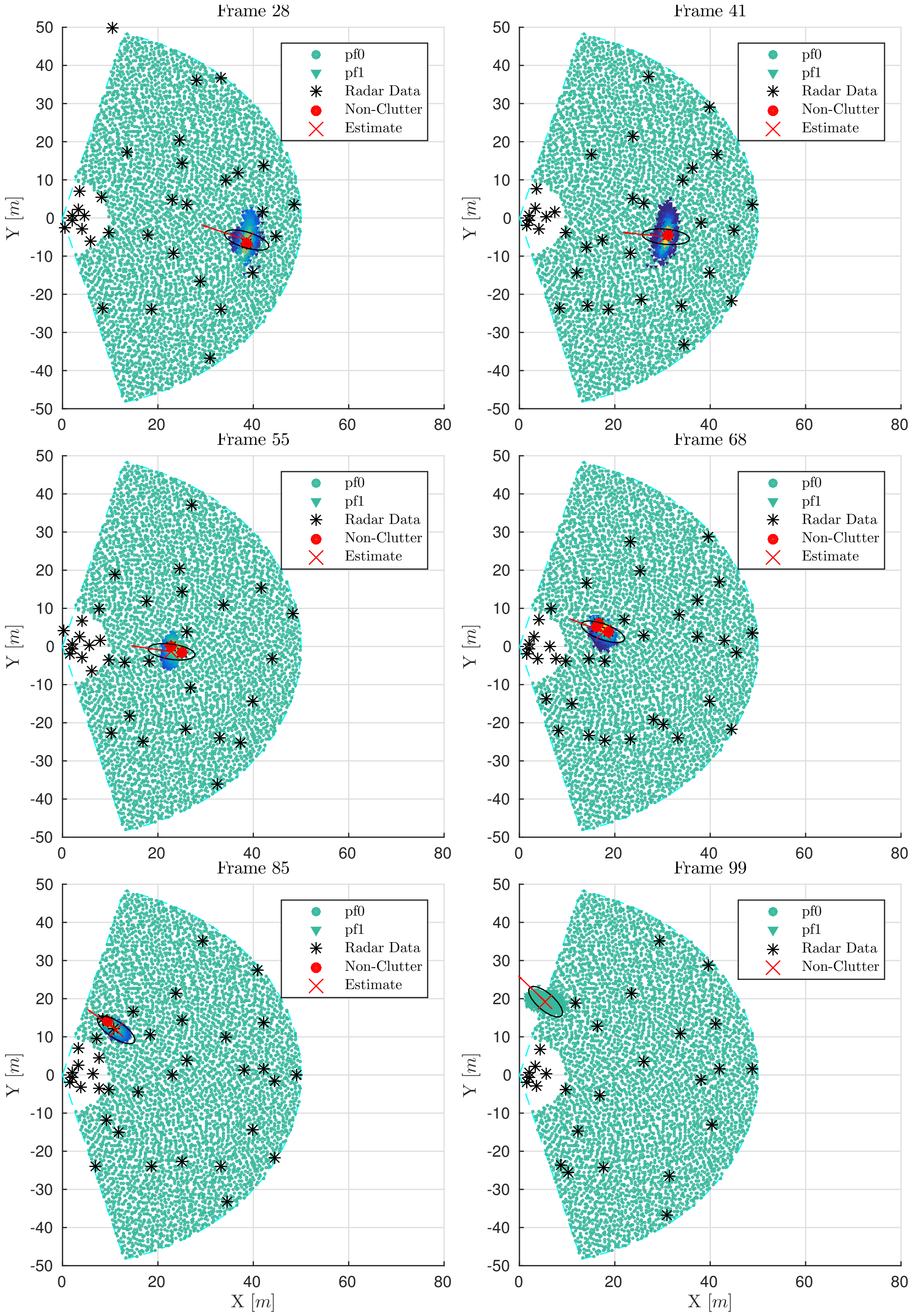


Figure 8: Progression of refining the estimated vehicle state.

### 4.1.2 Simultaneous Tracking

Using another set of experimental data, the ability to track multiple vehicles is tested. This test case involves two vehicles entering the FOV from behind and right of the radar-vehicle; the second vehicle enters roughly 2 seconds after the first. The progression of detection, refinement, and release of the vehicles can be seen in Figure 9.

Frame 21 of the sequence shows the point that the first vehicle is detected. At this point, the particle filter is detached to continue with refinement and another particle filter is brought online to act as a detector. In Frame 41 the second vehicle enters the FOV and the detector particles begin to converge, shown in Frame 50.

Just as with the first vehicle, when the second vehicle has been detected the particle filter is detached and a new detector is spawned. In Frame 58, now three individual sets of particles can be seen in the environment. The states of the tracked vehicles are then independently refined.

Frame 64 of the sequences shows the three particle filters operating alongside each other. Here the radar data points relevant to the vehicles are used by the trackers, *pf1* and *pf2*, and therefore the detector, *pf0*, stands idle once again.

The first vehicle exits the FOV in Frame 73. Frame 74 in Figure 9 shows the respective particle filter has been released without any effect on the others.

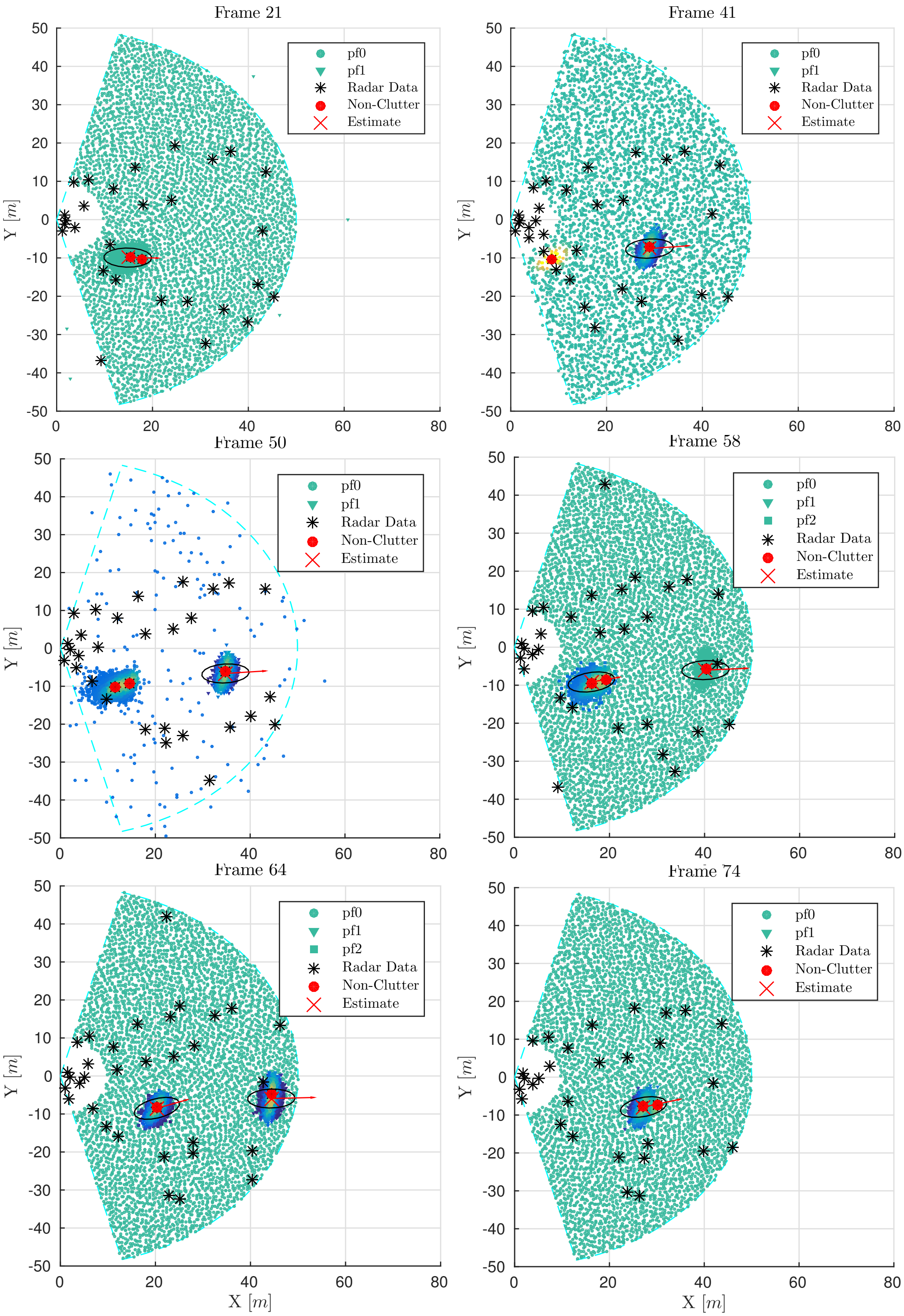


Figure 9: Progression tracking two vehicles simultaneously and independently.

## 4.2 Vehicle State Estimator

The Extended Kalman Filter, developed in section 3.4.1 to obtain an estimate of the radar-vehicle state, is tested below. The sensor data is obtained from a Mercedes Benz E-Class test vehicle. First, the vehicle parameters need by the EKF are presented, followed by the sensor. Next, the output signals produced by the EKF are analyzed.

### 4.2.1 System Inputs

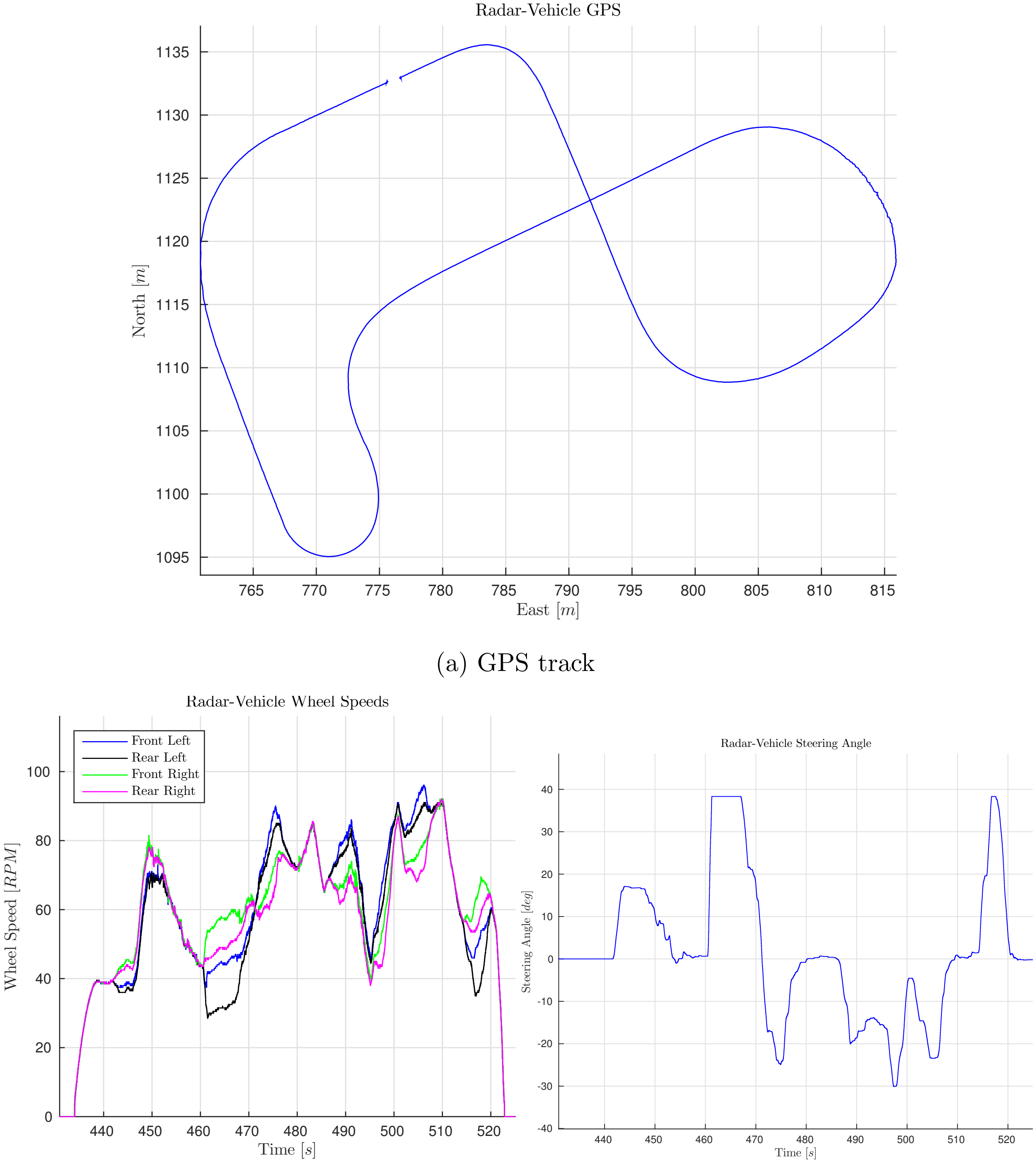
Recalling from the development of the EKF, the system needs vehicle-specific information to be used in the estimation procedure. These parameters are shown in Table 6; $L$ is the axle-to-axle length, $W$ is the axle width, and $R_{nom}$ is the nominal radius of the tires.

Table 6: Vehicle parameters used by the EKF.

| Vehicle Parameters | |
|---|---|
| $L$ | 2.939 $m$ |
| $W$ | 1.619 $m$ |
| $R_{nom}$ | 0.315 $m$ |

The EKF also requires the specification of the process and measurement noise covariance matrices, $Q$ and $R$, respectively. It also requires an initial guess of the state estimate and error covariance matrix, which amounts to an initial estimate of the posterior density. These parameters have been tuned using several test cases.

One of these example cases is shown in Figure 10. The input data provided to the system includes the GPS location, projected into a local cartesian coordinate system, individual wheel speeds, and steering angle.



(a) GPS track

(b) wheel speeds

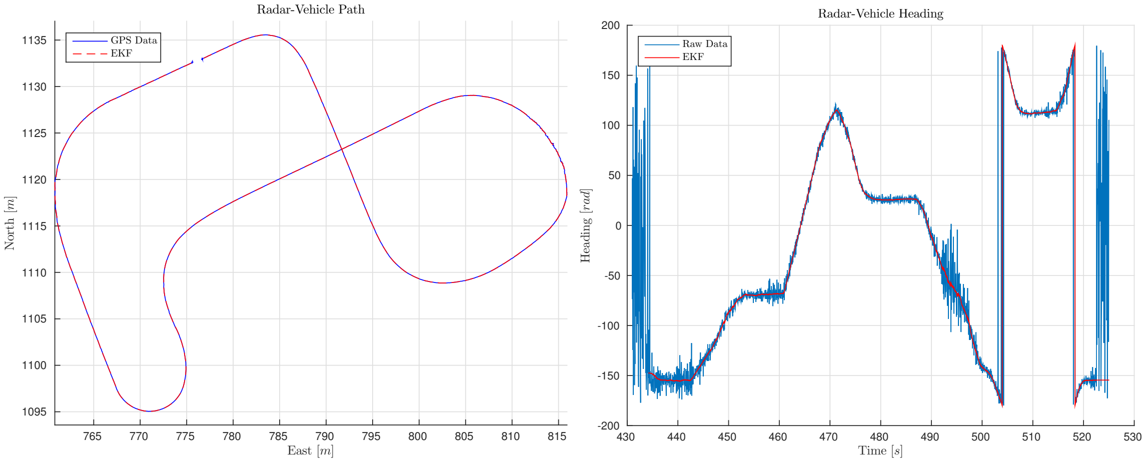
(c) steering angle

Figure 10: EKF input data. (a) The GPS track from the RTK GPS system. (b) The four wheel speeds provided by an onboard system. (c) Steering angle time series in degrees, also from the onboard system.

### 4.2.2 Results

Using the state vector estimated by the EKF, $\hat{s} = \left[x, y, \theta, \delta^{(3)}, \delta^{(4)}\right]$, where $\delta^{(3)}$ and $\delta^{(4)}$ are the wheel radius errors, the system output, defined in section 3.4.1, can be computed. The system output includes the vehicle location, $x$ and $y$, heading, $\theta$, velocity, $V$, and turn rate, $\omega$.

The position and heading output of the EKF compared with the raw data can be seen in Figure 11. Given the accuracy of the GPS system, which is realized in the measurement noise covariance matrix, the EKF track follows the GPS track closely. Also shown in Figure 11 is the raw and EKF heading time series. The raw heading is calculated as the angle formed between two sequential GPS points; this produces a noisy signal. Alternatively, the EKF produces a smoothed estimate. This smoothed signal is important when mapping the radar sensor data to the inertial frame, and vice versa.



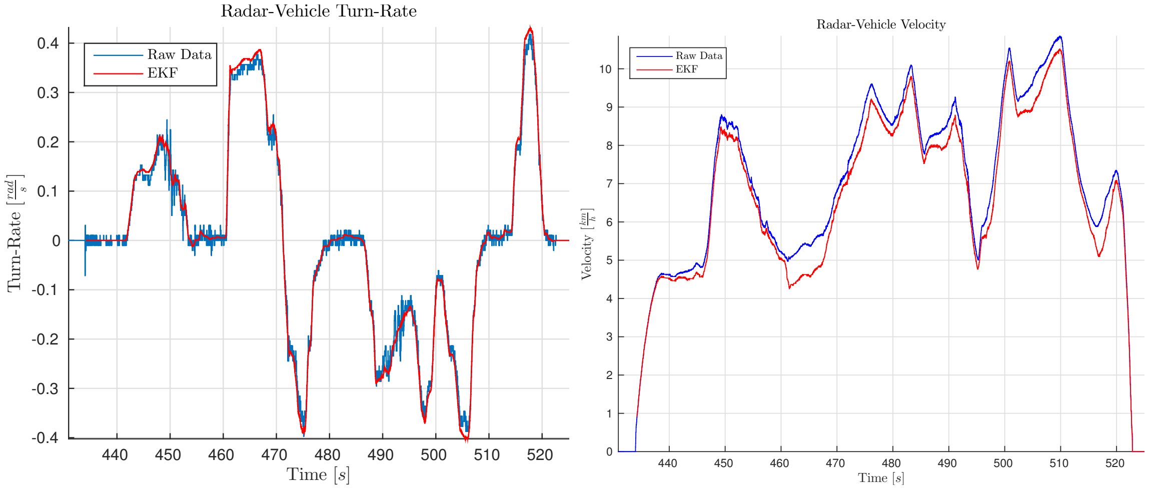
(a) position

(b) heading

Figure 11: Comparison between input data and EKF output for position and heading. Raw data is shown by the blue line, EKF data is represented by the red line.

The EKF turning-rate and velocity estimates can be seen in Figure 12. The raw data in the turning rate plot is the $\omega^{virt}$ time series (equation (68)). The signal is a function of the two rear-wheel speeds; as such, it is sensitive to the errors in those quantities. The discretization errors found in the individual wheel speeds can be seen as a source of noise about the signal.

The raw data in the velocity plot is the $V^{virt}$ time series (equation (67)), which is also a function of the two rear-wheel speeds. As opposed to the $\omega^{virt}$, it is an effective average of the speeds rather than a difference. Therefore, this source of noise in the $V^{virt}$ signal is not as prominent.



(a) turning-rate

(b) velocity

Figure 12: Comparison between input data and EKF output for turning-rate and velocity. Raw data is shown by the blue line, EKF data is represented by the red line.

Both the turning-rate and velocity signals are functions of the individual rear wheel radii. Given a nominal wheel radius, the EKF estimates the error of each rear-wheel. The time series of the estimated rear-wheel radii can be seen in Figure 13. The convergence of these signals to constant values is difficult given the limited sensors; no access to an onboard gyroscope, for example.

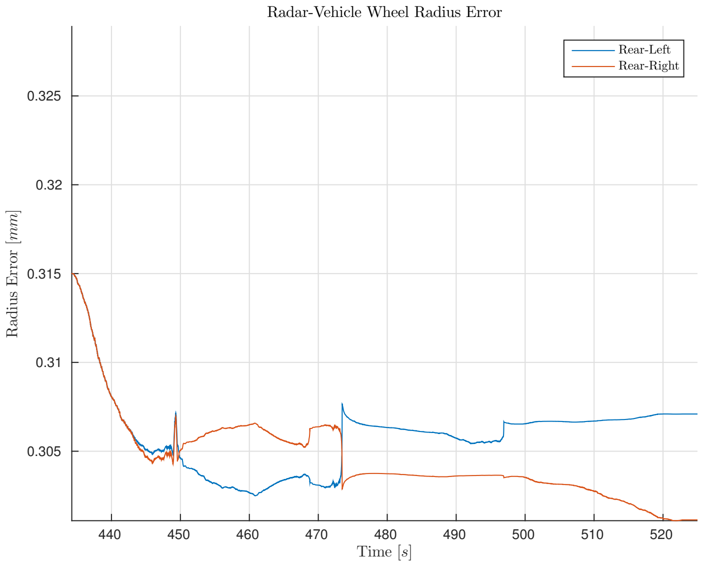


Figure 13: Estimated wheel radii.

No true-radii values are available but it can be seen that the estimates are less than the nominal value. Looking once again to the velocity plot, this fact can can be seen as the EKF velocity estimate is less than the virtual sensor time series. A similar statement can be made about the turning rate, the difference between in the rear-wheel radii produces the disparity between the estimated and virtual signals. This is most apparent at greater turning-rates.

Using additional test cases not presented here, it is verified that from the limited and noisy signals available to the system the EKF produces a full state estimate that can be used to extend the tracker to a moving radar-vehicle.

# CHAPTER V

# CONCLUSION

## *5.1 Closing Remarks*

The system developed in this thesis has met the objectives defined in the Introduction. Using the particle filter framework, the tracker is able to handle the nonlinear model describing the kinematics of the tracked vehicles as well as the nonlinear observation model mapping the state-space to the measurement-space. The particle filter is also unrestrictive on what noise distributions can be used. This was beneficial during the measurement update procedure.

Also provided by the particle filter framework is a mechanism to allow for the detection of new vehicles within the radar's field of view. Once a vehicle is identified it is tracked during the refinement stage until it exits or is lost. The management of tracked vehicles allows the system to run continously and only focus on relevant vehicles. This has been proven by the test case shown in the results.

The system has shown that it can handle tracking mutliple vehicles simultaneously using a collection of particle filters. To facilitate real-time operation the complex data association step has been relaxed. This allows the spawned trackers to operate independently of each other.

In both test cases the environment is cluttered with data points originating from non-vehicle objects and potential false alarms. The pruning and gating techniques developed sufficiently mitigate the effects of the irrelevant points so that the vehicles of interest may be detected and tracked. The success of these methods positions the system to be

Finally, a method to extend the tracking sytem has been developed. Given a limted set of measurements the state of the radar-vehicle is estimated using an Extended Kalman Filter. With this estimated state, mappings from the radar-frame to an inertial-frame, and vice versa, were derived such that a modified version of the tracking system could be used. The EKF was tested using experimental data and showed that a suitable state estimate could be produced with the limited, noisy sensor data.

## 5.2 Future Research

A number of research avenues exist to improve and extend the capability of the developed system. Testing of the current system in a controlled but real-world environment, where the tracked vehicle's states are knwon, is needed to rigorously verify the tracker's accuracy when coupled to the radar sensor; something simulations cannot adequately replicate. A controlled environment can also facilitate the testing of the system's limits, including vehicle size, separation, maneuvering, and occlusions.

A key step to fully extend the system's capability is the testing of the methods developed to enable the use of the tracking system on a moving vehicle. The modifications proposed, specifically the identification of stationary objects, pose a research endeavour unto itself. The problem of identifying data points from stationary objects and relevant vehicles jointly pushes the need for a more rigorous approach to data assocation.

# REFERENCES

[1] A.D.C GmbH, *SRR 20X/-2/-2C/-21 datasheet*, 4 2012. Version: 01.

[2] Arulampalam, M. S., Maskell, S., Gordon, N., and Clapp, T., "A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking," *IEEE Transactions on Signal Processing*, vol. 50, pp. 174–188, Feb. 2002.

[3] Bar-Shalom, Y., Daum, F., and Huang, J., "The probabilistic data association filter," *IEEE Control Systems*, vol. 29, pp. 82–100, Dec 2009.

[4] Bar-Shalom, Y. and Tse, E., "Tracking in a cluttered environment with probabilistic data association," *Automatica*, vol. 11, no. 5, pp. 451–460, 1975.

[5] Bellet, A., Habrad, A., and Sebban, M., *Metric Learning*. Synthesis Lectures on Artificial Intelligence and Machine Learning, Morgan & Claypool, 2015.

[6] Bergman, N., *Recursive Bayesian Estimation: Navigation and Tracking Applications*. PhD thesis, Linköping University, Linköping, Sweden, 1999.

[7] Bruno, M. G., *Sequential Monte Carlo Methods for Nonlinear Discrete-Time Filtering*. No. 11 in Sythesis Lectures on Signal Processing, San Rafael, California: Morgan & Claypool, 2001.

[8] Carlson, C. R., Gerdes, J. C., and Powell, J. D., "Practical position and yaw rate estimation with gps and differential wheelspeeds," in *Proc. 6th Int. Symp. AVEC*, 2002.

[9] Chen, S. and Huang, W., "Target tracking using particle filter with x-band nautical radar," in *2013 IEEE Radar Conference (RadarCon13)*, pp. 1–6, April 2013.

[10] Cox, I. J., "A review of statistical data association techniques for motion correspondance," *International Journal of Computer Vision*, vol. 10, no. 1, pp. 53–66, 1993.

[11] Daum, F., "Multitarget-multisensor tracking: Principles and techniques [bookshelf]," *IEEE Control Systems*, vol. 16, pp. 93–, Feb 1996.

[12] Douc, R., Cappé, O., and Moulines, E., "Comparison of Resampling Schemes for Particle Filtering," *eprint arXiv:cs/0507025*, July 2005.

[13] Doucet, A., de Freitas, N., and Gordon, N., "An introduction to sequential monte carlo methods," in *Sequential Monte Carlo Methods in Practice* (Doucet, A., de Freitas, N., and Gordon, N., eds.), ch. 1, pp. 3–14, New York, New York: Springer New York, 2001.

[14] Doucet, A., de Freitas, N., and Gordon, N., *Sequential Monte Carlo Methods in Practice.* New York, New York: Springer New York, 2001.

[15] Floudas, N., Polychronopoulos, A., and Amditis, A., "A survey of filtering techniques for vehicle tracking by radar equipped automotive platforms," in *2005 7th International Conference on Information Fusion*, vol. 2, pp. 8 pp.–, July 2005.

[16] Gelb, A., *Applied Optimal Estimation.* MIT Press, 1974.

[17] Gordon, N., Salmond, D., and Smith, A., "Novel approach to nonlinear/non-gaussian bayesian state estimation," *IEE Proceedings F. Radar and Signal Processing*, vol. 140, pp. 107–113, Apr. 1993.

[18] Gunnarsson, J., Svensson, L., Danielsson, L., and Bengtsson, F., "Tracking vehicles using radar detections," in *2007 IEEE Intelligent Vehicles Symposium*, pp. 296–302, June 2007.

[19] Gustafsson, F., Gunnarsson, F., Bergman, N., Forssell, U., Jansson, J., Karlsson, R., and Nordlund, P. J., "Particle filters for positioning, navigation, and tracking," *IEEE Transactions on Signal Processing*, vol. 50, pp. 425–437, Feb 2002.

[20] Gustafsson, F. and Isaksson, A. J., "Best choice of coordinate system for tracking coordinated turns," in *Decision and Control, 1996., Proceedings of the 35th IEEE Conference on*, vol. 3, pp. 3145–3150, Dec 1996.

[21] Hol, J. D., Schon, T. B., and Gustafsson, F., "On resampling algorithms for particle filters," in *Nonlinear Statistical Signal Processing Workshop, 2006 IEEE*, pp. 79–82, Sept 2006.

[22] Jammalamadaka, S. R. and SenGupta, A., *Series on Multivariate Analysis : Topics in Circular Statistics-Vol 5*, vol. 5. WSPC, 2001.

[23] Jaward, M., Mihaylova, L., Canagarajah, N., and Bull, D., "Multiple object tracking using particle filters," in *2006 IEEE Aerospace Conference*, pp. 8 pp.–, Mar 2006.

[24] Jazar, R. N., *Vehicle Dynamics: theory and application.* Springer New York, 2nd ed., 2014.

[25] Julier, S. J. and Uhlmann, J. K., "New extension of the kalman filter to nonlinear systems," vol. 3068, pp. 182–193, 1997.

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

[27] Kong, A., Liu, J. S., and Wong, W. H., "Sequential imputations and bayesian missing data problems," *Journal of the American Statistical Association*, vol. 89, pp. 278–288, Mar. 1994.

[28] Liu, F., Sparbert, J., and Stiller, C., "Immpda vehicle tracking system using asynchronous sensor fusion of radar and vision," in *Intelligent Vehicles Symposium, 2008 IEEE*, pp. 168–173, June 2008.

[29] Liu, P., Li, W., Wang, Y., and Ni, H., "On-road multi-vehicle tracking algorithm based on an improved particle filter," *IET Intelligent Transport Systems*, vol. 9, pp. 429–441, May 2015.

[30] Lymperopoulos, I., Chaloulos, G., and Lygeros, J., "An advanced particle filtering algorithm for improving conflict detection in air traffic control," in *International Conference on Research in Air Transportation (ICRAT)*, (Budapest, Hungary), jun 2010.

[31] McGee, L. A. and Schmidt, S. F., "Discovery of the kalman filter as a practical tool for aerospace and industry," Tech. Rep. NASA-TM-86847, National Aeronautics and Space Administration, Moffett Field, California, Nov. 1985.

[32] Morelande, M. R. and Musicki, D., "Fast multiple target tracking using particle filters," in *Proceedings of the 44th IEEE Conference on Decision and Control*, pp. 530–535, Dec 2005.

[33] Mr. Bayes, M. P., "An essay towards solving a problem in the doctrine of chances. by the late rev. mr. bayes, f. r. s. communicated by mr. price, in a letter to john canton, a. m. f. r. s.," *Philosophical Transactions (1683-1775)*, vol. 53, pp. 370–418, 1763.

[34] Mukhtar, A., Xia, L., and Tang, T. B., "Vehicle detection techniques for collision avoidance systems: A review," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, pp. 2318–2338, Oct 2015.

[35] Niederreiter, H., *Random Number Generation and Quasi-Monte Carlo Methods.* Society for Industrial and Applied Mathematics, 1992.

[36] Otto, C., Gerber, W., Len, F. P., and Wirnitzer, J., "A joint integrated probabilistic data association filter for pedestrian tracking across blind regions using monocular camera and radar," in *Intelligent Vehicles Symposium (IV), 2012 IEEE*, pp. 636–641, June 2012.

[37] Özkan, E., Lundquist, C., and Gustafsson, F., "A bayesian approach to jointly estimate tire radii and vehicle trajectory," in *2011 14th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, pp. 1622–1627, Oct 2011.

[38] Rachev, B. and Smrikarov, A., eds., *CompSysTech '03: Proceedings of the 4th International Conference Conference on Computer Systems and Technologies: E-Learning*, (New York, NY, USA), ACM, 2003.

[39] Reid, D. B., "An algorithm for tracking multiple targets," in *Decision and Control including the 17th Symposium on Adaptive Processes, 1978 IEEE Conference on*, pp. 1202–1211, Jan 1978.

[40] Schön, T., *On Computational Methods for Nonlinear Estimation.* PhD thesis, Linköping University, Linköping, Sweden, 2003.

[41] Sun, Z., Bebis, G., and Miller, R., "On-road vehicle detection: a review," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, pp. 694–711, May 2006.

[42] Swift Navigation, Inc., *Piksi Datasheet*, 3 2016. Version 2.3.1.