THE UNIVERSITY OF
SYDNEY

Cooperative Vehicle Tracking in Large Environments

Mao Shan

A thesis submitted in fulfillment
of the requirements of the degree of
Doctor of Philosophy

THE UNIVERSITY OF
SYDNEY

Australian Centre for Field Robotics
School of Aerospace, Mechanical and Mechatronic Engineering
The University of Sydney

August 2013

Declaration

I hereby declare that this submission is my own work and that, to the best of my knowledge and belief, it contains no material previously published or written by another person nor material which to a substantial extent has been accepted for the award of any other degree or diploma of the University or other institute of higher learning, except where due acknowledgement has been made in the text.

Mao Shan

31^{st} August, 2013

Reprinted with corrections and emendations 25^{th} February, 2014

Abstract

Mao Shan
The University of Sydney

Doctor of Philosophy
August 2013

Cooperative Vehicle Tracking in Large Environments

Vehicle position tracking and prediction over large areas is of significant importance in many industrial applications, such as mining operations. In a small area, this can be easily achieved by providing vehicles with a constant communication link to a control centre and having the vehicles broadcast their position. The problem changes dramatically when vehicles operate within a large environment of potentially hundreds of square kilometres and in difficult terrain. This thesis presents algorithms for cooperative tracking of vehicles based on a vehicle motion model that incorporates the properties of the working environment, and information collected by infrastructure collection points and other mobile agents.

The probabilistic motion prediction approach provides long-term estimates of vehicle positions using acceleration, speed, and timing profiles built for the particular environment and considering the probability that the vehicle will stop. A limited number of data collection points distributed around the field are used to update the position estimates, with negative (no communication) information also used to improve the estimation.

The thesis introduces the concept of observation harvesting, a process in which peer-to-peer (P2P) communication between vehicles allows egocentric position updates and inter-vehicle measurements to be relayed among vehicles and finally conveyed to the collection points for an improved position estimate. It uses a store-and-synchronise concept to deal with intermittent communication and aims to disseminate data in an opportunistic manner.

A nonparametric filtering algorithm for cooperative tracking is proposed to incorporate the information harvested, including the negative, relative, and time delayed observations. An important contribution of this thesis is to enable the optimisation of fleet scheduling when full coverage networks are not available or feasible. The proposed approaches were validated with comprehensive experimental results using data collected from a large-scale mining operation.

Acknowledgements

There are many people to whom I would like to express my gratitude. This thesis would not have been possible without their valuable support over the past four years.

First of all, I would like to thank my supervisor, Prof. Eduardo Nebot. Thank you for generously sharing your expertise and experience with me. Thanks for your inspiration when I ran out of ideas, your advice when I got confused, and your encouragement when I was frustrated. I would also like to thank my co-supervisor, Dr. Stewart Worrall. Thanks for your patience, your advice, and alternative perspective, in particular when discussing some details of the research.

Next, I would like to thank many people in the ACFR. Andres, thank you for sitting next to me for more than three years. You always seemed much more hard-working than I, which made me feel embarrassed. Vijay, David, Andrew, Gabriel, and Tim, the help from you guys is much appreciated. In addition, I would like to thank the faculty and ACFR for the financial assistance I have received as a student.

I would also like to thank my friends from the uni and the church. I would have turned into a robot without the distraction from you. In particular, I must thank Zhuo for helping me think positive. Special thanks go to Adam and Lily and their little lovely daughter Grace, for providing countless meals as well as massive computational resources. Without a doubt, both of these are of great significance in facilitating the research process.

Of course, I would like to thank my parents for always being there to back me up. Thanks for your love and care that will last a lifetime. And I should say a big thank you to my sister Carole for always being available whenever I need help.

Lastly, I thank Jesus Christ, my Lord, my Saviour. Without You, everything I have been doing is meaningless.

Dedicated to my family

Contents

List of Figures

List of Tables

Nomenclature

General Symbol Format

a, A	Scalar or scalar valued function
$\mathbf{a}, \mathbf{A}$	Vector/matrix or vector/matrix valued function (all vectors are assumed column vectors)

Notation

$\hat{(\cdot)}$	An approximation, estimate of $(\cdot)$
$\bar{(\cdot)}$	Mean value of $(\cdot)$
$\widetilde{(\cdot)}$	$(\cdot)$ that could be time delayed
$(\cdot)_t$	$(\cdot)$ at discrete time t
$(\cdot)_{t\|t-1}$	The difference of $(\cdot)$ at times $t-1$ and t
$(\cdot)_{t_1:t_2}$	The collection of $(\cdot)$ from t_1 to t_2
$(\cdot)_{a\to b}$	$(\cdot)$ for a to b
$(\cdot)^-$	Prediction for $(\cdot)$
$(\cdot)^+$	Posterior of $(\cdot)$
$(\cdot)^a$	$(\cdot)$ for a
$(\cdot)^{a\to b}$	$(\cdot)$ for a to b
$a \triangleq b$	a is defined as b
$a \setminus b$	Relative complement operation of b in a

Functions and Operators

$(\cdot)^T$	Vector/matrix transpose
$(\cdot)^{-1}$	Matrix inverse
$\mathbf{f}(\cdot)$	Non-linear state transition model
$\mathbf{h}(\cdot)$	Non-linear observation model

$H(\cdot)$	Entropy of a random variable $(\cdot)$
$min(\cdot)$	Minimum value of $(\cdot)$
$\mathcal{N}(\boldsymbol{\mu}, \mathbf{C})$	Gaussian probability density function defined with mean $\boldsymbol{\mu}$ and covariance matrix $\mathbf{C}$
$pm(\cdot)$	Probability mass
$P(\cdot)$	Probability distribution of a random variable $(\cdot)$
$Pm_{i \to j}$	Motion model for state i to j
$Ps_{i \to j}(\cdot)$	State transition function for state i to j
$q(\cdot)$	Proposed distribution
$St(\mu, \lambda, \nu)$	Student's t-distribution defined with mean μ, precision λ, and degrees of freedom ν
$\epsilon(\cdot)$	NEES value of $(\cdot)$
$O(\cdot)$	Time complexity of $(\cdot)$
$\nabla(\cdot)$	Gradient of vector/matrix $(\cdot)$

Variables and Symbols

$a, \mathbf{a}$	Acceleration, acceleration vector
B	Wheelbase
$\mathbf{B}$	Control input model, or local vehicle observation pool
d	Relative distance
$\mathbf{F}$	Linear state transition matrix
$\mathbf{G}$	Transition matrix
$\mathbf{H}$	Linear observation model
$\mathbf{i}$	Information gain vector
$\mathbf{I}$	Information gain matrix
$\mathbf{J}$	Jacobian matrix
L	Particle quantity
$\mathbf{L}$	The most informative observation vector
$\mathbf{Q}$	Process noise matrix
$r, \mathbf{r}$	Position, position vector
R	Threshold of acceleration speed correlation coefficient
$\mathbf{R}$	Sensor noise matrix
S	Motion state of a vehicle
t	Time
T	Time point
$\mathbf{u}$	Control input
v, V	Speed
$\mathbf{v}$	Speed, or sensor noise

w	Particle weight
$\mathbf{w}$	Process noise
$\mathbf{x}$	State vector of a node
$\mathbf{X}$	Joint state space of nodes
$\bar{\mathbf{X}}$	Joint state space of auxiliary nodes
$\mathbf{y}$	Information state vector
$\mathbf{Y}$	Information matrix
$z, \mathbf{z}$	Observation, observation vector
$\mathbf{Z}$	A collection of observations
γ	Steering angle
$\Delta t, \Delta T$	Time step length
Θ	A particle base
λ	Precision of Student's t-distribution
$\widetilde{\Lambda}$	Likelihood function of a relative observation
$\mu, \boldsymbol{\mu}$	Mean value, mean vector
ν	Degrees of freedom of Student's t-distribution
ξ	Condition for the state transition
ρ	Acceleration speed correlation coefficient
σ	Standard deviation
τ	Time length
Ψ	Likelihood function of an absolute referenced observation
$\widetilde{\Psi}$	Likelihood function of an absolute egocentric observation
$\boldsymbol{\omega}$	A subset of the global observation pool
$\boldsymbol{\Omega}$	Global observation pool

Abbreviations

ASP Acceleration Speed Profile

CL Cooperative Localisation

DSCPF Delayed-State Cooperative Particle Filter

DSKF Delayed-State Kalman Filter

DTN Delay/Disruption Tolerant Network

EKF Extended Kalman Filter

GPS Global Positioning System

GSCPF Gibbs Sampler based Cooperative Particle Filter

IA Intersections and Areas

IR In Range

KF Kalman Filter

MANET Mobile Ad-Hoc Network

MCMC Markov Chain Monte Carlo

MIOS Most Informative Observation Set

MIOV Most Informative Observation Vector

MOR Moving on Roads

NC Negative Connectivity

NEES Normalised Estimation Error Squared

OR Out of Range

P2P Peer-to-Peer

PC Positive Connectivity

PDF Probability Density Function

PF Particle Filter

PIA Passing Intersections and Areas

PVA Position-Velocity-Acceleration

RMSE Root Mean Squared Error

RS Road Segments

RSSI Received Signal Strength Intensity

SLAM Simultaneous Localisation and Mapping

SOR Stopping on Roads

UWB Ultra-Wide Band

V2I Vehicle-to-Infrastructure

V2V Vehicle-to-Vehicle

VANET Vehicular Ad-Hoc Network

WMN Wireless Mesh Network

WSN Wireless Sensor Network

Chapter 1

Introduction

1.1 Tracking Mobile Targets in Large Environments

The localisation and tracking of vehicles or personnel is a broad area of research with applications in traffic management, autonomous systems, and safety to name a few. The focus of this thesis is on the tracking of mobile agents in large areas in which environmental uncertainties make it difficult to pinpoint an accurate and real-time position.

Most existing tracking approaches concentrate on the tracking of vehicles and personnel in a small area, such as car parks, urban areas and indoor environments. The complexity of the process changes dramatically when the mobile target position tracking is required over a large environment of potentially hundreds of square kilometres. This is further complicated in environments with difficult terrains. The mining scenario is one of the examples of tracking applications with large operation areas and high environmental uncertainties. As shown in Figure 1.1, the Bingham Canyon Mine, is a typical large mining operation in a mountainous environment. Figure 1.1 only shows the dimensions of the pit. Some types of mines, in particular where the resources to extract are not deep, e.g. bauxite mines, can extend for more than 100 kilometres in length.

Figure 1.1 – The Bingham Canyon Mine, Salt Lake City, Utah, United States, the deepest open-pit mine in the world. Over 1.21 km deep and 4 km wide, the pit is reported as the world's largest man-made excavation [170].

Usually, a mining operation involves various types of mobile agents that could be mainly grouped into vehicles and personnel. The vehicles include haul trucks and light vehicles. Personnel refer to vehicle drivers, operators, and so on. Additionally, a mine also has infrastructures installed, such as a base station, data collection towers and mechanical equipments. Figure 1.2 illustrates some of the examples. The tracking of these mobile agents (e.g., vehicles) in the large outdoor environment is of great significance for the purposes of productivity optimisation, resource utilisation improvement and so on.

The most straightforward solution to the tracking problem is to have the vehicles transmit the position and state information obtained from their egocentric positioning devices (e.g., GPS) to a central "control room" using a communication infrastructure installed to give network coverage in the areas in which the vehicles operate. This

Figure 1.2 – Resources in a mining operation.

also requires the installation of wireless communication hardware on the vehicle-side, see Figure 1.3. In these mining scenarios, however, it is only feasible to install data collection infrastructure in a small number of locations - because of power, network and cost constraints - meaning that the vehicles will only be in communication with the network infrastructure for a fraction of the operating time.

This thesis examines the tracking (predicting) of vehicles' positions during a long-term observation outage and proposes novel algorithms that make use of long-term motion prediction, vehicle interactions and cooperative tracking. The following section presents a motivating example with more details on tracking vehicles in a large area.

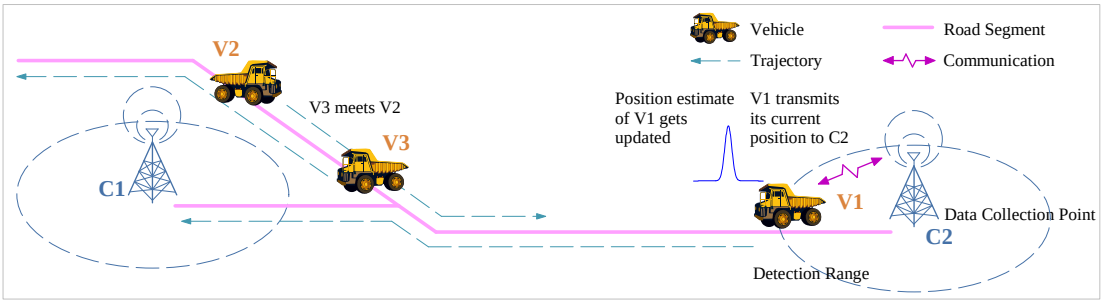
Figure 1.3 – Wireless communication devices mounted on a vehicle. The figure shows the installation of radio transceivers and associated antennas on mining haul truck.

1.2 A Motivating Example and Problem Statement

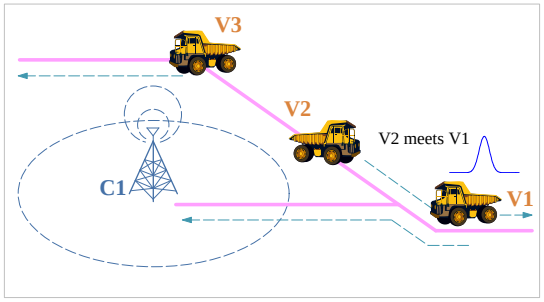
In this section, a motivating tracking example is presented and analysed to assist in the understanding of the practical constraints on tracking vehicles in large environments.

Figure 1.4 shows a multiple-vehicle tracking example in such a type of large environment: three vehicle agents V1, V2 and V3 equipped with GPS are moving in a large-scale area with two fixed data collection points, C1 and C2 installed, which are connected to a central base station. The base station (not shown in the figure) tracks these vehicles with information acquired from each data collection point, which has communication with the vehicles but is limited by range.

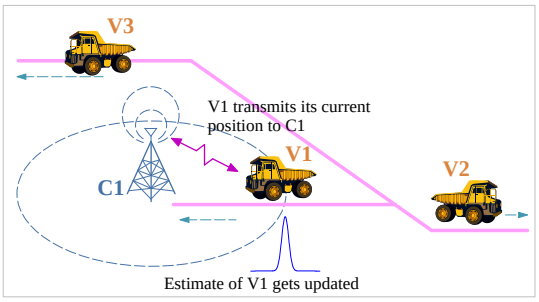
The GPS has been widely used to provide egocentric positioning information to mobile agents. With good visibility to many satellites, the positioning accuracy is sufficient (on the order of 10^0 metres) to localise vehicles' moving on roads. The communication devices mounted on the vehicles and the infrastructure allow the position information of a vehicle to be accurately known by the base station for the tracking purpose. It requires that both the vehicle and the infrastructure are in the communication



(a) $t_{now-m-n}$



(b) t_{now-n}



(c) t_{now}

Figure 1.4 – The motivating example of tracking multiple vehicles in a large environment. V1 started its journey from data collection point C2 at time $t_{now-m-n}$, heading towards C1, and in the meanwhile V2 and V3 met somewhere far away from both C1 and C2. At a later time t_{now-n} V1 met V2 on its way. Lastly V1 arrives at C1 at time t_{now}.

range of one another. The ideal solution to the tracking problem is to have complete site-wide communication coverage.

However, full network coverage is not practically feasible because of a large area of operation or geographical constraints of the terrain. The communication devices are of limited communication range because of power constraints, while deploying sufficient data collection points to provide complete site-wide communication becomes very expensive. Consequently, it is only feasible to provide sparse coverage and the targets may be out of the communication range of the infrastructure for a long period of time. In Figure 1.4, V1 was found to be only under detection by C2 and C1 before time point $t_{now-m-n}$ and after t_{now}, respectively. The fundamental problem was that, no information about the position of V1 was available while not in contact with the infrastructure. The information outage could last for minutes in the example large

environment. The problem is even more serious for V2 and V3, because they were constantly out of detection by any data collection points in the example.

Furthermore, even when a vehicle establishes communication with the infrastructure, the GPS information it provides may be useless because of the hardware malfunctions or limited visibility to GPS satellites. Often, a reduction in the quantity of visible GPS satellites, or a large geometric dilute of precision (GDOP) value, which indicates poor accuracy of positioning, is caused due to nearby mountains or canyon walls, or buildings in outdoor environments. Figure 1.5 demonstrates such an example, in which a degraded condition of GPS reception happens when vehicles descend into the pit of a mine. Under this circumstance, the base station also undergoes an outage of useful observations to constrain the position estimates of the vehicles.

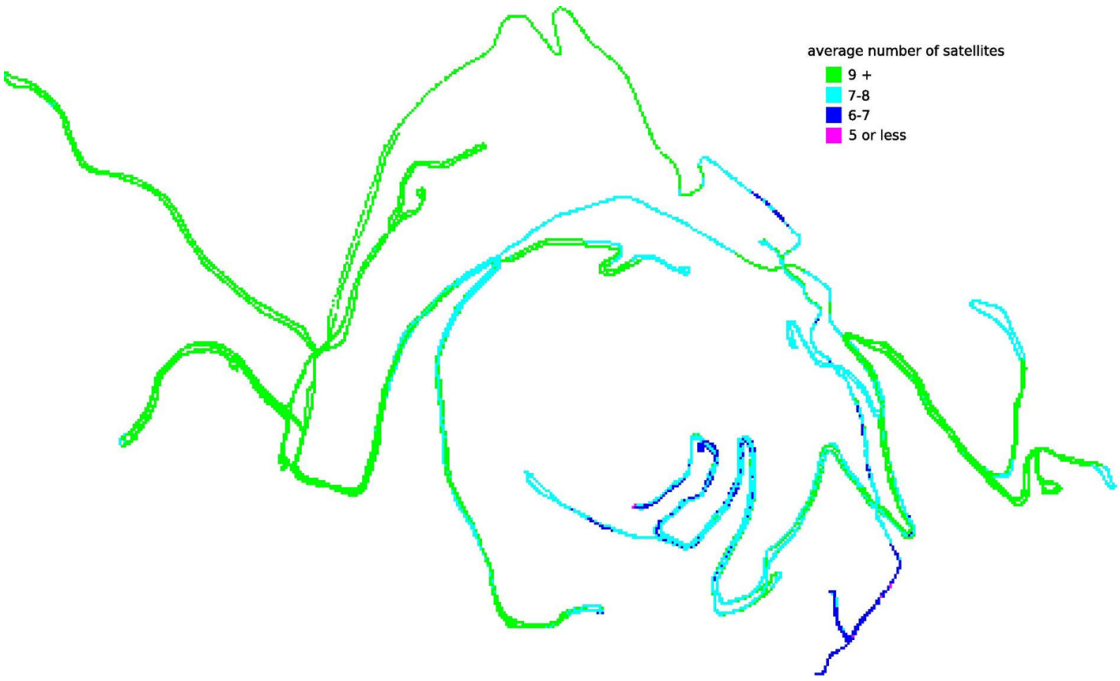


Figure 1.5 – The GPS reception in an open-pit mine [173]. Though the majority part of the site has good satellite coverage, the number of available satellites for positioning is reduced when vehicles descend into the pit. The degraded reception condition tends to deteriorate in a deeper pit.

During the observation outage period, the tracking system has to rely on motion prediction only. Generally, the uncertainty of the target position grows substantially in the absence of position observations and the target position estimate becomes

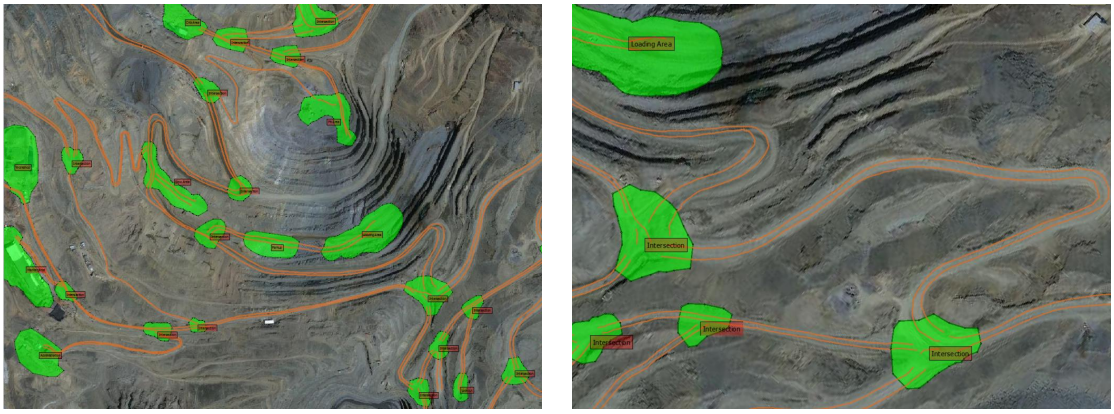
unusable after a short period of time. The lack of informative tracking observations about the target(s) for a considerable length of time, is defined as an "observation outage problem" in this thesis. A preliminary study on how the observation outage duration could be possibly reduced is presented in Appendix A.

The observation outage problem has not garnered much attention in the literature. Most of the existing work focuses on tracking mobile targets in small areas in which the targets tracked are constantly within the detection range of the sensors. In conventional vehicle tracking systems, frequent measurements are required to reduce the uncertainty of vehicles' positions. When the tracking is required in a large environment, the observation outage problem inevitably occurs, and new approaches should be developed to address this particular problem. This is the primary motivation for the research in this thesis. The next section presents an overview of the approaches proposed in this thesis to address the vehicle tracking problem in large environments.

1.3 Contributions of the Thesis

1.3.1 Overview

In the motivating example the observation outage period is between $t_{now-m-n}$ and t_{now} for V1, i.e., a time duration of $m + n$ seconds. And the base station accessed no information about V2 and V3 throughout the motivating example. When a standard vehicle motion model is used, which is ineffective for long-term motion prediction, the uncertainty of the position estimate grows dramatically with time. Therefore, the first component required for the tracking problem in large environments is a reliable and comparatively more accurate motion prediction model that is able to produce consistent position predictions of agents over a long-term time period. This is realised in the thesis by incorporating the environment properties and historical vehicle motion data into the motion model. The motion model also considers the probability that a vehicle may stop on a road anywhere, at any time. In addition,



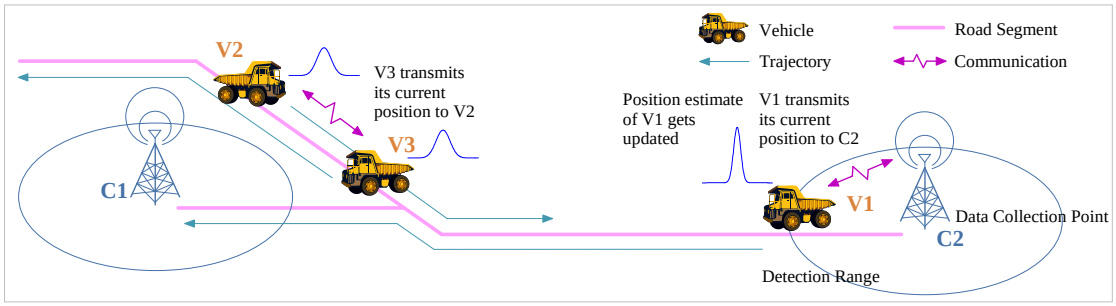
(a) Overview (b) Details

Figure 1.6 – Features in a mining operation. a) The environment can be represented by a road network based on context areas (loading areas, parking lots, etc.) connected by winding road segments and intersections formed by crossing roads. b) Several intersections and a loading area as well as roads connecting them are clearly demonstrated in a detailed picture.

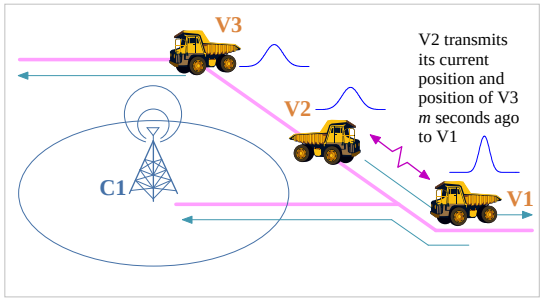
the road map information that is illustrated in Figure 1.6 could also be considered as *a priori* knowledge to assess the vehicle motion. The review on motion prediction models is elaborated upon in Section 2.4.

In the motivating example, vehicles were equipped with communication devices, though only to communicate with the data collection points. The tracking in the example is then improved when we enable a peer-to-peer (P2P) communication capability for the vehicles, see Figure 1.7. During the period when vehicles were out of the detection range of data collection points, they were able to communicate with each other and exchange information. As respectively shown in Figures 1.7(a) and 1.7(b), the inter-vehicle information exchange happened twice in the example, at times $t_{now-m-n}$ and t_{now-n}, and the position information of V2 and V3 is lastly brought to C1 (and the base station in the meanwhile) by V1 at time t_{now}, see Figure 1.7(c). Figure 1.8 clearly shows the vehicle-to-infrastructure (V2I) and vehicle-to-vehicle (V2V) interactions, along with time. The information, though time delayed, is used as observation updates to the position filter, and the estimation updates for all three vehicles are achieved at time t_{now}.

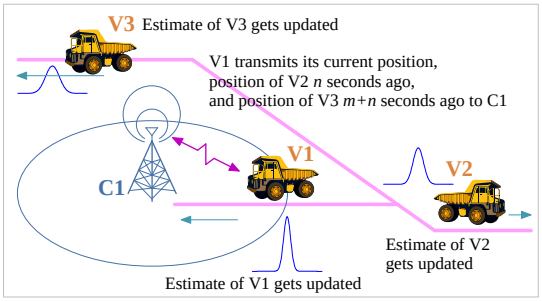
The V2V and V2I interactions described above, which allow the position information



(a) $t_{now-m-n}$



(b) t_{now-n}



(c) t_{now}

Figure 1.7 – The benefit of the observation harvesting comes from the interactions between vehicles within the communication range, and bringing information to the base station for the filter update stage. In (a), the position information of vehicle V3 was known to V2 when they met. The information was then forwarded to V1 when V1 and V2 moved close, as shown in (b). Lastly, in (c), through data collection point C1 the base station received the information, though delayed for V2 and V3, from returning V1 and estimation updates for all three can be evaluated.

of vehicles to be relayed and eventually collected by the infrastructure, are defined as "observation harvesting mechanism" proposed in the thesis. With the observation harvesting, a direct connection between a vehicle and a fixed data collection point is not required to update the position estimate. Each vehicle that returns to a fixed data collector acts as an information carrier for those vehicles not in contact with the infrastructure. In the example, even though no direct connection was established between the infrastructure and the two vehicles, V2 and V3, their information was able to be disseminated via V1. This is also illustrated in the form of an information flow in Figure 1.8. To give readers a better understanding of the related background, data dissemination techniques are reviewed in Section 2.5.

The information of interest in the research not only includes GPS updates, but also

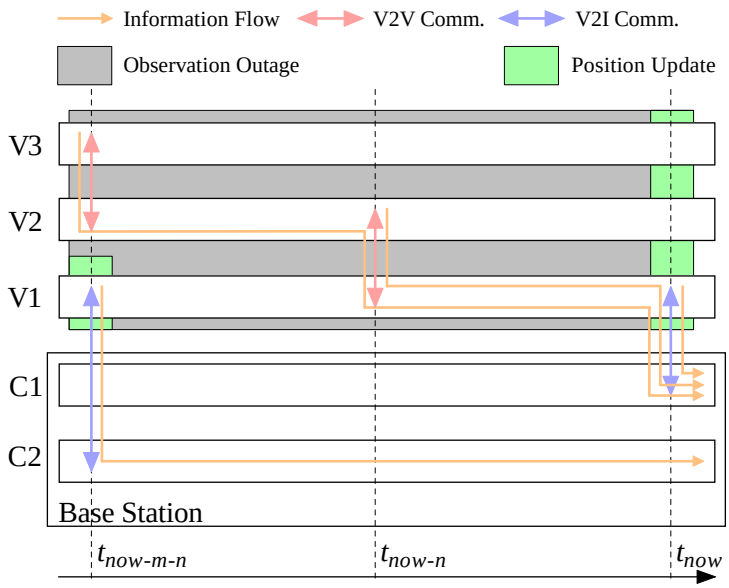


Figure 1.8 – Event time line in the motivating example.

comprises other types of observations, such as positive and negative detection of vehicles by a data collector, relative ranging measurements between vehicles and so forth. The observations received from the returning V1 are time delayed, but still useful to determine the positions of V2 and V3. These complimentary observations assist in constraining the possible vehicle locations in the absence of real-time GPS updates. In general, information collected about vehicles in the field is classified in this thesis, according to the information source, usage, method of transmission and reception. A detailed description on this part could be found in Section 3.2.

Every observation is collected and fused in the fusion centre to give estimation updates of every vehicle that is being tracked in the field. While most of the existing approaches deal with real-time and positive observations only, the fusion stage in the proposed approaches is capable of taking into account all kinds of information collected, including the negative, relative and time delayed. The tracking method that makes use of inter-node collaboration is also known as cooperative tracking in the literature. A key difference between the cooperative localisation/tracking and the non-cooperative counterpart is that the former considers relative measurements between these nodes without location awareness in its fusion stage, in addition to

measurements between position-unknown nodes and position-known/reference nodes. This allows for higher system robustness and tracking accuracy, particularly in scenarios with degraded GPS readings or even GPS outages. A comprehensive review of the data fusion approaches is presented in Section 2.6.

In summary, this thesis is concerned with long-term motion prediction and cooperative tracking in large environments. The tracking approaches presented include a motion prediction algorithm that incorporates parameters obtained from the environment and the vehicle history to obtain consistent long-term vehicle motion prediction. It uses a limited number of data collection points distributed around the field to update vehicle position estimates when in range and then predict vehicle positions at points in between. In addition, the thesis introduces the concept of observation harvesting, a process in which P2P communication between vehicles allows egocentric position updates and inter-vehicle measurements to be relayed among vehicles, and finally conveyed to the base station. The approaches also comprise a nonparametric algorithm for cooperative tracking using all kinds of information harvested, including negative and time delayed observations. Figure 1.9 illustratively summarises the challenges that emerge in tracking in large environments, and the corresponding approaches proposed in the thesis. The remainder of this section further introduces the proposed approaches with respect to three different categories.

1.3.2 Probabilistic Motion Prediction

The long-term motion prediction model (**A1** in Figure 1.9) proposed is based on the fact that the base station is likely to lose connection to vehicles for a significant period of time when working in large environments. Because of the lack of full coverage of sensing network (**C1** in the figure), the tracking has to rely on sporadic observations (**C2**) from V2I and V2V interactions; for the rest of time it depends on the prediction of vehicle positions over a long-term period (**C3**). Using a vehicle model that does not incorporate external environment properties, such as possible vehicle behaviour, interactions between individuals, weather conditions and traffic networks, will gen-

Figure 1.9 – An overview of the proposed approaches.

erally result in a very uncertain position estimation after a short period. As will be demonstrated in Section 2.3, the uncertainty of the vehicle position grows quickly in the absence of position observations when a standard vehicle dynamic model is used.

The algorithm proposed for long-term vehicle prediction and tracking is based on a model of the vehicle that incorporates the learnt properties of an environment, such as

roads and intersections and estimated probabilistic vehicle kinematics (**D1**). Motion profiles are built up for each position slot on each road segment, with historical motion data collected from vehicles in operation. It introduces acceleration speed profiles (ASPs) into the motion model and describes vehicle movement characteristics given a particular position on a road. The model has the correlation of vehicle motion parameters, i.e., acceleration, speed, and position, implicitly built into ASPs. It further considers the stopping probability of a vehicle on a road due to unpredictable factors, which is more representative of the real world situation than conventional models. In addition, road network information (**D2**) is also integrated into the model as prior knowledge. The experiment results show that by incorporating acceleration and speed probability density functions built for the particular environment, it is possible to constrain the potential trajectory of vehicles and provide an improved prediction of the position.

1.3.3 Observation Related

The work then extends the probabilistic motion prediction approach by introducing the concept of observation harvesting (**A2**), in which a delay/disruption tolerant network (DTN) is formed by resources in the field. P2P communication between vehicles allows position updates to be exchanged and brought forward to the base station. The degraded accuracy of the position estimates is caused when only discrete observations are available, compared to conventional scenarios in which continuous information is received for targets. To overcome this limitation, the observation harvesting concept is proposed for effective and robust data dissemination among agents and the infrastructure without a global network established in the field (**C4**). It collects all sorts of useful information about every vehicle running around the field and uses a store-and-synchronise concept to deal with intermittent communication. Under this configuration, data is disseminated in an opportunistic P2P manner.

The observations to harvest include absolute ones from on-board GPS devices, relative range measurements, and positive and negative types (**C5**). The positive information

is given by the detection of a vehicle in the detection range of a data collector or another vehicle indicating its presence, while the non-detection of the vehicle by a data collection point or another vehicle is considered negative information and can be used to exclude a vehicle in a particular area. If a vehicle is not seen for a period of time, the inference from this negative information is that the vehicle is likely to be somewhere outside the detection area.

Furthermore, the proposed approaches consider time delayed information (**C5**) in the fusion stage along with real-time observations. The observations from the fixed data collection points are instantly used to update position estimates when a vehicle is in the communication range. However, because vehicles are likely to be out of detection for varying periods of time, some of the information gathered in the harvesting stage would be known to the base station with some time delays. This kind of time delayed information is incorporated into the global position estimate when a vehicle brings it back to a data collection point or the base station. Every piece of the information mentioned above helps constrain the position estimate of the vehicle from the point of view of the base station, even in the absence of GPS information.

Information is interpreted in the tracking approaches using probabilistic observation/sensor models (**A3**). These models were built based on historical V2I and V2V observations collected during operation (**D3**), and aim at reliable close proximity detection between resources depending on received signal strength intensity (RSSI) and wireless connectivity measurements.

1.3.4 Cooperative Tracking

In many operations, not all mobile nodes will have continual access to GPS. This is dependent on GPS satellite availability, malfunctions of hardware, failure of antennas, and other issues. Even when the GPS provides position information, there can be significant errors due to signal interference, multipath, atmospheric effects, or poor satellite configuration. Cooperative tracking is able to improve position estimates when there is degraded GPS accuracy or even a GPS outage (**C6**) by fusing relative

measurements between mobile nodes. This method requires a mesh network for P2P communication and the distances between adjacent mobile nodes. Generally, cooperative tracking is beneficial for improving the overall estimate accuracy of mobile nodes, as more information is fused into the inference process.

This thesis proposes a novel nonparametric cooperative tracking algorithm that makes use of a P2P ad-hoc network. Those parametric cooperative tracking methods are not appropriate due to the non-Gaussian and non-linear properties of the tracking problem concerned in this work. A Gibbs sampler based cooperative particle filter (GSCPF), particularly for nonparametric cooperative tracking (**A4**), is proposed to estimate the state of every mobile node. The filter maintains the marginal state of every node, instead of a joint state of the group, and updates the estimates using a Gibbs sampler, which is known as a Markov chain Monte Carlo (MCMC) method. The Gibbs sampler is able to sample from each low-dimensional target space conditionally on others rather than directly from the high-dimensional joint state. Instead of keeping the joint state of all target nodes, the proposed approach works with the marginal distribution of each mobile node and fuses all available egocentric observations and relative measurements between nodes. The results demonstrate that the algorithm can improve the position estimates of the mobile nodes with or without egocentric position information available.

The proposed particle filter is then extended toward a cooperative tracking method that is able to use time delayed information. A delayed-state cooperative particle filter (DSCPF) (**A5**) is proposed to estimate the positions of every mobile agent in the field, with positions predicted by the proposed long-term motion model and information collected via observation harvesting. This filter is able to deal with time delayed observations regardless of the sequence of their arrivals. It jumps to a historical time point and performs a re-propagation of the filtering process when a vehicle brings back information to any data collection points. This approach also has been validated in both simulations and experiments.

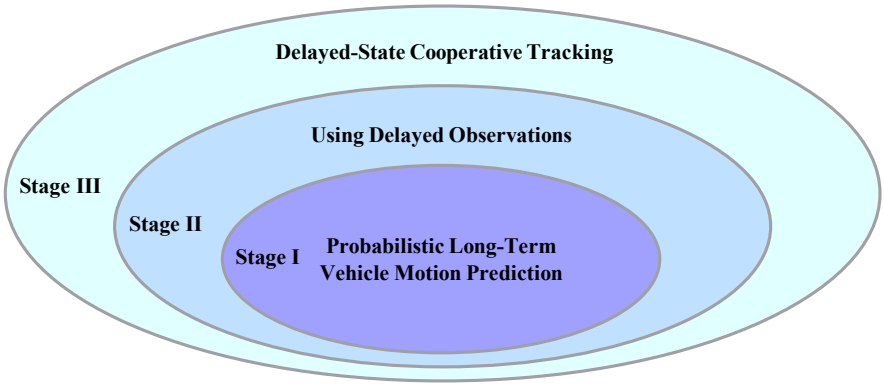


Figure 1.10 – Stages of the research.

1.3.5 Summary of Main Contributions

This thesis provides four main contributions from the three stages of the research clearly shown in Figure 1.10. The first stage focuses on the long-term vehicle motion prediction. The tracking of vehicles depends on the motion model and real-time absolute observations, which is the first main contribution. The second stage introduces the observation harvesting mechanism into the vehicle tracking, and delayed observations are considered in the tracking algorithm. This is the second main contribution. Up to this stage, vehicles have been tracked individually. The third, which is the final one of the research stages in this thesis, extends the work in the first two stages by proposing a nonparametric cooperative tracking approach followed by its extension with delayed-state filtering, which become the third and forth main contributions, respectively. The cooperative tracking approaches further consider the V2V relative ranging measurements between vehicles so that better tracking accuracy is achievable. Details on the four main contributions are given as follows.

I The thesis proposes a probabilistic long-term vehicle motion prediction and tracking approach that can be applied in large areas to track vehicle positions in the use of very sparse vehicle-to-infrastructure (V2I) communication. The vehicle model has the environment properties built into acceleration, speed and timing profiles using real historical data collected. The approach enables a con-

sistent prediction of vehicle motion over a long observation outage period, and update vehicle position when it is in the communication range of an infrastructure. Without the requirement of full network coverage, it only employs a limited number of data collection points distributed around the field to provide positive and negative information that can be used to improve position estimates. In addition, the approach is capable of coping with anomalies in a vehicle's true movement by incorporating a stopping probability within the vehicle model. Experiments were conducted to validate the proposed approach. This part of work has been published as a journal paper in the IEEE Transactions on Intelligent Transportation Systems, see [150].

II The thesis then proposes a prediction and tracking approach for multiple vehicles by extending the first main contribution. This is accomplished by additionally introducing the concept of observation harvesting, an effective and robust data dissemination process in which peer-to-peer (P2P) communication between vehicles allows egocentric position updates to be forwarded to the base station for an improved position estimate, again, without the requirement of an established global network. Every vehicle keeps the latest information for as many vehicles as possible in its local observation pool, so that the temporary failure of some vehicles, or even the majority of vehicles in the network, will not result in the failure of the entire network. This is an especially attractive characteristic for applications in which safety is primary. To fuse delayed information collected during vehicles' interactions, a particle filter capable of dealing with delayed observations is also proposed. The experimental results have clearly shown the improvements in position estimate when observation harvesting is enabled compared with individual tracking results. This part of work has been published as another journal paper in the IEEE Transactions on Intelligent Transportation Systems, see [151].

III The thesis also proposes a Markov chain Monte Carlo (MCMC) based approach, namely Gibbs sampler based cooperative particle filter (GSCPF), to cooperatively track positions of mobile agents. This part of the research is considered

one of the main contributions, because it provides a more generalised approach for cooperative tracking for cases in which the Gaussian assumption is not valid. Specifically, the proposed nonparametric approach involves a Gibbs sampler that replaces the importance sampling in traditional particle filters, and utilises it for sequential Bayesian inference. The approach could be implemented in a centralised or decentralised framework. In addition, some optimisations are introduced to lower the computation cost. The approach has been preliminarily tested in simulations of real-time cooperative tracking scenarios. The results revealed not only the advantage of cooperative tracking, but also the superior performance of the proposed GSCPF over parametric approaches when they are tested in non-linear and/or non-Gaussian systems. In scenarios assumed to be linear and Gaussian, the GSCPF presents a close performance to KF, which gives optimal results under this circumstance.

IV The thesis further extends the GSCPF approach to the delayed-state cooperative particle filter (DSCPF), which enables cooperative tracking of vehicles with delayed observations. The approach could work with the previous long-term motion prediction and observation harvesting approaches to cooperatively track vehicles in large environments without global communication. The DSCPF approach provides two versions, i.e., with or without backward smoothing. The former one further reduces uncertainty because the estimate of a past state could be improved by observations up to a later time. The experiment results validated the improvement of estimation with the DSCPF approach compared to the cases in which the DSCPF was not enabled. The experiment also demonstrated the benefits of the cooperative tracking approach in a situation in which a portion of the vehicles lose their location awareness, because they could be tracked with relative information through their neighbours. Meanwhile, the estimates of those with egocentric information available could also be improved.

The last two main contributions have been written as a third journal paper to be submitted for publication.

1.4 Thesis Structure

The remainder of the thesis is organised as follows.

- **Chapter 2: Tracking Multiple Agents in Large Environments** provides the necessary mathematical background information related to the tracking of multiple agents. Approaches in the literature regarding motion prediction, data dissemination and information fusion are reviewed, with their pros and cons discussed when used for tracking in large environments.
- **Chapter 3: From Information to Observations** introduces observation types used throughout this thesis.
- **Chapter 4: Probabilistic Long-Term Vehicle Motion Prediction** proposes a probabilistic long-term vehicle motion prediction approach, in which every vehicle is predicted and tracked individually. Experiment results are presented to validate the approach. This chapter presents the first main contribution.
- **Chapter 5: Using Delayed Observations for Vehicle Tracking** extends the work in the previous chapter by introducing the observation harvesting mechanism, making use of vehicle interactions to bring the delayed observations into vehicle tracking. The improvement in tracking performance is validated by the experiment results. Though vehicles communicate with each other, the inter-vehicle measurements have not yet been considered up to the end of this chapter. The chapter presents the second main contribution.
- **Chapter 6: Nonparametric Cooperative Tracking** proposes a novel nonparametric cooperative tracking approach that takes into account not only absolute observations from vehicles and the infrastructure, but also the relative ranging measurements between vehicles into the filtering process to achieve higher tracking accuracy. The superior performance of the novel approach is

demonstrated in simulations. The third main contribution is presented in this chapter.

- **Chapter 7: Cooperative Tracking with Delayed Observations** covers the last main contribution. The work in this chapter extends the cooperative tracking approach proposed in the previous chapter with the delayed-state concept. Combined with the long-term vehicle motion prediction and the observation harvesting, the tracking approach is able to cooperatively track vehicles with time delayed observations.

- **Chapter 8: Conclusions** presents conclusions and future work for the extension of this research.

Chapter 2

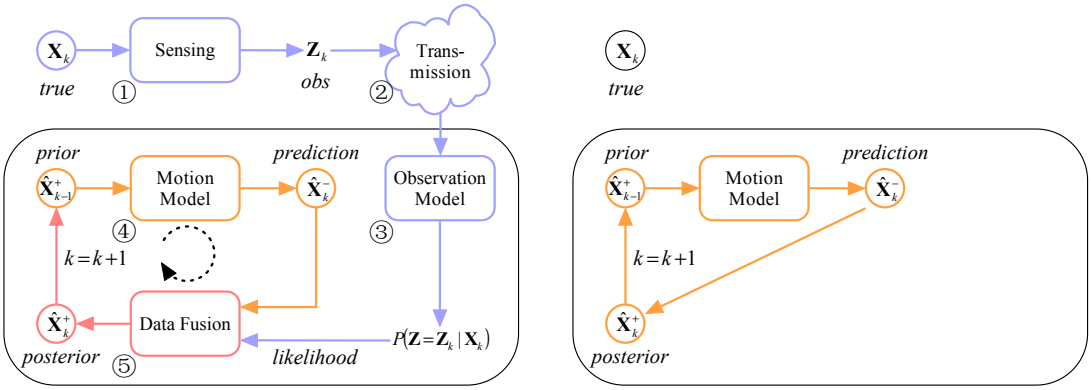
Tracking Multiple Agents in Large Environments

2.1 Introduction

This chapter examines the fundamental concepts about tracking multiple agents in large environments. It enables us to build up a concrete and comprehensive understanding of the challenges and techniques related to resource tracking over large scale areas from fundamental perspectives, prior to the original research work presented in subsequent chapters. After a brief introduction to the background of a standard tracking system in Section 2.2, the observation outage problem in tracking mobile agents in large environments is mathematically formulated in Section 2.3. A preliminary investigation on reducing the observation outage duration is elaborated in Appendix A, mostly from a theoretical point of view. This chapter also presents comprehensive reviews on mainly three topics: motion prediction approaches in Section 2.4, data dissemination methods in Section 2.5, and lastly information fusion in Section 2.6.

2.2 Background of Mobile Target Tracking

A typical tracking system consists of the following components, as also illustrated in Figure 2.1(a).



(a) A typical tracking system (b) Tracking in observation outage

Figure 2.1 – Components in a typical tracking system.

① **Sensing:** There are various existing sensors that can measure the state of people or vehicles being tracked, such as position, velocity, heading, relative range/bearing, etc. Existing sensing techniques include GPS based, vision based, radar based, ultrasonic based, laser based, received signal strength (RSS) based, ultra-wide band (UWB) based, radio frequency identification (RFID) based, magnetic field based and so on. As the sensing process contains noise, the observation made could be considered a random realisation of the true state of the target at time k from the probabilistic point of view.

$$P\left(\mathbf{Z}_k|\mathbf{X}_k\right)$$

② **Transmission:** The raw or pre-processed measurements are then transmitted to a fusion centre via a type of communication medium. The information transmission could be realised by a networking technique, such as wireless sensor network (WSN), mobile ad-hoc network (MANET), vehicular ad-hoc network

(VANET), delay/disruption tolerant network (DTN), cellular network to name a few.

③ **Sensor model:** Upon the arrival of measurements generated by sensors, the fusion centre interprets the sensed information using sensor models built beforehand for each type of sensors. The joint observation model is mathematically written as the product of N_s independent sensor models.

$$P\left(\mathbf{Z}|\mathbf{X}_k\right) = \prod_{i=1}^{N_s} P\left(\mathbf{z}^i|\mathbf{X}_k\right)$$

Lastly the information is represented in the form of likelihood functions, which could be used in the following data fusion stage. Given N_s independent observations, the likelihood functions are:

$$P\left(\mathbf{Z} = \mathbf{Z}_k|\mathbf{X}_k\right) = \prod_{i=1}^{N_s} P\left(\mathbf{z}^i = \mathbf{z}_k^i|\mathbf{X}_k\right)$$

④ **Motion model:** The state of the target at the next time step is predicted at the fusion centre using a motion model which well describes the dynamics of the target. The model could be linear, for example a simple position-velocity-acceleration (PVA) kinematic model, or a non-linear one. The latter includes neural network (NN), hidden Markov model (HMM), pattern recognition and more. Generally, the motion prediction process is written as:

$$P\left(\mathbf{X}_k|\mathbf{Z}_{1:k-1}\right) = \int P\left(\mathbf{X}_k|\mathbf{X}_{k-1}\right) P\left(\mathbf{X}_{k-1}|\mathbf{Z}_{1:k-1}\right) d\mathbf{X}_{k-1}$$

where $P\left(\mathbf{X}_{k-1}|\mathbf{Z}_{1:k-1}\right)$ is the estimate of the previous target state, a.k.a, the prior state. $P\left(\mathbf{X}_k|\mathbf{X}_{k-1}\right)$ is the motion model, which is to predict the state of the target given the knowledge of the previous state.

⑤ **Data fusion:** The likelihood information and prediction are fused in the data fusion stage, which is generally categorised to parametric and nonparametric

approaches. At the end of this stage, an estimate of the target's state is obtained. Using the Bayes rule, we have the posterior state computed as:

$$P\left(\mathbf{X}_k|\mathbf{Z}_{1:k}\right) \propto P\left(\mathbf{Z}=\mathbf{Z}_k|\mathbf{X}_k\right) P\left(\mathbf{X}_k|\mathbf{Z}_{1:k-1}\right)$$

2.3 The Observation Outage Problem

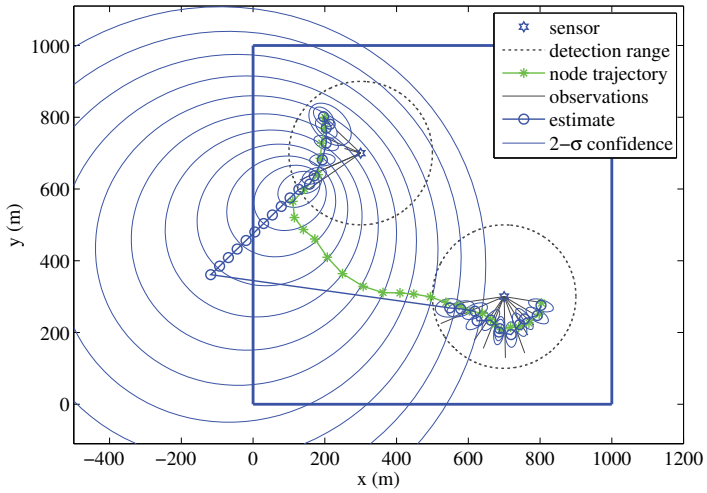
The transmission of the sensed data is not a concern when the target is doing self-tracking. In this case the target itself is where the data fusion takes place. However, in some applications where a central base station is attempting to track distant target(s), as what happens in large environments, it brings difficulties in transmitting data reliably and instantly over a long distance. The base station is likely to experience observation outages, and the tracking system, as a result, only has to depend on predictions, as shown in Figure 2.1(b). This is fundamentally because without the availability of likelihoods, the posterior comes from the prediction directly.

$$P\left(\mathbf{X}_k|\mathbf{Z}_{1:k}\right) \leftarrow P\left(\mathbf{X}_k|\mathbf{Z}_{1:k-1}\right)$$

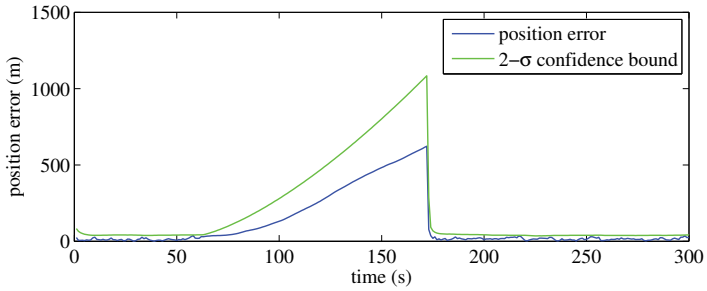
Suppose the motion prediction is required over m seconds, the resultant predicted state is written as:

$$P\left(\mathbf{X}_k|\mathbf{Z}_{1:k-m}\right) = \int \prod_{t=k-m+1}^{k} P\left(\mathbf{X}_t|\mathbf{X}_{t-1}\right) P\left(\mathbf{X}_{k-1}|\mathbf{Z}_{1:k-m}\right) d\mathbf{X}_{k-m:k-1}$$

The motion models used in conventional tracking problems are not required to perform a long-term motion prediction. Instead, they predict target movements only a few seconds into the future. Generally, the uncertainty of the target position grows quickly in the lack of position observations. An observation outage example in Figure 2.2 illustrates this situation using EKF with a linear motion model and range-bearing



(a) Tracking with range-bearing



(b) Position errors and uncertainty

Figure 2.2 – Tracking object with outage in observation. The target is accurately tracked when it is under detection of a sensor. Once it moves out of the detection circle from time 64s, uncertainty grows substantially along with time in the absence observations, until it is recaptured by the other sensor at time 173 s. Quantitatively (b) shows the growth of error and uncertainty during the near-two-minutes observation outage period (from time 64 s to 172 s).

measurements. The outage duration is nearly two minutes in the example. Besides, the update stage in most of the existing systems does not have to consider observations received with a large time delay.

The observation outage problem could be eased by adding more sensors and/or targets, increasing the sensor detection range, enabling the pair-wise collaboration between nodes and so forth. Interested readers could refer to Appendix A for a preliminary study on how this outage duration could be reduced.

2.4 Motion Prediction

2.4.1 Parametric and Nonparametric Models

The future kinematic states (e.g. positions, velocities) of an object could be predicted by motion models. They are usually presented in the form of dynamic/kinematic equations. Kalman filter techniques [92] are widely used to deal with linear systems, leaving non-linear cases to Extended Kalman filter (EKF) and Unscented Kalman filter (UKF) approaches.

The position and velocity of an object at time k are represented in the form of linear state space.

$$\mathbf{x}_k = \begin{bmatrix} x_k \\ \dot{x}_k \end{bmatrix}$$

where $\dot{x}_k$ is velocity, i.e., the derivation of position with respect to time.

They are easily calculated by using the Newton's law of motion, given the states at a previous time step.

$$x_k = x_{k-1} + \dot{x}_{k-1}\Delta t + \frac{1}{2}\ddot{x}_k\Delta t^2$$

$$\dot{x}_k = \dot{x}_{k-1} + \ddot{x}_k\Delta t$$

where $\ddot{x}_k$ is acceleration, i.e. the derivation of velocity with respect to time.

The prediction model for the Kalman filter approach can be written as follows:

$$\mathbf{x}_k = \mathbf{F}\mathbf{x}_{k-1} + \mathbf{G}a_k$$

where a_k is the acceleration, which is assumed to be a normally distributed random value with zero mean in the Kalman filter.

$\mathbf{F} = \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix}$ is the state transition function.

$\mathbf{G} = \begin{bmatrix} \frac{\Delta t^2}{2} \\ \Delta t \end{bmatrix}$.

Examples of this type of application are demonstrated in [77] [138] [95] and [134]. Nevertheless, in many situations, high non-linearity of system variables and noise existing in process and measurement make the formulation non-trivial. Due to a complex interrelationship between vehicular and environmental variables, it is not feasible to explicitly parametrise them into a vehicle kinematic model. This could be addressed with nonparametric methods such as a particle filter and its variants. However, vehicle models based on dynamic equations can be affected by changes in the vehicle conditions or operating contexts. Probabilistic approaches have been widely used for motion prediction making use of motion pattern techniques. These approaches are based on the assumption that the vehicle motion has typical patterns which can be learnt, and used to predict future states of the vehicle.

Generally speaking, the process of probabilistic motion prediction could be decomposed into two parts.

1. Learning: construct a set of representations describing motion behaviour of an object based on observations about it.

2. Prediction: estimate the future motion of the object on the basis of learnt knowledge about the present and past states of the object.

An off-line learning approach uses a "learn-then-predict" scheme, meaning that the system does not switch to the prediction stage until the learning process is accomplished. The method has to make the assumption that the model is well generalised by including enough datasets in the learning stage. This limitation is not found in on-line approaches, where a "learn-and-predict" concept is used. In this kind of system, an extra stage is introduced to improve the model in an incremental fashion.

3. Refining: continuously refine the learnt motion model by incorporating new observations.

To date, literature on vehicle tracking and motion prediction based on historical patterns mainly relies on an efficient pattern learning and recognition technique. More specifically, the strongly related research areas are trajectory matching and trajectory classifiers. As defined in [165], a trajectory is a sequence of successive locations in a 2D or 3D Euclidean space that is recorded along with its corresponding timestamps. That means the historical trajectories provide essential information to derive the future states of the object. [100] predicts target's future behaviour in the use of a clustering algorithm, which learns typical motion patterns of the target from trajectory data in a given environment. In [79], a visual tracker to predict behaviour of vehicles using learnt motion patterns from hierarchically clustered trajectories is applied. An online technique named Echo State Networks in [71] is adopted to predict movements of persons merely relying on historical trajectory data.

Different representations of trajectories have been studied in literature. As an early proposed method, cubic spline interpolation [42] is known to be sensitive to initial conditions and require re-computation for different operating conditions. Other examples include cubic Bézier curves [83], non-linear attractor dynamics [87], polynomial descriptors [93], linear dynamical system (LDS) [46] and trajectory shape signatures [43]. Given a sequence of location measurements, the observed trajectory is then compared to learnt knowledge in the use of a trajectory matching technique. Examples are dynamic time warping [63], longest common subsequence (LCSS) [166] and Levenshtein distance on trajectories (LTD) [66] to name a few. The LCSS is further extended in [73], which introduces the quaternion-based rotationally invariant longest common subsequence (QRLCS) metric. [43] proposes a trajectory matching method based on a pose normalisation process, which is invariant to translation, rotation and scale in 3D space. Apart from that, 3D trajectory matching is also adopted in [66] for long-term human motion predictions.

Hidden Markov model (HMM) based approaches have also attracted significant interest in recent years. Fraile and Maybank, [57], use HMM for classification of vehicle

motion given short segments of trajectories together with estimated speeds and steering angles in a known 2D area. Another example implementing HMM is given in [24] for people tracking applications. An extension of HMM is given in [163] that describes an approach named growing HMM (GHMM) that incrementally learns motion patterns on-line to predict vehicle motion. More works related to learning-driven approaches are surveyed in [182].

One of the interesting motion prediction examples is PRIDE, [109], which utilises an hybrid framework combining EKF and situation recognition for short-term and long-term predictions respectively. A different approach utilizing Neural Networks (NNs) is presented in [78] and [125]. In these cases the NNs were trained with actual vehicle measurements to predict the next stage of the vehicle given steering angle and velocity inputs.

2.4.2 Standard and Long-Term Prediction

In a general form of prediction formulations, we assume that a motion model is able to statistically predict the state of an object m seconds into the future, given the estimate at present $P(\mathbf{X}_k|\mathbf{Z}_{1:k})$.

$$P(\mathbf{X}_{k+m}|\mathbf{Z}_{1:k}) = \prod_{t=k+1}^{k+m} P(\mathbf{X}_t|\mathbf{X}_{t-1})\, P(\mathbf{X}_k|\mathbf{Z}_{1:k}) \tag{2.1}$$

Usually $m = 1$ is seen in a standard Kalman filtering formulation. Most of the existing approaches are suitable for very a short-term prediction and are only capable of predicting motion for no more than a few seconds. These approaches become less effective when they are used for prediction over a longer time. In [73] [109] and [72], long-term is defined to be on the order of only several seconds, which is far from the requirement of several minutes prediction (i.e. $m \sim 10^2 s$) dealt with in this thesis. The long-term motion prediction defined in this work has not yet been addressed by many researchers. The main challenge comes from the high diversity of road shapes and driver behaviour over time. The problem is further aggravated when turning

at a(n) crossing/intersection is taken into consideration. To simplify, it is therefore assumed in [52] and [108] that the mobile object will travel on the shortest/quickest paths.

In many circumstances, additional information is desirable for predictions in a long-term manner. Tracking of mobile users with a low-sampling-rate (one sample per 2 minutes) of GPS trajectories is achieved in [108], relying on a map-matching approach named ST-Matching. The map data is as well adopted as *a priori* knowledge in our approach, which is elaborated in the next section. In some approaches, high rate observations of internal states, such as steering angles and velocities are needed to assist predictions, see [125] [57] [163]. This is defined as dead reckoning in the literature, refer to Figure 2.3 as an example. However, this kind of information is not available in the tracking problem concerned in this thesis. Since the prediction is carried out at a central location, usually for a long period of time it lacks any observations about the vehicles to track.

Figure 2.3 – Dead reckoning in a GPS outage. During the outage period of GPS signal (red dots in the figure), the vehicle is still capable of keeping self-localised using the on-board sensors (e.g. compass, accelerometer, speedometer, gyrometer) and route information on the map.

2.4.3 Incorporating Environment Properties

Continuing the discussion in the previous subsection, the environment information helps physically constrain the potential trajectory of the mobile object. In other

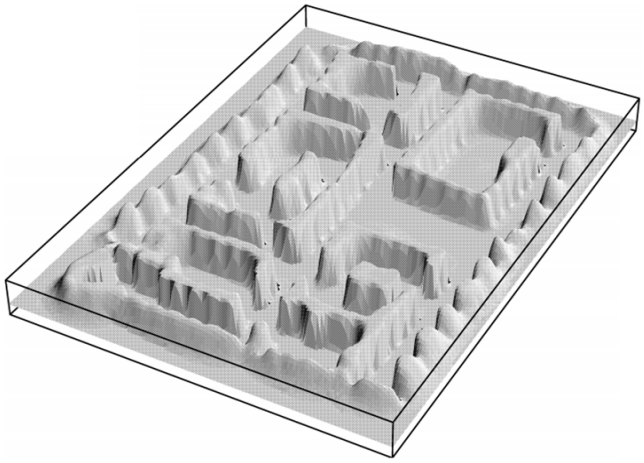


Figure 2.4 – Probabilistic map [159]. The height of each point corresponds to the probability that it is occupied.

words, the kinematic model is improved with the introduction of context awareness, with which each mobile object is considered as a part residing in a comprehensive system rather than as an individual. The first factor to include is the road map data, while other environmental information includes weather condition, traffic flow, interaction with others and more. By matching the observed trajectory to a route in the road network, a fast off-line map-matching approach is proposed in [52]. Another map-matching based approach is presented in [31], to compare GPS-observed vehicle trajectory to candidate paths in the digital map. The road network information could also be incorporated in the form of enhanced map (Emap) models [26] [161] to provide lane-level positioning for vehicles. With other environmental information (e.g. layout and geometry of surrounding buildings) additionally considered, 3D environment models are built up and used in [22] and [124] to detect multipath effects of GPS signal in urban areas, thereby improving the positioning accuracy.

In the statistical context, the geographical map information can be incorporated as a probabilistic map [159] [55] [158] [160], see Figure 2.4 for an example. It makes the assumption that a vehicle tends to move on roads, and its position in 2D space is bounded by the road network. Furthermore, the study in [108] incorporates temporal/speed constraints of trajectories in addition to the geometric information of a road network. This is taking into account further physical properties of environment

when modelling vehicle's movements. The most related work to the motion model proposed in this thesis is [175] and [174], in which road properties such as gradient and curvature are built into the velocity model for a vehicle on a particular road. A histogram of vehicle speed measurements is constructed for each section of the road, which is then used to predict the speed of a vehicle as a function of the location in the environment. The base station in [174] evaluates a long-term prediction for the fleet of vehicles. Nonparametric filters and delayed observations relayed through multi-hops in a mesh network are then fused to update estimates. Both papers utilised speed profiles, assuming that vehicles are moving all the time, and no differentiation is made between haul roads or intersections, making prediction sometimes not consistent due to un-expected vehicle behaviour.

The proposed model in the thesis, taking a step further, considers stopping cases and properties of the environment, such as roads, intersections, and special areas. The model has the properties built into acceleration, speed, and timing profiles. It is based on the fact that a vehicle has a certain probability to stop anywhere and any time due to operator requirements, machinery malfunctions or an accident. After a stopping event, the time taken before resuming is also included as a probabilistic value. The incorporation of these improvements has proven essential to be able to provide consistent predictions as will be shown in the experimental results section.

2.5 Data Dissemination

This section presents an overview of the current research issues on wireless networks, through which data of interest is able to disseminate out.

2.5.1 Infrastructure and Non-Infrastructure Networks

Networks could be classified according to topologies they employ. Widely used topologies for wireless networks are star, cellular and ad-hoc. Others such as ring and tree, usually used for wired networks, are not discussed in this section. Both star and

cellular are considered typical infrastructure networks, where nodes do not communicate with each other directly, but via a service provider, a.k.a "infrastructure". Every node in this kind of network connects to a central infrastructure, and all information flows from and to it. The central point plays the most significant role in the network. It acts as a master establishing and maintaining the whole network, while all other nodes are slaves. Bluetooth (IEEE 802.15) is a typical star network in which a master hub can connect to a maximum of 7 slaves. Cellular networks physically provide multiple base stations, which are connected from a higher level by either a wired or wireless way.

Despite easy implementation, infrastructure networks have limitations in terms of network scalability and robustness. The scale of infrastructure networks is limited to the signal coverage of the central hub(s). The robustness also becomes an issue as the whole network fails once the infrastructure malfunctions. Furthermore, the bandwidth and processing capability of the central point(s) would become a bottleneck with increasing quantities of nodes, when communicating and processing a large amount of data are required. Although cellular networks have comparatively better scalability and robustness owing to multiple base stations, they per se do not differentiate much from the star networks. As investigated in Appendix A, a fair large number of collection points, or long-range communication is needed when an infrastructure network is employed in an application with a large operation area. This is not feasible in practice due to concerns such as physical deployment, power, cost, etc.

These limitations could be overcome with a non-infrastructure ad-hoc network, in which no central node is required and nodes are their own service providers. An ad-hoc network is a collection of wireless mobile nodes dynamically forming a temporary network without the use of any existing network infrastructure or centralised administration [144]. Nodes are free to move randomly to form arbitrary network despite node positions, and capable of communicating with each other in an ad-hoc network. Therefore it is known to have the ability of self-organisation, self-discovering, self-healing and self-configuration. The basic communication in an ad-hoc network is forwarding. When a node receives a packet, it will forward the data to neighbours if

it is not the intended end-recipient. Multi-hop and P2P communication are of significance when a node is going to send data to those which are unable to communicate with directly. By removing the installation of the infrastructure, the ad-hoc network brings advantages such as a low setup cost, easy setup and excellent scalability. The ad-hoc network has been developed into many applications. For example, the well-known IEEE 802.11 or WiFi protocol also supports an ad-hoc configuration in the absence of a wireless access point.

In summary, the comparison of characteristics of star/cellular and ad-hoc networks is given in Table 2.1.

Table 2.1 – Comparison of star/cellular and ad-hoc networks

Topology	Star and Cellular	Ad-Hoc
Basic Type	Infrastructure Network	Non-Infrastructure Network
Central Point Requirement	Fixed and Pre-Located Central Point	No Central Point
Network Structure	Fixed	Dynamic Multi-Hop
Scalability	Poor, Limited by Central Node	Excellent
Mobility	Poor	Good
Setup Cost	High	Cost-Effective
Setup Speed	Low	High
Design Complexity	Low	High

2.5.2 Vehicular Ad-Hoc Networks

Generally, there are five primary categories of wireless mobile ad-hoc networks (MANETs) widely used in the literature. They are listed as below.

1. Pure Mobile Ad-Hoc Networks (Pure MANETs)
2. Wireless Mesh Networks (WMNs)
3. Wireless Sensor Networks (WSNs)

4. Vehicular Ad-Hoc Networks (VANETs)

5. Delay/Disruption Tolerant Networks (DTNs)

Among these, VANETs and DTNs are more related to the vehicle tracking problem this thesis is tackling. Respectively the two types of networks are reviewed in this section and Section 2.5.4, while the topic on the rest three MANETs would be expanded in Appendix B.

The development of vehicular ad-hoc networks (VANETs), a special kind of MANETs, has been stimulated mainly for applications in the area of intelligent transportation systems (ITS) and considered an important component therein [10]. In a VANET, those vehicles equipped with communication means are able to exchange information with each other through V2V communication, and also with an roadside infrastructure via V2I communication (also defined as vehicle-to-roadside, i.e., V2R in some literature). This kind of network is particularly attractive to autonomous and safety related applications, as the knowledge of real-time vehicle positions is assumed to be known by every node over the network.

Plenty of related work in this field focuses on communication and routing issues in VANETs. [54] studies the effect of communication parameters (e.g., transmission rates, ranges) of the data dissemination performance of a VANET. In terms of routing algorithms, topology-based and geographic routing protocols are two primary categories in VANETs. The choice depends on the requirements of applications, balancing pros and cons of the two. This section is not going to look into details on these protocols, interested readers could refer to [168] and [104] for reviews of routing protocols in VANETs.

The VANETs are different with pure MANETs basically in three aspects. (1) Power capacity of mobile nodes: the energy limitation no longer becomes a serious issue in a VANET as huge batteries are carried by vehicles. (2) Mobility of nodes: vehicles usually have higher mobility than nodes in traditional MANETs. (3) Motion pattern: vehicles are likely to move in a relatively more organised and predicable pattern rather than a purely random manner in a MANET.

In addition, there are two obvious concerns among many challenges existing in VANETs.

1. Location-Awareness [29]: the localisation requirement is essential in most of VANETs applications, please refer to Section 2.6.1 for an elaboration on this topic.

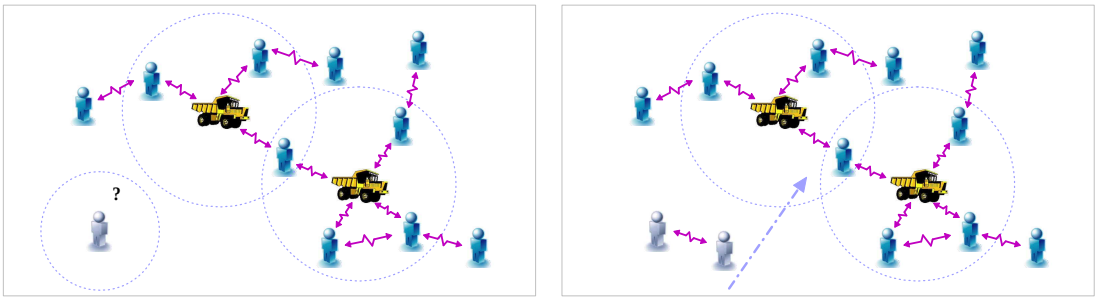
2. Disconnected Network [133]: the high mobility of vehicles tends to break connections in the network, which leads to an "isolated agent issue" defined in this thesis. Please refer to Section 2.5.3 for a discussion.

The IEEE P1609 standards family for Wireless Access in Vehicular Environments (WAVE) [1] is under further development and full use standards of IEEE 1609.2, 1609.3, and 1609.4 were released in April 2013 [84]. It consists of definitions of the architecture, communication model, networking service, MAC layer, etc. in vehicular networks. The ASTM E2213-03(2010) standard [13] describes the specifications of physical layer and MAC of wireless connectivity using 5 GHz band dedicated short-range communications (DSRC) services, which is based on IEEE Standards 802.11, for information exchange between roadside and vehicle systems.

2.5.3 Isolated Agent Issue

In a traditional network, it makes an assumption that real-time information of a node could be immediately transmitted to others using multi-hopping communication through the mesh network [174]. However, an isolated agent issue occurs when the connection gets interrupted to an agent without any neighbours around, illustrated in Figure 2.5(a).

More specifically for a vehicular case, the issue arises when VANETs are implemented in scenarios where the network is sparsely connected. A typical example case is rural areas, where a data packet is likely to be lost due to a low density of vehicles. The issue also occurs due to the lack of presence of neighbours within a vehicle's communication range to relay the message when the VANET is used in a large environment. One of



(a) The isolated agent issue in cooperative localisation

(b) Delayed information brought back from the isolated agent

Figure 2.5 – Cooperative localisation and the isolated agent issue. a) demonstrates a typical cooperative localisation application where every agent (but one) is connected with others nearby to form a mesh network. An isolated agent issue occurs when an agent does not have any neighbour around and loses connection to the rest. In b), another mobile agent passes and brings the information of the isolated agent, though delayed, back to the main group.

the solutions to this issue is by introducing another mobile agent to close pass the isolated agent and bring its information to the rest, though with some time delay. It requires a networking means allowing time delays in data transmission as well as a filtering method capable of fusing delayed information received.

The work in [172] studies the network disconnection phenomenon based on empirical data, and concludes that typical conventional routing protocols are unable to deal with the issue. To overcome the frequent disconnections of VANETs under circumstances discussed above, DTN based routing strategies under the category of geographic routing protocols have attracted the interest of researchers in recent years. These approaches allow packets to be stored when neighbouring vehicles are not in sight, and they are delivered when the carrier node meets with others. The following section presents details on the topic of DTNs.

2.5.4 Delay/Disruption Tolerant Networks

Delay/disruption tolerant networks (DTNs) are appropriate candidates for applications where continuous network connectivity is unable to be achieved using traditional

networks. As an overlay on top of regional networks including Internet, cellular networks and WSNs, a DTN provides content storage as a core network service across applications to deal with intermittent communication. A store-and-forward (or carry-and-forward) mechanism is used for routing asynchronous messages with various network transport technologies. Thus messages flowing within DTNs are at content level, rather than at packet level and focus on information dissemination rather than on node delivery [180].

DTNs were firstly introduced in deep space communication in the late 1990s, dealing with large round-trip propagation delays between spacecraft and ground base station. As part of NASA's efforts in extending Internet into outer space, a "Space Internet" system is being developed and tested in International Space Station (ISS) through implementation of the DTN-on-ISS [89]. At present the application of the new communication paradigm has been extended to terrestrial environments, where the delays could also be caused by communication disruptions and the data transmission is more opportunistic. This was initiated by the fact that, network disconnections inevitably occur due to variations in communication environments. [25] and [111] show examples of applications of DTNs in complicated underwater scenarios, where the acoustic communication channel suffers from frequent connection disruptions.

A key difference between DTNs and traditional networks is that a traditional one requires a reliable end-to-end route to be established and maintained until the communication is completed. Sufficiently good connectivity thus must be guaranteed in traditional networks. The communication between nodes is prone to frequent fragmentation otherwise. This is not a requirement for DTNs which use store-and-forward type overlay functions for dealing with disconnected operations and aiming at opportunistic transmission [180]. The nodes in DTNs buffer data until connections are available. This is shown in Haggle Project [156], where agents exchange messages with nearby devices. Each agent carries a message until it is close to another device, constructing a global mesh network. The vehicle-assisted data delivery (VADD) presented in [184] adopts the DTN concept and aims at reducing delivery delay in a disconnected vehicular network by making use of the predicable vehicle movement

constrained by traffic pattern and road network. [101] proposes GeOpps, a delay tolerant geographical opportunistic routing approach, to efficiently select the message forwarding path given the knowledge of vehicle routes suggested by on-board navigation systems. Readers may refer to [167] for a review on applications of DTNs.

Nowadays, the DTN Research Group (DTNRG) formed in 2002 is concerned with study on the architectural and protocol design of DTNs. DTN standards include RFC 4838 for architecture and RFC 5050 for the bundle protocol specification. Information on an up-to-date list of RFC standards is available on-line [85].

2.5.5 Flat and Hierarchical Architectures

Networks can be classified to flat and hierarchical depending on the hierarchy property. An infrastructure network is a typical example of physically hierarchical networks. The central node(s) take care of all communication and controls, so that ordinary nodes do not have to handle issues introduced in an ad-hoc network, such as network establishing, maintenance and control. However, as mentioned in Section 2.5.1, the central node(s) in a physically hierarchical network are constrained with limited communication bandwidth and processing capacity. While in a flat network, every node is not only physically equal in its capabilities with respect to power, performance, communication bandwidth, etc., but also logically plays the same role in the network. Nodes in a flat network connect to each other directly without forming clusters. Hence the management and coordination via central node(s) (cluster-head(s)) do not exist in a flat network. Networking follows a self-organised fashion and data is processed in a decentralised form. Because of this, scalability and communication bandwidth are not concerns in a flat network. However, the drawback of flat networks is that, the networking is of high complexity in the absence of coordination of central node(s). Apart from that, every node is also required to deal with network organising, routing and maintaining issues which substantially increase the system complexity and energy consumption.

MANETs are usually flat networks. Nevertheless when the scale is large in WSNs

or VANETs, nodes may form clusters according to deployment and some nodes are selected as cluster-heads acting as virtual central nodes [135]. This type of network is known as a clustered network. Multiple nodes within a local region connect to a local cluster-head to form a first-tier network. In the meanwhile all cluster-heads may connect together to construct a second-tier network. Therefore in other words, a clustered network is logically hierarchical. Nodes could use either single-hop or multi-hop mode of communication to send their data to their respective cluster-heads [115].

In some MANETs, the cluster-heads are static, which means that once selected, the cluster-heads would not change. The selection of the cluster-heads could be predetermined or following some algorithms. In [105], a multi-cluster, multi-hop packet radio network architecture for a wireless adaptive mobile information system is presented. It considers and compares two distributed clustering algorithms. The first is the lowest-ID algorithm, where the lowest-ID node is elected as the cluster-head in a region. The other is the highest-connectivity (degree) algorithm, in which the node with the highest degree in a neighbourhood becomes the head. A novel clustering algorithm named Distributed Score Based Clustering Algorithm (DSBCA) for MANETs is proposed in [2]. In DSBCA each node calculates its score by a linear algorithm taking into account the battery remaining, number of neighbours and stability. Then each node independently chooses one with the highest score as the cluster-head from its neighbours.

2.5.6 Consideration of Resource Constraints

The selected static cluster-heads in a clustered network are not competent as they are physically identical to other nodes because of its physically flat hierarchy of MANETs. As the same issues existing in a hierarchical network, the bandwidth and processing capability of cluster-heads would become a bottleneck along with the increase of node quantity [105]. The energy consumption of the cluster-heads is also higher than ordinary nodes. In order to solve these problems of resource constraints, a dynamic

selection of cluster-heads are adopted in the literature. They are dynamically elected according to a kind of randomised rotation rule. [68] proposes the virtual base station (VBS), with which a mobile node is elected from a set of nominees within its area. [19] suggests a distributed, randomised clustering algorithm to organize the sensors into clusters in a WSN for the purpose of energy-efficiency. Each sensor in the network volunteers to become a cluster-head with a probability of p and advertises itself to sensor nodes within its radio range. Other examples could be found in [70] [82]. Please note that in a WMN the limitation of energy capacity of cluster-heads (i.e. MRs) does not exist as a concern as it does in other types of MANETs, as mentioned before, thanks to its pre-configured hierarchical structure.

With the improvement in energy dissipation of cluster-nodes, the dynamic cluster-head technique, however, still could not properly sort out the inadequacies of communication bandwidth and processing capability in MANETs. Combining the advantages of flat and hierarchical networks, a hybrid scheme is preferred with the capability (e.g. power, computation performance, communication bandwidth) of cluster-heads increased. Under this circumstance, the cluster-heads are pre-determined and kept static rather than dynamic. The network is both flat and physically hierarchical, as nodes are flat and capable of communicating with each other while cluster-heads are equipped with more computational and communication resources. This kind of hybrid network is adopted in the thesis in the form of cooperative localisation, in which mobile nodes could communicate with both other mobile nodes and stationary data collection points (reference nodes in the localisation context, and static cluster-heads with respect to the network hierarchy). For details on cooperative localisation, please refer to Section 2.6.2.

Although infrastructures such as data collection points and a base station are used in this thesis, the network is considered ad-hoc or non-infrastructure, because the infrastructures are only to collect information and perform estimation. They are not involved in establishing and coordinating the network, which is whereas maintained by every node. Data is shared directly amongst vehicles in a P2P manner and ultimately gathered by the infrastructures.

2.6 Information Fusion

2.6.1 Improving Location-Awareness

Localising and tracking mobile nodes in an industrial environment is necessary for many autonomous and manned system applications. GPS receiver has been ubiquitously introduced as one of vehicle's on-board equipments, and various forms of GPS-based tracking algorithms are widely used for the localisation/tracking purpose. Unfortunately in a practical scenario, not all mobile nodes will have continual access to GPS position information as GPS-based systems might have difficulties in providing an accurate position, due to GPS degradation or even outages in some certain situations. The robustness of GPS-only systems therefore arises to be a critical issue in tracking applications.

An alternate approach is to use a number of beacons located at known positions to estimate the agent location. Using this approach, the mobile agent is fitted with a range/bearing sensor capable of detecting and identifying the beacons. Trilateration/triangulation algorithms, often involving filters, such as the Kalman filter, are used to estimate the position. This technique requires the working environment to be covered by a sufficient number of beacons to ensure that sufficient position accuracy can be achieved. To eliminate the requirement of the infrastructure, a few of other localisation techniques have been adopted in literature, by combining them with GPS into a joint solution which improves robustness and localisation accuracy. These supplementary localisation approaches include dead reckoning [56], map matching [52] [158], vision based [155], radio frequency identification (RFID) based [75] [14], laser based [120], radar based [141], [142] to name a few.

The localisation problem can also be addressed with cooperative localisation (CL). The advent of efficient P2P communication technology has made the cooperative tracking an active area of research.

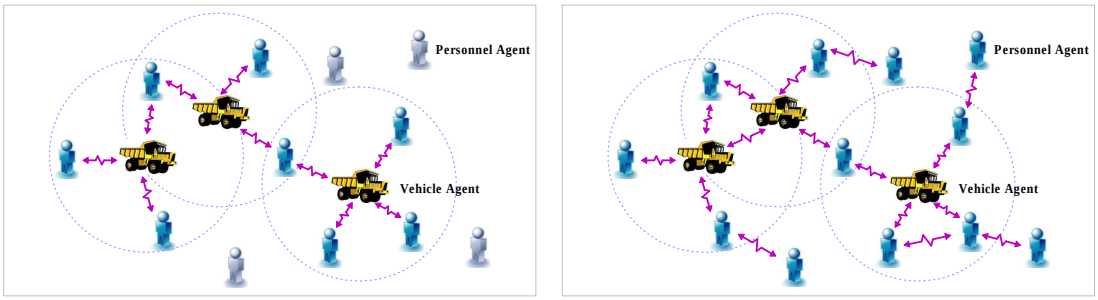
2.6.2 Non-Cooperative and Cooperative Localisation

Cooperative localisation has attracted a growing interest in the research community. In a non-cooperative centralised localisation scenario, interaction is only found between agents and infrastructures. In a large environment, the infrastructure usually fails to cover the whole area, which makes the global localisation ineffective. With cooperative localisation, the communication between vehicles can reduce the need for all vehicles to be constantly within the range of centralised infrastructure. [177] applies a cooperative approach in indoor localisation with benefits demonstrated from the cooperation between agents. A similar work is presented in [176], where a decentralised cooperative approach is adopted aiming for self-tracking in large mobile wireless networks. Interested readers may find more examples of cooperative localisation in [12], [15], [18], [131], [35], [110], and [3]. A comprehensive review of the current work is presented in [177].

Measurements of relative range and bearing can be used to estimate the relative position of all mobile nodes. This approach can have significant advantages since it can reduce or eliminate the requirement for fixed beacons, and the requirement for absolute egocentric position information (i.e., GPS) from all mobile nodes. Furthermore, cooperative localisation approaches also outperform non-cooperative ones in terms of localisation accuracy. The improvement is quantitatively illustrated in [171] with the concept of information ellipse from the geometric point of view.

A typical cooperative tracking example can be seen in Figure 2.6. In this case the mobile nodes are moving in a given area of operation, and the absolute Euclidean positions and pairwise measurements are shared between the nodes via a type of P2P communication such as a WSN [36] [126], pure MANET [47] or VANET [54]. The information can then be used to update the position estimates of each mobile node at a central observation point, for example, a base station.

2.6.3 Centralised and Decentralised Approaches



(a) Non-cooperative (b) Cooperative

Figure 2.6 – Non-cooperative and cooperative tracking in a mobile ad-hoc network. (a) Nodes without self-localisation capability (personnel agents in the figure) have to be localised by a sufficient number of reference nodes (vehicle agents) in non-cooperative tracking. (b) demonstrates a typical cooperative tracking application where every mobile node (vehicle or personnel agent) is connected with others nearby to form a mesh network. These position unknown nodes can communicate and help localise each other in cooperative tracking.

In a centralised CL architecture [36] [131] the estimation is performed at a central node. Each ordinary node in the framework broadcasts and forwards information to the central node for processing, which requires a direct or multi-hopping communication link established between the node and the central. The fusion centre is the network sink where measurements from all nodes are collected and the data fusion process occurs. High communication bandwidth and data processing capability of the central node are therefore in demand in the centralised network. The decentralised architecture, on the other hand, does not involve any central fusion facility. Each node is equipped with its own data processing capability and locally the data fusion takes place on the basis of local measurements and information from neighbouring nodes. An introduction to the fundamentals of decentralised data fusion (DDF) is presented in [51].

The centralised and decentralised CLs generally differ in the following aspects.

1. In a centralised approach, the central node is in charge of all necessary network coordination and data processing. The central node thus plays a pivotal role for the successful operation of the centralised structure. At the same time, the network is fragile at risk of the loss of the central node. The decentralised

framework substantially improves the robustness as no single node in this kind of approach acts as the fusion centre. Therefore it benefits from redundancy against the failure of the central node.

2. The bandwidth and computation capabilities become bottlenecks along with the increase of node quantity in the centralised framework, while the communication in decentralised CL is limited to P2P. The distributed data fusion paradigm and the removal of the requirement of the central node in the decentralised architecture make the network more scalable.

3. In the centralised network, global knowledge of the network topology is required by each node for the end-to-end transmission of data to the sink. The knowledge is also required to be kept up-to-date along with any change of the topology. These requirements are eliminated in the decentralised structure, where every node is concerned with communicating with its own instant neighbours.

Though distributed and decentralised approaches [126] [110] [3] have advantages such as higher network robustness and scalability, centralised approaches are attractive to applications where monitoring or surveillance is required from a "control room". A typical example is given in [174] to track a fleet of vehicles in a mining safety scenario. Furthermore, centralised algorithms usually offer more accurate position estimates in comparison to decentralised approaches [53], though at the expense of higher bandwidth, deployment and computation costs. A hybrid example is presented in [18]. Each node fuses egocentric information locally. Only inter-node measurements together with state estimates are sent to the central server for global localisation, reducing bandwidth and computational burden of the central node.

The approach in the thesis is defined to be centralised from the perspective of information processing. Measurements are shared among mobile vehicles and forwarded to the base station ultimately for the purpose of cooperative state estimation. In traditional centralised frameworks, data transmission follows an end-to-end route linking the transmitter and the determined recipient(s) (usually the central node), which

is known as unicast or multicast in terms of routing scheme. Instead, the information dissemination in the approach does not have an intentional recipient and thus is broadcast. A vehicle only shares its knowledge with neighbours in the use of P2P communication, and as time goes by the data is known by the rest of nodes in the network, including the fusion centre. This is much similar to the communication in decentralised approaches and brings benefits by eliminating the extra system overhead spent on routing and network maintenance.

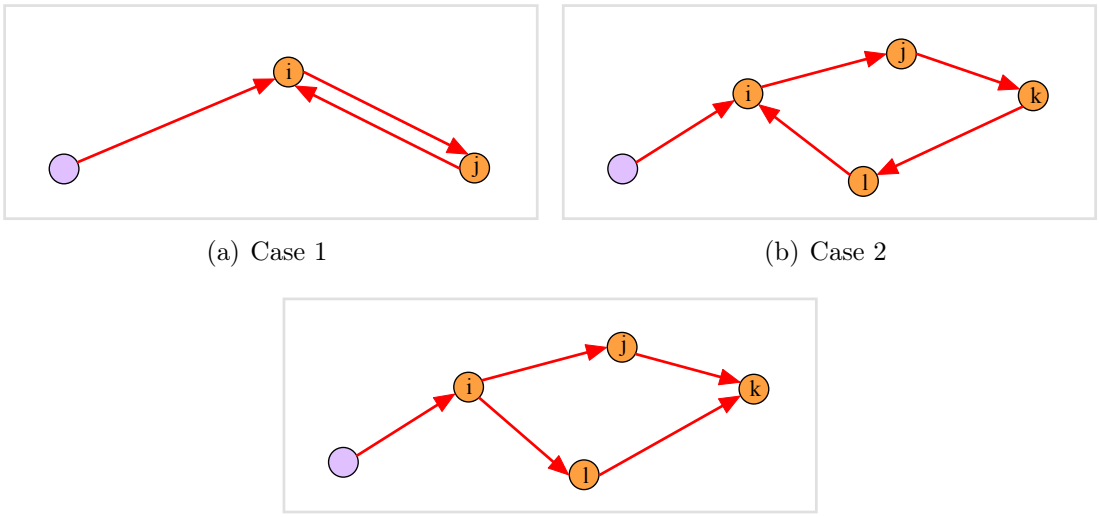
2.6.4 Inconsistency Issue

The over-confident (inconsistent) estimation issue [80] [56] (a.k.a, circular reasoning [76], or cyclic update [102] [103]) is usually present in decentralised CL problems when cross-correlation and dependency between nodes are not properly considered. This is primarily due to the double counting of information. Two nodes are said to become correlated since one of them uses the position estimate of the other to update its own. The inconsistency issue occurs when a node uses the information from another node that was previously correlated with, but without taking the inter-node dependency into consideration. It also happens when a node uses the information from multiple correlated nodes without properly considering the cross-correlation. Figure 2.7 shows a few cases leading to inconsistent estimates. In every case in the figure, a measurement is used more than once in updating a state estimate, causing an overconfident estimate, which then diverges.

The issue is addressed in [15] by keeping a record of measurements sources, while a dependency tree is introduced in [76] to trace the use of information. The work in [126] [62] [162] removes common past information by introducing channel filtering before the fusion stage. The channel filter records information that a pair of communicating nodes have in common and thus ensures that only new information is propagated between the two linking nodes. Nonetheless, it is subject to constraints in memory resource and scalability as a separate channel filter is required for each communication link [126]. Furthermore, the use of channel filters is limited to tree-connected

networks.

In centralised CL solutions, the most intuitive way to track every node cooperatively is to maintain a joint state. The cross-correlation is then handled implicitly in the filter, hence the potential inconsistency issue is no longer a concern. For this reason, [102] [103] develop a CL approach that is mathematically centralised-equivalent but realised in a decentralised fashion. However, the high-dimensional issue occurs when a joint state is manipulated in filtering. Please refer to Section 2.6.7 for details.



(a) Case 1

(b) Case 2

(c) Case 3

Figure 2.7 – Inconsistency issue caused by double counting information. (a) and (b) show cases when node i misuses the measurement of itself to update its estimate, defined as "i to i dependency problem" [132]. (c) demonstrates a case where node k double uses the information of i transmitted from j and l, which is also known as "common i problem". As the consequence of the inconsistency issue, an over-confident estimate happens when the cross-correlation is not properly considered and information is double counted.

2.6.5 Parametric and Nonparametric Approaches

Current CL approaches can be categorised into parametric and nonparametric ones. The parametric include Kalman filter (KF) and extended Kalman filter (EKF) based [47] [110] [15] [80] [102], maximum likelihood estimation (MLE) based [130], maximum

a posteriori (MAP) based [121], least-squares (LS) based [39] [127] to name a few. The nonparametric approaches are mostly particle filter (PF) based [126] [145] [3] [76] [56], such as nonparametric belief propagation (NBP), sum-product algorithm over a wireless network (SPAWN) and so forth.

In parametric filtering methods, the states to estimate are represented in the use of simple parameters, such as mean vectors and covariance matrices. Compared with nonparametric methods, the parametric approaches can produce more precise results when assumptions made are correct. On the other hand, they are comparatively more fragile against the violation of the assumptions. As examples of parametric filters, the KF and EKF are often the most attractive because of a minimal computational cost. Nevertheless, the application of KF and EKF based approaches is limited to problems presenting Gaussian properties. The information-filter based scheme in [18] is similar to the EKF based, except for a higher efficiency in communication and fusion stage due to the sparseness of canonical-form representation of Gaussian variables. There is some recent work in [6] [7] aiming to improve the robustness of parametric filters to non-Gaussian and heavy-tailed noise. Nonparametric algorithms are considered the ideal choice for the tracking of multiple objects in cluttered scenarios when the Gaussian assumption is not valid, however at the expense of computational complexity. Furthermore, it requires higher bandwidth in communication when a whole collection of particles is required to transmit from one node to another. To address this concern, in [126] a particle set in each node is converted to the form of continuous distribution before it is transmitted to another node nearby and used in DDF.

An approach receiving increasing attention in CL area is the NBP, which was initially proposed for localising in static networks [86]. Generally it is a distributed message-passing method on a graphical model. The belief of a target node, i.e. posterior marginal probability density function (PDF), is inferred by the product of its local evidence and all of the incoming messages. A message in NBP represents the "opinion" about the location of the target node from the viewpoint of a neighbouring node [146]. Then the target node broadcasts its updated belief to others in form of outgoing messages. These two steps are iterated until sufficient convergence. Variants of the

NBP have been developed to deal with the tracking of nodes in mobile networks, with the targets' dynamics models taken into consideration. One of the examples is the SPAWN [177], also known as loopy belief propagation (loopy BP), which operates on a factor graph and could be used in both static and dynamic networks. In summary the NBP based approaches own several advantages:

- It naturally allows for a distributed inference and easy implementation in ad-hoc networks.
- As a nonparametric algorithm, its applications are not limited to Gaussian models.
- It is capable of providing not only position estimates, but also the uncertainties associated.

Noteworthy is the last point, which is the main advantage of the NBP as a Bayesian approach over other non-Bayesian algorithms such as LS. The non-Bayesian estimators do not provide associated uncertainty information for all the target nodes, which is however crucial for most localisation/tracking applications. In Bayesian inference, all of the states and unknown parameters are treated as random variables [38], and the target's position is quantitatively and qualitatively inferred in the form of posterior marginal PDF given priors and finite observations [146]. Among these parametric CL approaches mentioned previously, besides the KF and EKF, the MAP also falls in the Bayesian category, while the LS and MLE are non-Bayesian.

In spite of their advantages, the limitations of NBP algorithms are apparent as well. They only produce exact estimates in non-loopy networks. That is to say, the NBP based approaches are only guaranteed to converge to the correct beliefs in singly connected networks (e.g. tree), where there exists a unique path between an arbitrary pair of nodes. However, less accurate results are obtained for networks multiply connected, or in other words networks with loops/cycles in terms of factor graphs [169] [178]. The reason of this issue, according to [146], is double counting of information as introduced in Section 2.6.4, which eventually results in overconfident beliefs. The

same limitation is also found in the SPAWN, which inherits properties from the belief propagation algorithms [177]. Only few solutions for networks with loops exist in the literature and they are rarely used in localisation [147].

Another limitation comes from a high communication cost in these sample-based message-passing algorithms, where a large number of packets is required, as during every time step, nodes repeat sending and receiving messages until the estimates converge.

2.6.6 Real-Time and Delayed-State Approaches

The real-time approaches are usually concerned with the present estimates of nodes only, which is reasonable under an assumption that the information from a node is known over the network instantly. However, measurement data from a node in many situations may not be received by another node or the base station immediately due to network communication delays or disruptions, as previously mentioned in Section 2.5.4. Information shared in the networks is hence often out of order, and this is defined to be an out-of-sequence measurements (OOSM) problem [20] [37] [183] in data fusion systems. It is possible to proceed with the estimation process to obtain sub-optimal estimates by incorporating the latest measurements, whilst discarding all out-of-date ones. The only way to achieve an optimal update under this situation is to have all measurements, including the time delayed, properly incorporated.

Delayed-state filtering emerges from simultaneous localisation and mapping (SLAM) domain to cope with time latency in landmark initialisation and communication. Approaches based on a delayed-state concept differ from the conventional cooperative localisation by maintaining a history of agents' states to allow the fusion of past data when received. [122] and [16] introduce the concept of delayed-state decentralised data fusion (DS-DDF), with past estimates retained in the state space vector for managing historical dependencies. Delayed-state information filter (DSIF) is adopted in both [118] and [32] with a canonical-form representation of Gaussian states kept in its sparse information matrix. These works demonstrated the advantages of the

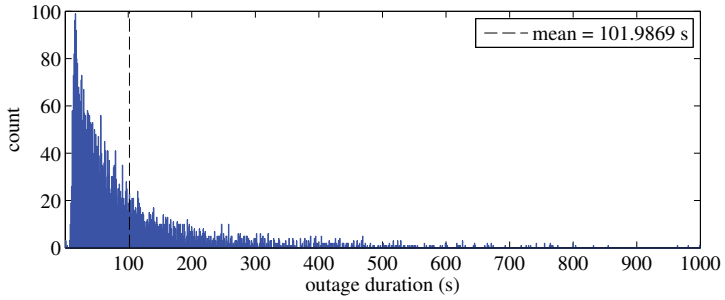
additivity property of the information filter in the update stage, and the ability to cope with information loss due to network latencies. However, this approach requires the Gaussian and linear assumptions to be satisfied.

The delayed-state filtering keeps a joint state space $\mathbf{X}_{k-m:k}$ storing states starting from the state m seconds ago until the present. Given that we have a good estimate of the state m seconds before, i.e. $P\left(\mathbf{X}_{k-m}|\mathbf{Z}_{1:k-m}\right)$, the estimate at present is predicted over time to yield $P\left(\mathbf{X}_{k-m:k}|\mathbf{Z}_{1:k-m}\right)$. Meanwhile it is updated by fusing available "asequent" measurements [122] (i.e., information that arrives with time delays and out of time sequence) received within the time window $(k-m,\ k]$, regardless of their arriving sequence. In short, the process is summarised in the equation below.

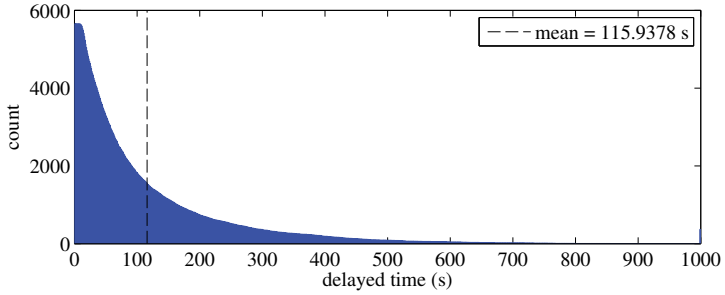
$$P\left(\mathbf{X}_{k-m:k}|\mathbf{Z}_{1:k}\right) \propto \underbrace{\prod_{t=k-m+1}^{k} P\left(\mathbf{Z}=\mathbf{Z}_t|\mathbf{X}_t\right)}_{likelihoods} \times \underbrace{\prod_{t=k-m+1}^{k} P\left(\mathbf{X}_t|\mathbf{X}_{t-1}\right) P\left(\mathbf{X}_{k-m}|\mathbf{Z}_{1:k-m}\right)}_{prediction}$$

The prediction portion in the equation appeared previously in Equation 2.1, which is for the long-term prediction in Section 2.4.2. The estimation process could be proceeded by running forward the time window.

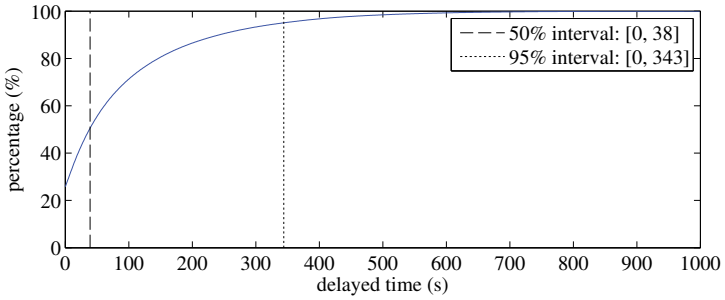
Continuing the analysis given in Figure A.3, Figure 2.8 revealed that the percentage of observations that could be included in the fusion stage, is substantially increased with the delayed-state filtering introduced. A larger time window enables more delayed observation to be incorporated, thereby improving the estimates, in the sense that a better estimate could be yielded with more measurements. Nevertheless, it is at the expense of higher memory and computational costs. The filtering results of a delayed-state Kalman filter along with the arrival of delayed information are demonstrated in Figure 2.9 as an example. Noteworthy is that, by comparing Figures 2.9(d) and 2.10, the fully updated delayed-state filter outperforms the standard Kalman filter provided with the same observation set but without time delays. This could be explained by a native smoothing effect in delayed-state filtering, which uses "future" information to further reason past state estimates.



(a) Distribution of observation outage duration

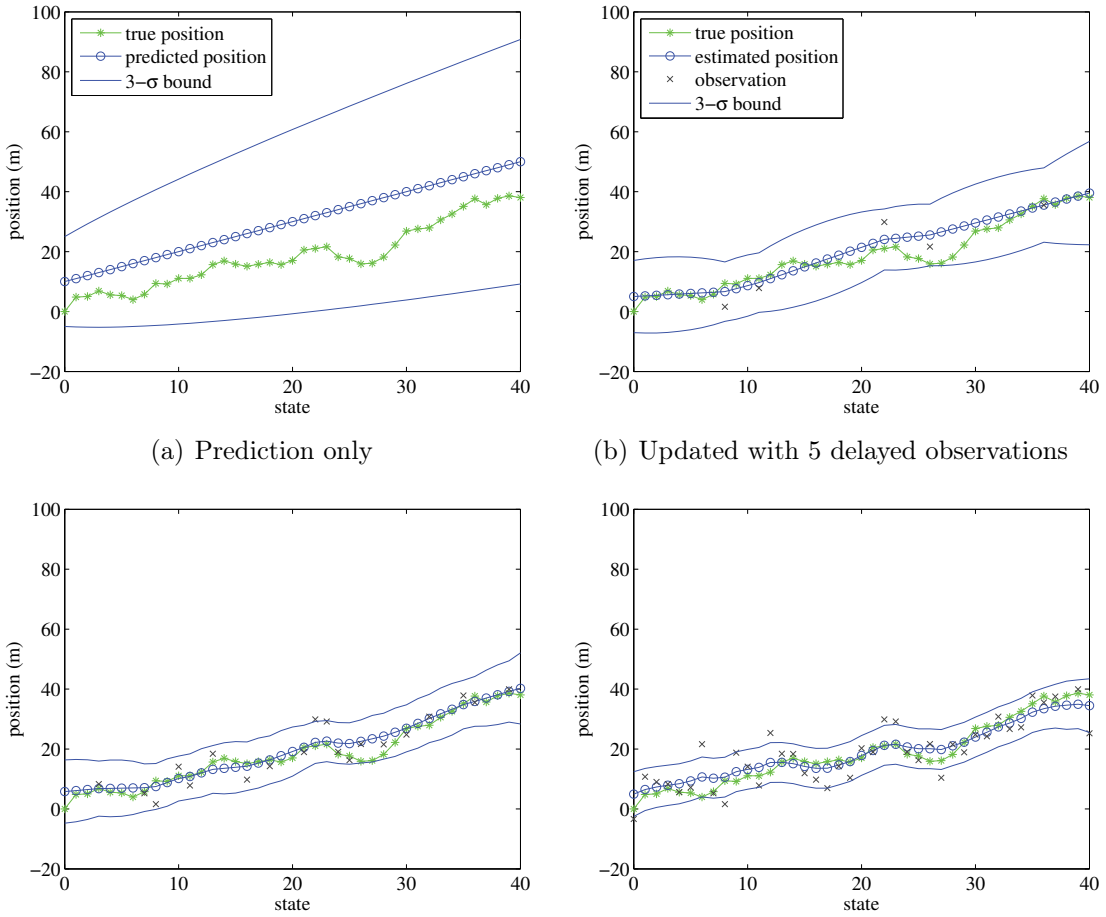


(b) Distribution of time delayed observations



(c) Cumulative observation percentage

Figure 2.8 – Analysis on delayed observations. (a) and (b) respectively show distributions of observation outage duration and time delayed observations in the case of 9 nodes and 9 sensors illustrated in Figure A.1 in Appendix A. (b) is processed under an assumption that each node generates one set of observations per second. Interestingly the distribution in (a) is Rayleigh-like while (b) seems a halved Gaussian distribution. The cumulated percentage of observations (real-time and time delayed) is illustrated in (c). According to (c), particularly for the case under study, less than 30% percent of observations are real-time and can be considered by a conventional real-time filtering algorithm. This is improved when the delayed-state concept is introduced to the filtering. A delayed-state filter with a time window of 38 seconds will have half of all observations fused. The number will increase to 95% if the time window size is expanded to 343 seconds.



(a) Prediction only

(b) Updated with 5 delayed observations

(c) Updated with 20 delayed observations

(d) Updated with all historical observations

Figure 2.9 – A Kalman filter with the delayed-state concept introduced is able to fuse delayed and out of order observations, while in a standard Kalman filter only real-time observations are accepted.

2.6.7 High-Dimensional State Space Issue

In centralised CL, the dimensional curse brought by high dimensionality of state space remains a challenge particularly when parametric filtering methods are not applicable. Generally speaking, the ensemble size and computations required in PF grows exponentially with the dimension of state space [154]. The high dimensional nature of cooperative tracking accounts for inefficiency of importance sampling [116]. Undeniably the use of PF for multiple target tracking is confronted with the well-known "curse of dimensionality" [44] [60]. According to the results presented in [23]

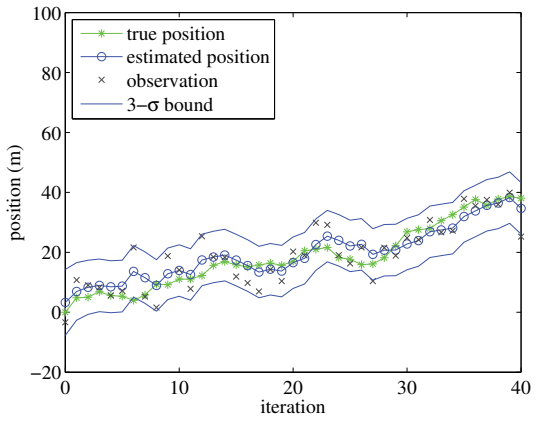


Figure 2.10 – Standard Kalman filter result assuming that all observations arrive in real time. The result is produced with lower accuracy, when compared to that of delayed-state Kalman filter in Figure 2.9(d). The standard filter uses observations up to time t to estimate the state $\mathbf{x}_t$, which means $P(\mathbf{x}_t|\mathbf{Z}_{0:t})$. In the delayed-state one, estimate of a historical state $\mathbf{x}_t$ is updated using information up to a later time $k > t$, i.e., $P(\mathbf{x}_t|\mathbf{Z}_{0:k})$. It is known as "smoothing process" when observations after time t are used to update state at time t.

and [27], the PF update suffers from a weight degeneracy (or weight collapse) problem in which a vast majority of the weights are placed on a single particle, unless the ensemble increases super-exponentially with the system dimension.

The remedies to the degeneracy issue include a form of resampling process [106], a proper choice of proposal distribution [38], diversifying the sample [59] and so on. The issue is however not addressed fundamentally for high-dimensional systems if the requirement of an enormous ensemble size is not met. By marginalising some variables, a Rao-Blackwellised Particle Filter (RBPF) [50] [38] [107] can outperform a standard PF with a reduced state space size. The prerequisite of using RBPF is that a subset of the state space can be handled by a parametric filtering approach.

2.6.8 Markov Chain Monte Carlo Approaches

Markov chain Monte Carlo (MCMC) approaches are generally used to sample from a complicated distribution of interest by using a Markov chain. A Markov chain refers to a sequence of random samples generated by a Markov process, a "memory-

less" stochastic process where the probability of a state is solely conditional on the immediately preceding state and independent on all history states by the preceding, i.e.,

$$P\left(\mathbf{x}_t = \theta_t | \mathbf{x}_{t-1} = \theta_{t-1}, \mathbf{x}_{t-2} = \theta_{t-2}, \cdots, \mathbf{x}_0 = \theta_0\right) = P\left(\mathbf{x}_t = \theta_t | \mathbf{x}_{t-1} = \theta_{t-1}\right)$$

A simple whilst effective example of MCMC methods is the Metropolis algorithm [112] [113]. It is particularly useful to draw samples to approximate a target distribution $P(\theta)$ that is difficult to sample from directly. Suppose a function $f(\theta)$ that is proportional to the desired distribution $P(\theta)$, briefly the Metropolis algorithm is described as:

- Start with any value θ_0 that satisfies $f(\theta_0 > 0)$.
- For each iteration $j = 1$ to k:
    1. Generate a candidate θ^* for the sample at j according to a proposal density $q\left(\theta^* | \theta_{j-1}\right)$, which must be symmetric, i.e. $q\left(\theta^* | \theta_{j-1}\right) = q\left(\theta_{j-1} | \theta^*\right)$.
    2. Accept the candidate $\theta_j = \theta^*$ with an acceptance probability of $\alpha = min\left(\frac{f(\theta^*)}{f(\theta_{j-1})}, 1\right)$, otherwise reject and $\theta_j = \theta_{j-1}$.

Iterative running of the algorithm produces a Markov chain $(\theta_0, \theta_1, \cdots, \theta_k)$, with the newer sample θ_t sometimes moving to a new random point otherwise staying at the previous point. Following a sufficient length of burn-in iterations N_{bi}, only every $(N_{si})^{th}$ sample in the generated Markov chain is saved to reduce autocorrelation and such a thinned output approaches the desired target distribution.

The Metropolis algorithm was generalised to Metropolis-Hastings by Hastings in [69] by adopting an arbitrary proposal density $q\left(\theta^* | x_{j-1}\right)$, and the acceptance ratio function becomes $\alpha = min\left(\frac{f(\theta^*) q(\theta^* | \theta_{j-1})}{f(\theta_{j-1}) q(\theta_{j-1} | \theta^*)}, 1\right)$. A detailed review of this method is provided in [41].

From the previous subsection, clearly we have learnt the limitation of the importance sampling strategy in PF to high-dimensional state space. The problem of low efficiency of sampling is that a traditional importance sampling based algorithm tries to sample all components of the state space simultaneously but with an inadequate ensemble size. As one of the most effective approaches to complex probabilistic systems, a Gibbs sampler breaks the high dimensional state space in CL by sampling from each target state conditionally to others instead of from the joint, thereby scaling well with the dimensionality. The fundamental idea is that drawing samples from a low-dimensional conditional distribution is usually much easier than drawing from the joint. The Gibbs sampler is a special case of the Metropolis-Hastings algorithm where the proposal densities are the conditionals and the acceptance rate of a random move is constantly one. Though having its origin from image processing in [58], the Gibbs sampler [33] has been extensively used for dynamic state space models in Bayesian statistics, see examples in [48] [81] [45] [139] [128].

Consider a joint distribution of N random variables $P\left(\theta^{1:N}\right) = P\left(\theta^1, \cdots, \theta^N\right)$ that is to sample from, each Gibbs sampling iteration after the initialisation step involves the updating of one of the variables by a new value drawn from the distribution of that variable conditional on the values of the rest variables. That is to say,

- Start with deterministically or randomly $\theta^1 = \theta_0^1, \cdots, \theta^N = \theta_0^N$.
- For each iteration $j = 1$ to k and each variable $i = 1$ to N, draw $\theta_j^i \sim P\left(\theta^i | \theta_j^1, \cdots, \theta_j^{i-1}, \theta_{j-1}^{i+1}, \cdots, \theta_{j-1}^N\right)$.

Note that instead of sampling from the state space of total N dimensions directly using the conventional importance sampling method, the Gibbs sampler samples N times in one-dimensional state spaces. The samples of the joint distribution is then obtained after removing the burn-in period and thinning the chain, as the same processes that are used in the Metropolis algorithm.

The performance of the MCMC algorithms, however, tends to be unreliable when the proposal densities are poorly chosen, or high correlations exist between some

variables [48]. In these situations, it becomes more difficult to have the state space well explored by the Markov chain and produce accurate results. Furthermore, it lacks an effective diagnostic method for convergence of the Markov chain [38], with which the sampling process could be monitored, and stopped when stationary of the sampler has been reached for a determined period of time. Another essential problem related to an MCMC method is its high computational demand, which makes it less feasible in on-line state estimation applications [59] [38]. Nonetheless, because of its attractive capability in addressing high-dimensional problems, there is some recent work towards the application of the MCMC algorithms for real-time multi-target tracking and analyses, by considering proper optimisations. For instance, [94] reduces the computational cost by adopting the concepts of auxiliary variable PF and Rao-Blackwellising, while [139] takes advantage of a hybrid structure of MCMC and importance sampling algorithms. The issues and applicable optimisations of MCMC algorithms for a real-time analysis of dynamic systems are discussed in [107].

2.7 Summary

In this chapter, we provided fundamental reviews on the background of mobile target tracking and considerations relevant to the tracking problem in large environments.

The long observation outage duration problem in a sparsely-connected network leads us to the conclusion that a consistent vehicle model for long-term motion prediction is in demand during the outage period. The nonparametric motion model is able to deal with the non-linear and non-Gaussian properties present in vehicle dynamics, and essentially to have environment properties incorporated to help constrain potential trajectories of vehicles.

Conventional MANETs are known to suffer from the isolated agent issue in geographically segmented networks in scenarios with large areas. From this emerges the use of DTNs to cope with the delays in data transmission and network disruptions. It does not require a reliable and real-time end-to-end route to be established for communication. The store-and-forward and opportunistic transmission properties of DTNs

have made it an appropriate candidate for data dissemination problems in large environments.

As one of the findings in the preliminary investigation in Appendix A, collaboration between nodes is beneficial in shortening the measurements outage length, thereby improving estimates. Cooperative tracking is able to eliminate the requirement of an infrastructure, and reduce the requirement of accessibility to absolute position information by introducing relative observations between two ordinary nodes. Furthermore, improved estimates are achievable in the use of delayed-state filtering, which is capable of fusing measurements that arrive with time delays in DTNs. The inefficient importance sampling from a high-dimensional state vector is yet an obstacle towards cooperative tracking using a traditional particle filter. It motivates the incorporation of MCMC methods into cooperative tracking, which is a sequential Bayesian inference problem with a high-dimensional state space.

In the upcoming chapter, we will define the information types and the observation models used throughout the thesis.

Chapter 3

From Information to Observations

3.1 Introduction

This section will present and define all types of information that could assist to constrain the possible position of a mobile agent in large environments. These types include absolute, relative, positive, negative, real-time, time delayed, which are used throughout this thesis. The definitions of the observation types are given in Section 3.2. We also make use of the motivating example in the first chapter to facilitate the understanding of each type. This part is included in Section 3.3. Section 3.4 explains observation types that are considered in different main contributions of the research. The vehicle proximity detection approach is introduced in Section 3.5, and details about it could be found in Appendix C. It is then followed by Section 3.6, which defines representations of the observations in the form of likelihood functions. The current arrangement of this chapter is for the convenience that the observation definition and observation models are discussed before their uses in subsequent chapters.

3.2 Definitions

3.2.1 Absolute or Relative Observations

As mentioned, GPS has been widely used in tracking applications to provide absolute position information of the target(s), see Figure 3.1(a). Egocentric observations of the absolute position for each vehicle could be obtained from an on-board GPS sensor. This state is dependent only on the vehicle itself. Conventionally, the likelihood function of an absolute egocentric observation for a vehicle v_p is represented by:

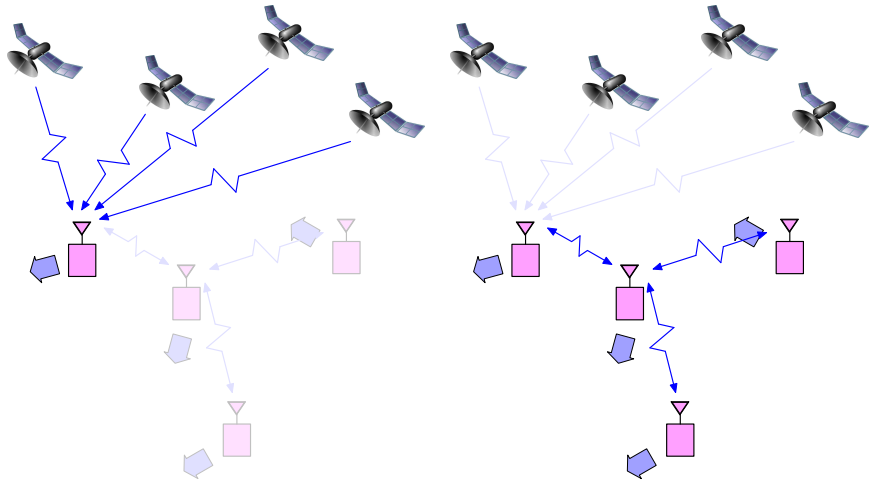
$$P\left(\mathbf{z}^{v_p} = \mathbf{z}_t^{v_p} | \mathbf{x}_t^{v_p}\right)$$

A solution to the accuracy degradation or the signal outage issue of GPS with a low cost and good scalability is to utilise the relative information between targets, as shown in 3.1(b). Some examples of the types of relative observations used in existing localisation/tracking approaches are time of arrival (TOA) [177], angle of arrival (AOA) [53], phase measurements, RSSI [36], amongst others. Trilateration, see Figure 3.2, or triangulation is then used to find the position of an unknown node. The TDOA and AOA data fusion methods, corresponding to trilateration and triangulation respectively, are investigated in [148]. Interesting analyses about the quality of the different types of relative observations are presented in [131] and [65].

In this thesis, the filter considers two types of relative information: V2V and V2I. The V2V relative measurement between vehicles v_p and v_q is dependent on the states of the two vehicles involved. Hence the likelihood function of it is represented by:

$$P\left(z^{v_p \to v_q} = z_t^{v_p \to v_q} | \mathbf{x}_t^{v_p}, \mathbf{x}_t^{v_q}\right)$$

Those V2I relative observations are about the relative measurements between vehicles and fixed data collection points. More specifically, observations generated by the data collection infrastructure are considered either as positive (vehicle is detected) or



(a) GPS (b) Pairwise measurement

Figure 3.1 – Absolute and relative observations. (a) a GPS device mounted on vehicles is a typical source of absolute referenced information. (b) the pairwise measurements between vehicles are considered relative observations.

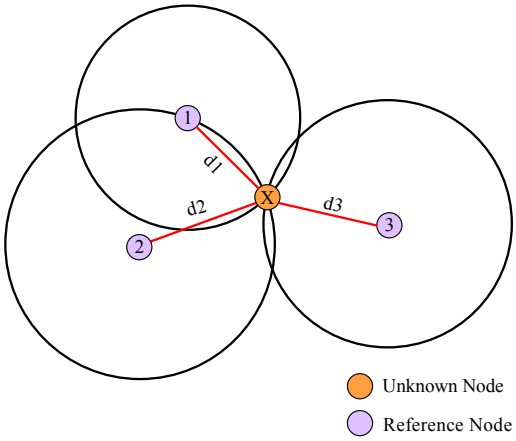


Figure 3.2 – Trilateration. The unknown node is localised with relative ranges to at least 3 reference nodes, which have their global position known.

negative (vehicle not detected). As the data collection point is fixed at a predetermined position, the relative information of a vehicle in proximity is converted into an absolute referenced observation. Given that the position of a stationary data collection point c_q is predetermined to be a Dirac delta function at $\left(x_{c_q}, y_{c_q}\right)$, the likelihood

function of the absolute referenced observation, that whether or not a vehicle v_p is in proximity of the data collector, is denoted by:

$$P\left(z^{v_p \to c_q} = z_t^{v_p \to c_q} | \mathbf{x}_t^{v_p}, \mathbf{r}^{c_q} = \begin{bmatrix} x_{c_q} & y_{c_q} \end{bmatrix}^T\right)$$

In the remainder of the thesis, we also adopt the term "absolute referenced observations" to represent the V2I relative observations, which are generated by fixed data collection points. This is because every V2I relative observation is converted in the first place.

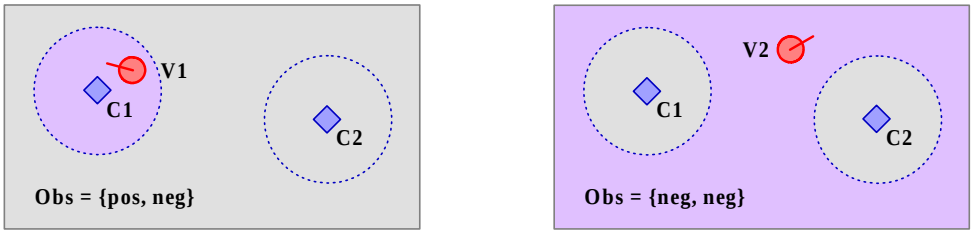
3.2.2 Positive or Negative Observations

The absolute referenced observation from a data collector contains either positive or negative detection of a vehicle. The relative information that whether or not a vehicle is nearby can be converted into an absolute referenced observation since the communication range and the location of the collection point are deterministic. When a vehicle establishes V2I communication with a data collection point, it is considered positive detection, which indicates the presence of the vehicle within the detection area of the collection point. The detection area around the data collection point where V2I communication is possible is bounded and represented by a circle centred at the collection point covering the potential communication range.

On the other hand, if the data collector does not communicate with a vehicle for a period of time, it is considered negative, absolute referenced information. The inference from this is that the vehicle is likely to be somewhere outside the detection area. It is important to note that this allows some level of localisation if the base station does not have available GPS information for the vehicle. There are two examples demonstrated in Figure 3.3, one showing positive information and the other negative.

Likewise, the V2V relative observation could also be either positive or negative. Such a relative observation between two vehicles indicates positive or negative detection by

a vehicle about another. In addition, positive detection of a vehicle is accompanied by an RSSI measurement, which can be used to refine the range information.



(a) Positive information (b) Negative information

Figure 3.3 – Positive and negative information scenarios with fixed data collection points C1 and C2. (a) Vehicle V1 is detected near collector C1. The location of V1 is constrained to the coverage area of C1, excluding the possibility that V1 is outside the area of C1. (b) V2 is outside the coverage area of both C1 and C2, reducing the probability that V2 is close to collectors C1 and C2.

3.2.3 Real-time or Time Delayed Observations

The absolute referenced observations from the infrastructure (i.e. data collection points) are considered real-time information, which becomes available to the base station as soon as it is generated. This includes positive (vehicle is detected) or negative (vehicle not detected) information about the likelihood of a vehicle being in the area around the data collection points. On the other hand, observations produced by vehicles running in the field are shared between vehicles via V2V interactions and eventually flow to a data collection point or the base station through V2I communication. An example of the information flow was illustrated in Figure 1.8 in Section 1.3.

The relaying of the observations is defined in this thesis as observation harvesting, which is the process where vehicles that are not in contact with the infrastructure can still have its information transmitted back to the central base station. The observations of interest in the observation harvesting include vehicle egocentric position updates and V2V relative measurements. They are shared amongst vehicles and before they are eventually brought to a fixed data collector, it could have been a

potentially significant time elapsed since they were generated. Suppose an observation generated at time t is received by the infrastructure at the present time k. It is considered a time delayed observation with a latency of $k - t$ seconds when $t < k$. The latency can be on the order of minutes for large environments.

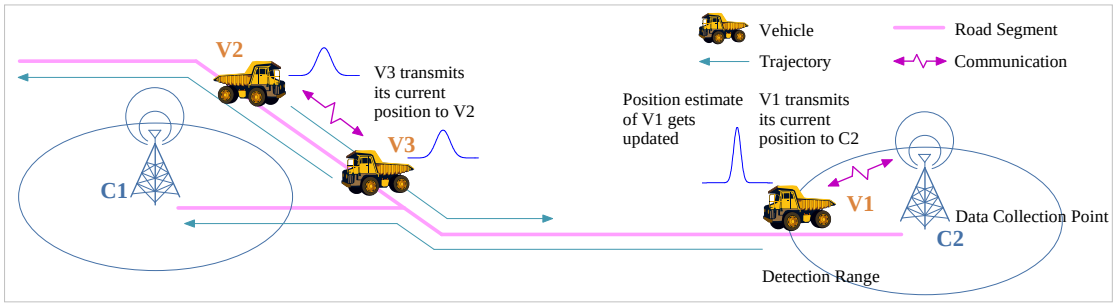
There is a special case where a harvested observation is considered real-time information. When a vehicle is within the communication range of any data collection point, its observations are immediately known by the base station through V2I communication, which means $t = k$. This is the best case in the observation harvesting process as the time delay is zero.

Delayed observations are also useful in reducing the uncertainty of position estimates, especially when real-time observations are not available. A filter must be able to utilise both real-time and delayed observations in this scenario to achieve a higher tracking accuracy. Figure 2.9 in Section 2.6.6 demonstrated how delayed observations help constrain the estimate of the target position by using a delayed-state filter.

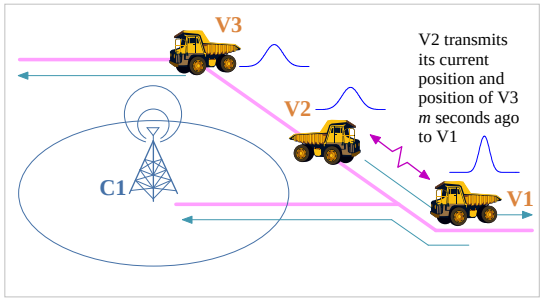
3.3 Continuing the Motivating Example

Recall the motivating example presented again in Figure 3.4. We will use this example to show the different types of observations described above.

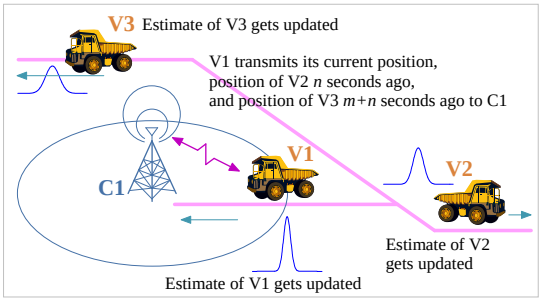
1. From the viewpoint of the base station, information at time $t_{now-m-n}$ is summarised as follows.
    1.1. C1 did not detect V2 and V3.
    1.2. C2 detected V1 in its detection range.
    1.3. C2 acquired the position of V1 at the time provided from GPS reading through a direct connection to it.
    1.4. C2 acquired the information that V1 did not detect V2 and V3 at this time.



(a) $t_{now-m-n}$



(b) t_{now-n}



(c) t_{now}

Figure 3.4 – A duplicate of Figure 1.7. In (a), the position information of vehicle V3 was known to V2 when they met. The information was then forwarded to V1 when V1 and V2 moved close, as shown in (b). Lastly in (c), through data collection point C1, the base station received the information, though delayed for V2 and V3, from returning V1 and estimation updates for all three are achieved.

1.5. C2 did not detect V2 and V3.

2. Information at time t_{now-n} from V1's perspective is summarised as below.

2.1. V1 detected V2 in its detection range.

2.2. V1 got the position of V2 at this time through a P2P connection to it.

2.3. V1 acquired the position of V3 m seconds ago, which was carried by V2.

2.4. V1 acquired the relative information that V2 and V3 detected each other m seconds ago.

2.5. V1 acquired the information that it was detected by V2.

2.6. V1 acquired the information that it was not detected by V2 and V3.

2.7. V1 acquired the information that V2 did not detect V3.

2.8. V1 did not detect V3.

3. At time t_{now}, all information carried by V1 is known to the base station via a direct connection to collection point C2. We summarise below the information available at the base station end.

   3.1. C1 detects V1 in its detection range.

   3.2. C1 gets the position of V1 at this time through a direct connection to it.

   3.3. C1 acquires the position of V2 n seconds ago through V1.

   3.4. C1 acquires the position of V3 $m + n$ seconds ago through V1.

   3.5. C1 acquires the information that V1 and V2 detected each other n sec ago.

   3.6. C1 acquires the information that V1 and V2 did not detect V3 n sec ago.

   3.7. C1 acquires the information that V2 and V3 used to detect each other $m + n$ sec ago.

   3.8. C1 gets the information that V2 and V3 did not detect V1 $m + n$ sec ago.

   3.9. C1 acquires the information that V1 does not detect V2 and V3.

   3.10. C1 does not detect V2 or V3.

   3.11. C2 does not detect V1, V2 or V3.

The observations from the information listed above are grouped into different types, as shown in Table 3.1. Please note that the "real-time" and "time delayed" are about the time an observer received the observation, relative to the time the observation was created. For example, in the information described above in 1.3, the absolute egocentric observation of V1 obtained by C2 was considered real-time as it was transmitted to the infrastructure as soon as it was created at time $t_{now-m-n}$. On the contrary, in the information indexed as 3.3, the absolute egocentric observation of V2 was a time delayed one because it was not known by C1 until n seconds later than the time it was generated.

Table 3.1 – Observations types of information in Figure 3.4(a)

Observation Type			Index of Information
absolute	real-time referenced	positive	1.2, 3.1
		negative	1.1, 1.5, 3.10, 3.11
	egocentric	real-time	1.3, 2.2, 3.2
		delayed	2.3, 3.3, 3.4
V2V relative	real-time	positive	2.1, 2.5, 2.7
		negative	1.4, 2.8, 3.9
	delayed	positive	2.4, 2.6, 3.5, 3.7
		negative	3.6, 3.8

Table 3.2 – Observations types considered in main research contributions

Observation Type			Main Contribution		
			I	II	IV
absolute	real-time referenced	positive	✓	✓	✓
		negative	✓	✓	✓
	egocentric	real-time	✓	✓	✓
		delayed		✓	✓
V2V relative	real-time	positive			✓
		negative			✓
	delayed	positive			✓
		negative			✓

3.4 Observation Types Considered in Main Research Contributions

In the first main research contribution (**Chapter 4**) of this thesis, only real-time absolute egocentric and referenced position observations from vehicles and data collectors respectively are used in the fusion process. Comparing Main Contributions I and II in Table 3.2, it can be seen that the second main contribution (**Chapter 5**) is improved based on the first one with delayed absolute egocentric observations taken into consideration in its filtering algorithm. In the first two main contributions of

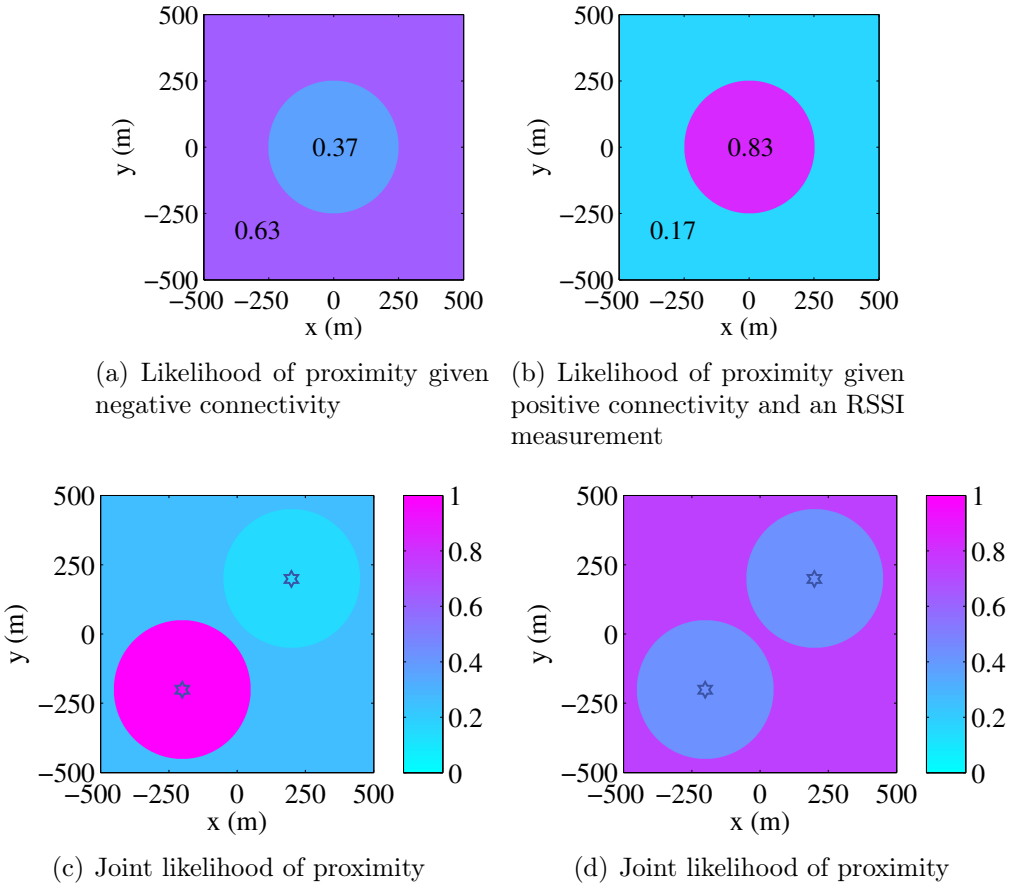
the research, the filters track vehicles individually, which means that V2V relative observations are not considered in these stages. This is because the fusion of this type of V2V information requires the consideration of cross-correlations between vehicles. In Main Contribution IV (**Chapter 7**), the tracking approach has all kinds of observations defined in this chapter considered in the filtering process. This cannot be realised without the proposed nonparametric cooperative tracking algorithm, which is able to deal with delayed and V2V relative observations. The cooperative tracking approach proposed in Main Contribution IV is extended from that in Main Contribution III (**Chapter 6**). Therefore the latter one is not listed in the table for the discussion.

3.5 Vehicle Proximity Detection

The thesis proposes a vehicle proximity detection approach, to infer the approximate position of a vehicle relative to another or to a fixed data collection point, given V2V or V2I relative information, respectively. This approximate relative range information is defined as "vehicle proximity" information in this thesis.

Specifically, the observation data required in the vehicle proximity detection approach is simply wireless connectivity and RSSI measurements. The information is provided by the wireless transceivers installed at each node (vehicle or infrastructure). The connectivity measurement could be either *Positive Connectivity (PC)* or *Negative Connectivity (NC)*, respectively corresponding to the positive or negative detection discussed previously.

However, geographical proximity of two nodes does not necessarily lead to positive connectivity between them. Similarly, negative connectivity does not necessarily mean two nodes are out of range, though positive connectivity certainly infers a high probability that two nodes are in range. Instead of considering a deterministic relationship between connectivity and proximity, the proposed vehicle proximity detection approach incorporated historical V2V and V2I communication data to build up probabilistic observation models. A detailed description of the approach could be



(a) Likelihood of proximity given negative connectivity

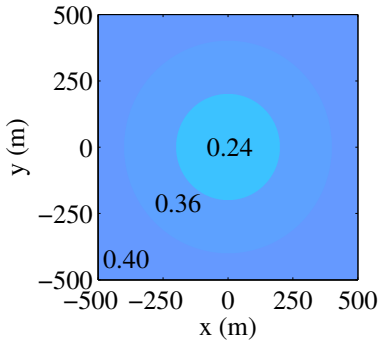
(b) Likelihood of proximity given positive connectivity and an RSSI measurement

(c) Joint likelihood of proximity

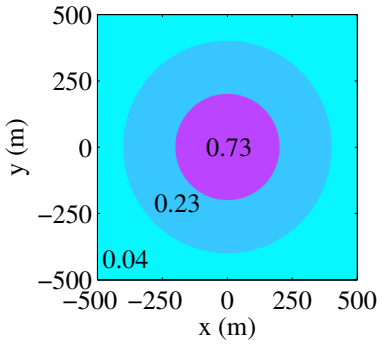
(d) Joint likelihood of proximity

Figure 3.5 – Examples of likelihood of proximity between a vehicle and an infrastructure. (a) demonstrates the likelihood of distance given a negative V2I connectivity, while the likelihood given positive connectivity and a RSSI measurement of -105 dBm is presented in (b). (c) illustrates the joint likelihood given two observations (one positive and the other negative) from two data collection points. Another example of the joint likelihood (two NCs) is given in (d).

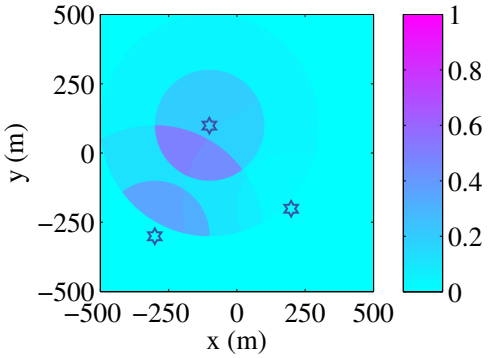
found in Appendix C. For an absolute referenced observation, it is a V2I observation $z_t^{v_p \to c_q}$ generated by a data collector c_q about a vehicle v_p. A *PC* event is observed, along with an RSSI measurement, when the data collection point receives a packet from the vehicle, otherwise an *NC* event occurs. Several likelihood examples given V2I relative observations are presented in Figure 3.5. A V2V relative observation $z_t^{v_p \to v_q}$ also contains RSSI information in the case of a positive connectivity event. Figure 3.6 demonstrates some examples of likelihood functions given V2V relative observations.



(a) Likelihood of proximity given negative connectivity



(b) Likelihood of proximity given positive connectivity and an RSSI measurement



(c) Joint likelihood of proximity

Figure 3.6 – Examples of likelihood of proximity between vehicles. (a) demonstrates the likelihood of distance given negative connectivity. The likelihood given positive V2V connectivity observation and an RSSI of -85 dBm is presented in (b). (c) shows the joint likelihood given multiple observations (two positive and one negative) from three observer vehicles.

3.6 Notation of Likelihood Functions

The likelihood function given an absolute egocentric observation of a vehicle v_p at time t from an on-board egocentric localisation device is denoted as:

$$\widetilde{\Psi}_t^{v_p} \triangleq P\left(\mathbf{z}^{v_p} = \mathbf{z}_t^{v_p} | \mathbf{x}_t^{v_p}\right) \tag{3.1}$$

Though an absolute egocentric observation could be either real-time or time delayed,

Table 3.3 – Likelihood from information in Figure 3.4(c) in Main Contribution I

	Observer (at $t = t_{now}$)				
Target	V1	V2	V3	C1	C2
V1	N/A	N/A	N/A	$\Psi_t^{V1\to C1}$	$\Psi_t^{V1\to C2}$
V2	N/A	N/A	N/A	$\Psi_t^{V2\to C1}$	$\Psi_t^{V2\to C2}$
V3	N/A	N/A	$\widetilde{\Psi}_t^{V3}$	$\Psi_t^{V3\to C1}$	$\Psi_t^{V3\to C2}$

Table 3.4 – Likelihood from information in Figure 3.4(c) in Main Contribution II

	Observer (at $t = t_{now}$)				
Target	V1	V2	V3	C1	C2
V1	$\widetilde{\Psi}_t^{V1}$	N/A	N/A	$\Psi_t^{V1\to C1}$	$\Psi_t^{V1\to C2}$
V2	N/A	$\widetilde{\Psi}_t^{V2}$	N/A	$\Psi_t^{V2\to C1}$	$\Psi_t^{V2\to C2}$
V3	N/A	N/A	$\widetilde{\Psi}_t^{V3}$	$\Psi_t^{V3\to C1}$	$\Psi_t^{V3\to C2}$

Table 3.5 – Likelihood from information in Figure 3.4(c) in Main Contribution IV

	Observer (at $t = t_{now}$)				
Target	V1	V2	V3	C1	C2
V1	$\widetilde{\Psi}_t^{V1}$	$\widetilde{\Lambda}_t^{V1\to V2}$	$\widetilde{\Lambda}_t^{V1\to V3}$	$\Psi_t^{V1\to C1}$	$\Psi_t^{V1\to C2}$
V2	$\widetilde{\Lambda}_t^{V2\to V1}$	$\widetilde{\Psi}_t^{V2}$	$\widetilde{\Lambda}_t^{V2\to V3}$	$\Psi_t^{V2\to C1}$	$\Psi_t^{V2\to C2}$
V3	$\widetilde{\Lambda}_t^{V3\to V1}$	$\widetilde{\Lambda}_t^{V3\to V2}$	$\widetilde{\Psi}_t^{V3}$	$\Psi_t^{V3\to C1}$	$\Psi_t^{V3\to C2}$

we do not differentiate these two types in terms of likelihood functions for the convenience of formulations. The real-time one is considered as a special case of the time delayed absolute egocentric observation with zero delay.

The likelihood function for the relative V2V observation between two vehicle nodes v_p and v_q at time t is denoted by:

$$\widetilde{\Lambda}_t^{v_p\to v_q} \triangleq P\left(z^{v_p\to v_q} = z_t^{v_p\to v_q} | \mathbf{x}_t^{v_p}, \mathbf{x}_t^{v_q}\right) \tag{3.2}$$

Again the real-time and delayed V2V relative observations are of the same notation of the likelihood function.

Provided that the position of a data collector c_q is a Dirac delta function at $\left(x_{c_q}, y_{c_q}\right)$,

the likelihood function given an absolute referenced observation between a vehicle v_p and the data collector at time t is represented by:

$$\Psi_t^{v_p \to c_q} \triangleq P\left(z^{v_p \to c_q} = z_t^{v_p \to c_q} | \mathbf{x}_t^{v_p}, \mathbf{r}^{c_q} = \begin{bmatrix} x_{c_q} & y_{c_q} \end{bmatrix}^T\right) \tag{3.3}$$

Continuing the example shown in Figure 3.4(c), Table 3.3 interprets observations gathered by the base station at time t_{now} and can be considered in the first main contribution of the research in the form of likelihood functions defined above. More observations are able to be fused in Main Contributions II and IV, see Tables 3.4 and 3.5, and naturally lead to improved tracking accuracy. In the last main contribution, V2V relative observations are considered as complimentary information to absolute ones in the fusion stage. The relative information is particularly useful to constrain vehicle positions in cases of degraded GPS and GPS outage, in the use of the cooperative tracking framework.

3.7 Summary

This chapter examines the information/observation aspect of the proposed approach. It defines observations according to various types of information that could be used to constrain positions of vehicles. It is important to note that the proposed vehicle proximity detection model is not aiming at accurate ranging, which requires consideration of complicated signal propagation models. Instead, an approximate distance between a pair of vehicles could be probabilistically determined by combining communication connectivity and RSSI measurements. This is easily achieved at no additional bandwidth cost, as nodes necessarily communicate when they meet. The next chapter will present the approach for the long-term vehicle motion prediction, in which the real-time absolute egocentric and referenced observations are used.

Chapter 4

Probabilistic Long-Term Vehicle Motion Prediction

4.1 Introduction

This chapter presents an innovative algorithm that incorporates parameters obtained from the environment and the vehicle history to obtain consistent long-term vehicle prediction. It introduces ASPs and timing profiles into the motion model, and to describe correlation of motion parameters for vehicle movement. The proposed approach is to install V2I communication capabilities in the fleet of vehicles and use a number of fixed data collection points to receive position updates when the vehicles are in range. The remainder of the chapter is organised as follows. Section 4.2 introduces the modelling of the environment including roads, intersections and areas, followed by a probabilistic formulation of a long-term motion prediction algorithm in Section 4.3. Experiment results are presented in Section 4.4. Lastly the work presented in this chapter is summarised in Section 4.5.

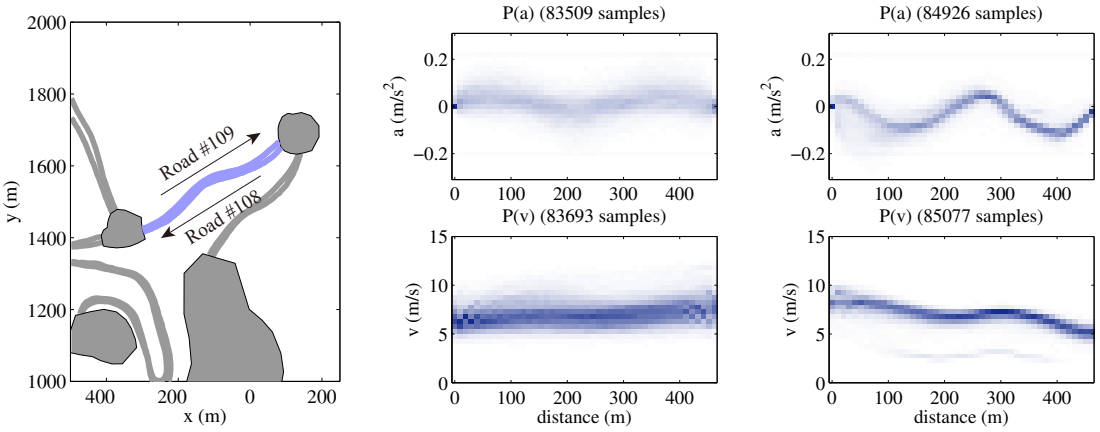
4.2 Modelling Environment Properties

One of the fundamental requirements to model the behaviour of the vehicle is to obtain an accurate description of the areas of operation. The operating environment of mining vehicles usually covers very large areas. This environment can be represented by a road network based on context areas (loading areas, parking lots, etc.) connected by winding road segments and intersections formed by crossing roads. As an example, Figure 1.6 in Section 1.3 illustrates this map representation for an actual mining operation. This information is built based on [4] and [5] and is incorporated as prior knowledge. The dynamics of a vehicle in such an environment is dictated by the physics of the possible vehicle motion. The motion of the vehicles in this context is affected by many different environmental factors, such as condition and curvature of the road, visibility, traffic stream variables [34], weather conditions and more. Another important constraint is that the vehicle motion is ultimately bounded by the road networks [67].

4.2.1 Behaviour of Vehicles

The constant velocity model is the most basic model to predict vehicle motion. In practice however, the vehicle velocity is constrained by many factors. A vehicle may run faster on a straight or downhill road. On the other hand, harsh weather conditions (e.g. foggy, rainy, snowy), and terrain variations such as corners and upside gradients, will generally result in lower velocity. In cargo handling applications, the vehicle load could also affect the speed. In addition to the physical constraints, a vehicle will always have certain probability to stop anywhere and at any time due to operational reasons (taking a break), engine problems, queuing/traffic or in the event of an accident.

A vehicle has a much higher probability to stop at an intersection, or in a “context area” such as parking lot, loading area, etc. than on a road. A driver could cross an intersection without deceleration when there are no other vehicles around. In



(a) Roads #108 and #109 on field map

(b) Road #108

(c) Road #109

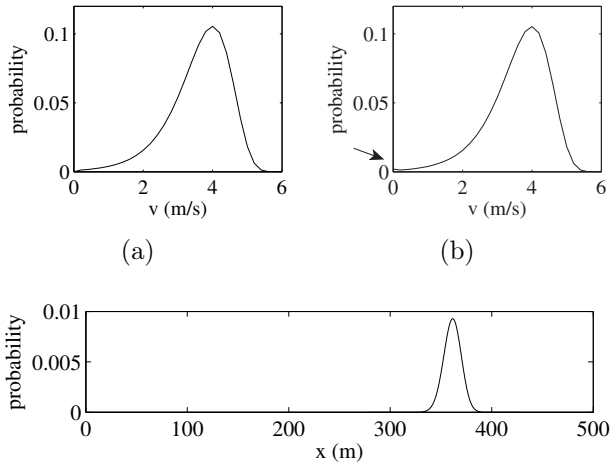
Figure 4.1 – Acceleration and speed profiles of two parallel roads #108 and #109. (b) and (c) illustrate distributions of acceleration and speed against distance along roads #108 and #109 respectively. In spite of having almost the same road curvature, the acceleration and speed profiles are quite different mainly due to the differing road gradients.

other cases, the driver could wait for a short period until it is safe to continue, or an extended period in the event of heavy traffic conditions. There could also be a particular context area where the vehicles are known to stop for a period of time. For example, a vehicle could spend 2 minutes picking up cargo at a loading area, then wait 30 seconds at an intersection when moving to dumping area, and take 1 minute to dump the load. There are many other possible stopping cases, such as the driver taking a break, swapping drivers, refuelling the vehicle to name a few.

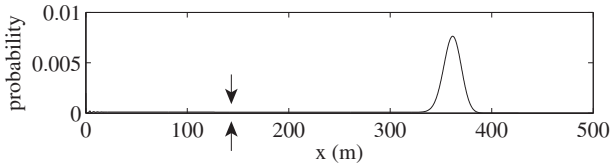
The next section presents a technique to incorporate these uncertainties into the prediction models.

4.2.2 Modelling Road Segments

Determining the potential range of vehicle dynamics for each part of the road is necessary to improve the long-term position estimate. In order to obtain this information, a road segment connecting two intersections or areas can be divided into slots. For each road segment in the road network, and each slot on the road segment, histograms



(c) Postion estimate with the speed PDF in (a)

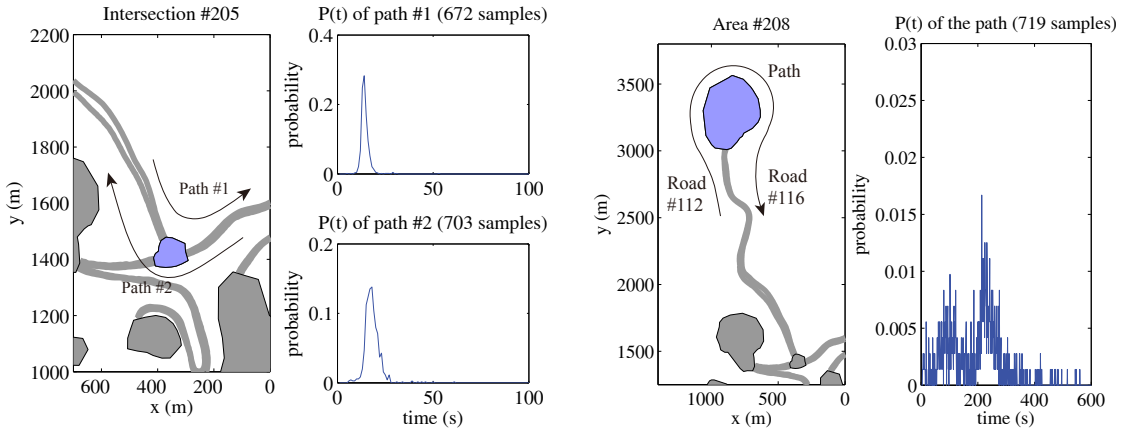


(d) Postion estimate with the speed PDF in (b)

Figure 4.2 – Speed PDF with and without stopping probability. (a) Speed PDF assuming the vehicle runs all the time. (b) Speed PDF with stopping probability. This model has a non-zero value at speed zero. The position estimate after 100 iteration for the non-stopping speed PDF gives a belief of position with all probability mass distributed around its mode, see (c). On the other hand, (d) shows that the prediction with a stopping probability has long tail behind to represent a non-zero probability of a stopping vehicle.

of the vehicle acceleration and speed are generated with the historical motion data collected from the vehicles in operation [175], [174]. An example obtained from experimental data is shown in Figure 4.1(a), showing two road segments connecting two intersections. Each road segment represents a one directional lane, meaning that a two-way road is made up of two road paths.

Predictions models generally assume that the vehicle is moving all the time, which is not always a correct assumption. The vehicle could stop anywhere because of unpredictable factors such as an engine fault or an accident. The kinematic model used in this chapter is improved by incorporating the stopping probability. Figure 4.2 demonstrates the difference in the prediction outcome resulting from the inclusion



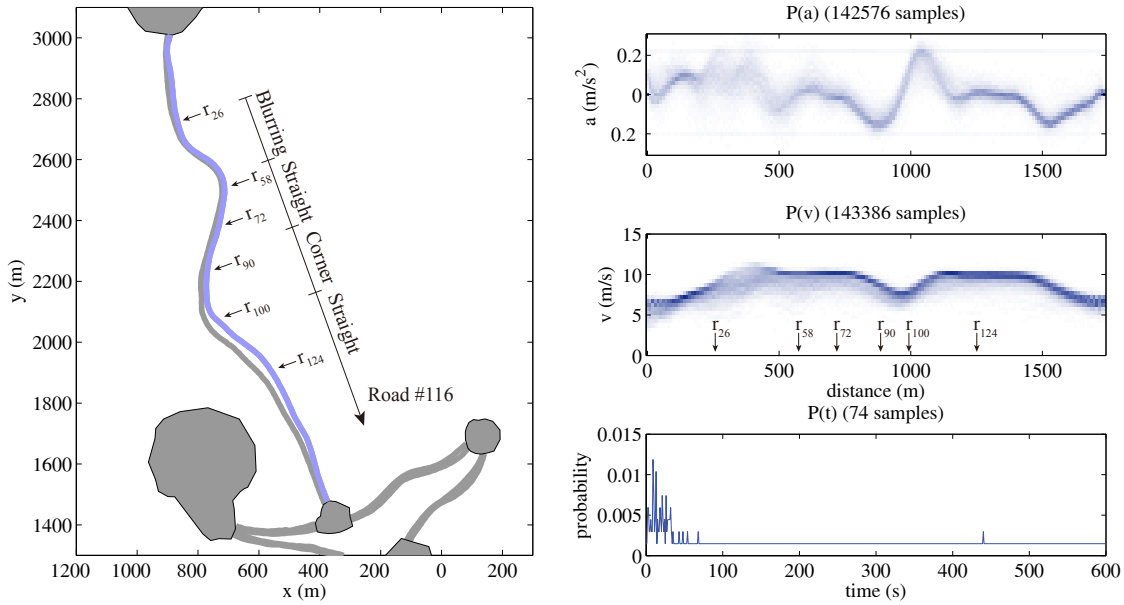
(a) Intersection #205 (b) Area #208

Figure 4.3 – Timing profiles of intersection and area. (a) illustrates 2 paths that can be chosen to cross the intersection #205. The timing profiles for these two opposite paths are clearly different. (b) demonstrates a spread timing distribution with roughly two modes, due to queuing time potentially spent in the loading area #208.

of the stopping probability. It is clear that the model that includes the stopping probability is more representative of the real world situation, where the probability of a vehicle stopping at any position along its trajectory is non-zero.

4.2.3 Modelling Intersections and Areas

Vehicle behaviour at intersections and other important context areas is less structured, with more complex activities when compared to a road. For this reason, it is not feasible to model the vehicle motion by considering a speed profile as the motion is primarily determined by other factors such as movement of nearby vehicles. The increased probability of the vehicle to stop for any amount of time can be represented using a "timing profile", which can empirically determine the probability of the time taken for a vehicle to traverse a context area. Suppose an intersection has M entry roads and N exit roads, generating $M \times N$ potential paths. For each path, a timing profile can be built with the statistical distribution of time taken to cross this intersection or area. It is necessary to determine the timing profile for each potential path as the timing is likely to be different depending on the entry and exit points of



(a) Road #116 on field map

(b) Road #116

Figure 4.4 – Acceleration, speed, and timing profiles of road #116. (a) shows segments on road #116. Some example position slots are labelled in both (a) and (b) (r_x means position slot #x). The blurring segment in (a) is characterised with blurred acceleration and speed distributions found in (b) (from distance about 250 to 500 m). It is followed by two straight segments, and a corner segment ranging approximately from 800 to 1000 m. The blurring effect reflects high motion diversity when vehicles pass the segment. When getting into the corner segment, vehicles generally have to slow down for safe pass, and accelerate when moving out. Furthermore, vehicles are likely to keep a constant speed when running on straight segments of road #116.

the context area. In Figure 4.3(a), the timing profile can be seen to be different for each path crossing the intersection. According to a spreading distribution in Figure 4.3(b), the time taken for vehicles traversing area #208 is rather uncertain.

A timing profile can also be used to determine the probability of the length of time a vehicle will stop on a particular road based on the empirical evidence. If a vehicle was to stop on a road, the timing profile is used to determine when the vehicle is likely to start moving again. Examples of timing profiles for road #116 can be seen in Figure 4.4(b). Note that a small uniform distribution is added into timing profile of a road during post-processing, to avoid empty bins caused by insufficient stopping samples.

4.2.4 Acceleration Speed Profiles

The kinematic model for moving vehicles was created using an acceleration speed profile (ASP) for each position slot on road segment, statistically representing vehicle movement characteristics given a particular position. From the kinematic point of view, a strong inter-relationship exists between acceleration, speed, and position. Vehicles are likely to speed up on a straight road but slow down when getting close to a corner. A vehicle driving at a slow speed could possibly be decelerating in preparation to stop, or preparing to accelerate in order to ramp up to the average speed for that section of road.

This section introduces a measure of the correlation between vehicle acceleration, speed, and position that is implicitly built into ASP, which describes $P(a, v|\mathbf{r})$.

The motion data along with the position on road segment #116 are given in Figure 4.4, where the variation in distributions of acceleration and speed can be seen. However, the correlation between acceleration and speed is not obvious. An example of ASP on a particular position slot is illustrated in Figure 4.5(a). There are very important relationships that are revealed by evaluating the acceleration speed correlation coefficient ρ, also known as Pearson's r which is defined by:

$$\rho = \frac{cov(\mathbf{a}, \mathbf{v})}{\sigma_a \sigma_v} \tag{4.1}$$

where $\mathbf{a}$ is a random Gaussian acceleration variable with mean μ_a and standard deviation σ_a. $\mathbf{v}$ is a random Gaussian speed variable with mean μ_v and standard deviation σ_v.

Positive value of ρ means a positive correlation of acceleration and speed and vice versa. A weak relationship between them is represented by zero or near zero ρ. Suppose the speed and acceleration at time k are $\mathbf{v}_k$ and $\mathbf{a}_k$ respectively, then the variance of speed at time $k+1$ can be evaluated using the basic kinematic equation $\mathbf{v}_{k+1} = \mathbf{v}_k + \mathbf{a}_k \Delta T$.

$$\sigma_{v_{k+1}}^2 = cov\left(\mathbf{v}_{k+1}, \mathbf{v}_{k+1}\right) = \sigma_{v_k}^2 + \Delta T^2 \sigma_{a_k}^2 + 2\Delta T cov\left(\mathbf{a}_k, \mathbf{v}_k\right)$$

Substituting Equation (4.1) into the above yields:

$$\sigma_{v_{k+1}}^2 = \sigma_{v_k}^2 + \Delta T^2 \sigma_{a_k}^2 + 2\Delta T \rho_k \sigma_{v_k} \sigma_{a_k}$$

Define R_k as a threshold of acceleration speed correlation coefficient used to determine whether or not the correlation coefficient ρ will lead to divergence of speed PDF at $k+1$.

$$R_k = -\frac{\Delta T \sigma_{a_k}}{2\sigma_{v_k}}$$

The variance of speed at time $k+1$ fulfils:

$$\sigma_{v_{k+1}}^2 \begin{cases} < \sigma_{v_k}^2, & \rho_k < R_k \\ \geq \sigma_{v_k}^2, & \rho_k \geq R_k \end{cases} \tag{4.2}$$

The evolution of speed distribution could be predicted by Equation (4.2). Generally, a negative acceleration speed correlation with $\rho < R_k$ makes the speed PDF converge at the next time step. On the other hand, divergence of speed PDF occurs with $\rho > R_k$. Together with μ_a, some typical vehicle instantaneous motion trends on roads are summarised in Table 4.1 and explained by ρ.

It can now be seen that the ASPs obtained from experimental data match the analysis of motion trend obtained by ρ and μ_a. When vehicles drive into a corner segment at approximately 800 to 1000 m along road #116, the ASP information in Figure 4.5 shows that vehicles running at a high speed have to slow down for safety while the slower speed ones do not. This is presented in vi) of Table 4.1. As predicted in v), vehicles at a low speed accelerate when they are moving out of the corner, see Figure 4.6.

An obvious blurring segment (from distance about 250 to 500 m) on road #116 is found on both acceleration and speed PDFs in Figure 4.4. The blurring effect reflects high motion diversity when vehicles pass this segment. The ASPs when vehicles are about to enter and leave the blurring segment are shown in Figures 4.7 and 4.8. On the position slot #26, the blurring of speed PDF begins with a positive correlation of acceleration and speed which matches the divergence case described as trend vii) in Table 4.1. The speed PDF then starts to become concentrated at the position slot #58, matching the trend iv). Two additional examples in Figures 4.9 and 4.10 show vehicles' intention to keep a constant speed on straight parts of road #116 via analysis of the ASPs. The cases correspond to i) in the table.

It should be emphasized that the term of acceleration speed correlation ρ is introduced only for analysis purposes. The correlation information has been built into the ASPs implicitly. In post-processing, the ASPs are added with a constant positive acceleration for low speed bins and a negative one for high speed ones. This is done to deal with bins without samples as well as to constrain the predicted speed PDF.

Table 4.1 – Instantaneous motion trends on a particular position slot

	Conditions	Motion Trend
i)	$\mu_a = 0, \rho = R$	Most vehicles keep constant speed.
ii)	$\mu_a > 0, \rho = R$	Vehicles tend to speed up.
iii)	$\mu_a < 0, \rho = R$	Vehicles are likely to slow down.
iv)	$\mu_a = 0, \rho < R$	Higher speed vehicles slow down, while lower speed ones accelerate.
v)	$\mu_a > 0, \rho < R$	Vehicles running at lower speed accelerate while higher speed ones maintain current speed.
vi)	$\mu_a < 0, \rho < R$	Lower speed vehicles keep constant speed but ones with higher speed decelerate.
vii)	$\rho > R$	Vehicles keep moving but slower ones tend to decelerate further, possibly preparing to stop.

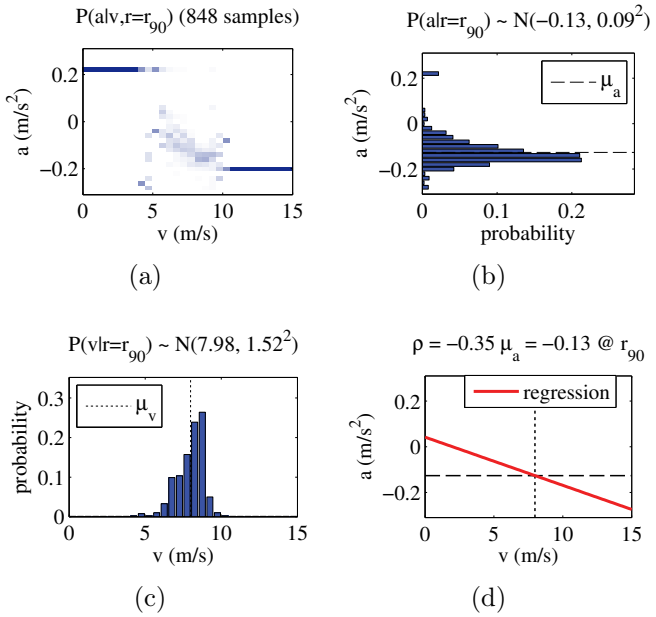


Figure 4.5 – Acceleration speed profile when entering the corner (r_{90}, i.e. position slot #90) on road #116. (a) describes acceleration PDF depending on speed on the position slot. (b)/(c) represents acceleration/speed distribution on the position slot, obtained by marginalising speed/acceleration away from the ASP. Mean acceleration μ_a is found to be -0.13, standard deviation $\sigma_a = 0.09$ and $\sigma_v = 1.52$. (d) shows the correlation between acceleration and speed. The ASP at the position suggests a negative μ_a and a correlation coefficient $\rho = -0.35$, smaller than threshold $R = -0.03$ on the particular position. The correlation information indicates that the vehicles running at a speed higher than average have to slow down for safety purposes while the slower ones do not, which is presented in vi) of Table 4.1.

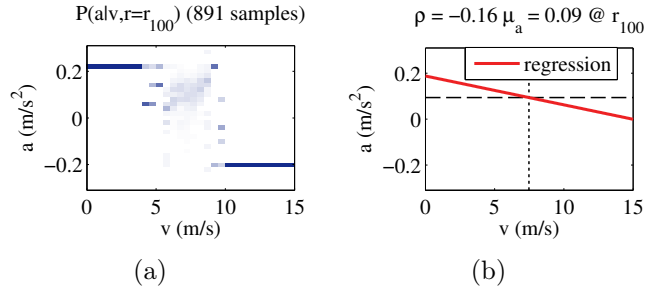


Figure 4.6 – Acceleration speed profile when leaving the corner (r_{100} on road #116. (a) shows acceleration conditional on speed on the position slot. (b) As predicted in v) of the table, vehicles at a low speed accelerate according to a positive μ_a and $\rho < R$ when they are moving out from the corner.

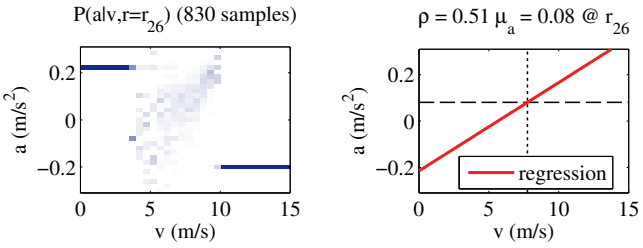


Figure 4.7 – Acceleration speed profile when entering the blurring segment on road #116. On the position slot #26, the blurring of speed PDF begins with a positive correlation of acceleration and speed i.e. $\rho = 0.51$, which matches the divergence case described as trend vii) in Table 4.1.

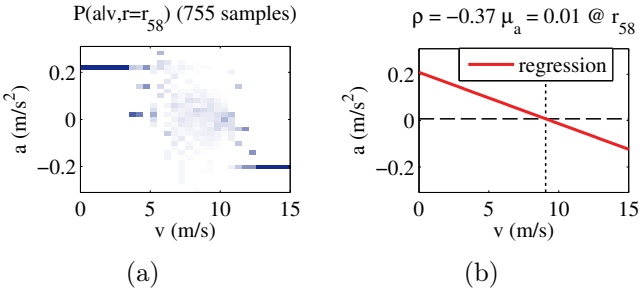


Figure 4.8 – Acceleration speed profile when leaving the blurring segment on road #116. The speed PDF keeps spreading out until the position slot #58, at which the ASP gives a $\rho < R$ on the position. With a μ_a almost zero, the speed PDF then begins to concentrate, matching the trend iv) in the table.

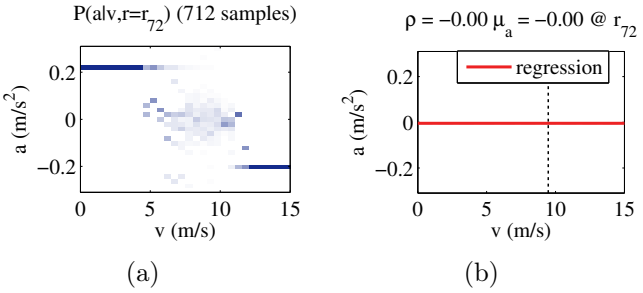


Figure 4.9 – Acceleration speed profile on a straight segments on road #116. It show vehicles' intention to keep a constant speed on straight parts of the road #116, according to near zero μ_a and ρ equal to threshold. These cases correspond to i) in Table 4.1.

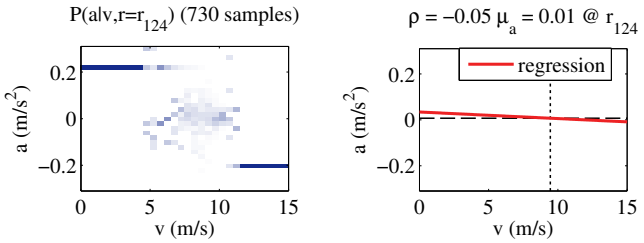


Figure 4.10 – Acceleration speed profile on another straight segments on road #116.

4.3 Long-Term Multiple-Model Motion Prediction

The prediction and tracking problem stated in this chapter can be formulated by multiple motion model approaches, in which multiple motion models are matched to different motion states of the tracked object. Interested readers may find details and examples on this topic in papers [137], [157] and [64].

A vehicle at time k can be in one of three motion states. These are defined as *Moving On Roads (MOR)*, *Stopping On Roads (SOR)* and *Passing Intersections and Areas (PIA)*, i.e.,

$$S_k \in \{s_1 = MOR, s_2 = SOR, s_3 = PIA\}$$

A vehicle may stay in the same motion state, or move from one state to another. Figure 4.11 illustrates the motion state machines and transitions between three states. The state vector for the vehicle at time k is written as:

$$\mathbf{x}_k = [(\mathbf{r}_k)^T \;\; v_k \;\; \tau_k]^T$$

where $\mathbf{r}_k \in \mathbb{R}^2$ is the position of the vehicle at time k, $v_k \in \mathbb{R}$ is the instantaneous speed of the vehicle, and τ_k is the time remaining when stopping on roads or passing intersections and areas. τ_k has a positive value for $S_k \in \{s_2, s_3\}$ states and it is constantly 0 for vehicles in $S_k = s_1$ state. The road map information is introduced

as a function of vehicle position describing two motion environment categories: *Road Segments (RS)*, and *Intersections and Areas (IA)*, which gives:

$$f_{map}\left(\mathbf{r}_k\right) \in \{RS, IA\}$$

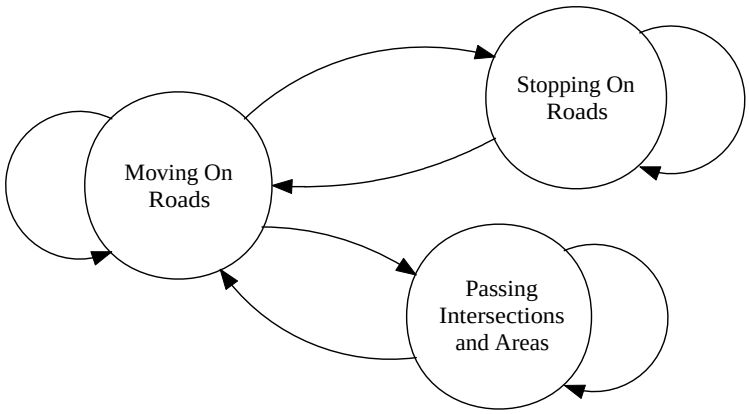


Figure 4.11 – Motion state machines. A vehicle may stay in a motion state, or transits from a state to another when conditions are fulfilled.

4.3.1 State Transition Functions

The transition function represents the probability that a vehicle motion state transits from initial state i to final state j at time k.

$$Ps_{i\to j}\left(k\right) \triangleq P\left(S_k = s_j | S_{k-1} = s_i, \xi_{k-1}\right) \tag{4.3}$$

where ξ_{k-1} is the condition for the state transition, such as if the stopping time runs out, a vehicle moves into an area, or even none.

Transit from *MOR* state to *SOR* state is statistically described by:

$$\begin{aligned} Ps_{1\to 2}\left(k\right) &\triangleq P\left(S_k = s_2 | S_{k-1} = s_1, f_{map}\left(\mathbf{r}_{k-1}\right) = RS\right) \\ &= P_{stopping}\left(\mathbf{r}_{k-1}\right) \end{aligned}$$

where $P_{stopping}(\mathbf{r}_{k-1})$ is stopping probability as a function of vehicle position. It describes the probability that a vehicle may stop when moving on roads.

The probability of staying in *MOR* state is written as:

$$\begin{aligned} Ps_{1\to1}(k) &\triangleq P(S_k = s_1 | S_{k-1} = s_1, f_{map}(\mathbf{r}_{k-1}) = RS) \\ &= 1 - P_{stopping}(\mathbf{r}_{k-1}) \end{aligned}$$

However, transitions from *SOR* state and *PIA* state to *MOR* state given stopping time are deterministic.

$$Ps_{2\to1}(k) \triangleq P(S_k = s_1 | S_{k-1} = s_2, \tau_{k-1} = 0) = 1$$

$$Ps_{3\to1}(k) \triangleq P(S_k = s_1 | S_{k-1} = s_3, \tau_{k-1} = 0) = 1$$

Probabilities of staying in *SOR* and *PIA* states given positive stopping time value are also equal to 1.

$$Ps_{2\to2}(k) \triangleq P(S_k = s_2 | S_{k-1} = s_2, \tau_{k-1} > 0) = 1$$

$$Ps_{3\to3}(k) \triangleq P(S_k = s_3 | S_{k-1} = s_3, \tau_{k-1} > 0) = 1$$

Lastly, a vehicle moving into intersections and areas instantly changes to *PIA* state.

$$Ps_{1\to3}(k) \triangleq P(S_k = s_3 | S_{k-1} = s_1, f_{map}(\mathbf{r}_{k-1}) = IA) = 1$$

Those state transition functions $Ps_{i\to j}(k)$ unlisted above are set to be 0.

4.3.2 Motion Models

The vehicle dynamics is described by a set of motion models. Each transition of states corresponds to a motion model written as:

$$Pm_{i\to j} \triangleq P\left(\mathbf{x}_k|\mathbf{x}_{k-1}, S_k = s_j, S_{k-1} = s_i\right) \tag{4.4}$$

For vehicles staying in *MOR* state, the motion model $Pm_{1\to 1}$ used contains acceleration information conditional on speed v_{k-1} and position $\mathbf{r}_{k-1}$ from ASP at time $k-1$, written as $P\left(a_{k-1}|v_{k-1}, \mathbf{r}_{k-1}\right)$, and simple acceleration-velocity-position kinematic model $P\left(\mathbf{r}_k, v_k|\mathbf{r}_{k-1}, v_{k-1}, a_{k-1}\right)$ with road map information incorporated.

The transitions from *MOR* state to *SOR* and *PIA* states respectively correspond to the motion models $Pm_{1\to 2}$ and $Pm_{1\to 3}$, in which the timing profile $P\left(\tau_k|\mathbf{r}_k\right)$ is included to describe a probabilistic stopping time that the vehicle will stay in either state.

During the time a vehicle is in the *SOR* or *PIA* state, motion models $Pm_{2\to 2}$ or $Pm_{3\to 3}$ are used respectively which involve the remaining stopping time shifting function $P\left(\tau_k|\tau_{k-1}\right)$.

When the stopping time is up, the vehicle in state *SOR* or *PIA* resumes to *MOR* state with motion model $Pm_{2\to 1}$ or $Pm_{3\to 1}$ respectively. Specifically, the latter one brings the vehicle position back to the roads.

4.3.3 Vehicle Egocentric and V2I Observations

Absolute vehicle egocentric observations are provided by on-board GPS devices. This information can contain geographical errors due to noise, multipath and poor satellite constellations. In addition, several data collection points are distributed in the field to operate as observers, offering V2I measurements (absolute referenced observations). The topic on observations has been discussed in **Chapter 3**.

Recall that the likelihood function given an absolute observation of vehicle v_p at time t from the on-board GPS is denoted in the form of:

$$\widetilde{\Psi}_t^{v_p} \triangleq P\left(\mathbf{z}^{v_p} = \mathbf{z}_t^{v_p} | \mathbf{x}_t^{v_p}\right)$$

Also recall that the likelihood function given an absolute referenced observation is represented by:

$$\Psi_t^{v_p \to c_q} \triangleq P\left(z^{v_p \to c_q} = z_t^{v_p \to c_q} | \mathbf{x}_t^{v_p}, \mathbf{x}_t^{c_q}\right)$$

where $z_t^{v_p \to c_q}$ is the observation measured by data collector c_q at time t.

4.3.4 Bayesian Estimation

Prediction Stage

The prediction stage of the multiple models motion tracking for vehicle v_p is written as:

$$P\left(\mathbf{x}_k^{v_p}, S_k^{v_p} | \mathbf{Z}_{1:k-1}^{v_p}\right) = \sum_{i=1}^{3} \sum_{j=1}^{3} \int Pm_{i \to j} Ps_{i \to j}(k)\, P\left(\mathbf{x}_{k-1}^{v_p}, S_{k-1}^{v_p} | \mathbf{Z}_{1:k-1}^{v_p}\right) d\mathbf{x}_{k-1}^{v_p} \quad (4.5)$$

Update with/without Egocentric Observations

If vehicle v_p transmits its information to one of data collection points, i.e. egocentric observation $\widetilde{\Psi}_k^{v_p}$ is available at time k, the filter updates with absolute referenced observations (one of them positive while the rest negative) together with egocentric GPS observation of vehicle node v_p at time k.

$$P\left(\mathbf{x}_k^{v_p}, S_k^{v_p} | \mathbf{Z}_{1:k}^{v_p}\right) \propto \left(\prod_{q=1}^{N_{dc}} \Psi_k^{v_p \to c_q}\right) \widetilde{\Psi}_k^{v_p} P\left(\mathbf{x}_k^{v_p}, S_k^{v_p} | \mathbf{Z}_{1:k-1}^{v_p}\right) \quad (4.6)$$

In Formulas (4.5), (4.6), the observations vector $\mathbf{Z}_{1:k-1}^{v_p}$ is holding all available observations regarding the node v_p up to time $k-1$.

The estimate also gets updated by fusing all negative absolute referenced observations when the vehicle is beyond the detection range of any data collection point. In this case, $\widetilde{\Psi}_k^{v_p}$ does not exist in Equation (4.6) as the egocentric observation is not available at time k.

4.3.5 Particle Filtering Algorithm

Gaussian based tracking algorithms are not appropriate due to non-Gaussian properties of the speed and timing profiles in the vehicle motion model. In addition, the vehicle prediction/tracking problem presents non-linearities in motion state transitions and dynamics models. For this reason, particle filtering [181] [38] [56] is well suited for the type of problem discussed in this paper. So far it has been widely used in vision-based traffic features recognition/detection approaches to track vehicles [153], pedestrians [119], lanes [96] or traffic signs [114].

Table 4.2 – Algorithm: Particle filtering for long-term motion prediction and tracking

$\{\mathbf{x}_k^i, S_k^i, w_k^i\}_{i=1}^L \leftarrow Particle\_Filter\left(\{\mathbf{x}_{k-1}^i, S_{k-1}^i, w_{k-1}^i\}_{i=1}^L, \mathbf{Z}_k\right)$	
1:	draw L particles from initial PDF
2:	**for** $i = 1$ to L **do**
3:	propagation: draw $S_k^i \sim P\left(S_k \vert S_{k-1}^i = s_i, \xi_{k-1}\right)$
4:	draw $\mathbf{x}_k^i \sim P\left(\mathbf{x}_k \vert \mathbf{x}_{k-1}^i, S_k^i, S_{k-1}^i\right)$
5:	update weight with absolute referenced observations: $w_k^i \sim \left(\prod_{q=1}^{N_{dc}} \Psi_k^{v_p \rightarrow c_q}\right) w_{k-1}^i$
6:	update weight with egocentric observation: $w_k^i \sim \widetilde{\Psi}_k^{v_p} w_k^i$ if $\widetilde{\Psi}_k^{v_p}$ exists
7:	**end for**
8:	normalise weights $\{w_k^i\}_{i=1}^L$
9:	**if** $\widehat{N_{eff}} < N_{thr}$ **do**
10:	Resample with replacement L particles from $\{\mathbf{x}_k^i, S_k^i, w_k^i\}_{i=1}^L$ according to $\{w_k^i\}_{i=1}^L$
11:	**end if**

In the proposed particle filtering algorithm, each vehicle running in the field is tracked by a separate particle filter maintained on a central base station, which uses a set of particles to represent state distribution at time k.

$$\{\mathbf{x}_k^i, S_k^i, w_k^i\}_{i=1}^L \sim P(\mathbf{x}_k, S_k|\mathbf{Z}_k)$$

where $\mathbf{x}_k^i$ is further factorised to $\mathbf{x}_k^i = [(\mathbf{r}_k^i)^T \ v_k^i \ \tau_k^i]^T$.

A description of the particle filtering algorithm is given in Table 4.2 in the form of a pseudocode. To prevent weight degeneracy, resampling is adopted in the algorithm when quantity of effective particles $\widehat{N_{eff}}$ is below threshold N_{thr}.

4.4 Experiment Validation

4.4.1 Experiment Setup

Data from a working mine operation (see Figure 4.12) was used to demonstrate the motion prediction algorithm presented. The vehicle state information collected includes vehicle position and speed (from GPS). Post-processing was performed to eval-

Figure 4.12 – Haul trucks running in a mining field.

uate stopping probability on roads, time taken to resume running as well as the length of time a vehicle spent on an area or intersection. The acceleration, speed and timing profiles are constructed for each road segment, intersection and area based on historical data of 25 days (600 hours) collected from 5 vehicles. A summary of real data used in building up the motion model for some road segments and areas/intersections is given in Table 4.3. An overall stopping probability of 0.03% is determined from the data.

It is important to note that there was no significant change to the operation of the filter when the vehicle models were generated using only half of the available historical data (approximately 2 weeks). It will be demonstrated in the results that the model generated using the first half of the data was very similar to the model using the second half, indicating that in the scenarios presented in this chapter the vehicle model generalises well to new instances of vehicle movement.

The approach presented in this chapter was successfully used to predict and track a fleet of vehicles over the entire site where data was collected. The results included in this chapter were taken from a typical set of road segments and areas using data from a number of vehicles. This scenario was selected as it demonstrates a wide range of road and area geometries and profiles. Similar results were obtained for the other vehicle activities on site.

Table 4.3 – Summary of true data collected

	Cases	Motion Samples	Timing Samples
Road #108	1273	83509	7
Road #109	1228	84926	58
Road #110	1260	72890	71
Road #111	1301	76134	21
Road #112	674	154980	30
Road #116	689	141888	74
Area #203	2711	N/A	2711
Area #205	2881	N/A	2881
Area #208	719	N/A	719
Area #209	2551	N/A	2551

4.4.2 Entropy of Target Agent

Entropy is adopted as a metric to estimate the algorithm performance. The indicator of the uncertainty contained in position estimates of the target vehicle, that is the entropy for a single vehicle at time k is calculated by:

$$H\left(\mathbf{x}_k\right) = -\sum_{s=1}^{n} pm_k\left(s\right) log_2 pm_k\left(s\right) \tag{4.7}$$

where $pm_k\left(s\right)$ is probability mass of position slot s from total n slots along the vehicle's route at time k. A path inside an area/intersection is considered as an individual position slot.

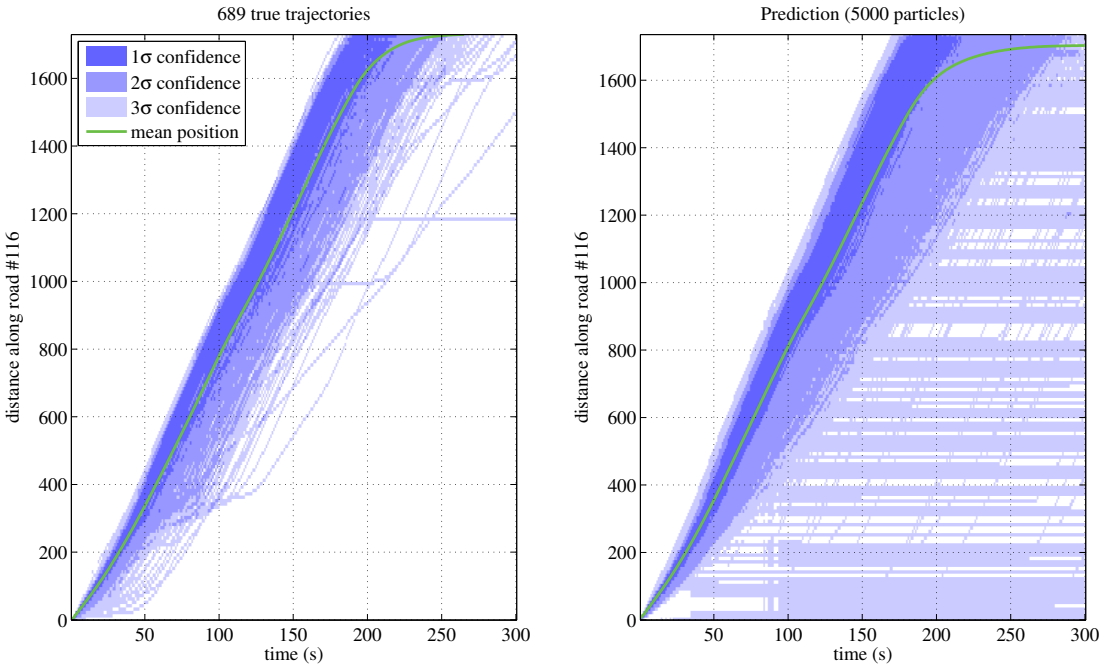
The probability mass on position slot s could be calculated by summing up all weights of particles on the slot, i.e.,

$$pm_k\left(s\right) = \sum_{\mathbf{r}_k^i \in s} w_k^i$$

A brief description of vehicle entropy could also be found in [3].

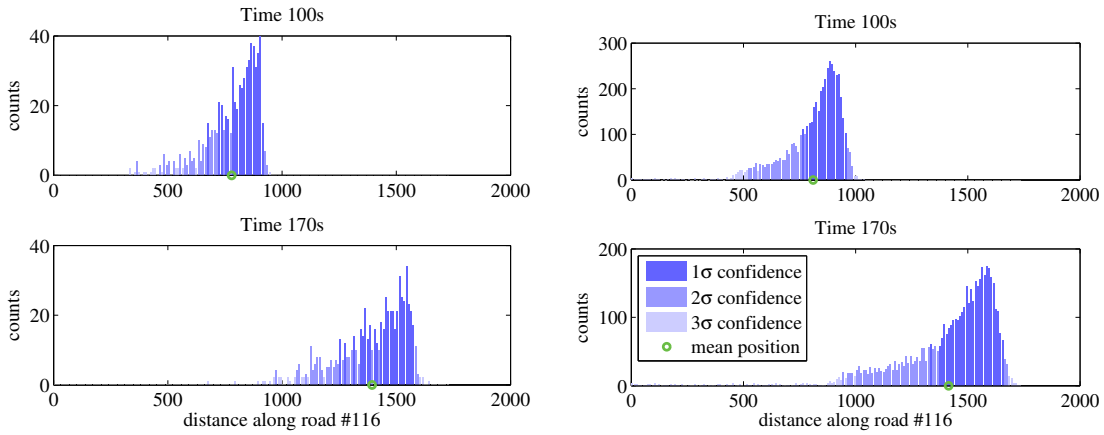
4.4.3 Generic Prediction

A total of 689 true trajectories data on road #116 are shown in Figure 4.13(a). As Figure 4.13(b) depicts, the algorithm gives a prediction with position distribution close to true data. The two figures match well in terms of 1-sigma and 2-sigma confidence areas. However, an apparent difference comes from the 3-sigma areas. We clearly see in the 3-sigma confidence area a few stopping cases running behind the majority in true data. Among them, a vehicle stopped for at least 2 minutes near position 1200 m. Nevertheless, it is not suggesting that a vehicle tends to stop at certain positions. The stopping position is assumed to be unpredictable on the road, as a vehicle could stop anywhere along its trajectory. The limited stopping samples



(a) True trajectories (b) Prediction

Figure 4.13 – True trajectories on road #116 and prediction. Generally the algorithm gives a consistent prediction result with position distribution rather close to true data.



(a) True positions (b) Prediction

Figure 4.14 – Details of true positions on road #116 and prediction. Position distributions on several time slots are demonstrated for a comparison of true positions and prediction. The prediction has a close match to the true data. Long thin tails along prediction trajectories could be clearly observed in the right figures.

contained in the true data set are far from enough to well describe the stopping behaviour on roads, as vehicle stopping is a rare event. As a result of an overall stopping probability along the road considered in the prediction algorithm, Figure 4.13(b) has almost all positions behind the main probability mass covered by 3-sigma confidence area, which is close to the real world situation. The long thin tail along the predicted trajectories can be clearly seen in Figure 4.14. Figure 4.15 illustrates the difference between average position in predictions and mean true positions along with time. It is seen that the prediction error remains under 40 m.

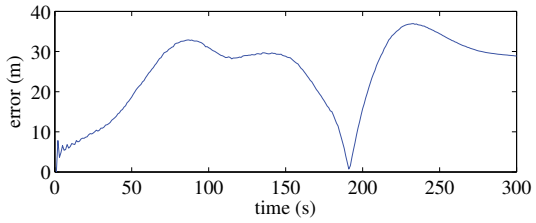
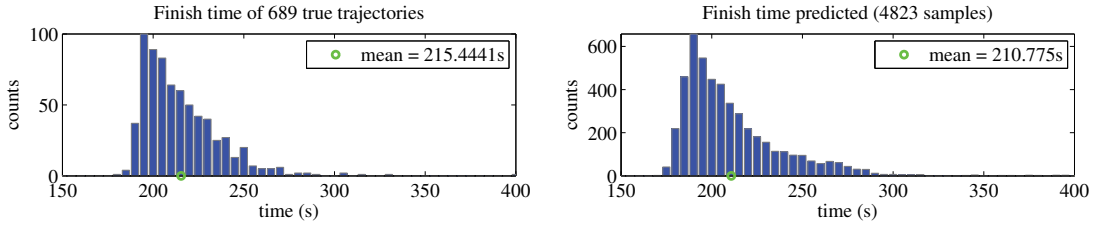


Figure 4.15 – Prediction error on road #116.

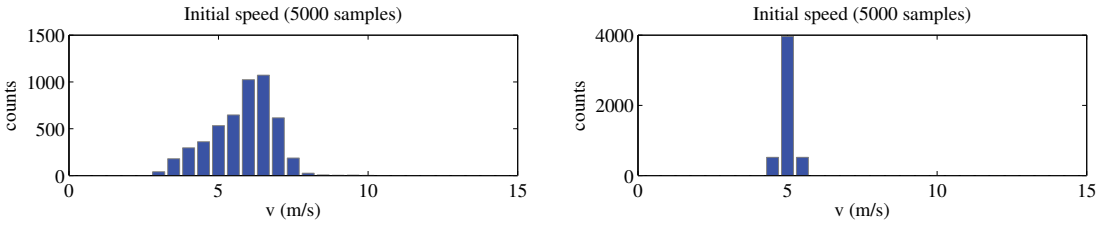


(a) True data (b) Prediction

Figure 4.16 – True finish time on road #116 and prediction by motion model. In (b), 4823 particles finished the trip in 400 s.

Figure 4.16 compares the time taken to finish road segment #116 from the true data with the prediction. It can be seen that the distribution and the mean are very similar for each. A few cases from the collected dataset are observed to have the finish time larger than 300 seconds. This is probably due to stopping on the road. Since the prediction takes into account a stopping probability, every bin over 300 seconds also has non-zero counts in the prediction result.

An interesting comparison is presented in Figure 4.17(a), where the initial vehicle state (position, velocity) are set from the true data. For the case where the real initial speed



(a) Initial speed in generic prediction (b) Initial speed in particular prediction

Figure 4.17 – Initial speed the prediction starts with. a) the generic prediction considers the initial speed of all true cases b) the prediction for a particular true case starts with an initial speed given as a Gaussian distribution with mean equal to true initial velocity, in this case 5 m/s (e.g. $N(5, 0.2^2)$).

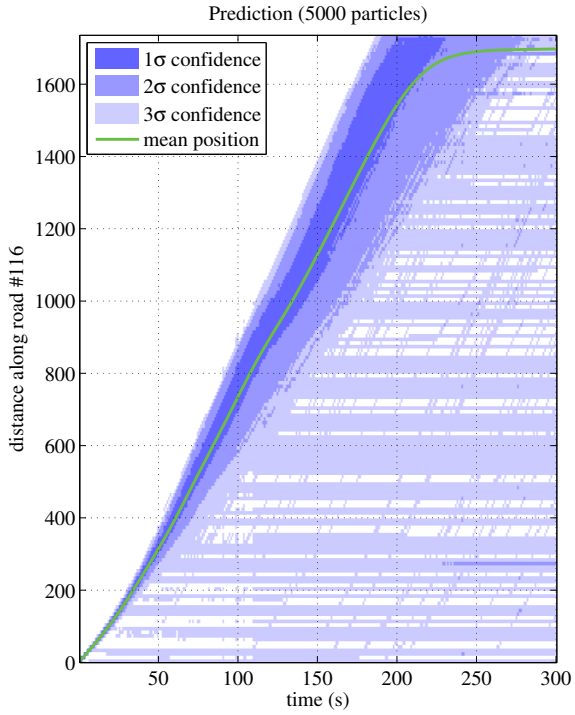
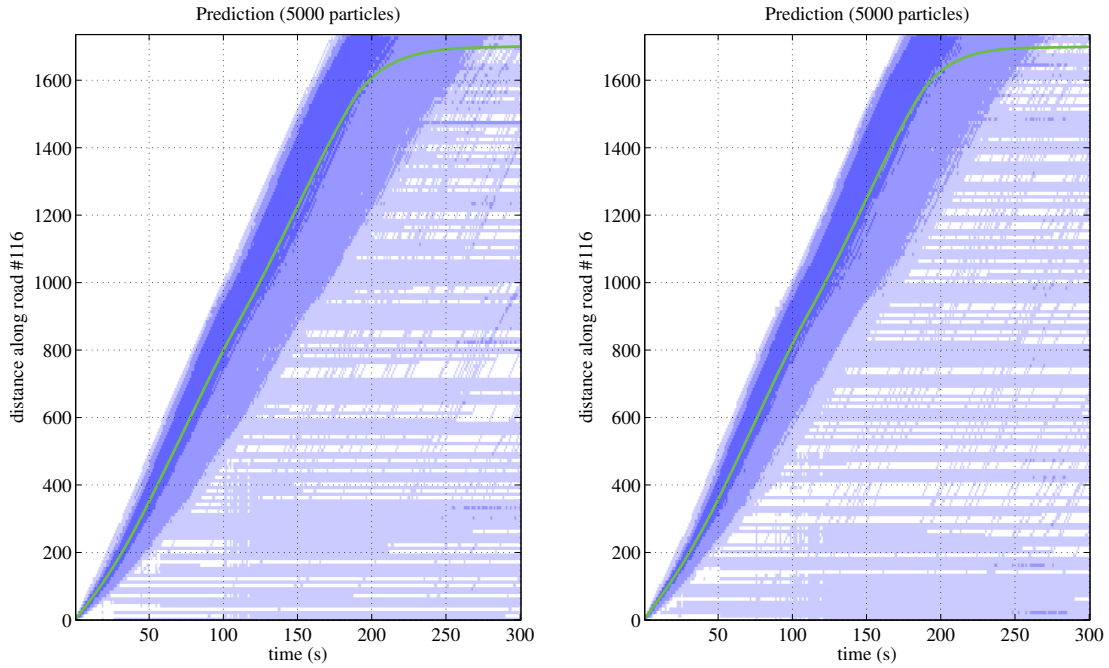


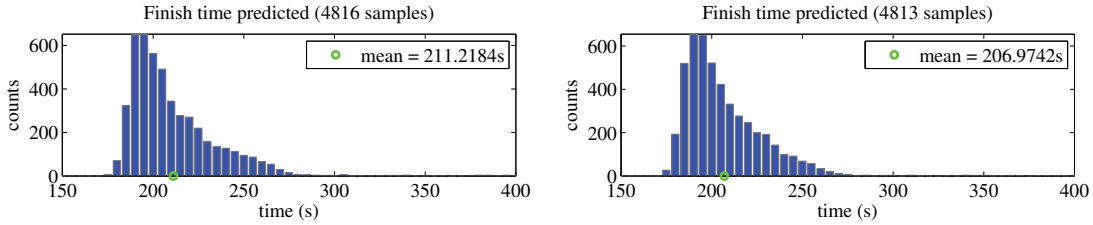
Figure 4.18 – Prediction on road #116 with initial true speed known

and position are known, the prediction yields a more accurate result. As shown in the figure, the algorithm is initialised with speed from 689 true trajectories for general motion predictions. When dealing with a specific true case, the prediction yields a more accurate result with the initial true speed given. As an example demonstrated in Figure 4.18, the prediction result with the initial speed determined as $N(5, 0.2^2)$ is observed to be more constrained compared to the general prediction in Figure 4.13(b).



(a) Prediction using the first half of data (b) Prediction using the second half of data

Figure 4.19 – Predicted trajectories on road #116 by two models independently built with the first and second half of motion data. Generally, the two models produce very close results.



(a) Prediction using the first half of data (b) Prediction using the second half of data

Figure 4.20 – Predicted finish times on road #116 by the two motion models. They are found to only have a difference of about 4 seconds in mean values.

To illustrate that the motion model generalises well with the 25 days worth of data, the total available historical data set was halved into two. Based on these two subsets, two independent motion models were built up and compared. Figure 4.19 show that the prediction results on road #116 by the models are fairly close. This also could be told by comparing the finish times on the road, see Figure 4.20.

4.4.4 Particular Tracking Cases

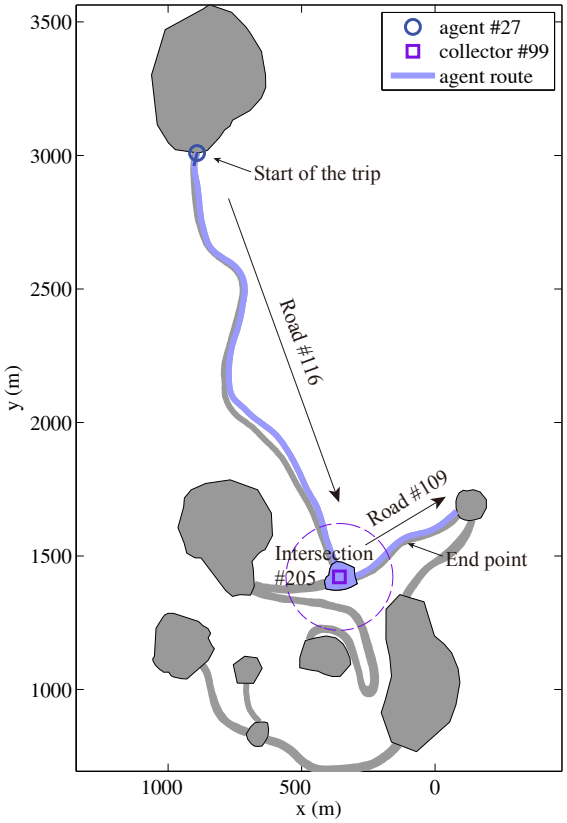
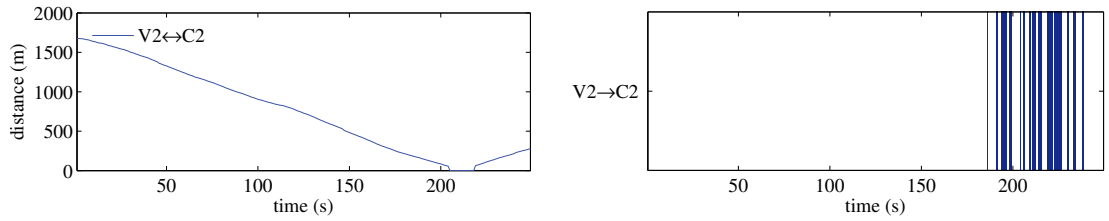


Figure 4.21 – Vehicle route on field map. V2 commenced the trip at the starting point of road #116, communicated with infrastructure when approaching to the data collection point C2 set at the centre of intersection #205, and finished the trip in the middle of road #109. The wireless signal of data collection point covered a circle with a radius of 200 metres.

A Single Vehicle Case without Stopping

The area selected for these experiments is shown in Figure 4.21. In this case one data collector C2 was installed in the intersection #205, and a vehicle V2 moved from one end of road #116, past the intersection #205 before finishing the trip in the middle of road #109. The first experiment used neither the data collector information nor GPS observations and the evaluation of the vehicle position relies on prediction only. The second experiment presents the tracking results considering the egocentric

GPS observations and V2I measurements provided by the data collection points. The third experiment is an identical tracking case to the second one, except for removal of the egocentric GPS observations. With only V2I measurements enabled in update process, the third experiment demonstrates the tracking performance of the algorithm during a GPS outage. When the vehicle trip was initiated, the algorithm predicted (without observations) or tracked (with V2I and GPS observations) the position of the vehicle along the route. Communication between the vehicle and the data collection point was possible when the relative distance between the vehicle and data collector is less than the communication range. A particle filter with 5000 particles was used to implement the estimation. Tracking with 1000 particles gives close result to 5000, however the latter was chosen for better visualisation effect of figures.

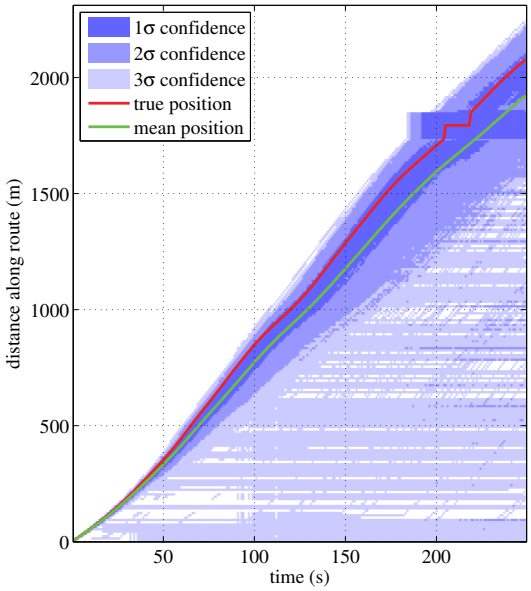


(a) V2I relative distance (b) V2I communication activity

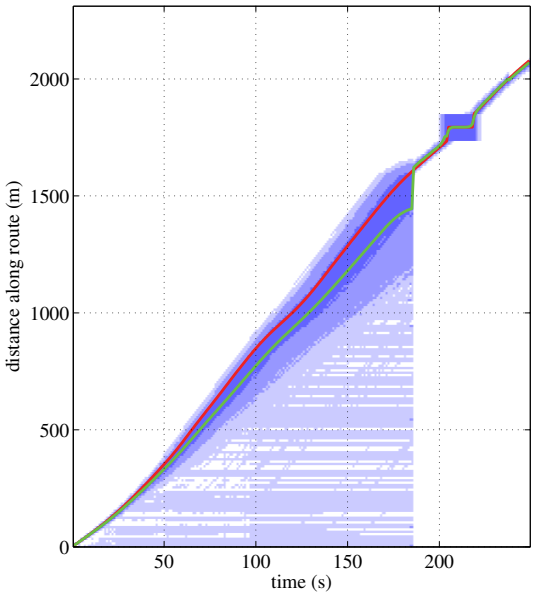
Figure 4.22 – V2I interactions in tracking with GPS case. a) demonstrates change of distance between the vehicle node and the data collection point. Blue lines in b) denote success of V2I communication and blank if fail. The same is the V2I communication in the case of tracking with GPS outage. However no communication activity existed for the prediction only case.

Figure 4.22 summarises interactions between the vehicle and the infrastructure. The relative distance between them descended when V2 approached C2 and increased after the agent passed and moved away. Communication between the vehicle and the data collection point began and retained as long as the relative distance was under the signal radius of the collection point. Each successful V2I communication in Figure 4.22(b) infers a information upload process from the agent to infrastructure.

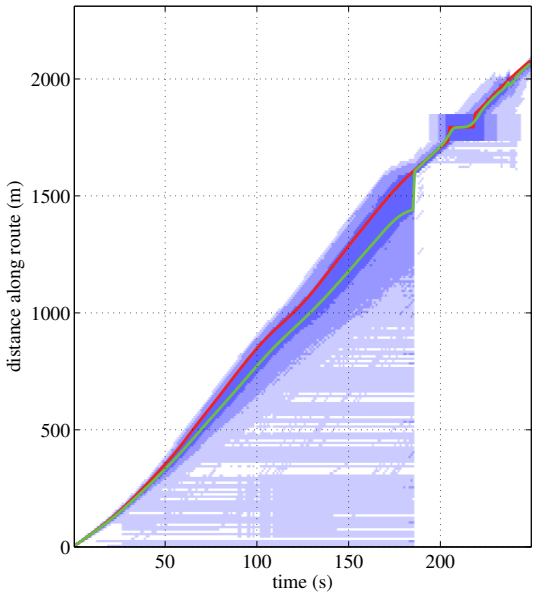
The overall experimental results show that the true vehicle positions were within the 2-sigma confidence bounds estimated by the algorithm for the majority of the time in both the prediction only and tracking scenarios. Figure 4.23(a) illustrates overall



(a) Prediction only

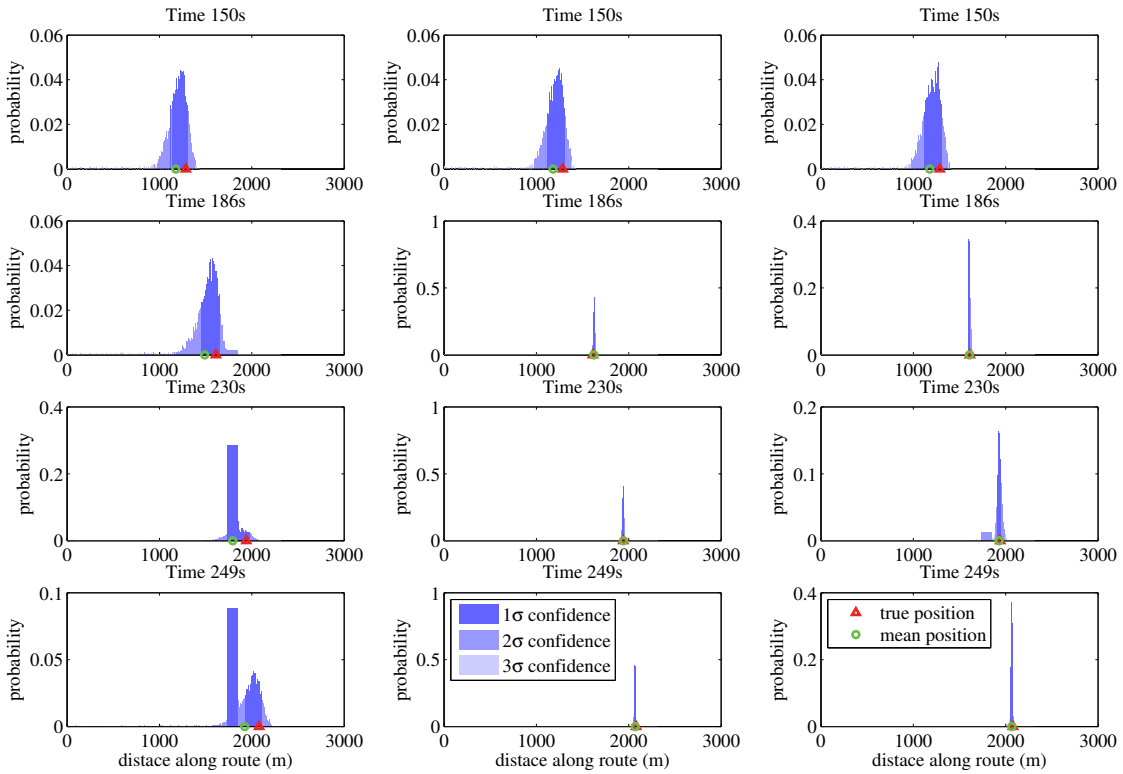


(b) Tracking with GPS



(c) Tracking without GPS

Figure 4.23 – Overall estimation. (a) prediction only, no reduction of uncertainty is observed. (b) tracking with GPS shows accurate estimation when the vehicle was moving under the signal range of the data collection point. (c) tracking with GPS outage suggests reduction of uncertainty at two times when the vehicle entered and left the data collection point's signal circle, but blurring out in between due to lack of GPS information.

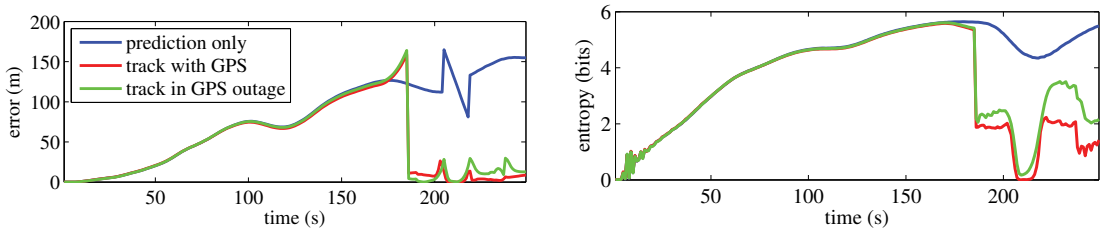


(a) Prediction only (b) Tracking with GPS (c) Tracking without GPS

Figure 4.24 – Estimation details. The estimation results in the three experiments show very slight difference in the first 150 seconds, before the vehicle got close to signal coverage of the data collection point. Nevertheless, uncertainties in tracking with GPS was reduced the most after the vehicle connected to the data collection point, followed by tracking in GPS outage.

results with prediction only, as a comparison to the other two scenarios that have updates from sporadic observations. Details about the vehicle position distribution at different time slots can be found in Figure 4.24. The pattern of the predicted position distribution is characterised by the majority probability mass concentrating on a head followed by a long thin tail, as a result of stopping probability on the road to cover sudden stopping cases. Generally the position uncertainty keeps growing along the route when no observations are included.

Figure 4.23(b) shows the results of the particle filter with sporadic observations. With the data collector located in intersection #205 enabled, the algorithm yields better estimation results as the position uncertainty is reduced when compared to prediction



(a) Estimation error (b) Entropy

Figure 4.25 – Comparison of estimation results. In both (a) and (b), tracking with GPS gives the best performance with respect to estimation error and entropy followed by tracking in GPS outage and case with prediction only.

only case. Though the uncertainty still grows along the road segment #116, it drops dramatically when the vehicle passes close to the intersection #205 and communicates with the data collector. The position uncertainty grows again when the vehicle leaves the data collector signal coverage area. It is important to see that the algorithm is capable of tracking the vehicle during the GPS outage, as shown in Figure 4.23(c). The filter estimates positions of the vehicle using only the positive and negative V2I observations generated due to the proximity to the data collector.

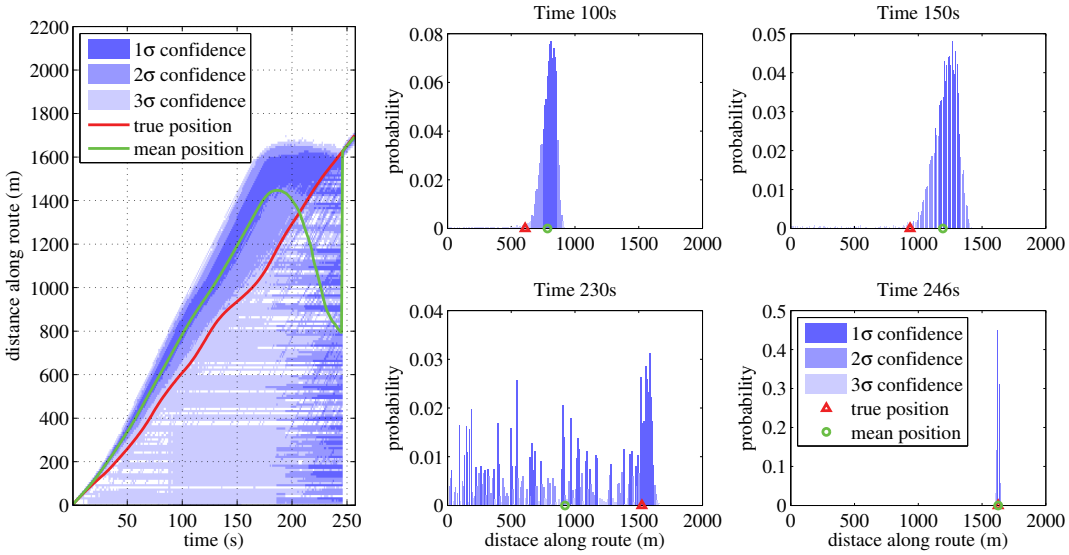
The estimation results from three experiments are also quantitatively summarised in terms of estimation error and entropy presented in Figure 4.25. The distribution of vehicle positions during the first 150 seconds was almost identical in all three experiments as shown in Figure 4.24. The distribution shape in the two tracking cases started to differ from the prediction only case when the vehicle was about to communicate with the data collector around time 188 s. Based on negative connectivity observations from the data collector, the tracking algorithm believed that the vehicle was probably still out of the data collector's communication range. The position probability mass outside the signal coverage of the base station was wiped out when the estimation got updated with information uploaded by the vehicle once it communicated with the data collector. In the meantime, the estimation error dropped, as seen in Figure 4.25(a). Compared to the tracking case without GPS, uncertainty in estimation was reduced further by the algorithm with egocentric GPS observations provided by the vehicle. Entropy in Figure 4.25(b) reflects the degree of estimation

uncertainty and varies accordingly. The entropy grew for the first 180 seconds in all three cases as no effective observation was available to constrain the position estimate. The entropy in the prediction only case then dropped when the estimated position moved close to intersection #205. This phenomenon is reasonable and explainable by the nature of tracking algorithm defined in this thesis. As we do not estimate position of the vehicle inside an area or intersection, the distribution temporarily becomes "more certain" when probability mass accumulates in an area or intersection. However the entropy grew to an even larger value after particles in *Passing Intersections or Areas* state left the intersection #205. This is due to additional motion diversity introduced after particles leave the intersection. When observations were enabled (as in the tracking cases), the entropy collapsed as soon as the vehicle moved into the signal coverage of the data collector at time 188 s. The entropy dropped to near zero when the majority of particles stayed inside the intersection #205 . Entropy started growing again after particles lost connection to the data collector as expected.

It is expected that the tracking with GPS observations outperforms the tracking without GPS case with lower estimation errors and entropies. However, it is demonstrated that the uncertainty can be reduced considerably by using only V2I information. This will be very important to enhance the prediction under all types of GPS outage, such as satellite availability, GPS malfunctions or broken antennas.

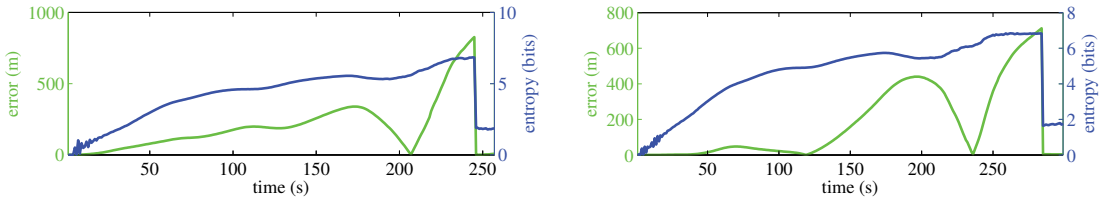
Single Vehicle Cases with Abnormality

Additional experimental results have also shown that positive and negative V2I observations help improve position estimates in unusual cases, such as a vehicle driving much more slowly than the model would suggest, or if a vehicle stops somewhere on the road. Figure 4.26 and Figure 4.28 demonstrate two examples of tracking results with anomalies in a vehicles true trajectory. In the first case, a vehicle was driven at a speed considerably slower than the average. Though a quite large error lies between the mean position and true value, the algorithm obtained consistent prediction before any positive observation was available, as illustrated in Figure 4.26(b). As the motion model generated a faster prediction than true vehicle movement, negative observa-



(a) Overall estimates (b) Details

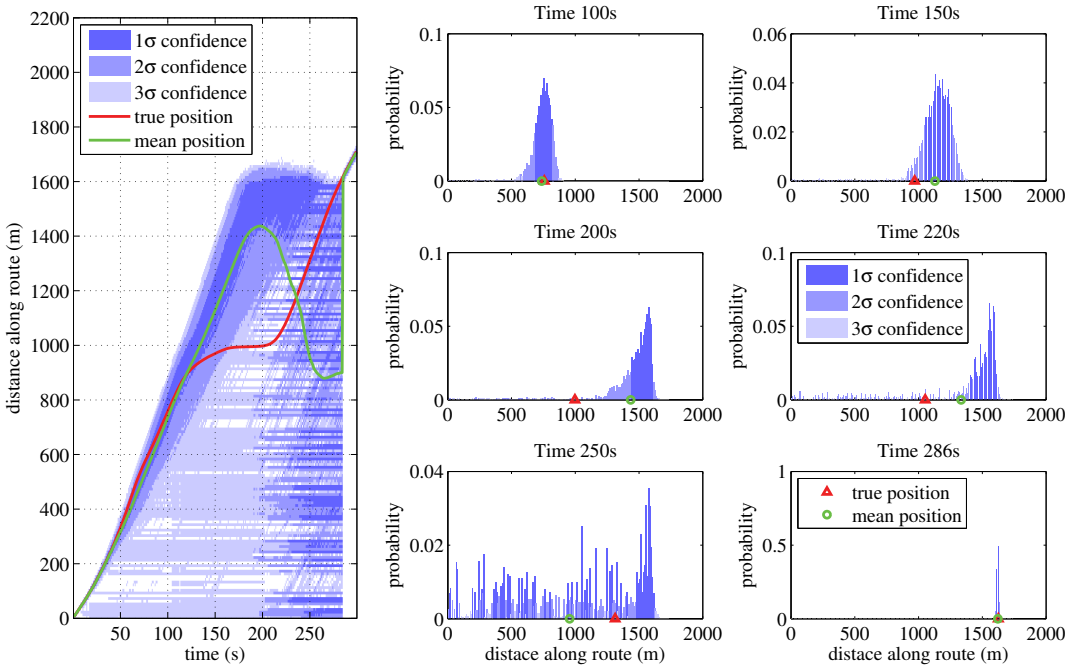
Figure 4.26 – Tracking vehicle with slow driving. a) clearly points out a consistent estimation of the vehicle all the time, though quite inaccurate due to rarely seen slow running from the beginning of the trip. Revealing details in tracking, b) shows an automatic correction process of the algorithm by eliminating the probability distribution head but rising up the tail. The tracking returned to be accurate once the vehicle established a connection to the data collection point.



(a) Tracking vehicle with slow driving (b) Tracking vehicle with stopping on road

Figure 4.27 – Tracking with abnormality on road. In a), estimation error as well as entropy kept growing before the abnormalities in true vehicle motion were corrected by the algorithm when the vehicle got connected to the data collection point. b) tells a same story except for an accurate estimation before the vehicle began to stop on road, at time 125 s.

tions before the vehicle being able to communicate with the data collector gradually eliminated the head part of the position distribution while probability mass of the tail increased. The estimation error and entropy peaked at time 243 s, as shown in Figure 4.27(a), right before the data collector connected to the vehicle. Once the con-



(a) Overall estimates (b) Details

Figure 4.28 – Tracking vehicle with stopping on road. Estimation was close to true trajectory before the vehicle began to stop as seen in a). Detailed process is given in b) to show how the algorithm dealt with the abnormal stopping case. During the stopping period, the algorithm adapted to the abnormality by rising up the tail part of the estimation distribution. The shape of the tail is close to a uniform distribution as the vehicle could be stopping anywhere behind the distribution head without additional information available. The tracking error was then reset with egocentric data uploaded when the vehicle ran into the signal circle of the data collection point.

nection was established, the algorithm obtained accurate position estimation using the egocentric observations collected from the vehicle.

In the second example, a vehicle stopped unexpectedly on road segment #116 at time 160 s for about 50 seconds. The mean estimation kept close to true trajectory in first 120 seconds. Taking the stopping probability on the vehicle into consideration, the algorithm included the possibility of the vehicle anomaly into the tail of the position estimate, and the tail increased during the stopping period of the vehicle. The uncertainty became almost uniform after the main position probability mass was removed due to negative information. This meant that the vehicle did not show up

when expected to the data collector, so the vehicle was then considered most likely to be stopped at some point along the road. Figure 4.28(b) shows that the estimation remained consistent, though the error began to increase with time. The estimation error was then corrected in the filter when the vehicle resumed moving and passed close to the data collection point located in intersection #205. This is seen as a steep error drop at time 284 s in Figure 4.27(b).

4.5 Summary

This chapter presents algorithms for long-term vehicle motion prediction and tracking based on a multiple-model approach that can be applied to large areas with very sparse vehicle communication. The algorithms incorporates probabilistic vehicle models and properties of the working environment such as roads, intersections and special areas. The prediction algorithm evaluates the vehicle position for a long period by using ASPs built for the particular environment, and considering vehicle stopping probability. It uses a limited number of data collection points distributed around the field to update vehicle position estimates when in range and then predict vehicle positions at points in between. The concept of using positive and negative information from the data collection points to improve the position estimate is introduced. A particle filter is adopted for global position estimates using both positive and negative information in the fusion stage.

The experimental results show that it is possible to obtain consistent prediction of vehicle position for long periods of time. A limited number of data collection points enable vehicle tracking without full network coverage. Considering only V2I information in update stage, the algorithm demonstrated to have the capability to track vehicles with GPS outage, reducing the uncertainty of the estimate even in case of GPS malfunctions. In addition, the algorithm is capable of coping with anomalies in a vehicle's true movement. The experimental results have shown that the position estimate in the case where a vehicle moves much slowly or unexpectedly stops on a road remains consistent. The anomalies are accounted for in the filter through the

incorporation of a stopping probability within the vehicle model.

The next chapter is addressing the prediction and tracking of multiple vehicles, and considering the interactions of all vehicles. This approach can significantly reduce the uncertainty of the whole fleet, since each mobile vehicle can potentially bring position estimates for other vehicles to a collection point.

Chapter 5

Using Delayed Observations for Vehicle Tracking

5.1 Introduction

This chapter extends the work in the previous chapter toward long-term multi-vehicle motion prediction and tracking. It introduces the concept of observation harvesting, where P2P communication between vehicles allows position updates to be exchanged and brought forward to any of the fixed data collection points, which then connect through to the centralised base station. New algorithms are introduced to incorporate the delayed information into the global position estimate when a vehicle brings back information to a data collection point. The remainder of the chapter is organised as follows. Following the observation harvesting mechanism introduced in Section 5.2, Section 5.3 proposes a probabilistic tracking algorithm to fuse the delayed information, as well as the implementation using a particle filter. Section 5.4 presents experimental results of tracking multiple vehicles followed by discussions. Finally, Section 5.5 summarises the work in this chapter.

5.2 Observation Harvesting

Observation harvesting refers to the mechanism where vehicles returning to communication range of fixed infrastructure (data collection points) provide delayed observations of other vehicles that were encountered during a trip. The centralised base station will update the estimates of vehicle position using direct observations from the vehicles that are currently in communication range (V2I), as well as the egocentric observations collected from interactions between vehicles (V2V) in other areas of the site that do not have direct communication with the base station. The observation harvesting mechanism involves the collection of the second hand (or third, fourth, etc.) information through V2V communication.

With observation harvesting, a direct connection between a vehicle and a fixed data collection point is not required to update the position estimate. Each vehicle that returns to a fixed data collector acts as an information carrier for the vehicles that do not return. In addition, when a vehicle is not detected by a data collection point it is considered negative information. This assists in constraining the possible vehicle location in the absence of new observations.

5.2.1 Observation Harvesting Algorithm Overview

The proposed observation harvesting algorithm allows egocentric position updates from vehicles, with some time delays, to be shared and harvested by the base station. The most informative information is kept in every vehicle and synchronised when a pair of vehicles are in communication range. When a vehicle passes a data collection point (fixed infrastructure), observations collected by the vehicle are downloaded and used to track all vehicles. Within the base station, real-time V2I observations generated by data collectors distributed in the field are also used in tracking process. An overview of the observation harvesting algorithm can be seen as Figure 5.1.

To summarise, the information conveyed in every vehicle is composed of:

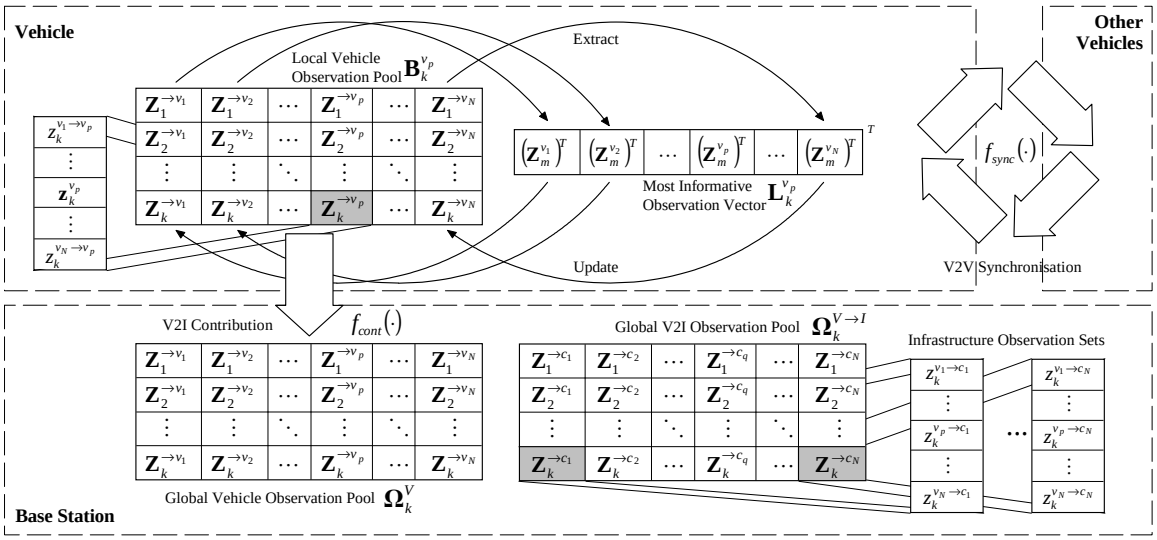


Figure 5.1 – An Overview of the observation harvesting algorithm.

1. $\mathbf{B}_k^{v_p}$: Local Vehicle Observation Pool, which is augmented over time with new observations based on new sensor information.

2. $\mathbf{L}_k^{v_p}$: Most Informative Observation Vector (MIOV), which refers to an aggregation of most informative observations about the vehicle itself and others. The MIOV information in a vehicle is extracted at every new time step from its local egocentric observation pool.

3. $T_{cont}^{v_p \to I}$: Last Contribution Time.

The base station keeps the information harvested in the following form.

1. $\mathbf{\Omega}_k^V$: Global Vehicle Observation Pool, where observations collected (harvested) from all N_v vehicles in the field are merged together.

2. $\mathbf{\Omega}_k^{V \to I}$: Global V2I Observation Pool, which keeps real-time observations from all N_c data collectors distributed in the field observing each vehicle.

The detailed description of the algorithm is presented in the remainder of this section.

The proposed observation harvesting algorithm with MIOV is capable of operating in a low bandwidth P2P network. It provides a simple but effective way to gain the

best position knowledge of all vehicles running in the field. To prevent overloading the available communication bandwidth, only the most informative observations are exchanged in a data synchronisation process between a pair of vehicles. Among all observation sets with absolute observations available, the most informative observation usually has the time stamp closest to the present time.

5.2.2 Local Vehicle Observation Pool

Observations collected by a vehicle at each time step are put into the local vehicle observation pool, which is a time-growing joint observation matrix. These observations are either ones measured directly by the vehicle (e.g. absolute GPS data, inter-vehicle relative measurements), or indirect ones acquired from other vehicles through V2V communication.

$$\mathbf{B}_k^{v_p} = \begin{bmatrix} \mathbf{Z}_1^{\rightarrow v_1} & \mathbf{Z}_1^{\rightarrow v_2} & \cdots & \mathbf{Z}_1^{\rightarrow v_p} & \cdots & \mathbf{Z}_1^{\rightarrow v_N} \\ \mathbf{Z}_2^{\rightarrow v_1} & \mathbf{Z}_2^{\rightarrow v_2} & \cdots & \mathbf{Z}_2^{\rightarrow v_p} & \cdots & \mathbf{Z}_2^{\rightarrow v_N} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ \mathbf{Z}_k^{\rightarrow v_1} & \mathbf{Z}_k^{\rightarrow v_2} & \cdots & \mathbf{Z}_k^{\rightarrow v_p} & \cdots & \mathbf{Z}_k^{\rightarrow v_N} \end{bmatrix} \tag{5.1}$$

At every time step, a vehicle generates a local vehicle observation set, describing all observations it records at this time step. The observation set $\mathbf{Z}_k^{\rightarrow v_p}$ for vehicle v_p at time k contains the absolute egocentric observation and the relative measurements to every other vehicles at that time slot.

$$\mathbf{Z}_k^{\rightarrow v_p} = \begin{bmatrix} z_k^{v_1 \rightarrow v_p} & z_k^{v_2 \rightarrow v_p} & \cdots & \left(\mathbf{z}_k^{v_p}\right)^T & \cdots & z_k^{v_N \rightarrow v_p} \end{bmatrix}^T$$

where

$\mathbf{z}_k^{v_p}$ is the absolute egocentric observation.

$z_k^{v_q \rightarrow v_p}$ $\forall q \neq p$ are the inter-vehicle relative observations.

The local egocentric observation pool is then augmented with the local observation set generated for a new time step. The local vehicle observation set is added into

the bottom of the v_p^{th} row in Equation (5.1), while the rest of rows are reserved for observation sets harvested from other vehicles.

5.2.3 Global Vehicle and V2I Observation Pools

The global vehicle and V2I observation pools are maintained by the base station only. Both of them grow size with time. The global vehicle observation pool is of the same structure as the local vehicle observation pool, but with all observations harvested from vehicles in the field merged together. The global vehicle observation pool keeping information about all N_v vehicles at time k is represented by:

$$\mathbf{\Omega}_k^V = \begin{bmatrix} \mathbf{Z}_1^{\rightarrow v_1} & \mathbf{Z}_1^{\rightarrow v_2} & \cdots & \mathbf{Z}_1^{\rightarrow v_N} \\ \mathbf{Z}_2^{\rightarrow v_1} & \mathbf{Z}_2^{\rightarrow v_2} & \cdots & \mathbf{Z}_2^{\rightarrow v_N} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{Z}_k^{\rightarrow v_1} & \mathbf{Z}_k^{\rightarrow v_2} & \cdots & \mathbf{Z}_k^{\rightarrow v_N} \end{bmatrix}$$

The global V2I observation pool records observations of all N_c data collection points distributed in the field observing N_v vehicles. The V2I observation pool at time k is written as:

$$\mathbf{\Omega}_k^{V\rightarrow I} = \begin{bmatrix} \mathbf{Z}_1^{\rightarrow c_1} & \mathbf{Z}_1^{\rightarrow c_2} & \cdots & \mathbf{Z}_1^{\rightarrow c_N} \\ \mathbf{Z}_2^{\rightarrow c_1} & \mathbf{Z}_2^{\rightarrow c_2} & \cdots & \mathbf{Z}_2^{\rightarrow c_N} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{Z}_k^{\rightarrow c_1} & \mathbf{Z}_k^{\rightarrow c_2} & \cdots & \mathbf{Z}_k^{\rightarrow c_N} \end{bmatrix}$$

Suppose there are N_c data collectors connected to base station as infrastructure points. Each of the data collectors generates a V2I observation set at each time step, containing real-time V2I observations it measures about all N_v vehicles at this time step.

$$\mathbf{Z}_k^{\rightarrow c_q} = \begin{bmatrix} z_k^{v_1\rightarrow c_q} & \cdots & z_k^{v_p\rightarrow c_q} & \cdots & z_k^{v_N\rightarrow c_q} \end{bmatrix}^T$$

Both global observation pools are augmented with new observations received at a new time step. Note that the new generated V2I observations are available to the global V2I observation pool immediately, while the global vehicle observation pool receives vehicle observations with some time delay.

5.2.4 Most Informative Observation Vector

The most informative observation vector (MIOV) is a vector composed by the most informative observation sets inside the local vehicle observation pool. Each vehicle node keeps the vector up-to-date at every time step and exchanges this information when it meets another vehicle. The MIOV maintained by vehicle node v_p at time k is written as:

$$\mathbf{L}_k^{v_p} = \left[(\mathbf{Z}_m^{v_1})^T \quad (\mathbf{Z}_m^{v_2})^T \quad \cdots \quad (\mathbf{Z}_m^{v_p})^T \quad \cdots \quad (\mathbf{Z}_m^{v_N})^T \right]^T$$

where each element is the most informative observation set (MIOS) about a vehicle chosen from the vehicle observation sets within local vehicle observation pool.

In vehicle v_p, the up-to-date MIOSs are extracted for each vehicle (including the vehicle itself) at every new time step. A function $f_{mios}(\cdot)$ is used here to obtain the MIOS about a vehicle v_q at the present time k.

$$\mathbf{Z}_m^{v_q} = f_{mios}\left(\mathbf{Z}_{1:k}^{\rightarrow v_q}\right)$$

The MIOS for the vehicle v_q is chosen to be the set that can best constrain the position estimate of the vehicle, among all observation sets available to vehicle v_p. Usually, it is the last known observation set that was generated by vehicle v_q, and is stored in the pool of vehicle v_p. When this last known information is not available to v_p, which is the case that the local vehicle observation pool of vehicle v_p contains no information from vehicle v_q at all, the returned MIOS for this vehicle is an empty set. The MIOS about the vehicle v_p itself is the new vehicle observation set generated at the present time.

5.2.5 Synchronisation Process

Synchronisation process of the MIOV occurs when two vehicles meet. This process involves bidirectional exchange of data between the two vehicles. After this process, the MIOVs in each of the two vehicles are expected to be identical.

Between two vehicles v_p and v_q, the whole synchronisation process $f_{sync}(\cdot)$ is as described in Table 5.1. The process begins with extracting and comparing time profiles of MIOVs (defined as $\mathbf{T}_k^{v_p}$, $\mathbf{T}_k^{v_q}$) of the two nodes. In order to minimise bandwidth, only information in the MIOV with a newer time stamp is transferred from one node to the other. This essentially means that each vehicle shares the best information they have about all other vehicles they have been in communication with. After the synchronisation process, the local vehicle observation pool is updated with the new information received, which is defined as back contribution of MIOV. A back contribution function $f_{bkct}(\cdot)$ is used to realise the mechanism in vehicle v_p.

$$\mathbf{B}_{k+}^{v_p} = f_{bkct}\left(\mathbf{B}_k^{v_p}, \mathbf{L}_{k+}^{v_p}\right)$$

The amount of data transferred in the synchronisation process is no larger than the size of MIOV, i.e. N_v absolute observations. With only a small amount of data exchanged, the synchronisation process provides a simple, short and effective method to disseminate the most informative information among all vehicles in the field.

5.2.6 Observation Contribution Process

Define a contribution process, whereby each vehicle uploads (contributes) observations in the local vehicle observation pool to the base station when in communication with a fixed data collection point. The data transmission in the process is unidirectional. A description on the contribution process $f_{cont}(\cdot)$ could be found in Table 5.2.

The last contribution time $T_{cont}^{v_p \to I}$ is kept in each vehicle to minimise the communication bandwidth requirements. When the process occurs between a vehicle v_p and the

Table 5.1 – Algorithm: Synchronisation process

$\mathbf{L}_{k+}^{v_p} = \mathbf{L}_{k+}^{v_q} \leftarrow f_{sync}\left(\mathbf{L}_k^{v_p}, \mathbf{L}_k^{v_q}\right)$	
1:	extract $\mathbf{T}_k^{v_p}$ from $\mathbf{L}_k^{v_p}$
2:	extract $\mathbf{T}_k^{v_q}$ from $\mathbf{L}_k^{v_q}$
3:	**for** $i = 1$ to N_v **do**
4:	**if** $T_{k,v_i}^{v_p}$ in $\mathbf{T}_k^{v_p} > T_{k,v_i}^{v_q}$ in $\mathbf{T}_k^{v_q}$ **do**
5:	replace $\mathbf{Z}_m^{v_i}$ in $\mathbf{L}_k^{v_q}$ by $\mathbf{Z}_m^{v_i}$ in $\mathbf{L}_k^{v_p}$
6:	**elseif** $T_{k,v_i}^{v_q}$ in $\mathbf{T}_k^{v_q} > T_{k,v_i}^{v_p}$ in $\mathbf{T}_k^{v_p}$ **do**
7:	replace $\mathbf{Z}_m^{v_i}$ in $\mathbf{L}_k^{v_p}$ by $\mathbf{Z}_m^{v_i}$ in $\mathbf{L}_k^{v_q}$
8:	**end if**
9:	**end for**
10:	$\mathbf{L}_{k+}^{v_p} = \mathbf{L}_k^{v_p}$ and $\mathbf{L}_{k+}^{v_q} = \mathbf{L}_k^{v_q}$

Table 5.2 – Algorithm: Contribution process

$\mathbf{\Omega}_{k+}^V, T_{cont}^{v_p \to I} \leftarrow f_{cont}\left(\mathbf{\Omega}_k^V, \mathbf{B}_k^{v_p}, T_{cont}^{v_p \to I}\right)$	
1:	**for** $t = T_{cont}^{v_p \to I} + 1$ to k **do**
2:	**for** $i = 1$ to N_v **do**
3:	vehicle v_p transmits $\mathbf{Z}_t^{v_i}$ in $\mathbf{B}_k^{v_p}$ to $\mathbf{\Omega}_k^V$
4:	**end for**
5:	**end for**
6:	$T_{cont}^{v_p \to I} = k$
7:	$\mathbf{\Omega}_{k+}^V = \mathbf{\Omega}_k^V$

base station, only information in the local vehicle observation pool of vehicle v_p with a time stamp later than the last contribution time $T_{cont}^{v_p \to I}$ is transferred to the global vehicle observation pool. After the contribution process, the last contribution time is updated to the present time.

5.2.7 Cost Analysis of Communication Bandwidth

Define:

1. $\bar{\tau}^{V \to V}$ as the average communication time interval between vehicles.

2. $\tau^{v_p \to I}$ and $\bar{\tau}^{v_p \to I}$ as the communication time interval between a vehicle v_p and

the infrastructure, and its average value respectively.

3. Interval ratio $\eta = \frac{\bar{\tau}^{v_p \to I}}{\bar{\tau}^{V \to V}}$, which means that on average the vehicle v_p meets another particular vehicle η times during the time interval $\bar{\tau}^{v_p \to I}$.

Table 5.3 – Maximum and mean bandwidth costs on P2P communication

Obs. Type		Quantity of Obs. Transmitted/Received	
		Maximum	Mean
V2V	Absolute	$N_v - 1$	$N_v/2$
	Relative	$(N_v - 1)^2$	$N_v (N_v - 1)/2$
	Total	$N_v (N_v - 1)$	$N_v^2/2$
V2I	Absolute	$N_v \tau^{v_p \to I}$	$\bar{\tau}^{v_p \to I} + (N_v - 1)\eta$
	Relative	$N_v (N_v - 1) \tau^{v_p \to I}$	$(N_v - 1)\bar{\tau}^{v_p \to I} + (N_v - 1)^2 \eta$
	Total	$N_v^2 \tau^{v_p \to I}$	$N_v \bar{\tau}^{v_p \to I} + N_v (N_v - 1)\eta$

Regardless of the communication intervals, the maximum inter-vehicle communication bandwidth is constant with a value of $N_v - 1$ for absolute observations and $(N_v - 1)^2$ for relative observations in a single V2V transaction. Table 5.3 also suggests that the maximum V2I bandwidth is proportional to the time elapsed since the last contribution time. Increasing either the quantity of data collection points or the communication range would reduce the V2I communication interval as well as requirements for V2I communication bandwidth. Alternatively, optimisation on the bandwidth cost could be achieved by introducing a sliding time window into the filter, which is elaborated in Section 5.3.3.

5.3 Tracking with Delayed Observations

This section mainly focuses on the tracking of vehicles using absolute referenced observations from data collection points, and absolute vehicle egocentric observations harvested by the base station. Please refer to **Chapter 3** for the topics on V2I proximity detection and egocentric observations from vehicles.

Note that though vehicles communicate with each other, V2V relative position measurements are not yet considered in the fusion stage of the tracking approach proposed in this section. This is because it requires the consideration of cross-correlations between vehicles. Therefore, the relative observations are not transmitted in the observation harvesting. Those inter-vehicle relative measurements would be considered in **Chapter 7**, where a delayed-state cooperative tracking approach is proposed.

5.3.1 Bayesian Estimation

As has been constructed in Section 3.6, the likelihood function given an absolute referenced observation about a vehicle v_p from a data collector c_q at time t is represented by $\Psi_t^{v_p \to c_q}$, while the likelihood given an absolute egocentric observation for a vehicle v_p is written as $\widetilde{\Psi}_t^{v_p}$.

Traditional approaches based on sequential Bayesian estimation only consider observations conditional on the present state. They are therefore of low performance in tracking problems with delayed observations. Instead, the filtering algorithm adopted in this chapter maintains the full history of vehicle states. With historical states maintained in the filter, information from delayed observations is able to be fused.

The following definitions are made.

1. Define $\mathbf{\Omega}_{k|k-1}^{v_p \to I}$ as a set of new real-time V2I observations from the infrastructure at time k.

2. $\mathbf{\Omega}_{k|k-1}^{v_p}$ is defined to be a collection of the delayed egocentric observations of vehicle v_p most recently received by the base station at time k.

Information contained in these two observation sets will be fused into the filter in its update stage. They are obtained by relative complement operations of global observation pools at two successive time steps. Respectively, we have:

$$\begin{aligned}\boldsymbol{\Omega}_{k|k-1}^{v_p \to I} &= \boldsymbol{\Omega}_{k}^{v_p \to I} \setminus \boldsymbol{\Omega}_{k-1}^{v_p \to I} \\ &= \left[z_k^{v_p \to c_1} \ \cdots \ z_k^{v_p \to c_q} \ \cdots \ z_k^{v_p \to c_N} \right]^T\end{aligned}$$

$$\boldsymbol{\Omega}_{k|k-1}^{v_p} = \boldsymbol{\Omega}_{k}^{v_p} \setminus \boldsymbol{\Omega}_{k-1}^{v_p}$$

where $\boldsymbol{\Omega}_k^{v_p \to I}$ holds all relative vehicle v_p to infrastructure observations up to time k in the global V2I observation pool $\boldsymbol{\Omega}_k^{V \to I}$, while $\boldsymbol{\Omega}_k^{v_p}$ keeps every egocentric observation of v_p up to time k, which are stored in the global egocentric observation pool $\boldsymbol{\Omega}_k^{V}$. Note that $\boldsymbol{\Omega}_{k|k-1}^{v_p}$ could be empty.

To track vehicles with delayed egocentric observations, the filter keeps full historical state information for each vehicle. At time k, the full states of vehicle v_p is represented as:

$$P\left(\mathbf{x}_{0:k}^{v_p}, S_{0:k}^{v_p} | \boldsymbol{\Omega}_k^{v_p \to I}, \boldsymbol{\Omega}_k^{v_p}\right)$$

The prediction stage presented in Equation (4.5) in Section 4.3.4 and the update/fusion stage can now be combined together to yield:

$$\begin{aligned}&P\left(\mathbf{x}_{0:k}^{v_p}, S_{0:k}^{v_p} | \boldsymbol{\Omega}_k^{v_p \to I}, \boldsymbol{\Omega}_k^{v_p}\right) \\ &\propto \left(\prod_{q=1}^{N_c} \Psi_k^{v_p \to c_q} \right) \left(\prod_{\mathbf{z}_t^{v_p} \in \boldsymbol{\Omega}_{k|k-1}^{v_p}} \widetilde{\Psi}_t^{v_p} \right) \\ &\times \sum_{i=1}^{3} \sum_{j=1}^{3} \int Pm_{i \to j} Ps_{i \to j}(k) \\ &\times P\left(\mathbf{x}_{0:k-1}^{v_p}, S_{0:k-1}^{v_p} | \boldsymbol{\Omega}_{k-1}^{v_p \to I}, \boldsymbol{\Omega}_{k-1}^{v_p}\right)\end{aligned} \tag{5.2}$$

5.3.2 Particle Filtering Algorithm

In the proposed algorithm, each vehicle running in the field is tracked by a separate particle filter maintained on a central base station, which uses $k+1$ sets of particles to keep track of the complete history states of the vehicle up to time k.

Define a particle base keeping $k+1$ collections of particles which correspond to $k+1$ states of the vehicle v_p.

$$\Theta_k^{v_p} = \begin{bmatrix} \{\mathbf{x}_0^i, S_0^i, w_0^i\}_{i=1}^L \sim P\left(\mathbf{x}_0^{v_p}, S_0^{v_p}\right) \\ \{\mathbf{x}_1^i, S_1^i, w_1^i\}_{i=1}^L \sim P\left(\mathbf{x}_1^{v_p}, S_1^{v_p} | \mathbf{\Omega}_1^{v_p \to I}, \mathbf{\Omega}_1^{v_p}\right) \\ \vdots \\ \{\mathbf{x}_k^i, S_k^i, w_k^i\}_{i=1}^L \sim P\left(\mathbf{x}_k^{v_p}, S_k^{v_p} | \mathbf{\Omega}_k^{v_p \to I}, \mathbf{\Omega}_k^{v_p}\right) \end{bmatrix}$$

where $\mathbf{x}_k^i$ is further factorised to $\mathbf{x}_k^i = \begin{bmatrix} (\mathbf{r}_k^i)^T & v_k^i & \tau_k^i \end{bmatrix}^T$.

Information in both the global V2I observation pool and global egocentric observation pool is fused in the filter in a non-synchronous mode. Without the availability of egocentric observations received from vehicles, the filtering continues with prediction and updates with real-time V2I measurements from fixed data collection points. The filtering is then restarted from a historical state on arrival of delayed observations. The earliest time stamp T_s in the new received egocentric observations $\mathbf{\Omega}_{k|k-1}^{v_p}$ at time k determines the time from which the particle filter restarts.

The particle filtering algorithm is initialised by drawing L particles to represent the initial state.

$$\Theta_0^{v_p} = \left[\{\mathbf{x}_0^i, S_0^i, w_0^i\}_{i=1}^L \sim P\left(\mathbf{x}_0^{v_p}, S_0^{v_p}\right)\right]$$

The algorithm for each vehicle is presented in Table 5.4 in the form of a pseudocode. To prevent weight degeneracy, resampling is adopted in the algorithm when the effective particles quantity $\widehat{N_{eff}}$ is below a threshold N_{thr}.

Table 5.4 – Algorithm: Particle filtering with delayed observations

$\Theta_k^{v_p} \leftarrow Particle\_Filter\left(\Theta_{k-1}^{v_p}, \mathbf{\Omega}_{k-1}^{v_p}, \mathbf{\Omega}_k^{v_p}, \mathbf{\Omega}_k^{v_p \to I}\right)$

calculate $\mathbf{\Omega}_{k|k-1}^{v_p} = \mathbf{\Omega}_k^{v_p} \setminus \mathbf{\Omega}_{k-1}^{v_p}$
if $\mathbf{\Omega}_{k|k-1}^{v_p}$ is not empty **do**
  $T_s = f_{min}\left(\mathbf{\Omega}_{k|k-1}^{v_p}\right)$
else do
  $T_s = k$
end if
load particles $\{\mathbf{x}_{T_s-1}^i, S_{T_s-1}^i, w_{T_s-1}^i\}_{i=1}^L$ from $\Theta_{k-1}^{v_p}$
for $t = T_s$ to k **do**
  for $i = 1$ to L **do**
    propagation: draw $S_t^i \sim P\left(S_t | S_{t-1}^i, \xi_{t-1}^i\right)$
      draw $\mathbf{x}_t^i \sim P\left(\mathbf{x}_t | \mathbf{x}_{t-1}^i, S_t^i, S_{t-1}^i\right)$
    update weight with V2I observations:
      $w_t^i = \left(\prod_{q=1}^{N_c} \Psi_t^{v_p \to c_q}\right) w_{t-1}^i$
    update weight with egocentric observation:
      $w_t^i = \widetilde{\Psi}_t^{v_p} w_t^i$ if $\widetilde{\Psi}_t^{v_p}$ exists
  end for
  normalise weights $\{w_t^i\}_{i=1}^L$
  if $\widehat{N_{eff}} < N_{thr}$ **do**
    resample with replacement L particles from $\{\mathbf{x}_t^i, S_t^i, w_t^i\}_{i=1}^L$ according to $\{w_t^i\}_{i=1}^L$
  end if
  replace $\{\mathbf{x}_t^i, S_t^i, w_t^i\}_{i=1}^L$ in $\Theta_{k-1}^{v_p}$ if $t < k$
end for
$\Theta_k^{v_p} \leftarrow \begin{bmatrix} \Theta_{k-1}^{v_p} \\ \{\mathbf{x}_k^i, S_k^i, w_k^i\}_{i=1}^L \end{bmatrix}$

5.3.3 Computational Complexity and Optimisation

The proposed particle filter is of higher computational cost than conventional sequential approaches, which have a time complexity of $O(L)$ for each vehicle in each iteration. It has been found that the motion prediction step in the proposed approach is the major contributor of the computational burden as it deals with complicated transitions of vehicle states and multiple vehicle models. Consequently, the computational complexity mainly depends on the number of steps the filter rewinds back

during an iteration. In the worst case the proposed particle filter restarts from the last contribution time $T_{cont}^{v_p \to I}$. With $\tau^{v_p \to I}$ previously defined as the time elapsed since the last contribution time of a vehicle v_p, T_s in Table 5.4 ranges from $k - \tau^{v_p \to I} + 1$ to the present time k, which means that the time complexity of the proposal particle filter is no more than $O\left(\tau^{v_p \to I} L\right)$ for the vehicle. The actual time complexity varies depending upon the V2V and V2I communication activities in the field, see Section 5.4.4 for detailed discussions.

The algorithm can be optimised by introducing a time sliding window with which only historical states within T_w seconds before the present time k are kept. With T_s then constrained by $[k - T_w + 1, k]$, the maximum time complexity in an iteration is reduced to $O(\tau_m L)$ where $\tau_m = min\left(\tau^{v_p \to I}, T_w\right)$. Therefore, by setting the maximum allowable time delay for vehicle observations in the use of the time sliding window, the computational cost of rerunning the filter is bounded. As observations with delay time larger than the time window T_w would be discarded, the choice of the time window length is essentially a trade-off between estimation accuracy and computational cost.

The communication bandwidth cost between vehicles and infrastructure will also be bounded by introducing the time sliding window, as observations out of the window are not transmitted. Consequently, the maximum amount of observations transmitted or received in a single V2I transaction will be reduced to $N_v \tau_m$.

5.4 Experimental Validation

5.4.1 Experiment Setup

Data from a working mine operation (see Figure 4.12 in Section 4.4) was used to demonstrate the vehicle tracking algorithm presented. The same vehicle motion prediction model built in that section was adopted in the experiment. During the experiment, the wireless transceiver mounted on each vehicle was enabled so that the vehicles could communicate when in proximity.

5.4.2 Entropy of Target Agents

The entropy for a single mobile agent has been introduced in Section 4.4.2, with its calculation presented in Equation (4.7). When analysing multiple agents cases (say, N_v vehicles), fleet entropy is used by adding up the entropy of each agent in the fleet. A brief description of the fleet entropy could also be found in [3].

$$H\left(\mathbf{X}_k\right) = \sum_{p=1}^{N_v} H\left(\mathbf{x}_k^{v_p}\right)$$

where $\mathbf{X}_k = \left[\left(\mathbf{x}_k^{v_1}\right)^T \quad \cdots \quad \left(\mathbf{x}_k^{v_p}\right)^T \quad \cdots \quad \left(\mathbf{x}_k^{v_N}\right)^T\right]^T$.

5.4.3 Experiment Results

Although the algorithm was run with a number of vehicles, a subset of them was used to show the results in a more clear manner. In this case three vehicles (V1, V2 and V3) were selected. They were moving in the area of operation with two fixed data collection points (C1 and C2) installed, see Figure 5.2 for the detailed layout and routes on the field map. According to Figure 5.3 which depicts the relative distance and interactions between two vehicles, events at some important time points are summarised in chronological order.

- $T_{V1\rightarrow C1}$: The last comm. time between V1 and C1.
- $T_{V2\rightarrow C2}$: The last comm. time between V2 and C2.
- $T_{V1\rightarrow V3}$: The last time V3 observed V1.
- $T_{V3\rightarrow V1}$: The last time V1 observed V3.
- $T_{V3\rightarrow C1}$: The first comm. time between V3 and C1.
- $T_{V1\rightarrow V2}$: The last time V2 observed V1.

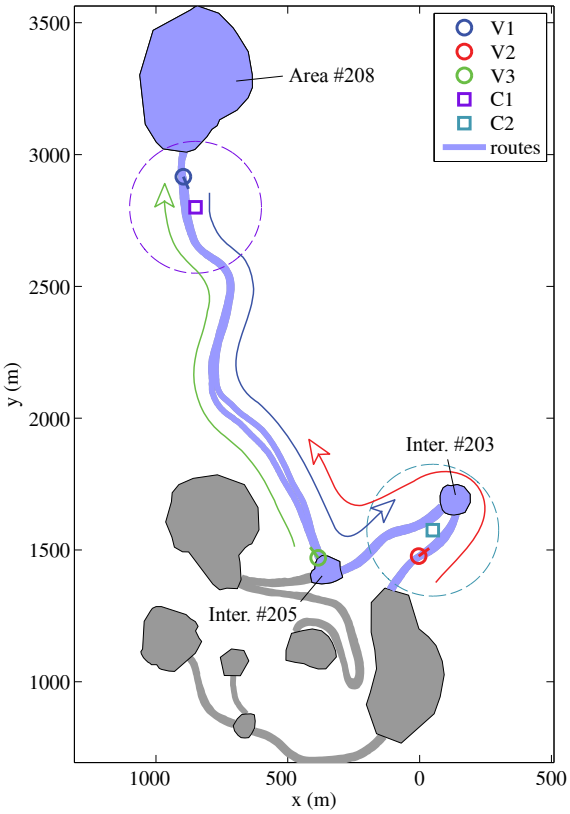
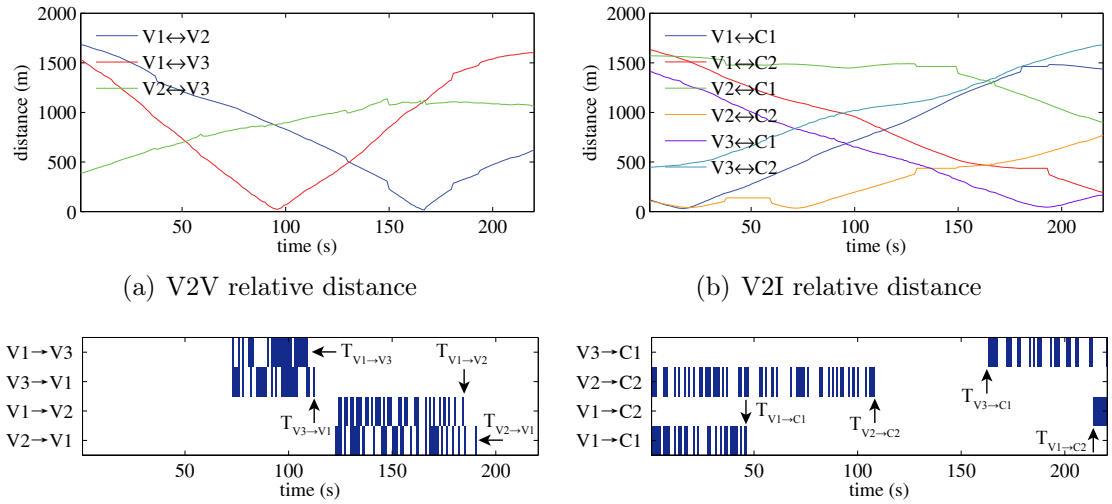


Figure 5.2 – Vehicles routes on field map. Three vehicles were running in a field with two data collection points installed near intersection #203 and area #208. Each data collection point established a signal circle with a radius of 250 metres. The vehicles were operating between intersections #203, #205, and area #208. V1 began a trip on the road near area #208, V2 began on a road adjacent to intersection #203, and V3 began near intersection #205.

- $T_{V2\to V1}$: The last time V1 observed V2.
- $T_{V1\to C2}$: The first comm. time between V1 and C2.

To describe the sequence of events, V1 left the communication range of C1 at time $T_{V1\to C1}$ and drove south, while V2 left C2 at time $T_{V2\to C2}$ and drove north. V1 passed V3 followed by V2 during this trip. V3 drove to C1 at time $T_{V3\to C1}$, after having passed V1 and collected its most informative position information up to time $T_{V1\to V3}$ through V2V synchronisation. The data collector C1 captured the time delayed position update from V3 and the base station used this to improve the estimate of the position of V1. This is illustrated in Figure 5.4(b) at the time marked $T_{V3\to C1}$.



(a) V2V relative distance

(b) V2I relative distance

(c) V2V communication activity

(d) V2I communication activity

Figure 5.3 – Interactions in V2I and V2V. (a) and (b) reveal change of relative distance in each of three vehicles pairs with time, and between vehicles and data collection points respectively. Troughs in distance lines represent physical proximity points of two nodes. In (c) and (d) blue denotes the successful communication between nodes was achieved and blank otherwise. More specifically, $T_{V3\rightarrow V1}$ is found to be 111 s, $T_{V1\rightarrow V3}$ to be 108 s, $T_{V1\rightarrow V2} = 183\ s$ and $T_{V2\rightarrow V1} = 189\ s$ in (c). For V2I communication in (d), $T_{V1\rightarrow C1} = 45\ s$, $T_{V2\rightarrow C2} = 107\ s$, whereas $T_{V3\rightarrow C1} = 162\ s$ and $T_{V1\rightarrow C2} = 213\ s$.

Similarly, V1 provided the base station with its egocentric observations and a position update for V2 after arriving at C2 at time $T_{V1\rightarrow C2}$. The position information from V2 that was harvested by V1 resulted in an improvement in the position estimate of V2 as illustrated in Figure 5.5(b) at the time marked $T_{V1\rightarrow C2}$.

Tracking results without V2V communication enabled are shown for comparison in Figures 5.4(a) and 5.5(a). These figures demonstrate the significant benefits of introducing inter-vehicle communication and the associated observation harvesting mechanism presented in this chapter. The position tracking of V3 does not show any difference with V2V enabled or disabled, as respectively illustrated in Figures 5.5(d) and 5.5(c). This is because V3 provides a direct position update to the base station before any of the other vehicles. Without observation harvesting (V2V disabled), the results are the same as the previous work in **Chapter 4**.

In terms of estimation errors and entropies, Figure 5.6 compares the tracking per-

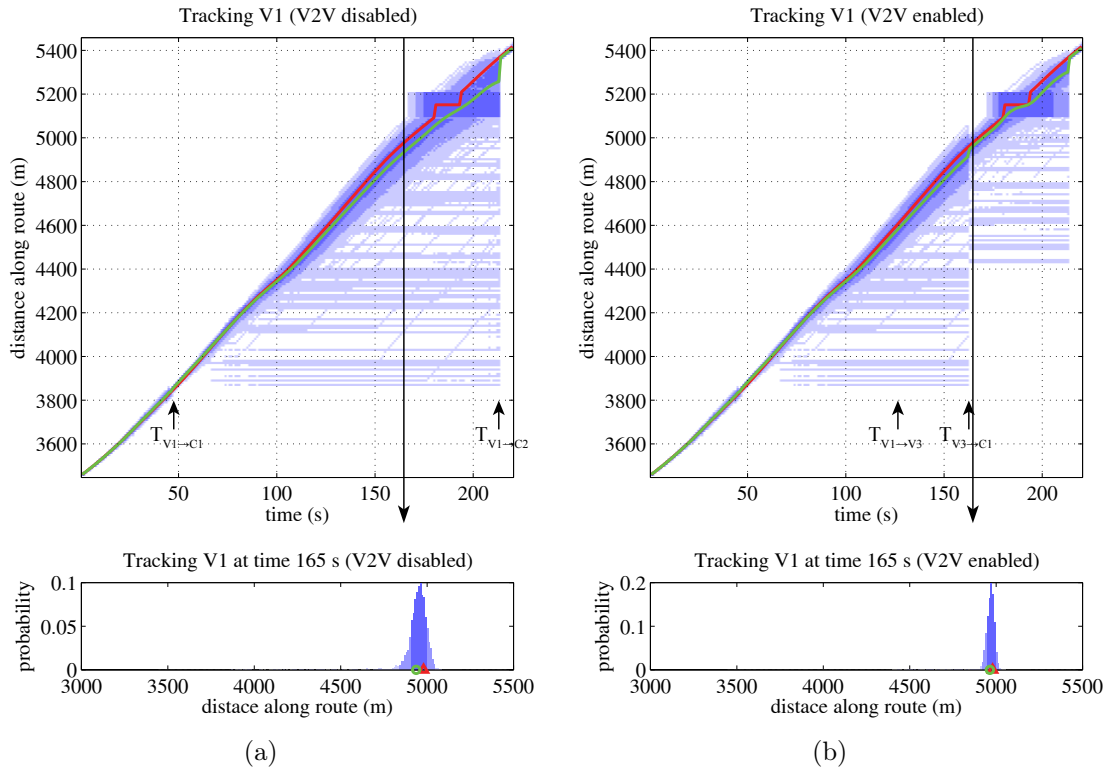


Figure 5.4 – Tracking results of the multiple vehicle scenario. Figure 5.5 continues the results in this figure. (a) shows the tracking result of V1 with V2V disabled, (b) shows the tracking with V2V enabled. (b) demonstrates a more accurate position estimate of V1 from time $T_{V3\rightarrow C1}$ (162 s) onward after a position update was delivered to the base station by V3. In Figure 5.5(b), an improved position estimate for V2 is shown with V2V enabled from time $T_{V1\rightarrow C2}$ (213 s) onward in comparison with the tracking results with V2V disabled in Figure 5.5(a). This is a result of V1 delivering a position observation from V2 to the base station as a delayed observation. Comparing Figures 5.5(c) and 5.5(d), there were no delayed updates provided for V3, meaning that the results are identical with or without V2V.

formance with V2V disabled and enabled, clearly illustrating the improvements with the incorporation of V2V communication. The tracking with V2V communication enabled is represented with blue lines. It can be seen that after some time it outperforms the tracking with V2V disabled (denoted by green lines). This is also true in both individual and fleet estimation errors and entropies, which are presented in Figure 5.7.

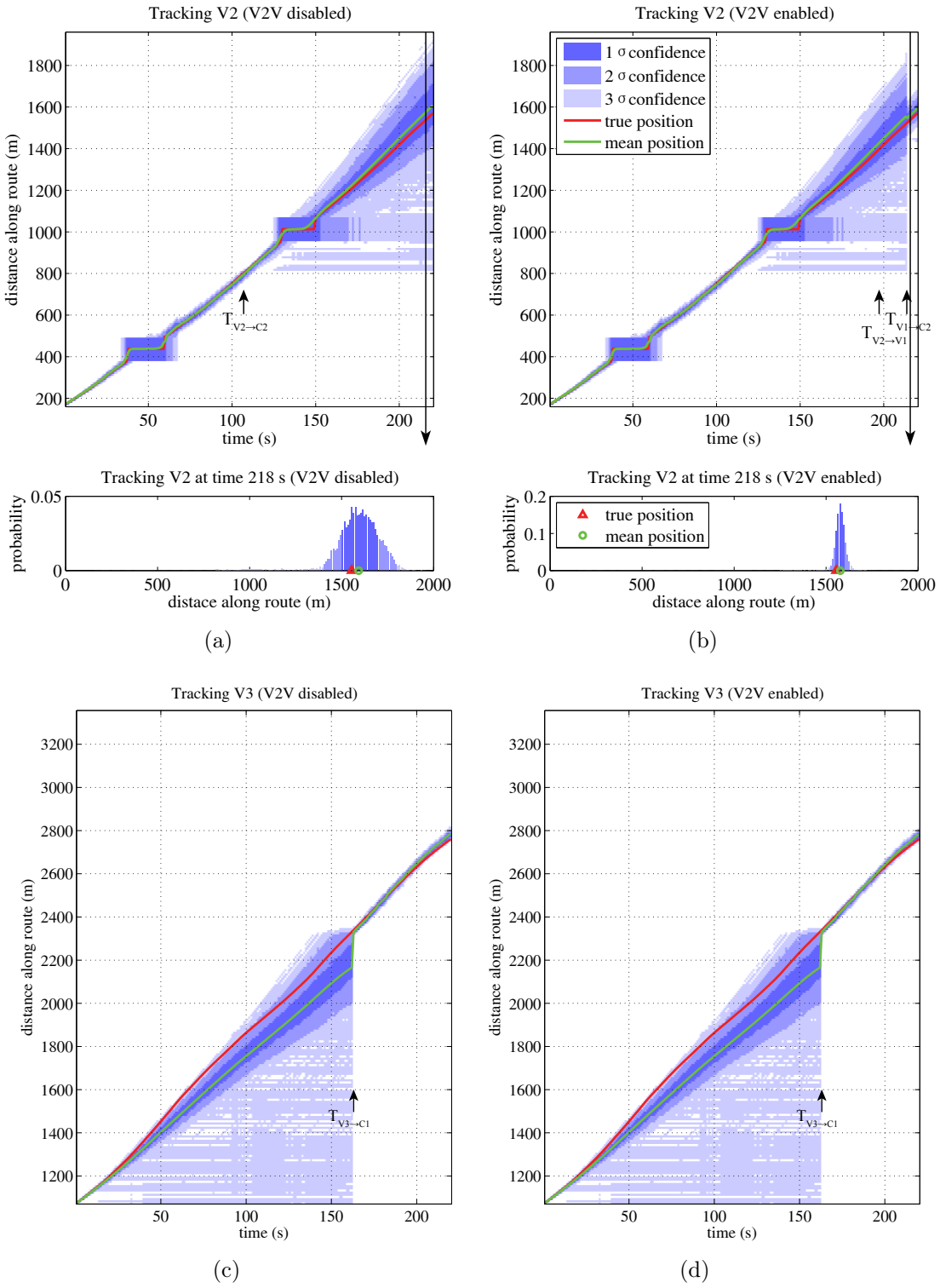
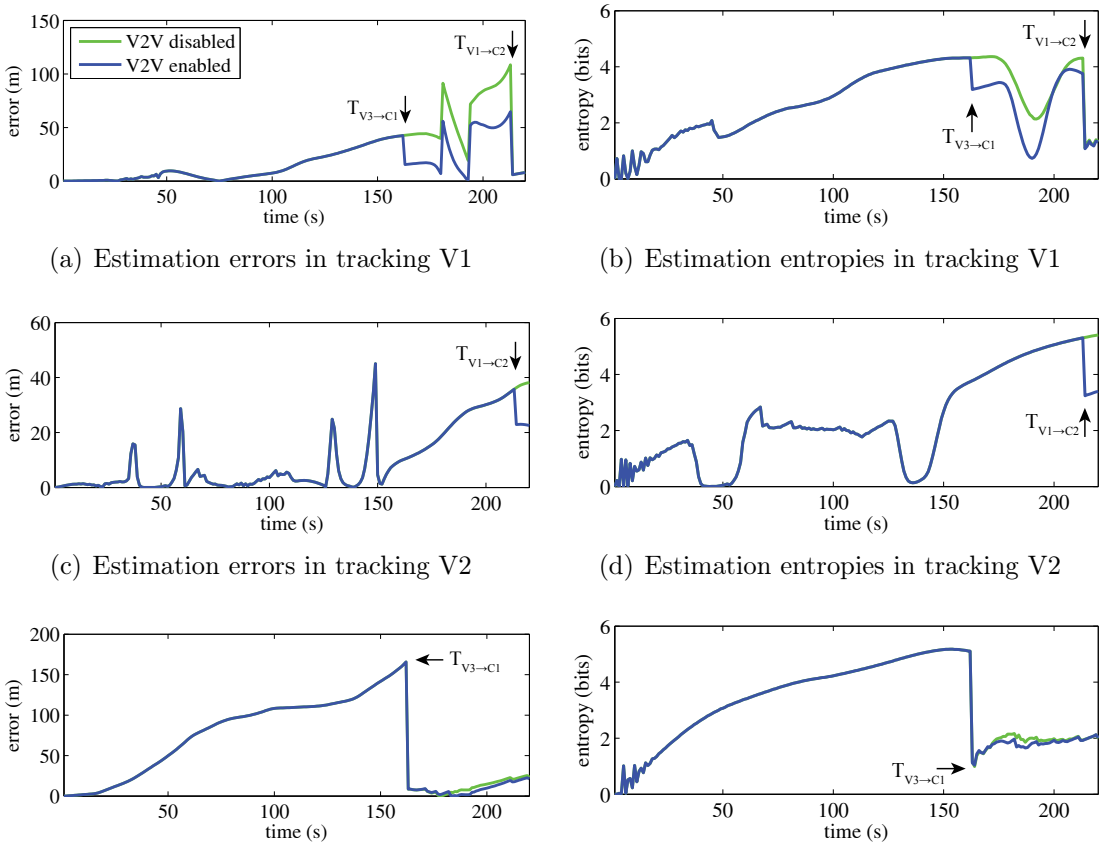


Figure 5.5 – Tracking results of the multiple vehicle scenario (cont.).



(a) Estimation errors in tracking V1

(b) Estimation entropies in tracking V1

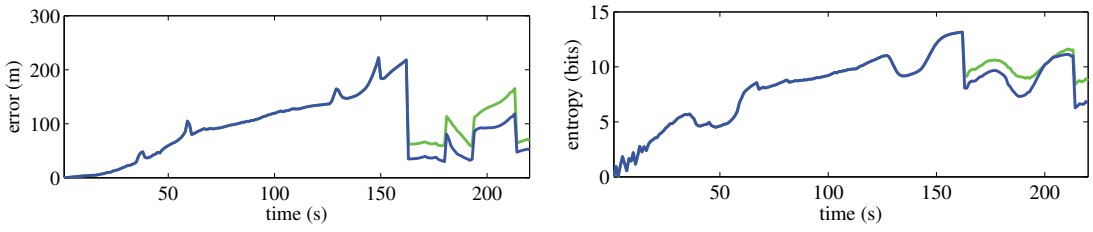
(c) Estimation errors in tracking V2

(d) Estimation entropies in tracking V2

(e) Estimation errors in tracking V3

(f) Estimation entropies in tracking V3

Figure 5.6 – Comparisons of estimation errors and entropies. Reduction in errors and entropies of V1 is shown in (a) and (b) from time $T_{V3\to C1}$ on. Likewise improvements can be seen for V2 after time $T_{V1\to C2}$ as illustrated in (c) and (d). The results for tracking V3 do not show much difference in (e) and (f).



(a) Total estimation errors

(b) Fleet entropies

Figure 5.7 – Comparisons of fleet estimation errors and entropies. (a) and (b) suggest an improvement in total estimation errors and fleet entropies respectively with observation harvesting introduced.

5.4.4 Discussions

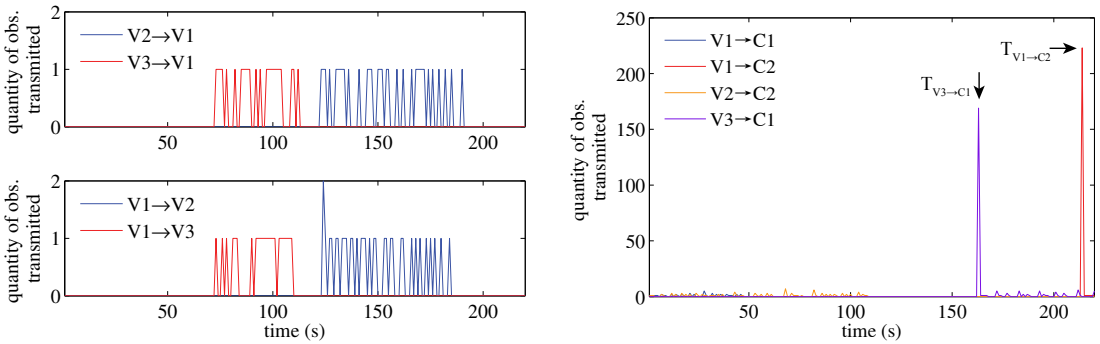
The accuracy of the GPS obtained position information can be degraded in certain conditions due to signal interference, multipath, poor satellite configuration and so forth. The tracking accuracy from the base station position estimates however are not heavily influenced by this as the position uncertainty of each vehicle is generally an order of magnitude higher than the GPS error after several minutes with no updates. Figures 5.4 and 5.5 shows how the position uncertainty of each vehicle grows to the order of hundreds of metres in minutes when the base station is not provided with new position information. This indicates that the potential GPS error (on the order of 10 metres) is not significant to the overall tracking performance.

Negative information contributes to improve the estimation performance in the absence of available positive observations. Vehicles that are not in direct communication with data collectors are able to be tracked with delayed observations brought back by returning vehicles.

Since information becomes less useful (or "diluted") as time goes on, tracking accuracy could be improved by reducing the delays in observations. This can be achieved by shortening the "blind time", i.e. the average time interval between the base station receiving observations about the same vehicles. The approaches can include:

- increasing quantity/density of data collection points: to shorten the time taken for a vehicle to move from a data collection point to another.
- optimising layout of data collection points: they are recommended to be placed at regions with heavy traffic and high timing uncertainty, such as special context areas and intersections, so that vehicles are detected by the data collectors more of the time.

For example, for a duration of almost 1 minute in the experiment, i.e. between time points $T_{V2\rightarrow C2}$ (107 s) and $T_{V3\rightarrow C1}$ (162 s), the infrastructure could not detect any of the vehicles and the position estimation of all three vehicles had to rely on



(a) V2V communication bandwidth

(b) V2I communication bandwidth

Figure 5.8 – Communication bandwidth cost in V2I and V2V. (a) vehicles communicate and synchronise position data with each other when in range. (b) vehicles contribute absolute observations to the base station when they are within communication range of a data collection point.

the motion prediction alone. In this case the blind time could be reduced by either deploying additional data collection point(s) somewhere between the two existing ones, or relocating C2 near the intersection #205, where vehicles generally take more time to traverse.

The blind time can also be reduced by the following:

- increasing the density of vehicles: to increase the frequency of V2V communication between vehicles, which consequently increases the sharing of information between vehicles. This will happen naturally in operations with a large number of vehicles, potentially making the proposed algorithm very close to a solution using full communication coverage.
- increasing the communication range of V2I and/or V2V: to increase time duration that a vehicle stays observable to data collection points and other vehicles.
- optimising the routes of vehicles: with vehicle density unchanged, an increased frequency of V2V communication could be achieved by planning the vehicles' routes to increase the number of interactions.

The algorithm additionally benefits from a shorter time delay of observations as the computational expense of the algorithm is to some extent determined by the length

of blind time. The particle filter rewinds back to a historical point once new delayed observations are received, the number of steps to rerun depends on the oldest observation in the received observations set $\mathbf{\Omega}^{v_p}_{k|k-1}$, and are ultimately bounded by the sliding time window size T_w as previously discussed in Section 5.3.3. All in all, the fundamental idea in optimising any implementation of the tracking approach proposed is to minimise the blind time.

The V2I bandwidth requirements from the experimental results were dominated by two major transactions illustrated in Figure 5.8(b). V3 transmitted 169 absolute position observations at time $T_{V3\rightarrow C1}$ (162 s) and V1 transmitted 223 observations at $T_{V1\rightarrow C2}$ (213 s). This was much less than the maximum V2I bandwidth cost predicted by $N_v \tau^{v_p \rightarrow I}$ for absolute observations. This was because in a large environment, moving vehicles generally are only in communication range for a relatively short period of time. The observations collected indirectly from other vehicles are transmitted to the base station together with the vehicle's own egocentric observations in a single transaction. The maximum V2I bandwidth cost would be reached in cases where a vehicle is in communication range with another vehicle for a long period before the transaction occurs, as would happen in a parking lot or when vehicles travel close together. For V2V communication, the bandwidth cost for absolute observations is negligible and is bounded by $N_v - 1$ according to Table 5.3. In a scenario containing three vehicles, the maximum V2V bandwidth cost is 2 absolute observations per iteration as validated by the experimental results shown in Figure 5.8(a).

The observation harvesting mechanism substantially increases the robustness of vehicle tracking in large environments with only a small bandwidth cost. A fast network connection time is crucial when a sparse collection of vehicles are moving in a large environment. Communication in a global wireless network can easily be interrupted by vehicle motion, as the signal propagation path between moving vehicles is constantly changing. The observation harvesting approach uses a store-and-synchronise concept to deal with intermittent communication and aims to disseminate data in an opportunistic manner. Without the additional overheads that come with packet routing, the fast and lightweight P2P communication makes the observation harvesting

mechanism an effective approach for sharing observations amongst mobile vehicles and fixed infrastructure in applications where a global network is not feasible.

The main characteristic of observation harvesting is that every vehicle keeps the latest information for as many vehicles as possible in its local observation pool. The temporary failure of some vehicles, or even the majority of vehicles in the network, will not result in the failure of the entire network. This is an especially attractive characteristic for applications where safety is strongly emphasised.

5.5 Summary

This chapter has presented a probabilistic algorithm to track multiple vehicles in a large area with delayed observations. The algorithm is based on the long-term vehicle motion prediction model presented in **Chapter 4**. The observation harvesting concept is proposed for effective and robust data dissemination amongst vehicles and fixed infrastructure without requiring global communication. The most informative position information is shared among vehicles moving around a site and forwarded to the base station via returning vehicles. This chapter presented a particle filter approach capable of dealing with delayed observations. The base station updates vehicle position estimates with absolute position information gathered from returning vehicles, together with V2I observations from fixed data collection points distributed in the area. The experiment results presented in this chapter show that it is possible to obtain consistent position estimates for multiple vehicles over long periods of time. V2V communication combined with a limited number of data collection points was used to demonstrate large scale multiple vehicle tracking without a full coverage communication network. The experimental results clearly showed the improvement from incorporating V2V communication with the modelling constraints, and using positive and negative information.

The next chapter will propose a nonparametric cooperative tracking approach, which enables the incorporation of relative observations to further improve the tracking performance.

Chapter 6

Nonparametric Cooperative Tracking

6.1 Introduction

This chapter is to present a more generalised approach for cooperative tracking in cases where the Gaussian assumption is not valid. A novel Gibbs sampler based cooperative particle filter (GSCPF) for nonparametric cooperative tracking is proposed in this chapter to estimate the state of every mobile node. This filter incorporates all available egocentric and relative inter-node observations and can be implemented in a centralised or decentralised architecture. The proposed GSCPF has the importance sampling step in a classical PF replaced by a Gibbs sampler, which is a MCMC method. The background of parametric approaches for cooperative tracking could be found in Appendix D. In the remainder of the chapter, Section 6.2 provides the Bayesian formulation on cooperative tracking. The principle of the proposed approach is introduced in Section 6.3, followed by simulation tests and results presented in Section 6.4. Results in a MANET are demonstrated in Section 6.5. Lastly Section 6.6 summarises the work presented in this chapter.

6.2 Bayesian Formulation on Cooperative Tracking

Suppose a joint state representing a set of N_n nodes moving in a field:

$$\mathbf{X} = \left[\left(\mathbf{x}^1\right)^T \quad \left(\mathbf{x}^2\right)^T \quad \cdots \quad \left(\mathbf{x}^{N_n}\right)^T \right]^T$$

To track the joint state cooperatively, a filter propagates states from time $k-1$ to k and updates estimates with all observations at time k.

$$P\left(\mathbf{X}_k|\mathbf{Z}_{1:k}\right) \propto P\left(\mathbf{Z}_k|\mathbf{X}_k\right) \int P\left(\mathbf{X}_k|\mathbf{X}_{k-1}\right) P\left(\mathbf{X}_{k-1}|\mathbf{Z}_{1:k-1}\right) d\mathbf{X}_{k-1} \tag{6.1}$$

We make the following assumptions.

- every node moves independently in the field, from which we have: $P\left(\mathbf{X}_k|\mathbf{X}_{k-1}\right) = \prod_{p=1}^{N_n} P\left(\mathbf{x}_k^p|\mathbf{x}_{k-1}^p\right)$.
- an egocentric position observation $\mathbf{z}_k^p$ regarding node p is only dependent on current state of the node $\mathbf{x}_k^p$.
- a relative range observation $\mathbf{z}_k^{p\to q}$ $(p \neq q)$ is only conditional on the current state of two involved nodes, i.e. $\mathbf{x}_k^p$ and $\mathbf{x}_k^q$.

Therefore the observation component of Equation (6.1) is able to be further factorised to absolute and relative observations.

$$P\left(\mathbf{Z}_k|\mathbf{X}_k\right) = \left(\prod_{p=1}^{N_n} P\left(\mathbf{z}_k^p|\mathbf{x}_k^p\right) \right) \left(\prod_{p=1}^{N_n} \prod_{q=1}^{N_n} P\left(\mathbf{z}_k^{p\to q}|\mathbf{x}_k^p, \mathbf{x}_k^q\right) \right)$$

where $p \neq q$.

A marginal distribution $P\left(\mathbf{x}_k^p|\mathbf{Z}_{1:k}\right)$ for node p at time k could be obtained by integrating with respect to the joint state of the rest nodes (denoted by $\bar{\mathbf{X}}_k$) in the joint posterior in Equation (6.1). This is achieved by:

$$P\left(\mathbf{x}_k^p|\mathbf{Z}_{1:k}\right) = \int P\left(\mathbf{X}_k|\mathbf{Z}_{1:k}\right) d\bar{\mathbf{X}}_k \tag{6.2}$$

where $\mathbf{X}_k = \left[\left(\mathbf{x}_k^p\right)^T \quad \bar{\mathbf{X}}_k^T\right]^T$.

6.3 A Novel Particle Filter for Cooperative Tracking

6.3.1 Theory

When a particle filter (PF) is required for cooperative tracking problems, it is usually not feasible to use a set of particles to approximate a high dimensional joint state. The required ensemble size grows exponentially with dimension of the joint state space (the dimension is proportional to node quantity in cooperative localisation/tracking), which is well known as "dimensional curse".

The proposed algorithm keeps the marginal state of each node and each node is tracked independently. Suppose at time k, each of N_n mobile nodes is tracked by a discrete set of L particles, we have:

$$\Theta_k = \begin{bmatrix} \{\mathbf{x}_k^{1,(i)}, w_k^{1,(i)}\}_{i=1}^L \sim P\left(\mathbf{x}_k^1|\mathbf{Z}_{1:k}\right) \\ \{\mathbf{x}_k^{2,(i)}, w_k^{2,(i)}\}_{i=1}^L \sim P\left(\mathbf{x}_k^2|\mathbf{Z}_{1:k}\right) \\ \vdots \\ \{\mathbf{x}_k^{N_n,(i)}, w_k^{N_n,(i)}\}_{i=1}^L \sim P\left(\mathbf{x}_k^{N_n}|\mathbf{Z}_{1:k}\right) \end{bmatrix}$$

For each time step and each target node p, we infer a joint posterior with an incomplete observation set based on the similar formulation in Equation (6.1).

$$\begin{aligned} P_{\mathbf{X}_k} &\triangleq P\left(\bar{\mathbf{X}}_k|\mathbf{x}_k^p, \mathbf{Z}_k\right) P\left(\mathbf{x}_k^p|\mathbf{Z}_{1:k}\right) \\ &= \frac{P\left(\bar{\mathbf{X}}_k, \mathbf{x}_k^p|\mathbf{Z}_k\right)}{P\left(\mathbf{x}_k^p|\mathbf{Z}_k\right)} \times \frac{P\left(\mathbf{Z}_k|\mathbf{x}_k^p\right) P\left(\mathbf{x}_k^p|\mathbf{Z}_{1:k-1}\right)}{P\left(\mathbf{Z}_k|\mathbf{Z}_{1:k-1}\right)} \\ &\propto P\left(\mathbf{Z}_k|\mathbf{X}_k\right) P\left(\mathbf{x}_k^p|\mathbf{Z}_{1:k-1}\right) \end{aligned} \tag{6.3}$$

where $P\left(\mathbf{x}_k^p|\mathbf{Z}_{1:k-1}\right) = \int P\left(\mathbf{x}_k^p|\mathbf{x}_{k-1}^p\right) P\left(\mathbf{x}_{k-1}^p|\mathbf{Z}_{1:k-1}\right) d\mathbf{x}_{k-1}^p$.

Please note that the joint posterior $P_{\mathbf{X}_k}$ does not contain prior knowledge of nodes except for node p. These nodes grouped in $\bar{\mathbf{X}}_k$ are defined as "auxiliary nodes", for they are temporarily aggregated together to help track the "primary node" p. The auxiliary nodes are then marginalised away to obtain the marginal probability of the primary node state $\mathbf{x}_k^p$.

$$P\left(\mathbf{x}_k^p|\mathbf{Z}_{1:k}\right) = \int P\left(\bar{\mathbf{X}}_k|\mathbf{x}_k^p, \mathbf{Z}_k\right) P\left(\mathbf{x}_k^p|\mathbf{Z}_{1:k}\right) d\bar{\mathbf{X}}_k \tag{6.4}$$

The resultant marginal distribution $P\left(\mathbf{x}_k^p|\mathbf{Z}_{1:k}\right)$ is inferred with a full set of observations up to time k, which is identical to that in Equation (6.2).

6.3.2 Gibbs Sampling

The joint posterior distribution $P_{\mathbf{X}_k}$ is usually too complicated to sample from directly. Instead, a Gibbs sampler [58] [33] is introduced to sample from conditional distributions.

Define N_n variables $\theta^1, \cdots, \theta^{N_n}$ corresponding to all N_n nodes in the joint posterior distribution. We have a replacing joint distribution written as:

$$\begin{aligned} &P\left(\theta^1, \cdots, \theta^{p-1}, \theta^p, \theta^{p+1}, \cdots, \theta^{N_n}\right) \\ &= P\left(\bar{\mathbf{X}}_k|\mathbf{x}_k^p, \mathbf{Z}_k\right) P\left(\mathbf{x}_k^p|\mathbf{Z}_{1:k}\right) \end{aligned}$$

We can initialise these variables deterministically or randomly.

$$\theta^1 = \theta_0^1, \cdots, \theta^{N_n} = \theta_0^{N_n}$$

The sampler draws a sample from the conditional distribution of each variable given the remaining variables at a time. A traversal of all N_n variables is defined as a scan of the sampler. Therefore for each auxiliary node q at each scan $j \geq 1$, we sample:

$$\begin{aligned} \theta_j^q &\sim P\left(\theta^q | \theta_j^1, \cdots, \theta_j^{q-1}, \theta_{j-1}^{q+1}, \cdots, \theta_{j-1}^{N_n}\right) \\ &\propto P\left(\mathbf{x}_k^q | \mathbf{z}_k^q\right) \left(\prod_{r=1}^{q-1} P\left(\mathbf{z}_k^{r \leftrightarrow q} | \mathbf{x}_k^r = \theta_j^r, \mathbf{x}_k^q\right) \right) \\ &\times \left(\prod_{r=q+1}^{N_n} P\left(\mathbf{z}_k^{r \leftrightarrow q} | \mathbf{x}_k^r = \theta_{j-1}^r, \mathbf{x}_k^q\right) \right) \end{aligned} \tag{6.5}$$

The equation is derived by substituting $\mathbf{x}_k^1 = \theta_j^1, \cdots, \mathbf{x}_k^{q-1} = \theta_j^{q-1}, \mathbf{x}_k^{q+1} = \theta_{j-1}^{q+1}, \cdots, \mathbf{x}_k^{N_n} = \theta_{j-1}^{N_n}$ into $P_{\mathbf{X}_k}$ in Equation (6.3).

Then for the primary node p at the scan, we sample:

$$\begin{aligned} \theta_j^p &\sim P\left(\theta^p | \theta_j^1, \cdots, \theta_j^{p-1}, \theta_j^{p+1}, \cdots, \theta_j^{N_n}\right) \\ &\propto P\left(\mathbf{z}_k^p | \mathbf{x}_k^p\right) \int P\left(\mathbf{x}_k^p | \mathbf{x}_{k-1}^p\right) P\left(\mathbf{x}_{k-1}^p | \mathbf{Z}_{1:k-1}\right) d\mathbf{x}_{k-1}^p \\ &\times \left(\prod_{r=1, r \neq p}^{N_n} P\left(\mathbf{z}_k^{r \leftrightarrow p} | \mathbf{x}_k^r = \theta_j^r, \mathbf{x}_k^p\right) \right) \end{aligned} \tag{6.6}$$

Likewise, the equation is obtained by substituting $\mathbf{x}_k^1 = \theta_j^1, \cdots, \mathbf{x}_k^{p-1} = \theta_j^{p-1}, \mathbf{x}_k^{p+1} = \theta_j^{p+1}, \cdots, \mathbf{x}_k^{N_n} = \theta_j^{N_n}$ into $P_{\mathbf{X}_k}$.

Iterative runs of Equations (6.5) and (6.6) generate a Gibbs chain. Figure 6.1 demonstrates one of the scans in the sampling process. Following a sufficient burn-in period (of, say, N_{bp} scans), the chain approaches its stationary distribution. From the Gibbs

Table 6.1 – Algorithm: GSCPF (naive)

$\Theta_k \leftarrow$ *Partile_Filter* $(\Theta_{k-1}, \mathbf{Z}_k)$

par for $p = 1$ to N_n **do**
  load prior particle set $\{\mathbf{x}_{k-1}^{p,(i)}, w_{k-1}^{p,(i)}\}_{i=1}^{L}$ from Θ_{k-1}
  initialise Gibbs variables $\theta^1 = \theta_0^1, \cdots, \theta^{N_n} = \theta_0^{N_n}$
  for $j = 1$ to $(N_{bp} + N_{si} \times L)$ **do**
    Sampling for each auxiliary node:
    for $q = 1$ to $p-1$, $p+1$ to N_n **do**
      draw $\theta_j^q \sim P\left(\theta^q | \theta_j^1, \cdots, \theta_j^{q-1}, \theta_{j-1}^{q+1}, \cdots, \theta_{j-1}^{N_n}\right)$ with observations from $\mathbf{Z}_k$ fused in
    end for
    Sampling for the primary node:
    draw $\theta_j^p \sim P\left(\theta^p | \theta_j^1, \cdots, \theta_j^{p-1}, \theta_j^{p+1}, \cdots, \theta_j^{N_n}\right)$ with observations from $\mathbf{Z}_k$ fused in
    if $(j - N_{bp})$ is an integer times of N_{si} **do**
      Extraction:
      particle index $i = (j - N_{bp}) / N_{si}$
      save sample $\mathbf{x}_k^{p,(i)} = \theta_j^p$
      set weight $w_k^{p,(i)} = 1/L$
    end if
  end for
  save posterior particles $\{\mathbf{x}_k^{p,(i)}, w_k^{p,(i)}\}_{i=1}^{L}$ to Θ_k
end for

chain we then extract a set of particles to approximate the marginal posterior distribution of the primary node $P\left(\mathbf{x}_k^p | \mathbf{Z}_{1:k}\right)$. Theoretically the desired posterior distribution could be approximated to any degree of accuracy with sufficient number of particles. To reduce autocorrelation, samples are extracted with an interval of N_{si} (or a thinning ratio of $1/N_{si}$ in other words), i.e. by taking every $(N_{si})^{th}$ value in the chain. Please note that the extracted posterior particles have equal weights, therefore resampling process is not required any more.

By repeating the above algorithm for every node at time k, all nodes are updated cooperatively with both egocentric and relative observations in $\mathbf{Z}_k$. Figure 6.2 illustrates how each node is treated as the primary node in turns. The algorithm is also summarised in Table 6.1 in the form of a pseudocode.

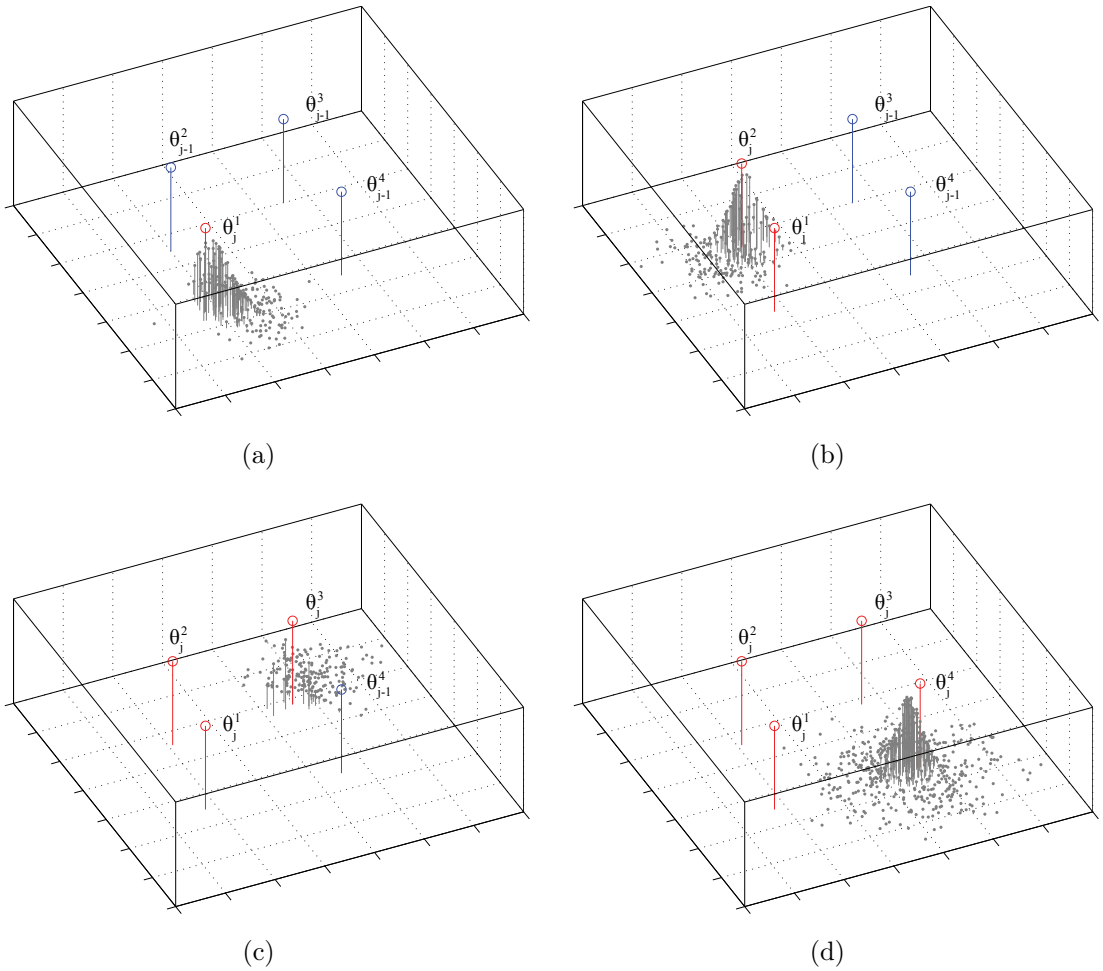
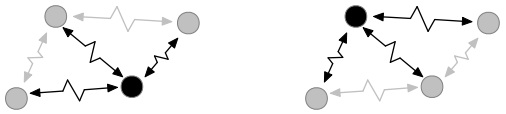


Figure 6.1 – A scan in the Gibbs sampling. (a) to (c) show the sampling for three auxiliary nodes and at last (d) for the primary node. The noise contained in the egocentric position observation of each node is $\mathcal{N}\left(\left[\begin{smallmatrix}0\\0\end{smallmatrix}\right], \left[\begin{smallmatrix}10^2 & 0\\0 & 10^2\end{smallmatrix}\right]\right)$, while the noise in inter-node distance measurements is written as $\mathcal{N}\left(0, 10^2\right)$. Particle quantity used for each node is 500.

6.3.3 Optimisations

Processing the First Iteration

When it is assumed that nodes are mutually independent at the first iteration, the prior information of the auxiliary node could be safely taken into account in the filtering. However, please note that this is only applicable at the first iteration, as from then on the nodes become cross-correlated. One could still use Equation (6.5)



(a) (b)

Figure 6.2 – Primary and auxiliary nodes. An arbitrary node works as the primary node in a) while the rest are auxiliary nodes. In b) another node becomes the primary node. This goes on so that every node is treated as the primary node. The processing of these four primary nodes in the figure is independent with each other, therefore computation for them could be processed in parallel.

at the first iteration, however at the expense of tracking accuracy at the first several iterations. To properly consider the prior information at the first iteration, it requires special formulation on the Gibbs sampling for auxiliary nodes, which differs from the one presented in Equation (6.5).

$$\begin{aligned}
\theta_j^q &\sim P\left(\theta^q|\theta_j^1,\cdots,\theta_j^{q-1},\theta_{j-1}^{q+1},\cdots,\theta_{j-1}^{N_n}\right) \\
&\propto P\left(\mathbf{z}_1^q|\mathbf{x}_1^q\right)\int P\left(\mathbf{x}_1^q|\mathbf{x}_0^q\right)P\left(\mathbf{x}_0^q\right)d\mathbf{x}_0^q \\
&\times\left(\prod_{r=1}^{q-1}P\left(\mathbf{z}_1^{r\leftrightarrow q}|\mathbf{x}_1^r=\theta_j^r,\mathbf{x}_1^q\right)\right) \\
&\times\left(\prod_{r=q+1}^{N_n}P\left(\mathbf{z}_1^{r\leftrightarrow q}|\mathbf{x}_1^r=\theta_{j-1}^r,\mathbf{x}_1^q\right)\right)
\end{aligned} \tag{6.7}$$

From the second iteration on, Equation (6.5) is used for every auxiliary node.

Construction of Uninformative Distribution for Auxiliary Nodes

In Equation (6.5), the prior distribution of an auxiliary node q is constructed with its egocentric observation $P\left(\mathbf{x}^q|\mathbf{z}_k^q\right)$ when it exists. However in cases where the auxiliary node is out of egocentric observations, the prior distribution should be an uninformative distribution, which means it does not contain any probabilistic information. Ideally the uninformative distribution is a uniform distribution over the entire state

space, which is not feasible to create. Instead, a more efficient method is proposed to construct regional uninformative distribution around the possible area of the auxiliary node's location.

Given an arbitrary proposed distribution $q(\mathbf{x})$ that is easy to sample from (say, Gaussian distribution), we could obtain an approximate of uniform distribution by dividing the proposed distribution by itself. More specifically, we first draw a set of equally weighted particles representing the proposed distribution, i.e. $\{\mathbf{x}^{(i)}, w^{(i)}\}_{i=1}^{L} \sim q(\mathbf{x})$. Then the weight of each particle is divided by its probability at its position, which means for particle $i = 1$ to L, we have:

$$\{\mathbf{x}^{(i)}, w^{(i)}\} \sim q(\mathbf{x})$$

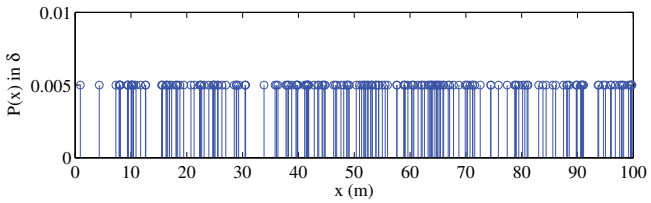
$$w^{(i)} = \frac{w^{(i)}}{q(\mathbf{x}^{(i)})}$$

Then the obtained set of weights are normalised. For instance, Figures 6.3 and 6.4 demonstrate two regional uninformative distributions constructed in 1D and 2D spaces respectively, compared to their corresponding ideals over the whole spaces.

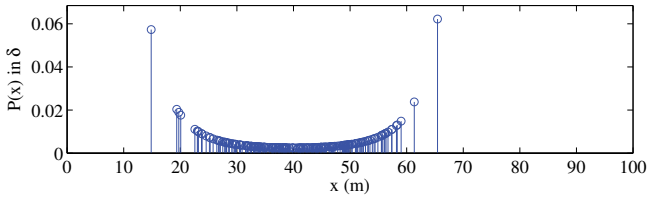
Pre-Processing of the Primary Node

In the naive implementation of the GSCPF, the intra-node fusion step of the primary node (i.e. the combination of prediction and fusion of its egocentric observation if it exits) is processed in each scan of the Gibbs sampling, see Equation (6.6). Moreover, each extraction from the Gibbs chain produces only one posterior particle for the primary node. Therefore in order to collect in total L posterior particles the Gibbs chain should meet a length of exactly $N_{bp} + N_{si} \times L$ scans. These two characteristics of the naive concept lower down the efficiency in computation.

An improvement to the GSCPF is to terminate the intra-node fusion process before the iterative Gibbs sampling step. The sampler loads the pre-processed particle set

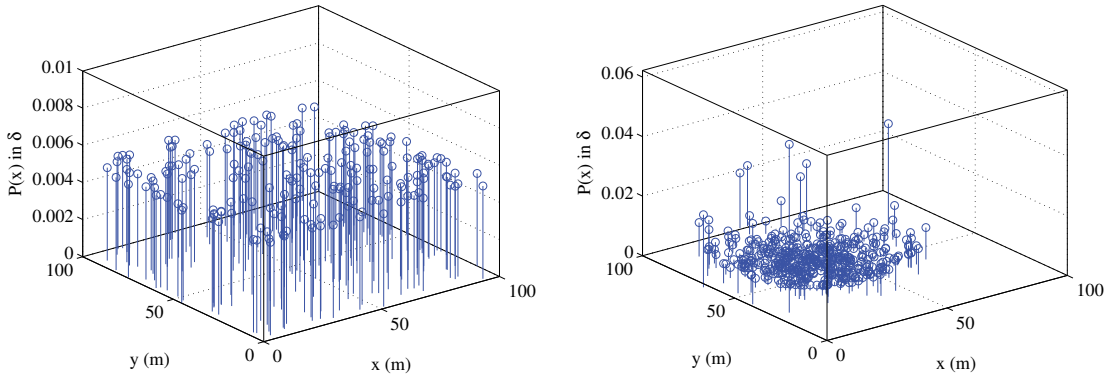


(a) Ideal



(b) Regional

Figure 6.3 – Uninformative distributions in 1D space. The distribution in (b) uses $\mathcal{N}\left(40, 10^2\right)$ as the proposed distribution.



(a) Ideal (b) Regional

Figure 6.4 – Uninformative distributions in 2D space. The distribution in (b) uses $\mathcal{N}\left(\left[\begin{smallmatrix}40\\60\end{smallmatrix}\right], \left[\begin{smallmatrix}15^2 & 0\\0 & 15^2\end{smallmatrix}\right]\right)$ as the proposed distribution.

of the primary node in each scan and only the inter-node information is left to process. That is to say, compared to the naive implementation, the improved GSCPF moves the intra-node fusion process out of the Gibbs sampling. In addition, when it comes to each $(N_{si})^{th}$ scan in the chain, the updated set of weights are extracted and accumulated. Lastly when the sampling procedure finishes, the accumulated weights are normalised to yield the posterior particles. The improvement brings a flexible requirement on the length of the Gibbs chain and thus the computation cost could be

Table 6.2 – Algorithm: GSCPF (Table 6.1 rewritten)

$\Theta_k \leftarrow Partile\_Filter\left(\Theta_{k-1}, \mathbf{Z}_k\right)$

par for $p = 1$ to N_n **do**
  load prior particle set $\{\mathbf{x}_{k-1}^{p,(i)}, w_{k-1}^{p,(i)}\}_{i=1}^{L}$ from Θ_{k-1}
  initialise Gibbs variables $\theta^1 = \theta_0^1, \cdots, \theta^{N_n} = \theta_0^{N_n}$
  for $j = 1$ to $(N_{bp} + N_{si} \times L)$ **do**
    Sampling for each auxiliary node: (Omitted)
    Sampling for the primary node:
    Intra-node Fusion:
    $P\left(\mathbf{x}_k^p|\mathbf{Z}_{1:k}\right)^- \propto P\left(\mathbf{z}_k^p|\mathbf{x}^p\right) \int P\left(\mathbf{x}_k^p|\mathbf{x}_{k-1}^p\right) P\left(\mathbf{x}_{k-1}^p|\mathbf{Z}_{1:k-1}\right) d\mathbf{x}_{k-1}^p$
    $\{\mathbf{x}_k^{p,(i)}, w_{k-}^{(i)}\}_{i=1}^{L} \sim P\left(\mathbf{x}_k^p|\mathbf{Z}_{1:k}\right)^-$
    Inter-node Fusion:
    $P\left(\mathbf{x}_k^p|\mathbf{Z}_{1:k}\right)^+ \propto P\left(\mathbf{x}_k^p|\mathbf{Z}_{1:k}\right)^- \left(\prod_{r=1,r\neq p}^{N_n} P\left(\mathbf{z}_k^{r\leftrightarrow p}|\mathbf{x}^r = \theta_j^r, \mathbf{x}^p\right)\right)$
    $\{\mathbf{x}_k^{p,(i)}, w_{k+}^{(i)}\}_{i=1}^{L} \sim P\left(\mathbf{x}_k^p|\mathbf{Z}_{1:k}\right)^+$
    draw $\theta_j^p \sim P\left(\mathbf{x}_k^p|\mathbf{Z}_{1:k}\right)^+$
    if $(j - N_{bp})$ is an integer times of N_{si} **do**
      Extraction:
      particle index $i = (j - N_{bp})/N_{si}$
      save sample $\mathbf{x}_k^{p,(i)} = \theta_j^p$
      set weight $w_k^{p,(i)} = 1/L$
    end if
  end for
  save posterior particles $\{\mathbf{x}_k^{p,(i)}, w_k^{p,(i)}\}_{i=1}^{L}$ to Θ_k
end for

reduced. Regardless of the chain length N_{cl} after the burning-period, the quantity of the posterior particles is kept identical to the prior. Nevertheless, the produced posterior particles are not equally weighted and the resampling process should be used when necessary to prevent particle degeneracy. The naive and improved algorithms respectively are shown in Tables 6.2 and 6.3.

Centralised and Decentralised Implementation

The algorithm can be implemented in a centralised or decentralised architecture. In the centralised structure, the estimation is performed in a base station. At every time step, observations including both absolute and relative ones are forwarded to the base

Table 6.3 – Algorithm: GSCPF (improved)

$\Theta_k \leftarrow Partile\_Filter\left(\Theta_{k-1}, \mathbf{Z}_k\right)$

par for $p = 1$ to N_n **do**
  load prior particle set $\{\mathbf{x}_{k-1}^{p,(i)}, w_{k-1}^{p,(i)}\}_{i=1}^{L}$ from Θ_{k-1}
  Intra-node Fusion:
  $P\left(\mathbf{x}_k^p|\mathbf{Z}_{1:k}\right)^- \propto P\left(\mathbf{z}_k^p|\mathbf{x}^p\right) \int P\left(\mathbf{x}_k^p|\mathbf{x}_{k-1}^p\right) P\left(\mathbf{x}_{k-1}^p|\mathbf{Z}_{1:k-1}\right) d\mathbf{x}_{k-1}^p$
  $\{\mathbf{x}_k^{p,(i)}, w_{k-}^{(i)}\}_{i=1}^{L} \sim P\left(\mathbf{x}_k^p|\mathbf{Z}_{1:k}\right)^-$
  initialise weights $\{w_k^{p,(i)}\}_{i=1}^{L} = 0$
  initialise Gibbs variables $\theta^1 = \theta_0^1, \cdots, \theta^{N_n} = \theta_0^{N_n}$
  for $j = 1$ to $(N_{bp} + N_{cl})$ **do**
    Sampling for each auxiliary node: (Omitted)
    Sampling for the primary node:
    Inter-node Fusion:
    $P\left(\mathbf{x}_k^p|\mathbf{Z}_{1:k}\right)^+ \propto P\left(\mathbf{x}_k^p|\mathbf{Z}_{1:k}\right)^- \left(\prod_{r=1,r\neq p}^{N_n} P\left(\mathbf{z}_k^{r\leftrightarrow p}|\mathbf{x}^r = \theta_j^r, \mathbf{x}^p\right)\right)$
    $\{\mathbf{x}_k^{p,(i)}, w_{k+}^{(i)}\}_{i=1}^{L} \sim P\left(\mathbf{x}_k^p|\mathbf{Z}_{1:k}\right)^+$
    draw $\theta_j^p \sim P\left(\mathbf{x}_k^p|\mathbf{Z}_{1:k}\right)^+$
    if $(j - N_{bp})$ is an integer times of N_{si} **do**
      Extraction:
      accumulate weights $\{w_k^{p,(i)}\}_{i=1}^{L} = \{w_k^{p,(i)}\}_{i=1}^{L} + \{w_{k+}^{(i)}\}_{i=1}^{L}$
    end if
  end for
  normalise weights $\{w_k^{p,(i)}\}_{i=1}^{L}$
  if $\widehat{N_{eff}} < N_{thr}$ **do**
    resample with replacement L particles from $\{\mathbf{x}_k^{p,(i)}, w_k^{p,(i)}\}_{i=1}^{L}$ according to $\{w_k^{p,(i)}\}_{i=1}^{L}$
  end if
  save posterior particles $\{\mathbf{x}_k^{p,(i)}, w_k^{p,(i)}\}_{i=1}^{L}$ to Θ_k
end for

station for estimates update via a communication network. If the tracking is realised in a decentralised manner, each node keeps sharing its egocentric observations and pairwise measurements with its neighbours. In the meanwhile every node collects all kinds of observations from the rest of nodes through the network, and lastly achieves self-localisation using its own egocentric information and all available information it receives. In this case, only the node itself is treated as the primary node. The computation burden is hence distributed over all nodes in the field. As transmitted between

nodes is the observation data only, the communication cost in both centralised and decentralised frameworks would not be a concern.

6.4 Simulation Results

Simulations with linear/non-linear and Gaussian/non-Gaussian cases were performed to evaluate the performance of the proposed GSCPF approach. The performance is also compared to that of parametric approaches in terms of tracking accuracy and consistency.

6.4.1 Linear and Gaussian Cases

Setup

The demonstration of the algorithm in linear and Gaussian cases is done with an example of 7 mobile nodes moving within a 1D space. Only 4 of the mobile nodes have absolute position information available from a GPS sensor. Each mobile node is equipped with a radio device that enables communication with other nodes, and also provides a measure of relative distance between the nodes. Both the motion process and observation models are set to be linear and Gaussian to ensure that an optimal performance is produced by Kalman filter, which the proposed algorithm is compared with. Nevertheless, as a nonparametric approach, the algorithm could also be applied in systems presenting non-linear and non-Gaussian properties, which will be demonstrated in the next sections. Both the Kalman and the particle filter are tracking the state of each node which is composed of position and velocity in 1 dimension.

$$\mathbf{x} = \begin{bmatrix} r_x & v_x \end{bmatrix}^T$$

If we consider an individual mobile node, its state transition model from time step $k-1$ to k is interpreted by:

$$\mathbf{x}_k = \begin{bmatrix} 1 & \Delta T \\ 0 & 1 \end{bmatrix} \mathbf{x}_{k-1} + \begin{bmatrix} \frac{\Delta T^2}{2} \\ \Delta T \end{bmatrix} \mathbf{a}$$

where the random acceleration $\mathbf{a} \sim \mathcal{N}(0, \mathbf{Q})$ and ΔT is the interval between the two time steps.

Figure 6.5 illustrates the true trajectories of all mobile nodes moving with random accelerations in the 1D space from one of the investigated cases. The communication range between nodes is assumed long enough so that each node is able to communicate with everyone else at all times. However, an exception is the last node (in orange colour in the figure), which does not have egocentric observations and is intentionally disabled to communicate with those neighbours with egocentric information available. Therefore it is localised/tracked by information that is passed via at least 2-hops. The purpose of this configuration is to evaluate the performance of the algorithm in a special cooperative tracking scenario in which a position unknown node (i.e. the last node in the simulation) is localised/tracked using only relative information from other position unknown nodes.

In each of the simulation iterations, all absolute and relative observation generated within each mobile node is forwarded to a central base station node to facilitate global cooperative tracking of all nodes. This central node is not used as part of the localisation process, and is only used to collect observations of every mobile node for data fusion. To simplify the analysis, an assumption is made that each piece of information arrives at the central base station in real time. Detailed parameters used in the simulation are given in Table 6.4. These parameters were chosen to produce explicit results. Some location errors were intentionally added to every node in the initial state.

In order to quantify the tracking performance, the root mean squared error (RMSE) is used as a measure. The location RMSE of overall N_n mobile nodes is computed by:

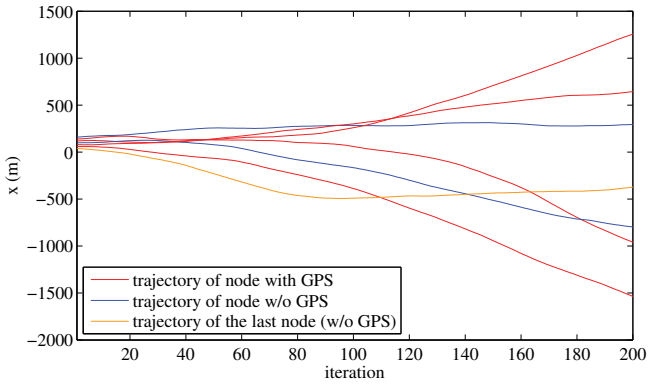


Figure 6.5 – True trajectories of mobile nodes. Each mobile node moves with a random acceleration in the 1D space.

$$RMSE = \sqrt{\frac{\sum_{p=1}^{N_n} \left(\hat{r}_x^p - r_x^p\right)^2}{N_n}}$$

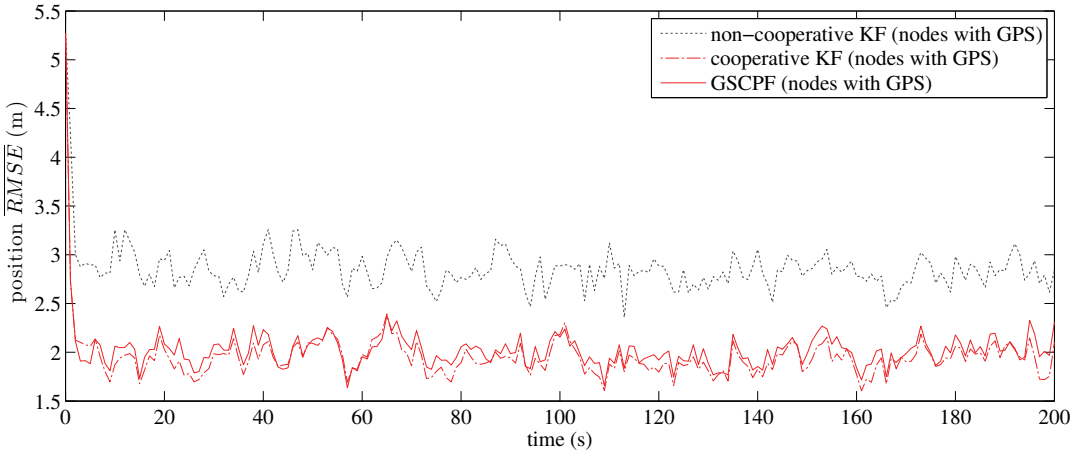
The RMSE value in velocity could be obtained in a similar way.

Results

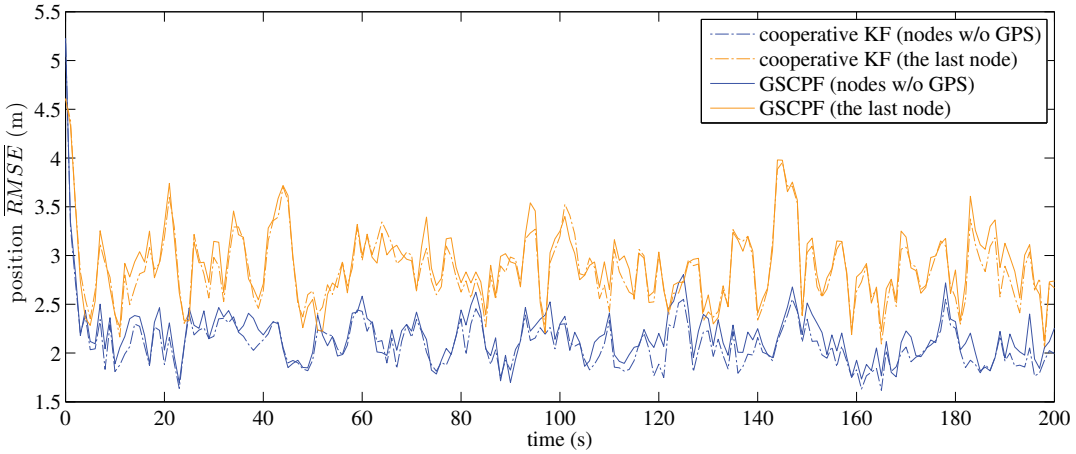
The tracking results of the parametric and nonparametric cooperative tracking approaches, i.e., KF and GSCPF respectively, are presented and compared in Figure

Table 6.4 – Simulation parameters

Parameter	Value
total Monte Carlo simulation cases	30
total iterations per case	200
total mobile nodes N_n	7
total mobile nodes with egocentric observation enabled	4
acceleration noise	$\mathcal{N}(0, 0.5^2)$
position observation noise $\mathbf{v}_{abs}$	$\mathcal{N}(0, 5^2)$
distance observation noise $\mathbf{v}_{rel}$	$\mathcal{N}(0, 10^2)$
particle quantity for primary node N_p	1000
particle quantity for auxiliary node N_a	500
burn-in length of Gibbs sampler N_{bp}	200
sampling interval of Gibbs sampler N_{si}	5
length of Gibbs chain N_{cl}	2000



(a) Mean position RMSE of nodes with GPS



(b) Mean position RMSE of nodes without GPS

Figure 6.6 – Mean RMSEs of position. The KF and GSCPF results are close in both (a) and (b). With cooperative tracking enabled, the relative observations are used to improve the tracking performance. This is particularly noticeable in the mobile nodes that do not have egocentric position observations from GPS. As a special case in nodes without GPS, the tracking result of the last node is illustrated in (b) separately. The tracking result of independent Kalman filter for nodes without GPS is not shown as the non-cooperative tracking algorithm fails in such a scenario.

6.6 in terms of mean RMSEs, which are calculated by averaging results of $N = 30$ Monte Carlo cases. As revealed in the figure, both cooperative algorithms can reduce the uncertainty of the mobile nodes with GPS by incorporating observations from other nearby nodes. The nodes without GPS only rely on position information of nearby nodes and relative measurements from them to get a position estimate. The

uncertainty of the absolute position in this case depends on the quality of the position information available to the GPS enabled neighbours as well as the relative ranging accuracy among them. As a special case set in the simulation, the last special node is tracked only by relative information from the other two GPS disabled nodes, which means that it is only allowed to use the third hand position information of these nodes with GPS enabled. This justifies the fact revealed in the figure that the tracking of the last node ends up with the largest errors.

To compare the tracking accuracy of the two tracking approaches, it is seen from the figure that GSCPF gives performance close to KF most of the time in position estimates for each type of nodes (with and without GPS and the last special node). We believe results with higher tracking accuracy are achievable with higher numbers of particles and/or longer lengths of Gibbs chain in the GSCPF. The tracking result from an independent KF where nodes do not communicate and are tracked independently is also presented. The advantage of cooperative tracking is clearly demonstrated in the diagram. Nodes without GPS are tracked cooperatively by nearby neighbours in the use of the cooperative tracking approaches, while the non-cooperative tracking algorithm fails in such a situation. In addition, estimates of the mobile nodes with GPS are also improved in cooperative tracking. This is manifestly because more information (both absolute and relative) is taken into account in the fusion stage, compared to the non-cooperative tracking scenario, where only absolute information is considered.

6.4.2 Linear and Non-Gaussian Cases

Setup

This section tests the proposed GSCPF with the presence of non-Gaussian noises in relative range observations. The purpose of the simulation is to evaluate the performance of KF and GSCPF when the Gaussian assumption is not kept valid any more, which is true for most applications in the real world. Of various kinds of candidates, a Student's t-distribution was chosen as the non-Gaussian distribution

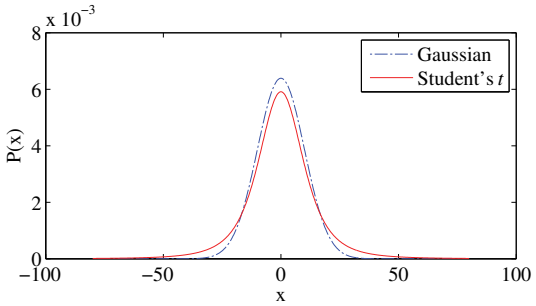


Figure 6.7 – Gaussian and non-Gaussian distributions. The blue curve demonstrates a zero-mean Gaussian distribution described by $\mathcal{N}\left(0, \sigma^2\right)$ where $\sigma = 10$. The red shows a Student's t-distribution as an example of non-Gaussian distribution. The distribution is similar to the Gaussian but comparatively heavier-tailed. It is formulated by $St\left(0, \lambda, \nu\right)$ where $\nu = 3$ is degrees of freedom, and the precision $\lambda = \frac{1}{\sigma^2}\frac{\nu}{\nu-2} = 0.03$. Along with the growing of ν to ∞, the Student's t-distribution approaches the Gaussian one.

used in the simulation. Though of similar shape to Gaussian distributions, it is heavier-tailed, which means that it is more likely to produce samples that fall far from its mean. When $\nu \to \infty$ is fulfilled, the Student's t-distribution turns to be Gaussian.

$$\lim_{\nu\to\infty} St\left(\mu, \lambda, \nu\right) = \mathcal{N}\left(\mu, \sigma^2\right)$$

where μ is mean; λ is precision; ν is degrees of freedom; and variance $\sigma^2 = \frac{1}{\lambda}\frac{\nu}{\nu-2}$ $(\nu > 2)$.

Simulation cases were created with relative observations corrupted by noises that follow the Student's t-distribution in Figure 6.7. And in the simulation, the GSCPF would use the same Student's t-distribution parameters to build up its observation model, while the KF uses the Gaussian in the figure to approximate the Student's t-distribution. The approximation is somewhat over-confident on excluding probabilities on the tails of the Student's t. However the approximation could never be perfect using only one Gaussian, as it turns to be over-pessimistic if a larger variance is used in the Gaussian. The rest of the components in the simulation such as node motion model and absolute observation were kept the same, i.e. linear and Gaussian, to the simulation conducted in Section 6.4.1. The purpose of this is to make sure that it is only relative observation models that would account for the difference in

tracking results of the two approaches.

Consistency Test

In addition to the RMSE which is to evaluate the tracking accuracy, normalised estimation error squared (NEES) is adopted as the metrics of consistency of the two tracking approaches in the simulation.

According to the definition in [21], a filter is considered "consistent" if its estimation errors are consistent with their statistical properties, i.e.,

1. Have mean zero, (i.e., unbiased estimates).
2. Have covariance matrix as calculated by the filter.

Respectively, they correspond to two equations:

$$E\left[\hat{\mathbf{x}}_k - \mathbf{x}_k\right] = 0$$

$$E\left[\left(\hat{\mathbf{x}}_k - \mathbf{x}_k\right)\left(\hat{\mathbf{x}}_k - \mathbf{x}_k\right)^T\right] = \mathbf{P}_k$$

where $\hat{\mathbf{x}}_k$ is the state estimate at time k, $\mathbf{x}_k$ is its true value, and $\mathbf{P}_k$ is its associated covariance calculated by the filter.

The NEES value for a given mobile node at time k is calculated by:

$$\epsilon\left(k\right) = \left(\hat{\mathbf{x}}_k - \mathbf{x}_k\right)^T \mathbf{P}_k^{-1}\left(\hat{\mathbf{x}}_k - \mathbf{x}_k\right)$$

The consistency test is based on averaging NEES results over N Monte Carlo runs of the filters, which yields:

$$\bar{\epsilon}\left(k\right) = \frac{1}{N}\sum_{i=1}^{N}\epsilon^i\left(k\right)$$

Then the $N\bar{\epsilon}(k)$ will have a χ^2 (chi-square) distribution with $Ndim(\mathbf{x}_k)$ degrees of freedom [21] [15], under the hypothesis that the tested filter is consistent and approximately linear and Gaussian. The state estimation errors are considered consistent with the filter-calculated covariances if:

$$\bar{\epsilon}(k) \in [r_1, r2]$$

where the interval $[r_1, r2]$ bounds the two-sided 95% probability concentration region and is calculated by:

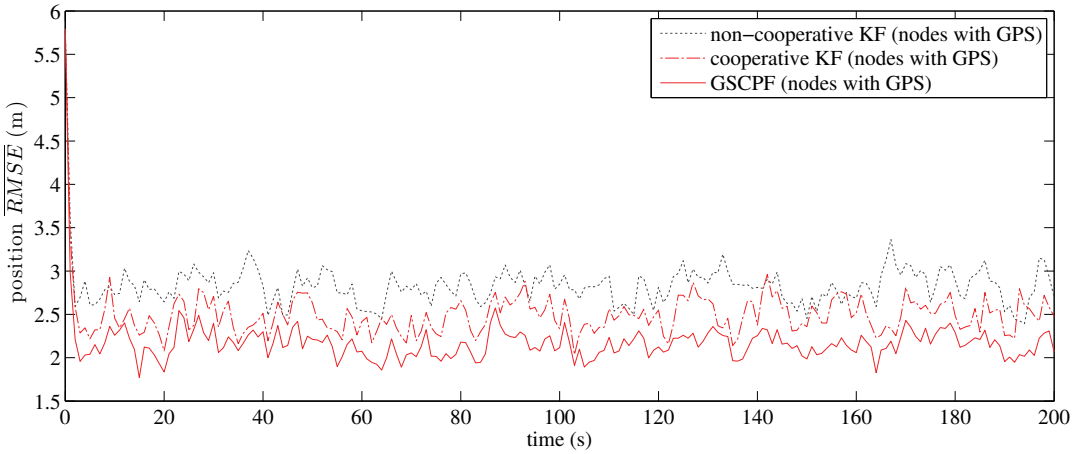
$$\left[\frac{\chi^2_{Ndim(\mathbf{x}_k)}(0.025)}{N}, \frac{\chi^2_{Ndim(\mathbf{x}_k)}(0.975)}{N}\right]$$

A filter tends to produce optimistic estimates if the $\bar{\epsilon}(k)$ rises significantly higher than the upper bound, while if it stays below the lower bound for a majority of time, the filter is considered conservative [17].

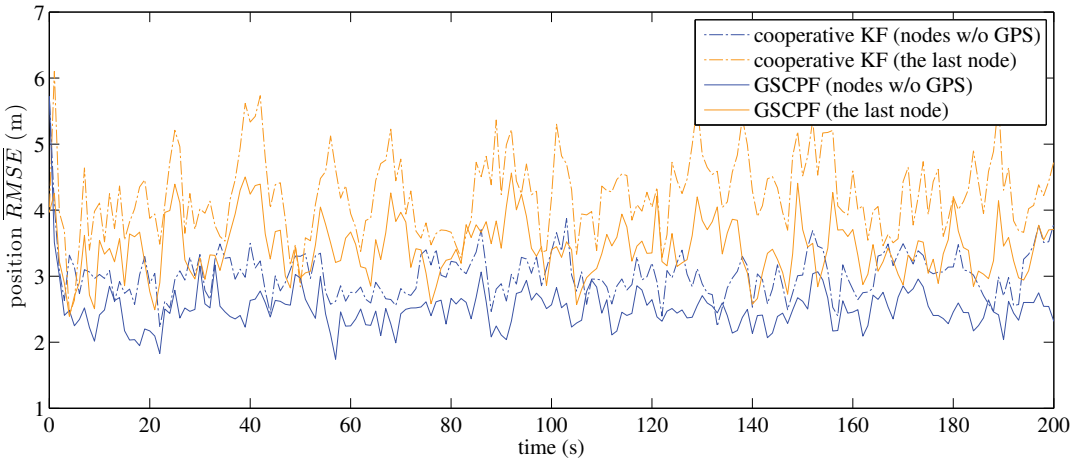
Results

The average RMSEs of the two filters, KF and GSCPF, are presented and compared in Figure 6.8. The average NEESs of the last node produced by the two filters could be found in Figure 6.9. These results were calculated by averaging outcomes of $N = 30$ Monte Carlo cases.

In terms of tracking accuracy according to the figure, apparently GSCPF demonstrates superior performance compared with that of KF in position estimation for each type of nodes (with and without GPS and the last node). As revealed by Figure 6.9, the GSCPF also outperforms KF in terms of estimation consistency. The mean NEES result of the GSCPF stays inside the 95% probability concentration region for the most of the simulation iterations. The mean NEES curve of the KF, however, keeps higher than the upper bound for a majority of the time. This indicates that the GSCPF is able to produce consistent tracking results, while the KF tends to be



(a) Mean position RMSE of nodes with GPS



(b) Mean position RMSE of nodes without GPS

Figure 6.8 – Mean RMSEs of position. The mean RMSE results of GSCPF outperform KF's in both (a) nodes with GPS and (b) nodes without GPS available. Note that the tracking result of independent Kalman filter is not affected by the non-Gaussian distributed noises in relative ranging observations, which do not exist in non-cooperative tracking scenarios. The independent KF tracking result for nodes without GPS is not shown in (b) as the independent tracking algorithm fails in such scenario.

optimistic in estimation. The comparisons in NEES for other nodes are not illustrated as they have the similar result to that in Figure 6.9.

To conclude, with the existence of the particular type of non-Gaussian noises (i.e. Student's t-distribution) in relative ranging observations in the simulation, the GSCPF performs better than KF in terms of both tracking accuracy and consistency. It is

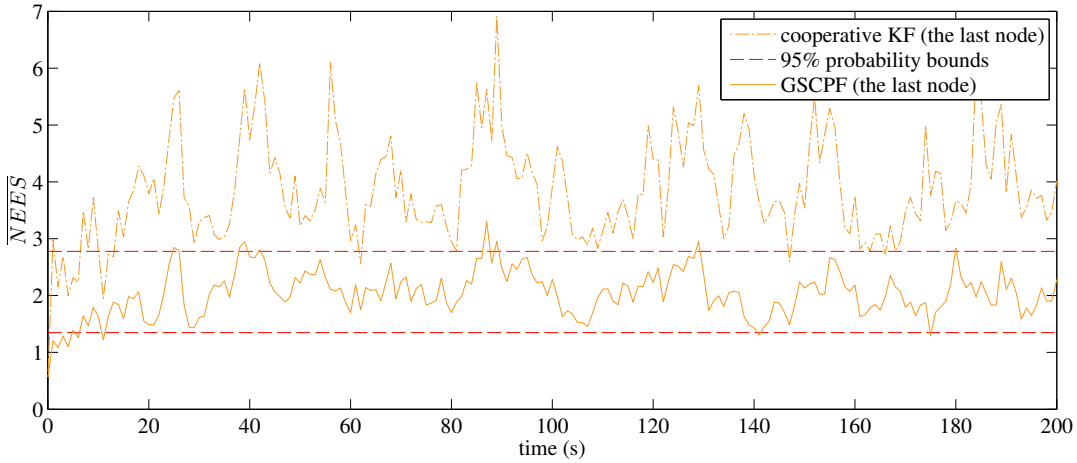


Figure 6.9 – NEES consistency test for the tracking of the last node. The in-bound rate of the GSCPF tracking result is about 91.04%, while for a majority of the time the KF result is out of the 95% probability bounds.

believed that the KF would perform even worse in tracking cases where more components present non-Gaussian properties or the existing non-Gaussian distribution is of lower similarity to Gaussian, such as triangular, rectangular, multi-modal and so on. Some methods, e.g. Gaussian mixtures [28], are used to approximate the non-Gaussian distribution in KF and its variants. For instance, the Student's t-distribution used in the simulation could be approximated by adding up a number of Gaussian distributions with the same mean but different variances. However, as a nonparametric approach, the GSCPF deals with the cooperative tracking problems with non-Gaussian distributions naturally and more precisely, and is certainly preferred in such types of applications.

6.4.3 Non-Linear and Gaussian Cases

Setup

To evaluate the performance of cooperative tracking algorithms in Gaussian but non-linear scenarios, a simulation was setup with 7 mobile nodes moving randomly in a 2D field. The trajectories of these nodes are illustrated in Figure 6.10. Four of them have GPS device fitted on-board while the rest three do not. The system's non-linearity

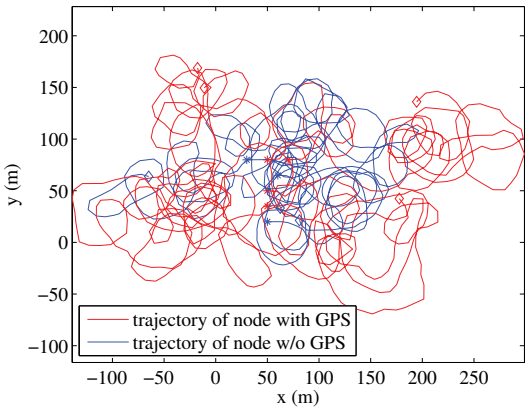


Figure 6.10 – True trajectories of mobile nodes. Every mobile node moves with random linear velocity and steering angle in the 2D area.

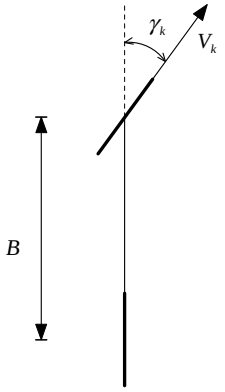


Figure 6.11 – Bicycle-like vehicle motion model.

comes from the node kinematic model and relative distance measuring model. The motion of each node could be described by a bicycle-like vehicle moving with random steering angle and rolling speed on ground, see Figure 6.11. Therefore we have the state of each node represented by:

$$\mathbf{x} = \begin{bmatrix} r_x & r_y & \theta \end{bmatrix}^T$$

where θ is the orientation of the node respect to the x axis.

Assuming that the motion control inputs are constant over the interval ΔT between two time steps, we have a discrete time vehicle model of the node from time step $k-1$ to k formulated by:

$$\mathbf{x}_k = f\left(\mathbf{x}_{k-1}, \mathbf{u}_k\right) = \mathbf{x}_{k-1} + \begin{bmatrix} \Delta T V_k cos\left(\theta_{k-1} + \gamma_k\right) \\ \Delta T V_k sin\left(\theta_{k-1} + \gamma_k\right) \\ \Delta T V_k \frac{sin(\gamma_k)}{B} \end{bmatrix}$$

where V_k represents the linear velocity of the front wheel on ground; γ_k refers to the steering angle; B is the wheelbase between axles of the front and rear wheels.

Having been added with process noises, the control inputs $\mathbf{u}_k$ could be expanded to:

$$\mathbf{u}_k = \begin{bmatrix} V_k \\ \gamma_k \end{bmatrix} = \begin{bmatrix} V_k^{true} \\ \gamma_k^{true} \end{bmatrix} + \begin{bmatrix} \mathbf{w}_V \\ \mathbf{w}_\gamma \end{bmatrix}$$

where V_k^{true} and γ_k^{true} respectively are true inputs of linear velocity and steering angle, and $\mathbf{w}_V$ and $\mathbf{w}_\gamma$ are zero-mean Gaussian noises on them respectively. For simplicity, we assume in the simulation that the control inputs for each node keep constant.

The kinematic model has to be linearised when these nodes are tracked in the use of EKF. For node p, we have the linearised motion model written as:

$$\begin{aligned} \boldsymbol{\delta}\mathbf{x}_k^p \approx & \begin{bmatrix} 1 & 0 & -\Delta T V_k^p sin\left(\hat{\theta}_{k-1}^p + \gamma_k^p\right) \\ 0 & 1 & \Delta T V_k^p cos\left(\hat{\theta}_{k-1}^p + \gamma_k^p\right) \\ 0 & 0 & 1 \end{bmatrix} \boldsymbol{\delta}\mathbf{x}_{k-1}^p \\ & + \begin{bmatrix} \Delta T cos\left(\hat{\theta}_{k-1}^p + \gamma_k^p\right) & -\Delta T V_k^p sin\left(\hat{\theta}_{k-1}^p + \gamma_k^p\right) \\ \Delta T sin\left(\hat{\theta}_{k-1}^p + \gamma_k^p\right) & \Delta T V_k^p cos\left(\hat{\theta}_{k-1}^p + \gamma_k^p\right) \\ \Delta T \frac{sin\left(\gamma_k^p\right)}{B} & \Delta T V_k^p \frac{cos\left(\gamma_k^p\right)}{B} \end{bmatrix} \begin{bmatrix} \mathbf{w}_V \\ \mathbf{w}_\gamma \end{bmatrix} \end{aligned}$$

In addition, a linearisation process of the non-linear relative ranging model between two neighbouring nodes p and q in 2D space is required.

$$h_k^{p \to q}\left(\mathbf{X}_k\right) = \sqrt{\left(r_x^p - r_x^q\right)^2 + \left(r_y^p - r_y^q\right)^2}$$

$$\nabla \mathbf{H}^{p\to q}_{\hat{\mathbf{x}}^p_k} = \begin{bmatrix} \frac{\hat{r}^p_x - \hat{r}^q_x}{d^{p\to q}_k} & \frac{\hat{r}^p_y - \hat{r}^q_y}{d^{p\to q}_k} & 0 & 0 \end{bmatrix}$$

$$\nabla \mathbf{H}^{p\to q}_{\hat{\mathbf{x}}^q_k} = \begin{bmatrix} \frac{\hat{r}^q_x - \hat{r}^p_x}{d^{p\to q}_k} & \frac{\hat{r}^q_y - \hat{r}^p_y}{d^{p\to q}_k} & 0 & 0 \end{bmatrix}$$

where $d^{p\to q}_k = \sqrt{(\hat{r}^p_x - \hat{r}^q_x)^2 + (\hat{r}^p_y - \hat{r}^q_y)^2}$.

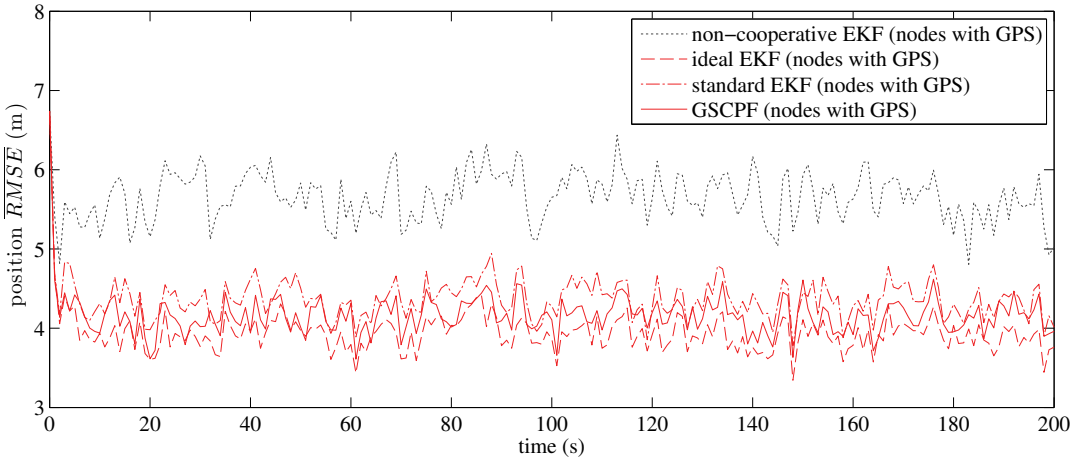
The relative range observations also contain zero-mean Gaussian noises denoted by v_{rel}.

The trajectories of all 7 nodes in one of the simulation cases are depicted in Figure 6.10. In Table 6.5 presented are detailed parameters used in the simulation. The constant true linear velocity was set to be $10m/s$, which represents a typical speed of haul trucks.

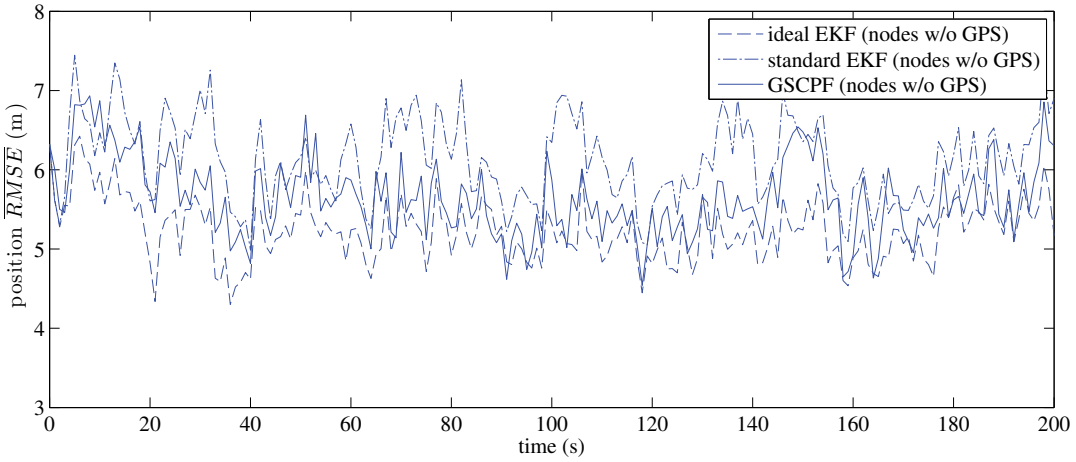
Table 6.5 – Simulation parameters

Parameter	Value
total Monte Carlo simulation cases	30
total iterations per case	200
total mobile nodes N_n	7
total mobile nodes with egocentric observation enabled	4
constant true linear velocity V	$10m/s$
linear velocity noise $\mathbf{w}_V$	$\mathcal{N}(0, 2^2)$
constant true steering angle γ	2° or -2°
steering angle noise $\mathbf{w}_\gamma$	$\mathcal{N}(0, 2.5^2)$
position observation noise $\mathbf{v}_{abs}$ in x and y	$\mathcal{N}\left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 5^2 & 0 \\ 0 & 5^2 \end{bmatrix}\right)$
distance observation noise $\mathbf{v}_{rel}$	$\mathcal{N}(0, 10^2)$
particle quantity for primary node N_p	2000
particle quantity for auxiliary node N_a	500
burn-in length of Gibbs sampler N_{bp}	200
sampling interval of Gibbs sampler N_{si}	5
length of Gibbs chain N_{cl}	2000

Results



(a) Mean position RMSE of nodes with GPS



(b) Mean position RMSE of nodes without GPS

Figure 6.12 – Mean RMSEs of position. The mean RMSE results of GSCPF outperform EKF's in both (a) nodes with GPS and (b) nodes without GPS available. The ideal EKF performs the best given the knowledge of true states. a) also demonstrates the advantage of cooperative tracking algorithms in improving tracking accuracy of nodes with GPS when their results are compared to that of the independent EKF. The independent EKF tracking result for nodes without GPS is not presented in (b) as the algorithm fails to track in such scenario.

By averaging results from $N = 30$ Monte Carlo simulation cases, the proposed GSCPF algorithm is compared with ideal and standard EKFs in Gaussian but non-linear scenarios. The difference between the two EKFs comes from the different choices of linearisation points. The standard EKF in the simulation performs linearisation at the point of last state estimates and current state predictions in the prediction

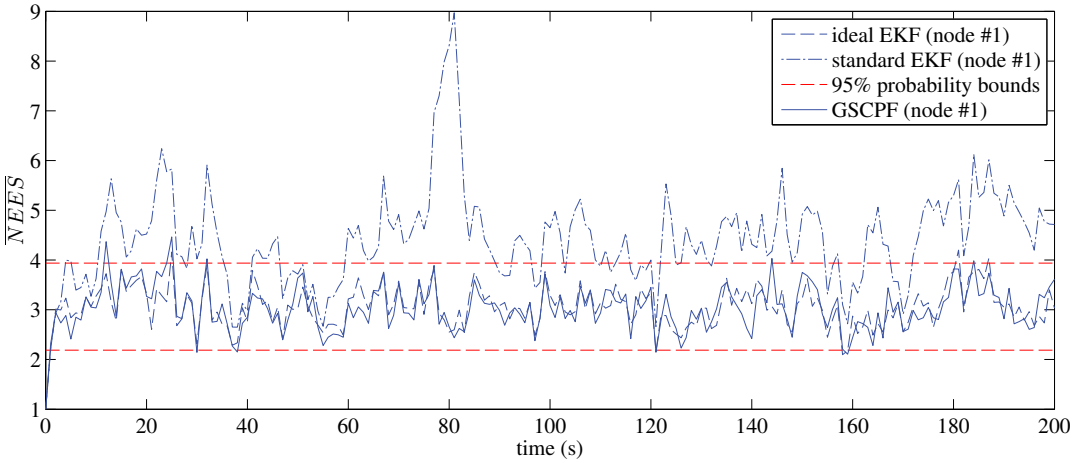


Figure 6.13 – NEES consistency test for the tracking of node #1. For 94.53% of the all simulated iterations the NEES of the GSCPF tracking result stays in the theoretical 95% probability bounds. The ideal EKF presents a close in-bound rate to the GSCPF, while the value for the standard EKF is about 26.37%.

and update stages respectively, while the ideal does it accurately with true values of the last and current states. Apparently the standard EKF has errors introduced in during linearisation processes, thereby generally producing less precise results. Under some particular circumstances the standard EKF easily ends up with inconsistency in tracking due to accumulated errors [17]. All random variables in the simulation cases were set to be Gaussian, so that it is only the linearisation errors that lead to degraded tracking accuracy of the standard EKF. The purpose of introducing the ideal EKF into the simulation is to demonstrate how much the standard EKF result is affected by the errors from linearisation.

Comparisons of the mean position RMSEs are made among the three cooperative tracking algorithms, ideal and standard EKFs and the proposed GSCPF. Respectively the tracking results for nodes with and without GPS could be found in Figures 6.12(a) and 6.12(b). In both figures clearly we observe that the ideal EKF performed the best, and the GSCPF outperformed the standard EKF. Though the results of the ideal EKF and GSCPF are shown in the same figures, there is no comparability between them, as the ideal EKF is provided with the knowledge of true states, which is not available in practical implementation. The NEES test results illustrated in Figure 6.13 also suggest a better consistency of the ideal EKF and GSCPF than that of the

standard EKF. In Figure 6.12(a) the tracking result of the non-cooperative EKF is presented as well, to again demonstrate the advantage of the cooperative algorithms in improving tracking of nodes with GPS. The non-cooperative EKF is unable to track these GPS disabled nodes, therefore its result is not shown for comparison in Figure 6.12(b).

6.4.4 Effects of Parameters in the Gibbs Sampler

There are three important parameters in the Gibbs sampler: the Gibbs chain length N_{cl}, the Gibbs sampling interval N_{si}, and the effective sample size N_{es}. Their relationship could be revealed by:

$$N_{es} = \frac{N_{cl}}{N_{si}}$$

This section is going to evaluate the effects of these parameters on the tracking accuracy.

Figure 6.14(a) compares the tracking performance with different N_{si} with a fixed N_{es}. As seen in the figure, the average RMSE descends along with the increase of N_{si}. This is apparently because of a lower autocorrelation between samples when a larger N_{si} is used. However, a larger N_{cl} and thus a higher computational cost are required. When N_{si} is not changed, the tracking accuracy is improved with the ascent in N_{cl} due to a greater N_{es}, see Figure 6.14(b).

When the length of Gibbs chain N_{cl} is fixed, it is seen from Figure 6.14(c) that the average RMSE reaches a trough at the point $N_{si} = 10$. This phenomenon could be explained by a combined effect of N_{si} and N_{es} shown in Figures 6.14(a) and 6.14(b), respectively. Before the point $N_{si} = 10$, the result is dominated by the improvement of accuracy brought by a larger N_{si}. However after this point, the degradation in accuracy due to a smaller N_{es} becomes manifest and dominates the result.

In addition, the increase in the particle quantity for primary nodes contributes to the improvement of tracking accuracy. This can be revealed in Figure 6.14(d).

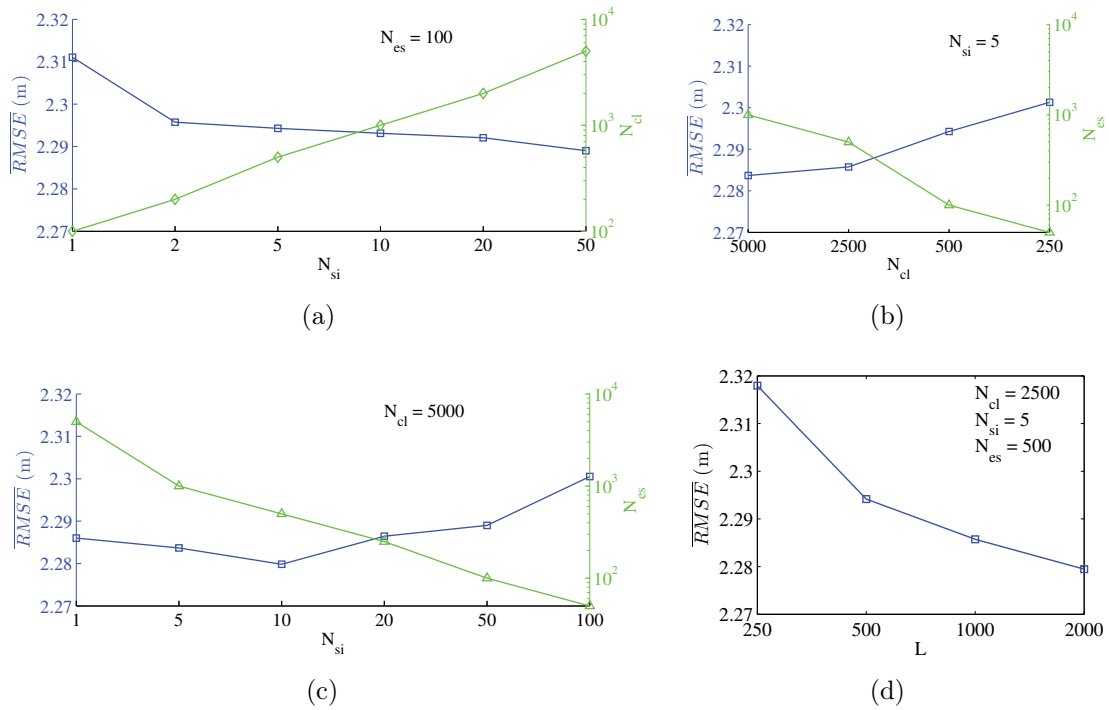


Figure 6.14 – Average position RMSEs with different parameters. (a) shows the decrease in average RMSE along with the ascent of N_{si}. The RMSE is also reduced when a larger N_{cl} is used, see (b). In (c), the average RMSE decreases before the point $N_{si} = 10$ and increases with a larger N_{si}. The x-axis in (b) is intentionally inverted in order to help understand the (c). (d) illustrates the decrease of RMSE when a larger quantity of particles is adopted for primary nodes.

It can be concluded that, a reasonable value of N_{si}, say, 5 or 10, is recommended for a trade-off between the tracking accuracy and the computational cost. Numbers in these figures were achieved by averaging tracking results of 7 nodes (4 nodes with location known and 3 unknown) from 50 Monte Carlo simulation cases.

6.5 Cooperative Tracking in Mobile Ad-Hoc Networks

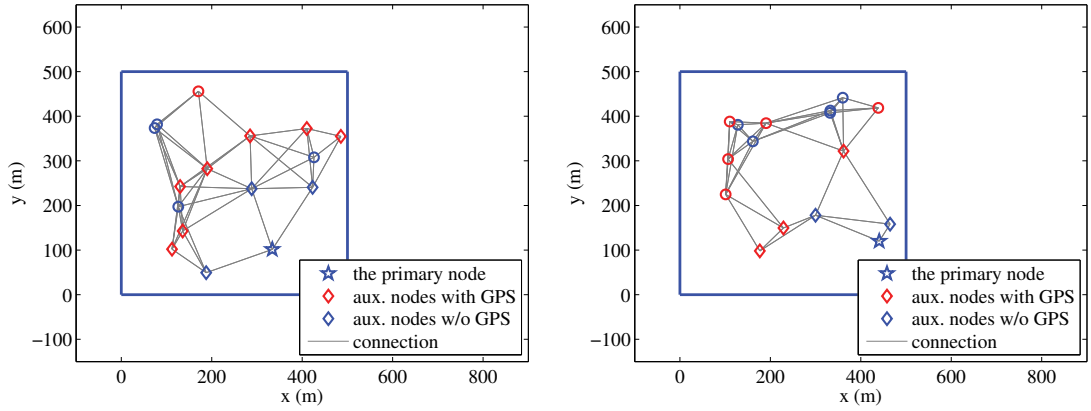
A MANET is a spontaneous network formed by a number of mobile nodes equipped with communication devices and no infrastructure is required. In a MANET, a node is only able to communicate with its instant neighbours that are within its com-

munication range. The MANET hence features multi-hop and P2P connections. A detailed description of the properties of this kind of network has been presented in Section 2.5.2. As nodes are moving around in the field, the network topology ever keeps changing, thereby a node might stay in sight of another for an unpredictable period of time. The tracking problem is therefore becoming more complicated due to the dynamics of network topology.

So far, the proposed GSCPF approach has been tested, throughout the previous several simulations, in tracking mobile nodes in a fully connected network, where nodes are assumed to be capable of communicating with each other all the time. The network environment is completely changed in a MANET, which is spontaneously and dynamically formed by a group of mobile nodes with limited communication ranges to their instant neighbours. Thus, the naive concept of the approach now should be optimised to fit the cooperative tracking problem in MANETs. As long as every node gets connected into the network, a node i is an N-hop ($N \geq 1$) neighbour of another arbitrary node j. Literally node i would help track the node j if it contains useful location information, though the information gets diluted along with the number of hops of transmission due to uncertainties introduced in relative ranging measurements. As a matter of fact, the gain from the information tends to be negligible when N is large. For this reason, in order to reach a balance between tracking accuracy and problem complexity, we would consider only a fraction of all other nodes as the auxiliary nodes of the node i in GSCPF algorithm. These nodes within a given threshold of hops of communication would be taken into account, while a node without new location information should be removed from the list of auxiliary nodes unless it acts as a medium relaying information for other auxiliary nodes.

6.5.1 Selection of Informative Auxiliary Nodes from Neighbours

Based on the previous considerations, the algorithm considers a node as an informative auxiliary when one of the following criteria is met.

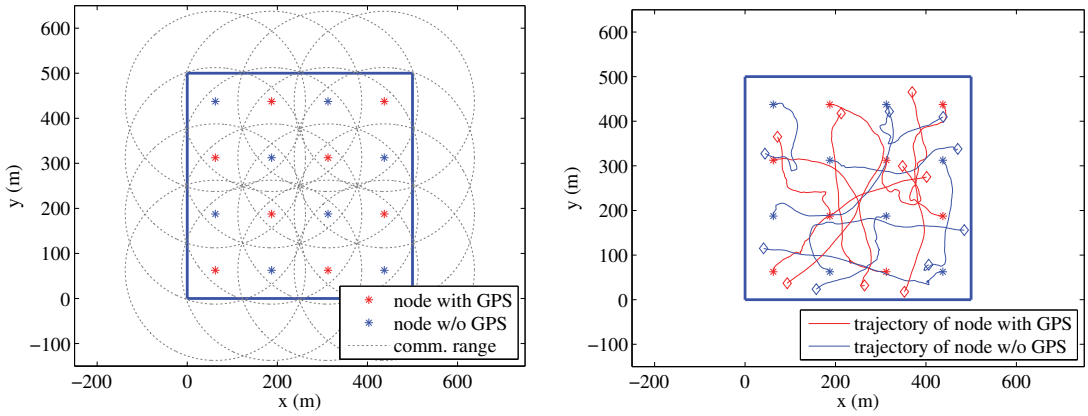


(a) Example 1 (b) Example 2

Figure 6.15 – Two examples of informative auxiliary nodes. In both (a) and (b), the nodes with GPS enabled are in red, while those without GPS information are in blue.

- It is a 1-hop neighbour with GPS enabled, as it is informative to the primary node.
- It is an informative 1-hop neighbour without GPS information, when it connects to at least one node that has GPS enabled and therefore could provide second-hand position information.
- It is a 2-hop neighbour with GPS enabled and connected by any of informative 1-hop neighbours, i.e. those falling in one of above two.

Those unfortunately not meeting the criteria, for example those nodes beyond 2-hop far away, are not taken into account as auxiliary nodes in the optimised GSCPF algorithm when it is applied in a MANET. Note that the selection process of auxiliary nodes is performed for each primary node at every time step, therefore each will have its own dynamic list of auxiliary nodes according to the network topology at the moment. At each time step, the GSCPF algorithm executes Gibbs sampling individually for every primary node according to its current list of auxiliary nodes. See Figure 6.15 for two examples of informative auxiliary nodes selected for two primary nodes respectively.



(a) Initial positions (b) True trajectories

Figure 6.16 – Initial positions and true trajectories of mobile nodes. (b) Each mobile node moves with random accelerations within the square field, with starting point represented by asterisk and ending point denoted by diamond shape. Each dashed circle represents the coverage of wireless communication of a node.

6.5.2 Simulation of Non-Linear and Non-Gaussian Cases

Setup

The algorithm in a MANET was tested with cases where 16 mobile nodes move within a square field with a dimension of $500m \times 500m$. Only half of the mobile nodes have absolute position information provided by on-board GPS sensors. The radio communication device mounted on each mobile node enables a node to communicate with others within a fixed range, and in the meanwhile measures relative range between the nodes with some noise. A central base station is adopted, the same way as how it was used in Section 6.4.1, to collect real-time observations from each mobile node via multi-hop communication. It is at the central station where the global tracking of all nodes takes place.

Initially, 16 mobile nodes are positioned in the form of a 4×4 array with even intervals. The communication device fitted in each mobile node is used to communicate between neighbouring nodes and provide distance measurements as shown in Figure 6.16(a). Trajectories of nodes are shown in Figure 6.16(b). An assumption is made that both

absolute and relative observations are received by the central base station in real time. The extended Kalman filter (EKF) is used as a comparison to the proposed GSCPF algorithm in the simulation. Both filters are tracking the state of each node which is composed of positions and velocities in two dimensions.

$$\mathbf{x} = \begin{bmatrix} r_x & r_y & v_x & v_y \end{bmatrix}^T$$

As the EKF is adopted, the non-linear relative ranging model between two neighbouring nodes p and q in 2D space is linearised as previously described in Section 6.4.3.

The relative distance observations were corrupted by noises that follow the same Student's t-distribution in Figure 6.7. In its linearised observation model, the EKF uses the Gaussian in the figure to approximate the Student's t-distribution. Detailed parameters used in the simulation could be found in Table 6.6. The particle filter was initialised with 2000 particles for each primary node, and 1000 for auxiliary nodes. Some location errors were intentionally added to every node in the initial state.

Table 6.6 – Simulation parameters

Parameter	Value
total Monte Carlo simulation cases	30
total iterations per case	200
total mobile nodes N_n	16
total mobile nodes with egocentric observation enabled	8
acceleration noise in x and y	$\mathcal{N}\left(\begin{bmatrix}0\\0\end{bmatrix}, \begin{bmatrix}0.2^2 & 0\\0 & 0.2^2\end{bmatrix}\right)$
position observation noise $\mathbf{v}_{abs}$ in x and y	$\mathcal{N}\left(\begin{bmatrix}0\\0\end{bmatrix}, \begin{bmatrix}5^2 & 0\\0 & 5^2\end{bmatrix}\right)$
distance observation noise $\mathbf{v}_{rel}$ used in EKF	$\mathcal{N}\left(0, 10^2\right)$
distance observation noise $\mathbf{v}_{rel}$ used in GSCPF	$St\left(0, 0.03, 3\right)$
initial position interval	125 m
P2P communication range	200 m
particle quantity for primary node N_p	2000
particle quantity for auxiliary node N_a	1000
burn-in length of Gibbs sampler N_{bp}	200
sampling interval of Gibbs sampler N_{si}	5
length of Gibbs chain N_{cl}	2000

To quantify tracking performance, the RMSE is again used as a measure. The RMSE of location of overall N_n mobile nodes in a 2D plane is computed by:

$$RMSE = \sqrt{\frac{\sum_{p=1}^{N_n} \left(\hat{r}_x^p - r_x^p\right)^2 + \left(\hat{r}_y^p - r_y^p\right)^2}{N_n}}$$

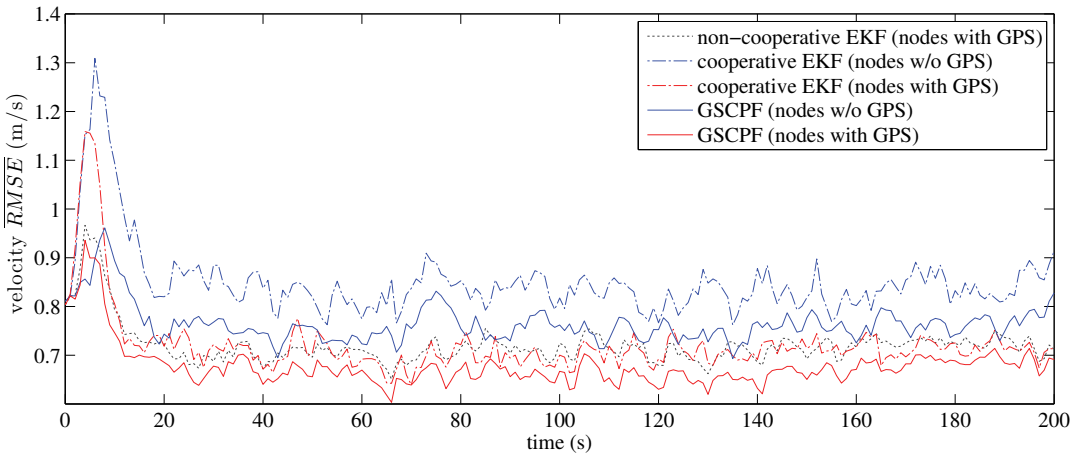
The similar way is used to calculate the RMSE in velocity.

Results

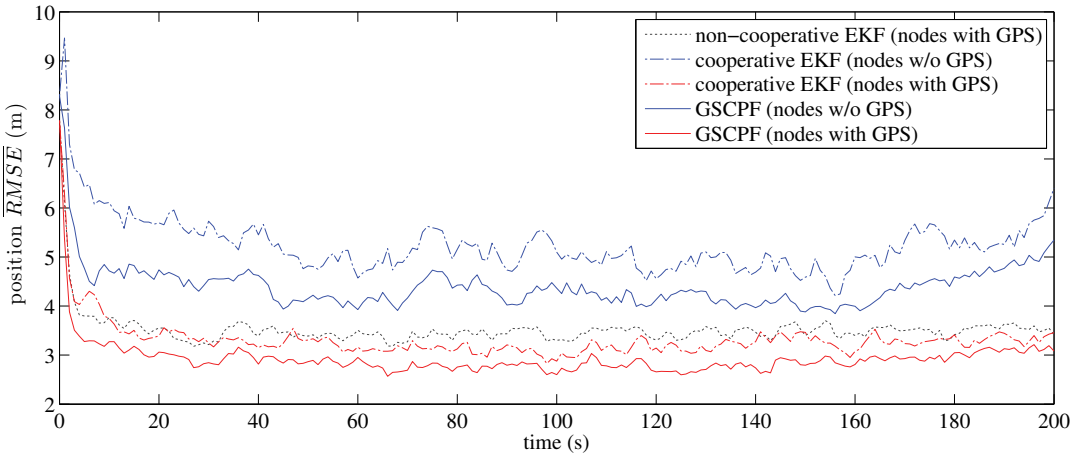
The GSCPF algorithm performs better than the EKF in both position and velocity estimation in the 2D MANET cases according to the simulation results illustrated in Figure 6.17. In all $N = 30$ Monte Carlo cases tested, the EKF performed the estimation given full knowledge of observations generated about every mobile node. On the other hand, the GSCPF tracked these nodes only based on a subset of all available information, which was automatically picked according to the informative auxiliary nodes selection mechanism presented in Section 6.5.1. Although the mechanism aims to reduce computational burden by excluding those neighbours not considered sufficiently informative, undeniably it results in a sacrifice in tracking accuracy to some extent due to an information loss.

Nevertheless, it is observed in Figure 6.17 that generally the GSCPF tracked nodes more accurately than the EKF in the simulation. In these non-linear and non-Gaussian testing cases, errors are introduced into the EKF during approximation of non-Gaussian noises and linearisation of the inter-node observation model. We believe that the errors accounted for an even larger degradation of performance in the EKF compared to what the information loss did in the GSCPF.

Furthermore, the results of NEES shown in Figure 6.18 also suggest better estimation consistency of the GSCPF algorithm than the EKF under the particular simulation environment. Only the result for one of the nodes is illustrated in the figure, as the similar result could be concluded about every other node. As previously mentioned, the Student's t-distribution is more likely to generate samples that are far away from



(a) Mean velocity RMSE



(b) Mean position RMSE

Figure 6.17 – Mean RMSEs of velocity and position. The mean RMSE results of GSCPF outperform EKF's in both (a) velocity and (b) position. The tracking results of independent EKF for nodes without GPS are not shown as the independent tracking algorithm fails in such scenarios.

its mean because of its "longer tails" than its Gaussian approximation. However, the EKF is sensitive to the presence of observations that are considered "outliers", i.e. those fall further than what its Gaussian observation model predicts, thereby an over-optimistic tracking is observed for the EKF in Figure 6.18. This is not an issue to the GSCPF as it has the real distribution of noises in observation more precisely modelled. The Student's t-distribution in the simulation is only used as an example out of various types of non-Gaussian noises, which exist ubiquitously in the real world.

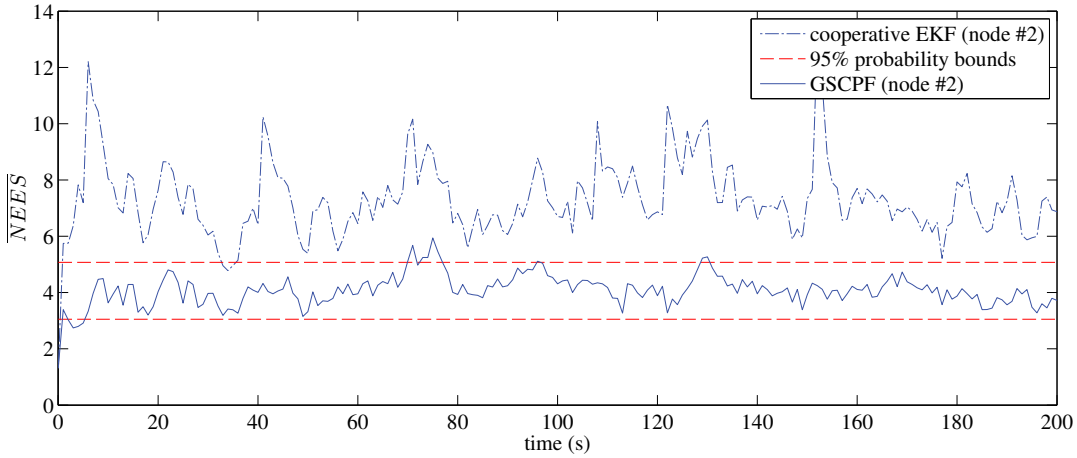


Figure 6.18 – NEES consistency test for the tracking of node #2. The in-bound rate of the GSCPF tracking result is about 93.03%, while most of the time the EKF result is out of the 95% probability bounds.

From this point of view, the GSCPF, as a nonparametric approach, tends to be of higher robustness than its parametric counterpart in practical implementation.

6.6 Summary

This chapter has presented a particle filter to track multiple mobile nodes in a cooperative manner. Relative range information is shared among nodes and forwarded to a central base station. The base station updates the estimate of nodes using the relative distance information between nodes, together with egocentric position information from the mobile nodes that are equipped with GPS sensors.

Simulation tests were performed in both fully connected networks and MANETs. The results show that it is possible to obtain accurate and consistent estimates of multiple nodes using the proposed GSCPF. In MANETs, P2P communication enables cooperative tracking of multiple nodes without a global, full coverage network. The results showed that the approach facilitates the tracking of mobile nodes which do not have egocentric position information. In addition, improvements in the tracking of mobile nodes that do have an egocentric position were demonstrated when compared to the independent tracking approaches. When the proposed algorithm is compared with

parametric approaches such as KF and EKF, it demonstrates superior performance when they are tested in non-linear and/or non-Gaussian systems. In scenarios assumed to be linear and Gaussian, the GSCPF presents close performance to the KF, which gives optimal results under this circumstance.

The next chapter is going to extend the GSCPF algorithm proposed in this chapter with the delayed-state concept introduced. The delayed-state approaches are preferred in tracking problems in large environments as they allow time delays in arriving observations. An approach namely DSCPF is proposed and applied to a fleet of mining vehicles in a large mining environment. This involves the cooperative tracking of multiple vehicles with highly non-linear motion dynamics and non-Gaussian noise in the relative position observations.

Chapter 7

Cooperative Tracking with Delayed Observations

7.1 Introduction

The nonparametric cooperative tracking approach proposed in the previous chapter is extended toward delayed-state cooperative filtering in this chapter. The new proposed approach is capable of fusing delayed inter-vehicle relative information harvested to further constrain the estimates of vehicle positions. The research work presented in this chapter completes the main contributions of the thesis. Please refer to Appendix E for the background of parametric delayed-state filtering approaches. The remainder of this chapter is organised as below. Section 7.2 defines how delayed observations are represented in the base station before they are used. Then the Bayesian formulation of the delayed-state cooperative filtering is elaborated in Section 7.3. It is followed by Section 7.4, which introduces the proposed particle filter in details. Simulation results are presented in Section 7.5, while experiments results are available in Section 7.6. Finally the work presented in this chapter is summarised in Section 7.7.

7.2 Keeping Delayed Observations

In **Chapter 5**, $\mathbf{\Omega}_k^V$ and $\mathbf{\Omega}_k^{V\to I}$ respectively have been defined as the global vehicle egocentric observation pool and global V2I observation pool in the base station. In this chapter, for convenience we join these two pools together to obtain a global observation pool $\mathbf{\Omega}_k$, which stores all kinds of observations (absolute or relative, positive or negative, real-time or time delayed, etc.) that can be used to constrain estimates of vehicle positions.

To better describe these observations in the $\mathbf{\Omega}_k$, we define $\boldsymbol{\omega}_t^k$ to represent the collection of every observation that was generated at a historical time t ($1 \leq t \leq k$) and is available in the global observation pool. Note that $\boldsymbol{\omega}_t^k$ could be blank, since the corresponding information might be absent in the $\mathbf{\Omega}_k$. The global observation pool now can be rewritten as:

$$\mathbf{\Omega}_k = \boldsymbol{\omega}_{1:k}^k$$

From the global observation pool we also define a collection of delayed observations $\mathbf{\Omega}_{k|k-1}$ most recently received by the base station at time k. It also could be empty, as the base station might not receive any information at the time. $\mathbf{\Omega}_{k|k-1}$ can be obtained by a relative complement operation of the global observation pools at two successive time steps, which means:

$$\mathbf{\Omega}_{k|k-1} = \mathbf{\Omega}_k \setminus \mathbf{\Omega}_{k-1}$$

7.3 Bayesian Formulation on Delayed-State Cooperative Tracking

Delayed-states filtering is based on the product rule in Bayesian estimation based on Markov assumption.

$$P\left(\mathbf{x}_{0:k}|\mathbf{z}_{1:k}\right) = \left(\prod_{t=1}^{k} P\left(\mathbf{z}_t|\mathbf{x}_t\right)\right)\left(\prod_{t=1}^{k} P\left(\mathbf{x}_t|\mathbf{x}_{t-1}\right)\right) P\left(\mathbf{x}_0\right)$$

The posterior estimate is achieved with observations regardless of their arrival sequence.

Suppose a joint state representing a set of N_v vehicles moving in a field at time t.

$$\mathbf{X}_t = \left[\left(\mathbf{x}_t^{v_1}\right)^T \quad \left(\mathbf{x}_t^{v_2}\right)^T \quad \cdots \quad \left(\mathbf{x}_t^{v_N}\right)^T\right]^T$$

Suppose a sliding time window T_w, within which the states of all vehicles running in the field are tracked by a central base station. To track the joint state cooperatively, a delayed-state filter keeps present and historical states within the time window T_w, and updates estimates using real-time and delayed observations from the global observation pool.

$$P\left(\mathbf{X}_{k-T_w+1:k}|\mathbf{\Omega}_k\right) \propto P\left(\mathbf{\Omega}_{k|k-1}|\mathbf{X}_t\right) \int P\left(\mathbf{X}_k|\mathbf{X}_{k-1}\right) P\left(\mathbf{X}_{k-T_w:k-1}|\mathbf{\Omega}_{k-1}\right) d\mathbf{X}_{k-T_w} \tag{7.1}$$

where $P\left(\mathbf{X}_k|\mathbf{X}_{k-1}\right) = \prod_{p=1}^{N_v} P\left(\mathbf{x}_k^{v_p}|\mathbf{x}_{k-1}^{v_p}\right)$, based on the assumption that every node moves independently in the field.

In Equation (7.1), $\mathbf{\Omega}_{k|k-1}$ has been defined to be the collection of delayed observations received at time k. As mentioned before it could be empty.

Recall Section 3.6 for the likelihood functions given various observations. The $P\left(\mathbf{\Omega}_{k|k-1}|\mathbf{X}_t\right)$ then could be further factorised to:

$$P\left(\mathbf{\Omega}_{k|k-1}|\mathbf{X}_t\right) = \left(\prod_{\mathbf{z}_t^{v_p} \in \mathbf{\Omega}_{k|k-1}} \widetilde{\Psi}_t^{v_p}\right)\left(\prod_{\mathbf{z}_t^{v_p \to c_r} \in \mathbf{\Omega}_{k|k-1}} \Psi_t^{v_p \to c_r}\right)\left(\prod_{\mathbf{z}_t^{v_p \to v_q} \in \mathbf{\Omega}_{k|k-1}} \widetilde{\Lambda}_t^{v_p \to v_q}\right)$$

The above equation is for all p and q from 1 to N_v ($p \neq q$) and r from 1 to N_c.

A marginal $P\left(\mathbf{x}_t^{v_p}|\mathbf{\Omega}_k\right)$ for vehicle v_p at an arbitrary time step t within the time window could be obtained by integrating the joint state density function with respect to the states except for $\mathbf{x}_t^{v_p}$ in Equation (7.1).

$$P\left(\mathbf{x}_t^{v_p}|\mathbf{\Omega}_k\right) = \int P\left(\mathbf{X}_t|\mathbf{\Omega}_k\right) d\bar{\mathbf{X}}_t \tag{7.2}$$

where $P\left(\mathbf{X}_t|\mathbf{\Omega}_k\right) = \int P\left(\mathbf{X}_{k-T_w+1:k}|\mathbf{\Omega}_k\right) d\mathbf{X}_{k-T_w+1:t-1,t+1:k}$ and $\bar{\mathbf{X}}_t$ is the joint state of the vehicles except for v_p at time t.

7.4 A Delayed-State Particle Filter for Cooperative Tracking

To track the joint state of vehicles cooperatively, in total there are $(T_w + 1) \times N_v$ sets of particles to keep track of the complete T_w historical states plus one present state of the N_v tracked agents in the sliding time window up to the present time k. The filter jumps to a historical time point to perform forward filtering (followed by backward smoothing, optionally), when new time delayed observation(s) are received, as shown in Figure 7.1. The forward filtering refers to the process that the algorithm performs filtering, in chronological order, for historical states and present state using delayed observations received. The backward smoothing further improves estimates of historical states by incorporating "future" information, in reversed time sequence.

At time k, for each historical time slot $t \in [k - T_w, k]$, we define a particle base keeping N_v collections of particles which correspond to N_v vehicles.

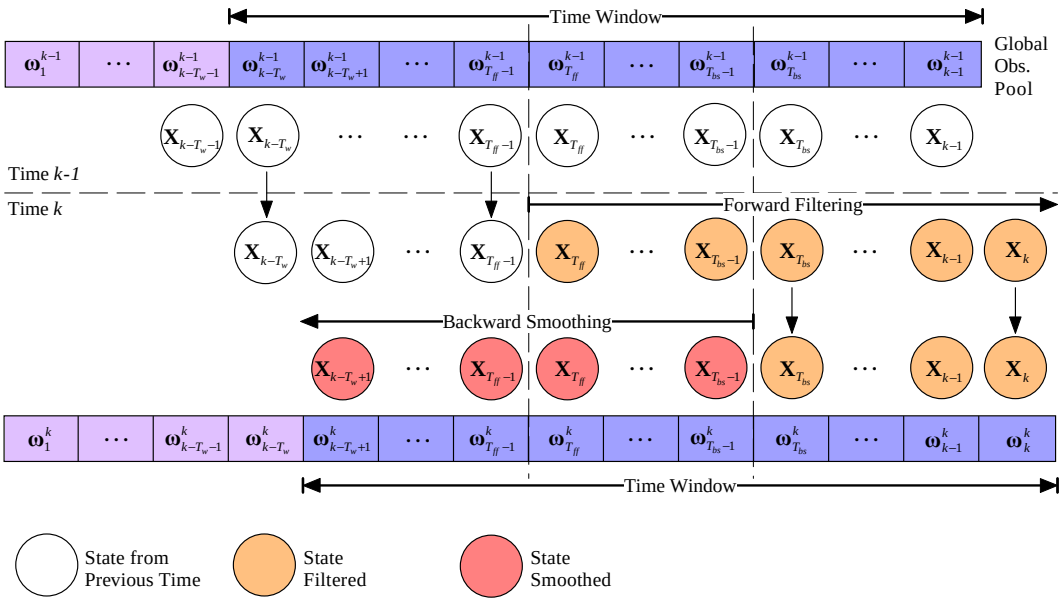


Figure 7.1 – Delayed-state cooperative particle filter.

$$\Theta_t^k = \begin{bmatrix} \{\mathbf{x}_t^{v_1,(i)}, w_t^{v_1,(i)}\}_{i=1}^{L} \sim P\left(\mathbf{x}_t^{v_1} | \boldsymbol{\omega}_{1:t}^k\right) \\ \{\mathbf{x}_t^{v_2,(i)}, w_t^{v_2,(i)}\}_{i=1}^{L} \sim P\left(\mathbf{x}_t^{v_2} | \boldsymbol{\omega}_{1:t}^k\right) \\ \vdots \\ \{\mathbf{x}_t^{v_N,(i)}, w_t^{v_N,(i)}\}_{i=1}^{L} \sim P\left(\mathbf{x}_t^{v_N} | \boldsymbol{\omega}_{1:t}^k\right) \end{bmatrix}$$

Thus to keep track of the total $T_w + 1$ states of all vehicles, we need $T_w + 1$ such particle bases.

$$\Theta_{k-T_w:k}^k = \begin{bmatrix} \Theta_{k-T_w}^k & \Theta_{k-T_w+1}^k \cdots & \Theta_{k-1}^k & \Theta_k^k \end{bmatrix} \tag{7.3}$$

7.4.1 Forward Filtering

Define a starting point of forward filtering, which determines where the particle filter starts a repropagation process from, when time is evolved from $k-1$ to k.

$$T_{ff} = f_{min}\left(\boldsymbol{\Omega}_{k|k-1}\right)$$

As the earliest time stamp of observations in $\mathbf{\Omega}_{k|k-1}$ is found to be T_{ff} by function $f_{min}(\cdot)$, the algorithm is optimised to jump back and restart its filtering process from the time point T_{ff} instead of $k - T_w$. This is apparently because for $1 \leq t \leq T_{ff}$, we have $\boldsymbol{\omega}_{1:t}^{k} = \boldsymbol{\omega}_{1:t}^{k-1}$.

For time step $t = T_{ff}$ to k and each target node v_p, a joint posterior is inferred by generalising the nonparametric cooperative tracking formulation proposed in Equation (6.1) in Section 6.3.1.

$$\begin{aligned} P_{\mathbf{X}_t}^{k} &\triangleq P\left(\bar{\mathbf{X}}_t | \mathbf{x}_t^{v_p}, \boldsymbol{\omega}_t^{k}\right) P\left(\mathbf{x}_t^{v_p} | \boldsymbol{\omega}_{1:t}^{k}\right) \\ &= \frac{P\left(\bar{\mathbf{X}}_t, \mathbf{x}_t^{v_p} | \boldsymbol{\omega}_t^{k}\right)}{P\left(\mathbf{x}_t^{v_p} | \boldsymbol{\omega}_t^{k}\right)} \times \frac{P\left(\boldsymbol{\omega}_t^{k} | \mathbf{x}_t^{v_p}\right) P\left(\mathbf{x}_t^{v_p} | \boldsymbol{\omega}_{1:t-1}^{k}\right)}{P\left(\boldsymbol{\omega}_t^{k} | \boldsymbol{\omega}_{1:t-1}^{k}\right)} \\ &\propto P\left(\boldsymbol{\omega}_t^{k} | \mathbf{X}_t\right) P\left(\mathbf{x}_t^{v_p} | \boldsymbol{\omega}_{1:t-1}^{k}\right) \end{aligned} \tag{7.4}$$

where $P\left(\mathbf{x}_t^{v_p} | \boldsymbol{\omega}_{1:t-1}^{k}\right) = \int P\left(\mathbf{x}_t^{v_p} | \mathbf{x}_{t-1}^{v_p}\right) P\left(\mathbf{x}_{t-1}^{v_p} | \boldsymbol{\omega}_{1:t-1}^{k}\right) d\mathbf{x}_{t-1}^{v_p}$.

As having been defined in Section 6.3.1, these auxiliary nodes grouped in $\bar{\mathbf{X}}_t$ are temporarily aggregated together at a historical time point t to help track the primary vehicle node v_p. The auxiliary nodes are then marginalised away to obtain the marginal probability of the primary node state $\mathbf{x}_t^{p}$.

$$P\left(\mathbf{x}_t^{v_p} | \boldsymbol{\omega}_{1:t}^{k}\right) = \int P\left(\bar{\mathbf{X}}_t | \mathbf{x}_t^{v_p}, \boldsymbol{\omega}_t^{k}\right) P\left(\mathbf{x}_t^{v_p} | \boldsymbol{\omega}_{1:t}^{k}\right) d\bar{\mathbf{X}}_t \tag{7.5}$$

By repeating the algorithm in Equations (7.4) and (7.5) for every node at each time $t \geq T_{ff}$, all nodes are updated cooperatively with both absolute and relative observations in $\boldsymbol{\omega}_{1:t}^{k}$. When the t comes to the present time k, we will have the posterior distribution inferred based on all observations available in the global observation pool, as $\mathbf{\Omega}_k = \boldsymbol{\omega}_{1:k}^{k}$.

A Gibbs sampler is used to sample from conditionals of the joint posterior distribution $P_{\mathbf{X}_t}^{k}$. The Gibbs sampler adopted is an extended version of the one described in

Section 6.3.2, which was previously used for real-time filtering. That sampler is now generalised to fit the delayed-state framework.

Define N_v variables $\theta^{v_1}, \cdots, \theta^{v_N}$ corresponding to all N_v vehicles in the joint posterior distribution. We have a replacing joint distribution written as:

$$\begin{aligned} & P\left(\theta^{v_1}, \cdots, \theta^{v_p-1}, \theta^{v_p}, \theta^{v_p+1}, \cdots, \theta^{v_N}\right) \\ & = P\left(\bar{\mathbf{X}}_t | \mathbf{x}_t^{v_p}, \boldsymbol{\omega}_t^k\right) P\left(\mathbf{x}_t^{v_p} | \boldsymbol{\omega}_{1:t}^k\right) \end{aligned}$$

We initialise these variables deterministically or randomly and we have:

$$\theta^{v_1} = \theta_0^{v_1}, \cdots, \theta^{v_N} = \theta_0^{v_N}$$

In the Gibbs sampling, the V2V relative observation could be converted to an absolute referenced observation. Given $\mathbf{z}_t^{v_p \to v_q}$ and $\mathbf{z}_t^{v_q \to v_p}$ and $\mathbf{x}_t^{v_q} = \theta_j^{v_q}$ at a scan $j \geq 0$, for convenience we joined two likelihood functions together and lastly denote the converted likelihood function as:

$$\widetilde{\Psi}_t^{v_p \leftrightarrow \theta_j^{v_q}} \triangleq P\left(\mathbf{z}_t^{v_p \to v_q} | \mathbf{x}_t^{v_p}, \mathbf{x}_t^{v_q} = \theta_j^{v_q}\right) P\left(\mathbf{z}_t^{v_q \to v_p} | \mathbf{x}_t^{v_q} = \theta_j^{v_q}, \mathbf{x}_t^{v_p}\right)$$

For each scan $j \geq 1$ and for each auxiliary node v_q, we sample:

$$\begin{aligned} \theta_j^{v_q} & \sim P\left(\theta^{v_q} | \theta_j^1, \cdots, \theta_j^{v_q-1}, \theta_{j-1}^{v_q+1}, \cdots, \theta_{j-1}^{v_N}\right) \\ & \propto \widetilde{\Psi}_t^{v_q} \left(\prod_{r=1}^{N_c} \Psi_t^{v_q \to c_r}\right) \left(\prod_{r=1}^{q-1} \widetilde{\Psi}_t^{v_q \leftrightarrow \theta_j^{v_r}}\right) \left(\prod_{r=q+1}^{N_v} \widetilde{\Psi}_t^{v_q \leftrightarrow \theta_{j-1}^{v_r}}\right) \end{aligned} \tag{7.6}$$

for all existing $\mathbf{z}_t^{v_q}, \mathbf{z}_t^{v_q \to c_r}, \mathbf{z}_t^{v_q \to v_r}, \mathbf{z}_t^{v_r \to v_q} \in \boldsymbol{\omega}_t^k$.

Then for the primary vehicle node v_p at the j^{th} scan, we sample:

$$\begin{aligned}\theta_j^p &\sim P\left(\theta^{v_p}|\theta_j^1,\cdots,\theta_j^{v_p-1},\theta_j^{v_p+1},\cdots,\theta_j^{v_N}\right)\\ &\propto \widetilde{\Psi}_t^{v_p}\left(\prod_{r=1}^{N_c}\Psi_t^{v_p\to c_r}\right)\left(\prod_{r=1,r\neq p}^{N_v}\widetilde{\Psi}_t^{v_p\leftrightarrow\theta_j^{v_r}}\right)P\left(\mathbf{x}_t^{v_p}|\boldsymbol{\omega}_{1:t-1}^k\right)\end{aligned} \tag{7.7}$$

for all existing $\mathbf{z}_t^{v_p}, \mathbf{z}_t^{v_p\to c_r}, \mathbf{z}_t^{v_p\to v_r}, \mathbf{z}_t^{v_r\to v_p} \in \boldsymbol{\omega}_t^k$. And in the equation we also have $P\left(\mathbf{x}_t^{v_p}|\boldsymbol{\omega}_{1:t-1}^k\right) = \int P\left(\mathbf{x}_t^{v_p}|\mathbf{x}_{t-1}^{v_p}\right) P\left(\mathbf{x}_{t-1}^{v_p}|\boldsymbol{\omega}_{1:t-1}^k\right) d\mathbf{x}_{t-1}^{v_p}$.

Iterative runs of the two equations generate a Gibbs chain. A set of posterior particles for the primary vehicle node v_p is then extracted from the chain, as described in Section 6.3.2.

By repeating the above algorithm for every node at time t, all nodes are updated cooperatively with both egocentric and relative observations in $\boldsymbol{\omega}_t^k$. In the form of a pseudocode, the algorithm is also summarised in Table 7.1.

7.4.2 Backward Smoothing

After the forward filtering process, a particle set kept in the Θ_t^k approximates the distribution of a vehicle at time t by fusing a subset of observations, denoted by $\boldsymbol{\omega}_{1:t}^k$, in the global observation pool $\boldsymbol{\Omega}_k$.

Equation 7.2 for delayed-state filtering in Section 7.3 computes $P\left(\mathbf{x}_t^{v_p}|\boldsymbol{\Omega}_k\right)$ (i.e. $P\left(\mathbf{x}_t^{v_p}|\boldsymbol{\omega}_{1:k}^k\right)$), which means $\mathbf{x}_t^{v_p}$ is not only filtered by observations up to time t, but also smoothed by observations up to a later time k. Obviously estimates based on $P\left(\mathbf{x}_t^{v_p}|\boldsymbol{\Omega}_k\right)$ tend to be more certain than that on $P\left(\mathbf{x}_t^{v_p}|\boldsymbol{\omega}_{1:t}^k\right)$ unless $t = k$.

The smoothed distribution $P\left(\mathbf{x}_t^p|\boldsymbol{\Omega}_k\right)$ could be elaborated to:

Table 7.1 – Algorithm: Forward filtering

$\Theta_t^k \leftarrow$ *Forward_Filtering* $\left(\Theta_{t-1}^k, \boldsymbol{\omega}_t^k\right)$

par for $p = 1$ to N_v **do**
  load particle set $\{\mathbf{x}_{t-1}^{v_p,(i)}, w_{t-1}^{v_p,(i)}\}_{i=1}^{L}$ from Θ_{t-1}^k
  initialise Gibbs variables $\theta^{v_1} = \theta_0^{v_1}, \cdots, \theta^{v_N} = \theta_0^{v_N}$
  for $j = 1$ to $(N_{bp} + N_{si} \times L)$ **do**
    Sampling for each auxiliary node:
    for $q = 1$ to $p-1$, $p+1$ to N_v **do**
      draw $\theta_j^{v_q} \sim P\left(\theta^{v_q} | \theta_j^{v_1}, \cdots, \theta_j^{v_{q-1}}, \theta_{j-1}^{v_{q+1}}, \cdots, \theta_{j-1}^{v_N}\right)$
      with delayed observations from $\boldsymbol{\omega}_t^k$ fused in
    end for
    Sampling for the primary node:
    draw $\theta_j^{v_p} \sim P\left(\theta^{v_p} | \theta_j^{v_1}, \cdots, \theta_j^{v_{p-1}}, \theta_j^{v_{p+1}}, \cdots, \theta_j^{v_N}\right)$
    with delayed observations from $\boldsymbol{\omega}_t^k$ fused in
    if $(j - N_{bp})$ is an integer times of N_{si} **do**
      Extraction:
      particle index $i = (j - N_{bp}) / N_{si}$
      save sample $\mathbf{x}_t^{v_p,(i)} = \theta_j^{v_p}$
      set weight $w_t^{v_p,(i)} = 1/L$
    end if
  end for
  save posterior particles $\{\mathbf{x}_t^{v_p,(i)}, w_t^{v_p,(i)}\}_{i=1}^{L}$ to Θ_t
end for

$$
\begin{aligned}
P\left(\mathbf{x}_t^{v_p} | \boldsymbol{\omega}_{1:k}^k\right) &= \int P\left(\mathbf{x}_t^{v_p} | \mathbf{x}_{t+1}^{v_p}, \boldsymbol{\omega}_{1:t}^k\right) P\left(\mathbf{x}_{t+1}^{v_p} | \boldsymbol{\omega}_{1:k}^k\right) d\mathbf{x}_{t+1}^{v_p} \\
&= \int \frac{P\left(\mathbf{x}_t^{v_p} | \boldsymbol{\omega}_{1:t}^k\right) P\left(\mathbf{x}_{t+1}^{v_p} | \mathbf{x}_t^{v_p}\right)}{P\left(\mathbf{x}_{t+1}^{v_p} | \boldsymbol{\omega}_{1:t}^k\right)} P\left(\mathbf{x}_{t+1}^{v_p} | \boldsymbol{\omega}_{1:k}^k\right) d\mathbf{x}_{t+1}^{v_p} \\
&= P\left(\mathbf{x}_t^{v_p} | \boldsymbol{\omega}_{1:t}^k\right) \int \frac{P\left(\mathbf{x}_{t+1}^{v_p} | \mathbf{x}_t^{v_p}\right) P\left(\mathbf{x}_{t+1}^{v_p} | \boldsymbol{\omega}_{1:k}^k\right)}{\int P\left(\mathbf{x}_{t+1}^{v_p} | \mathbf{x}_t^{v_p}\right) P\left(\mathbf{x}_t^{v_p} | \boldsymbol{\omega}_{1:t}^k\right) d\mathbf{x}_t^{v_p}} d\mathbf{x}_{t+1}^{v_p}
\end{aligned}
\tag{7.8}
$$

Equation (7.8) enables the obtaining of the smoothed $P\left(\mathbf{x}_t^{v_p} | \boldsymbol{\omega}_{1:k}^k\right)$ with the original $P\left(\mathbf{x}_t^{v_p} | \boldsymbol{\omega}_{1:t}^k\right)$ at time t and $P\left(\mathbf{x}_{t+1}^{v_p} | \boldsymbol{\omega}_{1:k}^k\right)$, which is the smoothed density function at the next time step.

To represent the smoothed distributions, we introduce additional particle bases similar

to those defined in Equation (7.3).

$$\Theta^{*k}_{k-t_w+1:k} = \left[\Theta^{*k}_{k-t_w+1} \quad \Theta^{*k}_{k-t_w+2} \cdots \quad \Theta^{*k}_{k-1} \quad \Theta^{*k}_{k}\right]$$

where

$$\Theta^{*k}_{t} = \begin{bmatrix} \{\mathbf{x}_t^{v_1,(i)}, w_t^{*v_1,(i)}\}_{i=1}^{L} \sim P\left(\mathbf{x}_t^{v_1}|\mathbf{\Omega}_k\right) \\ \{\mathbf{x}_t^{v_2,(i)}, w_t^{*v_2,(i)}\}_{i=1}^{L} \sim P\left(\mathbf{x}_t^{v_2}|\mathbf{\Omega}_k\right) \\ \vdots \\ \{\mathbf{x}_t^{v_N,(i)}, w_t^{*v_N,(i)}\}_{i=1}^{L} \sim P\left(\mathbf{x}_t^{v_N}|\mathbf{\Omega}_k\right) \end{bmatrix}$$

Define a starting point of backward smoothing.

$$T_{bs} = f_{max}\left(\mathbf{\Omega}_{k|k-1}\right)$$

Since the largest time stamp of observations in $\mathbf{\Omega}_{k|k-1}$ is found to be T_{bs} by function $f_{max}(\cdot)$, the algorithm is optimised to start its smoothing process from the time point $T_{bs}-1$ back to $k-T_w$, instead of from time $k-1$. This is evidently because $\boldsymbol{\omega}^k_{1:t} = \boldsymbol{\omega}^k_{1:k}$ when $T_{bs} \leq t \leq k$.

Hence each Θ^k_t is copied to Θ^{*k}_t in particles bases when $t_{bs} \leq t \leq k$.

Then for each vehicle v_p, for $t = T_{bs}-1, T_{bs}-2, \cdots, k-t_w+1$ and $i = 1:L$, recursively we have the smoothed particle weights calculated as:

$$w_t^{*v_p,(i)} = w_t^{v_p,(i)} \sum_{j=1}^{L} \frac{P\left(\mathbf{x}_{t+1}^{v_p,(j)}|\mathbf{x}_t^{v_p,(i)}\right) w_{t+1}^{*v_p,(j)}}{\sum_{m=1}^{L} P\left(\mathbf{x}_{t+1}^{v_p,(j)}|\mathbf{x}_t^{v_p,(m)}\right) w_t^{v_p,(m)}}$$

To sum up, the smoothing is processed with particles re-weighted to obtain an approximation to the smoothed density, but the original particle locations are maintained. The resultant particle set $\{\mathbf{x}_t^{v_p,(i)}, w_t^{*v_p,(i)}\}_{i=1}^{L}$ approximates the marginal distribution $P\left(\mathbf{x}_t^{v_p}|\mathbf{\Omega}_k\right)$ of vehicle v_p, which is identical to that yielded in Equation (7.2). The backward smoothing algorithm is summarised in Table 7.2 in the form of a pseudocode.

Table 7.2 – Algorithm: Backward smoothing

$\Theta_t^{*k} \leftarrow$ *Backward_Smoothing* $\left(\Theta_t^k, \Theta_{t+1}^{*k}\right)$

for $p = 1$ to N_v **do**
  load particle set $\{\mathbf{x}_t^{v_p,(i)}, w_t^{v_p,(i)}\}_{i=1}^L$ from Θ_t^k
  load particle set $\{\mathbf{x}_{t+1}^{v_p,(i)}, w_{t+1}^{*v_p,(i)}\}_{i=1}^L$ from Θ_{t+1}^{*k}
  for $i = 1$ to L **do**
    for $j = 1$ to L **do**
      $\alpha^{(j)} = \sum_{m=1}^L P\left(\mathbf{x}_{t+1}^{v_p,(j)} | \mathbf{x}_t^{v_p,(m)}\right) w_t^{v_p,(m)}$
    end for
    $w_t^{*v_p,(i)} = w_t^{v_p,(i)} \sum_{j=1}^L \frac{P\left(\mathbf{x}_{t+1}^{v_p,(j)} | \mathbf{x}_t^{v_p,(i)}\right) w_{t+1}^{*v_p,(j)}}{\alpha^{(j)}}$
  end for
  save smoothed particles $\{\mathbf{x}_t^{v_p,(i)}, w_t^{*v_p,(i)}\}_{i=1}^L$ to Θ_t^{*k}
end for

The smoothed estimates are extracted from the additional particles bases $\Theta_{k-t_w+1:k}^{*k}$. However it is important to note that, only filtered particle bases $\Theta_{k-t_w:k}^k$ are kept and evolved to the next time step, while the $\Theta_{k-t_w+1:k}^{*k}$ are calculated at every new step. More information about backward smoothing is available in [97], [98] and [49].

The particles re-weighted by the smoothing process are only for historical states. The last/present state estimates would be always identical before and after smoothing. The pseudocode of the whole DSCPF algorithm is also presented in Table 7.3. Nevertheless, the backward smoothing process in the DSCPF is optional. It could be skipped to conserve the computational cost in applications where only the estimation of present states is cared about.

7.5 Simulation Results

7.5.1 Setup

The proposed DSCPF algorithm has been evaluated in two types of simulation cases, one linear and Gaussian, and the other linear and non-Gaussian. In both types,

Table 7.3 – Algorithm: DSCPF

$\Theta^{*k}_{k-t_w+1:k}, \Theta^{k}_{k-t_w:k} \leftarrow DSCPF\left(\Theta^{k-1}_{k-t_w-1:k-1}, \mathbf{\Omega}_k, \mathbf{\Omega}_{k-1}\right)$

$\mathbf{\Omega}_{k|k-1} = \mathbf{\Omega}_k \setminus \mathbf{\Omega}_{k-1}$
Forward filtering:
$T_{ff} = f_{min}\left(\mathbf{\Omega}_{k|k-1}\right)$
for $t = k - T_w$ to $T_{ff} - 1$ **do**
  for $p = 1$ to N_v **do**
    copy particles $\{\mathbf{x}_t^{v_p,(i)}, w_t^{v_p,(i)}\}_{i=1}^{L}$ from Θ_t^{k-1} to Θ_t^k
  end for
end for
for $t = T_{ff}$ to k **do**
  extract $\boldsymbol{\omega}_t^k$ from $\mathbf{\Omega}_k$
  $\Theta_t^k \leftarrow Forward\_Filtering\left(\Theta_{t-1}^k, \boldsymbol{\omega}_t^k\right)$
end for
Backward smoothing:
$T_{bs} = f_{max}\left(\mathbf{\Omega}_{k|k-1}\right)$
for $t = T_{bs}$ to k **do**
  for $p = 1$ to N_v **do**
    copy particles $\{\mathbf{x}_t^{v_p,(i)}, w_t^{v_p,(i)}\}_{i=1}^{L}$ from Θ_t^k to Θ_t^{*k}
  end for
end for
for $t = T_{bs} - 1$ to $k - T_w + 1$ **do**
  $\Theta_t^{*k} \leftarrow Backward\_Smoothing\left(\Theta_t^k, \Theta_{t+1}^{*k}\right)$
end for

the simulation of the DSCPF algorithm is setup with 7 mobile nodes moving within one-dimensional space. Four of the mobile nodes have self-localisation capability (providing absolute global position measurements), and the rest do not. Each node is equipped with a radio device to communicate with others, and to measure relative distance to them in the meanwhile. All observations about these nodes, including absolute and relative ones, are forwarded to a central infrastructure set somewhere to achieve the global tracking of each mobile node. Please note that the infrastructure is only for data gathering and fusion purpose in the simulation, but not involved in assisting localisation or tracking. Different to the assumption of real-time observations made in the conventional scenarios (as in **Chapter 6**), we assume that every observation will arrive at the fusion centre with some random time delay. The fu-

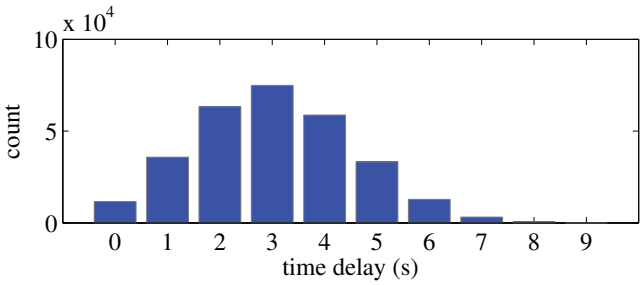


Figure 7.2 – Random time delays of observations. The pattern of the histogram generally follows a binomial distribution. The count values of the last three bins (7, 8 and 9) are hard to determine from the figure. Their values are 3199, 504, and 14, respectively.

sion centre is using the delayed observations collected to track every node in the field. With cooperative tracking introduced, nodes help each other improve tracking performance in the use of relative observations between them. Especially those without the self-localisation means have to be localised and tracked by the distance measurements from and to their neighbours.

7.5.2 Linear and Gaussian Cases

In the first type, the simulation uses a linear motion dynamics model, and noises in observations are set to be Gaussian. Certainly DSCPF could also be applied in systems presenting non-linear and/or non-Gaussian properties, such as the experiments in Section 7.6. The purpose of this simulation is to evaluate the performance of the proposed DSCPF in comparison with the optimal tracking approach, which is the delayed-state Kalman filter (DSKF) in the linear and Gaussian settings.

Initially, $N_n = 7$ mobile nodes are positioned with even intervals. The communication device mounted on each node allows a connection to other nodes and relative distances to be measured. During the simulation, every mobile node was moving with random velocities. Both absolute and relative observations arrive with random time delays, which are set to present a binomial probability distribution within a time window, see Figure 7.2. In the simulation, both filters are tracking the state of each node, which is composed by position and velocity in the one-dimensional space.

Table 7.4 – Simulation parameters

Parameter	Value
total Monte Carlo simulation cases	30
total iterations per case	200
total mobile nodes N_n	7
time window size T_w	10 s
total mobile nodes with egocentric observation enabled	4
speed noise	$\mathcal{N}(0, 0.2^2)$
position observation noise $\mathbf{v}_{abs}$	$\mathcal{N}(0, 5^2)$
distance observation noise $\mathbf{v}_{rel}$	$\mathcal{N}(0, 15^2)$
particle quantity for primary node N_p	2000
particle quantity for auxiliary node N_a	500
burn-in length of Gibbs sampler N_{bp}	200
sampling interval of Gibbs sampler N_{si}	5
length of Gibbs chain N_{cl}	2000

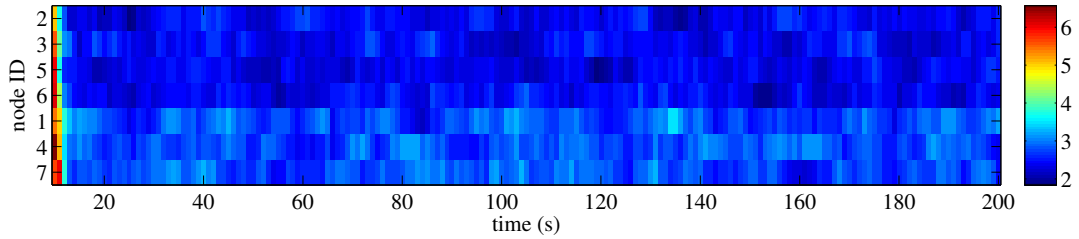
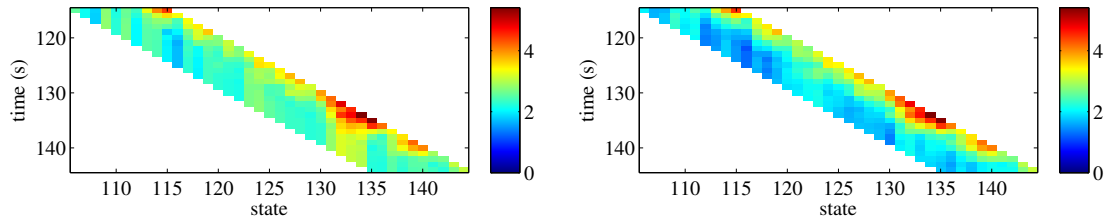


Figure 7.3 – Mean RMSEs of position estimates of nodes (in metres). The bottom three are those without GPS, and they have to be localised by relative distances to neighbours.



(a) Without smoothing (b) With smoothing

Figure 7.4 – DSCPF without vs. with smoothing. (a) shows the position estimates of node #1 (without GPS) from DSCPF without smoothing. Generally, the DSCPF with smoothing gives a better tracking result with smoother variation of estimates, as shown in (b). The results are in metres.

$$\mathbf{x} = \begin{bmatrix} r_x & v_x \end{bmatrix}^T$$

The simulation is initialised with parameters given in Table 7.4. These parameters were determined in order to produce explicit results. The time window is chosen to be as narrow as 10 seconds. This is because the motion model used in the simulation is not designed for long-term prediction. In addition, the particle filter is initialised with 2000 particles for each primary node, and 500 only for every auxiliary node. Some location errors were intentionally added to every node in the initial state.

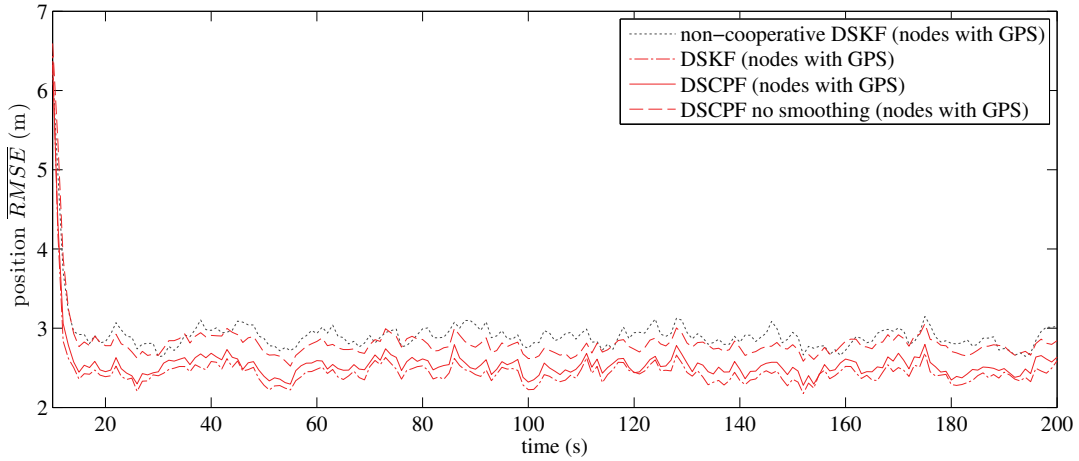
In order to quantify the performance, the RMSE is again used as a measure of performance. The RMSE of position for overall N_n mobile nodes could be computed by:

$$RMSE = \sqrt{\frac{\sum_{p=1}^{N_n} \sum_{t=1}^{T_w} \left(\hat{r}_x^{p,t} - r_x^{p,t}\right)^2}{N_n \times T_w}}$$

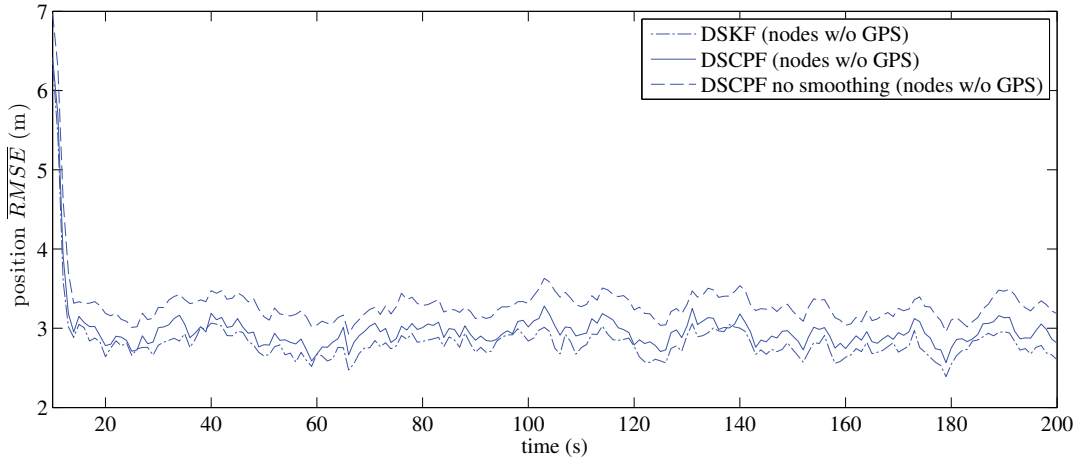
Overall mean RMSE is then calculated by averaging over $N = 30$ independent Monte Carlo cases. Mean RMSE for an individual node could also be obtained by averaging RMSEs of the particular node in these cases.

The mean RMSEs of positions for all 7 individual mobile nodes in the DSCPF tracking result are demonstrated in Figure 7.3. As observed, the algorithm tracks GPS enabled nodes at a higher accuracy level, with both absolute and relative measurements provided, than the nodes without GPS, which have to be localised by distance measurements from and to their neighbours. Figure 7.4 gives details and makes a comparison between the tracking results with and without smoothing enabled in DSCPF. With the backward smoothing enabled, estimates of past states could be improved by observations up to a later time. Also noteworthy is the identical last state estimates in both DSCPFs.

To compare DSCPF and DSKF, Figure 7.5 summarises the tracking results in a straightforward manner. The tracking result of the DSCPF without backward smooth-



(a) Mean position RMSE of nodes with GPS



(b) Mean position RMSE of nodes without GPS

Figure 7.5 – Mean RMSEs of positions. The tracking results of the non-cooperative DSKF for nodes without GPS are not shown in (b) and (d) as the independent tracking algorithm fails in such scenarios.

ing is also provided. It is seen that, the DSCPF with smoothing gives close performance to the optimal most of the time, for both types of nodes (with and without GPS). It is believed that better results are achievable with higher quantities of particles in DSCPF. The tracking result from non-cooperative version of DSKF is also illustrated, where nodes are tracked independently. The advantage of cooperative tracking is clearly demonstrated in the figure. The independent tracking algorithm is unable to track positions of nodes without GPS, therefore its result is not shown in

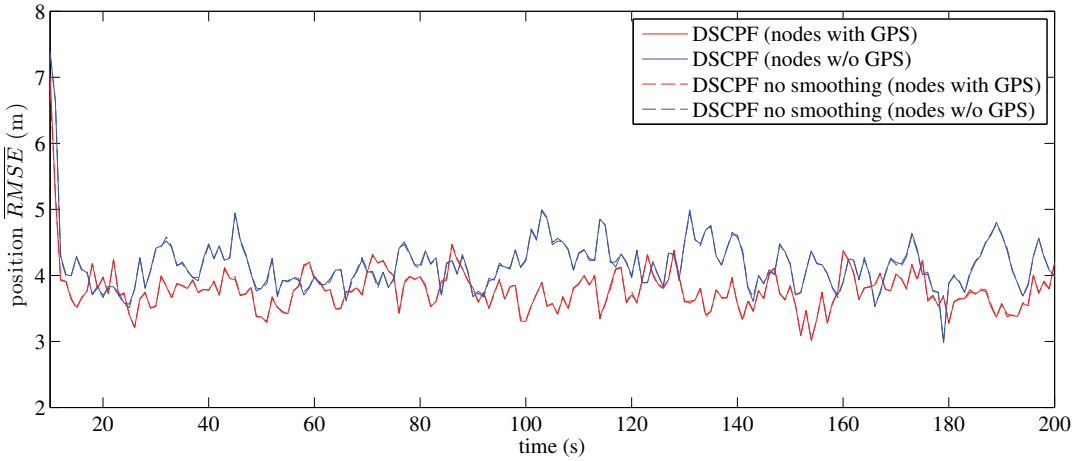


Figure 7.6 – Mean RMSEs of the last state positions. The tracking results of the last state positions for DSCPF with and without smoothing are almost the same. Specifically, the line representing the result of the nodes with GPS by DSCPF with smoothing almost overlaps with that by DSCPF without smoothing. Similarly, the results of the nodes without GPS by DSCPFs with and without smoothing almost overlap.

Figure 7.5(b). In addition, estimates of nodes with GPS are also improved when the cooperative tracking is enabled.

Though the DSCPF with smoothing outperforms that without smoothing, which has been clearly shown in Figure 7.5, their average RMSEs of the last state estimates are highly close, see Figure 7.6. It can be concluded that in implementation where only the present/last state estimates are considered, the DSCPF without smoothing is preferred due to a comparatively lower computational expense.

7.5.3 Linear and Non-Gaussian Cases

The proposed DSCPF has also been tested in $N = 30$ Monte Carlo linear and non-Gaussian cases. A Student's t-distribution illustrated in Figure 7.7 is chosen as the non-Gaussian distribution used in the simulation, as it is similar to Gaussian but heavier-tailed. In these simulation cases, the non-Gaussian noises exist in the relative range measurements between nodes. As a nonparametric approach, the DSCPF is able to accurately deal with the non-Gaussian noises, while the DSKF has to approx-

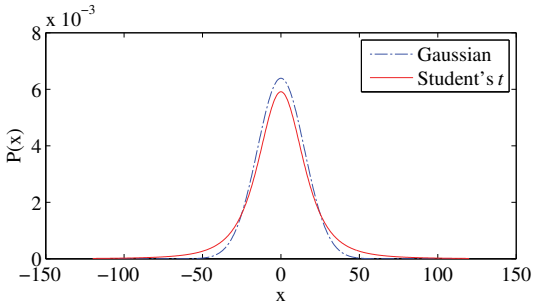


Figure 7.7 – A non-Gaussian distribution and its Gaussian approximation. The red curve shows a Student's t-distribution as an example of the non-Gaussian distribution, and it could be formulated by $St(0, \lambda, \nu)$, where $\nu = 3$ is degrees of freedom, and the precision $\lambda = 0.0133$. Its Gaussian approximation is shown in blue, which is represented by $\mathcal{N}(0, \sigma^2)$, where $\sigma = \sqrt{\frac{1}{\lambda}\frac{\nu}{\nu-2}} = 15$.

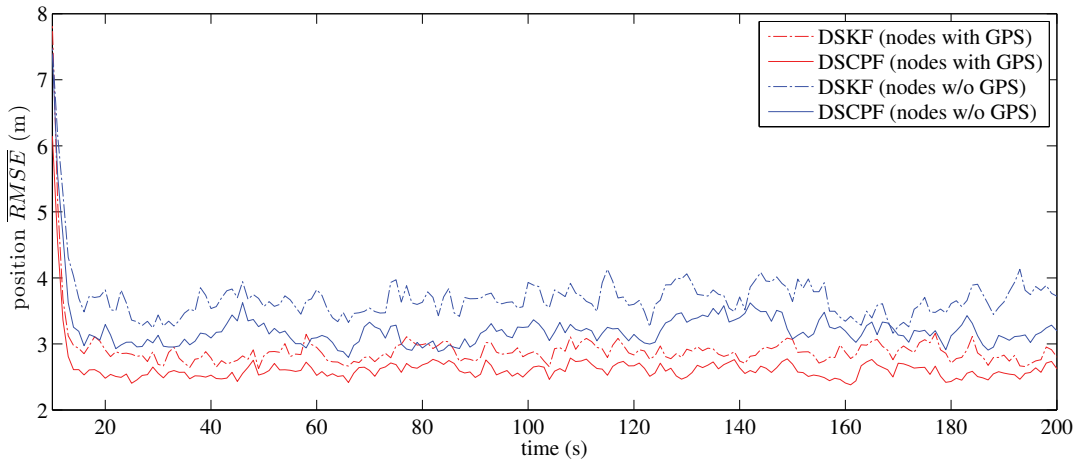


Figure 7.8 – Mean RMSEs of positions in the non-Gaussian cases.

imate the non-Gaussian distribution in the use of a Gaussian illustrated in Figure 7.7. In terms of the overall mean RMSE, Figure 7.8 compares the tracking results of the DSCPF algorithm and DSKF in linear and non-Gaussian cases. It could be easily told from the figure that the DSCPF algorithm outperforms DSKF in terms of tracking accuracy for both node types (with and without GPS).

The DSCPF also tracks nodes with better estimation consistency than DSKF. The mean NEES curve of the DSCPF about node #1 stays inside the 95% probability concentration region for most of the simulation iterations, as illustrated in Figure 7.9. However, the result of DSKF has a much higher proportion of the iterations above

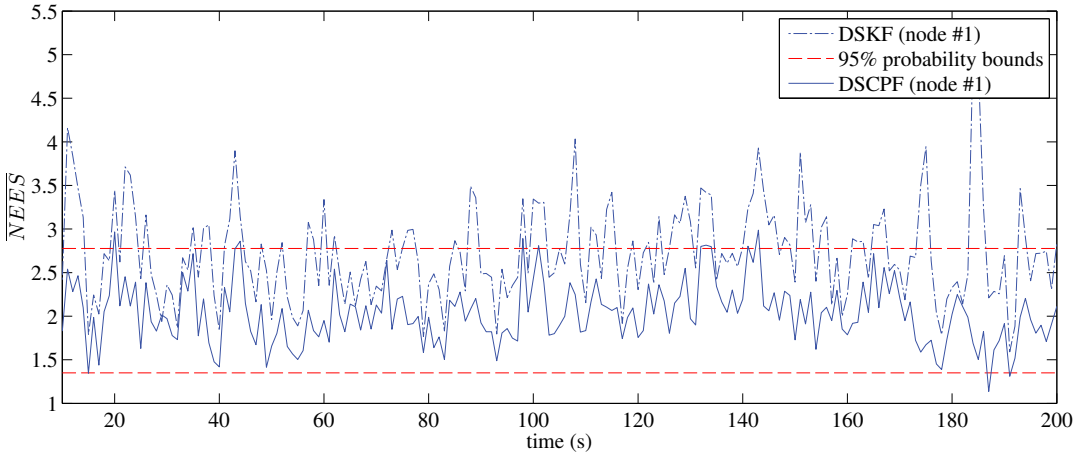


Figure 7.9 – NEES consistency test for the tracking of node #1. The in-bound rate of the DSCPF tracking result is about 93.72%, close to the optimal 95%, while the value for DSKF is only 59.16%.

the upper bound than expected. A conclusion can be drawn that DSKF tends to be over-optimistic in estimation in the simulation cases. The NEES results for other nodes are similar to that of the node presented.

7.6 Experiment Validation

Two experiments were conducted to validate the delayed-state cooperative tracking approach proposed. The backward smoothing process was not enabled in the DSCPF algorithm when implemented in the experiments, in which the last state estimates were considered important. In each experiment, the DSCPF algorithm was run with 5000 primary particles, and the effective length of Gibbs chain was set to be 5000.

7.6.1 Experiment I

Setup

The first experiment was to follow up the previous experiment in Section 5.4, where three vehicles (V1, V2 and V3) were tracked in a mining operation using delayed ob-

Table 7.5 – GPS status of vehicles in Experiment I

	Vehicles		
	V1	V2	V3
GPS Enabled	✗	✓	✓

servations received through observation harvesting. Details, such as vehicles routes, field map, communication activities and so on, could be found in that section. This time, the vehicles were tracked in the use of the new proposed approach. To demonstrate the advantage of cooperative tracking, the absolute egocentric observations (i.e. GPS data) of V1 were intentionally disabled in the experiment, see Table 7.5. This is equivalent to the case that V1 was not equipped with GPS device.

Another difference to the previous experiment is that, each vehicle kept recording the wireless connectivity and RSSI measurements when communicating with the other two along with its movement. The V2V relative observations, as a part of delayed observations, were brought to the base station when any vehicles got connected to either data collection points (C1 or C2), which has been explained in the observation harvesting. The positions of those vehicles out of contact with the infrastructure were predicted with the long-term motion prediction approach. The performance of the tracking is again evaluated by the estimation errors and entropies, which were used in the previous experiment.

The cooperative tracking results were compared to the tracking results with V2V communication enabled, i.e. the observation harvesting enabled. In a non-cooperative tracking case, vehicles do not communicate with each other. The tracking with V2V enabled is the one between the non-cooperative and cooperative tracking, as it enables inter-vehicle communication so that vehicles help each other bring GPS updates to the base station, but the fusion stage does not consider V2V relative information. This is the scenario studied in the previous experiment. This experiment was to demonstrate how tracking performance would be further improved by enabling cooperative tracking in the same experiment conducted in Section 5.4.

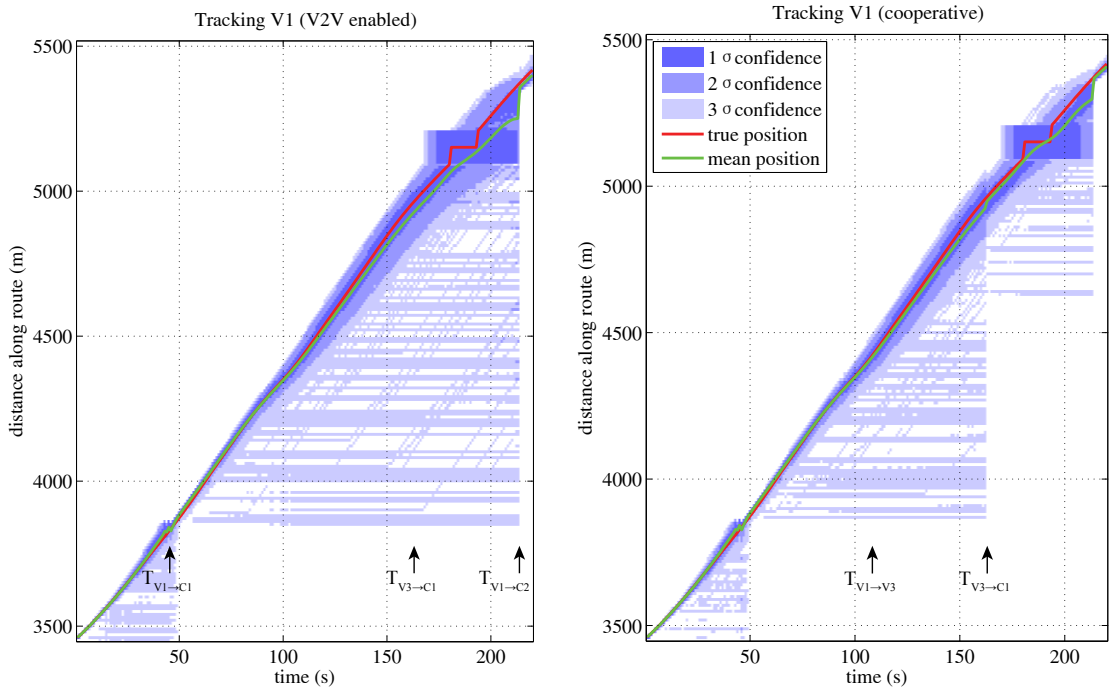


Figure 7.10 – Cooperative tracking results of Experiment I in a comparison to the case without cooperative tracking. This figure is continued by Figure 7.11. Recall the vehicles routes presented in Figure 5.2, V1 moved along a road near area #208 and met V3 followed by V2. After V1 drove out of the communication range of the data collection point C1 at $T_{V1\to C1}$ (45 s), the base station lost connection with it and had to estimate its position by motion prediction until the connection recovered at $T_{V1\to C2}$ (213 s). Through V2V communication between V1 and V3, the V2V relative range observations between them up to time $T_{V1\to V3}$ (108 s) was taken to the base station when V3 established a connection with C1 at $T_{V3\to C1}$ (162 s). The interactions between V1 and V2 also allow the egocentric information and the V2V relative measurements about V2 up to $T_{V2\to V1}$ (189 s) to be carried to C2 at $T_{V1\to C2}$ (213 s) via V1. Though V2 never communicated with V3, it uploaded negative V2V relative connectivity information about V3 before it lost connection with C2 at $T_{V2\to C2}$ (107 s). Compared to the tracking result of V1 with V2V communication enabled in (a), the cooperative tracking result in (b) shows a more accurate estimation of V1 from the time $T_{V3\to C1}$ (162 s) on. The result of V3 with cooperative tracking in Figure 7.11(b) also outperforms the case with only V2V communication enabled in Figure 7.11(a) roughly between time 80 s and $T_{V3\to C1}$ (162 s). The tracking results about V2 with and without cooperative tracking are found to be the same in Figures 7.11(d) and 7.11(c) respectively.

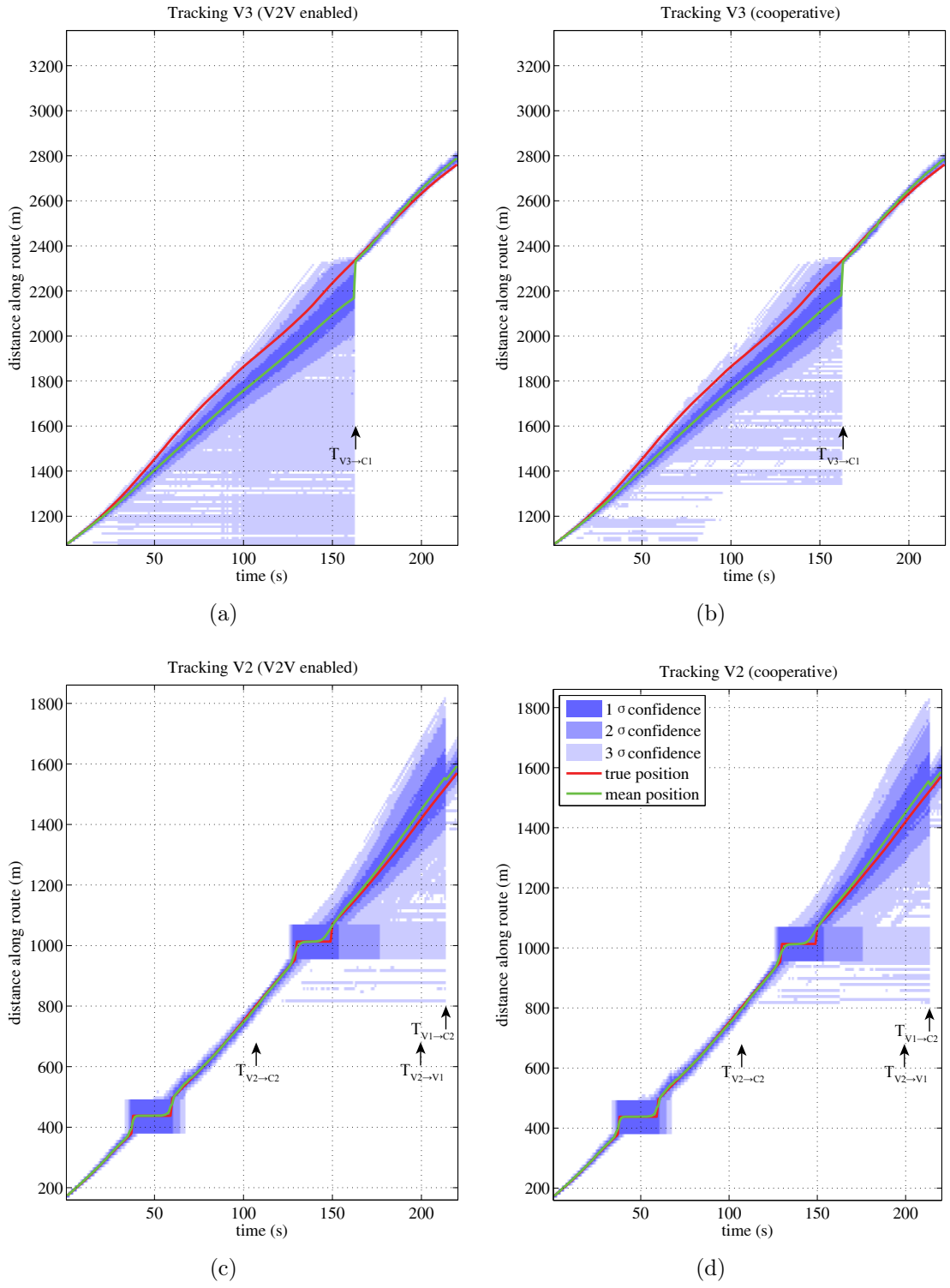
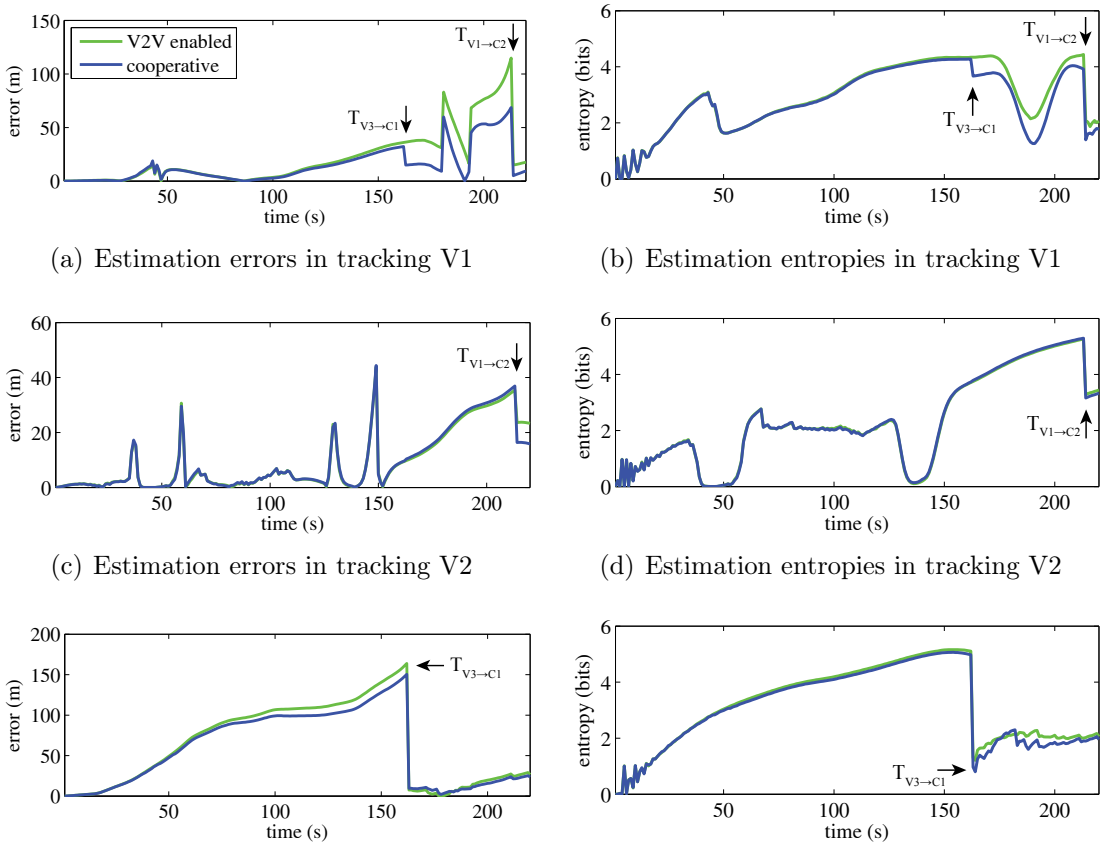


Figure 7.11 – Cooperative tracking results of Experiment I (cont.).



(a) Estimation errors in tracking V1

(b) Estimation entropies in tracking V1

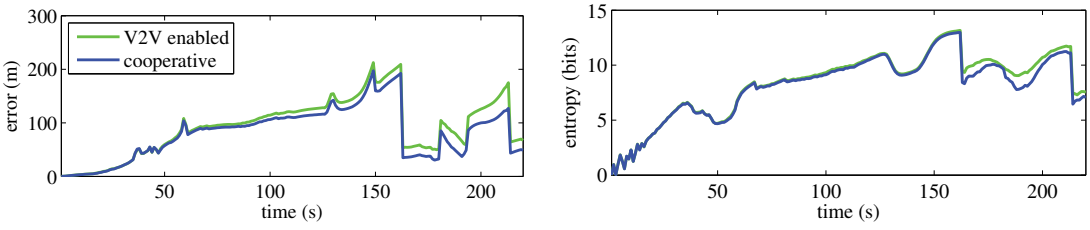
(c) Estimation errors in tracking V2

(d) Estimation entropies in tracking V2

(e) Estimation errors in tracking V3

(f) Estimation entropies in tracking V3

Figure 7.12 – Comparisons of estimation errors and entropies. Drops in errors and entropies of V1 are respectively found in (a) and (b) from time $T_{V3\to C1}$ (162 s) on. Though hard to tell by comparing Figures 7.11(c) and 7.11(d), improvement in estimation errors is achieved for V2 right after time $T_{V1\to C2}$ (213 s) according to (c). Lastly for V3, (e) and (f) also suggest more accurate results between time 80 s and $T_{V3\to C1}$ (162 s) when it was tracked in a cooperative way.



(a) Total estimation errors

(b) Fleet entropies

Figure 7.13 – Comparisons of fleet estimation errors and entropies.

Results

Compared to the tracking case with V2V communication enabled, the results with cooperative tracking enabled show more accurate position estimates of V1 and V3. Without cooperative tracking, the filter at the base station could not make use of the V2V relative information received, resulting in unchanged position estimation of V1 at $T_{V3\rightarrow C1}$, see Figure 7.10(a). When cooperative tracking was enabled, the delayed V2V relative observations received helped refine the position estimate of V1, as shown in Figure 7.10(b), even though V1 had neither egocentric observations nor a direct connection with any data collection points at that time. V1 brought the egocentric position data of V2 to C2 at $T_{V1\rightarrow C2}$, therefore the tracking results for V2 with and without cooperative tracking were almost the same, as receptively presented in Figures 7.11(d) and 7.11(c). In Figure 7.11(b), V3 is also observed to be better estimated with cooperative tracking than it is in Figure 7.11(a), as a result of V2 bringing the negative V2V relative information to the base station since it did not observe V3 along its way. This V2V relative range information about V3 cannot be considered in the fusion stage without cooperative tracking.

In terms of estimation errors and entropies, Figure 7.12 quantitatively compares the performance of two tracking cases and clearly illustrates improvements in the cooperative one. Represented by blue lines in the figure, the tracking results of every agent with cooperative tracking enabled after some time points outperform the other one denoted by green lines. It is also found to be true in the comparisons of the fleet estimation errors and entropies presented in Figure 7.13.

The comparisons of these figures demonstrate benefits brought by cooperative tracking. With the delayed-state cooperative tracking approach introduced, improvements in tracking performance are achieved by taking advantage of inter-vehicle relative range measurements, with which the estimation of agents both with and without egocentric observations could be refined.

7.6.2 Experiment II

Setup

The second experiment was to further demonstrate the advantage of cooperative tracking in a more complicated case: three vehicles (V1, V2 and V3) were tracked with only one of them equipped with GPS and one data collection point (C1) installed in the field, see Figure 7.14. More specifically, two out of these three vehicles, V1 and V3 were not provided with GPS information (see Table 7.6), and they ran together with an almost constant distance between them. This means that the tracking algorithm in this experiment has to rely more on V2V relative observations.

As the same to the setup in the first experiment, each vehicle had the V2V relative observations together with other delayed observations brought to the base station through the observation harvesting. To make a comparison, the experiment was also run with a non-cooperative tracking configuration, where inter-vehicle communication was disabled. The long-term vehicle motion prediction approach was again adopted to predict positions of vehicles when they were not in the communication range of the infrastructure.

Table 7.6 – GPS status of vehicles in Experiment II

	Vehicles		
	V1	V2	V3
GPS Enabled	✗	✓	✗

Results

The relative distance and communication activities of V2V and V2I are presented in Figure 7.15. Some important time points in the communication activities are summarised out as below.

- $T_{V3 \rightarrow V2}$: The first time V2 observed V3.

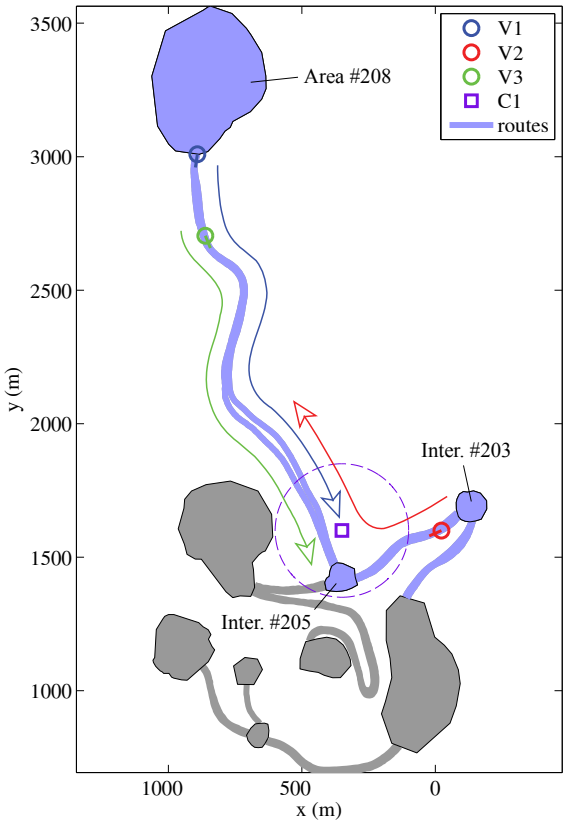
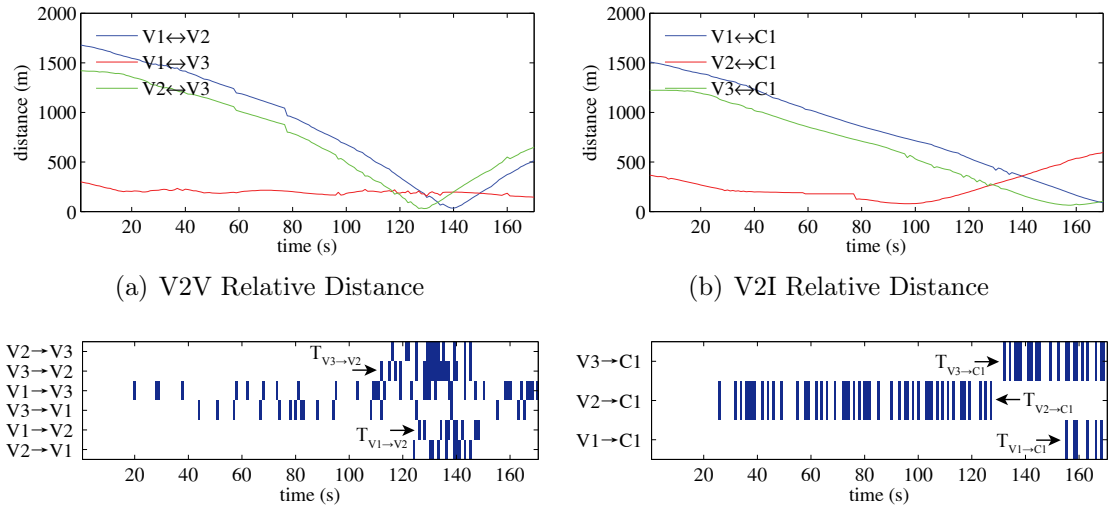


Figure 7.14 – Vehicles routes on field map. Three mobile agents were running in a field with only one data collection point installed near intersection #205. The data collection point established a signal circle with a radius of 250 meters. The figure also shows vehicles' routes on the field map. V1 and V3 initialised a trip on the road next to area #208, whereas V2 was coming from a road near intersection #203.

- $T_{V1 \to V2}$: The first time V2 observed V1.
- $T_{V2 \to C1}$: The last communication between V2 and C1.
- $T_{V3 \to C1}$: The first communication between V3 and C1.
- $T_{V1 \to C1}$: The first communication between V1 and C1.

V2 firstly arrived at intersection #205 and drove north. During its trip, it met V3 at time $T_{V3 \to V2}$ and then V1 at $T_{V1 \to V2}$, which were moving together to the south. Lastly V3 followed by V1 drove to C1 at $T_{V3 \to C1}$ and $T_{V1 \to C1}$ respectively.



(a) V2V Relative Distance

(b) V2I Relative Distance

(c) V2V Communication Activity

(d) V2I Communication Activity

Figure 7.15 – Interactions in V2I and V2V. (a) and (b) reveal change of relative distance in each of the three vehicle pairs with time, and between vehicles and the data collection point respectively. Troughs in distance lines represent physical proximity points of two nodes. In (c) and (d) blue strips denote that successful communication between nodes was achieved, and blank otherwise. More specifically, $T_{V3\rightarrow V2}$ is found to be 111 s, and $T_{V1\rightarrow V2}$ to be 125 s. For V2I communication in (d), $T_{V2\rightarrow C1} = 126s$, $T_{V3\rightarrow C1} = 131s$ and $T_{V1\rightarrow C1} = 154s$. As V1 and V3 were travelling close together, we could observe comparatively more V2V communication activities between them throughout the trip time.

The motion model tended to predict the positions of V3 faster than the true values due to a slow start up of V3. Before the time point $T_{V3\rightarrow V2}$, V2 did not detect V3 in its communication range. This negative detection information was soon known by C1 through V2I communication and helped correct the faster prediction of V2's position. V2 had a positive relative range measurement about V3 through V2V communication with it at the time point, which was again known by the base station and used to improve the estimate of the position of V3. This is demonstrated in Figure 7.16(b) at around the time marked $T_{V3\rightarrow V2}$. Noteworthy is that V3 did not have egocentric GPS updates, so the estimate improvement was contributed by the V2V relative information and the GPS update of V2. The position estimate was further refined when V3 established V2I communication with C1, as illustrated in Figure 7.16(b) at the time marked $T_{V3\rightarrow C1}$.

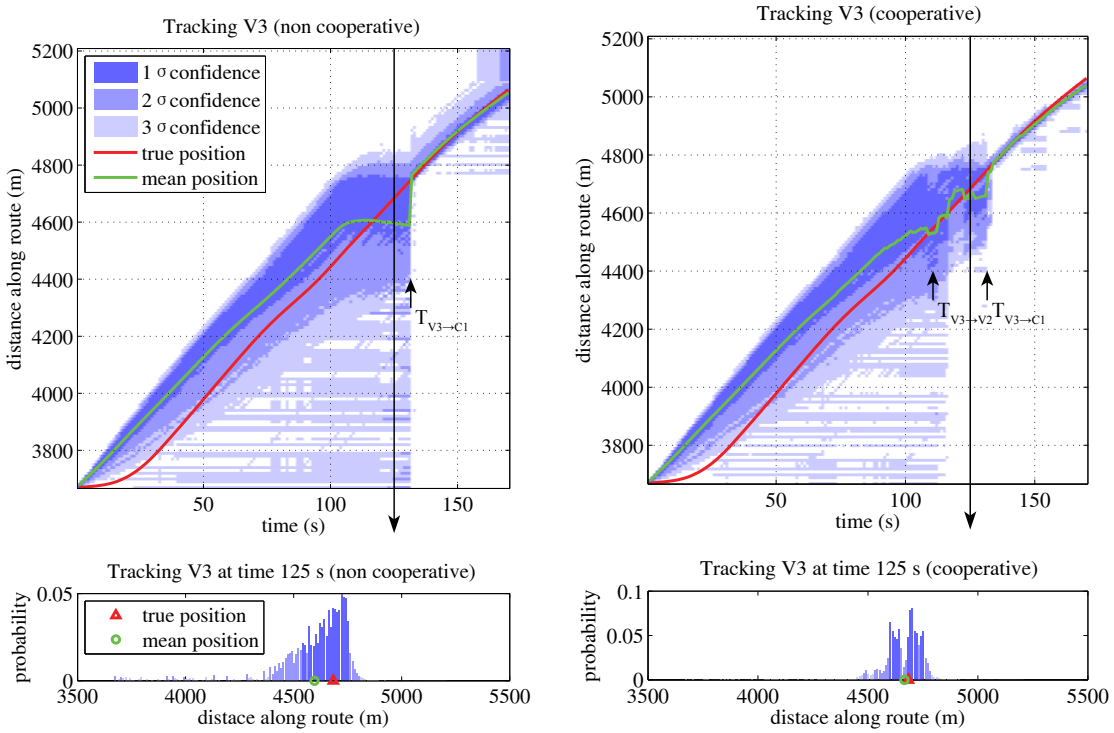


Figure 7.16 – Cooperative tracking results of Experiment II. The experiment results presented in this figure are continued in Figure 7.17. In the experiment, V2 met V3 followed by V1. The base station had to estimate the positions of V3 and V1 by prediction until they established V2I communication with C1 at time $T_{V3\rightarrow C1}$ (131 s) and $T_{V1\rightarrow C1}$ (154 s) respectively. With cooperative tracking enabled, the positive V2V relative range observation between V2 and V3 at around time $T_{V3\rightarrow V2}$ (111 s) were known to the base station via V2, which was in the communication range of C1. The base station was able to update the position estimate of V3, see (b), even when it was out of contact with the data collection point. The non-cooperative tracking result only shows an update in estimation when V3 arrived at C1 at time $T_{V3\rightarrow C1}$. Similarly V2 brought the relative measurements about V1 at around time $T_{V1\rightarrow V2}$ (125 s) to the base station in the cooperative tracking, thereby the position estimate of V1 got updated, as shown in Figure 7.17(b). Noteworthy is that, after V3 established communication with C1 at time $T_{V3\rightarrow C1}$, the relative range observations between V1 and V3 were available to the base station via V3. Consequently the position of V1 was better estimated after that time point, as illustrated in Figure 7.17(b). The non-cooperative tracking result for V1 is given in Figure 7.17(a) for comparison. The tracking results of V2 with cooperative and non-cooperative tracking are almost the same, as respectively shown in Figure 7.17(d) and Figure 7.17(c).

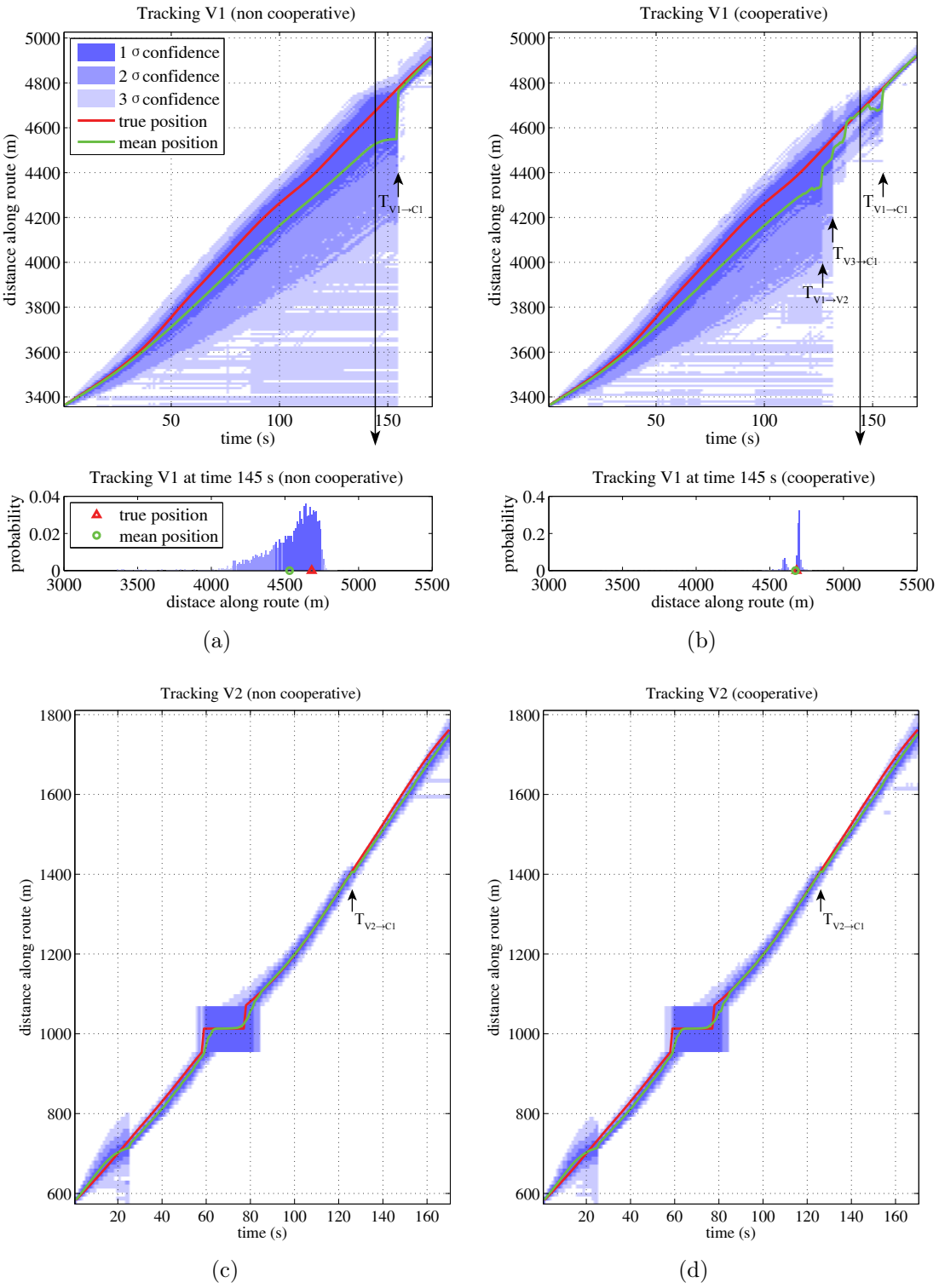
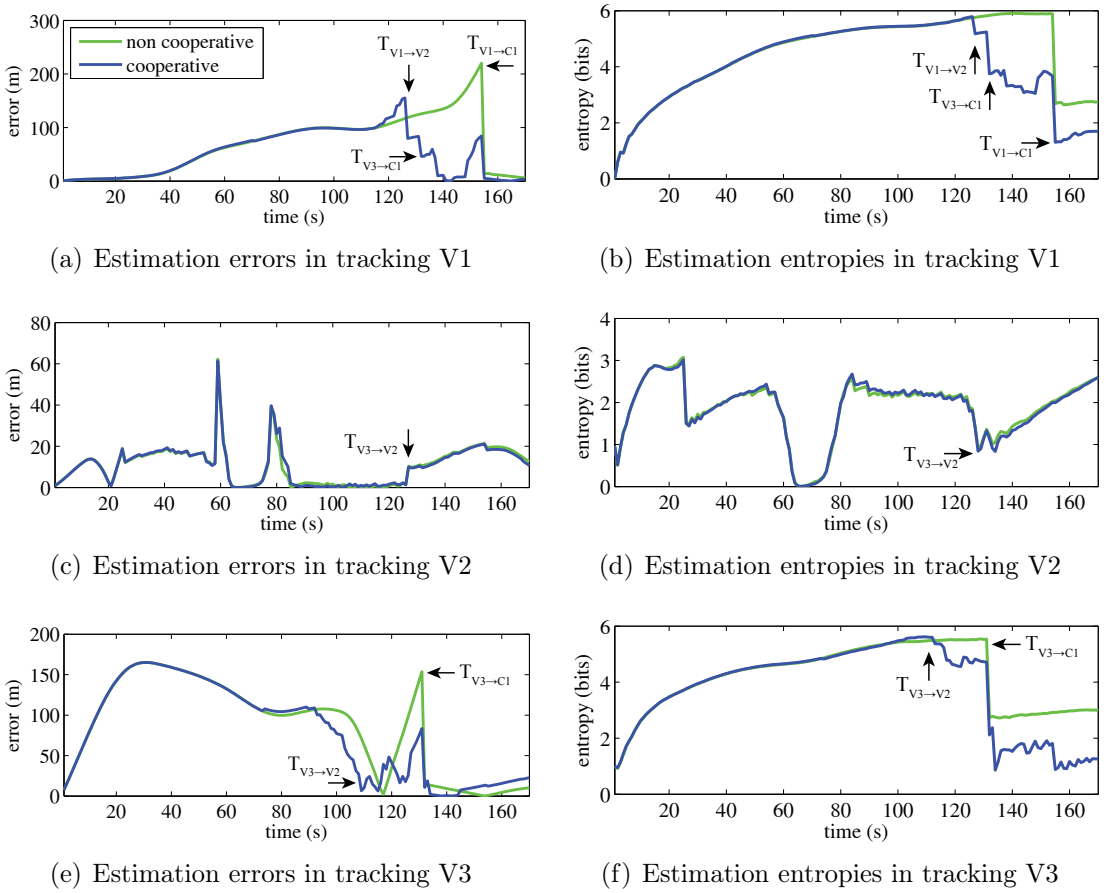


Figure 7.17 – Cooperative tracking results of Experiment II (cont.).



(a) Estimation errors in tracking V1

(b) Estimation entropies in tracking V1

(c) Estimation errors in tracking V2

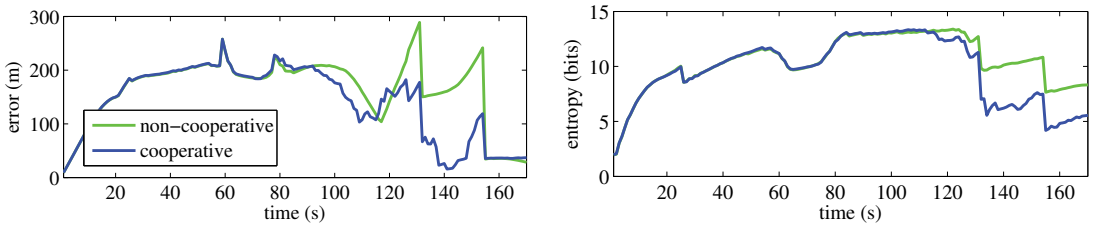
(d) Estimation entropies in tracking V2

(e) Estimation errors in tracking V3

(f) Estimation entropies in tracking V3

Figure 7.18 – Comparisons of estimation errors and entropies. The position errors of V1 with cooperative tracking in (a) initially rose before time $T_{V1\to V2}$ (125 s) and soon dropped, and its entropies was reduced from $T_{V1\to V2}$ onward in (b). The tracking results of V2 with cooperative or non-cooperative tracking did not show a difference that could be told from (c) and (d). For V3, (e) suggests generally a smaller tracking error starting before the time $T_{V3\to V2}$ (111 s), due to the negative V2V relative information enabled in cooperative tracking. The entropies of V3 also began to descend at the time point in (f), when it was tracked in a cooperative way.

Likewise, V1 did not have GPS information and its position estimate was firstly improved at time $T_{V1\to V2}$, when it had V2V communication with V2. During the time $T_{V3\to C1}$ and $T_{V1\to C1}$, the estimate was further improved a few times when V3 from time to time uploaded positive V2V relative observations about V1 to the base station through C1. Lastly, the position estimate was updated when it had a direct V2I transaction with C1, which is illustrated at the time $T_{V1\to C1}$ in Figure 7.17(b).



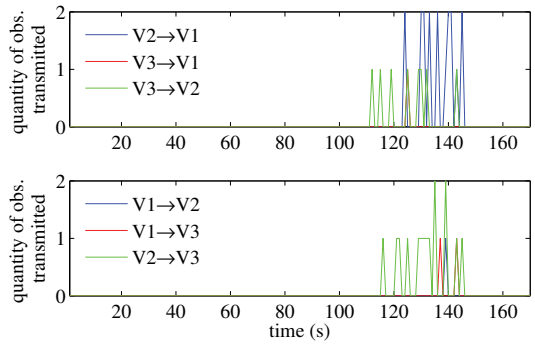
(a) Total estimation errors (b) Fleet entropies

Figure 7.19 – Comparisons of fleet estimation errors and entropies.

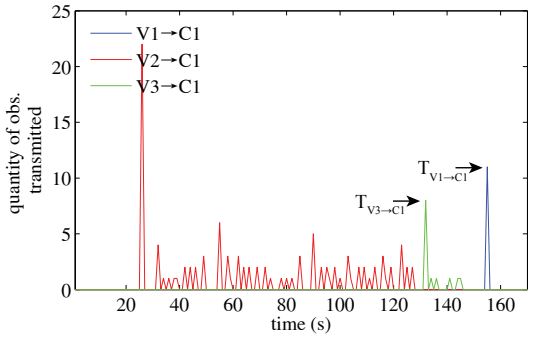
Estimation results without cooperative tracking are demonstrated in Figures 7.16(a) and 7.17(a), in comparison to the cooperative ones. The comparison demonstrates an improvement in tracking performance by the use of cooperative tracking, which incorporates the relative measurements between vehicles. This could also be concluded by comparing estimation errors and entropies of these vehicles, as illustrated in Figures 7.18 and 7.19.

The tracking of V2 does not show much difference with cooperative or non-cooperative tracking, as respectively shown in Figures 7.17(d) and 7.17(c), also in Figures 7.18(c) and 7.18(d). This could be explained by that, V2 kept in contact with C1 for most of the time. V2 moved out of the communication range of C1 at the time $T_{V2\rightarrow C1}$. After that, there was a position update in the cooperative tracking when V1 brought the delayed GPS update of V2 to C1. The improvement, however, is not obvious enough when we compare the two figures.

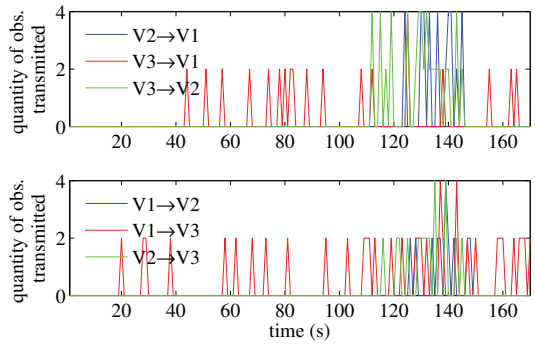
The bandwidth costs of V2V communication for both absolute and relative observations are negligible. They are bounded by $N_v - 1$ and $(N_v - 1)^2$ respectively as presented in Table 5.3. The values in the experiment containing $N_v = 3$ vehicles are validated in Figure 7.20. The V2I bandwidth costs in the experiment were dominated by three major transactions, at the time V2 firstly communicated with C1 (25 s), and time $T_{V3\rightarrow C1}$ (131 s) and $T_{V1\rightarrow C1}$ (154 s), as illustrated in Figures 7.20(b) and 7.20(d). At 25 s, V2 transmitted 22 absolute observations and 52 relative. V3 contributed 8 absolute and 322 relative at time $T_{V3\rightarrow C1}$ (131 s). Lastly these values for V1 were 11 and 378 respectively at $T_{V1\rightarrow C1}$ (154 s). These values are much less than



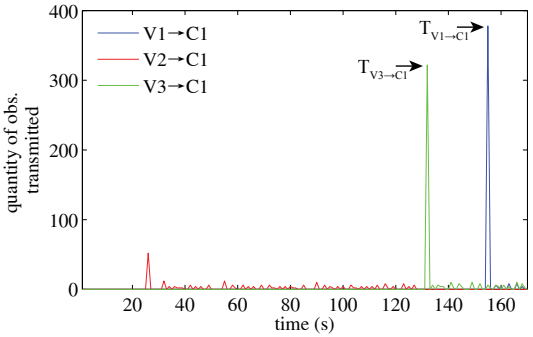
(a) V2V communication bandwidth (absolute)



(b) V2I communication bandwidth (absolute)



(c) V2V communication bandwidth (relative)



(d) V2I communication bandwidth (relative)

Figure 7.20 – Communication bandwidth costs in V2I and V2V. (a) and (b) respectively show the quantity of absolute and relative observations that vehicles synchronise with each other when in the communication range. When vehicles are within the communication range of any data collection points, the numbers of absolute and relative observations they contribute to the base station are illustrated in (c) and (d) respectively.

the maximum V2I bandwidth costs predicted by $N_v \tau^{v_p \to I}$ for absolute observations, and $N_v \left(N_v - 1\right) \tau^{v_p \to I}$ for relative measurements. These actual bandwidth costs of relative observations could be further reduced when compression of data is enabled, as a negative relative observation contains only simple non-detection information.

7.7 Summary

This chapter has presented a particle filter, which can be used to track multiple agents cooperatively with delayed observations. The proposed approach, namely delayed-

state cooperative particle filter (DSCPF), extends the previous work in **Chapter 6** toward delayed-state filtering. It features forward filtering and backward smoothing, to fuse observations that are received by the fusion centre with some time delay.

There are several conclusions that could be drawn according to the simulation and experiment results. The first is that the DSCPF performs close to the optimal parametric tracking approach in linear and Gaussian simulation cases. It is also confirmed that the DSCPF, as a nonparametric tracking approach, demonstrates superior performance in non-Gaussian cases. Furthermore, it has been illustrated in the experiments that inter-vehicle interactions combined with a limited number of data collection points enable vehicle cooperative tracking without the assumptions of full network coverage, and that every vehicle is equipped with GPS sensor. Compared to cases where vehicles are tracked individually, the cooperative tracking is proven to be beneficial in improving estimates according to the experiment results.

The proposed cooperative tracking algorithm is considered an essential component of the approaches that are developed in this thesis, to particularly deal with the vehicle tracking problem in large environments. The positions of a vehicle could be predicted by the long-term motion prediction approach proposed in **Chapter 4**, when its information is not available to the base station. With the observation harvesting mechanism proposed in **Chapter 5**, V2V relative ranging measurements, together with vehicle egocentric observations, are shared among vehicles and forwarded to the base station via returning vehicles. Lastly, in the use of the particle filter proposed in this chapter, the base station is able to cooperatively track vehicle positions with delayed information harvested, combined with V2I measurements from data collection points distributed in the area.

Chapter 8

Conclusions

8.1 Introduction

This chapter summarises the contributions of this thesis in Section 8.2 and also outlines the directions of future research in Section 8.3. A conclusion is finally presented in Section 8.4.

8.2 Summary of Contributions

This thesis studies the problem of tracking multiple vehicles in large environments. It presents a set of approaches to address the observation outage problem, which inevitably occurs when the tracking is required over a large area of operation.

Most of the existing techniques are ineffective for tracking mobile agents in large environments. This is because of the lack of environmental properties that need to be incorporated into the prediction of vehicle positions when it is required over a long period. It is also essentially due to the incapableness of fusing the time delayed tracking information, which is disseminated through a fragmented network. Furthermore, the traditional cooperative tracking approaches are unable to work in applications with the presence of non-Gaussian properties in motion and observation models. The

thesis overcomes these limitations and proposes novel approaches to address the tracking problem in large environments. The contributions of this thesis are described as follows.

Contribution 1: Definition of Observation Types and Their Uses

The thesis defines all types of observations that could be used in constraining possible vehicle positions. The traditional tracking approaches only consider a portion of these types. The observations are grouped into absolute and relative, positive and negative, real-time and time delayed. Among them, negative and time delayed information also assists in improving estimates in the absence of positive and real-time observations, respectively. The fusion of V2V relative measurements is also used in the proposed cooperative tracking approach.

Contribution 2: Development of Probabilistic Observation Models for Vehicle Proximity Detection

The observation models were developed based on historical V2V and V2I communication data. Their purpose is to evaluate proximity information about a vehicle's position relative to another, or to a data collection point, given the availability of the relative observations defined in Contribution 1. The models are based on wireless connectivity information and RSSI measurements extracted during communication activities, and can be continuously improved with more communication data incorporated. They require no additional bandwidth, as nodes necessarily communicate when they meet.

Contribution 3: Development of a Probabilistic Approach for Long-Term Vehicle Motion Prediction

The thesis develops a new approach for long-term vehicle motion prediction, based on a vehicle model that incorporates the learnt properties of the environment. It

enables consistent position prediction for vehicles over a long period, while most of the traditional methods are only suitable for short-term prediction. The model is also based on the assumption that a vehicle has a non-zero probability of stopping on a road. This makes the model a better representation of the real-world situation than conventional approaches that assume non-stopping movements.

Contribution 4: Development of the Observation Harvesting Approach Using V2V and V2I Interactions

The thesis proposes the observation harvesting mechanism, which aims to share the observations amongst vehicles and a fixed infrastructure in an opportunistic manner. Conventional data transmission techniques require continuous network connectivity between a transmitter and a recipient. Different from the conventional methods, the proposed approach introduces a store-and-synchronise concept to deal with network disruptions in large environments. It turns the observation harvesting into an effective, robust, and lightweight data dissemination method without full network coverage.

Contribution 5: Development of a Multiple-Vehicle Tracking Approach Using Delayed Observations

A new particle filtering algorithm is proposed for the tracking of multiple vehicles, by combing Contributions 2, 3, and 4. The algorithm incorporates the delayed information into the global position estimation when a vehicle brings back, to a data collection point or the base station, egocentric position information of those vehicles not in contact with the infrastructure. During the outage periods of observations, the algorithm estimates the positions of vehicles using the long-term motion prediction model proposed. In this approach, V2V relative position measurements are not yet considered in the fusion stage.

Contribution 6: Development of a Nonparametric Approach for Cooperative Tracking

Despite a minimal computational cost, parametric cooperative tracking approaches require the Gaussian assumption to be satisfied. For this reason, a novel nonparametric algorithm is developed to provide a more generalised approach particularly for cooperative tracking. In traditional PFs, the high-dimensional nature of cooperative tracking problems leads to low efficiency of importance sampling. This limitation is overcome in the thesis by introducing the Gibbs sampler into particle filtering, to sample from low-dimensional target state space conditionally on others, rather than directly from the joint state. The research applies equally to various applications where cooperative localisation/tracking is a requisite.

Contribution 7: Development of a Cooperative Vehicle Tracking Approach with Delayed Observations

A cooperative vehicle tracking approach is developed, by extending the work in Contribution 6 toward delayed-state filtering and combining together Contributions 2, 3, and 4. The approach is able to perform long-term motion prediction and to fully utilise all types of observations harvested, including the time delayed and relative. Compared to Contribution 5, the approach is additionally capable of considering V2V relative measurements in its fusion stage to further constrain position estimates. This, essentially, enables the tracking of those vehicles with degraded GPS accuracy or GPS outages.

Publication Information

Contribution 3 was included in the first journal paper [150], published in the IEEE Transactions on Intelligent Transportation Systems, and also in an earlier paper [149], in the 14^{th} IEEE International Conference on Intelligent Transportation Systems (*ITSC'11*). Contributions 1, 2, 4, and 5 were included in the second journal paper [151], published in the IEEE Transactions on Intelligent Transportation Systems.

Contribution 6 was included in a paper [152] in the IEEE International Conference on Robotics and Automation (*ICRA'14*). Contributions 6 and 7 were included in the third journal paper, which is to be submitted for publication.

8.3 Directions for Future Work

The future work of the proposed approaches in this thesis generally contains four components, which are detailed as follows.

Improving the Vehicle Motion Model

The probabilistic motion prediction approach has demonstrated that a motion model must incorporate both the vehicles' and environmental properties to successfully perform long-term prediction of a fleet of vehicles. This approach can be improved further with an on-line estimation of model parameters to provide continual improvement of the vehicle model. This will allow the properties of the individual vehicle to be incorporated into the model. Additional information regarding weather conditions can be used to consider the model changes caused by environmental conditions such as rain, fog and snow. Also, the motion model could be improved by taking into account different driving patterns, as results of different drivers. It is clear that improving the target motion model using these techniques will improve the overall performance of the algorithm. Furthermore, the approach could be extended to include other types of resources, such as personnel. These concepts indicate the general future research directions for this approach.

Further Study on Optimising the Harvesting Mechanism

The proposed observation harvesting mechanism has proven to be an effective method for sharing information in a delayed/disrupted network, by taking advantage of V2V and V2I interactions. As mentioned in Section 5.4.4, the accuracy level of the tracking

approach is mainly determined by the length of the "blind time". The future work will include a further examination of the optimisation of locations for data collection points and other mechanisms for reducing the blind time, increasing the availability of harvested information, as well as optimisation in the robustness of the approach against conflicting information. This is expected to lead to a reduced, or potentially bounded position estimation error for the fleet of vehicles. The optimisation will be an important outcome since it would enable other productivity-related tasks, such as dispatching.

More Accurate and Reliable P2P Measurements

The accuracy provided for V2V and V2I relative measurements is adequate for the long-term vehicle tracking scenario investigated in this thesis. Nevertheless, the tracking approaches will benefit from a higher degree of accuracy of the P2P relative measurements. The RSSI data, with which the observation models were built up, is known to be noisy and unreliable due to environmental factors. In the future work, a more advanced P2P ranging means (e.g., UWB based) is preferred, in order to pursue a higher precision level and better reliability of relative position measurements between nodes.

A Closer Look at the Gibbs Sampler based Tracking Approach

The Gibbs sampler based tracking algorithm introduced in this thesis has shown its great potential as an accurate, and more importantly a tractable nonparametric cooperative tracking approach in applications where the Gaussian assumption is not valid. Admittedly, the study and implementation of this novel approach in this thesis are at a preliminary stage. The future work will include a more detailed investigation into the fundamental principle of the Gibbs sampler based tracking framework. More simulations and field experiments, along with possibly better analysis methodologies are required for an in-depth examination. Furthermore, the algorithm will be optimised toward a more efficient sampling process and a reduced computational cost.

8.4 Summary

The thesis has made a significant contribution to the field of mobile agents tracking in large environments. The research paves the way for the implementation of resource optimisation algorithms when full coverage networks are not available or feasible. This facilitates another area of research, regarding resource planning incorporating vehicle uncertainty due to discrete positioning information.

It is of fundamental importance to optimise the utilisation of vehicle resources in industrial applications. Current fleet monitoring and dispatch systems require full network coverage to facilitate the planning of vehicle operations. The algorithms presented in this thesis will provide an estimate of vehicle position, with associated uncertainty, at all times. This will enable the development of a new set of optimisation algorithms to implement fleet monitoring and vehicle dispatching without the infrastructure and maintenance expenses of a full coverage communication network.

Appendix A

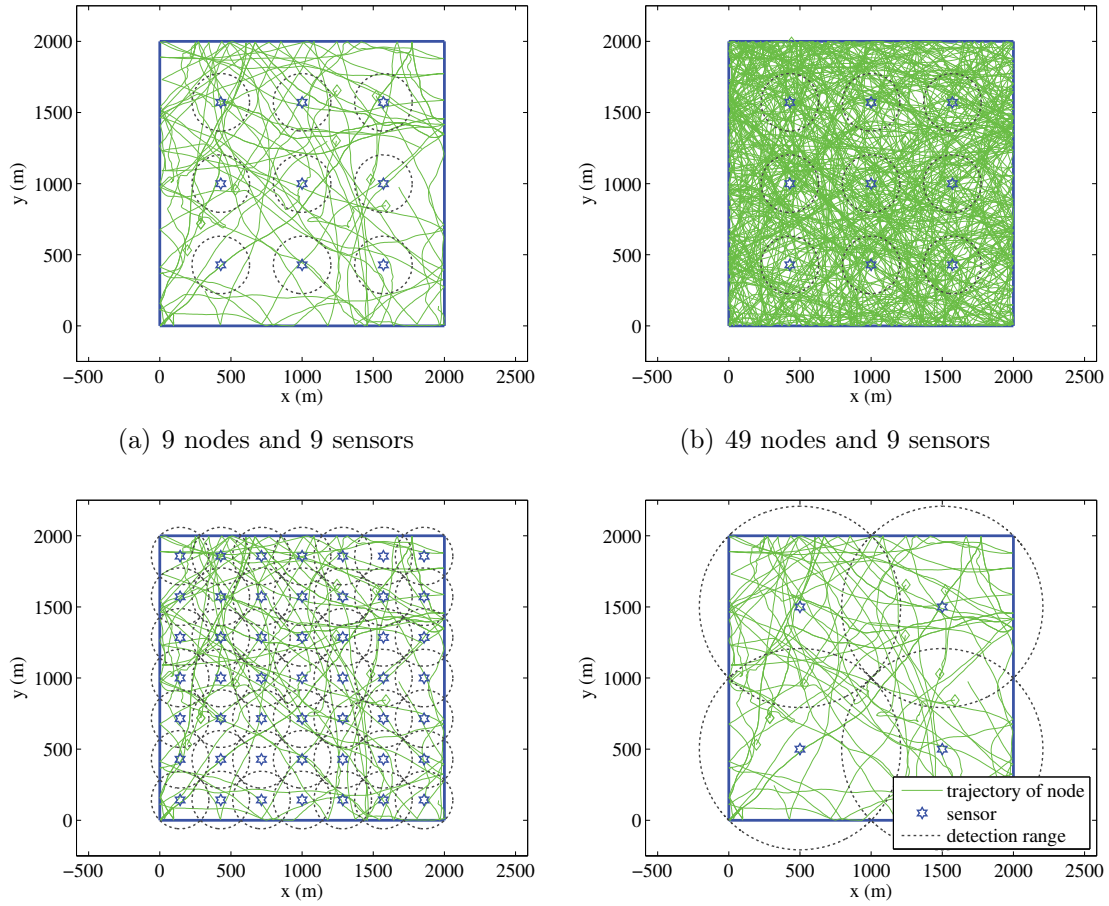
Reducing Observation Outage Duration: A Preliminary Study

The observation outage rate per se is a simple mathematical problem based on the percentage of the area that is not yet covered by sensors. Roughly, as a complement number to the observation outage rate, the sensor coverage rate is proportional to the number of sensors and the area that a sensor can cover, when not taking into account the overlapping of multiple sensors.

$$\eta_s \propto N_s \times A_s$$

Hence, the ease of observation outage issue appears to be achievable by considering one or two of the following options.

1. Increasing sensor quantity/density: With detection range unchanged, a larger number of sensors, or a higher sensor density in other words, results in a higher portion of the field area being covered. And because of this, for a more percentage of time a node could be detected by any of these sensors when moving inside the area. Theoretically, the quantity of sensors needed to fully cover a square area should fulfil:



(a) 9 nodes and 9 sensors

(b) 49 nodes and 9 sensors

(c) 9 nodes and 49 sensors

(d) 9 nodes and 4 sensors with longer detection range

Figure A.1 – Setups of nodes and sensors in a large area. Diamond shape represents starting points of nodes trajectories. Sensors in both (a) and (b) only cover a portion of the area. In (c), covering the entire $2km \times 2km$ area requires in total 49 sensors with a detection radius of 202 m. As shown in (d), the quantity of sensors in need for full area coverage is reduced to only 4 with the detection range extended to 707 m. The figures only show nodes trajectories during the first half an hour.

$$N_s \geq \frac{L^2}{2 \times R_s^2} \tag{A.1}$$

where L is the side length of the square. R_s is the detection range of a sensor. The denominator equals the area of the largest square that a circle with radius R_s is able to cover.

Comparing Figure A.1(a) and A.1(c), the latter one illustrates a full sensor cov-

erage case, where observation outage no longer occurs by distributing 49 sensors around the area. Furthermore, the estimate is improved with observations coming from more sensors.

2. Increasing sensor detection range: Alternatively, by extending sensor detection range a moving node could stay under detection of sensors for a longer period of time. According to Equation A.1, the minimum sensor quantity required for full coverage could be reduced with a longer sensor detection range. The sensor quantity for area-wide coverage in Figure A.1(d) is dropped to 4, much less than 49 in Figure A.1(c). This is obtained by adding detection range from 202 m to 707 m.

Another improvement is to deal with the distribution of sensors, taking the environment properties into consideration.

3. Optimising sensor arrangement: These target nodes may concentrate their activities on particular parts of the operation area, depending on tasks that they undertake, as well as structure of the field, such as geographical shape, terrain, road network and etc. Figure 1.6 has demonstrated features in a mining operation. Under this situation, an optimised sensor arrangement will necessarily bring a lower probability of observation outage. One of examples is to place sensors near to intersections with heavy traffic load.

A brief investigation is carried out aiming at reducing observation outage duration. In addition to the first two options, we also consider the following two methods in the investigation.

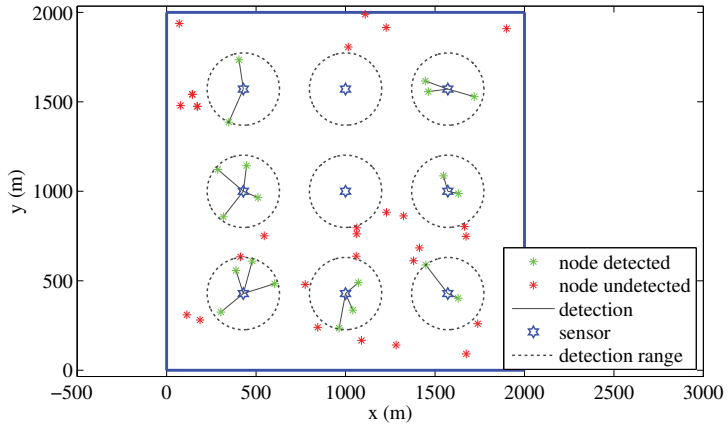
4. Introducing pair-wise collaboration between target nodes: In independent tracking scenarios, targets are tracked by sensors only. This is changed by enabling sensing techniques on targets, such as range finder, wireless communication, and so on, so that multiple targets can measure the pair-wise information between each other. This information is shared among targets and forwarded to

infrastructure. Inside the fusion centre, the inter-target information is considered in addition to the measurements generated by infrastructure sensors for an improved position estimate. Figure A.2 illustrates the difference between tracking cases before and after the pair-wise collaboration is introduced. The undetected nodes are significantly reduced in Figure A.2(b) as some of those beyond the detection range of sensors could be tracked by nearby neighbours.

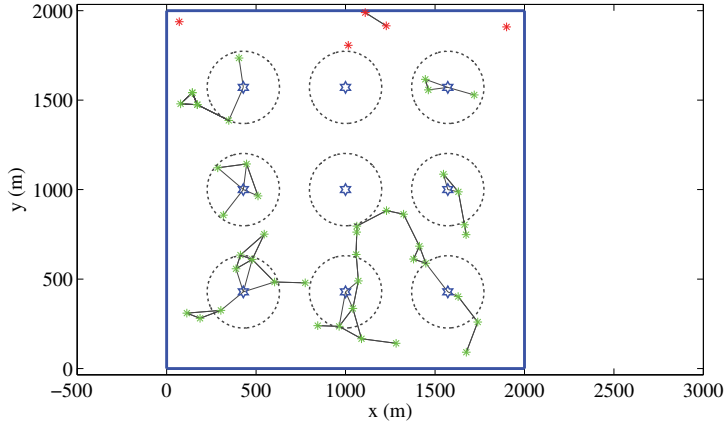
5. Increasing node quantity/density: The quantity or density of nodes does not affect the tracking performance in tracking cases where every node is tracked independently. When the inter-node collaboration concept is used however, the quantity of the nodes matters. These nodes are now considered "moving sensor" under this configuration. Thus an increased target node quantity brings the similar effect to raising up the count of sensors.

The investigation begins with setting up various numbers of stationary sensors and mobile nodes moving randomly in a square of $2km \times 2km$, as having been illustrated in Figure A.1. The effect by different detection ranges of the sensors is also analysed. In addition, the problem is further studied with inter-node collaboration concept incorporated. The blue line in Figure A.3(a) depicts the descent of average outage duration along with increase of sensor quantity. However, placing sensors everywhere in many applications with large environments is not an economically practical option. Besides, it also ends up with a limited scalability of the tracking system. Figure A.3(b) also suggests a lower average outage duration in the use of a longer detection range between sensors and targets. However accurate and reliable detection over a long range remains a challenge in reality using existing technologies, because of constraints such as power, size, and cost. Long range wireless communication is also prone to frequent interruption due to noise and environmental dynamics. Despite their limitations in practice, long-range detection and full sensor coverage theoretically are considered the ultimate solution to realise global tracking of targets in large environments.

According to Figures A.3(a) and A.3(b), enabling collaboration between nodes further reduces the mean observation outage time in the study. This is achieved when nodes



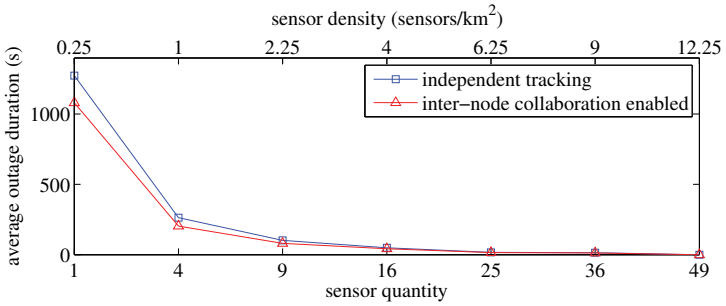
(a) Before



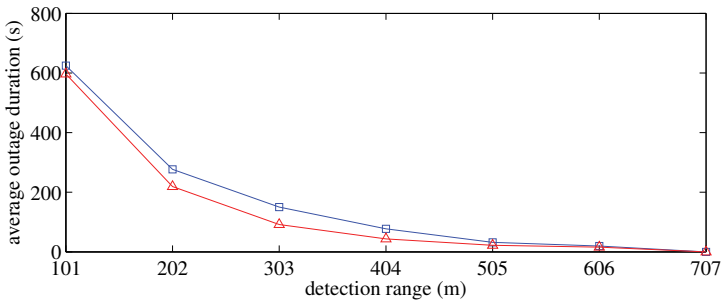
(b) After

Figure A.2 – Tracking 49 nodes before and after inter-node collaboration is enabled. (a) shows 20 nodes detected, leaving the rest 29 out of detection range. The undetected nodes are reduced to only 5 in (b), where nodes are able to help track each other.

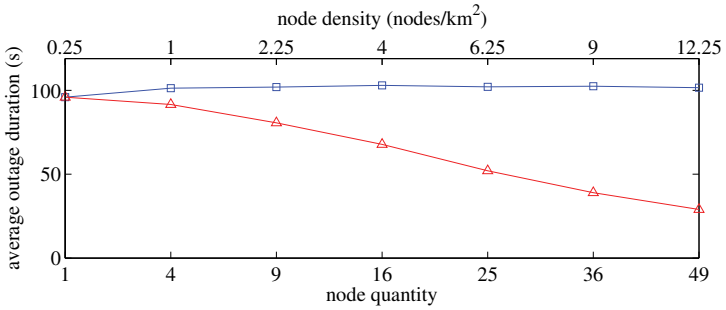
themselves can help detect each other in addition to the existing sensors. Figure A.3(c) demonstrates the improvement by further taking advantage of the pairwise collaboration. A more reduced outage duration is observed in the figure, as a result of a larger number of nodes in the collaboration enabled case. While the average outage time is not affected by node quantity when the collaboration capability is not equipped.



(a) Sensor quantity/density



(b) Detection range



(c) Node quantity/density

Figure A.3 – Analysis of average observation outage duration against nodes and sensors quantities, and detection range. (a) and (b) show results with 9 nodes, while quantity of sensors in (b) and (c) is set to be 9 as well. Sensors in both (a) and (c) have a pre-determined detection range of 202 m.

Appendix B

A Supplementary Review on Mobile Ad-Hoc Networks

B.1 Pure MANETs

Mobile ad-hoc networks (MANETs) consist of mobile nodes which are able to communicate with each other via multi-hop and P2P wireless connection. A MANET is spontaneously and dynamically formed by these mobile nodes without the requirement of any infrastructure or prior network configuration, which is considered the key benefit of it. The properties of MANETs have been discussed in Section 2.5.1, in a comparison to the infrastructure networks. A pure MANET refers to a mobile ad-hoc network with a general purpose, without dependency on applications. It possesses highlights like good mobility, dynamic multi-hop network structures, and direct P2P communication [144].

In order to meet different requirements of various applications, the pure MANETs could be further extended to one of the following variants: WMNs, WSNs, VANETs and DTNs. There are some salient challenges in design of a pure MANET or one of its variants, three of which are summarised as below.

1. Extra System Overhead for Routing: Communication within a MANET could

be categorised into application and infrastructure communication. Application communication refers to transfer of application related information. The networking process is infrastructure communication, which is correlated to data transmission required in establishment, configuration and optimisation of the network. The system overhead for networking grows along with the quantity of nodes in the network. The routing techniques could be generally classified to proactive and reactive types [91], with the difference of whether or not a routing state table is maintained in each node. The extra overhead for routing could not be eliminated as long as the routing process is necessary in the network, though reactive approaches will cost lower communication overhead compared to the case with proactive ones.

2. Network Dynamics: The topology of a MANET changes very rapidly due to mobility of nodes [179]. It is further complicated by node enrolment and leave without prior notification. Traditional routing approaches that are based on fixed infrastructures are difficult to be applied to MANETs as routing messages from or to moving nodes is more challenging. Unlike the case of a deterministic deployed network, where positions of nodes are pre-determined and data routing paths remain unchanged once established, the mobile nodes in a MANET have to deal with dynamic routing issue to keep the network self-organised and self-configured. Along with the change of network topology, the route discovery operation is required to perform frequently in order to update routing information and maintain the whole network. Many potential paths may have disappeared before they can be utilised due to rapid and unpredictable changes in connectivity.

3. Resource Constraints: Most of wireless nodes in MANETs are tightly resource constrained in terms of energy, processing and storage, due to lack of infrastructure. For this reason, efficiency of resource utilisation ought to be fully considered when designing ad-hoc wireless networks. Furthermore, in some cases nodes are of different wireless communication capability, power capacity, performance and so on, which require complicated coordination of these nodes

as well as distributed algorithms to maintain a dynamic network topology [179]. This is particularly essential when the network presents hierarchical property. The topic is further elaborated in Section 2.5.6.

Wireless LAN with ad-hoc configuration in IEEE 802.11b standard has been adopted in "Virtual Mine" [123], "Virtual Network System" [99], and the work by Ruff and Holden [140] for close proximity warning systems.

B.2 WMNs

Wireless mesh networks (WMNs) [117] emerge in recent years as a particular type of MANETs and it is aiming to provide high bandwidth Internet access service for civilian applications. As different to the flat architecture of pure MANETs, a WMN is a hierarchical network consisting of two tiers: mesh backbone and mesh clients. The mesh backbone comprises stationary wireless mesh routers (MRs) inter-connecting to form a mesh through wireless links. Those MRs with wired connections further act as Internet gateways (IGWs) bridging Internet and the mesh network. Mobile wireless devices such as cell phones, laptops and tablets turn to be the mesh clients in a WMN. These clients connect to adjacent MRs and are able to access Internet through IGWs in the use of multi-hop communication. Because of the self-configuration and self-organisation properties inherited from pure MANETs, a WMN can be flexibly modified by adding MRs into or removing away from the network. The wireless connection between MRs also enables rapid and cost-efficient deployment of the network.

The main improvements of WMNs over pure MANETs could be summarised to the following three points.

1. More stable topology. The mesh backbone (i.e. MRs and IGWs) is usually stationary or with little mobility. This reduces the networking overhead consumed on re-establishing the network when the topology is changed. It also lowers down the probability of connection breakage and thus improves network throughput.

2. Rich energy of MRs. The MRs and IGWs, which form mesh backbone in WMNs, are usually equipped with abundant power supply. Nevertheless, the mesh clients are still subject to power constraint, the same as these mobile nodes operating on batteries in MANETs.

3. Multichannel of transmission. The radio on each MR is configured with multiple orthogonal channels through which data could be transmitted and received simultaneously without interference [117]. The network bandwidth in WMNs is thus substantially improved as compared to the situation of transmitting data over a shared channel in pure MANETs.

The technical standards of WMNs have been defined in the IEEE 802.11s, 802.15, 802.16, and 802.20. Among them, IEEE 802.11s defines the physical and MAC layers protocols for WMNs.

B.3 WSNs

A wireless sensor network (WSN) contains hundreds or thousands of sensor nodes, which are able to collect various data from the physical world [11]. The goal of WSNs is to detect and estimate some certain phenomena or events, or to track states of certain objects in an area of interests according to data type collected. The information is collected, preliminarily processed by sensor nodes and finally forwarded to a base station (a.k.a fusion centre). So sensor nodes are capable of communicating either among each other or directly to the base station. Basically, each sensor is comprised of sensing, processing, transmission, mobiliser, position finding system, and power unit (some of these components are optional like the mobiliser). As a single sensor node is usually a tiny wireless device operating on a battery, energy efficient routing and information processing are amongst main areas of research on WSNs. Some related work on the topic of energy efficiency issue in a WSN will be presented in Section 2.5.5. Applications of WSNs include searching [30], military [12], traffic surveillance

[40], environment monitoring [9] [90] and many more. A comprehensive review on WSNs and their applications could be found in [10].

WSNs have been widely used for localising/tracking single or multiple mobile agents. In this application scenario, the measurements made by sensor nodes on states of the targets are sent over the network to the fusion centre possibly via multi-hop communications. The fusion centre is then responsible to perform data fusion process and infer positions of the targets. The work in [164] develops an approach to track a single moving target in a WSN. In an experiment presented in [145], 33 position-unknown nodes out of which 7 are mobile, are tracked accurately by using an ad-hoc network and 5 GPS referenced nodes. The unknown nodes in [12] are localised for fire-fighter or military operation application in indoor environment in the use of anchors, which have access to GPS position. A decentralised anchor-free algorithm is proposed in [136] to localise a set of static position unaware nodes in a sensor network given pairwise distance information. The work presented in [126] focuses on the decentralised tracking of multiple mobile targets in a WSN. More examples about localising/tracking nodes in WSNs are presented in [131], [61], [39] and [88].

The standard IEEE 802.15.4 defines the physical and MAC layer protocols for remote monitoring and control, as well as sensor network applications [143]. ZigBee is an IEEE 802.15.4 standard based technology for industrial and commercial applications. Depending on RF environment and the power consumption required for a given application, ZigBee compliant wireless devices are expected to transmit a range of 10-75 meters operating in the ISM band.

Appendix C

Vehicle Proximity Detection

C.1 Introduction

The details on the vehicle proximity detection approach is presented in this appendix. Section C.2 presents background on proximity detection of vehicles. The probabilistic observation models built with V2V and V2I communication data are introduced in Section C.3, and the likelihood functions are presented in Section C.4. Lastly, Section C.5 summarises this appendix.

C.2 Proximity Detection

In the environment under investigation, apart from the absolute egocentric observations provided by on-board GPS devices, relative information comes from the proximity detection between vehicle nodes (V2V) and between vehicle and infrastructure (V2I), i.e. identifying the approximate distance between a pair of nodes. Conventionally, we use received signal strength (RSS) as a means of relative range measurement. However, it suffers from low accuracy as the RSS varies greatly due to environmental dynamics. Rather than using RSS information as the only method, a better solution is to combine RSS measurements with other detection means, one of which is wireless

connectivity. The advantage of using the hybrid method is manifest because no additional bandwidth is required to measure RSS and connectivity, as nodes necessarily communicate as they meet.

Connectivity means whether or not a packet from the transmitter can be successfully received and decoded by the receiver. It depends a lot on transmission power, environment and noise brought in the transmission channel. Wireless transceivers installed at each node are used to transmit and receive data, and in the meanwhile to provide connectivity information, i.e., binary quantisation of RSSI measurements [129], without considering signal propagation models. Therefore, the wireless connectivity information is a binary variable, either *Positive Connectivity (PC)* which means two nodes can successfully communicate, or *Negative Connectivity (NC)* if cannot.

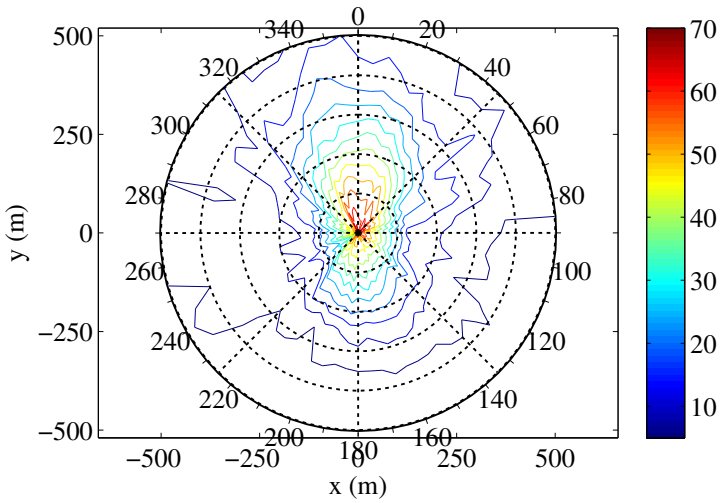


Figure C.1 – Example of irregularity in connectivity: A haul truck with two antennas mounted at head and rear shows areas with higher chance of wireless connectivity near two ends, as a result of more concentrated radio signal propagating along with front and back directions.

Proximity refers to identifying only whether or not a vehicle is within the detection range of a sensor/observer. In wireless communication context, the detection range of a node theoretically equals to its signal coverage. Accordingly, the proximity can be mathematically defined as a binary value: either *In Range (IR)* or *Out of Range (OR)*. Nevertheless, inequality pattern of radio power propagation along with different directions in reality complicates signal coverage shape. Figure C.1 shows

an example of connectivity as a function of distances and angles. This is because of electromagnetic interference (EMI), antenna configuration and other environmental factors, such as reflection, blockage and multipath effects. Therefore in the real world, negative connectivity does not necessarily mean two nodes are out of range, while positive connectivity infers a high probability that two nodes are in communication range.

Instead of a simply deterministic relationship between connectivity and proximity, probabilistic observation models are adopted in the thesis aiming at reliable vehicle proximity detection given V2V or V2I connectivity and received signal strength intensity (RSSI) measurements. The thesis also considers both positive and negative information [8] [74] in the tracking algorithms. As mentioned in Section 3.2.2, the positive information is given by the detection of a vehicle in sensor range indicating its presence. The non-detection of the vehicle is considered negative information and is useful to exclude possibility that the vehicle is in this particular area. As the signal coverage range is limited to only a few hundred metres, the vehicle proximity detection, as an approximate ranging approach, provides only rough information on a vehicle's position relative to another, or a fixed data collection point.

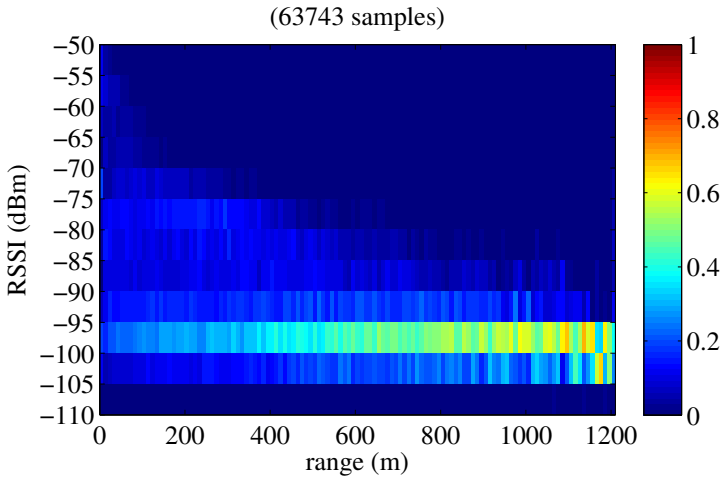
In the proposed approach, an observer is only required to provide connectivity observations, and RSS when the connectivity is positive. There are two types of observers:

1. Wireless transceivers mounted at fixed data collection point.
2. Wireless transceivers attached to mobile vehicles.

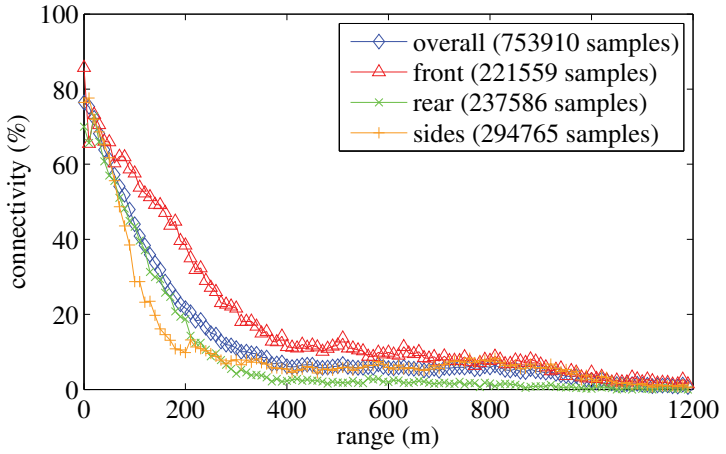
Based on this, two proximity detection models for V2I and V2V were constructed, in the use of historical communication data.

C.3 Distance-Bound Sensor Models

The correlation between RSSI and distance for V2V communication is revealed by Figure C.2(a), while Figure C.2(b) summarises the connectivity information. The



(a) RSS along with distance



(b) Chance of connectivity along with distance

Figure C.2 – Overall RSSI and connectivity along with distance in V2V communication. a) demonstrates a generally logarithmic dropping of received signal power along with linear increment of range between transmitter and receiver, though with great noise and fluctuation as a consequence of multipath fading. The descent rate of connectivity differs at front, rear and sides directions in b), due to the non-uniform distribution of radio signal power.

different descending curves of connectivity could be explained by the non-uniform radio signal power density at different directions. The irregularity in connectivity also results in an ambiguous boundary between the *IR* and *OR*. Therefore a slightly more sophisticated definition of proximity should be used to better describe irregularity characteristic of wireless connectivity in V2V communication.

A distance-bound sensor model was built up for V2V proximity detection, which divides the range from 0 to infinite to three partitions: *Close Proximity (CP)*, *Intermediate Proximity (IP)* and *Low Proximity (LP)* and are defined as:

$$V2V\ Proximity = \begin{cases} CP, & 0 \leq distance < 200 \\ IP, & 200 \leq distance < 400 \\ LP, & 400 \leq distance < \infty \end{cases}$$

Based on the connectivity and RSSI data presented in Figure C.2, the sensor model for V2V communication could be built with histogram data shown in Figure C.3. Lastly, the V2V proximity detection model could be built up with likelihood matrices demonstrated in Figure C.5(a).

Similarly the sensor model for V2I proximity detection was also constructed and illustrated in Figure C.4, but with the distance partitioned into two:

$$V2I\ Proximity = \begin{cases} CP, & 0 \leq distance < 250 \\ LP, & 250 \leq distance < \infty \end{cases}$$

The built vehicle proximity detection model for V2I is illustrated in Figure C.5(b).

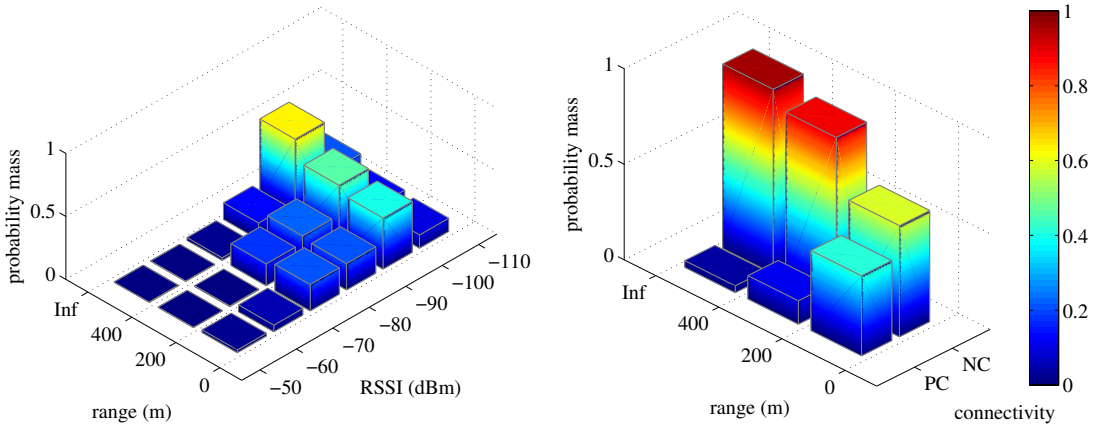
C.4 Likelihood Functions

The likelihood function given a connectivity observation is represented by:

$$P\left(z^{c,p\to q} = z_t^{c,p\to q} | \mathbf{x}_t^p, \mathbf{x}_t^q\right)$$

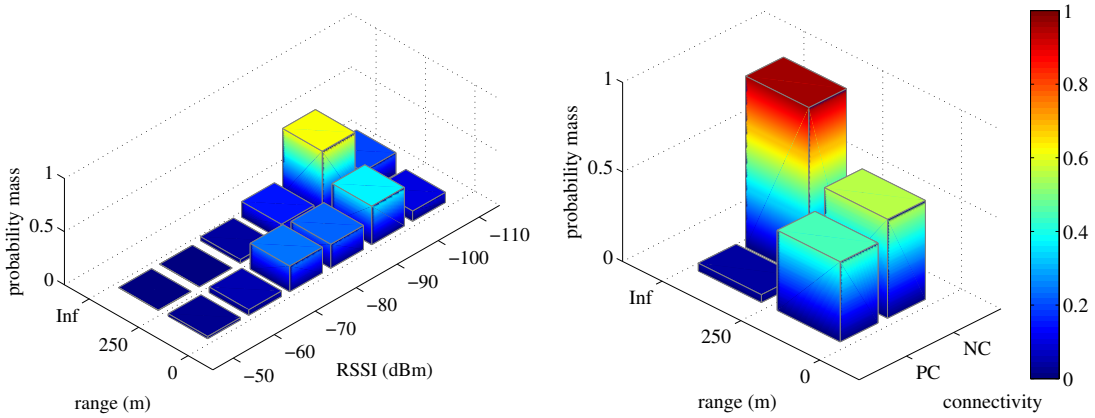
where $z_t^{c,p\to q} \in [PC,\ NC]$ is a relative connectivity observation measured by observer q at time t.

Every observation of a PC event, i.e., $z_t^{c,p\to q} = PC\, p \neq q$, is accompanied by an RSSI measurement in the communication channel, which means the RSSI measurement is



(a) Histogram of RSS (b) Histogram of connectivity

Figure C.3 – Histograms of RSS and connectivity along with distance segments in V2V communication.



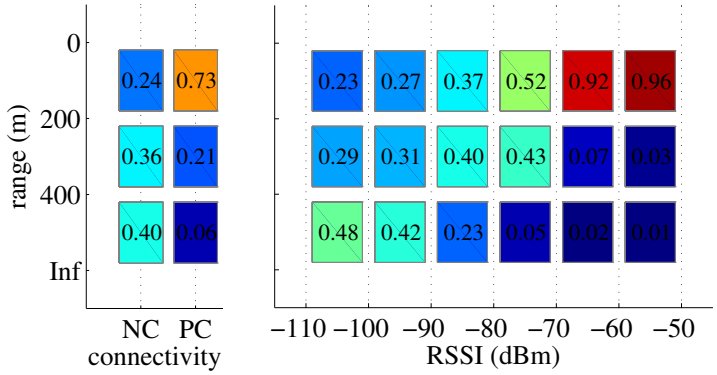
(a) Histogram of RSS (b) Histogram of connectivity

Figure C.4 – Histograms of RSS and connectivity along with distance segments in V2I communication.

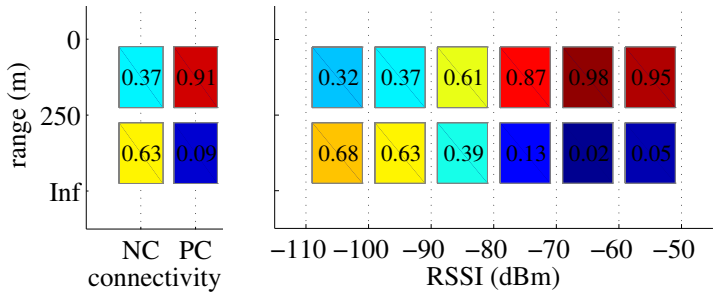
conditional on the occurrence of a PC observation. Therefore the likelihood function given an RSSI observation is denoted by:

$$P\left(z^{r,p\to q} = z_t^{r,p\to q} | \mathbf{x}_t^p, \mathbf{x}_t^q, z^{c,p\to q} = PC\right)$$

where $z_t^{r,p\to q}$ is an RSSI observation measured by observer q at time t.



(a) V2V



(b) V2I

Figure C.5 – Likelihood matrices of distance given wireless connectivity (left) and RSSI observations (right). (a) shows the vehicle proximity detection model for V2V, while (b) shows that for V2I.

Thus the joint of the two likelihood functions is written as:

$$
\begin{aligned}
&P\left(z^{r,p\to q}=z_t^{r,p\to q}, z^{c,p\to q}=PC|\mathbf{x}_t^p,\mathbf{x}_t^q\right)\\
&=\underbrace{P\left(z^{r,p\to q}=z_t^{r,p\to q}|\mathbf{x}_t^p,\mathbf{x}_t^q,z^{c,p\to q}=PC\right)}_{likelihood\ given\ RSSI}\times\underbrace{P\left(z^{c,p\to q}=PC|\mathbf{x}_t^p,\mathbf{x}_t^q\right)}_{likelihood\ given\ connectivity}
\end{aligned}
$$

For the convenience of notation, the left-hand side of the equation is simplified as $P\left(z^{p\to q}=z_t^{p\to q}|\mathbf{x}_t^p,\mathbf{x}_t^q\right)$ where $z_t^{p\to q}=\{z_t^{r,p\to q}, z_t^{c,p\to q}=PC\}$.

On the contrary, if a node does not receive a packet from another node, an NC event occurs, i.e., $z_t^{c,p\to q}=NC$, and no RSSI measurement is available. The likelihood function in this situation is written as:

$$P\left(z^{p\to q} = z_t^{p\to q} | \mathbf{x}_t^p, \mathbf{x}_t^q\right) = \underbrace{P\left(z_t^{c,p\to q} = NC | \mathbf{x}_t^p, \mathbf{x}_t^q\right)}_{likelihood\ given\ connectivity}$$

where $z_t^{p\to q} = \{z_t^{c,p\to q} = NC\}$.

C.5 Summary

The tracking approaches proposed in this thesis are capable of tracking vehicles with the availability of rough pairwise measurements. It should be emphasised that vehicle proximity detection approach does not pursue accurate ranging between vehicles, and between vehicle and infrastructure. As a rule of thumb, the more accurate an observation/sensor model, the better tracking accuracy. However, developing an accurate observation model is time consuming, and not worthwhile either for an observing system that deals with unreliable wireless connections and extremely noisy RSSI measurements. Nevertheless, a trivial way to improve the existing vehicle proximity detection approach is to incorporate more communication data in operation. It is possible to introduce an on-learning mechanism for the continuous refining of the models.

Appendix D

Background of Parametric Cooperative Tracking

D.1 Means and Covariance

In Kalman filtering and its variants, the joint state of all N_n nodes at time slot k are denoted by means of estimates together with covariance matrix.

$$\hat{\mathbf{X}}_k = \begin{bmatrix} \hat{\mathbf{x}}_k^1 & \hat{\mathbf{x}}_k^2 & \cdots & \hat{\mathbf{x}}_k^n \end{bmatrix}^T$$

$$\mathbf{P}_k = \begin{bmatrix} \mathbf{P}_k^1 & \mathbf{P}_k^{12} & \cdots & \mathbf{P}_k^{1n} \\ \mathbf{P}_k^{21} & \mathbf{P}_k^2 & \cdots & \mathbf{P}_k^{2n} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{P}_k^{n1} & \mathbf{P}_k^{n2} & \cdots & \mathbf{P}_k^n \end{bmatrix}$$

D.2 State Transition

- Linear State Transition Model: Given a linear state transition model:

$$\mathbf{X}_k = \mathbf{F}\mathbf{X}_{k-1} + \mathbf{B}\mathbf{u}_k + \mathbf{G}\mathbf{w} \tag{D.1}$$

where $P(\mathbf{w}) \sim N(0, \mathbf{Q})$.

The Kalman filter predicts joint state of nodes at time k with

$$\hat{\mathbf{X}}_k^- = \mathbf{F}\hat{\mathbf{X}}_{k-1} + \mathbf{B}\mathbf{u}_k$$

$$\mathbf{P}_k^- = \mathbf{F}\mathbf{P}_{k-1}\mathbf{F}^T + \mathbf{G}\mathbf{Q}\mathbf{G}^T$$

- Non-Linear State Transition Model: When the state transition model presents non-linearity, which means:

$$\mathbf{X}_k = \mathbf{f}(\mathbf{X}_{k-1}, \mathbf{u}_k) + \mathbf{g}(\mathbf{X}_{k-1})\mathbf{w} \tag{D.2}$$

then the extended Kalman filter (EKF) is adopted.

Applying a 1^{st} order Taylor series expansion about a linearisation point $\mathbf{X}_{k-1}^*$ for Equation (D.2), we get:

$$\begin{aligned}\mathbf{X}_k^* + \boldsymbol{\delta}\mathbf{X}_k &= \mathbf{f}\left(\mathbf{X}_{k-1}^* + \boldsymbol{\delta}\mathbf{X}_{k-1}, \mathbf{u}_k\right) + \mathbf{g}\left(\mathbf{X}_{k-1}^* + \boldsymbol{\delta}\mathbf{X}_{k-1}\right)\mathbf{w} \\ &\approx \mathbf{f}\left(\mathbf{X}_{k-1}^*, \mathbf{u}_k\right) + \mathbf{g}\left(\mathbf{X}_{k-1}^*\right)\mathbf{w} + \nabla\mathbf{F}_{k-1}\boldsymbol{\delta}\mathbf{X}_{k-1} + \nabla\mathbf{G}_{k-1}\mathbf{w}\end{aligned}$$

where $\nabla\mathbf{F}_{k-1}$ and $\nabla\mathbf{G}_{k-1}$ are Jacobian matrices of $\mathbf{f}(\cdot)$ and $\mathbf{g}(\cdot)$ about the point $\mathbf{X}_{k-1}^*$ respectively.

As $\mathbf{X}_k^* = \mathbf{f}\left(\mathbf{X}_{k-1}^*, \mathbf{u}_k\right) + \mathbf{g}\left(\mathbf{X}_{k-1}^*\right)\mathbf{w}$, the transition model is then linearised to:

$$\boldsymbol{\delta}\mathbf{X}_k \approx \nabla\mathbf{F}_{k-1}\boldsymbol{\delta}\mathbf{X}_{k-1} + \nabla\mathbf{G}_{k-1}\mathbf{w}$$

The Taylor series expansion is only accurate when the linearisation point $\mathbf{X}_{k-1}^*$ is equal to true state $\mathbf{X}_{k-1}$. Under this situation the EKF is ideal, which cannot

be reached in real world. Normally the last state estimate $\hat{\mathbf{X}}_{k-1}$ is chosen to be the linearisation point by assuming that $\hat{\mathbf{X}}_{k-1}$ is close to $\mathbf{X}_{k-1}$.

Respectively the Jacobian matrices are obtained by:

$$\nabla \mathbf{F}_{k-1} = \frac{\partial \mathbf{f}\left(\mathbf{X}_{k-1}, \mathbf{u}_k\right)}{\partial \mathbf{X}_{k-1}} \Big|_{\mathbf{X}_{k-1}=\hat{\mathbf{X}}_{k-1}}$$

$$\nabla \mathbf{G}_{k-1} = \frac{\partial \mathbf{g}\left(\mathbf{X}_{k-1}\right)}{\partial \mathbf{X}_{k-1}} \Big|_{\mathbf{X}_{k-1}=\hat{\mathbf{X}}_{k-1}}$$

More specifically, the $\nabla \mathbf{F}_{\hat{\mathbf{X}}_{k-1}}$ is factorised to:

$$\nabla \mathbf{F}_{k-1} = \begin{bmatrix} \nabla \mathbf{F}_{\hat{\mathbf{x}}_{k-1}^1} & \cdots & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & \cdots & \nabla \mathbf{F}_{\hat{\mathbf{x}}_{k-1}^p} & \cdots & 0 \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & \cdots & \nabla \mathbf{F}_{\hat{\mathbf{x}}_{k-1}^{Nn}} \end{bmatrix}$$

where $\nabla \mathbf{F}_{\hat{\mathbf{x}}_{k-1}^p} = \frac{\partial \mathbf{f}(\cdot)}{\partial \mathbf{x}_{k-1}^p} \Big|_{\mathbf{x}_{k-1}^p=\hat{\mathbf{x}}_{k-1}^p}$, while the $\nabla \mathbf{G}_{k-1}$ is of the similar form.

To sum up, we have predictions:

$$\hat{\mathbf{X}}_k^- = \mathbf{f}\left(\hat{\mathbf{X}}_{k-1}, \mathbf{u}_k\right)$$

$$\mathbf{P}_k^- = \nabla \mathbf{F}_{k-1} \mathbf{P}_{k-1} \nabla \mathbf{F}_{k-1}^T + \nabla \mathbf{G}_{k-1} \mathbf{Q} \nabla \mathbf{G}_{k-1}^T$$

D.3 Observation Models

- Linear Absolute Observation Model: Suppose an egocentric measurement is generated for node p at time k according to the absolute observation model:

$$\mathbf{z}_k^p = \mathbf{H}_k^p \mathbf{X}_k + \mathbf{v}_{abs} \tag{D.3}$$

where $P(\mathbf{v}_{abs}) \sim N(0, \mathbf{R}_{abs})$.

- Linear Relative Observation Model: Given a linear relative measurement model for nodes p and q at time k:

$$\mathbf{z}_k^{p\to q} = \mathbf{H}_k^{p\to q}\mathbf{X}_k + \mathbf{v}_{rel} \tag{D.4}$$

where $P(\mathbf{v}_{abs}) \sim N(0, \mathbf{R}_{rel})$. And the observation function $\mathbf{H}_k^{p\to q}$ only depends on the two related states $\mathbf{x}_k^p$ and $\mathbf{x}_k^q$.

- Non-Linear Relative Observation Model: The non-linear relative observation model for nodes p and q at time step k is denoted by:

$$\mathbf{z}_k^{p\to q} = \mathbf{h}_k^{p\to q}(\mathbf{X}_k) + \mathbf{v}_{rel}$$

where $P(\mathbf{v}_{rel}) \sim N(0, \mathbf{R}_{rel})$. The observation function $\mathbf{h}_k^{p\to q}(\cdot)$ is only related to $\mathbf{x}_k^p$ and $\mathbf{x}_k^q$.

Then the observation model is linearised about point $\hat{\mathbf{X}}_k^-$ by performing 1^{st} order Taylor series expansion on $\mathbf{z}$ and $\mathbf{X}$.

$$\boldsymbol{\delta}\mathbf{z}_k^{p\to q} \approx \nabla\mathbf{H}_k^{p\to q}\boldsymbol{\delta}\mathbf{X}_k + \mathbf{v}_{rel}$$

where $\boldsymbol{\delta}\mathbf{z}_k^{p\to q} = \mathbf{z}_k^{p\to q} - \mathbf{h}_k^{p\to q}\left(\hat{\mathbf{X}}_k^-\right)$ and $\boldsymbol{\delta}\mathbf{X}_k = \mathbf{X}_k - \hat{\mathbf{X}}_k^-$ assuming that $\hat{\mathbf{X}}_k^-$ is close to $\mathbf{X}_k$. The EKF is ideal when the linearisation point is chosen to be true state $\mathbf{X}_k$ instead of the prediction $\hat{\mathbf{X}}_k^-$. The $\nabla\mathbf{H}_k^{p\to q}$ is the Jacobian matrix of the non-linear observation model $\mathbf{h}_k^{p\to q}(\cdot)$.

The Jacobian matrix can be derived by:

$$\nabla\mathbf{H}_k^{p\to q} = \frac{\partial\mathbf{h}_k^{p\to q}(\mathbf{X}_k)}{\partial\mathbf{X}_k}\Big|_{\mathbf{X}_k=\hat{\mathbf{X}}_k}$$

And $\nabla\mathbf{H}_k^{p\to q}$ is factorised to:

$$\nabla\mathbf{H}_k^{p\to q} = \begin{bmatrix} 0 & \cdots & \nabla\mathbf{H}_{\hat{\mathbf{x}}_k^p}^{p\to q} & \cdots & \nabla\mathbf{H}_{\hat{\mathbf{x}}_k^q}^{p\to q} & \cdots & 0 \end{bmatrix}$$

where

$\nabla \mathbf{H}^{p \to q}_{\hat{\mathbf{x}}^p_k} = \frac{\partial \mathbf{h}^{p \to q}_k(\cdot)}{\partial \mathbf{x}^p_k} \Big|_{\hat{\mathbf{x}}^p_k, \hat{\mathbf{x}}^q_k}$

and

$\nabla \mathbf{H}^{p \to q}_{\hat{\mathbf{x}}^q_k} = \frac{\partial \mathbf{h}^{p \to q}_k(\cdot)}{\partial \mathbf{x}^q_k} \Big|_{\hat{\mathbf{x}}^p_k, \hat{\mathbf{x}}^q_k}$.

D.4 Batch Update

Given a collection of arbitrary observations (for example, m absolute observations and n non-linear relative observations at time k) received by the fusion centre:

$$\mathbf{\Phi}_k = \begin{bmatrix} \mathbf{z}^{p_1}_k & \cdots & \mathbf{z}^{p_m}_k & \mathbf{z}^{p_1 \to q_1}_k & \cdots & \mathbf{z}^{p_n \to q_n}_k \end{bmatrix}^T \tag{D.5}$$

The Kalman filter updates estimates by:

$$\hat{\mathbf{X}}^+_k = \hat{\mathbf{X}}^-_k + \mathbf{K}_k \widetilde{\mathbf{y}}_k$$

$$\mathbf{P}^+_k = \mathbf{P}^-_k - \mathbf{K}_k \mathbf{S}_k \mathbf{K}^T_k$$

where the measurement residual $\widetilde{\mathbf{y}}_k = \mathbf{\Phi}_k - \mathbf{\Gamma}_k$ and

$$\mathbf{\Gamma}_k = \begin{bmatrix} \begin{bmatrix} \mathbf{H}^{p_1}_k \hat{\mathbf{X}}^-_k & \cdots & \mathbf{H}^{p_m}_k \hat{\mathbf{X}}^-_k \end{bmatrix}^T \\ \begin{bmatrix} \mathbf{h}^{p_1 \to q_1}_k \left(\hat{\mathbf{X}}^-_k \right) & \cdots & \mathbf{h}^{p_n \to q_n}_k \left(\hat{\mathbf{X}}^-_k \right) \end{bmatrix}^T \end{bmatrix}.$$

the residual covariance $\mathbf{S}_k = \mathbf{W}_k \mathbf{P}^-_k \mathbf{W}^T_k + \mathbf{\Upsilon}_k$.

the optimal Kalman gain $\mathbf{K}_k = \mathbf{P}^-_k \mathbf{W}^T_k \mathbf{S}^{-1}_k$.

the measurement matrix

$$\mathbf{W}_k = \begin{bmatrix} \mathbf{H}^{p_1}_k & \cdots & \mathbf{H}^{p_m}_k & \nabla \mathbf{H}^{p_1 \to q_1}_k & \cdots & \nabla \mathbf{H}^{p_n \to q_n}_k \end{bmatrix}^T$$

the measurement covariance matrix

$$\boldsymbol{\Upsilon}_k = \begin{bmatrix} \mathbf{R}_{abs} & \cdots & 0 & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & \mathbf{R}_{abs} & 0 & \cdots & 0 \\ 0 & \cdots & 0 & \mathbf{R}_{rel} & \cdots & 0 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & 0 & \cdots & \mathbf{R}_{rel} \end{bmatrix}$$

Appendix E

Background of Delayed-State Filtering

E.1 Delayed-State Kalman Filter

Delayed-state Kalman filter keeps a record of joint states within sliding time window.

$$\hat{\mathbf{X}}_{k-t_w+1:k} = \begin{bmatrix} \hat{\mathbf{X}}_k & \hat{\mathbf{X}}_{k-1} & \cdots & \hat{\mathbf{X}}_{k-t_w+1} \end{bmatrix}^T$$

$$\mathbf{P}_{k-t_w+1:k} = \begin{bmatrix} \mathbf{P}_k & \mathbf{P}_{k,k-1} & \cdots & \mathbf{P}_{k,k-t_w+1} \\ \mathbf{P}_{k-1,k} & \mathbf{P}_{k-1} & \cdots & \mathbf{P}_{k-1,k-t_w+1} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{P}_{k-t_w+1,k} & \mathbf{P}_{k-t_w+1,k-1} & \cdots & \mathbf{P}_{k-t_w+1} \end{bmatrix}$$

E.1.1 State Transition

The joint state of all N_n nodes at time slot k are denoted by means of estimates together with covariance matrix in Kalman filtering.

$$\hat{\mathbf{X}}_k = \begin{bmatrix} \hat{\mathbf{x}}_k^1 & \hat{\mathbf{x}}_k^2 & \cdots & \hat{\mathbf{x}}_k^n \end{bmatrix}^T$$

$$\mathbf{P}_k = \begin{bmatrix} \mathbf{P}_k^{11} & \mathbf{P}_k^{12} & \cdots & \mathbf{P}_k^{1n} \\ \mathbf{P}_k^{21} & \mathbf{P}_k^{22} & \cdots & \mathbf{P}_k^{2n} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{P}_k^{n1} & \mathbf{P}_k^{n2} & \cdots & \mathbf{P}_k^{nn} \end{bmatrix}$$

Suppose a linear state transition model:

$$\mathbf{X}_k = \mathbf{F}\mathbf{X}_{k-1} + \mathbf{B}\mathbf{u}_k + \mathbf{G}\mathbf{w} \tag{E.1}$$

where $P(\mathbf{w}) \sim N(0, \mathbf{Q})$.

For $k \geq t_w$, the delayed-state Kalman filter predicts joint states of nodes with:

$$\hat{\mathbf{X}}_{k-t_w+1:k}^{-} = \mathbf{T}\hat{\mathbf{X}}_{k-t_w:k-1} + \mathbf{D}\mathbf{u}_k$$

$$\mathbf{P}_{k-t_w+1:k}^{-} = \mathbf{T}\mathbf{P}_{k-t_w:k-1}\mathbf{T}^T + \mathbf{E}\mathbf{Q}\mathbf{E}^T$$

where

$$\mathbf{T} = \begin{bmatrix} \mathbf{F} & 0 & \cdots & 0 & 0 \\ \mathbf{I} & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & \cdots & \mathbf{I} & 0 \end{bmatrix}$$

$$\mathbf{D} = \begin{bmatrix} \mathbf{B} & 0 & \cdots & 0 \end{bmatrix}^T$$

$$\mathbf{E} = \begin{bmatrix} \mathbf{G} & 0 & \cdots & 0 \end{bmatrix}^T$$

For $k < t_w$, the states in delayed-state Kalman filter are evolved with:

$$\hat{\mathbf{X}}_{0:k}^{-} = \mathbf{T}'\hat{\mathbf{X}}_{0:k-1} + \mathbf{D}\mathbf{u}_k$$

$$\mathbf{P}_{0:k}^{-} = \mathbf{T}'\mathbf{P}_{0:k-1}\mathbf{T}'^{T} + \mathbf{E}\mathbf{Q}\mathbf{E}^{T}$$

where

$$\mathbf{T}' = \begin{bmatrix} \mathbf{F} & 0 & \cdots & 0 \\ \mathbf{I} & 0 & \cdots & 0 \\ 0 & \mathbf{I} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \mathbf{I} \end{bmatrix}$$

E.1.2 Update with Delayed Observations

- Linear Absolute Observation Model: Suppose an egocentric measurement is generated for node p at time t according to the absolute observation model:

$$\mathbf{z}_t^p = \mathbf{H}_t^p \mathbf{X}_t + \mathbf{v}_{abs} \tag{E.2}$$

  where $P(\mathbf{v}_{abs}) \sim N(0, \mathbf{R}_{abs})$.

  In delayed-state Kalman filter, a delayed absolute observations received by fusion centre at time k are represented by:

$$\mathbf{z}_t^p = \mathbf{A}_t^p \mathbf{X}_{k-t_w+1:k} + \mathbf{v}_{abs}$$

  where $\mathbf{A}_t^p = \begin{bmatrix} 0 & \cdots & \mathbf{H}_t^p & \cdots & 0 \end{bmatrix}$.

- Non-Linear Relative Observation Model: The non-linear relative observation model for nodes p and q at time step t $(k - t_w + 1 \leq t \leq k)$ is denoted by:

$$\mathbf{z}_t^{p\to q} = \mathbf{h}_t^{p\to q}\left(\mathbf{X}_{k-t_w+1:k}\right) + \mathbf{v}_{rel}$$

  where $P(\mathbf{v}_{rel}) \sim N(0, \mathbf{R}_{rel})$. The observation function $\mathbf{h}_t^{p\to q}(\cdot)$ is only related to $\mathbf{x}_t^p$ and $\mathbf{x}_t^q$.

So we have Jacobian matrix of $\mathbf{h}_t^{p\to q}(\cdot)$ evaluated at $\hat{\mathbf{X}}_{k-t_w+1:k}$ and written as:

$$\begin{aligned}\mathbf{J}_t^{p\to q} &= \frac{\partial \mathbf{h}_t^{p\to q}(\mathbf{X}_{k-t_w+1:k})}{\partial \mathbf{X}_{k-t_w+1:k}}\Big|_{\mathbf{x}_{k-t_w+1:k}=\hat{\mathbf{x}}_{k-t_w+1:k}} \\ &= \begin{bmatrix} 0 & \cdots & \nabla\mathbf{H}_t^{p\to q} & \cdots & 0 \end{bmatrix}\end{aligned}$$

And $\nabla\mathbf{H}_t^{p\to q}$ is further factorised to:

$$\nabla\mathbf{H}_t^{p\to q} = \begin{bmatrix} 0 & \cdots & \nabla\mathbf{H}_{\hat{\mathbf{x}}_t^p}^{p\to q} & \cdots & \nabla\mathbf{H}_{\hat{\mathbf{x}}_t^q}^{p\to q} & \cdots & 0 \end{bmatrix}$$

where

$$\nabla\mathbf{h}_{\hat{\mathbf{x}}_t^p}^{p\to q} = \frac{\partial \mathbf{h}_t^{p\to q}(\cdot)}{\partial \mathbf{x}_t^p}\Big|_{\hat{\mathbf{x}}_t^p,\hat{\mathbf{x}}_t^q}$$

$$\nabla\mathbf{h}_{\hat{\mathbf{x}}_t^q}^{p\to q} = \frac{\partial \mathbf{h}_t^{p\to q}(\cdot)}{\partial \mathbf{x}_t^q}\Big|_{\hat{\mathbf{x}}_t^p,\hat{\mathbf{x}}_t^q}$$

Then the observation model is linearised by performing 1^{st} order Taylor series expansion on $\mathbf{z}$ and $\mathbf{X}$.

$$\boldsymbol{\delta}\mathbf{z}_t^{p\to q} \approx \mathbf{J}_t^{p\to q}\boldsymbol{\delta}\mathbf{X}_{k-t_w+1:k} + \mathbf{v}_{rel}$$

where

$\boldsymbol{\delta}\mathbf{z}_t^{p\to q} = \mathbf{z}_t^{p\to q} - \mathbf{h}_t^{p\to q}\left(\hat{\mathbf{X}}_{k-t_w+1:k}^{-}\right)$,

$\boldsymbol{\delta}\mathbf{X}_{k-t_w+1:k} = \mathbf{X}_{k-t_w+1:k} - \hat{\mathbf{X}}_{k-t_w+1:k}^{-}$ assuming that $\hat{\mathbf{X}}_{k-t_w+1:k}^{-}$ is close to $\mathbf{X}_{k-t_w+1:k}$.

- Batch Update: Given a collection of arbitrary observations (say, m absolute observations and n relative observations within the sliding time window) received by the fusion centre at time k:

$$\boldsymbol{\Phi}_k = \begin{bmatrix} \mathbf{z}_{t_1}^{p_1} & \cdots & \mathbf{z}_{t_m}^{p_m} & \mathbf{z}_{t_1}^{p_1\to q_1} & \cdots & \mathbf{z}_{t_n}^{p_n\to q_n} \end{bmatrix}^T \tag{E.3}$$

The delayed-state Kalman filter updates estimates by:

$$\hat{\mathbf{X}}^{+}_{k-t_w+1:k} = \hat{\mathbf{X}}^{-}_{k-t_w+1:k} + \mathbf{K}_k \widetilde{\mathbf{y}}_k$$

$$\mathbf{P}^{+}_{k-t_w+1:k} = \mathbf{P}^{-}_{k-t_w+1:k} - \mathbf{K}_k \mathbf{S}_k \mathbf{K}_k^T$$

where

the measurement residual $\widetilde{\mathbf{y}}_k = \mathbf{\Phi}_k - \mathbf{\Gamma}_k$

and

$$\mathbf{\Gamma}_k = \begin{bmatrix} \left[\mathbf{A}^{p_1}_{t_1} \hat{\mathbf{X}}^{-}_{k-t_w+1:k} \quad \cdots \quad \mathbf{A}^{p_m}_{t_m} \hat{\mathbf{X}}^{-}_{k-t_w+1:k}\right]^T \\ \left[\mathbf{h}^{p_1 \to q_1}_{t_1}\left(\hat{\mathbf{X}}^{-}_{k-t_w+1:k}\right) \quad \cdots \quad \mathbf{h}^{p_n \to q_n}_{t_n}\left(\hat{\mathbf{X}}^{-}_{k-t_w+1:k}\right)\right]^T \end{bmatrix}$$

the residual covariance $\mathbf{S}_k = \mathbf{W}_k \mathbf{P}^{-}_{k-t_w+1:k} \mathbf{W}_k^T + \mathbf{\Upsilon}_k$.

the optimal Kalman gain $\mathbf{K}_k = \mathbf{P}^{-}_{k-t_w+1:k} \mathbf{W}_k^T \mathbf{S}_k^{-1}$.

the measurement matrix

$$\mathbf{W}_k = \begin{bmatrix} \mathbf{A}^{p_1}_{t_1} & \cdots & \mathbf{A}^{p_m}_{t_m} & \mathbf{J}^{p_1 \to q_1}_{t_1} & \cdots & \mathbf{J}^{p_n \to q_n}_{t_n} \end{bmatrix}^T$$

the measurement covariance matrix

$$\mathbf{\Upsilon}_k = \begin{bmatrix} \mathbf{R}_{abs} & \cdots & 0 & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & \mathbf{R}_{abs} & 0 & \cdots & 0 \\ 0 & \cdots & 0 & \mathbf{R}_{rel} & \cdots & 0 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & 0 & \cdots & \mathbf{R}_{rel} \end{bmatrix}$$

E.2 Delayed-State Information Filter

E.2.1 Converting States to Information Form

The joint state of all nodes at time k can be converted to information form:

$$\mathbf{Y}_k = \mathbf{P}_k^{-1}$$

$$\hat{\mathbf{y}}_k = \mathbf{P}_k^{-1}\hat{\mathbf{X}}_k$$

Delayed-states information filter keeps a record of historical joint states up to time slot k:

$$\hat{\mathbf{y}}_{k-t_w+1:k} = \begin{bmatrix}\hat{\mathbf{y}}_k & \hat{\mathbf{y}}_{k-1} & \hat{\mathbf{y}}_{k-2} & \cdots & \hat{\mathbf{y}}_{k-t_w+1}\end{bmatrix}^T \tag{E.4}$$

$$\mathbf{Y}_{k-t_w+1:k} = \begin{bmatrix} \mathbf{Y}_k & \mathbf{Y}_{k,k-1} & 0 & \cdots & 0 \\ \mathbf{Y}_{k-1,k} & \mathbf{Y}_{k-1} & \mathbf{Y}_{k-1,k-2} & \cdots & 0 \\ 0 & \mathbf{Y}_{k-2,k-1} & \mathbf{Y}_{k-2} & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & \mathbf{Y}_{k-t_w+1} \end{bmatrix} \tag{E.5}$$

The delayed-state information matrix here is a sparse matrix.

E.2.2 States Evolving

- Augmentation of a New State: Given the motion model in Equation (E.1), augmentation process of new time step k is described by:

$$\hat{\mathbf{y}}_{k-t_w:k} = \begin{bmatrix} \mathbf{V}^{-1}\mathbf{B}\mathbf{u}_k \\ \hat{\mathbf{y}}_{k-1} - \mathbf{F}^T\mathbf{V}^{-1}\mathbf{B}\mathbf{u}_k \\ \vdots \\ \hat{\mathbf{y}}_{k-t_w} \end{bmatrix}$$

$$\mathbf{Y}_{k-t_w:k} = \begin{bmatrix} \mathbf{V}^{-1} & -\mathbf{V}^{-1}\mathbf{F} & \cdots & 0 \\ -\mathbf{F}^T\mathbf{V}^{-1} & \mathbf{Y}_{k-1} + \mathbf{F}^T\mathbf{V}^{-1}\mathbf{F} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \mathbf{Y}_{k-t_w} \end{bmatrix}$$

where $\mathbf{V} = \mathbf{G}\mathbf{Q}\mathbf{G}^T$ must be non-singular.

- Marginalisation of the Tail State: Then the oldest state $\mathbf{x}_{k-t_w}$ is marginalised away to keep a constant window size. The marginalisation process here is trivial, as the only affected state is $\mathbf{x}_{k-t_w+1}$.

$$\begin{aligned} \hat{\mathbf{y}}^{mar}_{k-t_w+1:k} &= \hat{\mathbf{y}}_{k-t_w+1:k} - \mathbf{Y}_{k-t_w+1:k,k-t_w+1}\mathbf{Y}^{-1}_{k-t_w}\hat{\mathbf{y}}_{k-t_w} \\ &= \begin{bmatrix} \hat{\mathbf{y}}_k \\ \hat{\mathbf{y}}_{k-1} \\ \hat{\mathbf{y}}_{k-2} \\ \vdots \\ \hat{\mathbf{y}}_{k-t_w+1} \end{bmatrix} - \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ \mathbf{Y}_{k-t_w+1,k-t_w} \end{bmatrix} \mathbf{Y}^{-1}_{k-t_w}\hat{\mathbf{y}}_{k-t_w} \\ &= \begin{bmatrix} \hat{\mathbf{y}}_k \\ \hat{\mathbf{y}}_{k-1} \\ \hat{\mathbf{y}}_{k-2} \\ \vdots \\ \hat{\mathbf{y}}_{k-t_w+1} - \mathbf{Y}_{k-t_w+1,k-t_w}\mathbf{Y}^{-1}_{k-t_w}\hat{\mathbf{y}}_{k-t_w} \end{bmatrix} \end{aligned}$$

$$
\begin{aligned}
\mathbf{Y}^{mar}_{k-t_w+1:k} &= \mathbf{Y}_{k-t_w+1:k} - \mathbf{Y}_{k-t_w+1:k,k-t_w+1}\mathbf{Y}^{-1}_{k-t_w}\mathbf{Y}_{k-t_w+1,k-t_w+1:k} \\
&= \begin{bmatrix}
\mathbf{Y}_k & \mathbf{Y}_{k,k-1} & 0 & \cdots & 0 \\
\mathbf{Y}_{k-1,k} & \mathbf{Y}_{k-2} & \mathbf{Y}_{k-2,k-1} & \cdots & 0 \\
0 & \mathbf{Y}_{k-1,k-2} & \mathbf{Y}_{k-3} & \cdots & 0 \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
0 & 0 & 0 & \cdots & \mathbf{Y}_{k-t_w+1}
\end{bmatrix} \\
&- \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ \mathbf{Y}_{k-t_w+1,k-t_w} \end{bmatrix} \mathbf{Y}^{-1}_{k-t_w} \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ \mathbf{Y}_{k-t_w,k-t_w+1} \end{bmatrix}^T \\
&= \begin{bmatrix}
\mathbf{Y}_k & \mathbf{Y}_{k,k-1} & 0 & \cdots & 0 \\
\mathbf{Y}_{k-1,k} & \mathbf{Y}_{k-1} & \mathbf{Y}_{k-1,k-2} & \cdots & 0 \\
0 & \mathbf{Y}_{k-2,k-1} & \mathbf{Y}_{k-2} & \cdots & 0 \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
0 & 0 & 0 & \cdots & \mathbf{Y}'_{k-t_w+1}
\end{bmatrix}
\end{aligned}
$$

where $\mathbf{Y}'_{k-t_w+1} = \mathbf{Y}_{k-t_w+1} - \mathbf{Y}_{k-t_w+1,k-t_w}\mathbf{Y}^{-1}_{k-t_w}\mathbf{Y}_{k-t_w,k-t_w+1}$.

The above formulation is based on the information vector and matrix marginalisation lemma:

Given an information matrix:

$$
X = \begin{bmatrix} A & B \\ C & D \end{bmatrix}
$$

Then:

$$
A^{mar} = A - BD^{-1}C
$$

E.2.3 Data Fusion

- Linear Absolute Observation Model: Given an absolute observation $\mathbf{z}_t^p$ where $k - t_w + 1 \leq t \leq k$ for node p at time t, according to observation model in Equation (E.2), the information contribution is:

$$\mathbf{i}_t^p = (\mathbf{A}_t^p)^T \mathbf{R}_{abs}^{-1} \mathbf{z}_t^p$$

and

$$\mathbf{I}_t^p = (\mathbf{A}_t^p)^T \mathbf{R}_{abs}^{-1} \mathbf{A}_t^p$$

- Non-linear Relative Observation Model: Given a relative observation $\mathbf{z}_t^{p \to q}$ where $k - t_w + 1 \leq t \leq k$, the information contribution is:

$$\mathbf{i}_t^{p \to q} = (\mathbf{J}_t^{p \to q})^T \mathbf{R}_{rel}^{-1} \left(\mathbf{z}_t^{p \to q} - \mathbf{h}_t^{p \to q} \left(\hat{\mathbf{X}}_{k-t_w+1:k}^- \right) + \mathbf{J}_t^{p \to q} \hat{\mathbf{X}}_{k-t_w+1:k}^- \right)$$

$$\mathbf{I}_t^{p \to q} = (\mathbf{J}_t^{p \to q})^T \mathbf{R}_{rel}^{-1} \mathbf{J}_t^{p \to q}$$

- Batch Update: To fuse the absolute and relative delayed observations in $\mathbf{\Phi}_k$ given in Equation (E.3), the update stage is written as:

$$\hat{\mathbf{y}}_{k-t_w+1:k}^+ = \hat{\mathbf{y}}_{k-t_w+1:k}^- + \sum_{i=1}^{m} \mathbf{i}_{t_i}^{p_i} + \sum_{i=1}^{n} \mathbf{i}_{t_i}^{p_i \to q_i}$$

$$\mathbf{Y}_{k-t_w+1:k}^+ = \mathbf{Y}_{k-t_w+1:k}^- + \sum_{i=1}^{m} \mathbf{I}_{t_i}^{p_i} + \sum_{i=1}^{n} \mathbf{I}_{t_i}^{p_i \to q_i}$$

The collected information pieces are fused into the estimator in an additive manner.

E.2.4 Extraction of the Last State Estimates

After fusion step at each time step, the last state estimates could be extracted from the joint states. When non-linear observation models are used, the extraction process is necessary in order to obtain the last state predictions before the data fusion stage.

$$
\begin{aligned}
\hat{\mathbf{y}}_k^{mar} &= \hat{\mathbf{y}}_k - \mathbf{Y}_{k,k-t_w+1:k-1}\mathbf{Y}_{k-t_w+1:k-1}^{-1}\hat{\mathbf{y}}_{k-t_w+1:k-1} \\
&= \hat{\mathbf{y}}_k - \begin{bmatrix} \mathbf{Y}_{k,k-1} \\ 0 \\ \vdots \\ 0 \end{bmatrix}^T \mathbf{Y}_{k-t_w+1:k-1}^{-1} \begin{bmatrix} \hat{\mathbf{y}}_{k-1} \\ \hat{\mathbf{y}}_{k-2} \\ \vdots \\ \hat{\mathbf{y}}_{k-t_w+1} \end{bmatrix} \\
&= \hat{\mathbf{y}}_k - \begin{bmatrix} \mathbf{Y}_{k,k-1} \\ 0 \\ \vdots \\ 0 \end{bmatrix}^T \begin{bmatrix} \mathbf{Y}_{k-1} & \mathbf{Y}_{k-1,k-2} & \cdots & 0 \\ \mathbf{Y}_{k-2,k-1} & \mathbf{Y}_{k-2} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \mathbf{Y}_{k-t_w+1} \end{bmatrix}^{-1} \begin{bmatrix} \hat{\mathbf{y}}_{k-1} \\ \hat{\mathbf{y}}_{k-2} \\ \vdots \\ \hat{\mathbf{y}}_{k-t_w+1} \end{bmatrix}
\end{aligned}
$$

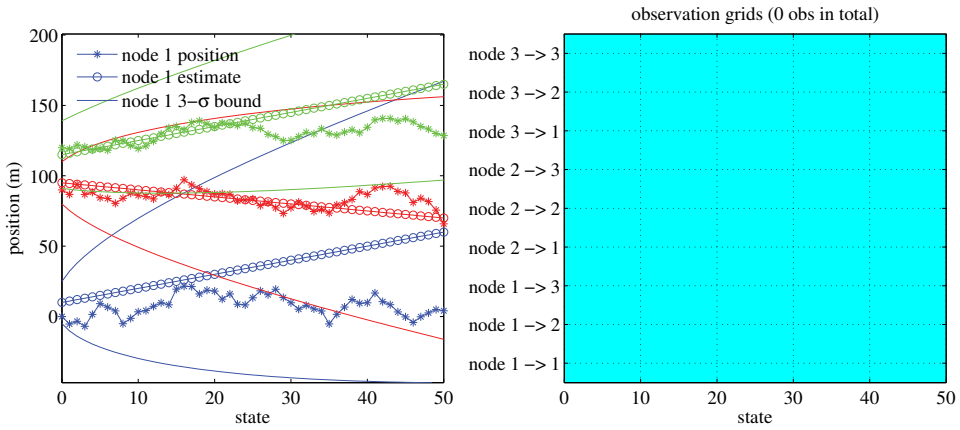
$$
\begin{aligned}
\mathbf{Y}_k^{mar} &= \mathbf{Y}_k - \mathbf{Y}_{k,k-t_w+1:k-1}\mathbf{Y}_{k-t_w+1:k-1}^{-1}\mathbf{Y}_{k-t_w+1:k-1,k} \\
&= \mathbf{Y}_k - \begin{bmatrix} \mathbf{Y}_{k,k-1} \\ 0 \\ \vdots \\ 0 \end{bmatrix}^T \mathbf{Y}_{k-t_w+1:k-1}^{-1} \begin{bmatrix} \mathbf{Y}_{k-1,k} \\ 0 \\ \vdots \\ 0 \end{bmatrix} \\
&= \mathbf{Y}_k - \begin{bmatrix} \mathbf{Y}_{k,k-1} \\ 0 \\ \vdots \\ 0 \end{bmatrix}^T \begin{bmatrix} \mathbf{Y}_{k-1} & \mathbf{Y}_{k-1,k-2} & \cdots & 0 \\ \mathbf{Y}_{k-2,k-1} & \mathbf{Y}_{k-2} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \mathbf{Y}_{k-t_w+1} \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{Y}_{k-1,k} \\ 0 \\ \vdots \\ 0 \end{bmatrix}
\end{aligned}
$$

This step is not trivial as the $\mathbf{Y}_{k-t_w+1:k-1}^{-1}$ is inverted, though $\mathbf{Y}_{k-t_w+1:k-1}$ is a sparse matrix and most elements in $\mathbf{Y}_{k,k-t_w+1:k-1}$ and $\mathbf{Y}_{k-t_w+1:k-1,k}$ are zero.

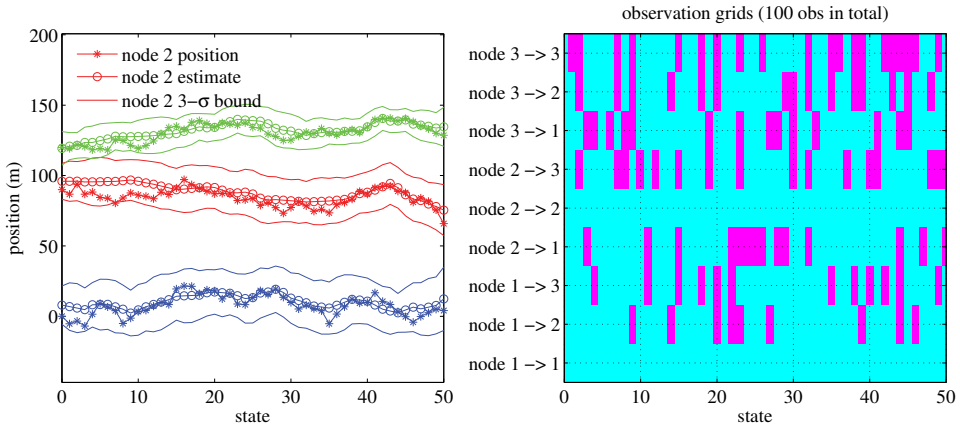
Inverse operation of Equations (E.4) and (E.5) is used to convert these extracted states back to forms of mean and covariance matrix.

E.2.5 Simulation Example

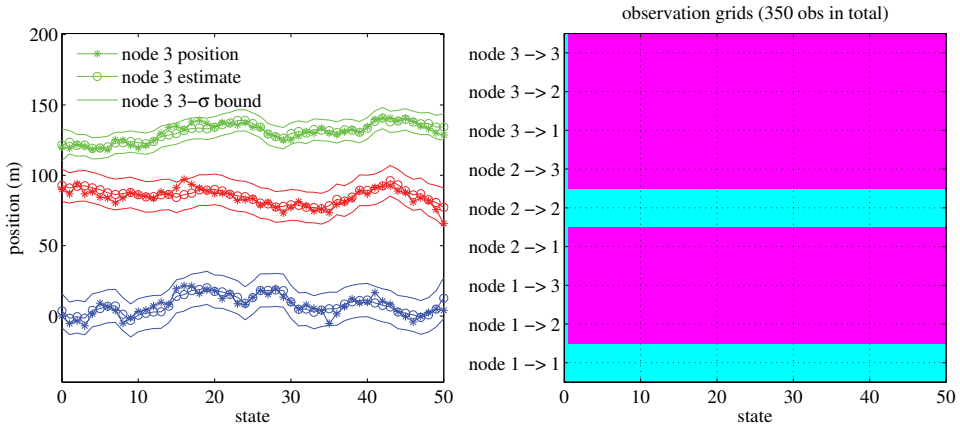
A simulation example is presented in Figure E.1, with three mobile nodes moving in a one-dimensional space. Only node #3 (denoted in green in the figure) is equipped with egocentric positioning means. This means the localisation/tracking of the rest two depends on the egocentric location information of node #3 and the relative position measurements between them. Furthermore, these observations are received by the fusion centre with random time delays. The figure reveals incrementally improved tracking results for all of these three nodes along with the arrival of the delayed observations, in the use of a delayed-state filter.



(a) Prediction only



(b) Updated with 100 delayed observations



(c) Updated with all delayed observations

Figure E.1 – Results of delayed-state cooperative tracking of three nodes using an information filter. This is an example showing a linear and Gaussian system with three nodes being tracked in a one-dimensional space. Though without self-localisation capability, nodes 1 and 2 are tracked accurately using the relative measurement between nodes.

Bibliography

[1] Ghassan M.T. Abdalla, Mosa Ali Abu-Rgheff, and Sidi Mohammed Senouci. Current trends in vehicular ad hoc networks. In *Proc. of IEEE Global Information Infrastructure Symposium*, Morocco, July 2007.

[2] S. Adabi, S. Jabbehdari, A. Rahmani, and S. Adabi. A novel distributed clustering algorithm for mobile ad-hoc networks. *Journal of Computer Science*, 4(2):161–166, 2008.

[3] Emrah Adamey and Umit Ozguner. Cooperative multitarget tracking and surveillance with mobile sensing agents: A decentralised approach. In *Proc. the 14th IEEE International Conference on Intelligent Transportation Systems (ITSC'11)*, Washington, DC, USA, October 2011.

[4] Gabriel Agamennoni, Juan Nieto, and Eduardo Mario Nebot. Mining GPS data for extracting significant places. In *Proc. IEEE International Conference on Robotics and Automation (ICRA'09)*, pages 855–862, 2009.

[5] Gabriel Agamennoni, Juan Nieto, and Eduardo Mario Nebot. Robust inference of principal road paths for intelligent transportation systems. *IEEE Trans. Intell. Transport. Syst.*, 12(1):298–308, January 2011.

[6] Gabriel Agamennoni, Juan I. Nieto, and Eduardo M. Nebot. An outlier-robust Kalman filter. In *Proc. IEEE International Conference on Robotics and Automation (ICRA'11)*, Shanghai, China, May 2011.

[7] Gabriel Agamennoni, Juan I. Nieto, and Eduardo M. Nebot. Estimation of multivehicle dynamics by considering contextual informaiton. *IEEE Trans. Robot.*, 28(4):855–870, August 2012.

[8] Craig S. Agate, Robert M. Wilkerson, and Keven J. Sullivan. Utilizing negative information to track ground vehicles through move-stop-move cycles. In *Proc. SPIE*, volume 5429, pages 273–283, 2004.

[9] Jon Agre and Loren Clare. An integrated architechture for cooperative sensing networks. *IEEE Computer Magazine*, pages 106–108, May 2000.

[10] I.F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci. Wireless sensor networks: a survey. *Journal of Computer Networks*, 38(4):393–422, March 2002.

[11] J. N. Al-Karaki and A. E. Kamal. Routing techniques in wireless sensor networks: a survey. *IEEE Wireless Communications*, 11(6):6–28, 2004.

[12] Nayef Alsindi and Kaveh Pahlavan. Cooperative localisation bounds for indoor ultra-wideband wirelss sensor networks. *EURASIP Journal on Advances in Signal Processing*, 2008, 2008.

[13] ASTM. ASTM E2213 - 03(2010). `http://enterprise.astm.org/filtrexx40.cgi?+REDLINE_PAGES/E2213.htm`. 2010. [Online; accessed 17-Aug-2013].

[14] P. Bahl and V. Padmanabhan. RADAR: An in-building RF-based user location and tracking system. In *Proc. of IEEE Infocom*, Tel-Aviv, Israel, March 2000.

[15] Alexander Bahr, Matthew R. Walter, and John J. Leonard. Consistent cooperative localisation. In *Proc. IEEE International Conference on Robotics and Automation (ICRA'09)*, Kobe, Japan, May 2009.

[16] Tim Bailey and Hugh Durrant-Whyte. Decentralised data fusion with delayed states for consistent inference in mobile ad hoc networks. Technical report, Australian Centre for Field Robotics, University of Sydney, 2007. URL `www-personal.acfr.usyd.edu.au/tbailey/papers/delayedstateddf.pdf`.

[17] Tim Bailey, Juan Nieto, Jose Guivant, Michael Stevens, and Eduardo Nebot. Consistency of the EKF-SLAM algorithm. In *Proc. IEEE/RSJ International Conference of Intelligent Robots and Systems (IROS)*, Beijing, China, October 2006.

[18] Tim Bailey, Mitch Bryson, Hua Mu, John Vial, Lachlan McCalman, and Hugh Durrant-Whyte. Decentralised cooperative localisation for heterogeneous teams of mobile robots. In *Proc. IEEE International Conference on Robotics and Automation (ICRA'11)*, Shanghai, China, May 2011.

[19] S. Bandyopadhyay and E. Coyle. An energy efficient hierarchical clustering algorithm for wireless sensor networks. In *Proc. of IEEE Infocom*, San Francisco, April 2003.

[20] Yaakov Bar-Shalom. Update with out-of-sequence measurements in tracking: Exact solution. *IEEE Trans. Aerosp. Electron. Syst.*, 28(2):769–778, July 2002.

[21] Yaakov Bar-Shalom, X.-Rong Li, and Thiagalingam Kirubarajan. *Estimation with Applications to Tracking and Navigation.* John Wiley and Sons, 2001.

[22] Sven Bauer, Marcus Obst, and Gerd Wanielik. 3D environment modeling for GPS multipath detection in urban areas. In *Proc. the 9th IEEE International Multi-Conference on System, Signal and Devices*, pages 1–5, Chemnitz, Germany, March 2012.

[23] Thomas Bengtsson, Peter Bickel, and Bo Li. Curse-of-dimensionality revisited collapse of the particle filter in very large scale systems. In *Probability and statistics: essays in honor of David A. Freedman*, volume 2, pages 316–334. Institute of Mathematical Statistics, 2008.

[24] Maren Bennewitz, Wolfram Burgard, Grzegorz Cielniak, and Sebastian Thrun. Learning motion patterns of people for compliant robot motion. *Internationl Journal of Robotics Research*, 24(1):31–48, 2005.

[25] Alessandro Berni, Diego Merani, John Potter, and Robert Been. Heterogeneous system framework for underwater networking. In *Proc. of IEEE Military Communications Conference (MILCOM) 2011*, pages 2050–2056, September 2011.

[26] David Bétaille and Rafael Toledo-Moreo. Creating enhanced maps for lane-level vehicle navigation. *IEEE Trans. Intell. Transport. Syst.*, 11(4): 786–798, December 2010.

[27] Peter Bickel, Bo Li, and Thomas Bengtsson. Sharp failure rates for the bootstrap particle filter in high dimensions. In *Pushing the limits of contemporary statistics: contributions in honor of Jayanta K. Ghoshreedman*, volume 3, pages 318–329. Institute of Mathematical Statistics, 2008.

[28] Christopher M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.

[29] Azzedine Boukerche, Horacio A.B.F. Oliveira, Eduardo F. Nakamura, and Antonio A.F. Loureiro. Vehicular ad hoc networks: A new challenge for localisation-based systems. *Computer Communication*, 2008.

[30] Frédéric Bourgault and Hugh F. Durrant-Whyte. Communication in general decentralized filters and the coordinated search strategy. In *Proc. of the 7th International conference on information fusion (FUSION'04)*, pages 723–770, Stockholm, Sweden, June 2004.

[31] Sotiris Brakatsoulas, Dieter Pfoser, Randall Salas, and Carola Wenk. On map-matching for vehicle tracking data. In *Proc. of the 31st VLDB Conference*, Trondheim, Norway, 2005.

[32] J. Capitán, L. Merino, F. Caballero, and A. Ollero. Delayed-state information filter for cooperative decentralised tracking. In *Proc. IEEE International Conference on Robotics and Automation (ICRA'09)*, Kobe, Japan, May 2009.

[33] George Casella and Edward I. George. Explaining the Gibbs sampler. *The American Statistician*, 46(3):167–174, 1992.

[34] Dewang Chen, Junping Zhang, Jue Wang, and Fei-Yue Wang. Freeway traffic stream modeling based on principle curves. In *Proc. the 6th IEEE International Conference on Intelligent Transportation Systems (ITSC'03)*, pages 368–372, Shanghai, China, 2003.

[35] Hongyang Chen, Marcelo H.T. Martins, Pei Huang, Hing-Cheung So, and Kaoru Sezaki. Cooperative node localization for mobile sensor networks. In *Proc. of IEEE/IPIP International Conference on Embedded and Ubiquitous Computing (EUC2008)*, pages 607–308, Shanghai, China, December 2008.

[36] Wenping Chen and Xiaofeng Meng. A cooperative localization scheme for zigbee-based wireless sensor networks. In *Proc. the 14th IEEE International Conference (ICON'06)*, volume 2, pages 1–5, September 2006.

[37] Yaakov Bar-Shalom Huimin Chen and Mahendra Mallick. One-step solution for the multistep out-of-sequence measurement problem in tracking. *IEEE Trans. Aerosp. Electron. Syst.*, 40(1):27–37, January 2004.

[38] Zhe Chen. Bayesian filtering: From Kalman filters to particle filters, and beyond. Technical report, Adaptive Syst. Lab., McMaster Univ., Hamilton, ON, Canada, 2003. URL `http://soma.crl.mcmaster.ca/zhechen/homepage.htm`.

[39] Bing Hwa Cheng, Ralph E. Hudson, Flavio Lorenzelli, Lieven Vandenberghe, and Kung Yao. Distributed Gauss-Newton method for node localisation in wireless sensor networks. In *Proc. of the 6th IEEE Workshop on Signal Processing Advances in Wireless Communications (SPAWC2005)*, New York, USA, June 2005.

[40] Sing-Yiu Cheung and Pravin Varaiya. Traffic surveillance by wireless sensor networks. Technical Report UCB-ITS-PRR-2007-4, Institute of Transportation Studies, University of California, Berkeley, 2007.

[41] Siddhartha Chib and Edward Greenburg. Understanding the Metropolis-Hastings algorithm. *American Statistician*, 49(4):327–335, November 1995.

[42] J. J. Craig. *Introduction to Robotics: Mechanics and Control*. Addison Wesley, 2 edition, 1989.

[43] Arie Croitoru, Peggy Agouris, and Anthony Stefanidis. 3D trajectory matching by pose normalisation. In *Proc. of the 13th annual ACM international worship on GIS*, pages 153–162, New York, NY, USA, 2005.

[44] Fred Daum and Jim Huang. Curse of dimensionality and particle filters. In *Proc. of the IEEE Aerospace Conference*, volume 4, pages 1979–1993, March 2003.

[45] Jean Diebolt and Christian P. Robert. Estimation of finite mixture distributions through Bayesian sampling. *Journal of the Royal Statistical Society Series B Statistical Methodology*, 56:363–375, 1994.

[46] Kevin R. Dixon and Pradeep K. Khosla. Trajectory representation using sequenced linear dynamical systems. In *Proc. of IEEE International Conference on Robotics and Automation*, New Orleans, Louisiana USA, April 2004.

[47] Liang Dong. Cooperative localisation and tracking of mobile ad hoc networks. *IEEE Trans. Signal Processing*, 60(7):3907–3913, July 2012.

[48] Arnaud Doucet and Roman Holenstein. Particle Markov chain Monte Carlo methods. *Journal of the Royal Statistical Society Series B Statistical Methodology*, 72(Part 3):269–342, 2010.

[49] Arnaud Doucet and Adam M. Johansen. A tutorial on particle filtering and smoothing: fifteen years later. In *The Oxford Handbook of Nonlinear Filtering*. Oxford University Press, Oxford, 2009.

[50] Arnaud Doucet, Nando de Freitas, Kevin Murphy, and Stuart Russell. Rao-Blackwellised particle filtering for dynamic Bayesian networks. In *Proc. the 16th Annual Conference on Uncertainty in Artificial Intelligence (UAI'00)*, pages 176–183, San Francisco, California, USA, 2000.

[51] Hugh Durrant-Whyte and Mike Stevens. Data fusion in decentralised sensing networks. In *Proc. of the 4th International Conference on Information Fusion*, Montreal, QC, Canada, 2001.

[52] Jochen Eisner, Stefan Funke, Andre Herbst, Andreas Spillner, and Sabine Storandt. Algorithms for matching and predicting trajectories. In *Proc. of the 13th Workshop on Algorithm Engineering and Experiments (ALENEX)*, pages 84–95, San Francisco, California, USA, January 2011.

[53] Tolga Eren. Cooperative localisation in wireless ad hoc and sensor networks using hybrid distance and bearing (angle of arrival) measurements. *EURASIP Journal on Wireless Communications and Networking*, pages 541–552, August 2011. 2011:72.

[54] Yaser P. Fallah, Ching-Ling Huang, Raja Sengupta, and Hariharan Krishnan. Analysis of information dissemination in vehicular ad-hoc networks with application to cooperative vehicle safety systems. *IEEE Trans. Veh. Technol.*, 60(1):233–247, January 2011.

[55] Dieter Fox, Wolfram Burgard, Hannes Kruppa, and Sebastian Thrun. A monte carlo algorithm for multi-robot localization. Technical Report CMU-CS-99-120, Computer Science Department, Carnegie Mellon University, Pittsburgh, PA, 1999.

[56] Dieter Fox, Wolfram Burgard, Hannes Kruppa, and Sebastian Thrun. A probabilistic approach to collaborative multi-robot localisation. *Auton. Robots*, 8(3):325–344, June 2000.

[57] Roberto Fraile and Steve Maybank. Vehicle trajectory approximation and classification. In *Proc. British Machine Vision Conference*, 1998.

[58] Stuart Geman and Donald Geman. Stochastic relaxation, Gibbs distribution and Bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6:721–741, 1984.

[59] Walter R. Gilks and Carlo Berzuini. Following a moving target: Monte Carlo inference for dynamic Bayesian models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 63(1):127–146, 2001.

[60] Paul Bui Quang Christian Musso Francois Le Gland. An insight into the issue of dimensionality in particle filtering. In *Proc. of the 13th IEEE Conference on Information Fusion (FUSION)*, pages 1–8, July 2010.

[61] Aloor Gopakumar and Lillykutty Jacob. Performance of some metaheuristic algorithms for localisation in wireless sensor networks. *International journal of network management*, 19(5):355–373, October 2009.

[62] S. Grime and Hugh F. Durrant-Whyte. Data fusion in decentralised sensor networks. *Control Engineering Practice*, 2(5):849–863, 1994.

[63] Alexei Gritai, Yaser Sheikh, Cen Rao, and Mubarak Shah. Matching trajectories of anatomical landmarks under viewpoint antrhopometric and temporal transforms. *International Journal of Computer Vision*, 84(3): 325–343, September 2009.

[64] Yang Gu and Manuela Veloso. Effective multi-model motion tracking using action models. *International Journal of Robotics Research*, 28(1):3–19, January 2009.

[65] Fredrik Gustafsson and Fredrik Gunnarsson. Mobile positioning using wireless networks: Possibilities and fundamental limitations based on available wireless network measurements. *IEEE Signal Processing Mag.*, 22(4):41–53, July 2005.

[66] Markus Hahn, Lars Krüger, and Christian Wöhler. 3D action recoginition and long-term prediction of human motion. In *ICVS Lecture Notes in Computer Science*, pages 23–32, Springer, 2008.

[67] Carsten Hasberg, Stefan Hensel, and Christoph Stiller. Simultaneous localisation and mapping for path-constrained motion. *IEEE Trans. Intell. Transport. Syst.*, 13(2):541–552, June 2012.

[68] H. Hassanein and A. Safwat. Virtual base stations for wireless mobile ad hoc communications: and infrastructure for the infrastructure-less. *International Journal of Communication Systems*, 14:763–782, 2001.

[69] W. K. Hastings. Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, 57(1):97–109, April 1970.

[70] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan. Energy-efficient communication protocol for wireless microsensor networks. In *Proc. of the Hawaii International Conference on System Sceinces*, Hawaii, 2000.

[71] Sven Hellbach, Sören Strauss, Julian P. Eggert, Edgar Körner, and Horst-Michael Gross. Echo state networks for online prediction of movement data - comparing investigations. In *Proc. International Conference on Artificial Neural Networks (ICANN)*, pages 710–719, Prague, 2008.

[72] Christoph Hermes, Christian Wöhler, Konrad Schenk, and Franz Kummert. Long-term vehicle motion prediction. In *Proc. IEEE Intelligent Vehicles Symposium*, Xi'an, China, June 2009.

[73] Christoph Hermes, Markus Hahn, Julian Einhaus, Christian Wöhler, and Franz Kummert. Vehicle tracking and motion prediction in complex urban scenarios. In *Proc. IEEE Intelligent Vehicles Symposium*, San Diego, USA, 2010.

[74] Jan Hoffmann, Michael Spranger, Daniel Göhring, and Matthias Jüngel. Making use of what you don't see: Negative information in Markov localization. In *Proc. IEEE/RSJ International Conference of Intelligent Robots and Systems (IROS)*, Beijing, China, October 2006.

[75] D. Höhnel, W. Burgard, D. Fox, K. Fishkin, and M. Philipose. Mapping and localization with RFID technology. In *Proc. IEEE International Conference on Robotics and Automation (ICRA'04)*, New Orleans, USA, 2004.

[76] Andrew Howard, Maja J Matarić, and Gaurav S Sukhatme. Putting the 'I' in 'team': an ego-centric approach to cooperative localisation. In *Proc. IEEE International Conference on Robotics and Automation (ICRA'03)*, Taipei, Taiwan, September 2003.

[77] Congwei Hu, Wu Chen, Yongqi Chen, and Dajie Liu. Adaptive Kalman filtering for vehicle navigation. *Journal of Global Positioning Systems*, 2(1): 42–47, 2003.

[78] W. Hu, D. Xie, T. Tieniu, and S. Maybank. Learning activity patterns using fuzzy self-organizing neural network. *IEEE Trans. Syst., Man, Cybern.*, 34(3): 1618–1626, June 2004.

[79] Weiming Hu, Xuejuan Xiao, Zhouyu Fu, Dan Xie, Tieniu Tan, and Steve Maybank. A system for learning statistical motion patterns. *IEEE Trans. Pattern Anal. Machine Intell.*, 28:1450–1464, 2006.

[80] Guoquan P. Huang, Nikolas Trawny, Anastasios I. Mourikis, and Stergios I. Roumeliotis. Observability-based consistent EKF estimators for multi-robot cooperative localization. *Auton. Robots*, 30(1):99–122, 2011.

[81] Carine Hue, Jean-Pierre Le Cadre, and Patrick Pérez. Sequential monte carlo methods for multiple target tracking and data fusion. *IEEE Trans. Signal Processing*, 50(2):309–325, 2002.

[82] Sajid Hussain and Abdul W. Matin. Hierarchical cluster-based routing in wireless sensor networks. In *Proc. of the 5th International Conference on Information Processing in Sensor Networks*, 2006.

[83] Jung-Hoon Hwang, Ronald C. Arkin, and Dong-Soo Kwon. Mobile robots at your fingertip: Bezier curve on-line trajectory generation for supervisory control. In *Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2003.

[84] IEEE. IEEE 1609 - family of standards for wireless access in vehicular environments (WAVE). `http://www.standards.its.dot.gov/Factsheets/Factsheet/80`. 2013. [Online; accessed 17-Aug-2013].

[85] IETF. IETF RFC standards. `http://www.dtnrg.org/wiki/Docs`. 2013. [Online; accessed 17-Aug-2013].

[86] Alexander T. Ihler, John W. Fisher, Randolph L. Moses, and Alan S. Willsky. Nonparametric belief propagation for self-localisation of sensor networks. *IEEE Journal on Selected Areas in Communications*, 23(4):809–819, 2005.

[87] Auke Jan Ijspeert, Jun Nakanishi, and Stefan Schaal. Trajectory formation for imitation with nonlinear dynamical systems. In *Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2001.

[88] Chalermek Intanagonwiwat, Ramesh Govindan, and Deborah Estrin. Directed diffusion: A scalable and robust communication pradigm for sensor networks. In *Proc. of the 6th international ACM conference on mobile computing and networking (MOBICOM)*, pages 56–67, Boston, MA, August 2000.

[89] Andrew Jenkins, Sebastian Kuzminsky, Kevin K. Gifford, Robert L. Pitts, and Kelvin Nichols. Delay/disruption-tolerant networking: Flight test results from the international space station. In *Proc. of IEEE Aerospace conference*, March 2010.

[90] Philo Juang, Hidekazu Oki, Yong Wang, Margaret Martonosi, Li-Shiuan Peh, and Daniel Rubenstein. Energy efficient computing for wildlife tracking: Design tradeoffs and early experiences with zebranet. In *Proc. of Conference on Architechtural Support for Programming Languages and Operating Systems (ASPLOS)*, pages 96–107, October 2002.

[91] R. Jurdak. *Wireless ad hoc and sensor networks: A cross-layer design perspective*. Springer, New York, 2007.

[92] R. E. Kalman. A new approach to linear filtering and prediction problems. *Transactions of the ASME Journal of Basic Engineering*, 82:35–45, 1960.

[93] Shehzad Khalid and Andrew Naftel. Classifying spatiotemporal object trajectories using unsupervised learning of basis function coefficients. *Multimedia Systems*, 12(3):227–238, December 2006.

[94] Zia Khan, Tucker Balch, and Frank Dellaert. MCMC data association and sparse factorization updating for real time multitarget tracking with merged and multiple measurements. *IEEE Trans. Pattern Anal. Machine Intell.*, 28 (12):1960–1972, December 2006.

[95] Sung-Soo Kim and Yong-Bin Kang. Congestion avoidance algorithm using extended Kalman filter. In *Proc. International Conference on Convergence Information Technology (ICCIT'07)*, pages 913–918, 2007.

[96] ZuWhan Kim. Robust lane detection and tracking in challenging scenarios. *IEEE Trans. Intell. Transport. Syst.*, 9(1):16–26, March 2008.

[97] Genshiro Kitagawa. Non-Gaussian state-space modeling of nonstationary time series. *Journal of the American Statistical Association*, 82:1032–1063, 1987.

[98] Mike Klaas, Mark Briers, Nando de Freitas, Arnaud Doucet, Simon Maskell, and Dustin Lang. Fast particle smoothing: If i had a million particles. In *Proc. International Conference on Machine Learning (ICML)*, pages 25–29, 2006.

[99] G. Kloos, J. E. Guivant, S. Worrall, A. Maclean, and E. M. Nebot. Wireless network for mining applications. In *Proc. Australasian Conference on Robotics and Automation*, Canberra, Australia, December 2004.

[100] Frédéric Large, Dizan Vasquez, Thierry Fraichard, and Christian Laugier. Avoiding cars and pedestrians using velocity obstacles and motion prediction. In *Proc. IEEE Intelligent Vehicle Symposium*, Parma, Italy, June 2004.

[101] Ilias Leontiadis and Cecilia Mascolo. Geopps: Geographical opportunistic routing for vehicular networks. In *Proc. of IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM 2007)*, pages 1–6, Helsinki, Finland, 2007.

[102] Keith Y. K. Leung, Timothy D. Barfoot, and Hugh H. T. Liu. Decentralised localisation of sparsely-communicating robot networks: A centralised-equivalent approach. *IEEE Trans. Robot.*, 26(1):62–77, February 2010.

[103] Keith Y. K. Leung, Timothy D. Barfoot, and Hugh H. T. Liu. Decentralised cooperative slam for sparsely-communicating robot networks a centralised-equivalent approach. *Journal of Intelligent and Robotic Systems*, 66(3):321–342, 2012.

[104] Fan Li and Yu Wang. Routing in vehicular ad hoc networks: A survey. *IEEE Vehicular Tech. Mag.*, 2(2):12–22, June 2007.

[105] C. R. Lin and M. Gerla. Adaptive clustering for mobile wireless networks. *Journal on Selected Areas in Communication*, 15:1265–1275, 1997.

[106] Jun S. Liu. *Monte Carlo strategies in scientific computing.* Springer-Verlag, New York, 2001.

[107] Jun S. Liu and Rong Chen. Sequential monte carlo methods for dynamic systems. *Journal of the American Statistical Association*, 93(443):1032–1044, September 1998.

[108] Yin Lou, Chengyang Zhang, Yu Zheng, Xing Xie, Wei Wang, and Yan Huang. Map-matching for low-sampling-rate GPS trajectories. In *Proc. of the 17th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems (ACM GIS'09)*, Seattle, Washington, USA, 2009.

[109] R. Madhavan, Z. Kootbally, and C. Schlenoff. Prediction in dynamic environments for autonomous on-road driving. In *Proc. the 9th International Conference on Control, Automation, Robotics and Vision (ICARCV'06)*, pages 1–6, Singapore, December 2006.

[110] Raj Madhavan, Kingsley Fregene, and Lynne E. Parker. Distributed cooperative outdoor multirobot localisation and mapping. *Auton. Robots*, 17 (1):23–39, July 2004.

[111] Diego Merani, Alessandro Berni, John Potter, and Ricardo Martins. Heterogeneous system framework for underwater networking. In *Proc. of IEEE Baltic Congress on Future Internet Communications (BCFIC)*, pages 103–108, February 2011.

[112] Nicholas Metropolis and S. Ulam. The Monte Carlo methods. *Journal of the American Statistical Association*, 44(247):335–341, September 1949.

[113] Nicholas Metropolis, Arianna W. Rosenbluth, Marshall N. Rosenbluth, Augusta H. Teller, and Edward Teller. Equation of state calculations by fast computing machines. *Journal of Chemical Physics*, 21(6):1087–1092, June 1953.

[114] Mirko Meuter, Anton Kummert, and Stefan Müller-Schneiders. 3D traffic sign tracking using a particle filter. In *Proc. the 11th IEEE International Conference on Intelligent Transportation Systems (ITSC'08)*, pages 168–173, Beijing, China, October 2008.

[115] V. Mhatre and C. Rosenberg. Design guidelines for wireless sensor networks: Communication, clustering and aggregation. *Ad Hoc Networks*, 2(1):45–63, January 2004.

[116] Lyudmila Mihaylova and Avishy Carmi. Particle algorithms for filtering in high dimensional state space: A case study in group object tracking. In *Proc. of the IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 5932–5935, 2011.

[117] Sudip Misra, Subhas Chandra Misra, and Issac Woungang. *Guide to Wireless Mesh Networks*. Springer, London, 2009.

[118] Hua Mu, Tim Bailey, Paul Thompson, and Hugh Durrant-Whyte. Decentralised solutions to the cooperative multi-platform navigation problem. *IEEE Trans. Aerosp. Electron. Syst.*, 47(2):1433–1449, 2011.

[119] Stefan Munder, Christoph Schnörr, and Dariu M. Gavrila. Pedestrian detection and tracking using a mixture of view-based shape-texture models. *IEEE Trans. Intell. Transport. Syst.*, 9(2):333–343, June 2008.

[120] Eduardo M. Nebot, Jose Guivant, and Stewart Worrall. Haul truck alignment monitoring and operator warning system. *Journal of Field Robotics*, 23(2): 141–161, 2006.

[121] Esha D. Nerurkar, Stergios I. Roumeliotis, and Agostino Martinelli. Distributed maximum a posteriori estimation for multi-robot cooperative localisation. In *Proc. IEEE International Conference on Robotics and Automation (ICRA'09)*, Kobe, Japan, May 2009.

[122] Eric W. Nettleton and Hugh F. Durrant-Whyte. Delayed and asequent data in decentralised sensing networks. In *Society of Photo-Optical Instrumentation Engineering (SPIE) Conference*, volume 4571, pages 1–9, 2001.

[123] A. Nieto and K. Dagdelen. Development of a dump edge and vehicle proximity warning system based on GPS and wireless networks to improve safety in open pit mines. *SME Transactions*, 320:11–20, 2006.

[124] Marcus Obst, Sven Bauer, and Gerd Wanielik. Urban multipath detection and mitigation with dynamic 3D maps for reliable land vehicle localisation. In *Proc. IEEE/ION Position Location and Navigation Symposium (PLANS)*, pages 685–691, Myrtle Beach, South Carolina, April 2012.

[125] Se-Yong Oh and Younguk Yim. Modelling of vehicle dynamics from real vehicle measurements using a neural network with two-state hybrid learning for accurate long-term prediction. *IEEE Trans. Veh. Technol.*, 53(4): 1076–1084, July 2004.

[126] Lee-Ling Ong, Tim Bailey, and Hugh Durrant-Whyte. Decentralised particle filtering for multiple target tracking in wireless sensor networks. In *Proc. the 11th International Conference on Information Fusion*, pages 342–349, United States, June 2008.

[127] Nayef A. Alsindi Kaveh Pahlavan, Bardia Alavi, and Xinrong Li. A novel cooperative localisation algorithm for indoor sensor networks. In *Proc. the 17th Annual IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC'06)*, Helsinki, Finland, September 2006.

[128] Liam Paninski, Kamiar Rahnama, and Michael Vidne. Robust particle filters via sequential pairwise reparameterised Gibbs sampling. In *Proc. the 46th annual conference on information science and systems (CISS 2012)*, Princeton, NJ, USA, March 2012.

[129] N. Patwari and A. Hero-III. Using proximity and quantized RSS for sensor localization in wireless networks. In *Proc. the 2nd International ACM Workshop on Wireless Sensor Networks and Applications (WSNA)*, San Diego, USA, September 2003.

[130] Neal Patwari, Alfred O. Hero-III, Matt Perkins, Neiyer S. Correal, and Robot J. O'Dea. Relative location estimation in wireless sensor networks. *IEEE Trans. Signal Processing*, 51(8):2137–2148, August 2003.

[131] Neal Patwari, Joshua N. Ash, Spyros Kyperountas, Alfred O. Hero-III, Randolph L. Moses, and Neiyer S. Correal. Locating the nodes: Cooperative localisation in wireless sensor networks. *IEEE Signal Processing Mag.*, 22(4): 54–69, July 2005.

[132] Rong Peng and Mihail L. Sichitiu. Probabilistic localisation for outdoor wireless sensor networks. *ACM SIGMOBILE Mobile Computing and Communications Review*, 11(1):53–64, 2007.

[133] Mohammad Jalil Piran, G. Rama Murthy, and G. Praveen Babu. Vehicular ad hoc and sensor networks: Principles and challenges. *International Journal of Ad hoc, Sensor & Ubiquitous Computing (IJASUC)*, 2(2):38–49, June 2011.

[134] D. Ponsa, A. López, J. Serrat, F. Lumbreras, and T. Graf. Multiple vehicle 3D tracking using an unscented Kalman filter. In *Proc. the 8th IEEE International Conference on Intelligent Transportation Systems (ITSC'05)*, pages 1108–1113, Vienna, Austria, September 2005.

[135] G. J. Pottie and W. J. Kaiser. *Principles of Embedded Networked Systems Design.* Cambridge University Press, New York, 2005.

[136] Nissanka B. Priyantha, Hari Balakrishnan, Erik D. Demaine, and Seth J. Teller. Anchor-free distributed localisation in sensor networks. In *Proc. of the 1st international conference on Embedded networked sensor systems (SensSys'03)*, pages 340–341, 2003.

[137] K. Punithankumar, T. Kirubarajan, and A. Sinha. Multiple-model probability hypothesis density filter for tracking maneuvering targets. *IEEE Trans. Aerosp. Electron. Syst.*, 44(1):87–98, January 2008.

[138] H. Qasem and L. Reindl. Unscented and extended Kalman estimators for non linear indoor tracking using distance measurements. In *Proc. the 4th Workshop on Positioning, Navigation and Communication (WPNC'07)*, pages 177–181, 2007.

[139] Olivier Rabaste. Multi-target tracking with MCMC-based particle filters. In *Proc. the 18th European Signal Processing Conference (EUSIPCO-2010)*, Aalborg, Denmark, August 2010.

[140] Tod M. Ruff and Thomas P. Holden. Preventing collisions involving surface mining equipment - A GPS based approach. *Journal of Safety Research*, 34 (2):175–181, 2003.

[141] Todd M. Ruff. Test results for collision warning systems for surface mining dump trucks. Technical Report RI 9652, National Institute for Occupational Safety and Health (NIOSH), 2000.

[142] Todd M. Ruff. Recommendations for testing radar-based collision warning systems on heavy equipment. Technical Report RI 9657, National Institute for Occupational Safety and Health (NIOSH), 2002.

[143] P. Santi. *Topology Control in Wireless Ad Hoc and Sensor Networks.* John Wiley & Sons Ltd, Chichester, 2005.

[144] S. K. Sarkar, T. G. Basavaraju, and C. Puttamadappa. *Ad Hoc Mobile Wireless Networks: Principles, Protocols, and Applications*. Auerbach Publications, New York, 2008.

[145] Thuraiappah Sathyan and Mark Hedley. A particle filtering algorithm for cooperative tracking of nodes in wireless networks. In *Proc. the 22nd IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC'11)*, pages 1304–1308, 2011.

[146] Vladimir Savic and Santiago Zazo. Cooperative localisation in mobile networks using nonparametric variants of belief propagation. *Ad Hoc Networks*, 11:138–150, 2013.

[147] Vladimir Savic, Adrián Población, Santiago Zazo, and Mariano García. Indoor positioning using nonparametric belief propagation based on spanning trees. *EURASIP Journal on Wireless Communications and Networking*, 2010, 2010.

[148] Ali H. Sayed, Alireza Tarighat, and Nima Khajehnouri. Network-based wireless location: challenges faced in developing techniques for accurate wireless location information. *IEEE Signal Processing Mag.*, 22(4):24–40, July 2005.

[149] Mao Shan, Stewart Worrall, and Eduardo Nebot. Long term vehicle motion prediction and tracking in large environments. In *Proc. the 14th IEEE International Conference on Intelligent Transportation Systems (ITSC'11)*, Washington, DC, USA, October 2011.

[150] Mao Shan, Stewart Worrall, and Eduardo Nebot. Probabilistic long-term vehicle motion prediction and tracking in large environments. *IEEE Trans. Intell. Transport. Syst.*, 14(2):539–552, June 2013.

[151] Mao Shan, Stewart Worrall, Favio Masson, and Eduardo Nebot. Using delayed observations for long-term vehicle tracking in large environments. *IEEE Trans. Intell. Transport. Syst.*, PP(99):1–15, January 2014.

[152] Mao Shan, Stewart Worrall, and Eduardo Nebot. Nonparametric cooperative tracking in mobile ad-hoc network. In *Proc. IEEE International Conference on Robotics and Automation (ICRA'14)*, pages 1–8, Hongkong, China, June 2014. accepted.

[153] Sayanan Sivaraman and Mohan Manubhai Trivedi. A general active-learning framework for on-road vehicle recognition and tracking. *IEEE Trans. Intell. Transport. Syst.*, 11(2):267–276, June 2010.

[154] Chris Snyder, Thomas Bengtsson, Peter Bickel, and Jeff Anderson. Obstacles to high-dimensional particle filtering. *Monthly Weather Review*, 136(12): 4629–4640, 2008.

[155] J. Steele, C. Debrunner, and M. Whitehorn. Stereo images for object detection in surface mine safety applications. Technical Report TR20030109, Colorado School of Mines, Western Mining Resource Center, 2003.

[156] J. Su, J. Scott, P. Hui, J. Crowcroft, E. Lara, C. Diot, A. Goel, M.H. Lim, and E. Upton. Haggle: Seamless networking for mobile applications. In *Proc. the 9th IEEE International Conference on Ubiquitous Computing*, Innsbruck, 2007.

[157] D. Svensson and L. Svensson. A new multiple model filter with switch time conditions. *IEEE Trans. Signal Processing*, 58(1):11–25, January 2010.

[158] Sebastian Thrun. Learning metric-topological maps for indoor mobile robot navigation. *Artificial Intelligence*, 99(1):21–71, 1998.

[159] Sebastian Thrun, Dieter Fox, and Wolfram Burgard. Probabilistic methods for state estimation in robotics. In *Proc. of the Workshop SOAVE'97, VDI-Verlag*, 1997.

[160] Sebastian Thrun, Steffen Gutmann, Dieter Fox, Wolfram Burgard, and Benjamine J. Kuipers. Integrating topological and metric maps for mobile robot navigation: A statistical approach. In *Proc. of the 15th National Conference on Artificial Intelligence*, 1998.

[161] Rafael Toledo-Moreo, David Bétaille, and François Peyret. Lane-level integrity provision for navigation and map matching with GNSS, dead reckoning, and enhanced maps. *IEEE Trans. Intell. Transport. Syst.*, 11(1):100–112, March 2010.

[162] Simukai Utete and Hugh F. Durrant-Whyte. Reliability in decentralised data fusion networks. In *Proc. of the 1994 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI'94)*, pages 215–221, Las Vegas, NV, 1994.

[163] Dizan Vasquez, Thierry Fraichard, and Christian Laugier. Incremental learning of statistical motion patterns with growing hidden Markov models. *IEEE Trans. Intell. Transport. Syst.*, 10(3):403–416, September 2009.

[164] Tom Vercauteren and Xiaodong Wang. Decentralized sigma-point information filters for target tracking in collaborative sensor networks. *IEEE Trans. Signal Processing*, pages 2997–3009, 2005.

[165] Michail Vlachos, George Kollios, and Dimitrios Gunopulos. Discovering similar multidimensional trajectories. In *Proc. of the 18th International Conference on Data Engineering (ICDE)*, pages 673–684, San Jose, California, USA, 2002.

[166] Michail Vlachos, George Kollios, and Dimitrios Gunopulos. Elastic translation invariant matching of trajectories. *Machine Learning*, 58(2-3):301–334, 2005.

[167] Artemios G. Voyiatzis. A survey of delay- and disruption-tolerant networking applications. *Journal of Internet Engineering*, 5(1):331–344, June 2012.

[168] Mohamed Watfa. *Advances in Vehicular Ad-Hoc Networks: Developments and Challenges*. Information Science Reference, Hershey, New York, 2010.

[169] Yair Weiss. Correctness of local probability propagation in graphical models with loops. *Neural Computation*, 12(1):1–41, 2000.

[170] Wikipedia. Bingham canyon mine. `http://en.wikipedia.org/wiki/Bingham_Canyon_Mine`. 2013. [Online; accessed 14-Aug-2013].

[171] Moe Z. Win, Yuan Shen, and Henk Wymeersch. On the position error bound in cooperative networks: A geometric approach. In *Proc. of IEEE International Symposium on Spread Spectrum Techniques & Applications*, pages 637–643, Bologna, Italy, August 2008.

[172] Nawaporn Wisitpongphan, Fan Bai, Priyantha Mudalige, Varsha Sadekar, and Ozan Tonguz. Routing in sparce vehicular ad hoc wireless networks. *IEEE J. Select. Areas Commun.*, 25(8):1538–1556, October 2007.

[173] Stewart Worrall. *Providing Situation Awareness in Complex Multi-Vehicle Operations*. PhD thesis, The University of Sydney, March 2009.

[174] Stewart Worrall and Eduardo Mario Nebot. Using non-parametric filters and sparse observations to localise a fleet of mining vehicles. In *Proc. IEEE International Conference on Robotics and Automation (ICRA'07)*, pages 509–516, Roma, Italy, April 2007.

[175] Stewart Worrall and Eduardo Mario Nebot. A probabilistic method for detecting impending vehicle interactions. In *Proc. IEEE International Conference on Robotics and Automation (ICRA'08)*, volume CD, pages 1787–1791, CA, USA, May 2008.

[176] Henk Wymeersch, Ulric Ferner, and Moe Z. Win. Cooperative Bayesian self-tracking for wireless networks. *IEEE Commun. Lett.*, 12(7):505–507, July 2008.

[177] Henk Wymeersch, Jaime Lien, and Moe Z. Win. Cooperative localization in wireless networks. *Proc. of the IEEE*, 97(2):427–450, February 2009.

[178] Jonathan S. Yedidia, William T. Freeman, and Yair Weiss. Constructing free energy approximations and generalised belief propagation algorithms. *IEEE Trans. Inform. Theory*, 51:2282–2312, 2005.

[179] E. Yoneki. ECCO: Data centric asynchronous communication. Technical Report UCAM-CL-TR677, University of Cambridge, 2006.

[180] E. Yoneki and J. Crowcroft. Towards data driven declarative networking in delay tolerant networks. In *Proc. the 2nd International Conference on Distributed Event Based Systems*, Rome, Italy, 2008.

[181] Yihua Yu and Qiansheng Cheng. Particle filters for maneuvering target tracking problem. *Journal of Signal Processing*, 86(1):195–203, January 2006.

[182] Junping Zhang, Fei-Yue Wang, Kunfeng Wang, Wei-Hua Lin, Xin Xu, and Cheng Chen. Data-driven intelligent transportation systems: A survey. *IEEE Trans. Intell. Transport. Syst.*, 12(4):1624–1639, December 2011.

[183] Keshu Zhang, X. Rong Li, and Yunmin Zhu. Optimal update with out-of-sequence measurements. *IEEE Trans. Signal Processing*, 53(6): 1992–2004, June 2005.

[184] Jing Zhao and Guohong Gao. VADD: Vehicle-assisted data delivery in vehicular ad hoc networks. In *Proc. of the 25th IEEE International Conference on Computer Communications (INFOCOM 2006)*, pages 1–12, Barcelona, Spain, 2006.