Contextual Information Aided Target Tracking and Path Planning for Autonomous Ground Vehicles

By

Runxiao Ding

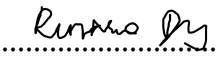
A doctoral thesis submitted in partial fulfilment of the requirements for the award of the degree of Doctor of Philosophy (PhD)

June 2016

Loughborough University Centre for Autonomous Systems,
Department of Aeronautical and Automotive Engineering,
Loughborough University, Loughborough,
Leicestershire, UK, LE11 3TU

CERTIFICATE OF ORIGINALITY

This is to certify that I am responsible for the work submitted in this thesis, that the original work is my own except as specified in acknowledgements or in footnotes, and that neither the thesis nor the original work contained therein has been submitted to this or any other institution for a degree.


..... (Signed)

Runxiao Ding
..... (candidate)

I would like to dedicate this thesis to my loving parents...

Acknowledgements

I would like to register my sincere thanks to my supervisor, Professor Wen-Hua Chen, for his guidance during my PhD study. He provided me such a wonderful opportunity to do research on autonomous vehicle subject which not only leads me to the world most cutting-edge technology area but also improves my abilities and brings me the confidence of solving any tough tasks. I have the great respect and appreciation for his erudite knowledge and rigorous work attitude. Without his brilliant advice, trust and great encouragement, the completion of this project would be impossible.

Special thanks to Dr Miao Yu and Dr Hyondong Oh for their assistance and advices during the process of my research especially in academic writing and knowledge sharing. Their broad knowledge and useful discussion always inspire me and make me keep on moving forward.

Many thanks to all the members from Aeronautical and Automotive Engineering Department who provided the funding for my work through the university studentship and special thanks to my colleagues from Loughborough University Centre of Autonomous Systems (LUCAS) for providing such a great academic and research environment.

Finally I would like to thank my parents Wei Ding and Xuan Xin for their support and inspiration through my life and all my close friends Qianhan Gong, Chunyan Wu, etc. for their encouragement during the final stage of my PhD.

Abstract

Recently, autonomous vehicles have received worldwide attentions from academic research, automotive industry and the general public. In order to achieve a higher level of automation, one of the most fundamental requirements of autonomous vehicles is the capability to respond to internal and external changes in a safe, timely and appropriate manner. Situational awareness and decision making are two crucial enabling technologies for safe operation of autonomous vehicles.

This thesis presents a solution for improving the automation level of autonomous vehicles in both situational awareness and decision making aspects by utilising additional domain knowledge such as constraints and influence on a moving object caused by environment and interaction between different moving objects. This includes two specific sub-systems, model based target tracking in environmental perception module and motion planning in path planning module.

In the first part, a rigorous Bayesian framework is developed for pooling road constraint information and sensor measurement data of a ground vehicle to provide better situational awareness. Consequently, a new multiple targets tracking (MTT) strategy is proposed for solving target tracking problems with nonlinear dynamic systems and additional state constraints. Besides road constraint information, a vehicle movement is generally affected by its surrounding environment known as interaction information. A novel dynamic modelling approach is then proposed by considering the interaction information as 'virtual force' which is constructed by involving the target state, desired dynamics and interaction information. The proposed modelling approach is then accommodated in the proposed MTT strategy for incorporating different types of domain knowledge in a comprehensive manner.

In the second part, a new path planning strategy for autonomous vehicles operating in partially known dynamic environment is suggested. The proposed MTT technique is utilized to provide accurate on-board tracking information with associated level of uncertainty. Based on the tracking information, a path planning strategy is developed to generate collision free paths by not only predicting the future states of the moving objects but also taking into account the propagation of the associated estimation uncertainty within a given horizon. To cope with a dynamic and uncertain road environment, the strategy is implemented in a receding horizon fashion.

Contents

List of Figures

List of Tables

Symbols

General State Estimation

x	State vector for target vehicle in MTT
x	Position along x axis in Cartesian coordinate
y	Position along y axis in Cartesian coordinate
$\dot{x}$	Velocity along x axis in Cartesian coordinate
$\ddot{x}$	Acceleration along x axis in Cartesian coordinate
$\ddot{x}$	Acceleration along x axis in Cartesian coordinate
$\ddot{y}$	Acceleration along y axis in Cartesian coordinate
$\hat{\mathrm{x}}_{k\|k}$	Estimated state at time k
$P_{k\|k}$	Covariance of the estimated state at time k
$\hat{\mathrm{x}}_{k\|k-1}$	Predicted state at time k
$P_{k\|k-1}$	Covariance of the predicted state at time k
$\tilde{\mathrm{x}}_{k\|k-1}$	Constrained state prediction
z_k	Measurement at time k
z_k^m	Position measurement at time k
$\hat{\mathrm{z}}_{k\|k-1}$	Predicted measurement at time k
$\tilde{\mathrm{z}}_k$	The projected constrained measurement at time k
$\tilde{\mathrm{x}}_{k\|k-1}^p$	The constrained predicted target position
r_k	Measurement range at time k
θ_k	Measurement bearing angle at time k

s_k	State of sensor (ego vehicle) at time k
$S_{k\|k-1}$	Innovation covariance of the predicted measurement at time k
$\tilde{S}_{k/k-1}$	Constrained Innovation covariance at time k
K	The Kalman gain
Q	Covariance matrix for process noise
R	Covariance matrix for measurement noise
$\tilde{R}$	The projected measurement error covariance matrix
$\{\cdot\}^T$	Matrix transpose operator
F	Linear state transition matrix
G	(Acceleration) Input matrix
H	Linear measurement matrix
$f(\cdot)$	Nonlinear dynamic function
$h(\cdot)$	Nonlinear measurement function
$C(\cdot)$	Linear inequality constraint matrix
$\mathcal{L}(\cdot)$	Nonlinear inequality constraint function
C_k	The constraint function
$\tilde{C}_k$	The active constraint function
$\tilde{Q}$	Projected process error covariance matrix
$\tilde{v}_k$	The measurement noise for the projected measurement
ω_k	Process noise at time k
v_k	Measurement noise at time k
a_k^e	Acceleration introduced by the environmental force
T	Sample period

Moving Horizon Estimation

$\Gamma_{k-N}(\mathrm{x}_{k-N})$	MHE arrival cost for initial state x_{k-N}
$\hat{\mathrm{x}}_{k-N}^{mh}$	Estimated state at $k-N$ which is used for arrival cost calculation
$\hat{\mathrm{x}}_{k-N}^{*}$	Optimised initial state
$\hat{\mathrm{x}}_{k}^{*}$	Optimised estimated state at time k by MHE
$\{\hat{\omega}_{j}^{*}\}_{j=k-N}^{k-1}$	The optimised process noise sequence for a horizon of *N* steps
$p(\cdot)=\mathcal{N}(\mu,R)$	Normal (Gaussian) probability distribution with mean μ and covariance Σ

Multiple Target Tracking Parameters

P_D	Detection probability
β_{FA}	False alarm density
β_{NT}	New target density
$p_{k/k-1}^{j}\left(\mathrm{Z}_{k}^{\Theta_k^{-1}(j)}\right)$	Predicted measurement density
p_G^j	The gate probability of the *j*th target
$\{\theta_k^i\}$	A set of $i=1,\ldots,N$ hypotheses at time k
$p(\theta_k^i)$	The probability for θ_k^i at time k
V	Gate region
Z_k	A set of all measurements at time k
t_i^n	New target initiation time
t_d^n	Existing target deletion time
T^n	The set of all states for target *n* during the tracking process
m_k^{FA}	The number of false alarms in gate region V at scan k

m_k^{NT} The number of detected new target at scan k

Θ_k An arbitrary association combination about the current measurement set Z_k

A_{ij} The Assignment matrix with dimension of $i \times j$ where i represents the number of measurements and j is the total number of potential tracks

$\widehat{N}_{k/k}^{l}$ Estimate of cardinality at time k for the l^{th} iteration

N_k^{true} The true number of targets at time step k

ub Upper bound of road boundary

lb Lower bound of road boundary

$\boldsymbol{f}_{i,j}^{(\cdot)}$ The interactive force between object i and j

f^e The summed environmental force

$d_{ij}^{prediction}$ The relative Euclidean distance between $\mathrm{x}_i^{predict}$ and position of the object j

T_g^l Transformation matrix representing the rotation from global coordinate l to the road network local coordinate g

$T_g^s(\cdot)$ & $T_s^g(\cdot)$ Transformation matrix for the global coordinate to sensor coordinate and sensor coordinate to the global one, respectively

Path Planning

$q(k)$ State for ego vehicle in path planning

X_{free} Obstacle-free space in path planning

X_{obs} Obstacle region in path planning

c_s Cost parameter related to stationary obstacle

c_m Cost parameter related to moving target vehicles of interest

c_g Cost parameter rewarding moving towards the short term goal

c_u	Cost parameter penalizing control input considering the energy cost
$u^*_{k:k+N-1}$	Full sequence of control input for a horizon of N steps
$u = (u_1, u_2)$	System control input the vehicle heading velocity u_1 and the steering rate u_2

List of Acronyms

ADAS	Advanced Driver Assistance Systems
ACC	Adaptive Cruise Control
AEB	Automated Emergency Braking
APF	Artificial Potential Field
CV	Constant Velocity Model
CA	Constant Acceleration Model
CLA	Cellular Automata Model
CT	Coordinated Turn Model
CMHE	Constrained Moving Horizon Estimation
DDF	Divided Difference Filter
DMHE	Domain Knowledge aided Moving Horizon Estimation
DMHE-MHT	Domain Knowledge aided Moving Horizon Estimation based Multiple Hypothesis Tracking
EKF	Extended Kalman Filter
EM	Expectation Maximization
FISST	Finit Set Statistics
GNN	Global Nearest Neighbour
GM-PHD	Gaussian Mixture Probability Hypothesis Density
GMF	Gaussian Mixture Filter
GPS	Global Positioning System

IMM Interacting Multiple Model

JPDAF Joint Probabilistic Data Association Filter

KF Kalman Filter

LDW Lane Departure Warning

MAP *maximum a posteriori*

MCMC Markov Chain Monte Carlo

MHE Moving Horizon Estimation

MHE-MHT Moving Horizon Estimation based Multiple Hypothesis Tracking

MPC Model Predictive Control

MTT Multiple Target Tracking

MHT Multiple Hypothesis Tracking

MILP Mixed Integer Linear Programming

OSPA Optimal Sub-Pattern Assignment Metric

OV Optimal Velocity

PA Parking assist

PDF Probability Density Function

PF Particle Filter

PHD Probability Hypothesis Density

PMHT Probabilistic Multiple Hypothesis Tracking

PRM Probabilistic Roadmap Planners

RRT Rapidly-Exploring Random Trees

RMSE Root Mean Square Error

TBD Track-Before-Detect

UKF Unscented Kalman Filter

VFH Vector Field Histogram

Chapter 1

Introduction

1.1 Overview

Autonomous vehicles are a rapidly evolving technology which only a few years ago was still considered as science fiction. Recently the field of autonomous vehicles is rapidly growing worldwide, not only from robotics researchers but also in military and civilian applications. Self-driving car is considered as a strategic solution for governments and automotive industries. The advent of autonomous vehicles, which can move autonomously and navigate in daily life including highway, urban and unstructured scenarios, would offer a profound influence in many aspects including the advantages of enhanced road safety, increasing operational efficiency, and also lead to economic benefits such as reduced (fuel) energy usage and personnel. As a result, autonomous vehicles could provide significant economic, environmental and social benefits.

One of the most fundamental requirements of autonomous vehicles is the capability to respond to internal and external changes in a safe, timely and appropriate manner. This process is also known as situational awareness and decision making which are two crucial enabling technologies for safe operation of autonomous vehicles. To a large extent, it could determine the automation level of autonomous vehicle [1] from assisting the driver such as advanced driver assistance systems (ADAS), to fully controlling the vehicle in an autonomous way.

Figure 1.1 illustrates an overview of this thesis and their relationships. The basic framework of autonomous intelligent vehicle has several real-time systems, including perception, localization and map building, path planning and motion control [2]. As shown in Figure 1.1, while perception refers to understanding its measurement through appropriate sensors and tracking moving objects, finding the robot/vehicle pose or configuration in the surrounding environment is localization and map building. Planning the path in accordance with the task by using cognitive decision making is an essential phase before actually accomplishing the preferred trajectory by controlling the motion.

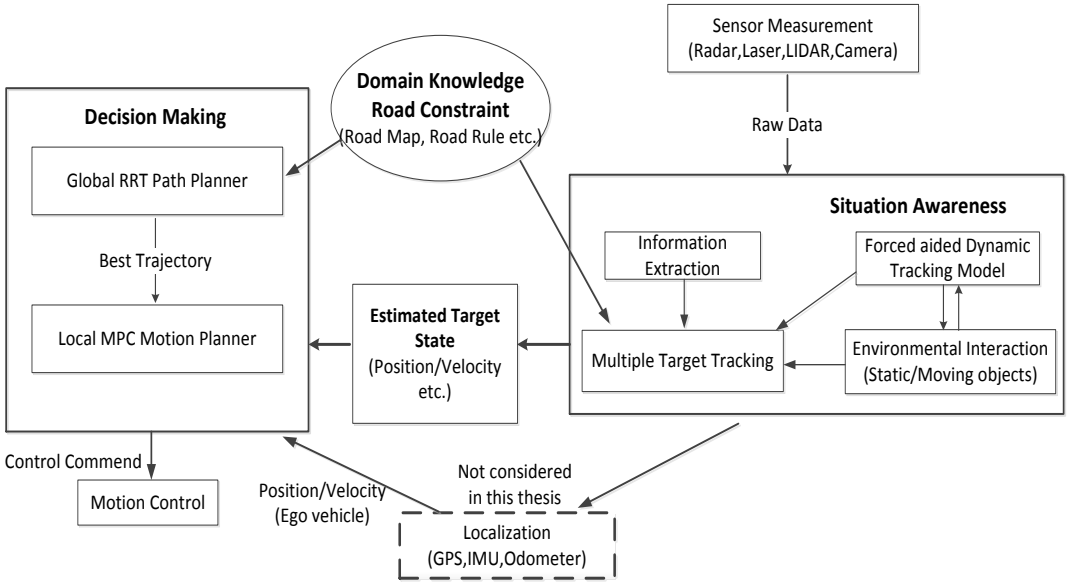


Figure 1.1 An overview of the system presented in this thesis

In situational awareness, sensing and developed algorithms are used to understand the vehicle's surrounding environment. The problem of estimating the movement states (including position, velocity or acceleration) of the surrounding objects such as humans, other vehicles and road users, is an important problem in autonomous vehicles. Knowledge about the position of moving objects can be used for early warning and collision avoidance system in ADAS and further automation decision making systems such as motion planning and control. The technique required to solve this problem is known as multiple target tracking (MTT) which are widely addressed in various military and civilian applications. This thesis concentrates on its use in autonomous vehicle and automotive safety area.

In traditional MTT, the core problem includes the process of collecting data from sensors which may contain potential multiple obstacles and partitioning these data into different sets of observations or tracks. Autonomous vehicles gain its cognitive capabilities by acquiring information from the environment using various automobile sensors like radar, sonar and LIDAR. MTT algorithms are used for perception of the objects in the environment within a volume of time and space. In autonomous vehicles, MTT becomes a very challenging topic since they usually operate in a dynamic, unpredictable environment with incomplete (or inaccurate) sensory information. In probabilistic robotics and statistical approaches, environmental information is usually represented with a known distribution and thus such uncertainty is difficult to be handled with. In order to achieve more effective and accurate traffic information and reduce the perception uncertainty, information in the world model (also known as domain knowledge) such as the operation environment, the rules of the road and interactions between the environmental objects could be used in autonomous vehicle MTT.

This thesis is focused on improving the automation level for autonomous ground vehicle especially on two systems: model-based target tracking in environmental perception module and motion planning in path planning module, by utilising additional domain knowledge such as road constraint and force based interaction information. However, this extra information makes the traditional Gaussian distribution assumption invalid, which is fundamental for most of the current statistical Bayes approaches such as Kalman filtering. Under the Gaussian distribution assumption, the estimated state about a moving object can be represented by its mean and a covariance. In other words, the motion uncertainty can be represented by an uncertain region which is defined by the state mean (located at the centre) under a specified confidence level represented by the covariance. The introduction of domain knowledge such as road constraints due to the road network and interaction between targets or target to environment makes this Gaussian assumption not true anymore.

In this thesis, to address the challenge of non-Gaussian distributions imposed by making use of external domain knowledge information from the world model, a rigorous Bayesian framework is developed for pooling road constraint information and sensor measurement data to provide the better multiple ground targets state estimation. Among various state estimation algorithms, the moving horizon estimation (MHE) is of particular considered in this thesis, because by applying the

optimization based MHE, not only nonlinear dynamic systems but also additional state constraints in target tracking problems can be naturally handled. Besides, the unique moving horizon property of the MHE provides the natural benefit for ground target tracking especially in cluttered environment with noisy measurements and occlusion problems. To solve tracking ambiguity (data association) problem in MTT, an improved multiple hypothesis tracking (MHT) framework is developed by implementing the constrained MHE as state estimation technique namely Moving Horizon Estimation based Multiple Hypothesis Tracking (MHE-MHT). Comparing with traditional MHT, the new MHE-MHT framework inherits the advantages from MHE which makes it suitable for system with nonlinear measurement and capable to systematically deal with state constraint based environmental information such as road width and speed limit in both state estimation and data association layer.

To further improve situational awareness for autonomous vehicle, interaction information for the MTT problem is considered in this thesis. In a realistic ground tracking scenario, a target's movement is generally affected by its surrounding environment considering both stationary and moving objects, which means there are interactions between the tracked target and its surrounding environment in addition to constraint information. For example, the vehicle may be repulsive away or attracted to certain objects in the environment and thus the surrounding environment may interact with vehicle's movement to some level. A novel dynamic modelling approach is proposed in this thesis by considering the interaction between a vehicle and the environment by using a 'virtual force' concept. A new MHE algorithm, domain knowledge aided MHE (DMHE) is then proposed in this thesis. The proposed DMHE framework could effectively incorporate the domain knowledge, including both the constraint information and interaction information in a comprehensive way. The DMHE is then incorporated into MHE-MHT framework namely DMHE-MHT for ground MTT with complicated environmental information.

According to the framework [2] of autonomous vehicle, the improved estimation performance is paramount for autonomous vehicle navigation problems. In this case, a new vehicle motion planning algorithm for partially known dynamic environment is finally developed in this thesis. The environmental estimation information provided by the improved situational awareness approach is fed to this novel path planner. The proposed sensor based MTT technique is utilized to provide accurate on-board tracking information. A horizon length of trajectory prediction information

considering target motion uncertainty is also accommodated in the motion planning strategy implemented by a model predictive control (MPC) method to achieve a collision free motion planning with moving objects.

1.2 Outline

This thesis details the development of domain knowledge aided situational awareness and decision making systems for autonomous vehicle. The contents of each chapter are outlined below.

Chapter 2 - Literature Review

Situational awareness and decision making are two of the most important enabling technologies for safe operation of unmanned vehicles. This research is carried out towards a higher automation level for unmanned vehicles. In this case, a detailed literature review for autonomous ground vehicle application in carried out in both target tracking and motion planning aspects.

Chapter 3 - Single Target Tracking using Constrained Moving Horizon Estimation

A new target tracking strategy by using constrained Moving Horizon Estimation approach is proposed in this chapter. By applying optimization based MHE, not only nonlinear dynamic systems but additional state constraints in target tracking problems such as road width can be naturally handled. The proposed MHE algorithm is demonstrated by single target tracking scenarios verified by both linear and nonlinear measurement models.

Chapter 4 - Multiple Target Tracking in Cluttered Environment with Road Constraint Information

In order to extend the target tracking method developed in the previous chapter for solving multiple target issues, a new MTT strategy MHE-MHT is proposed in this chapter. The proposed MHE-MHT algorithm is demonstrated by a multiple ground vehicle tracking scenario considering road constraints with an unknown and time

varying number of targets observed in clutter environments using nonlinear measurement model.

Chapter 5 - Environmental Interaction Modelling and Target Tracking

In Chapter 5, a novel dynamic modelling approach is the proposed by considering the interaction information between the target and its surrounding environment. The proposed model is then utilised in (multiple) target tracking strategies developed in previous chapters incorporating comprehensive domain knowledge information which include both environmental physical constraints and target interaction force information. Compared with the results in previous Chapter 4, the results in this chapter can provide a better tracking performance in both estimation RMSE and data association accuracy aspects when utilise limited/no interaction information.

Chapter 6 - Autonomous Vehicle Motion Planning in Dynamic Environment with Trajectory Prediction based Collision Avoidance System

Utilising the MTT strategy developed in the previous chapters, an autonomous vehicle motion planning strategy in dynamic environment is suggested in this chapter. The proposed MTT technique is utilized to provide accurate on-board tracking information. Meanwhile, targets' motion prediction information is also accommodated in the motion planning strategy in a stochastic way to achieve a collision free planning.

Chapter 7 - Conclusion and Future Work

This chapter draws conclusions and presents directions and recommendations for further research.

Chapter 2

Literature Review

This chapter presents a review of the literature relevant to the project area, with the discussion broken down into two major areas: **situational awareness** and **decision making** for autonomous vehicles. Firstly, some relevant techniques about situational awareness are discussed including state estimation, multiple target tracking, road constraint information, tracking model and target interaction. Then the relevant challenges associated with autonomous vehicle decision making area, especially algorithms for solving vehicle path planning problem are also discussed in this chapter.

I Situational Awareness

2.1 Autonomous Vehicle Situational Awareness

2.1.1 Definition

Situational awareness is crucial for autonomous driving. A number of different definitions of situational awareness can be found in [3]. Among them, the one given in [4] is most suited to the concept of autonomous vehicle systems and its relevant applications:

"Situational awareness is the perception of the elements in the environment within a volume of time and space, the comprehension of their meaning and the projection of their state in the future" Endsley, 1988 [4]

Endsley in [4] also defined a model of situational awareness. This was reduced and adapted to enable the original definition to be operationalised. The model is called '3-Q' model which includes three fundamental questions shown below in Figure 2.1.

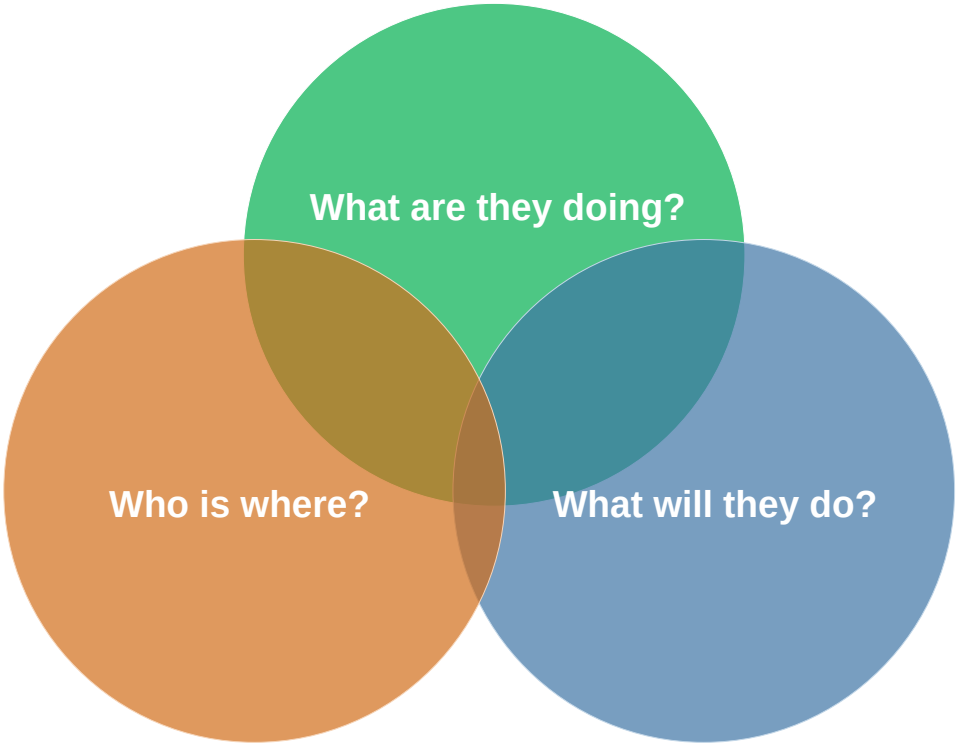


Figure 2.1 '3-Q' model for situational awareness

The 3-Q model can be applied to any activity where the situation is in constant change, and where all data, evaluations and assessments (answers to the three questions) are of a transitory nature [5]. Other paper [3] splits the 3-Q model into three types: Transitory Awareness, Local Awareness, and Global Awareness. Different use cases may require different perception models. Principles of situational awareness when used in different scenarios can be used to derive perception requirements for ADAS and autonomous vehicle features. In other words, the autonomous vehicle perception capabilities rely heavily on its situational awareness.

2.1.2 Contextual Information aided Target Tracking and Situational Awareness

The perception capabilities in autonomous vehicle and relative ADAS are widely considered a key technological component to enhance safety, flow, and efficacy of traffic logistic [7]. Autonomous vehicles and relative ADAS systems acquire information from their environment by sensors such as camera, radar, and LIDAR and build a mental model of the real world based on an interception of this information. In autonomous vehicle systems, this is also known as environmental perception system which usually includes three different domains: road structure, stationary obstacles and dynamic obstacles [6].

The road structure defines where and how vehicles are allowed to drive, encoding traffic rules as necessary. This includes information such as: the road boundaries, lane widths and land markings. Note that this does not only correspond directly to physical constraints such as the boundaries and curbs but also ‘soft constraints’ such as road markings and speed limit. The road structure is a logical interpretation of the environment. In environmental perception system this is usually called contextual information.

Stationary obstacles are defined as obstacles which are assumed not to move during the perception period. Dynamic obstacles on the other hand are defined as objects which are moving or potentially moving in the perception period. With this definition, not only the vehicles participating actively in traffic but these temporarily stopped or occluded by other road users are considered as dynamic obstacles. The overall architecture of environmental perception system is shown in Figure 2.2.

As shown in Figure 2.2, target tracking is an integral part of larger environmental perception system, which is responsible for identification of dynamic obstacle hypotheses. The process of target detection and tracking is assisted by the contextual information generated from both stationary obstacle estimation and road estimation.

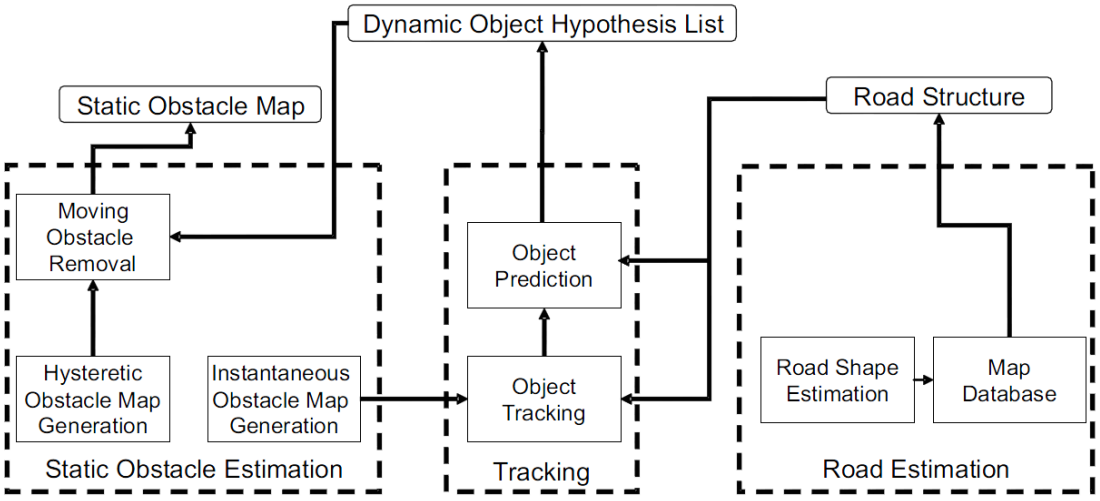


Figure 2.2 Environmental perception system [6]

Target tracking is of great importance in situational awareness system. Knowledge about the status of moving objects such as position, velocity and acceleration can be used to improve the perception of the surrounding environment. Furthermore, from the autonomous vehicle development perspective, target tracking can even determine the automation level [15]. For example, this ability allows an autonomous vehicle to improve its collision avoidance behaviour in populated environment or high speed maneuver. This thesis focus on developing an autonomous vehicle target tracking strategy.

The main difference from previous target tracking algorithms is the contextual information is implemented to assist target tracking taking under consideration of environmental uncertainty. The proposed methods incorporates different types of contextual information in a comprehensive way considering both environmental physical constraints and interaction behaviour between targets and the environment, the details are explained in the following Chapters 3, 4 and 5.

2.2 Background for Target Tracking Problem

The problem of estimating the information of objects movement state (including position, velocity or acceleration) is very important in autonomous vehicles. This problem can be basically summarized as target tracking issue. Knowledge about the state of moving objects can be taken as powerful information to improve the autonomy for autonomous vehicles especially in urban environments.

Moving target tracking was founded on Kalman filtering algorithm in the early 1960s [8]. Since then, the relative approaches have been widely used in various military and civilian applications such as aerospace navigation, air traffic control, sensor networks, biomedical research and environment monitoring [9-12] in the last thirty decades. Recently due to the rapid growing in the field of autonomous vehicles and its relevant ADAS, researchers and companies start showing great interest in automotive MTT area [13-15]. These intelligent features have advantages of significantly increasing road safety and improving the quality and efficiency of people and transportation. Besides, these intelligent features and autonomous functionalities on vehicles can also bring major economic benefits from reduced fuel consumption, efficient exploitation of the road network, and reduced personnel [16].

2.2.1 Autonomous Vehicle and Target Tracking

ADAS's have become a widespread class of automotive applications in nowadays commercial vehicles. They lend even more confidence to driving and improve road safety in complicated and challenging driving conditions (e.g. at night or bad weather conditions, crowed urban environments, high speed motorway maneuver). By using state-of-the-art onboard sensors such like radar, LIDAR, GPS and camera vision systems together with accurate online global map, ADAS extend extend the sensed information beyond the ego vehicle state to environmental information. This enables a wide field of latest ADAS functions, such as, e.g., lane departure warning (LDW), parking assist (PA) based on sonar, radar, or video. The adaptive cruise control (ACC) automatically adjusts velocity to keep a comfortable distance to predeceasing vehicles. In order to mitigate collision hazards, automated emergency braking (AEB) engages a strong braking when an immanent and inevitable collision is detected. Due to uncertainties in the processing chain, so far this action only works with low speed environment (e.g. $\leq$35mph) [17] and mainly in vehicle following scenarios. The main aim for AEB at the moment is to reduce the kinetic energy of the impact but not to completely avoid the collision. To achieve a more safety collision avoidance driving, more advanced AEBs such as cyclist AEB [18], pedestrian AEB [19], junction AEB [20] are now under developing. An illustration of different ADAS based on different sensing technologies is shown in Figure 2.3.

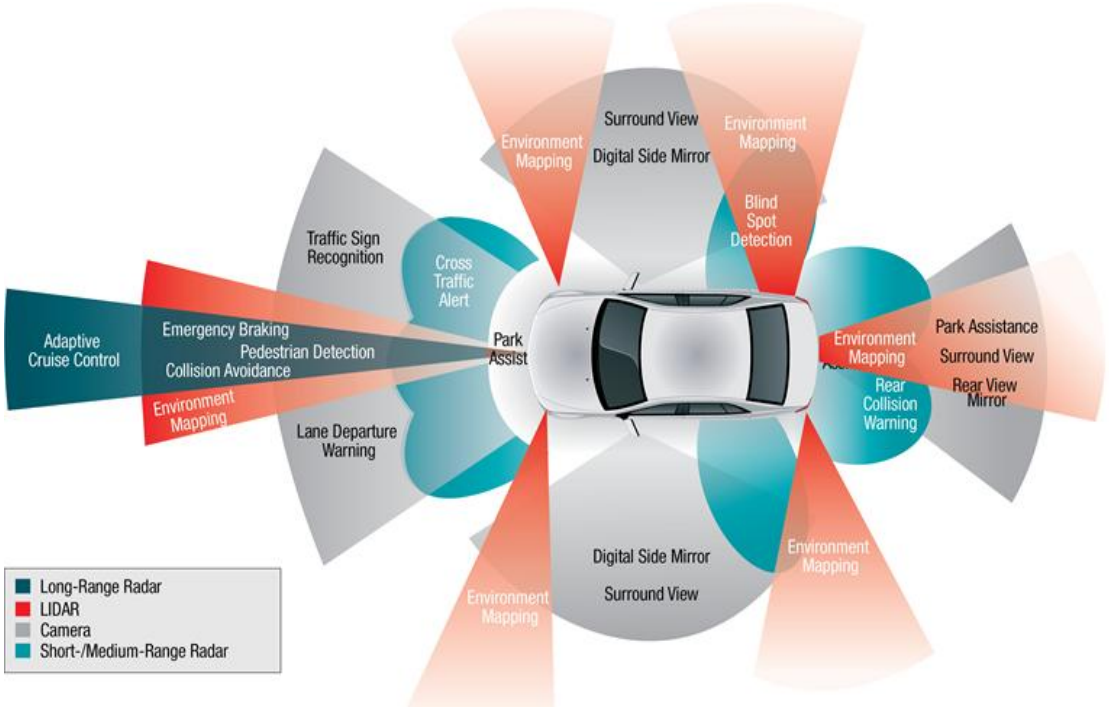


Figure 2.3 Advanced driver assistance systems [21]

Despite impressive recent merits in research in this field, the uncertainty of environment information is far too high as to allow fully autonomous driving in the near future. So far the majority of the ADAS functions spectrum is restricted to information, warning, and comfort enhancement, while the final responsibility stays with in driver. Corresponding to the five levels of automation [22] shown in Figure 2.4, so far the current ADAS have achieved Level 2 driver assistance and moving from Level 3 partial automation towards Level 4 conditional automation in the next couple of years. The aim is to achieve a high automation level around 2025 and eventually the full driverless automation under complex scenarios in the future.

In order to improve the autonomy level for autonomous vehicles, it is essential to have a better understanding of driving scenarios/environments. This would lead to much more complete sensing and perception requirements. The enhancement of reliability and certainty of the perception system is of prevailing importance. Since detection and tracking moving objects is especial important in environmental perception, recently more research are started on a MTT based ADAS feature which aims for an early warning and collision avoidance system [23] [24]. This includes the usage of multisensory and multisystem capabilities [25], extended usage of data-fusion [26], improvement of sensor capability [27], usage of a new set of information

source (digital map, v2X, eHorizon, infrastructure data) [28] and smarter algorithms and interoperability [29].

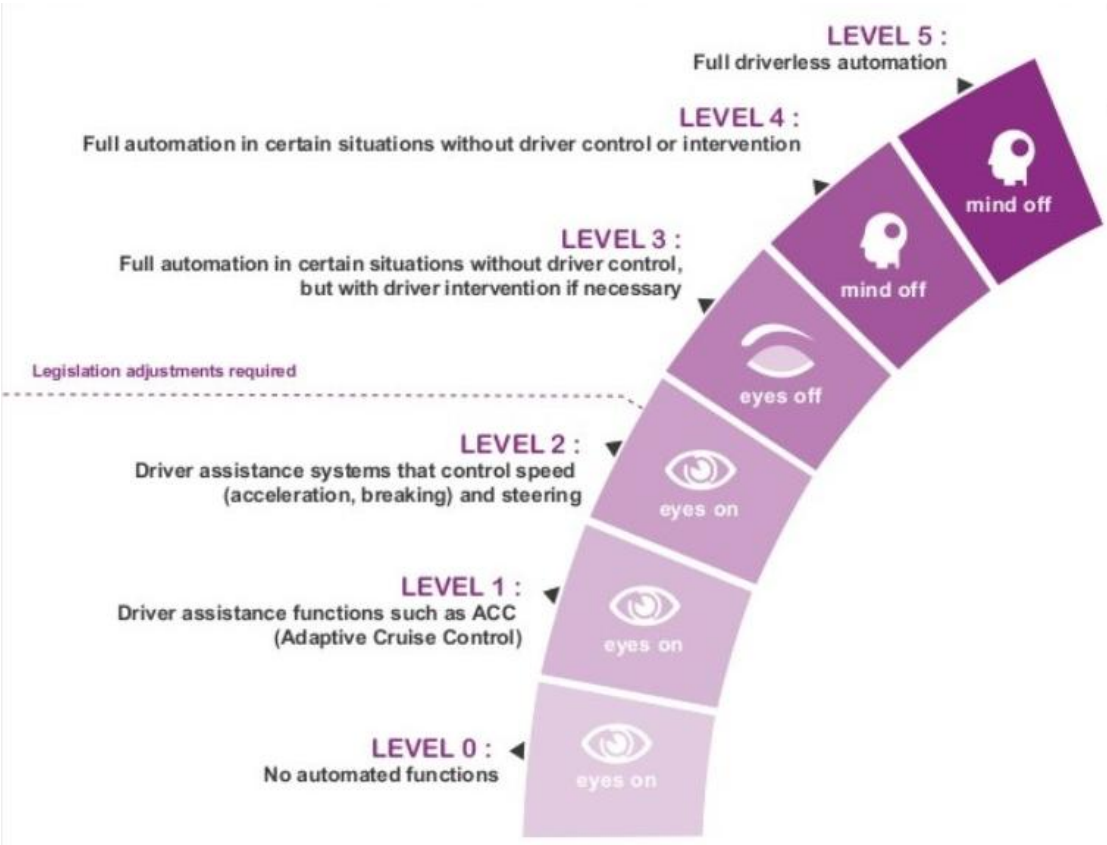


Figure 2.4 Levels of autonomous vehicle development [22]

A number of algorithms have been studied and proposed for target tracking problems. The essence of target tracking problems can be defined as finding tracks (states of targets) from a sequence of noisy measurements [15]. Based on the complicity of tacking situations, the target tracking problem is divided into two main different classes: single target tracking and multiple target tracking (MTT). MTT techniques are theoretically and fundamentally different from single target tracking techniques. In single target tracking, only one target state is modelled, which makes all the other detected objects not considered or updated with the state of the target. Unlike single target tracking, inherent data association problems arise in MTT which makes the algorithms capable of tracking closely-spaced and even crossing multiple targets with tracking occlusion. The process of data association is to partition and assign the observed data into different sets of observations or tracks that are produced

by the same object, in other words it's the process of associating uncertain measurements to known tracks [26]. The details of the differences between single target tracking and MTT are discussed in the following part of Chapter 1.

2.2.2 Target Dynamic Models

In target tracking problems, the target is represented by a state vector which usually includes target position, velocity and/or acceleration. An example of a state vector is shown below:

$$\mathrm{x}_k = [x_k, \dot{x}_k, y_k, \dot{y}_k]^T \tag{2.1}$$

where x_k and y_k are the position along x and y axis respectively in Cartesian coordinate and $\dot{x}_k$ and $\dot{y}_k$ are the respective velocities.

A target Dynamic model is used to describe the evolution of the target state with respect to time. Surveys of full different dynamic models that are used for tracking are presented in [30]. In this thesis, only some of the most commonly used models are considered such as constant velocity model (CV), constant acceleration model (CA) and coordinated turn model (CT). The detail explanation of each of the models is shown below.

According to the state vector (2.1) shown above, the CV model process equation can be written as:

$$\mathrm{x}_{k+1} = F\mathrm{x}_k + G\omega_k \tag{2.2}$$

where

$$F = \begin{bmatrix} 1 & T & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & T \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.3}$$

$$G = \begin{bmatrix} \frac{1}{2}T^2 & 0 \\ T & 0 \\ 0 & \frac{1}{2}T^2 \\ 0 & T \end{bmatrix} \tag{2.4}$$

T is defined as the sample period. In tracking problems, the acceleration is usually considered as noise added in system model. In this case, $\omega_k = [a_x, a_y]^T$ here is the process noise which contains acceleration in the x and y axis respectively and G

represents the input matrix (acceleration matrix). ω_k is defined as a zero mean white noise with covariance matrix Q_{CV}.

Next is the CA model. The state vector now includes the position, velocity and acceleration:

$$\mathrm{x}_k = [x_k, \dot{x}_k, \ddot{x}_k, y_k, \dot{y}_k, \ddot{y}_k]^T \tag{2.5}$$

where $\ddot{x}_k$ and $\ddot{y}_k$ are the acceleration in x and y axis.

Similar to CV model, CA model is linear and the process equation is the same as (2.2) with different state transition and process noise matrix:

$$F = \begin{bmatrix} 1 & T & \frac{1}{2}T^2 & 0 & 0 & 0 \\ 0 & 1 & T & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & T & \frac{1}{2}T^2 \\ 0 & 0 & 0 & 0 & 1 & T \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.6}$$

$$G = \begin{bmatrix} \frac{1}{2}T^2 & T & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{2}T^2 & T & 1 \end{bmatrix}^T \tag{2.7}$$

The process noise $\omega_k = [a_x, a_y]^T$ is the same variable shown in CV model with different covariance Q_{CA}

The third model is CT model considering the following state vector which contains position, velocity, acceleration and turn rate:

$$\mathrm{x}_k = [x_k, \dot{x}_k, y_k, \dot{y}_k, \omega]^T \tag{2.8}$$

where ω stands for the turn rate at time k which is assumed as a known parameter.

The CT model can be described by the following equation:

$$\mathrm{x}_{k+1} = F(\omega)\mathrm{x}_k + \eta_k \tag{2.9}$$

where
$$F(\omega) = \begin{bmatrix} 1 & \frac{\sin(\omega_k T)}{\omega_k} & 0 & \frac{1-\cos(\omega_k T)}{\omega_k} \\ 0 & \cos(\omega_k T) & 0 & -\sin(\omega_k T) \\ 0 & \frac{1-\cos(\omega_k T)}{\omega_k} & 1 & \frac{\sin(\omega_k T)}{\omega_k} \\ 0 & \sin(\omega_k T) & 0 & \cos(\omega_k T) \end{bmatrix} \tag{2.10}$$

The zero-mean (Gaussian) white noise η_k is used to model the perturbation of the trajectory from the ideal CT model in both x and y axis.

2.2.3 Measurement Models

For target tracking systems, measurement model is also a compulsory part. These measurements are related to the state vector via the measurement function which is shown in (2.11) for the most general form:

$$\mathrm{z}_k = h(\mathrm{x}_k) + v_k \tag{2.11}$$

where z_k is the source of detected measurements at time k and v_k is called the measurement noise that is caused by impreciseness of the sensor or human error and other environmental factors. It is assumed to be normal (Gaussian) probability distribution $v_k \sim \mathcal{N}(v_k; 0, R)$, with zero mean and covariance denoted as R. The measurement function h varies with different tracking systems from linear to nonlinear equations. It is always assumed to be a known function which reflects the relation between the target state vectors x_k and the detected measurements z_k. For example, a sensor may receive observation in a local Cartesian coordinate (such as target state position in along x and y axis) and in this case the h would be a linear equation; or observation in a polar coordinate system (such as range and bearing based radar) which is described as a nonlinear equation. The details of different measurement models are shown in the following chapters in this thesis.

2.3 Single Target Tracking State Estimation (Filter) Approaches

The primary concern of target tracking is to use state estimation tools of determining the position and velocity of an object in realistic environments. State estimation theory is used to solve the problem of recovering unobserved state variables from original measurement.

2.3.1 Bayesian State Estimation

The most general state estimation algorithm is given by the Bayesian algorithm. The algorithm calculates the belief distribution from measurement and prior state data. Consider the discrete state-space system:

$$x_{k+1} = f(x_k) + \omega_k \tag{2.12}$$

$$z_k = h(x_k) + v_k \tag{2.13}$$

where x_k is the state with initial state x_0 and its distribution $p(x_0)$; z_k is the measurement; ω_k is the white process noise with a known distribution $p(\omega)$ independent from x_k; v_k is the white measurement noise with a known distribution $p(v)$ independent from x_k. The aim is to use these given prior information to find the posterior density of the state $p(x_k|z_{1:k})$. The process noise represents the lack of knowledge about the system dynamics. The larger the process noise, the smaller will be the trust on the state equation. The measurement noise represents the imperfections in acquiring the data. The larger the measurement noise, the smaller the trust will be on the measurements. Basic probability theory gives a recursive solution in the form:

$$p(x_{k-1}|z_{1:k-1}) \xrightarrow{\text{prediction}} p(x_k|z_{1:k-1}) \xrightarrow{\text{update}} p(x_k|z_{1:k}) \tag{2.14}$$

The Bayesian algorithm is recursive, that is, the belief $p(x_k|z_{1:k})$ at time k (current step) is calculated from the belief $p(x_{k-1}|z_{1:k-1})$ at time k-1. The recursive process can be represented below in a 'for' loop shown on the top of next paper:

- Start with $p(x_0)$, at k = 1.
- For each k
- Prediction Update

$$p(x_{k-1}|z_{1:k-1}) = \int p(x_k|x_{k-1})p(x_{k-1}|z_{1:k-1})\, dx_{k-1} \tag{2.15}$$

- Measurement Update

$$p(x_k|z_{1:k}) = \frac{p(z_k|x_k)p(x_k|z_{1:k-1})}{p(z_k|z_{1:k-1})} \tag{2.16}$$

where $p(z_k|z_{1:k-1}) = \int p(z_k|x_k)p(x_k|z_{1:k-1})\, dx_k$ is constant with respect to x_k

- End for
- Let k=k+1 and repeat the process recursively.

The process includes two essential steps. First, prediction update gives a result of predicted probability distribution $p(\mathrm{x}_k|\mathrm{z}_{1:k-1})$ which is obtained by integrating two beliefs: the prior state probability distribution $p(\mathrm{x}_{k-1}|\mathrm{z}_{1:k-1})$ and the state transition probability distribution $p(\mathrm{x}_k|\mathrm{x}_{k-1})$. The nature of prediction update is actually an implementation of the total probability theory $p(\mathrm{x}) = \int p(\mathrm{x}|\mathrm{z})p(\mathrm{z})\,d\mathrm{y}$. Then the next step is called measurement update, which implements the basic Bayes rule $p(\mathrm{x}) = \frac{p(\mathrm{x}|\mathrm{z})p(\mathrm{x})}{p(\mathrm{z})}$. In this step, the predicted state probability $p(\mathrm{x}_k|\mathrm{z}_{1:k\cdot 1})$ is multiplied by the measurement likelihood function $p(\mathrm{z}_k|\mathrm{x}_k)$, which is a belief of measurement z_k been observed. Finally the posterior (estimated) state probability at time k is given as $p(\mathrm{x}_k|\mathrm{z}_{1:k-1})$.

2.3.2 Kalman Filter

Kalman filter (KF) is a one of the most studied technique for implementing Bayes filters which has been widely applied in linear Gaussian systems. If the system and measurement process are linear with Gaussian noise, different from the basic Bayes filter, KF has a benefit of representing state probability by only two sufficient statistics: the estimated mean $\hat{\mathrm{x}}_{k|k} = E[\mathrm{x}_k|\mathrm{z}_{1:k}]$ and covariance $P_{k|k} = E\left[(\mathrm{x}_k - \hat{\mathrm{x}}_{k|k})\left(\mathrm{x}_k - \hat{\mathrm{x}}_{k|k}\right)^T \middle| \mathrm{z}_{1:k}\right]$ at each time step resulting in a Gaussian distributed system posterior density $p(\mathrm{x}_k|\mathrm{z}_{1:k})$. In other words, instead of propagating densities (2.14) only mean and covariance are considered:

$$\hat{\mathrm{x}}_{k-1|k-1}, P_{k-1|k-1} \xrightarrow{\text{prediction}} \hat{\mathrm{x}}_{k|k-1}, P_{k|k-1} \xrightarrow{\text{update}} \hat{\mathrm{x}}_{k|k}, P_{k|k} \qquad (2.17)$$

As a result, the infinite dimensional estimation problem reduces to finite dimensional estimation problem.

The KF operation estimates a process by using a form of feedback loop: the filter estimates the process state at some time and then obtains feedback from noisy measurements. In this case, the operation is based on two steps: a prediction and an update step. The prediction step is responsible for projecting forward (in time) the current state and error covariance estimates to obtain the *a priori* estimates for the next time step. Then the measurement update equations, also known as correction step, are responsible for incorporating a new measurement into the *a priori* estimate to obtain an adjusted *a posteriori* estimation of mean and covariance.

The specific equations prediction and measurement updates are presented below:

- Start with $\hat{\mathrm{x}}_{0|0}$, $P_{0|0}$, at *k*=1.
- For each k:
- Prediction Update

$$\hat{\mathrm{x}}_{k|k-1} = F\hat{\mathrm{x}}_{k-1|k-1} \tag{2.18}$$

$$P_{k|k-1} = FP_{k-1|k-1}F^T + Q \tag{2.19}$$

- Measurement Update

$$\hat{\mathrm{x}}_{k|k} = \hat{\mathrm{x}}_{k|k-1} + K_k(\mathrm{z}_k - \hat{\mathrm{z}}_{k|k-1}) \tag{2.20}$$

$$P_{k|k} = P_{k|k-1} - K_k S_{k|k-1} {K_k}^T \tag{2.21}$$

where $$\hat{\mathrm{z}}_{k|k-1} = H\hat{\mathrm{x}}_{k|k-1} \tag{2.22}$$

$$S_{k|k-1} = HP_{k|k-1}H^T + R \tag{2.23}$$

$$K_k = P_{k|k-1}H^T {S^{-1}}_{k|k-1} \tag{2.24}$$

where $\hat{\mathrm{x}}_{k|k-1}$ is the predicted state and $P_{k|k-1}$ is the covariance of the predicted state; $\hat{\mathrm{x}}_{k|k}$ is the estimated state and $P_{k|k}$ is the covariance of the estimated state. $\hat{\mathrm{z}}_{k|k-1}$ is the predicted measurement; $\mathrm{z}_k - \hat{\mathrm{z}}_{k|k-1}$ is the measurement error/innovation; $S_{k|k-1}$ is the covariance of the predicted measurements/ innovation covariance. K_k is the Kalman gain.

The first task during the measurement update is to compute the Kalman gain K_k. The Kalman gain serves as a 'weighted compensator', if *R* is large then K_k will be small and therefore the measurement will be given less weight. Similarly, if the measurement noise is low, R will be small and hence the Kalman Gain will be large, therefore giving more importance to the measurement innovation $\mathrm{z}_k - \hat{\mathrm{z}}_{k|k-1}$. After each time the measurement update pair, the process is repeated with the previous *a posteriori* estimates used to project or predict the new *a priori* estimates.

2.3.3 Nonlinear State Estimation (Filters)

The crucial for the correctness of KF is that the estimated state comes from a linear Gaussian system. However, the state transitions and measurement relationship to the process are rarely linear in practice. E.g. the target moves with constant translational and rotational velocity but on a circular trajectory, which cannot be represented by linear state transitions. This problem is handled by linearized KFs, such as extended Kalman filter (EKF) [31] and Unscented Kalman filter (UKF) under a Gaussian noise assumption [32]. Other methods, such as sequential Monte Carlo methods known as Particle filters (PF) are used to deal with nonlinear systems under non-Gaussian noise assumption [33]. The rest of this section will discuss EKF, UKF and PF in details.

Extended Kalman Filter

As described above, the KF addresses the general problem of estimating the state of a linear system with Gaussian noise. This is crucial for the correctness of KF. However, state transitions and measurement relationship to the process are rarely linear in practice.

The EKF is a possible solution to implement a recursive nonlinear estimation filter [31]. EKF relaxes one of the assumptions shown above, the linearity assumption, so that the state transitions probability and the measurement probabilities can be governed by nonlinear function f and h. The essence of EKF is to linearize about the current mean and covariance of the process and measurement functions so as to calculate a Gaussian approximation to the true belief. EKFs utilize a method called first order Taylor expansion to constructs a linear approximation to function f and h about the current estimate of the state. The matrix of Jacobian partial derivatives for f and h are calculated separately and then applied for calculating the error covariance.

The most important factor in EKF is the linear approximation to the non-linear functions. Besides, the EKF is very similar to the standard KF in prediction and update steps which makes it easy in implementation. However, if the linearization is poor, estimation biases and divergence is expected resulting in bad filter performance. In order to achieve better estimation performance especially for highly nonlinear systems, more advanced nonlinear filter techniques such as UKF and PFs are required.

Unscented Kalman Filter

In linearization problems, if the uncertainty is small in the variable to be transformed, the EKF usually can provide good performance. However, as the uncertainty grows, the performance of EKF may degrade and sometimes lead to terrible results. This is because EKF (based on Taylor expansion) cares only about the information of transformation around the linearization point. Hence it only works well locally. When the uncertainty grows, the error propagation of linear transformations cannot be well approximated by a linear or quadratic function which will result in an extremely poor performance [34].

The Unscented Kalman filter (UKF) performs a stochastic linearization through the use of a weighted statistical linear regression process. Instead of approximating the nonlinear function f by a (first order) Taylor expansion; the UKF unscented transformation is based on using a number of 'sigma points' to represent the original Gaussian density [35]. The sigma points are transformed with the nonlinear transformation f and are generally located at the mean and symmetrically along the main axes of the covariance. For an n-dimensional Gaussian with mean μ and covariance Σ, there are 2n+1 sigma points χ with four parameters: mean $\hat{\mathrm{x}}_{k-1|k-1}$ and covariance $P_{k-1|k-1}$, mean weight ω_m and covariance weight ω_c .

In UKF, the target state posterior density and covariance are reconstructed from the sampling sigma points and assumed to be Gaussian. The main advantage of the UKF over the EKF is that instead of calculating the Jacobians linearization based on one specific state point, a more accurate estimate of the exact man and covariance can be obtained by simply increasing the number of sigma points via a wide variety of processes.

Particle Filter

Particle filtering (PF) is an alternative nonparametric implementation of the Bayes filter. Unlike KF based filters which summarise the target density using a mean and covariance, the PF describes the density directly using a finite number of randomly sampled points and weights. In some sense, a PF can be described as a generalization of UKF to random particles instead of sigma-points. This gives PF the advantages of solving state estimation for nonlinear and non-Gaussian systems.

The main idea of PF is to approximate the posterior $p(\mathrm{x}_k|\mathrm{z}_{1:k})$ as:

$$p(\mathrm{x}_k|\mathrm{z}_{1:k}) \approx \sum_{i=1}^{N} \pi_{k|k}^{i} \delta(\mathrm{x}_k) \tag{2.25}$$

where some state values $\{\mathrm{x}_{k|k}^{i}\}_{i=1}^{N}$ called particles and weights $\{\pi_{k|k}^{i}\}_{i=1}^{N}$ are used to characterise the posterior density $p(\mathrm{x}_k|\mathrm{y}_{1:k})$. N is the number of support points with the weights are normalised such that $\sum_i \pi_{k|k}^{i} = 1$.

In this case, instead of propagating state densities as (2.17), a PF propagates only the particles and weights

$$\{\pi_{k-1|k-1}^{i}, \mathrm{x}_{k-1|k-1}^{i}\}_{i=1}^{N} \xrightarrow{\text{prediction}} \{\pi_{k|k-1}^{i}, \mathrm{x}_{k|k-1}^{i}\}_{i=1}^{N} \xrightarrow{\text{update}} \{\pi_{k|k}^{i}, \mathrm{x}_{k|k}^{i}\}_{i=1}^{N} \tag{2.26}$$

It can be shown that as the number of particles increase $N \to \infty$, the approximation (2.25) approaches the true posterior density $p(\mathrm{x}_k|\mathrm{z}_{1:k})$. The PF algorithm follows a similar prediction and update step as other filters described above. The filter starts with a set of particle points and weights, which describes the posterior density at time $k-1$ as $p(\mathrm{x}_{k-1}|\mathrm{z}_{1:k-1})$. The particles are then propagated using the mean proposal distribution to form a new set of particles at time k, which is the same the prediction step. The weights at time k are a function of the process dynamics and measurement likelihood function. This new set of particles and weights $\{\pi_{k|k}^{i}, \mathrm{x}_{k|k}^{i}\}_{i=1}^{N}$ gives an approximation for the posterior density at time k, which is the update step.

One main problem with PF is known as degeneracy problem which is the situation where one specific particle dominates and the weights of the others are negligible. This is guaranteed to occur after a finite number of filter recursions [33]. In order to solve this problem, resampling technique is used which removes particles with negligible weights and replicates the particles with high weights. As a result, particle weights become all equal at the end of resampling. Another main disadvantage of PF is its high computational requirement. Due to high computational cost, PF are currently limited to solve smaller dimensional state space estimation problems.

2.3.4 Moving Horizon Estimation

In this section, a constrained state estimation method is introduced which is different from Kalman filter based approaches: the moving horizon estimation (MHE). MHE, which reformulates the estimation problem as quadratic programming over a moving

fixed-size estimation window, becomes an important approach to constrained nonlinear estimation [36]. In MHE, the basic strategy is to consider explicitly a fixed amount of data. In this case, the state estimate is determined online by solving a finite horizon state estimation problem. As new measurements arrive, the old measurements are discarded from the estimation window but approximately summarized by the estimator [37]. Mathematical optimization strategies are required and essential for MHE to handle explicitly nonlinear and constraints on the system. The name of 'moving horizon' arises from the visual process of using a sliding estimation window as shown in Figure 2.5.

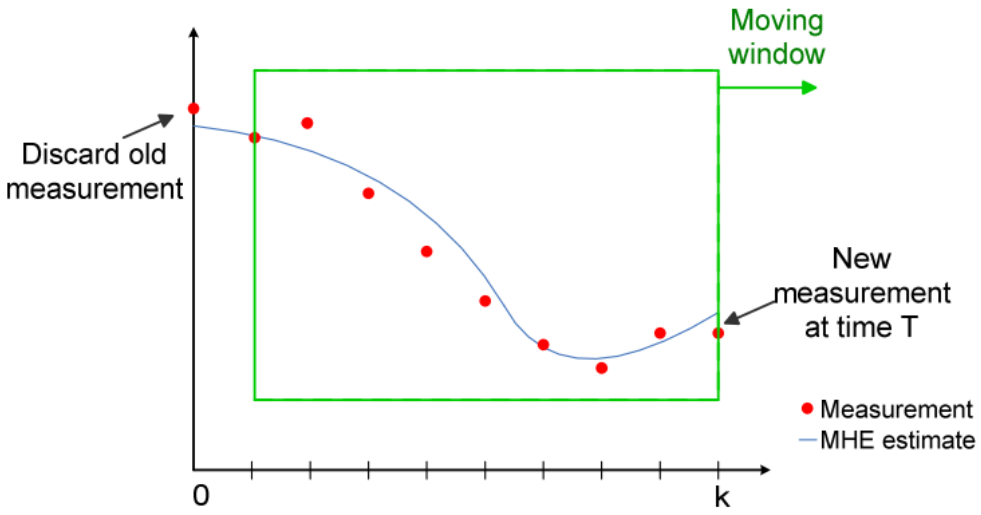


Figure 2.5 Illustration of moving horizon estimation (MHE) for one time step

Different from other methods, MHE has a great advantage of capable to handle different forms of constraints naturally in its framework by reconstructing the mode of the posteriori distribution via constrained optimisation [38]. In addition to state constraints, such constraints can also be placed on the state process and/or measurement noise of a linear/nonlinear system which are typically used to model bounded disturbance or random variables with truncated distribution/densities. The interest on MHE was originally motivated by its robustness, which makes the approach suitable for solving modelling uncertainties and numerical errors [39]. Recently, MHE has been widely applied in chemical process [40], fault detection [41, 42], system identification [43] and process control. The technique has also been implemented in linear systems [36, 44], hybrid systems [45] and more recently in

nonlinear system with constraints explicitly [37]. However, literature shows that MHE has been rarely implemented in target tracking problems especially for autonomous vehicle applications due to its relatively high computational load .

For this very reason, this research is mainly focus on how to adopt constrained MHE as an alternatively efficient state estimation technique into target tracking process so as to improve the target tracking performance even with nonlinear system and additional state constraint. The details of MHE mathematical explanation and relevant adaption for solving target tracking problems will be explained in Chapter 4.

2.4 Multiple Target Tracking --- Data Association Approaches

In MTT, useful information containing one or more (potential) targets is collected from raw data by sensor data processing and measurement formation. Then the (new) tracks are formed based on the existing tracks and a new set of input measurements. However in practical applications, false alarms or clutter are also presented in the original measurement which makes the relation between which measurement corresponds to the target of interest not clear, especially when multiple targets are presented. In this case it is necessary to partition and assign the observed data into different sets of observations or tracks that are produced by the same object. This process is known as data association [34], which is also the core of MTT problems. Once observations are assigned to tracks, the states of the tracks are updated and predicted using state estimation algorithms, such as the well-known Kalman filter. The basic structure of MTT is shown below in Figure 2.6. The details of each of the block will be discussed in the following of this chapter.

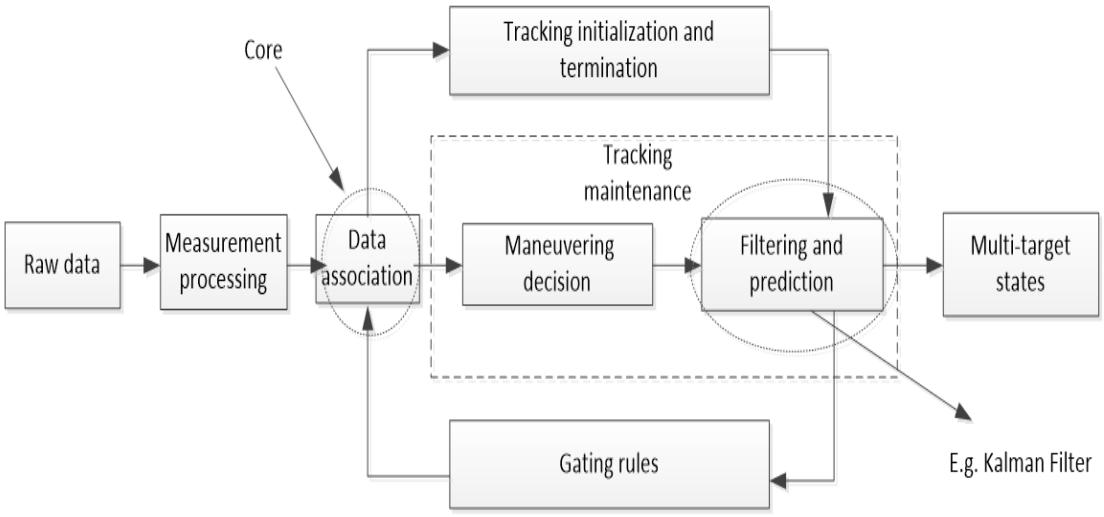


Figure 2.6 Structure of multiple target tracking algorithms

Before moving to MTT issues, a series basic concepts and structures are worth to be covered first. Target tracking is basically to use the state estimation tools in realistic environments. The outline of target tracking includes three types of data: measurements, targets states, trajectory; and four main modules: tracking initialization/formation, tracking maintenance (maneuvering decision, filtering, and prediction), tracking termination, data association and state estimation, gating.

2.4.1 Basic Concepts in Multiple Target Tracking

Measurements

During the first observation stage, the term measurement contains all quantities included in a (processed) output from the raw data of a sensor. Such measurements may include velocity, distance, bearing angle or range. Because the data received by sensors is affected by measurement noise and other forms of interference, there is a need of pre-process of raw data. Inherent uncertainties are always with measurements in target tracking problems. Such as false measurements (false alarm probability $P_{FA} > 0$) and missed measurements (the detection probability $P_D < 1$). Thus each measurement has three types of sources including detected target, a new target or false alarm (Clutter). False alarm (Clutter) refers to erroneous detection events such as those caused by random noise or clutter that do not persist over several scans.

Data Association

As there are multiple targets considered in the sensor's field of view, all the confirmed measurements need to be associated with existing objects. This process is also known as data association (assignment process) which is technically the core of MTT process. Examples of data association algorithms including: Nearest neighbours (NN), Global nearest neighbours (GNN), Joint probabilistic data associations (JPDA) and Multiple hypothesis tracking (MHT). The details of data association techniques will be discussed in the next section in this chapter.

Gating

In the filter and prediction stage, the location of each target is predicted in the subsequent measurement step using filtering and state estimation algorithm. This prediction is based on an estimate of the location data computed by filter under relative target model from previous iteration and the actual location data measured during current observation stage. The difference between the predicted information of each target and the actual current measurement is known as innovation vector which can be calculated based on different mathematic equations, such as by *Mahalanobis* distance [46]. The results of this step are needed by the following gating process. Only the measurements within the gating region are considered for update of the track. The shape of the gate varies in different algorithms such as rectangular, circle and ellipsoid. The most common choice is ellipsoidal gate which is defined by a probability contour obtained when intersecting a Gaussian with a hyper-plane (ellipsoid). The gate checking is needed before data association so as to minimize the measurements candidates and the number of possible combinations.

Targets

In practical applications, vehicles and objects may leave or enter the sensor's field of view at any time which makes the tracking initialization and termination essential in MTT structure. The target candidate is confirmed or deleted only if it is detected or missed for a specific number of iterations. The threshold is varied for different situations and usually obtained empirically.

2.4.2 Data Association Algorithms

In this chapter, some concept and basic data association algorithms are discussed. Several approaches for MTT have been developed over the last decades, overviews can be found in Pulford [15] and Christoph [14]. Basically, these methods can be divided into two categories – the data association based 'classic' methods and the more recent finit set statistics (FISST) based approaches. In this thesis only the data association based MTT algorithms are considered. GNN, JPDA and MHT are the most widely used data association techniques.

The basic and simpliest solution is the nearest neighbour standard filter (NNSF) [47]. This techinique associates each target with only the closest measurement in statistiacal distance. Because of this feature, such a method is also known as making 'hard dicision'. However, this simple procedure prunes away many feasible hypotheses. To overcome this problem, a combination of observation associations is made in the GNN approach [48] and the dicision is made only based on the most likely combination. In GNN, an essential association matric concept is introduced to 'score' the distance between all measurements and all targets, resulting in a matrix of values. The combination of associations with the highest score is then chosen, taking into account the fact that a single target may only result in one observation in each combination. Combinatorial optimization methods are required in GNN to solve this optimal assignment problem. In standard GNN method, the current measurements are associated to existing tracks with only the most likely association hypothesis at each step. Only one hypothesis (most likely assignment combination) is considered for existing track update and new track initiation. Because of this inherent property, GNN only works well in the case of widely spaced targets, accurate measurements, and few alarms in the track gates [48].

Rather than making 'hard decisions', a suboptimal Bayesian approach which is based on mimimum mean square error (MMSE) is designed, known as JPDA [49]. JPDA makes soft decision equivalent of GNN. All the possible association conditions are now considered by allowing a track to be updated by a weighted sum of all observations in its gate. This means that each measurement may contribute to more than one track which makes it different from GNN. In this approach, all measurements that are close to the predicte target location are considered in data association. Each observation is essentially weighted in a probabilistic way based on

the maginitude of deviation from the predicted location and then the Bayesian measurement update step is performed using all these weighted measurements in a Gaussian mixture pdf form. The main shortcoming of JPDA filter is that the final estimate is collapsed to a single Gaussian, thus discarding some pertinent information. The derivations of JPDA such as sample-based JPDA [51] and Monte Carlo JPDA [50], suboptimal/fast JPDA [52], N-best JPDA [53], etc. Some of these subsequent work addressed the JPDA shortcoming by reducing the number of mixture components but many feasible hypothese may be discarded by the pruning mechanisms.

The problems result from standard GNN and JPDA together with the increase in computational capabilities makes MHT a preferred data association method [48]. MHT method form alternative association hypotheses in case of observation to track conflict situations. Rather than choosing the best hypothesis at current step as in GNN and JPDA, MHT keeps a set of multiple hypotheses and thus the assignment ambiguity will be resolved in future when subsequent new observations are arrived. In this case, possible error association could be corrected when more evidences are updated.

2.4.3 Multiple Hypothesis Tracking

MHT methods form alternative association hypotheses in case of observation to track conflict situations and the basic structure of MHT is shown in Figure 2.7.

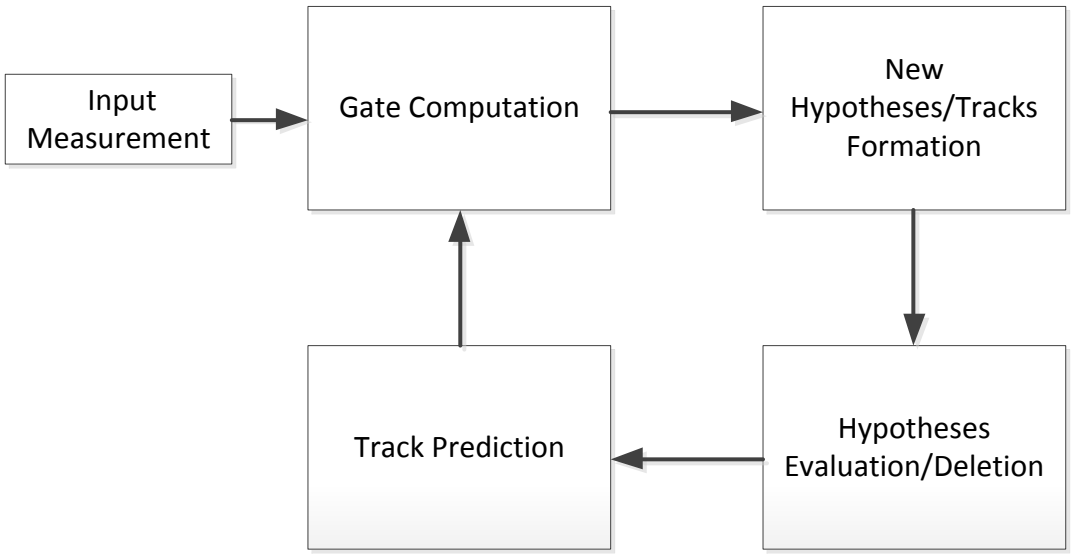


Figure 2.7 MHT algorithm logic overview

Rather than choosing the best hypothesis or combining the hypotheses as in JPDA method at current step, the set of hypotheses is propagated into the future in anticipation that subsequent data will resolve the assignment ambiguities. Unlike other data association algorithms like JPDA, MHT algorithm does not use a separate track initialization procedure and hence track initiation is integrated into the algorithm. Fig 2.8 illustrates how MHT manages these hypotheses using an example.

As shown in Figure 2.8, two tracks T1 and T2 are in predicted positions $\hat{z}^1_{k/k-1}$and $\hat{z}^2_{k/k-1}$. Four measurements are received at the same scan. Assume that the statistical distance between track n and measurement m is d_{nm} and only these pairs with a distance less than gate size are considered as candidate for data association. In this case, the unlikely observation z^1_k is eliminated. The MHT will form different hypotheses by taking into account all the possible sources of a measurement: new track, false alarm and existing track. For example, two very likely hypotheses would both update T1 with z^2_k but update T2 with either z^3_k or z^4_k. Other unlikely but feasible hypotheses would be that all observations represent new targets or false alarms.

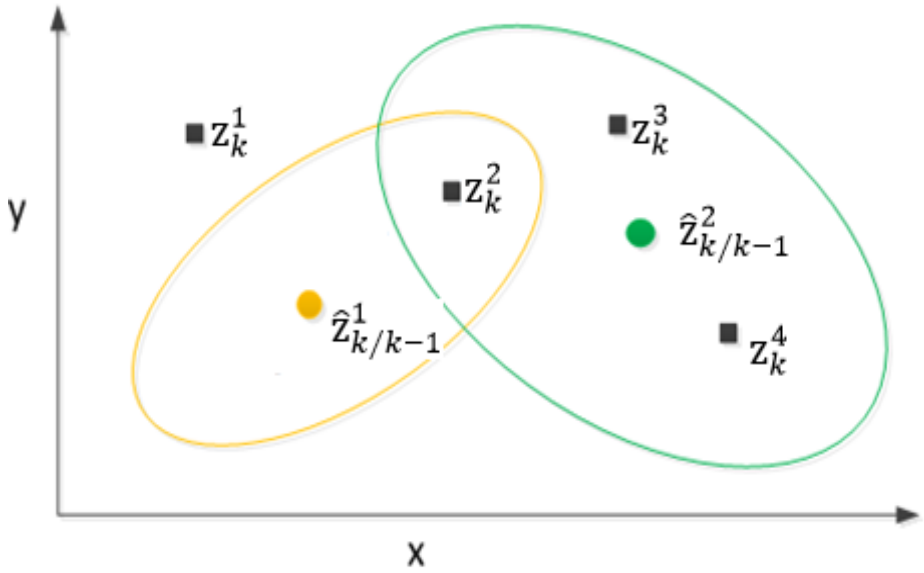


Figure 2.8 Gating procedure for MHT with tracks drawn in circle and measurements in rectangle

Hypothesis based MHT (Reid's Algorithm)

Reid's algorithm is a hypothesis based MHT implementation which keeps the past different hypotheses in the memory between consecutive time steps. When a new

measurement is received, observations that fall within the gate region will set a possible measurement to track assignment and thus an existing hypothesis is expanded to a set of new hypotheses. Each hypothesis contains a set of compatible observation to track assignments, leading to an exhaustive approach of enumerating all the possible assignment combinations. Tracks are defined to be compatible if they have no measurements in common which means each of the new measurement can only be taken to updated with of one of the existing tracks; defined as a new track or a false alarm.

Hypotheses formation:

Assuming N hypotheses are generated from original measurements at time k-1. Each of the hypotheses $\{\theta_{k-1}^{i}\}, i = 1, \dots, N$ is characterized by their assumed number of targets (tracks) and corresponding hypothesis probability $p(\theta_{k-1}^{i})$. The new hypotheses formation is demonstrated in Figure 2.9 (left) using one of the N hypotheses. Assuming a set of three measurements is received at time k. Then an assignment problem based hypotheses tree can be generated, which is shown in Figure 2.9 (right). Regarding to measurement one z_k^1, four hypotheses can be made: creating a false alarm, a new target, updating the existing track one T_1 or track two T_2. The measurement two z_k^2 can only generate three hypotheses, updating target two, creating new target or false alarms while only two hypotheses for measurement three z_k^3: creating a new target or false alarm. The depth of the tree is equal to the number of measurements in the current scan.

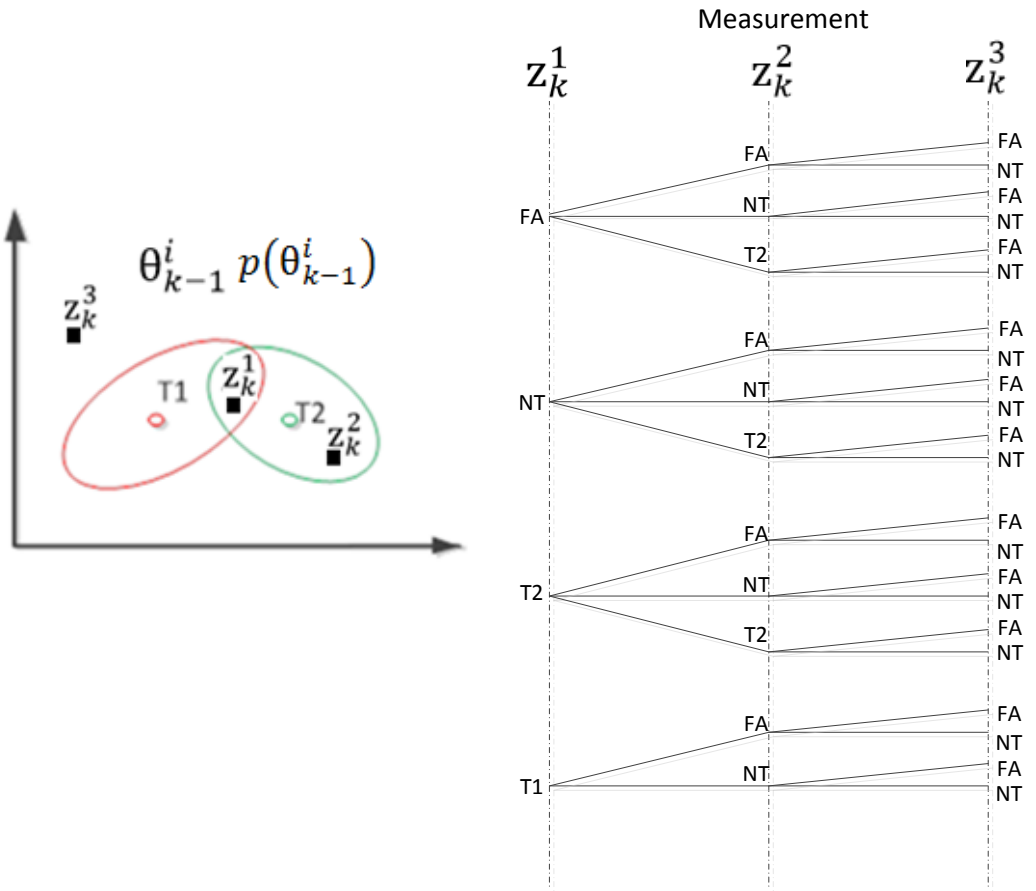


Figure 2.9 MHT new hypothesis formulation

Hypotheses probabilities calculation:

The evaluation of alternative hypotheses formation is based on a probabilistic expression known as the hypotheses probability $p(\theta_{k-1}^{i})$. It includes prior probability of existing target, the false alarms density, the detection sequences and dynamic (kinematic) consistency of the measurements in the tracks.

Assuming at time k (an intermediate stage of tracking), there are $j = 1 \dots n$ targets established previously and $Z_k = \{z_k^1 \dots z_k^m\}$ measurements which are just received. As mentioned above, there are θ_{k-1}^{i}, $i = 1, \dots, N$ different hypotheses about the past kept in the memory between consecutive scans. Let $\theta_k^l \triangleq \{\Theta_k, \theta_{k-1}^{i}\}$, $l = 1, \dots, M$, denotes M posterior hypotheses at current scan k. Each θ_k^l combines a relative past hypothesis θ_{k-1}^{i} with a current generated assignment set Θ_k. (Θ_k is an arbitrary association combination about the current measurement set Z_k) Thus its probability can be represented as:

$$p\left(\theta_k^l|Z_{0:k}\right) \propto \tag{2.27}$$

$$\underbrace{\beta_{FA}^{(m_k^{FA})}\beta_{NT}^{(m_k^{NT})}\left[\prod_{j\in J_D^i}\frac{p_D^j p_{k/k-1}^j\left(Z_k^{\Theta_k^{-1}(j)}\right)}{(1-p_D^j p_G^j)}\right]}_{\textbf{Maximied by Assignment Problem(Murty's)}}\underbrace{C_i p\left(\theta_{k-1}^i|z_{0:k-1}\right)}_{\textbf{Previous Hypothesis probability}}$$

Each parameter in the above equation is explained below:

- m_k^{FA} is the number of false alarms in gate region V at scan k. Thus the probability distribution of m_k^{FA} false alarms in the region V is defined as:

$$p_{FA(m_k)} = \frac{(\beta_{FA}V)^{m_k^{FA}}\exp(-\beta_{FA}V)}{m_k^{FA}!} \tag{2.28}$$

  where β_{FA}(number of FAs/area/scan) is the density of false alarms
- m_k^{NT} is the number of new targets in gate region V at scan k. Thus the probability distribution of m_k^{NT} new targets in the region V is defined as:

$$p_{FA(m_k)} = \frac{(\beta_{NT}V)^{m_k^{NT}}\exp(-\beta_{NT}V)}{m_k^{NT}!} \tag{2.29}$$

  where β_{NT} is the density of new targets is (number of NTs/area/scan)
- The detection probability of the jth target is p_D^j.
- The gate probability of the jth target is p_G^j. (Detected target is in the gate.)
- J_D^i and J_{ND}^i are the set of indices of detected targets and non-detected targets respectively depending on the previous hypothesis $\theta_{k-1}^i, i = 1, \dots, N$.
- $\Theta_k^{-1}(j)$ is the index of the measurements (compressed in the hypothesis) that is assigned to target when $j \in J_D^i$.
- Predicted measurement density (innovation likelihood function) of jth target based on the measurement from $\Theta_k(j)$ hypothesis is:

$$p_{k/k-1}^j\left(Z_k^{\Theta_k^{-1}(j)}\right) \triangleq \mathcal{N}\left(Z_k; \hat{Z}_{k/k-1}, S_{k/k-1}\right) \tag{2.30}$$

  which can be calculated using the normal distribution density function:

$$p(\mathrm{Z}_k|\mathrm{x}_k) = \det(2\pi S_k)^{-\frac{1}{2}} \exp\left(-\frac{1}{2}(\mathrm{Z}_k - H\mathrm{x}_k)^T {S_k}^{-1}(\mathrm{Z}_k - H\mathrm{x}_k)\right) \quad (2.31)$$

- The term $\prod_{j=1}^{n_T^i}(1 - p_D^j p_G^j)$ is a constant value for each hypothesis represented below as C_i:

$$C_i \triangleq \prod_{j=1}^{n_T^i}(1 - p_D^j p_G^j) = \prod_{j\in J_D^i}(1 - p_D^j p_G^j)\prod_{j\in J_{ND}^i}(1 - p_D^j p_G^j) \quad (2.32)$$

The MHT hypothesis probability function (2.27) can then be simplified by taking a logarithm transformation, the result is shown below:

$$\log p\left(\theta_k^l|\mathrm{Z}_{0:k}\right) = m_k^{FA}\log\beta_{FA} + m_k^{NT}\log\beta_{NT} \quad (2.33)$$

$$+ \sum_{j\in J_D}\log\frac{p_D^j p_{k/k-1}^j\left(\mathrm{Z}_k^{\Theta_k^{-1}(j)}\right)}{\left(1 - p_D^j p_G^j\right)} + \log p\left(\theta_{k-1}^i|\mathrm{Z}_{0:k-1}\right)$$

From an implementation perspective, it is easier and practical to represent each hypothesis tree with corresponding statistic assignment probability in a matrix form known as the assignment matrix [58]. Taking the example of the first hypothesis tree θ_{k-1}^1 in Figure 2.9, the assignment matrix is formed as below in Table 2.1.

Table.2.1 Assignment matrix A_{ij}

A_1	T_1	T_2	FA_1	FA_2	FA_3	NT_1	NT_2	NT_3
$\mathbf{z_k^1}$	l_{11}	l_{12}	$\log\beta_{FA}$	$-\infty$	$-\infty$	$\log\beta_{NT}$	$-\infty$	$-\infty$
$\mathbf{z_k^2}$	$-\infty$	l_{21}	$-\infty$	$\log\beta_{FA}$	$-\infty$	$-\infty$	$\log\beta_{NT}$	$-\infty$
$\mathbf{z_k^3}$	$-\infty$	$-\infty$	$-\infty$	$-\infty$	$\log\beta_{FA}$	$-\infty$	$-\infty$	$\log\beta_{NT}$

The assignment matrixA_{ij} has a dimension of $i \times j$ where i represents the number of measurements and j is the total number of potential tracks (including existing

tracks, new targets and false alarms). The variable l_{ij} is equal to $log\frac{p_D^j p_{k|k-1}^j(z_k^i)}{(1-p_D^j p_G^j)}$ which has been derived in (5); β_F and β_N are the density of false alarms and new targets respectively. Now finding the optimal association hypothesis is equivalent to find a corresponding combination of column j (1 to 8) and row i (1 to 3) such that the sum $\sum_{i=1}^{3} A_{ij(j=1\ldots8)}$ is maximized. This is also known as assignment problem [54] in optimization subject. Computation expense is the main issue in assignment problem as the computation cost is extremely heavy when the matrix dimension increases. Some relative algorithms and the corresponding programming tool boxes can be founded such as: linear programming technique (Hungarian method) [54], Munkres algorithm [55], Jonker and Volgenant (JVC) algorithm [56], Auction algorithm [57] and Murty's algorithm [58].

Reducing number of hypotheses:

As suggested in Reid's algorithm [58], instead of generating all possible hypotheses that are possibly deleted later, only the best m hypotheses are generated from each prior hypothesis. Auction algorithm is used to find the m-best assignment solutions from the assignment matrix A_1 by maximum reward (minimum cost).

The key principle of the MHT method is that difficult data association decisions are deferred until more data are received which could be achieved by using N-scan pruning. The structure provides a convenient mechanism for implementing deferred decision logic and for presenting a coherent output from the MHT. As a result, uncertainty at time k-N is resolved by the hypotheses given at time k.

2.5 Other Multiple Target Tracking Approaches

Some nonenumerative approach based MTT algorithms have also been developed such as Probabilistic Multiple Hypotheses Tracker (PMHT) [59]. These methods do not require explicit enumeration of data association hypotheses which leads to an incomplete data association problem that can be efficiently solved using expectation maximization algorithm [60]. However, PMHT is a batch strategy which is not suitable for online applications and the standard version of PMHT is also generally outperformed by JPDA [61].

Particle filters on the other hand have also been implimented in MTT problems to replace EKF for solving nonlinear models especially when the performance of the algorithm degrades as the non-linearities become more severe [62]. The data association problem has also been addressed directly in the context of particle filtering (Sequential Monte Carlo) [63]. The main problem with these MC strategies is that they are iterative in nature and take an unknown number of iterations to converge which makes them not entirely suitable for real-time applications.

Most of the above conventional approaches implicitly assume that some form of thresholding has been applied to raw sensor data in order to reduce the amount of measurement data for processing. In contrast, track-before-detect (TBD) strategies do not apply thresholding and construct a generative model for the raw measurements/state in terms of a multiple target state hypothesis and thus completely avoid an explicit association process [64]. However, such TBD strategies are motivated by highly cluttered sensor data, which can not provide a reasonable detection of object features prior to a valid track. In practial systems, such measurements are not always readily available and may lead to a larger computational complexity. Thereful, TBD techniques have limited applicability comparing with conventional thresholded measurement procedure.

Very recently, a new concept has been introduced in MTT area - the random FISST [65]. While the convential MTT methods try to solve the problem explicitly by expending single target tracking with data association capabilities, the number of targets is also a random variables (random set) in FISST and explicit data association are avoided. The innovation of FISST is to model both the system and measurement as random finite sets (RFSs) and directly apply the Bayes recursion to these set-valued random variables and thus solve the data association problem implicitly. In contrast to explicit data association methods, conventional probability-mass functions are replaced by belief-mass functions. Probability hypothesis density filter (PHD) [66] and multi-target multi-Bernoulli (MeMBer) [67] filter proposed by Mahler have successful implemented the FISST concept into MTT algorithms. Furthermore, the performance of MHT and Gaussian mixture cardinalized probability hypothesis density (GM-PHD) is compared in [68]. The results show that MHT is more stable with lower RMSE while GM-PHD has the advantage of faster response to new /vanishing targets.

II Decision Making

2.6 Path Planning for Autonomous vehicle

This section focus on discussing how to implement situational awareness based information to assist decision making for autonomous vehicle. As shown in Figure 1.1 in Chapter 1, while perception refers to understanding its measurement through appropriate sensors and tracking moving objects, finding the vehicle's pose or configuration in the surrounding is localization and map building. Planning the path in accordance with the task by using cognitive decision making is an essential phase before actually accomplishing the preferred trajectory by controlling the motion.

The path planning problem has been studied extensively over the past decades. See, for instance, the textbooks of Choset et al. [69], De Berg et al. [70], Latombe [71] and LaValle [72] for detailed introductions into path planning and many references to related work. Although path planning problems have well studied in stationary environment, less attention has been given to path planning in dynamic environments. Besides stationary obstacles, dynamic environments contain moving obstacles with which collisions must be avoided as well. This is especially the case for mobile robot and automated vehicle systems.

The main purpose of autonomous vehicle path planning is to determine a safe and collision-free path from a starting point to a goal point optimizing a performance criterion such as distance, time or energy [73] while taking into account the vehicle dynamics and manoeuver capabilities [74]. Some examples of common assumptions are listed below, these topics are not discovered in the thesis and more details can be found in [75]:

- Vehicle models vary in complexity from velocity controlled linear models to realistic car-like.
- Different levels of knowledge about the obstacles and other vehicles are required by different planning scheme. This ranges from abstracted obstacle set information to allowance for the actual nature of realistic noisy sensor data obtained from range-finding sensors.
- Different assumptions about the shape of stationary obstacles have been proposed.
- Uncertainty is always present in real time autonomous vehicle systems. To better

reflect this, assumptions can be made describing bounded disturbance from the nominal model, bounded sensor errors and the presence of communication errors [76].

2.6.1 Path Planning Levels

Based on the availability of environmental information, the autonomous vehicle planning can be roughly classified into different levels. The highest level is global planning (also known as offline path planning) which is primarily concerned in environments where workspace information about stationary obstacles and limited digital map are known in advance. Path planning is therefore the problem of finding a geometric feasible path form a known initial position to a given goal. The feasible path is required not to result in collision with stationary obstacles and therefore must adhere to any environment physical constraints.

On the other hand, the lower level motion planning is concerned with real-time online planning of the actual vehicle transition from one feasible state (including position, velocity, rotation etc.) to another satisfying the vehicle's dynamic constraints while avoiding obstacles in both stationary and moving form.

2.6.2 Sampling based Path Planning

In recent years, various probabilistically complete approaches that do not constrain the nature of the robot's motion have been suggested to solve the path planning problem in known environments. These methods are also known as sampling based planning [77] which are well suited for kinodynamic motion planning problem. Examples include the randomized path planner (RPP) [78], Ariadne's clew [79], probabilistic roadmap planners (PRM) [80], and rapidly-exploring random trees (RRT) [81]. The success of these planners in solving challenging problems can be explained by the fact that no explicit representation of the free configuration space is required. These practical planners satisfy a weaker form of completeness. They use randomization to treat the high dimensionality of configuration space and connect the collision free regions of the configuration space without requiring to explicitly computing this subset. The term probabilistically complete was introduced to characterize these sampling-based algorithms, able to find a solution if sufficient running time is given.

According to the survey [82], the sampling-based approaches can be grouped in two main families: 1) those using sampling techniques for constructing a roadmap in free configuration space [80] and 2) those using sampling within incremental search methods for exploring the configuration space looking for a particular path [78,81,83].

The roadmap based methods are more suitable when several motion planning queries involving the same mechanical system moving in a static environment must be solved. Computing time is spent in a pre-processing phase and then planning queries can be solved in real-time. In some papers, they are also called multiple-query methods. Among one the well-known method is the PRM which creates a roadmap by randomly sampling configuration from the configuration space. If these configurations are collision-free, they are added as nodes to the roadmap. The PRM is probabilistically complete which means it can guarantee a solution, if it exists, from the start to goal configuration as time approaches infinity.

The incremental search methods, also known as single-query methods are in general faster since they need not pre-processing. Among them the most popular method is called the RRT. Unlike PRMs which require lots of effort into the pre-processing and thus not very suitable for dynamic environment planning, RRTs aim to solve a specific query as quickly as possible, without using pre-processing. In RRT, a tree of valid paths is grown outward from the start configuration by random sampling, until any possible branch reaches the goal configuration. More details of RRT-like approaches related to this thesis are presented in the Chapter 6.

2.6.3 Planning in Partially Known Dynamic Environments

A natural feature of autonomous vehicle path planning problem as mentioned above is planning in dynamic environments, in which besides stationary obstacles, also moving obstacles are present. The simplest instance of the planning problem in dynamic environments is when the motions of the obstacles are predictable, that is, they are fully known and given beforehand.

In many cases, however, the motions of the moving obstacles are only predictable for the very near future, or are not predictable at all but with current measurement only. In this case, such on-board sensors are used for environment perception, providing information about the moving obstacles during the execution of the vehicle path. This can be used to extend the vehicle’s plan, or adapt a previously planned

global path to make it suitable for the new situation online. This is repeated until the goal has been reached, that is, during the motion of the vehicle there is a continuous cycle of interleaved sensing and planning. Hence, only lower level online planners are suitable for planning in partially known environments. There are two fundamental problems in partially known dynamic planning scheme. Assuming a global path planner is used at current time t_C giving a future path of the initial environmental situation till time $t_c + h$ (h is the horizon of prediction time interval). First, the predicted situation from of the world at near future $t_c + h$ may differ from the actual situation when moving obstacles change their trajectories during planning. This may result in invalid paths. Second, the path generated from global path planner that the vehicle will follow between t_C and $t_c + h$ is not guaranteed to be collision-free, since the global planner can only consider limited environment information and is based on the previous trajectories of moving obstacles.

2.7 Optimal Control Based Motion Planning

The offline planner for known environments has been discussed in previous sections. However in these planning algorithms, the environment would have be to be perfectly known in advance which is not conductive to autonomous vehicle planning applications especially collision avoidance based motion planning.

Recently, model predictive control (MPC) architectures have been applied to collision avoidance problems [75]. They have many favourable properties compared to the commonly used artificial potential field (APF) [84] methods and velocity obstacle based methods [85], which could be generally more conservative when extended to higher order vehicle models and easily extends to robust and nonlinear problems. MPC is also increasingly being applied to autonomous vehicle and mobile robot motion planning problems. It is useful as it naturally combines path planning with on-line stability and convergence guarantees [86, 87]. In general, MPC planning methods are more optimal, allow for the future position of the vehicle, and account for situations where multiple obstacles, vehicles and complex vehicle dynamics are concurrently present more naturally, and do not necessarily carry an excessive computational burden.

As all methods, optimal control has some limitations. For instance, the solution may be a local maximum or minimum, instead of the global one as intended. Also,

usually the complexity of this kind of problems is very high and sometimes even computationally impossible to solve. Besides MPC, other optimization based path planning algorithms are also listed below according to the review [75]:

Graph search algorithms: examples include A* [88], D* [89], and fast marching [90]. Most methods hybridize the environment into a square graph, an irregular graph, or a Voronoi diagram (the skeleton of points, which separates all obstacles). A search can then be performed to calculate the optimal sequence of node transitions. In addition, this may be used as the first step to find a bounded area within which further path planning operations can take place.

Optimization of predefined paths: examples include Bezier curves [91], splines [92], and polynomial basis functions [93]. While these are inherently smoother, showing completeness may be more difficult in some situations.

Artificial potential field methods: these methods are also ideally suited to on-line reactive navigation of vehicles (without path planning). These can also be used as path planning approaches, essentially by using more information about the environment [94, 95]. However, the resultant trajectories would not be optimal in general. APF methods have lower computational requirements than local planning approaches, but this is becoming less of a concern with ever increasing computational powers of unmanned vehicles.

Mathematical programming and optimization: this usually is achieved using mixed integer linear programming (MILP) constraints to model obstacles as multiple convex polygons [96]. Currently, this is commonly used for MPC approaches.

Evolutionary algorithms, simulated annealing, particle swarm optimization: these are based on a population of possible trajectories, which follow some update rules until the optimal path is reached [97, 98]. However, these approaches seem to be suited to complex constraints, and may have slower convergence for normal path planning problems.

Partially observable Markov decision processes: this calculates a type of decision tree for different realizations of uncertainty, and uses probabilistic sampling to generate plans that may be used for navigation over long time frames [99]. However, this may not be necessary for all MPC-based navigation problems.

2.8 Collision Free Planning using MPC

MPC-type approaches have previously been used to navigate vehicles in unknown environments [100, 101, 102]. Here, the MPC algorithm is combined with some type of mapping algorithms, however some of the rigorous guarantees normally provided in MPC approaches are harder to show. An approach to collision avoidance using these types of methods is to estimate obstacle positions based on bearing measurements combined with some state estimation method [103]. In this case, observability constraints can be taken into account during planning. When compared to potential field methods, MPC methods generally perform better as they consider a more optimal path that plans ahead as obstacles are approached. They are also less conservative, bringing the vehicle closer to the edge of its control capability.

2.9 Summary

This chapter provides a systematic literature review on two main aspects that will be covered in this thesis, namely situational awareness and decision making for autonomous vehicle. Particularly, the reviews of a wide variety of target tracking and path planning based techniques are covered and explained. The environmental information produced by target tracking (perception) system is used for achieving a collision free motion planning for autonomous vehicles.

First, the background of target tracking is explained with details discussion in two areas: the single target state estimation (filter) approaches and data association approaches based MTT algorithms. A detailed explanation of MHT algorithm is presented in this review.

Next, existing path planning techniques used in autonomous vehicle are surveyed and classified in details. Two specific algorithms RRT and MPC based path planning algorithms for solving autonomous vehicle motion planning in dynamic environment with collision avoidance functionality are also discussed.

Chapter 3

Single Target Tracking using Constrained MHE

3.1 Introduction

Although current automotive tracking technologies can give relative reliable performance in ADAS applications such as automatic emergency braking systems. It has been agreed in common that the accuracy of tracking (state estimation) performance can be greatly improved by extra domain knowledge [104]. Besides using advanced sensors such as radar, laser and cameras, the tracking performance can also be greatly improved by utilizing trajectory constraints and other environmental related information imposed from the road network and digital maps [105]. Such additional prior information can be treated as different types of constraints and subsequently implemented in Bayes' rule together with system measurement and other prior knowledge about the system dynamics [106]. As a result, the posterior distribution of the system state, taking into account constraints, is derived after measurement update process. Incorporating the constraint related information into state estimation can improve on the accuracy of tracking.

Different types of state constraints and the corresponding methods have been developed for solving the constrained state estimation problem. Based on D. Simon's recent research in papers [106] and [107], an overview of various ways to incorporate

state constriants in the KF is provided including: model reduction, pseudo-measurement approaches and projection based approaches. However theses approaches are restricted to deal with linear system and linear state constraints only and all these approaches result in the same state estimate under certain conditions [107]. To extend the above methods to inequality contraints, paper [108] mentioned a method of using an active set approach. An active set method uses the fact that it is only those constraints that are active at the solution of the problem that are significant in the optimality conditions. Further more, to apply the constrained KF to nonlinear systems and nonlinear state constrains, a basic linearization idea is used in [106] for both the system and constraints which is equivalent to the core concept of EKF. Although constrained KF methods are relatively easy in implementation, the above methods have several disadvantages even for basic linear and equality constraints [106]. Moreover, the technique used in projecting the unconstrained state estimate onto linearized state constraints is subject to constraint approximation errors which may result in convergence issues [109]. This makes KF not the optimal solution for constrained state estimation problem especially for the case of system inequality constraints when recursive analytic solution is not available.

Recently, some methods such as the constrained UKFs [110] and interior point likelihood maximization (IPLM) [111] are developed based on linearization approaches. Others such as Gaussian mixture filter (GMF) [112] and (constrained) Particle filter approches [113] are also developed using projection and trunction approaches. The majority of filters proposed to solve the constrained estimation problems focus on linear (in)equality or nonlinear equality constraints. A little research has been conducted on nonlinear inequality constraints so far [112]. However in ground vehicle tracking problems, (non)linear inequality constraints have played an important role for most tracking scenarios, e.g. highway road and roundabout boundary when road width is considered.

In this case, another strategy for determining an optimal state estimate is suggested and its core concept is to reformulate the estimation problem as an optimization approach based quadratic programming problem. More specifically, Rao et al. [114] have proposed a constrained state estimation for nonlinear discrete-time systems, where the state estimation is developed based on a moving horizon concept known as MHE. The basic strategy of MHE in determining the optimal state estimation is to reformulate the estimation problem as an optimisation problem using

a fixed-size estimation window. This method has been widely used in chemical engineering. Other applications include hybrid systems, distributed or network systems, large-scale systems and so on. However, the implementation of moving horizon approach based estimation methods in target tracking is still relatively an uncharted area.

Advantages for using MHE to solve target tracking state estimation could be significant. Since the method is optimization based, the road constraint and other relevant information in target tracking problems can be natually handled by MHE as additional restrictions in (non)linear and/or (in)equality form for both linear and nonlinear systems. In addition to state constraints, MHE is also capable of incorporating constraints on the state process and/or observation noises. In vehicle tracking, such constraints are typically used to model bounded disturbance or truncated distribution representing the influence of the operation environment on vehicle movement such as vehicle acceleration and deceleration.

Another advantage of using MHE as a state estimation method in target tracking is that it always considers a window of N latest measurements. Such feature is very meaningful in target tracking problems especially when targets are occluded by each other/stationary obstacles which leads to no reliable measurement at specific time step/steps. MHE utilizes the measurements in a receding horizon window could reduce the effect of unreliable measurements such as in the above situation in state estimation. Simulation results in [107] show that MHE achieves the smallest estimation error for nonlinear systems and nonlinear constraints. Theoretically, for a linear system without constraints and with a quadratic cost, MHE reduces to KF when the horizon length reduces to one [114].

A new target tracking strategy by using constrained MHE approach is proposed in this chapter. By applying optimization based MHE, not only the nonlinear measurement model but additional state constraints in target tracking problems such as road boundary are naturally handled. The proposed MHE algorithm is demonstrated by single target tracking scenarios verified by both linear and nonlinear measurement models. Compared with other filters, constrained MHE can produce high estimation accuracy while taking an acceptable computational load.

3.2 Constrained State Estimation

In the operation of automated vehicles, it is necessary to track all the nearby road users to make sure the safety of the vehicles and other road users. This chapter considers situation of tracking a single vehicle that is moving on road. This is in fact a constrained estimation problem as the objects of interest must be on the road. In this section, both the road constrained state estimation problem and MHE based target tracking are described.

3.2.1 Problem Formulation

Consider the movement of objects of interest described by the discrete system:

$$\mathrm{x}_{k+1} = f(\mathrm{x}_k) + \omega_k \tag{3.1}$$

$$\mathrm{z}_k = h(\mathrm{x}_k) + v_k \tag{3.2}$$

where

$$\mathrm{x}_k = [x_k, \dot{x}_k, y_k, \dot{y}_k]^T \tag{3.3}$$

where $f: \mathbb{R}^n \to \mathbb{R}^n$ is the nonlinear system dynamic function and $h: \mathbb{R}^n \to \mathbb{R}^m$ is the $nonlinear\ measurement\ model$. $\mathrm{x}_k \in \mathbb{R}^n$ is the state vector which contains position along x and y axis respectively in Cartesian coordinate and $\dot{x}_k$ and $\dot{y}_k$ are the respective velocities. $\mathrm{z}_k \in \mathbb{R}^m$ is the vector of available measurements. The vectors $\omega_k \in \mathbb{R}^n$ and $v_k \in \mathbb{R}^m$ are Gaussian noises of the process and the measurement described by independent density $p(\omega_k) = N(0, Q)$ and $p(v_k) = N(0, R)$ respectively, where Q and R are covariance matrices. It is commonly assumed that the initial distribution of the state vector x_0 is known as a Gaussian distribution $p(\mathrm{x}_0) = N(\tilde{\mathrm{x}}_0, \tilde{P}_0)$.

If let F_k and H_k be the linear (linearized) matrices respect to x_k and z_k, respectively, then the system shown above is now a linear time-invariant discrete-time system with dynamic function and measurement equation shown below:

$$\mathrm{x}_{k+1} = F\mathrm{x}_k + \omega_k \tag{3.4}$$

$$\mathrm{z}_k = H\mathrm{x}_k + v_k \tag{3.5}$$

3.2.2 State Constraints

As discussed in Introduction, ground targets are constrained when moving on road. Thus the knowledge of terrain database and road maps can be used as constraints and incorporated into the tracking algorithm. In most existing techniques, the road maps constraint target motion in a one-dimensional physical space [115] (by ignoring the road width) and incorporate them as equality constraints. This is a fairly good approach when an observer is far away from the moving objects such as in the scenario of unmanned aircraft tracking a ground vehicle using GMTI radar. This however would result in highly nonlinear constraint formulation and low tracking accuracy for autonomous vehicle tracking scenarios where road width is comparable to the sensor accuracy (e.g. high accuracy with shorter detection range sensors such as LIDAR). Besides, from the operation point of view, the autonomous vehicle or relevant ADAS must know which lane or precise location of the other moving targets on the road. In this chapter, road network information is considered as road width inequality constraints and the target motion is restricted by these physical constraints in both straight and curved segment.

Linear state inequality constraints

Suppose that at each time step k, x_k is subject to the following linear inequality constraint:

$$a_k \leq C(\mathrm{x}_k) \leq b_k \tag{3.6}$$

where C: $\mathbb{R}^n \rightarrow \mathbb{R}^c$, a_k, $b_k \in \mathbb{R}^c$, and the inequality $\leq$ holds for all elements of the vectors and $a_k \neq b_k$, $\forall k$. C is a known $c \times n$ matrix, a_k and b_k are the known vectors each with a dimension of $c \times 1$ representing the lower and upper road boundary individually, c is the number of constraints, n is the number of states, and $c \leq n$.

Specifically, for target tracking with straight (linear) road width constraint shown in Figure 3.1, equation (3.6) is expressed as:

$$\begin{bmatrix} -I \\ I \end{bmatrix} * T_g^l(\mathrm{x}_k) \leq \begin{bmatrix} -ub \\ lb \end{bmatrix} \tag{3.7}$$

where T_g^l is known as the transformation matrix representing the rotation from global coordinate l to the road network local coordinate g (with orientation along and

orthogonal to the road) by rotation angle θ. ub and lb are the upper bound and lower bound of the straight road respectively. The details of mathmatic expression of T_g^l is given in seciton 3.3.2.

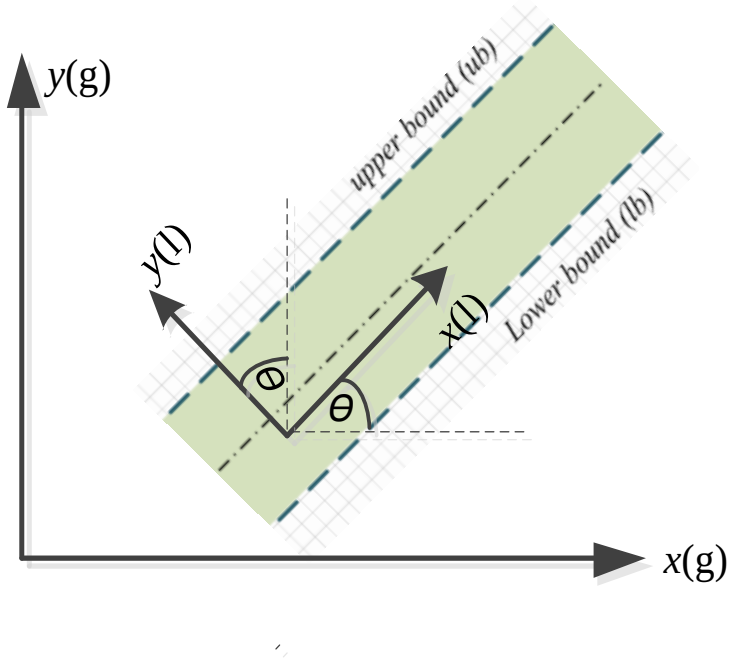


Figure 3.1 Straight road width linear constraint

Nonlinear state inequality constraints

In the same form as the linear road width constraint shown in (3.6), a circular or curved road segment shown in Figure 3.2 can be represented as a nonlinear inequality constraint represented by function $\mathcal{L}$ as:

$$r_1 \leq \mathcal{L}(\mathrm{x}_k) \leq r_2 \tag{3.8}$$

At each time step k, the road is defined by two arcs with radii $r_1$and r_2 representing the lower/upper road boundary, with the center at the origin of the Cartesian coordinate system.

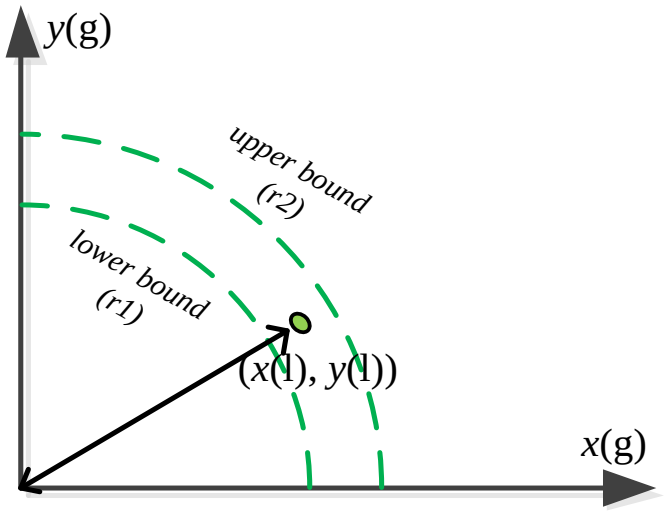


Figure 3.2 Curved road width nonlinear constraint

3.3 Constrained MHE for Target Tracking

MHE is an optimization approach based state estimation method which can take into account state constraints during estimation process. Essentially, MHE follows Bayes rule which maximizes the *a posteriori* probability density function (pdf) $p(X_N|Z_N)$ given a fixed horizon of measurements. Here X_N represents a horizon of N states $\mathrm{x}_{k-N,\ldots,}\mathrm{x}_{k-1}$ and Z_N is a horizon length of N past measurements $\mathrm{z}_{k-N,\ldots,}\mathrm{z}_{k-1}$. The joint conditional density is then given by:

$$p(X_N|Z_N) \propto p(Z_N|X_N)\, p(X_N|Z_{0:k-N-1}) \tag{3.9}$$

where

$$p(X_N|Z_{0:k-N-1}) = p(\mathrm{x}_{k-N}, \ldots, \mathrm{x}_{k-1}|\mathrm{z}_0, \ldots, \mathrm{z}_{k-N-1}) \tag{3.10}$$

$$p(Z_N|X_N) = p(\mathrm{z}_{k-N}, \ldots, \mathrm{z}_{k-1}|\mathrm{x}_{k-N}, \ldots, \mathrm{x}_{k-1}) \tag{3.11}$$

where $p(X_N|Z_{0:k-N-1})$ is the *a priori* state density given the measurements before the horizon; $p(Z_N|X_N)$ is the joint measurement likelihood function.

Assuming X_N is a first order Markovian chain, the *a posteriori* joint conditional density $p(X_N|Z_N)$ in (3.9) is reformed:

$$p(X_N|Z_N) = c \prod_{j=k-N}^{k-1} p\left(\mathrm{z}_j|\mathrm{x}_j\right) \prod_{j=k-N}^{k-1} p\left(\mathrm{x}_{j+1}|\mathrm{x}_j\right)\, p(\mathrm{x}_{k-N}|Z_{0:k-N-1}) \tag{3.12}$$

where c is the constant and $p(\mathrm{z}_j|\mathrm{x}_j)$ is the likelihood function for each measurement within the horizon. $p(\mathrm{x}_{j+1}|\mathrm{x}_j)$ is the state transition pdf and $p(\mathrm{x}_{k-N}|Z_{0:k-N-1})$ is the *a priori* density of the initial state x_{k-N}.

3.3.1 Constrained MHE

By using Bayesian *maximum a posteriori* (MAP) criteria on equation (3.12), it becomes $\arg max_{\{\mathrm{x}_{k-N},\ldots,\mathrm{x}_{k-1}\}} p(X_N|Z_N)$ shown in equation (3.13):

$$\arg max_{\{\mathrm{x}_{k-N},\ldots,\mathrm{x}_{k-1}\}} p(X_N|Z_N) \tag{3.13}$$

$$= \arg \max_{\{\mathrm{x}_{k-N},\ldots,\mathrm{x}_{k-1}\}} \sum_{j=k-N}^{k-1} log\, p(\mathrm{x}_{j+1}|\mathrm{x}_j) + log\, p(\mathrm{z}_j|\mathrm{x}_j) + log\, p(\mathrm{x}_{k-N}|Z_{0:k-N-1})$$

$$= \arg \min_{\{\mathrm{x}_{k-N},\ldots,\mathrm{x}_{k-1}\}} \sum_{j=k-N}^{k-1} -log\, p(\mathrm{x}_{j+1}|\mathrm{x}_j) + \sum_{j=k-N}^{k-1} - log\, p(\mathrm{z}_j|\mathrm{x}_j)$$

$$- log\, p(\mathrm{x}_{k-N}|Z_{0:k-N-1})$$

According to (3.13), the MHE cost function, subject to Eq. (3.4) and (3.5) with Gaussian noise, is then shown below as a quadratic programming problem:

$$\min_{\left\{\mathrm{x}_{k-N},\{\omega_j\}_{j=k-N}^{k-1}\right\}} \sum_{j=k-N}^{k-1} \left(\left\|\mathrm{x}_j - F\mathrm{x}_{j-1}\right\|_{Q^{-1}}^2 + \left\|\mathrm{z}_j - H\mathrm{x}_j\right\|_{R^{-1}}^2 \right) + \Gamma_{k-N}(\mathrm{x}_{k-N}),$$

$$\text{for } \{\mathrm{x}_{k-N},..,\mathrm{x}_k\} \in C_x \tag{3.14}$$

where C_x represents the constrained region. $\omega_k = \mathrm{x}_{k+1} - F\mathrm{x}_k$ denotes the optimal system process noise and $v_k = \mathrm{z}_k - H\mathrm{x}_k$ denotes the optimal measurement noise at each time k. $\{\mathrm{x}_{k-N:k}\}$ is the ensemble of states from time instance $k-N$ to k which solves the quadratic programming problem (3.14) while giving the optimal estimate solution. N is a moving horizon length which is chosen to give a trade-off between the estimation accuracy and the computational cost. For (3.14), different optimisation methods could be used to compute the states $\{\mathrm{x}_{k-N:k}\}$.

$\Gamma_{k-N}(\mathrm{x}_{k-N}) = -\log\left(p(\mathrm{x}_{k-N}|Z_{0:k-N-1})\right)$ as shown in (3.15) represents an arrival cost defined in [116] which plays an important role in summarising the effect of the past measurement as a *priori* information on the initial state x_{k-N}.

$$\Gamma_{k-N}(\mathrm{x}_{k-N}) \approx \left\|\mathrm{x}_{k-N} - \hat{\mathrm{x}}_{k-N}^{mh}\right\|_{P_{k-N}^{-1}}^{2} \tag{3.15}$$

where $\hat{\mathrm{x}}_{k-N}^{mh}$ and P_{k-N} represents the previous moving horizon state estimate and covariance at $k-N$, respectively. The unconstrained EKF [116] is adopted as the approximate method for calculating the arrival cost error covariance P_{k-N} as:

$$P_{k+1} = Q + FP_kF^T - FP_kH^T(R + HP_kH^T)^{-1}HP_kF^T \tag{3.16}$$

For nonlinear system equation (3.1) and (3.2), the linearized Jacobian matrix of dynamic function f and the measurement function h is calculated as: $F_k = \frac{\partial f(\hat{\mathrm{x}}_{k-1|k-1})}{\partial \hat{\mathrm{x}}_{k-1|k-1}}$ and $H_k = \frac{\partial h(\hat{\mathrm{x}}_{k|k-1})}{\partial \hat{\mathrm{x}}_{k|k-1}}$ respectively.

The state estimate of the MHE optimisation function (3.14) at time *k* is denoted as: $\mathrm{x}\left(k; \hat{\mathrm{x}}_{k-N}^{*}, \{\widehat{\omega}_j^{*}\}_{j=k-N}^{k-1}\right)$, including the optimised initial state $\hat{\mathrm{x}}_{k-N}^{*}$ and the optimised process noise sequence $\{\widehat{\omega}_j^{*}\}_{j=k-N}^{k-1}$. Then, at time *k*, the optimised estimated state $\hat{\mathrm{x}}_k^{*}$ considering linear dynamic function (3.4) can be calculated as:

$$\hat{\mathrm{x}}_k^{*} = \mathrm{x}\left(k; \hat{\mathrm{x}}_{k-N}^{*}, \{\widehat{\omega}_j^{*}\}_{j=k-N}^{k-1}\right) = F^k\hat{\mathrm{x}}_{k-N}^{*} + \sum_{j=k-N}^{k-1} F^{k-j-1}\widehat{\omega}_j^{*} \tag{3.17}$$

In constrained MHE, a horizon length of *N* states instead of only the current step is considered at each iteration. If considering the measurement update function (3.5) as a linear equality constraint, then the optimised state $\hat{\mathrm{x}}_k^{*}$ is constrained by the following equation:

$$H_kF^k\hat{\mathrm{x}}_{k-N}^{*} + H_k\sum_{j=k-N}^{k-1} F^{k-j-1}\widehat{\omega}_j^{*} + \sum_{j=k-N}^{k-1}\hat{v}_j^{*} = \{\mathrm{z}_j\}_{j=k-N}^{k-1} \tag{3.18}$$

where $\{\hat{v}_j^{*}\}_{j=T-N}^{k-1}$ is the optimised measurement noise for *N* horizon length and $\{\mathrm{z}_j\}_{j=k-N}^{k-1}$ is the horizon of *N* latest measurements.

As illustrated in Figure 3.3, a lower bound constraint is incorporated in the MHE. Correspondingly the estimated state is constrained to be only above the constraint value. The details of implementing road boundary based inequality constraints in MHE is discussed in the next section.

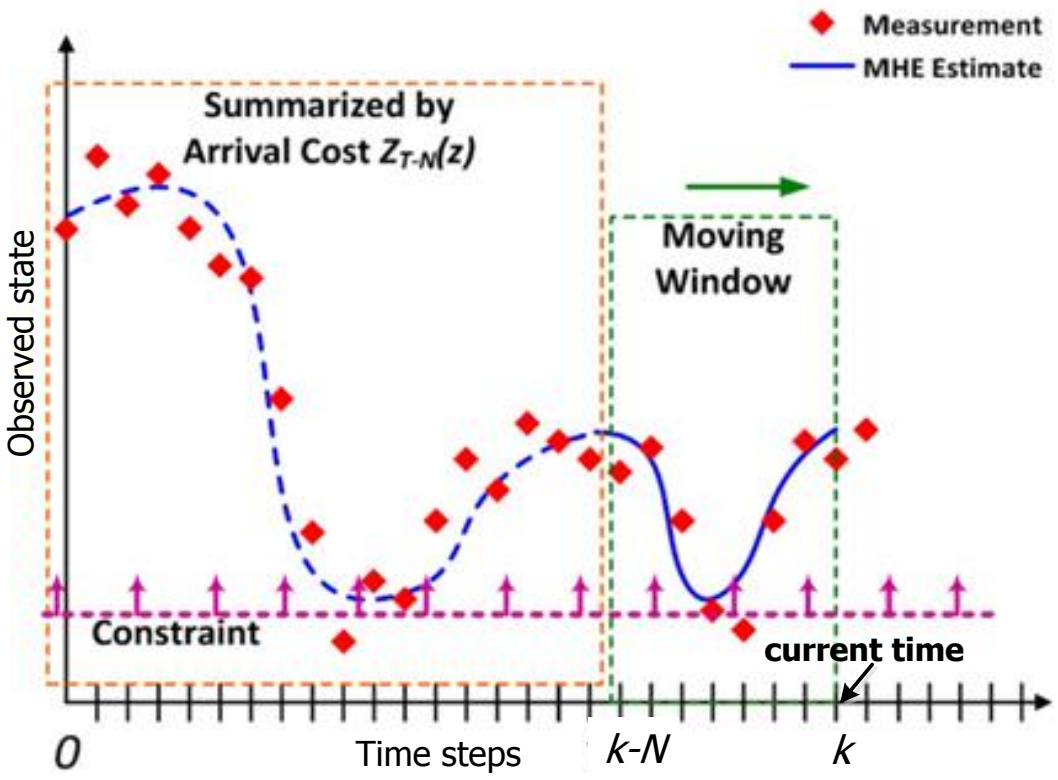


Figure 3.3 Illustration of MHE with constraints

3.3.2 Constrained MHE with Road Inequality Constraint

Since MHE is an optimization framework based state estimation algorithm, the physical road width constraints discussed above could be easily imposed in the MHE process.

Linear Inequality Constraint of The Road:

Linear inequality constraints such as the straight road shown in Figure 3.1, the estimated positon $[\hat{x}_k^*, \hat{y}_k^*]$ in global Cartesian coordinate x(g) and y(g) axis respectively, is transformed into a local coordinate $[x_k^l, y_k^l]$ by an counter-clockwise rotation angle θ from x(g) direction if considering a same coordinate origin. The constraint matrix T_g^l can be defined using a homogeneous transformation matrix from the global to local coordinate, $T_g^l = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix}$ and the local position is formed in (3.19):

$$\begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} \hat{\mathrm{x}}^*_{k,1} \\ \hat{\mathrm{x}}^*_{k,3} \end{bmatrix} = \begin{bmatrix} x^l_k \\ y^l_k \end{bmatrix} \tag{3.19}$$

Since the vehicle's maneuver is only limited in the lateral direction by the width of the road, the road inequality constraint for y_l in local coordinate is represented below in (3.20)

$$\begin{bmatrix} -I \\ I \end{bmatrix} y^l_k \leq \begin{bmatrix} -ub \\ lb \end{bmatrix} \tag{3.20}$$

Road Nonlinear Inequality Constraint:

For nonlinear inequality constraints such as the curved road shown in Figure 3.2, the estimated positon $[\hat{x}^*_k,\ \hat{y}^*_k]$ is constrained by the upper/lower road boundary r_1/r_2. In this case, the road width inequality constraint (3.8) is represented by equation (3.21). Thus the nonlinear inequality equation $\mathcal{L}$ is represented by $\sqrt{\hat{x}^{*\,2}_k + \hat{y}^{*\,2}_k}$.

$$r_1 \leq \sqrt{\hat{x}^{*\,2}_k + \hat{y}^{*\,2}_k} \leq r_2 \tag{3.21}$$

For a nonlinear measurement model (3.2) with two measurements: range r and bearing angle θ. The nonlinear measurement equation is given below:

$$\mathrm{z}_k = \begin{bmatrix} r_k \\ \theta_k \end{bmatrix} = \begin{bmatrix} \sqrt{x_k{}^2 + y_k{}^2} \\ arctan\left(\frac{y_k}{x_k}\right) \end{bmatrix} + v_k \tag{3.22}$$

Then the nonlinear equality constraint according to the nonlinear measurement function (3.2) can be formulated similar to equation (3.18), however, the linear measurement function H_k is substituted by $h(\mathrm{x}_k) = \begin{bmatrix} \sqrt{x_k{}^2 + y_k{}^2} \\ arctan\left(\frac{y_k}{x_k}\right) \end{bmatrix}$ and the measurements are $\{\mathrm{z}_j\}^{k-1}_{j=k-N} = \begin{bmatrix} \{\theta_j\}^{k-1}_{j=k-N} \\ \{r_j\}^{k-1}_{j=k-N} \end{bmatrix}$.

Besides the road width constraint on state values (position), constraint could also be combined in non-state vectors such as process noise. As mentioned in previous section, process noise is usually traded as acceleration in target tracking model. E.g. a

target is under acceleration mode if the process noise is greater than zero. MHE has a great advantage of capable to directly add inequality constraints on any variables defined in the objective function (e.g. state vector, process noise and measurement noise).

3.3.3 Constrained MHE with Missing Measurement

As discussed in Chapter 2, different from pure state estimation, in target tracking problems the data received by sensors is affected by measurement noise and other forms of interference. In this case the inherent uncertainties are always with measurements in target tracking problems and one main issue is measurement miss detection problem.

In MHE framework, a problem arises when missed measurement happens among a horizon of measurements since there is no individual prediction update process (unlike KF) and the estimation problem is solved by an optimization toolbox. In this chapter, the missed measurement is presumed as one step predicted state calculated by KF prediction update process shown in equation (2.18) and thus the estimated process noise ω_k and measurement noise v_j for the current step are taken as null sets. This assumption is equivalent to treat the non-available measurement updated a posterior estimate as a prior predicted state which is used in KF target tracking problem [117]. However, instead of directly using the predicted state as the estimation result, the estimation problem is resolved by the MHE quadratic programming cost function (3.14) considering a horizon of mixed measurements. In order to accommodate the road constraints, the missed measurements for the constrained MHE are replaced by the constrained predicted states. The detail of the mathematical explanation is shown in Chapter 4.

3.4 Simulations

In this section, two simulation examples are presented in the context of ground vehicle tracking. The first example is single target tracking with a linear measurement model and linear road inequality constraint. The second one is based on a nonlinear measurement model and nonlinear road inequality constraints. The results are compared with general KF and other constrained filters.

3.4.1 Linear MHE with Linear Road Inequality Constraint

In this section, a single target tracking scenario is considered using linear measurement models with road boundary constraints. The vehicle dynamics is described by a CV with noisy acceleration:

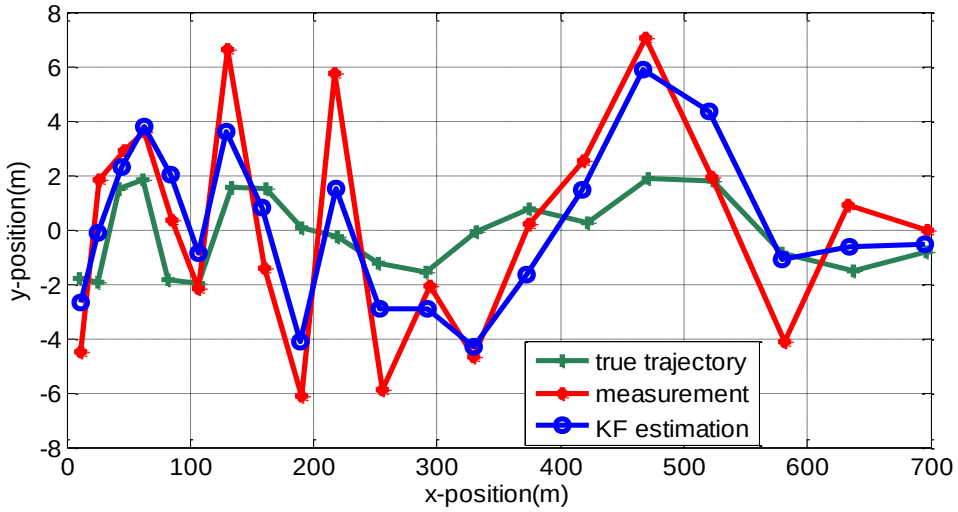
$$\mathrm{x}_{k+1} = \begin{bmatrix} 1 & T & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & T \\ 0 & 0 & 0 & 1 \end{bmatrix} \mathrm{x}_k + \begin{bmatrix} T^2/2 & 0 \\ T & 0 \\ 0 & T^2/2 \\ 0 & T \end{bmatrix} \omega_k \tag{3.23}$$

where the state vector $\mathrm{x}_k = [x_k, \dot{x}_k, y_k, \dot{y}_k]^T$ consists of the vehicle position and velocity in *x* and *y* directions, and $T = 1$ is the sampling interval, $\omega_k = \left[u_k^x, u_k^y\right]^T$ is a two-dimensional Gaussian process noise with zero mean and covariance matrix Q = *diag*{8,4} in a local coordinate where $diag\{.\}$ represents a diagonal matrix. This covariance represents higher motion uncertainty along the centre line direction and smaller uncertainty orthogonal to the road. The vehicle measurement model is a linear matrix in *x* and *y* potion with a Gaussian measurement noise v_k and covariance matrix R = *diag*{10, 10} in a global Cartesian coordinate as:

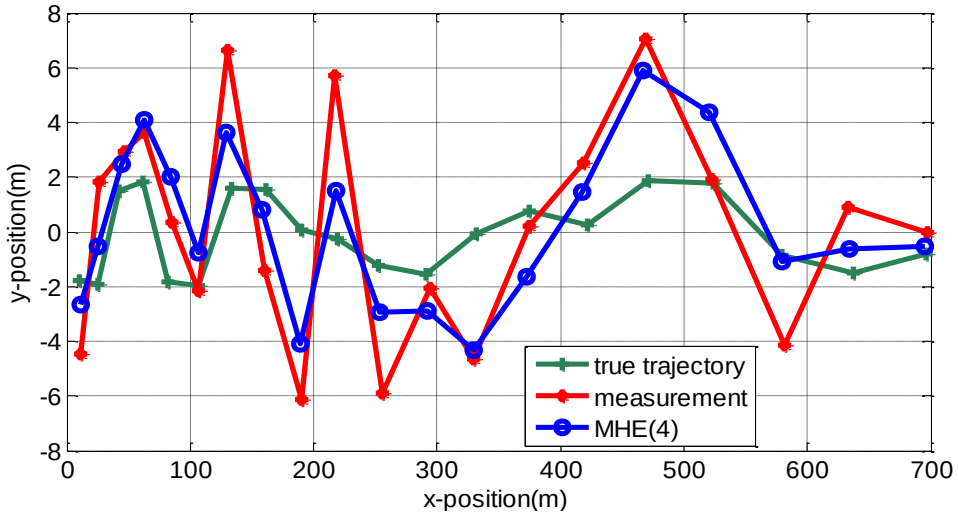
$$\mathrm{z}_k = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \mathrm{x}_k + v_k \tag{3.24}$$

A vehicle is moving on a single carriage way starting from a position of the middle of the road with coordinates (0 m, 0 m). The road is assumed to have a total width of 4 meters and the vehicle's trajectory is limited within the road width constraint. It is assumed that the vehicle accelerates straight to the east with an initial velocity of [10m/s 0m/s].

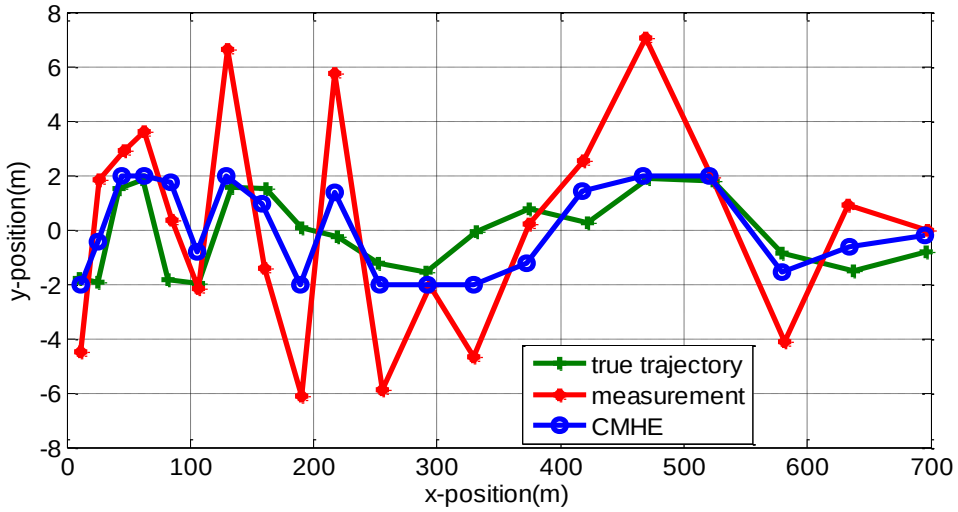
The constrained MHE (CMHE), standard MHE and KF are compared with a horizon size for MHE/CMHE chosen as 4. The results are shown below in Figure 3.4.


y-position(m)
x-position(m)
true trajectory
measurement
KF estimation

(a)


y-position(m)
x-position(m)
true trajectory
measurement
MHE(4)

(b)



(c)

Figure 3.4 Comparison between KF (a), MHE (b), CMHE (c)

As shown in Figure 3.4, the KF and MHE without using road boundary constraint have the estimation result outside the road. The performance is improved in the CMHE with the tracking results being projected on the road boundary.

For further comparison, the root mean squares errors (RMSEs) for different filters are calculated. Table 3.1 shows the RMSE for the position states, each is calculated for an average of 100 times Monte Carlo simulations running MATLAB. The Mathworks' Optimization Toolbox™ *quadprog* in MATLAB software is used.

Different MHE horizon length are also compared in Table 3.1. Usually, the choice of the horizon length N is a turning parameter in MHE. The horizon size is chosen to give a trade-off between estimation accuracy and computational effort. From the theoretical standpoint, MHE can keep stability as long as the horizon length is greater than the observability index of the system measurement model [118] which is two in this study. A practical rule of a proper value is usually to choose the horizon length as a positive integer as twice the order of the system [116].

From Table 3.1 some conclusions can be drawn. First it shows that the generic MHE with a horizon length of 1 is identical with KF when considering linear dynamic/measurement model without additional constraints. This is because for linear system without constraints, MHE often tends to perform the same as the iterated KF [107]. Second, by comparing MHE or CMHE using horizon of 1, 4 and 10, there is no

significant improvement when using a longer horizon in this study. This is because the system and constraint considered in this study are both linear equation which covers the advantage of using MHE for solving nonlinear dynamics. Third, the arrival cost in this study is calculated by KF covariance and all the results using different horizon length show a stable result. As proved in [118], when the process model is linear, the approximate arrival cost calculated by KF covariance, regardless of whether there are constraints, can yield a stable MHE with guaranteed convergence/stability. Last but not least, the constrained MHE with a proper horizon length presents a much better tracking result than unconstrained state estimation methods.

Table 3.1 Averaged RMSEs for KF, MHE and CMHE.

Filter Type	RMSE Estimation Error (m)
Kalman Filter	3.2666
MHE (Horizon size 1)	3.2666
MHE (Horizon size 4)	3.2574
MHE (Horizon size 10)	3.2461
CMHE (Horizon size 1)	2.8212
CMHE (Horizon size 4)	2.8057
CMHE (Horizon size 10)	2.7960

3.4.2 MHE with Missing Measurements

In this section, the target tracking measurement miss detection problem is considered. The same single target tracking scenario discussed in section 3.4.1 is used. However this time the sensor has a low detection rate $P_D = 0.5$ which means only partially measurements are available during the tracking process. The tracking scenario is shown in Figure 3.5 and the index of the measurement according to the sampling instance is shown in Figure 3.6. In this case, measurements 4, 8, 9, 12, 13, 14, 17, 18, 19, 20 are miss detected which are shown with y position at 0 for illustration purpose.

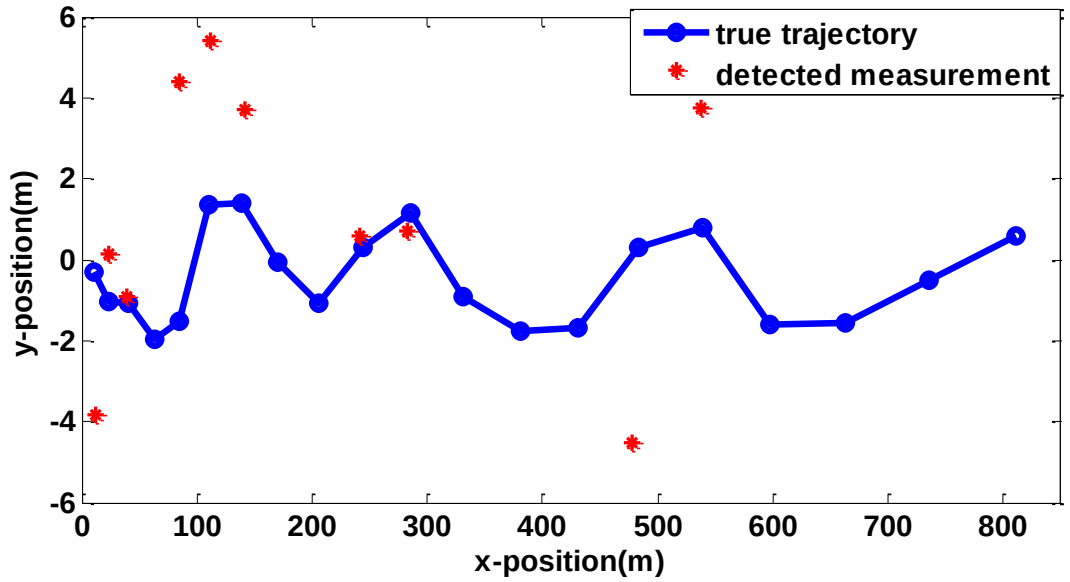


Figure 3.5 True trajectory with detected measurement

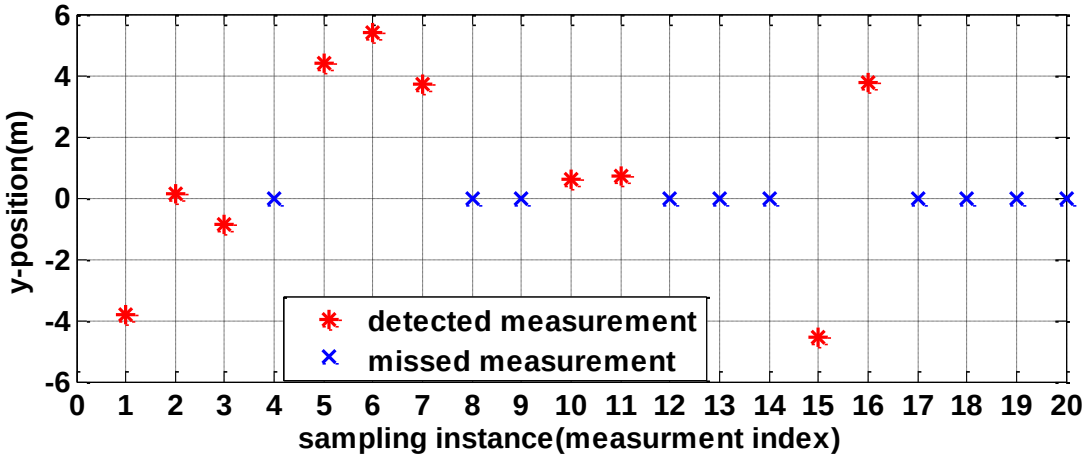


Figure 3.6 Index of missed/detected measurements

The CMHE, MHE and KF are compared with different horizon size. As discussed in section 3.3.3, the *a prior* predicted state and covariance are used as the estimation result for KF while miss detection happens while MHE resolves the estimation problem using a quadratic programming solver.

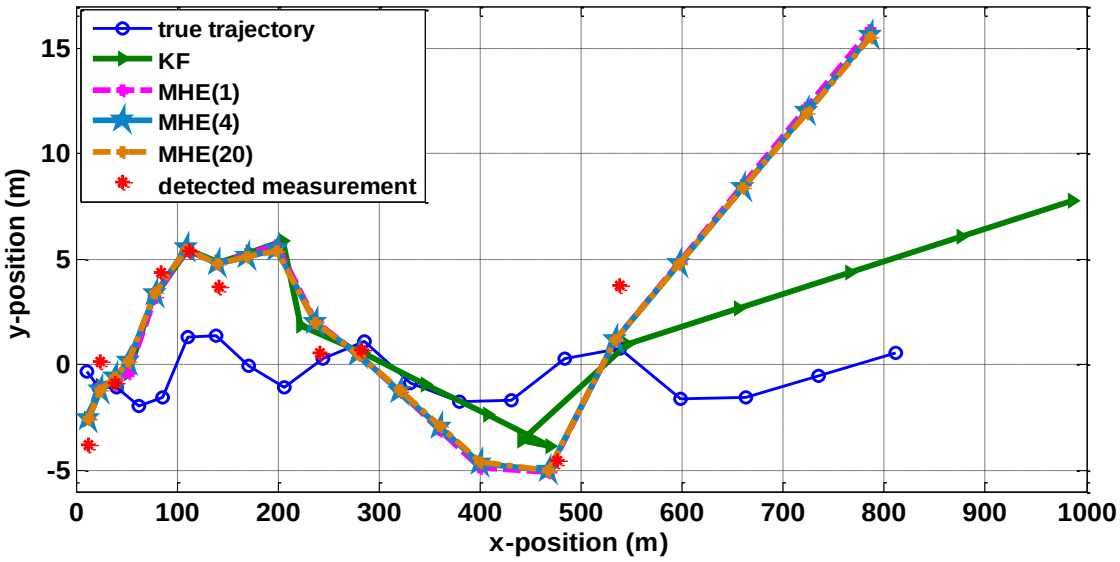


Figure 3.7 Tracking result with missed measurements using MHE with horizon 1, 4 and 20 against KF

As shown in Figure 3.7, the results between general MHE and KF without road constraints are compared. First, let's focus on the beginning 7 sampling steps where target is only temperately missing for one sampling step at instance 4. Both methods despite of a longer horizon length show a very similar estimation result which bias to the detected measurements. This is because for a linear dynamic system with CV model the target maneuver is relatively simply and a one-step maneuver is predictable when the (a prior) velocity estimation does not change too much from the previous steps. However when a longer window of miss detection happens, such as step 12, 13 and 14, the prediction information is not reliable any more. This is shown by the result from KF estimation between x position 300m and 500m. Due to very poor velocity estimation (prediction), the position estimation error is accumulated and results in an unrealistic turning around maneuver. On the other hand, the MHE using the optimization solver shows a much more accurate state estimation especially in orthogonal direction since the target vehicle has much higher maneuver uncertainty in

the orthogonal direction (x-axis) than lateral direction (y-axis). The same result is also shown in Figure 3.8 at the last couple of steps (17, 18, 19 and 20) for the tracking process when no measurement is detected for a continuous 4 steps. The MHE shows more accurate position estimation while the rough prediction from KF makes the estiamtion far away from the true trajecotory.

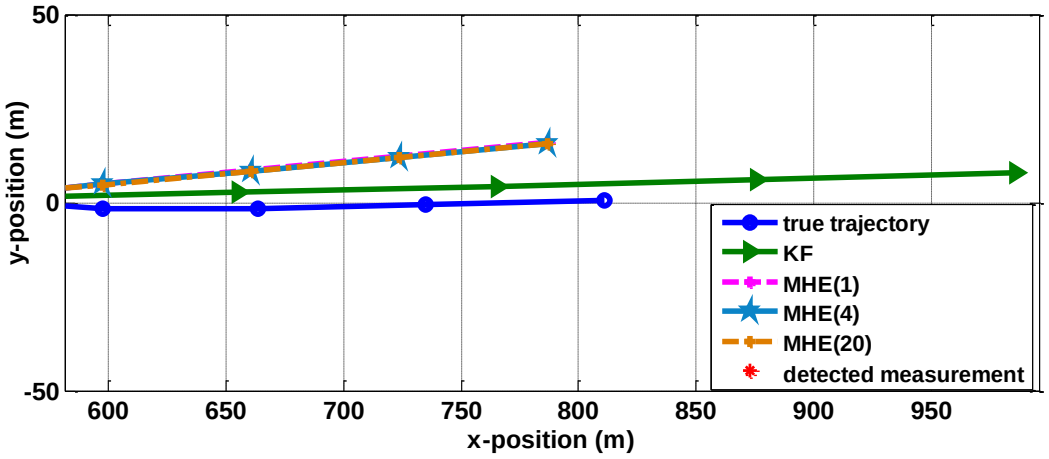


Figure 3.8 Tracking result with missed measurements using MHE with horizon 1, 4 and 20 against KF from step 17 to 20

The results of the CMHE with different horizon length are also compared as shown in Figure 3.9. Comparing the results between CMHE, MHE and KF shown in Figure 3.7 and 3.9, once again it confirms that the road constraints play a signification role in improving the tracking accuracy. The utilising of longer horizon length improves the tracking results but with only limited effect due to the linear dynamic system and constraints. When a continuous multiple steps miss detection happens (as shown in Figure 3.9 from x position 300m to 500m as well as 600m to 800m), the CMHE with longer horizon length tends to produce smoother and more stable estimation result which is less affected by missed measurement. This gives MHE a great benefit of solving autonomous vehicle target tracking problem where challenging tracking occlusion problem happens. On the other hand, the result from x position 0m to 100m shows that the reduced CMHE with only one horizon length, which can be taken as a KF solved by optimization method with constraints, is more sensitive for temporally missed measurement. The horizon length should be tuned for different tracking scenarios in practical tracking applications.

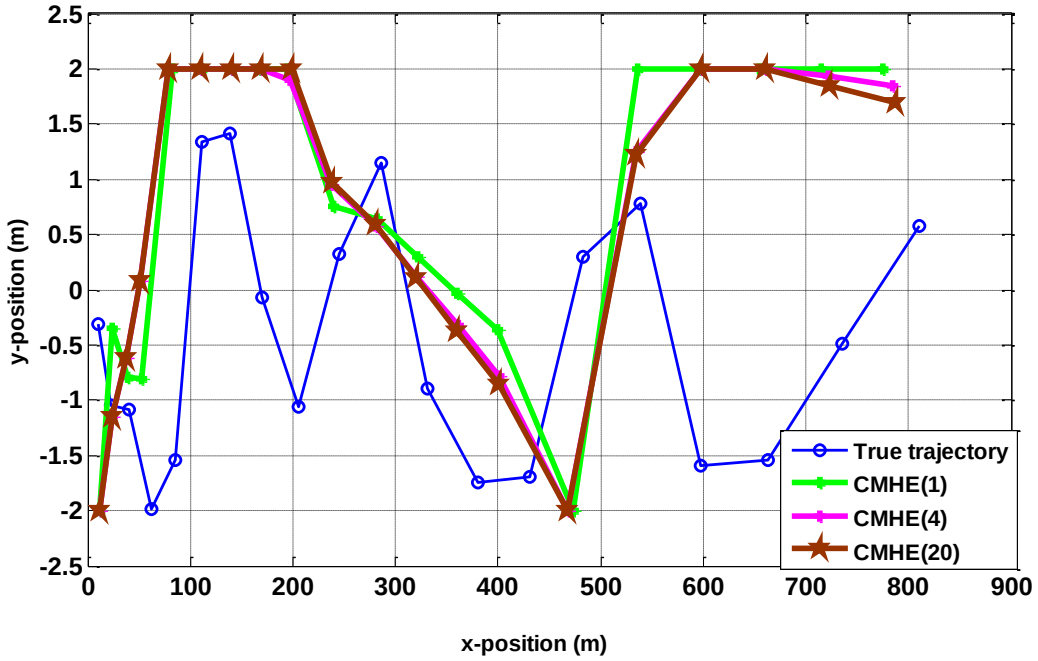


Figure 3.9 Tracking result with missed measurements using CMHE with horizon 1, 4 and 20

For further comparison, the position RMSEs of different filters with an average of 100 times Monte Carlo simulations is calculated. As shown in Table 3.2, overall the MHE based optimization solver especial CMHE is more suitable for solving target tracking with missed measurement issue than KF. Even for general MHE with a horizon of 1 the result is still slightly better than KF. This is because in KF, the fundamental principle is to make a trade-off between the *a priori* predicted information and the current measurement by tuning a 'weighted compensator' the KF gain. When no sufficient measurement is available, the KF turns to completely trust the predicted state. However in MHE, the estimation problem is solved by minimizing the quadratic cost function (3.14) considering (linear) constraints. The decision is made based on the measure of confidence of the prediction model, measurement and the knowledge of the initial state x_{k-N} (using the arrival cost term) which is the state estimation of the previous step when using a horizon length of N=1. When no measurement information is available, instead of purely relying on the predicted state, the MHE makes a trade-off between the last step estimation x_{k-1} and the current step state prediction. This strategy works better when accommodating the road constraint which makes the on the constrained predicted state closer to the true trajectory.

Especially in this study, where the sensor measurement has less measure of confidence than the prediction model (with larger uncertainty/noise), the constrained state prediction in some sense is more suitable for representing the true trajectory than the measurement.

The increasing of horizon length could improve the tracking result under some conditions e.g. using a horizon of 20, which makes the MHE a full information estimation algorithm [119], for solving a temporary missed measurement during the horizon window though with diminishing returns once N is sufficiently large. However under some extreme situations, e.g. as shown in Figure 3.8 when a continuous of 4 measurements are missing in one iteration of MHE (with a horizon length of 4), the accumulated prediction error makes the longer horizon MHE worse than a single horizon length one.

Table 3.2 RMSEs for KF, MHE and CMHE with missed measurement $P_D = 0.5$

Filter Type	RMSE Estimation Error (m)
Kalman Filter	8.4034
MHE (Horizon size 1)	8.0524
MHE (Horizon size 4)	8.0668
MHE (Horizon size 20)	7.8840
CMHE (Horizon size 1)	7.0985
CMHE (Horizon size 4)	7.1668
CMHE (Horizon size 20)	6.7617

3.4.3 Nonlinear MHE with Nonlinear Road Inequality Constraint

In this example, the test scenario is set up following the previous study of [112]. A moving vehicle on a circular road section is considered as shown in Figure 3.5. The road is defined by two boundaries with two arcs of r_1=96m and r_2=100m, respectively, centred at the origin of a Cartesian coordinate system. The same vehicle dynamics shown in (3.23) is used. ω_k is a two-dimensional Gaussian process noise with zero mean and covariance matrix $Q = diag\{1,1\}$. The initial state of the vehicle is $\mathrm{x}_0 = [98,0,0,10]^T$. The vehicle is supposed to move for 20 seconds with $T = 1$. The vehicle is tracked by a nonlinear range and bearing model shown in (3.22). v_k is a two-dimensional Gaussian zero-mean measurement noise with a diagonal covariance matrix $R = diag\{8, 10^{-3}\}$. Given the road boundaries shown in Figure 3.10, the nonlinear state inequality constraint (3.21) is considered.

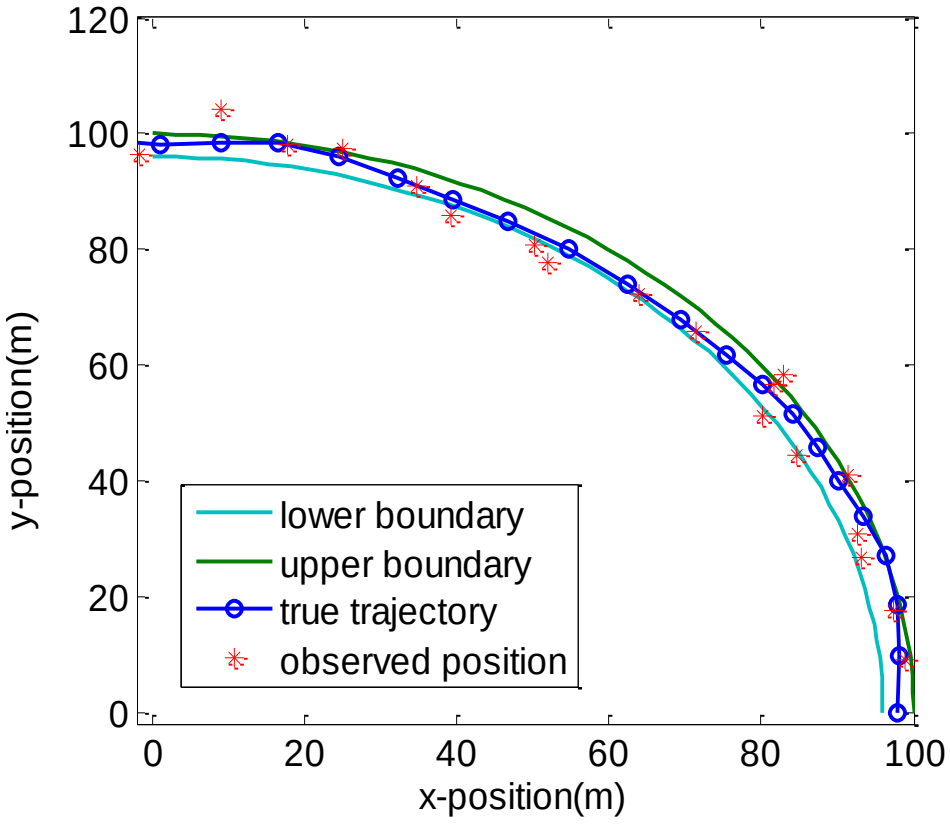


Figure 3.10 The simulated circular road tracking scenario

The performance of CMHE with different horizon length (N=2 and 8) with some other conventional filters [112] aiming at illustration of handling nonlinear in system and constraints in position estimation are compared. A brief introduction of each filter used in comparison study is provided below while more details can be found in the provided relevant reference.

- *Unconstrained filters:*

UKF [120]: As explained in Chapter 2, instead of approximating the nonlinear function, replaces the distribution of a state estimate by a set of deterministically chosen sigma-points and associated weights.

Divided difference filter (DDF) [121]: Similar to the EKF, the DDF approximates the nonlinear mappings, but instead of the Taylor series expansion. Stirling's interpolation formula of the first order is used [121] which means the derivatives are replaced by differences.

Gaussian mixture filter (GMF) [112]: The GMF is based on the analytical solution to the Bayesian framework for solving nonlinear non-Gaussian dynamic stochastic system, where all the pdfs are assumed in the Gaussian mixture form. The GMF can be interpreted as an approximation of the true pdf e.g., using the EM algorithm. The GMF can be imagined as a parallel run of several local filters depending on the chosen approximation weighted with respect to the measurement. In this study, the GMF is based on the UKF and DDF [112].

- *Constrained filters:*

tUKF [112] and tDDF [112]: are the extension of the generic UKF and DDF discussed above using truncation approach [112] handling nonlinear inequality constraint. The aim is to find the estimate subject to the constraint by approximating the truncated pdf. The Monte Carlo based truncation method are using with 500 samples.

tGMF[112]: the extension of the generic GMF by using the truncated Gaussian mixture pdf. The details of the mathematical explanation can be found in [112].

cPF[113]: the extension of the PF discussion in Chapter 2 using a global sampling technique and truncation approaches to directly trim the conditional pdf of the state with respect to the nonlinear constraints. 10^3 samples are used based on [112].

CMHE: the proposed CMHE in this chapter is used. However it needs to be wary of the divergence (instability) of the arrival cost when the process model/constraint is nonlinear. Since the EKF covariance does not guarantee stability, additional measures might be used to guarantee stability. In this case, a degree of forgetting [38] is implemented in this study to guarantee convergence /stability: the MHE should not weigh the past data too heavily. One property of the KF is that it exponentially forgets the past data [38]. A 'forgetting factor' to the approximate arrival cost is added to premultiply the approximate arrival cost by a scalar $\alpha \in (0,1)$:

$$\Gamma_{k-N}(\mathrm{x}_{k-N}) = \alpha \left\| \mathrm{x}_{k-N} - \hat{\mathrm{x}}_{k-N}^{mh} \right\|^2_{P_{k-N}^{-1}} \tag{3.25}$$

The further discussion regarding forgetting factor in constrained MHE can be found in [38]. More recently, other sampling based nonlinear filters such as PF [122] and UKF [123] are proposed for calculating the arrival cost. The further study are suggested in the future work.

The constrained nonlinear optimization problem can be solved by various methods [88], therefore all of the theory that applies to the particular optimization algorithm that is used also applies to CMHE. In this study, the optimization problem is solved by *fmincon* solver in MATLAB on a 2.4 GHz CPU.

The performance of different filters was measured using the position RMSE. The results are shown in Table.3.3 with average results of 100 times Monte Carlo simulations. 7

Table 3.3. Estimation performance of filters

	UKF	DDF	GMF	tUKF	tDDF	tGMF	cPF(10^3)	CMHE(2)	CMHE(8)
RMSE	2.79	4.50	2.51	2.06	2.21	1.91	2.07	2.11	1.99
Time(s)	0.019	0.027	0.042	3.280	3.458	6.612	20.010	1.09	2.97

From Table 3.3, the impact of different constrained filters, considering nonlinear measurement model and nonlinear inequality constraint, on the estimation quality with different increase of computational cost can be seen. It can be seen from Table

3.3 that the tUKF, tDDF, tGMF outperform their unconstrained conventional filters UKF, DDF and GMF. The cPF provides high quality estimates however at an expense of high computational cost. The optimization based CMHE algorithm in this case provide similar performance to those based on truncation approach such as the tUKF, tDDF and tGMF. This fact is supported by the results achieved by the proposed CMHE filter using the approximate arrival cost (3.25), which provides reasonable good performance especially when increasing the horizon length. When N=8 the CMHE provides the second best RMSE=1.99 among all filters in Table 3.3 which is slightly worse than tGMF with RMSE=1.91 however CMHE provides a much better the computational cost with only half time taken for tGMF.

By comparing the result of CMHE using horizon length of 2 and 8 it can be found that the performance of CMHE improves as one increase the horizon length. The improvement is much more significant than the one in Table 3.1 since nonlinear system and nonlinear constraints are considered in this study. However, the computational cost also increases with the horizon length. Thus the estimation effect and the computing speed are needed to be balanced by choosing N.

Other recent studies on constrained state estimation algorithm, such as [107], [112], have also provide similar results. In [107] it is proven that for nonlinear measurement model and nonlinear constraints, the results indicate that of all the algorithms investigated, the CMHE results in the smallest estimation error. The related simulation results are provided in the Appendix A. However, this performance comes at the expense of programming effort and computational effort that is orders of magnitude higher than other methods. [112] also comprises CMHE with other truncation and projection based techniques for nonlinear dynamic system with linear state inequality constraint. The results vary when using different arrival cost and horizon length in CMHE. For and zero arrival cost with full horizon length where the CMHE now represents the full information filter, the relevant CMHE approach provides slightly better MSE results than other techniques. However the MHE computational costs is by two orders higher than others. If computational expense is a consideration then the truncation based techniques such as tGMF performs better than CMHE when a shorter and more realistic horizon length is implemented.

In summary, the “best” constrained estimation algorithm depends on the applications. For ground target tracking scenarios, the CMHE provides generally better performance but with relatively higher computational cost.

3.5 Summary

This chapter proposes CMHE algorithm for solving single target tracking problems for discrete-time linear and nonlinear measurement systems. External road information is employed by CMHE filters such as road boundary inequality constraints in both linear and nonlinear forms. The proposed MHE algorithm is demonstrated by single target tracking scenarios verified by both linear and nonlinear measurement models considering linear and nonlinear inequality constraints. Missed measurement issues is also considerd. Since in target tracking problems, targets are often occluded by other obstacles which leads to no reliable measurement at specific time step/steps. Simulation results show that, (C)MHE, utilizes the measurements in a receding horizon window, reduces the effect of unreliable measurements and produces more accurate tracking result. Comparing with other filters, CMHE can produce high estimation accuracy while taking an acceptable computational load. Hence, this research will extend the applicability of the MHE techniques to a wider application area in solving target tracking problems. In the next chapter, the CMHE will be implemented for solving more complicated MTT problems.

Chapter 4

MHE-MHT with Road Constraint Information

4.1 Introduction

Tracking multiple ground moving objects (e.g. vehicles, pedestrians and motorbikes) is playing a significant role in autonomous vehicles and ADASs. Different from multiple target tracking approaches applied in aerospace area, the motion of the ground vehicles are likely limited by road and terrain constraints. This information could be taken as additional domain knowledge to enhance tracking quality and continuity. In Chapter 3, it has been proved that information of road constraint can be exploited to improve the tracking performance in single target tracking scenarios. In this chapter, the constrained MHE (CMHE) is extended to solve MTT when situations become more complex with missed detection, false alarm and tracking occlusion.

In this chapter, a new MTT strategy namely Moving Horizon Estimation based Multiple Hypothesis Tracking (MHE-MHT) is proposed. To solve tracking ambiguity (data association) problem in MTT, an improved multiple hypothesis tracking (MHT) framework is developed by implementing the constrained MHE as a state estimation technique. Different from most of other recent researches which focus on MTT data association process [14], this chaper focus on improving MTT performace utilising

extra domain knowledge which considers environmental physical constraints. Comparing with traditional MHT, the new MHE-MHT framework inherits the advantages from MHE which makes it suitable for systems with nonlinear measurement and capable to systematically deal with state constraints derived from environmental information. In addition to the state estimation layer, in order to explicitly deal with environmental constraints based extra domain knowledge, modifications have been made in the MHE-MHT framework comparing with the original MHT structure such as constrained state prediction and data association, target maintenance logic and *m*-best *N* scan prunning technique. The details of the improved MHT structure used in MHE-MHT is explained in Section 4.4 in this chapter.

Performance of the proposed MHE-MHT algorithm is demonstrated by multiple ground vehicle tracking scenarios considering road constraints with an unknown and time varying number of targets observed in clutter environments with both linear and nonlinear measurement models. Simulation results at the end of the chapter show that the proposed technique efficiently tracks multiple vehicles accurately and reliably even when targets approach or cross each other in a highly cluttered environment. The proposed MHE-MHT contributes a further improvement in reducing the tracking error in both state estimation and data association aspects by incorporating the road boundary constraints explicitly.

4.2 Background

The problem of estimating the position of multiple moving targets, also known as MTT, has become an important part in autonomous vehicles and advanced driver assistance systems. Knowledge about the state of moving objects can be taken as valuable information to improve the level of autonomy for vehicles. The aim is to achieve an improved collision avoidance behavior and safe road safety driving even in populated environments.

As mentioned in Chapter 2, several approaches for MTT have been developed over the last decades. Among them, the data association based methods have achieved a great success in a wide range of applications, especially in the autonomous vehicle area recently [14].

Different from GNN and JPDA which consider data association decisions one scan at a time, summarising previous data by a single hypothesis, MHT algorithm in [48], however, is a more complex approach that considers data association across multiple scans and a number of hypotheses. In other words, MHT algorithm attempts to keep all possible association hypotheses over multiple frames of data. This will result in an exponentially growing number of hypotheses and thus a NP-hard problem. Cox [58] in 1997 developed an efficient implementation by using polynomial time optimization algorithms to find the k-best solutions to an assignment problem along with pruning and merging techiniques to reduce the number of low probability hypotheses. MHT essentially keeps a set of multiple hypotheses and thus the assignment ambiguity will be resolved in future when subsequently new observations are arrived. In this case, hard decisions are not made until they need to be with the fact of using more information rather than just the current data frame, thus possible association error could be corrected when more evidences are updated. MHT also has the advantage of being able to deal with track creation, confirmation, occlusion and deletion in a probabilistically consistent way and is very suitable for autonomous surveillance. Such features along with the dramatic increases in computational capabilities have made MHT a preferred data association method for modern systems [47].

4.3 Problem Formulation

The aim of MTT algorithms is to track the state of a number of *M* targets. As with the formulation in Chapter 2, let the state of target *n* at discrete time *k* be denoted x_k^n, and let T^n denote the set of all states for target *n* during the tracking process, i.e. $T^n = \{\mathrm{x}_k^n\}$ for $k = 1 \ldots T$, which is also known as a track for target *n*. In MHT based MTT, tracking initiation, maintenance and deletion is naturally considered without running any separate high level logic process. In this case, in addition to target states, each track T^n contains other parameters such as initiation time t_i^n (the time step when a new target is detected), life time l^n (the age of a detected target) and deletion time t_d^n (the time step when a detected target is deleted from the tracking list).

Let $\mathrm{Z_k} = \{\mathrm{z}_k^i\}_{i=1}^{m_k}$ denote the set of m_k measurements received at time *k*. Since the data received by sensors are affected by measurement noise and other forms of interference, inherent uncertainties are always with measurements, e.g. false alarms

(clutter) represented by false alarm probability p_{FA} ($p_{FA} > 0$) and missed measurements represented by detection probability p_D ($p_D < 1$). In this case, each of the measurement may belong to one of the three possible circumstances: i) the measurement starts a new target, ii) the measurement is a false alarm, and iii) the measurement belongs to an existing target.

Such measurement to track assignment is known as a hypothesis. It is assumed that: i) each hypothesis contains a set of compatible measurement to track assignments and ii) assignments are defined as 'compatible' if they have no measurements in common which means that each measurement can only update with one of the existing tracks in each hypothesis. The detail of hypothesis generation process is illustrated in Figure 2.9 in Chapter 2. Let N_h denotes the total number of hypotheses $\{\theta_{k-1}^i\}_{i=1}^{N_h}$ at time $k-1$. Each of the hypothesis θ_{k-1}^i is a history of assignment sets to time *k*. Each of the assignment sets Θ_k is characterized by three elements: i) the assumed number of targets (tracks), ii) the assignment (data association) result and iii) the corresponding assignment probability $p(\Theta_k)$. Each existing hypothesis is extended to a set of new hypotheses $\{\theta_k^l\}_{l=1}^{N_h}$ at time k by considering all possible track-measurement assignment sets when a new set of measurements $\mathrm{Z_k} = \{\mathrm{z}_k^i\}_{i=1}^{m_k}$ are received.

The evaluation of alternative hypotheses formation is based on the hypothesis probability $p(\theta_k^l)$. The formulation of $p(\theta_k^l)$ includes the prior probability of the existing hypothesis $p(\theta_{k-1}^i|\mathrm{Z}_{0:k-1})$; the false alarms density $\beta_{FA}^{(m_k^{FA})}$considering m_k^{FA} false alarms; the probability of detection sequences $\beta_{NT}^{(m_k^{NT})}$considering m_k^{NT}detected new targets; dynamic consistency of the measurements in the tracks based on the predicted measurement density $p_{k/k-1}^j\left(\mathrm{Z}_k^{\Theta_k^{-1}(j)}\right)$. The detail of hypothesis probability calculation is explained in Section 4.5.

Instead of generating all possible hypotheses such as in the generic MHT [58], techniques are used in this chapter so as to avoid combinatoric explosion and make the proposed algorithm more feasible for real time application. The details are also explained in Section 4.5.

4.4 Road Constraint aided MHT

4.4.1 Mobile Sensor Model

In autonomous vehicle target tracking scenarios, the tracking sensors such as radar or LIDAR are equipped with the ego vehicle and thus moving during the tracking process. This is known as a positioning (as a part of navigation) problem in robotics [124] where the ego vehicle utilises on-board GPS or inertial sensors to identify its own position. To simplify the system complexity, in this chapter it is assumed that the positioning sensors can give perfect global position information about the ego vehicle without measurement noise and motion uncertainty. General state estimation methods such as KF, PF could be utilized to solve the localization position uncertainty problem [125] [126].

The other moving vehicles (targets) are tracked based on the sensor (ego vehicle's) current location using either linear measurement model (3.23) with position information or nonlinear measurement model (3.20) with bearing and range information.

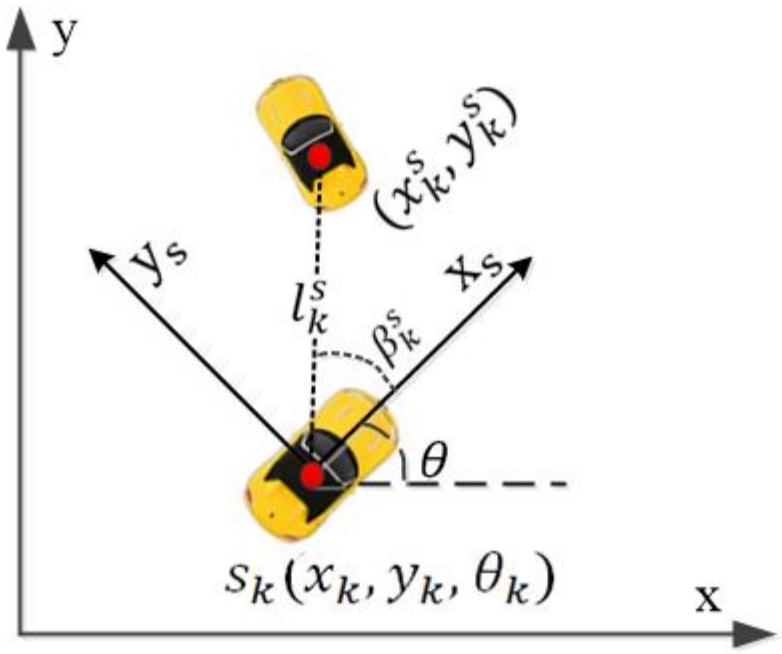


Figure 4.1 Measurement of target vehicle according to the ego vehicle state s_k

As illustrated in Figure 4.1, the state of sensor (ego vehicle) at time k is defined as $\mathrm{s}_k = [x_k, y_k, \theta_k]^T$ located in the global Cartesian coordinate $\mathcal{G}$ with position x_k and y_k in global coordinate x and y axis respectively and a rotation angle θ_k which is the angle between target moving direction and the global x axis. Besides the global coordinate, a sensor (ego vehicle) coordinate $\mathcal{L}_s$ is defined by $\mathrm{x_s}$ and $\mathrm{y_s}$ axis where $\mathrm{x_s}$ represents the target moving direction and $\mathrm{y_s}$ is the direction perpendicular to $\mathrm{x_s}$. The origin of $\mathcal{S}$ is located at the ego vehicle's position $[x_k, y_k]$.

In this case, the state of the target at time k under the sensor coordinate $\mathcal{L}_s$ is defined as $\mathrm{x}_k^s = [x_k^s, \dot{x}_k^s, y_k^s, \dot{y}_k^s]^T$ (as shown in Figure 4.1 with position $[x_k^s, y_k^s]$). $\mathrm{z}_k^s \in \mathbb{R}^n$ is the measurement provided by the sensor under the sensor coordinate and n is the measurement dimension. The measurement in $\mathcal{L}_s$ can either be the position of the target $\mathrm{z}_k^s = [x_k^s, y_k^s]^T$ or bearing-range $\mathrm{z}_k^s = [l_k^s, \beta_k^s]^T$ (as shown in Figure 4.1) based on the sensor's current position. z_k^s which is associated with the state x_k^s and measurement noise v_k is modelled as:

$$\mathrm{z}_k^s = h(\mathrm{x}_k^s, v_k) \tag{4.1}$$

As mentioned in Chapter 3, h is the general nonlinear measurement function and $v_k \in \mathbb{R}^{n_z}$ is Gaussian noises of the measurement uncertainty described by independent density $p(v_k) = N(0, R)$.

It is also assumed the availability of two kinds of transformation functions $T_g^s(\cdot)$ and $T_s^g(\cdot)$ representing the transformation for the global coordinate to sensor coordinate and sensor coordinate to the global one, respectively. Thus the sensor coordinate target state x_k^s and measurement z_k^s can be converted to the global Cartesian coordinate x_k and z_k in the same form as shown in Chapter 3 equation (3.1) and (3.2) using the transformation function considering the ego vehicle state s_k:

$$\mathrm{x}_k = T_s^g(\mathrm{x}_k^s, \mathrm{s}_k) \tag{4.2}$$

$$\mathrm{z}_k = T_s^g(\mathrm{z}_k^s, \mathrm{s}_k) \tag{4.3}$$

In this case, the target tracking problem using a mobile sensor model is now converted to the standard tracking model formulation in the global coordinate [127] as described in equation (4.21).

4.4.2 Road Constraint

Assuming the road network information is given by a prior digital road map, then the relative road constraints can be accommodated in the MHE-MHT algorithm so as to improve the tracking performance. As mentioned in Chapter 3, the knowledge of road network could also be used as state constraints incorporated in the CMHE algorithm. Therefore in this chapter, road network information is considered as road boundary inequality constraint and the target motion is restricted by these physical constraints as illustrated in Figure 4.2.

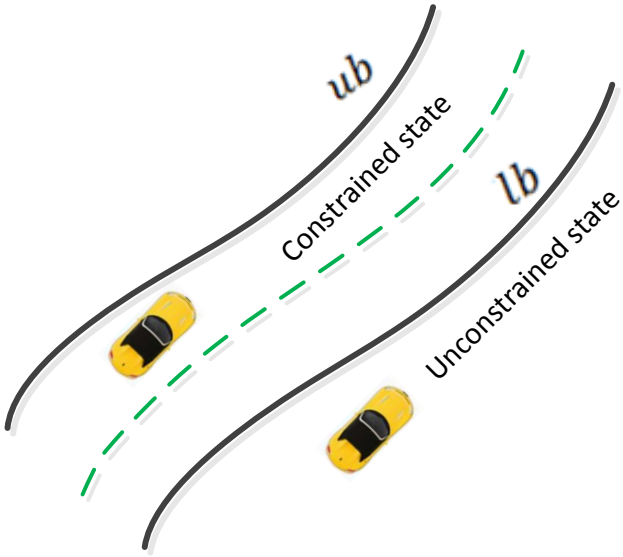


Figure 4.2 Vehicle with road inequality constraint

In order to incorporate road constraints related to different road segments. In this chapter, local coordinates associated to different road segments are used when accommodating the road constraints on the target states. The movement of the target vehicle is constrained on the road network projected on two directions: along and perpendicular to the road curve. Similar to the mobile sensor problem described above, two kinds of coordinate systems are employed. Besides the global Cartesian coordinate $\mathcal{G}$, a local coordinate frame associated to each road segment r is defined by $\mathcal{L}_r$. The origin of $\mathcal{L}_r$ is attached to the starting point of the road predefined in the global coordinate, the x_r axis of $\mathcal{L}_r$ aligns with the continuous curve representing the road, whereas the y_r axis is perpendicular to x_r. Two kinds of transformation

functions are defined as $T_g^r(\cdot)$ and $T_r^g(\cdot)$ to represent the transformation for the global coordinate to road local coordinate and road local coordinate to the global one, respectively. The global coordinate state of the target vehicle $\mathrm{x}_k = [x_k, \dot{x}_k, y_k, \dot{y}_k]^T$ is converted to road coordinate state $\mathrm{x}_k^r = [x_k^r, \dot{x}_k^r, y_k^r, \dot{y}_k^r]^T$ for the following state constraint formulation.

For the state constraint $\{\mathrm{x}_{k-N}^r, .., \mathrm{x}_k^r\} \in C_x$ according to equation (3.14), the road boundary constraint can be represented in (4.4):

$$a_k \leq \mathcal{L}(\mathrm{x}_k^r) \leq b_k \tag{4.4}$$

where $\mathcal{L}$ is the constraint function represented by a full rank $c \times n$ matrix, a_k and b_k are the known vectors each with a dimension of $c \times 1$ representing the lower and upper road boundary individually, c is the number of constraints, n is the number of states. In terms of the geographic information, the constraint function is defined based on different roads types (straight/curved road shown in Chapter 3), and each of the road boundary can be represented by either a first order linear equation (4.5) for straight road or nonlinear equation (4.6) for curved junction under a road local coordinate $\mathcal{L}_r$ based on the digital map.

$$a_r x_k^r + b_r y_k^r + c_r = 0 \tag{4.5}$$

$$a_r (x_k^r)^2 + b_r (y_k^r)^2 + c_r = 0 \tag{4.6}$$

where the coefficient a_r, b_r and c_r are the parameters representing one boundary of the road r ; x_k^r and y_k^r represent the position in road local Cartesian coordinate in x_r and y_r axis respectively.

The road inequality boundary constraints can then be presented by linear/nonlinear programming problems [109] defined by the inequality constrains below:

$$lb_r \leq A_r x_k^r + B_r y_k^r \leq ub_r \tag{4.7}$$

$$lb_r \leq A_r (x_k^r)^2 + B_r (y_k^r)^2 \leq ub_r \tag{4.8}$$

where A_r and B_r are the normalized parameters according to equation (4.5) and (4.6) respectively; lb_r and ub_r represents the lower and upper boundary for road r.

Due to the road boundary constraints, the vehicle motion uncertainty in the road local coordinate x_r direction and y_r direction is unequal. The maneuver along the road direction (x_r) has more uncertainty than one in orthogonal direction (y_r). For model based target tracking problems covered in this thesis, the motion uncertainty is represented by the process noise shown in (3.1). In this case the system process noise ω_k with the covariance matrix $\sigma_\omega = diag\{\sigma_x, \sigma_y\}$ defined in global coordinate x and y axis need to be converted to the road local coordinate $\mathcal{L}_r$ (according to different road *r*) by the transformation function $T_r^g(\cdot)$ denoted as ω_k^r. Thus, the process noise covariance matrix along the road direction and orthogonal direction is defined as $\sigma_\omega^r = diag\{\sigma_{x_r}^r, \sigma_{y_r}^r\}$.

4.5 MHE-MHT Structure

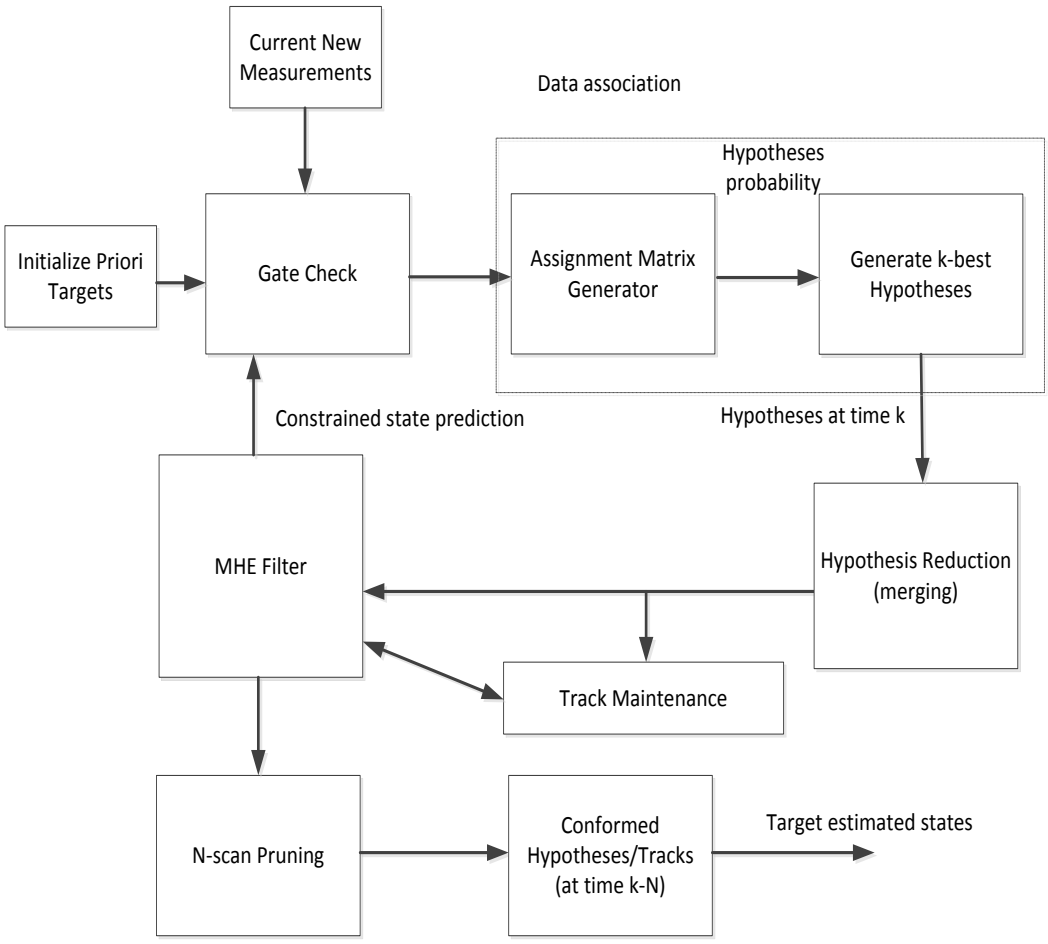


Figure 4.3 Flow diagram of MHE-MHT algorithm

The derivation of the standard MHT structure is presented in detail in Chapter 2, which contains three main processes: hypothesis generation, probability calculation and hypothesis reduction.

The performance of the MHT filter heavily depends on the particular implementation of gating and pruning techniques [58] that are *ad-hoc* in general and requires a careful design of the structures and algorithm. In this case, this section focus on the structure development for the proposed MHE-MHT. In MHE-MHT, in order to explicitly deal with environmental constraints based extra domain knowledge, modifications have been made comparing with the original MHT structure and the details are expliained below. The flow diagram of the MHE-MHT algorithm is presented in Figure 4.3. The high-level pseudo-code for MHE-MHT is provided in the Appendix B. The formation of MHE-MHT structure is set forth explicitly:

Constrained state prediction

In (E)KF, the state prediction is calculated below, following (2.18):

$$\hat{\mathrm{x}}_{k|k-1} = F\hat{\mathrm{x}}_{k-1|k-1} \tag{4.9}$$

where $\hat{\mathrm{x}}_{k-1|k-1}$ is the state estimate at time $k-1$; $\hat{\mathrm{x}}_{k|k-1}$ is the predicted state at time *k*; F is a linear state dynamic function. Since there is no process noise or measurement noise, the *a priori* predicted state $\hat{\mathrm{x}}_{k|k-1}$ is equivalent to the *a posteriori* state estimate $\hat{\mathrm{x}}_{k|k}$ at time *k*, considering a perfect measurement case where $\mathrm{z}_k = \hat{\mathrm{z}}_{k|k-1}$. However, as discussed in previous chapters, in realistic target tracking problems the data received by sensors is affected by measurement noise and other forms of interference. In this case the inherent uncertainties are always with measurements and the so called 'perfect measurement' is only considered when no actually measurement is received. This is also known as the missed measurement issue mentioned in Section 3.3.3, Chapter 3.

Due to the road boundary considered in this chapter, the road inequality constraints (4.4) are also accommodated in the predicted state. In MHE framework, the missed measurement issue is presumed as a one-step constrained state prediction and the additional road constraints can be naturally accommodated in the measurement using optimization based MHE quadratic programming cost function.

In addition to the state estimation process, the constrained state prediction is also utilised in *gate check* and *data association* process in the MHE-MHT framework as discussed below. To solve this problem, the predicted state is projected within the road boundaries using estimation projection method with the active set approach in this chapter.

The position of any off road unconstrained state prediction $\hat{\mathrm{x}}_{k|k-1}$ is projected onto the nearest road boundary (upper or lower). To deal with inequality constraint, an active set method uses the fact that it is only those constraints that are active at the solution of the problem that are significant in the optimality conditions. Suppose that there are c inequality constraints in (4,4), and q of the c inequality constraints are active at time k. Denote by $\tilde{C}_k$ the q rows of C_k that correspond to the active constraints, and denote by a_k or b_k that correspond to the constraint vector representing the lower or upper boundary respectively. The constrained prediction $\tilde{\mathrm{x}}_{k|k-1}$ can therefore be written as a solution of the equality constrained problem:

$$min_{\tilde{\mathrm{x}}_{k|k-1}}\left(\tilde{\mathrm{x}}_{k|k-1} - \hat{\mathrm{x}}_{k|k-1}\right) W\left(\tilde{\mathrm{x}}_{k|k-1} - \hat{\mathrm{x}}_{k|k-1}\right) \tag{4.10}$$

such that

$$\tilde{C}_k \tilde{\mathrm{x}}_{k|k-1} = d_k = \begin{cases} a_k \\ b_k \end{cases} \tag{4.11}$$

where W is a positive-definite weighting matrix. The inequality constrained problem in (4.4) is equivalent to the equality constrained problem (4.11). Therefore the estimate projective method [106] is used giving the following solution:

$$\tilde{\mathrm{x}}_{k|k-1} = \hat{\mathrm{x}}_{k|k-1} - W^{-1}\tilde{C}_k{}^T\left(\tilde{C}_k W^{-1}\tilde{C}_k{}^T\right)^{-1}(\tilde{C}_k\hat{\mathrm{x}}_{k|k-1} - d_k) \tag{4.12}$$

In this work, we set $W = I$ to obtain the least squares estimate of the state constraints.

According to [106], the projected process error covariance matrix $\tilde{Q}$ can be expressed:

$$\tilde{Q} = (I - Q^{-1}\tilde{C}^T\left(\tilde{C}R^{-1}\tilde{C}^T\right)^{-1}\tilde{C})R(I - R^{-1}\tilde{C}^T\left(\tilde{C}R^{-1}\tilde{C}^T\right)^{-1}\tilde{C})^T \tag{4.13}$$

Gate Check

Similar to the standard MHT framework shown in Chapter 2, gate check is carried out

by calculating the *Mahalanobis distance* between the predicted target position and the current measurements. The prediction of target position is done by constrained (E)KF prediction update as mentioned above and only the measurements whose *Mahalanobis distances* with particular targets are smaller than a particular threshold are used for the further data association.

$$\left(\mathrm{z}_k^m - \tilde{\mathrm{x}}_{k|k-1}^p\right)^T \tilde{S}_{k/k-1}\left(\mathrm{z}_k^m - \tilde{\mathrm{x}}_{k|k-1}^p\right) \leq Gating \tag{4.14}$$

where z_k^m is the position measurement *m* at time *k*, $\tilde{\mathrm{x}}_{k|k-1}^p$ is the constrained predicted target position calculated by (4.12) and $\tilde{S}_{k/k-1}$ is the constrained covariance of innovation vector which can be calculated according to (2.23) using (4.13). $Gating$ is a matrix of binary values which indicates maximum possible distance between measurement and targets. Only the measurements inside the gate are considered for assignment.

Data Association

MHE-MHT implements similar data association process as the Cox's algorithm [58] which has been explained in Chapter 2. The assignment matrix is generated to represent all possible target-to-measurement associations. Then each new hypothesis contains a set of potential target-to-measurement assignments, leading to an exhaustive approach of enumerating all the possible assignment combinations. To solve this problem, the Murty's algorithm is used to find the *k*-best assignment/ hypotheses generated from each parent hypothesis.

In MHE-MHT, the evaluation of alternative hypotheses formation is based on a probabilistic expression, the hypotheses probability $p\left(\theta_k^l\right)$ mentioned in section 4.3. Here $\theta_k^l = \{\theta_{k-1}^i, \Theta_k\}$ denotes a new hypothesis generated at time *k* which combines a relative past hypothesis θ_{k-1}^i and a currently generated assignment Θ_k. Here Θ_k is one of the assignment combination generated by the assignment matrix (Table 2.1). Based on the Bayes' rule, the probability $p\left(\theta_k^l|\mathrm{Z}_{0:\mathrm{k}}\right)$ of hypothesis θ_k^l considering a sequence of past measurements $Z_{0:k}$ is represented as:

$$p\left(\theta_k^l|\mathrm{Z}_{0:\mathrm{k}}\right) = p\left(\theta_{k-1}^i, \Theta_k|\mathrm{Z}_{0:\mathrm{k}}\right) \tag{4.15}$$

$$= \frac{1}{c} p\left(\mathrm{z}_\mathrm{k}|\theta_{k-1}^i, \Theta_k, \mathrm{Z}_{0:\mathrm{k}-1}\right) p\left(\Theta_k|\theta_{k-1}^i, \mathrm{Z}_{0:\mathrm{k}}\right) p\left(\theta_{k-1}^i\middle|, \mathrm{Z}_{0:\mathrm{k}-1}\right)$$

where c is a normalization constant; $p\left(\mathrm{z_k}|\theta_{k-1}^{i},\Theta_k,\mathrm{Z_{0:k-1}}\right)$ represents the measurement likelihood function corresponding to current assignment Θ_k ; $p\left(\Theta_k|\theta_{k-1}^{i},\mathrm{Z_{0:k}}\right)$ is the assignment probability representing the current data association certainty; the last term of (4.15), $p\left(\theta_{k-1}^{i}\middle|,\mathrm{Z_{0:k-1}}\right)$, represents the probability of the parent global hypothesis and is therefore available from the previous iteration.

Assuming at time k there are $j = 1 \ldots n$ targets existed from previous hypothesis θ_{k-1}^{i}. Among the j targets, J_D^i targets are detected while J_{ND}^i targets are non-detected according to the current measurement set $\mathrm{Z_k} = \left\{\mathrm{z}_k^i\right\}_{i=1}^{m_k}$ and assignment Θ_k. Thus the current *a posteriori* hypothesis probability (4.15) can be represented as:

$$p\left(\theta_k^l|\mathrm{Z_{0:k}}\right) \propto \tag{4.16}$$

$$\underbrace{\beta_{FA}^{(m_k^{FA})}\beta_{NT}^{(m_k^{NT})}\left[\prod_{j\in J_D^i}\frac{p_D^j p_{k/k-1}^j\left(\mathrm{z}_k^{\Theta_k^{-1}(j)}\right)}{\left(1-p_D^j p_G^j\right)}\right]}_{\textbf{Maximied by Assignment Problem(Murty's)}}\underbrace{C_i p\left(\theta_{k-1}^{i}|\mathrm{Z_{0:k-1}}\right)}_{\textbf{Previous Hypothesis probability}}$$

where m_k^{FA} is the number of false alarms in gate region at time k while the density of false alarms is β_{FA}(number of FAs/area/scan) ; m_k^{NT} is the number of new targets in gate region at time k and the density of new targets is β_{NT}(number of NTs/area/scan). P_D^j is the detection probability of the jth target and the gate probability of the jth target is P_G^j.

The expression in (4.16) can be further simplified by taking a logarithm transformation. The result is shown below in (4.17):

$$\log p\left(\theta_k^l|\mathrm{Z_{0:}}_k\right) = m_k^{FA}\log\beta_{FA} + m_k^{NT}\log\beta_{NT} + \tag{4.17}$$

$$\sum_{j\in J_D}\log\frac{p_D^j p_{k/k-1}^j\left(\mathrm{z}_k^{\Theta_k^{-1}(j)}\right)}{\left(1-p_D^j p_G^j\right)} + \log p\left(\theta_{k-1}^{i}|\mathrm{Z_{0:}}_{k-1}\right)$$

Here $p_{k/k-1}^j\left(\mathrm{z}_k^{\Theta_k^{-1}(j)}\right)$ is the predicted measurement density of jth target based on measurements z_k at time k and the current assignment $\Theta_k(j)$. In Bayesian framework

it can be calculated by the innovation likelihood function represented by a normal distribution function:

$$p^{j}_{k/k-1}\left(\mathrm{z}_k^{\Theta_k^{-1}(j)}\right) \triangleq \mathcal{N}\left(\mathrm{z}_k; \hat{\mathrm{z}}_{k/k-1}, S_{k/k-1}\right) \tag{4.18}$$

where $\hat{\mathrm{z}}_{k/k-1}$ denotes the predicted measurement(position) for target *j* and $S_{k/k-1}$ is the corresponding associated innovation covariance. Both $\hat{\mathrm{z}}_{k/k-1}$ and $S_{k/k-1}$ are calculated using (E)KF in the standard MHT. However in the constrained MHE-MHT, the constrained position prediction $\tilde{\mathrm{x}}^{p}_{k|k-1}$ and constrained innovation covariance $\tilde{S}_{k/k-1}$ as discussed above in (4.12) and (4.13) are used instead shown below:

$$p^{j}_{k/k-1}\left(\mathrm{z}_k^{\Theta_k^{-1}(j)}\right) \triangleq \mathcal{N}\left(\mathrm{z}_k; \tilde{\mathrm{x}}^{p}_{k|k-1}, \tilde{S}_{k/k-1}\right) \tag{4.19}$$

As a result, $p^{j}_{k/k-1}\left(\mathrm{z}_k^{\Theta_k^{-1}(j)}\right)$ can be calculated using the multivariate normal distribution density function:

$$p^{j}_{k/k-1}\left(\mathrm{z}_k^{\Theta_k^{-1}(j)}\right) = \det\left((2\pi)^{d}\tilde{S}_{k/k-1}\right)^{-\frac{1}{2}} \tag{4.20}$$

$$\exp\left(-\frac{1}{2}\left(\mathrm{z}_k - \tilde{\mathrm{x}}^{p}_{k|k-1}\right)^{T}{\tilde{S}_{k/k-1}}^{-1}\left(\mathrm{z}_k - \tilde{\mathrm{x}}^{p}_{k|k-1}\right)\right)$$

It is impractical to enumerate all possible global hypotheses and calculate the probability for each of the hypothesis. In the following part of this report, an improved pruning method is implemented in MHE-MHT framework.

MHE Filter

The details about implementing CMHE for road constrained target tracking have been discussed in Chapter 3. In the generic MHT, the 'filter' process is based on KF including two individual steps: prediction update and measurement update. However, the two steps are combined in MHE and solved directly by optimisation solver. In MHE, the state estimation is determined online by considering a finite horizon of latest measurements. The filtering process would be similar to KF if measurements are always observed and updated with the target. However, a problem arises in MHE when missed measurements (temporarily target missing) happens among a horizon of

measurements. Similar to the way that is used in Chapter 3, in the proposed MHE-MHT, any missed measurement is represented by the constrained predicted position $\tilde{\mathrm{x}}_{k|k-1}$ calculated by (4.12) assuming no process noise ω_j and measurement noise v_j at current step.

Target Maintenance:

For ground target tracking scenarios, vehicles may enter or leave the surveillance field of view during the tracking process. Moreover, occlusion or miss detection is also possible when a vehicle is hidden behind another one. In order to achieve a fully functional tracking algorithm, a target maintenance logic is developed for in the MHE-MHT structure. Basically, there are three possible states for a set of targets in this logic: target initiation, confirmation/deletion and maintenance. The targets present at a time step are a combination of existing targets from the parent tracks and any new targets resulting from the set of measurement associations. For any target *n* in existence at time *k*-1, the possible associations at time *k* are shown:

- Target initiation: If the measurement is associated with a new target *n* and the relevant hypothesis is selected from the current hypotheses tree. Add a *lifetime* index l to the target with value one and the relevant time step t_i^n is recorded.
- Target confirmation/deletion: The new target is confirmed only if the detected target appears along the same track over a consecutive iteration of *Ct* (confirmation threshold) times. Once the tentative target is confirmed, the time step is recorded as t_c^n. The *lifetime* index is accumulated by one whenever the tentative target is detected but not over *Ct*. On the contrary, the *lifetime* index for any existing target is reduced by one whenever the target is not associated with the current measurement and will be permanently deleted from target list when the *lifetime* is zero. The time step is also recorded as t_d^n.
- Target maintenance: The confirmed target may be temporally occluded or undetected by the sensor without measurements being associated. For this situation, the track is updated according to the predicted position calculated by KF of the target last associated states.

Correspondingly, the high level logic for MHE-MHT target maintenance is shown in Table 4.1:

Table 4.1 High level logic for MHE-MHT target maintenance

-- At time k, for nExistedTarg number of detected target in a hypothesis
For n=1: nExistedTarg
(Case one: permanently deleted targets)
If Lifetime $l^n == 0$
Continue; (the target is permanently deleted/already disappeared)
End
(Case two: target maintenance—target updating with measurement or temporally miss detection)
If Targ≠asso (Target not associated with current measurement)
$l^n = l^n - 1$;
If $l^n > 0$ *(Target temporally miss detection)*
Implement KF prediction with road constraint for CMHE estimation result;
Else (Target permanently deleted from target tree)
Deletion time $t_d^n = k$;
End
Else (Target associated with current measurement)
Implement CMHE estimation;
If $l^n <$ *Conformation threshold Ct*
$l^n = l^n + 1$;
If $l^n == Ct$*; (Tentative target confirmed)*
$t_c^n = k$;
End
Else
$l^n = Ct$;
End
End
(Case three: Target initiation)
-- At time k, for nNewTarg number of detected new target
For i=1: nNewTarg (measurement is associated to a new target)
Use current measurement as initial position;
Initiation time $t_i^n = k$;
$l^i = 1$;
End

m-best *N* Scan Pruning

As mentioned above, hypothesis pruning or merging process is essential for MHT to reduce the computational cost due to exponential increase in the number of track hypotheses over time. This thesis considers on road vehicle tracking scenarios in relatively low target density environment. In this case, an *m*-best *N* scan pruning approach is proposed by combining both the standard *N*-scan pruning and m-best merging approaches. This enables us to minimize the computational load for real time tracking while maintains a relatively high tracking continuity and accuracy.

The key principle of the proposed method is that difficult data association decisions are deferred until more data are received which also matches the fundamental principle in MHE. The continued growth of the hypotheses is controlled by keeping only the *N* latest scans in the hypothesis trees. At each time step, only *k* global hypotheses (among all the generated hypotheses) with the highest probabilities are kept. The scan number *N* here is chosen as the same value of the horizon length in MHE. Thus, the association uncertainty at time *k*-*N* is resolved by the best hypothesis given at time *k*. In the meanwhile, the estimation process considers a horizon length of measurements within the last *N* scans. A lager *N* and *k* implies a deeper window gap and a wider range of different data associations hence the solution might be more accurate, but makes the running time longer. The number should be selected corresponding to different tracking scenarios.

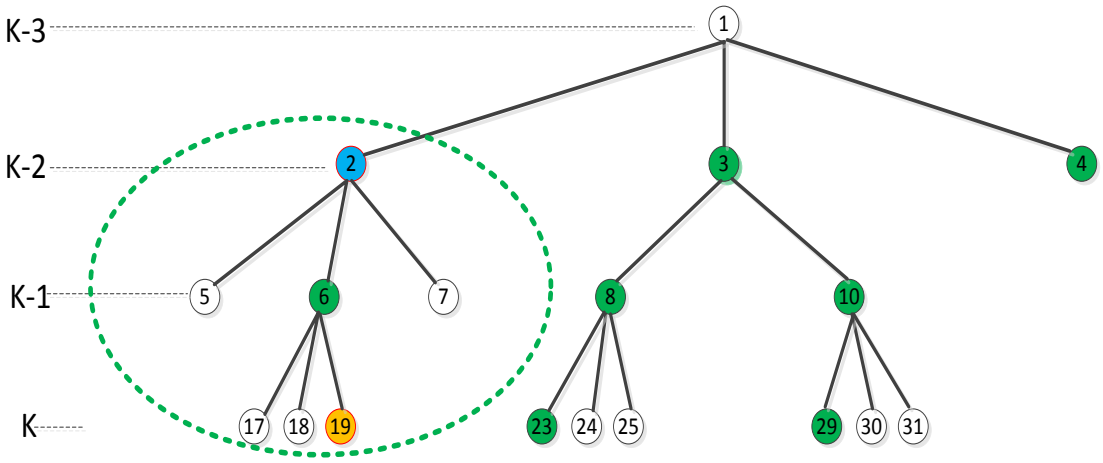


Figure 4.4 *m*-best *N* scan pruning

An example is shown in Figure 4.4 with a 3-best 2 scan pruning. At each time step, 3 hypotheses are generated from each of the parent hypothesis using the Murty's algorithm with relatively highest probability. Then all the hypotheses generated at the same step are compared and only the best 3 among them all are kept in the hypothesis tree. Since hypothesis 19 has the highest probability among 19, 23, and 29 at time step *k*, its origin at step *k*-2 is chosen for states update in filtering procedure which is hypothesis 2 in this case.

4.6 Simulations

In this section, two simulation examples are presented in the context of autonomous vehicle MTT. The first example is a MTT using a linear measurement model. The second one is based on a nonlinear measurement model. The road constraints generated from maps are considered in both scenarios.

4.6.1 MHE-MHT with Linear Measurement Model

In this section, a MTT scenario is considered using linear measurement models with road boundary constraints. As shown in Figure 4.5, consider a two-dimensional (2-D) unsupervised crossroad scenario with four vehicles observed in clutter over the surveillance of $100\text{m} \times 100\text{m}$ (meters). Four vehicles start moving with initial position at (0m, 40m), (100m, 45m), (40m, 0m) and (45m, 100m) respectively. Each vehicle is moving on a single carriage way starting from a position of the middle of the road. The road is assumed to have a total width of 4 meters and the vehicle's trajectory is limited within the road width constraint. All the vehicles have a same initial speed of 9m/s in a straight direction along the road network. The road speed limit is set as 10m/s. The autonomous ego vehicle starts moving from position (0m, 40m) 5 seconds after vehicle 3 with a speed of 1m/s. It is assumed that the ego vehicle is equipped with on-board radar which has an observation angle of 180 degree and a long detection range of 150m.

The vehicle dynamics is similar to the CV model shown in (3.22) under the global coordinate $\mathcal{G}$ with the state vector $\mathrm{x}_k = [x_k, \dot{x}_k, y_k, \dot{y}_k]^T$. Given road model r, the process noise ω_k^r is defined under the road local coordinate $\mathcal{L}_r$ which is a two two-dimensional Gaussian process noise with zero mean and covariance matrix

$\sigma_{\omega}^{r} = diag\{10, 5\}$ in $\mathcal{L}_r$ where $diag\{.\}$ represents a diagonal matrix. This covariance represents higher motion uncertainty along the centre line direction and smaller uncertainty orthogonal to the road. Considering the road inequality constraint defined in (4.7) with state $\mathrm{x}_k^r = [x_k^r, \dot{x}_k^r, y_k^r, \dot{y}_k^r]^T$ under $\mathcal{L}_r$, the global dynamic function (3.23) can be written as:

$$\mathrm{x}_{k+1} = T_r^g(F \cdot T_g^r(\mathrm{x}_k) + G \cdot \omega_k^r) \tag{4.21}$$

The vehicle measurement z_k^s model is a linear matrix with $\mathrm{x_s}$ and $\mathrm{y_s}$ potion under the sensor coordinate $\mathcal{L}_s$:

$$\mathrm{z}_k^s = h(\mathrm{x}_k^s, v_k) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \mathrm{x}_k^s + v_k \tag{4.22}$$

where v_k is a Gaussian noise with covariance matrix R = $diag\{25, 25\}$ under $\mathcal{L}_s$. In order to keep the tracking consistency, the measurement z_k^s is then converted to the global Cartesian coordinate $\mathcal{G}$ measurement z_k using the transformation function (4.3).

Each target is detected with a probability of $P_d = 0.98$. The detected measurements are immersed in a clutter environment that can be modelled as a Poisson distribution with clutter density of $\beta_{FA} = 1.2 \times 10^{-3}$ (false alarms/area/scan) over the $10^4 m^2$ region (i.e., 12 clutter returns over the surveillance region at each scan). The sampling time interval is T=0.1s. New target density is $\beta_{NT} = 2 * 10^{-4}$ and the gate region $P_G = 0.97$.

It is worth to mention that if the collision volume is regardless for each vehicle, occlusions happen when two vehicles cross each other and more likely when one vehicle is in the line of sight of another vehicle (depending on the observer angle and direction) which makes the tracking problem more challenging.

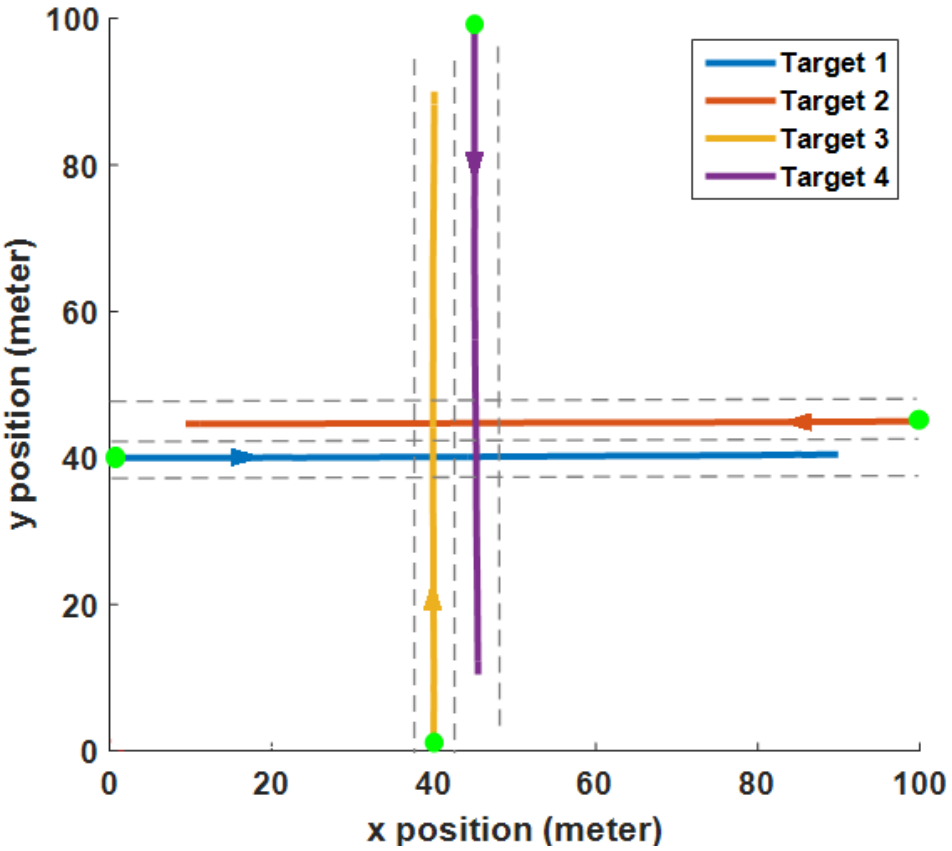


Figure 4.5 Multiple target trajectories for scenario 1

The position estimates between generic KF base MHT (KF-MHT) (Figure 4.6), and constrained MHE-MHT (CMHE-MHT) (Figure 4.7) are demonstrated in a cluttered environment with false alarms and missed detection (i.e., 12 false alarms return over the surveillance region and one miss detection at k=99). It can be shown that the road boundary and speed limit constraints play a significant part in improving the tracking accuracy. Due to the inequality state constraints, the estimation results are limited within the road under the speed limit. For further comparing different algorithms, 100 trials of Monte-Carlo simulations are performed. The performances of different algorithms are measured using the root mean-square error (RMSE). As shown in Table 4.2, the CMHE-MHT gives better tracking results for all four targets by introducing both the road boundary and speed limit constraint.

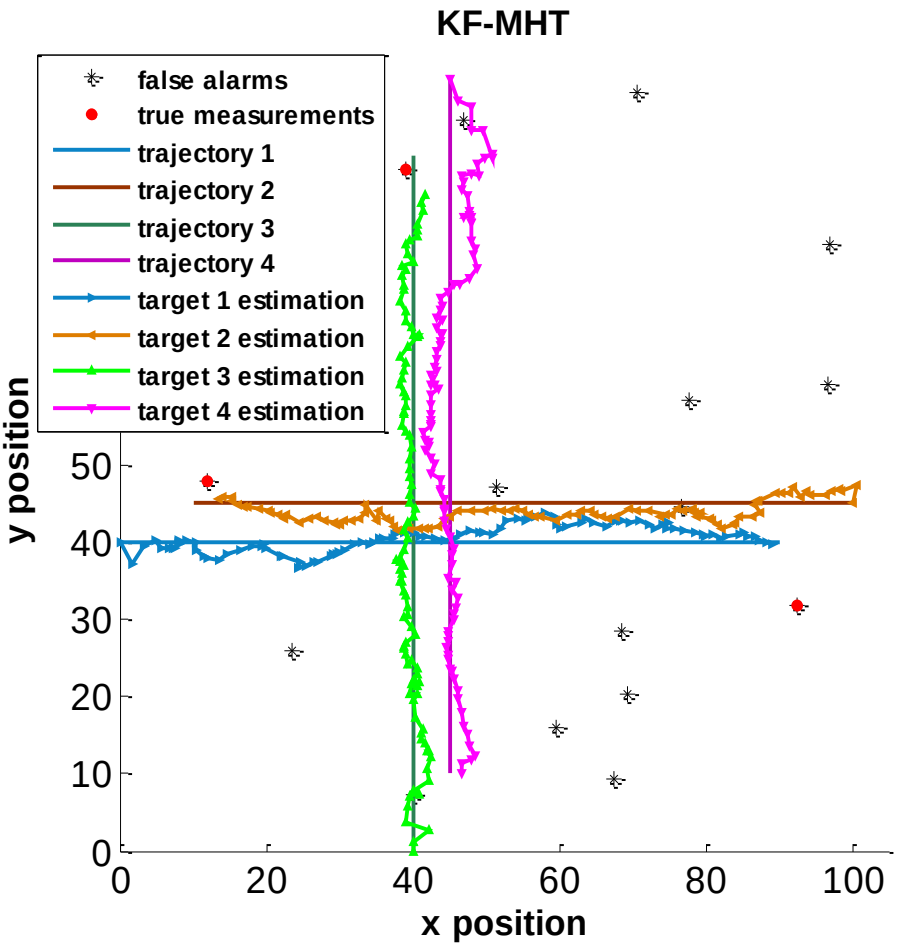


Figure 4.6 KF-MHT tracking result for scenario 1(meter)

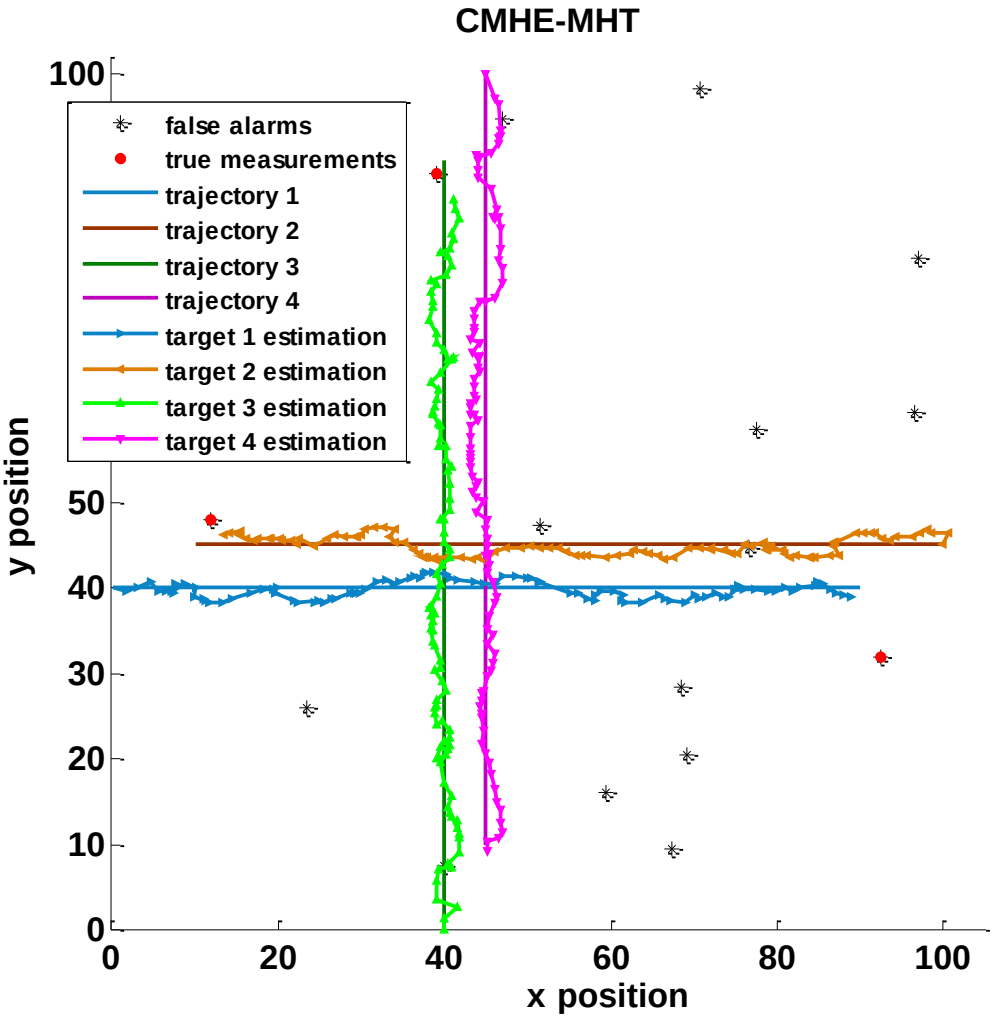


Figure 4.7 CMHE-MHT tracking result for scenario 1 (meter)

Table 4.2 Averaged RMSEs for four vehicles by different approaches

RMSE(m)	Overall RMSE position	Target 1	Target 2	Target 3	Target 4
KF-MHT	4.7359	3.6943	4.8592	4.9702	5.4200
CMHE-MHT(horizon=4)	2.4884	2.1009	2.1685	2.6053	3.0790

To further analyse the proposed algorithm in terms of data association accuracy and tracking continuity, the target maintenance logic is implemented by setting the lifetime threshold *Ct* as 4 with a relatively low detection probability $P_d = 0.60$. The total tracking life time is 99 time steps. The *m*-best N scan pruning technique is used with *m*=5 meaning only five best global hypotheses are kept at each scan and N=4 for N-scan pruning. The tracking result is defined as success only if all of the four tracks are maintained during the tracking process. The CMHE-MHT algorithm with different horizon length (N=4 and N=8) are also compared and the simulation results are shown in Table 4.3.

Table 4.3 Data association simulation results

Estimation Indexes	KF-MHT	CMHE-MHT(N=4)	CMHE-MHT(N=8)
Tracking success rate (all four targets)	67%	96%	98%
Average number of tracks (4 true tracks)	7.3	4.3	4.1
Average true track life (99 steps in total)	78.2	97.8	99
Average RMSE for position (m)	5.1926	2.5764	2.3882

From the results, it can be seen that the proposed CMHE-MHT method improves both in the tracking quality and continuity. The successful rate proves that the CMHE-MHT is capable of steadily tracking all four targets when using an appropriate horizon length and maintenance logic parameters. The successful rate is extremely high in this tracking scenario because a relatively low clutter density is considered and most of the false alarms are rejected by implementing gating with constrained state prediction. The constrained position prediction also provides a more realistic and reliable substitution measurement when miss detection happens. Most tracking

failures are caused by the occlusion where two vehicles cross each other. When no road constraints are used, the general KF may generate off road position estimate which could be associated with false alarms instead of the true measurement. As a result, the incorrect data association results generate the redundant false tracks and fail in tracking the true target after occlusion. The higher average number of tracks and average RMSE implies that the generic KF-MHT does not provide a stable data association and accurate state estimation comparing with the proposed method. On the other hand, the road knowledge aided CMHE-MHT algorithm can track the moving objects robustly. The use of map information can effectively reduce the number of incorrect assignments and, as a result, the number of false hypotheses and targets are also minimized.

A brief computational cost comparison between KF-MHT and MHE-MHT is also considered in this work. In order to exclude the effect of the structure complicity of the proposed method, the two algorithms are compared under the same structure of generic MHT with same parameters. In this case, the main computational difference comes from the state estimation step, which then becomes a comparison between KF and linear MHE. For linear and Gaussian target dynamic and measurement model used in this simulation, the 'quadprog' optimization toolbox in MATLAB software is used to solve the constrained linear optimization for MHE. The computational time for MHE-MHT is almost identical as KF-MHT when using a horizon length of 1. It is only mildly increased when using a longer horizon length of 4 and suitable for real time application in our simulation. It takes about double computational time when using a horizon length of 8. Considering the relatively small estimation improvement (shown in Table 3.1 and Table 4.3) while significant increase of computational cost, we would suggest using a horizon length of 4 for real time applications and the rest of work in this Chapter.

4.6.2 MHE-MHT with Nonlinear Measurement Model

To further analysis the proposed algorithm, in the second example, a more challenging MTT simulation for an interacting road scenario using a nonlinear Gaussian measurement model is set up. As illustrated in Figure 4.8, the surveillance region of a two-dimensional scenario is $[-1000\text{m}, 1000\text{m}] \times [-1000\text{m}, 1000\text{m}]$ with an unknown and time varying number of targets observed in clutter environment.

The vehicle dynamics and relative state vector are the same as Scenario 1 with a two-dimensional Gaussian process noise which has a covariance matrix $Q = diag\{25,16\}$. The sampling interval is $T = 1$ and the total simulation time is *k*=100.

Initially, two targets start moving in the environment with initial position$(250m, 250m)$ and $(-250m, -250m)$ respectively. Each vehicle is moving on a single carriage way starting from a position on the middle of the road. The road is assumed to have a total width of 4 meters and the vehicle's trajectory is limited within the road width constraint. The initial speed of two vehicles is 12m/s in along each road network. The road speed limit is set as 14m/s. The target initial covariance is defined as $P_0 = diag[100,25,100,25]^T$ for both two targets. The two vehicles cross each other at *k*=53 when tracking occlusion happens. A new target appears at time *k*=66 with an initial velocity of 21m/s and the speed limit on road 3 is 23m/s. The ego vehicle starts to follow target 1 eight seconds later. It is assumed that the ego vehicle has a full range detection ability of all three targets during the whole tracking process.

The measurement model $z_k^s = h(x_k^s, v_k)$ is defined as a nonlinear range and bearing model shown in (3.20). v_k is a two-dimensional Gaussian zero-mean measurement noise with a covariance matrix $R = diag\{25, 2.5^{-3}\}$ under $\mathcal{L}_s$. The measurement z_k^s is also converted to the global Cartesian coordinate $\mathcal{G}$ measurement z_k using the transformation function (4.3). Each target is detected with a probability of $P_d = 0.98$. The detected measurements are immersed in high clutter environment with clutter density of β_{FA}=12.5×10^{-6} over the $4 \times 10^6 m^2$ surveillance region (i.e., 50 false alarms return over the surveillance region).

The lifetime threshold *Ct* is defined as 5 in the CMHE-MHT implementation, which means any new target can only be confirmed if successfully detected in 5 consecutive time steps. Similarly, tracking any existing target will be terminated after miss detection of 5 sequential time steps. The horizon length used in the MHE is set as 4 and so as for N-scan pruning. Since only a small number of targets are considered in this study, at each time step, 3 best hypotheses are kept so as to reduce the computational cost.

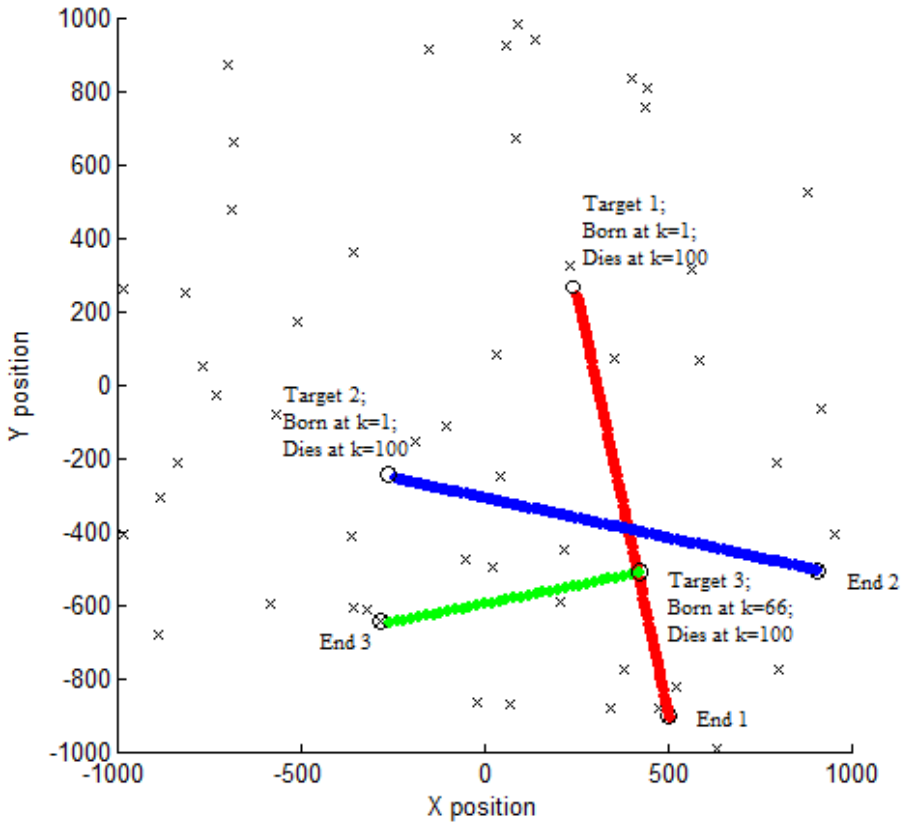


Figure 4.8 Multiple target trajectories for scenario 2 (meter)

An example of the overall tracking performance of three different algorithms, (nonlinear)MHT [58] (Figure 4.9), GM-PHD [66] (Figure 4.10) and CMHE-MHT (Figure 4.11) are shown below. As proposed in [66], the GM-PHD recursion is a closed-form solution to the PHD recursion, which is proposed for jointly estimating the time-varying number of targets and their states from a sequence of noisy measurements sets in the presence of data association uncertainty, clutters and missed measurements. In contrast to explicit data association methods such as MHT, the posterior PHD function is approximated by a sum of of weighted Gaussian components for all target candidates whose weights, means and covariance are propagated analytically in time. In original GM-PHD [66], the mean and covariance of each Gaussian components are propagated by KF. The mean disadvantage of GM-PHD is that it does not provide identities of individual target state estimates which are needed in this study for constructing tracks of each individual targets. To solve this problem, in this study, a separately method [128] is implemented to directly determine the state trajectories of the individual targets from the evolution of the Gaussian

mixture. In this case, the solution provides not only the state estimates of targets at each time step but also association amongst state esitmates of targets over time so that estimates and labels of state trajectories for individual targets can be obtained.

The GM-PHD filter has been extended to accommodate non-linear target dynamic models [129] using EKF. In this case, the generic MHT and GM-PHD used in this study are based on the EKF filter. For mildly nonlinear problems such as this example, the EKF provides good approximations and the performance gap between EKF and other nonlinear filters such as UKF, PF may not be noticeable. It is not hard to tell that the proposed CMHE-MHT gives the best tracking performance by comparing these three figures.

The (nonlinear)MHT successfully picks up most of the target positions of Target 1 and Target 2 in the front half of the trajectory 1 and 2. However the tracking performance decays rapidly after two targets cross each other at k=53 where tracking occlusion happens. The (nonlinear)MHT fails to continue tracking Target 2 while quite a few false tracking points are picked up which are obviously off road. (nonlinear)MHT successfully picks up the new target appears on trajectory 3, however the position estimate is far away from the true trajectory and wrong tracks are also picked up.

The GM-PHD filter outputs fewer false tracks than the generic KF-MHT filter during the whole tracking process. All three targets are successfully tracked in this case, however the position estimate are not very accurate with a lot of off road estimates. Target 3 is directly picked up just after its appearance, however there are a few wrong tracks picked up after tracking occlusion at k=53 and new target appearance at k=66 that suggest not very stable association performance.

Lastly, the proposed CMHE-MHT gives a relatively more accurate and stable tracking performance comparing with the other two filters. The three targets are not only successfully tracked but also with very accurate position estimation (within road boundaries). CMHE-MHT outputs least false tracks during the whole tracking process and relatively accurate and stable data association after tracking occlusion problem. It should be noted that for CMHE-MHT to pick up target tracks, the target must be present in the scene for at least a number of 5 time steps so as to confirm a new track hypothesis. The results are also presented with a horizon length of 5 steps delay. In this case, it gives the slowest reaction to the new target appearance.

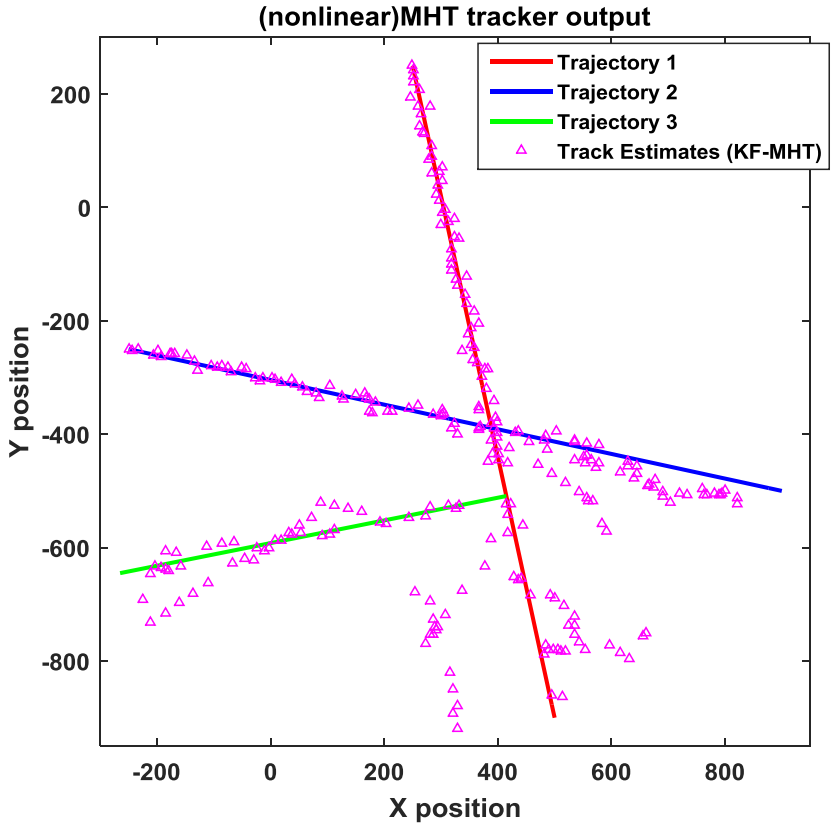


Figure 4.9 Trajectories given by (nonlinear) MHT tracker for scenario 2 (m)

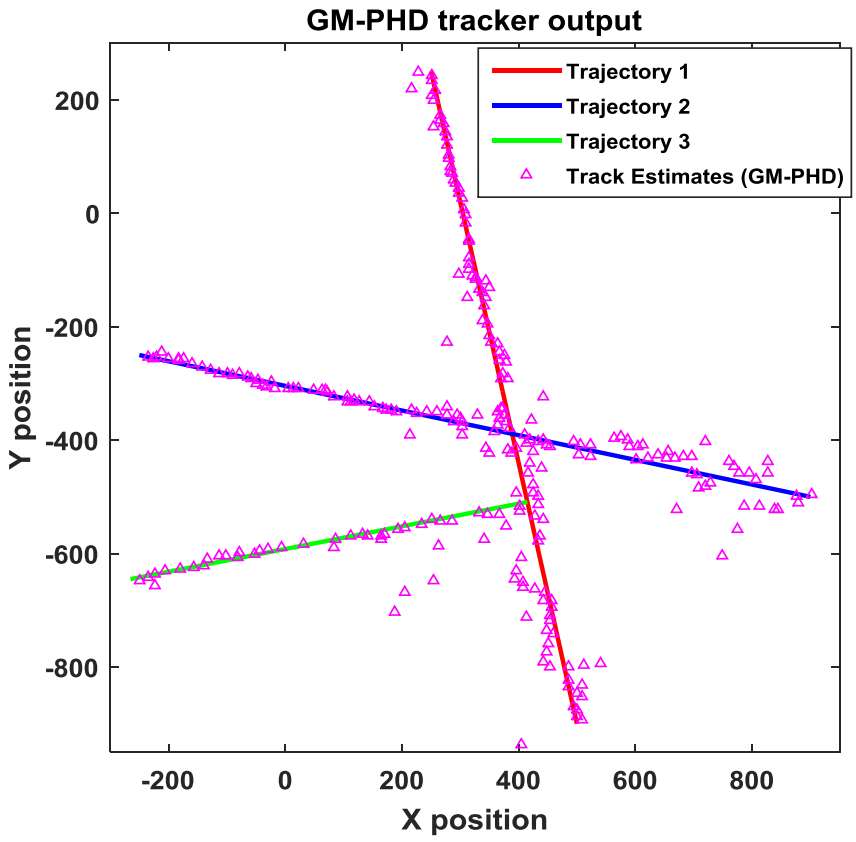


Figure 4.10 Trajectories given by the GM-PHD tracker for scenario 2 (m)

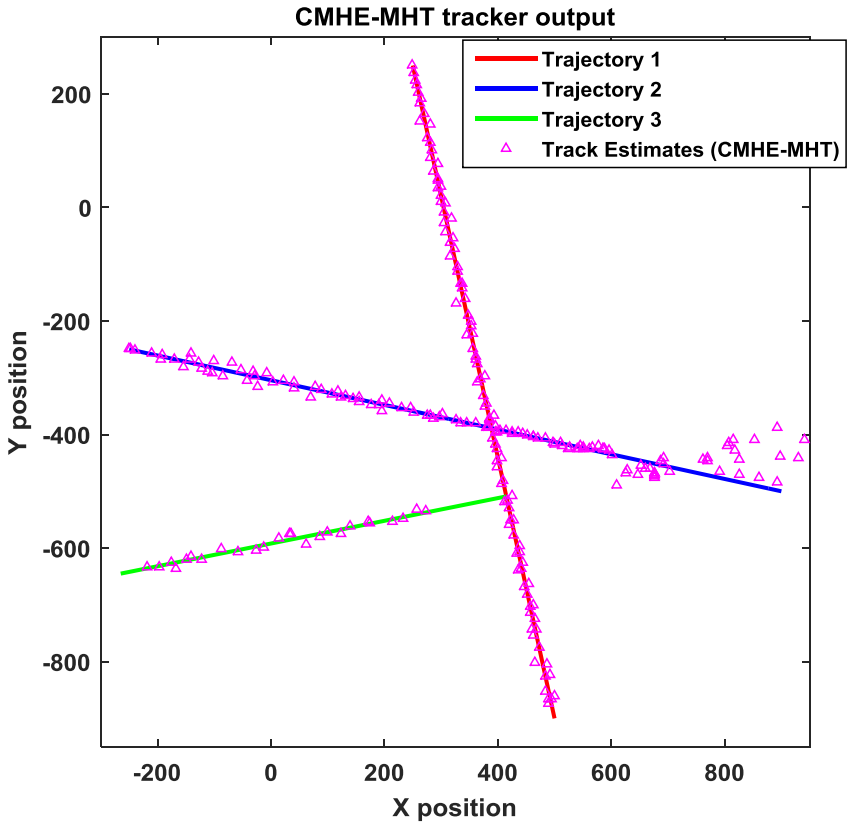


Figure 4.11 Trajectories given by the CMHE-MHT tracker for scenario 2 (m)

Figure 4.12 to 4.14 show the details of tracking result from step 40 to 60 where target 1 and 2 cross each other with tracking occlusion problem. The results show that generic MHT tracker fails to keep tracking the true tracks after occlusion and also picks up a number of false tracks. GMPHD tracker correctly identifies target 1 and 2 when they cross however the performance of position estimate is poor with lots of off road points. The CMHE-MHT presents the best tracking result which not only successfully overcomes the tracking failure when two targets cross over but also produces an accurate position estimate due to the benefit from the road information.

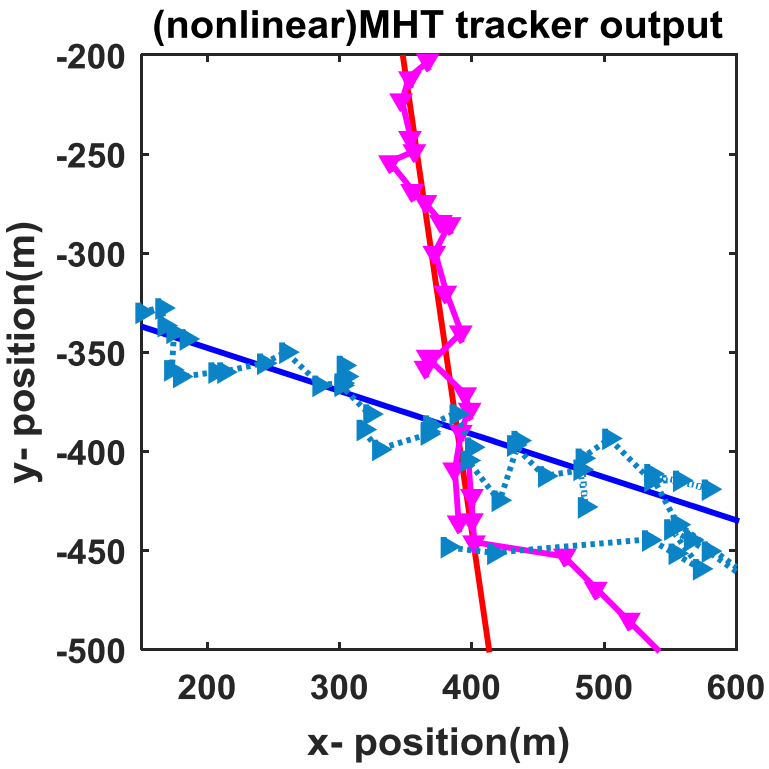


Figure 4.12 Trajectories given by the (nonlinear)MHT tracker. The results from time step 40 to step 60 is presented.

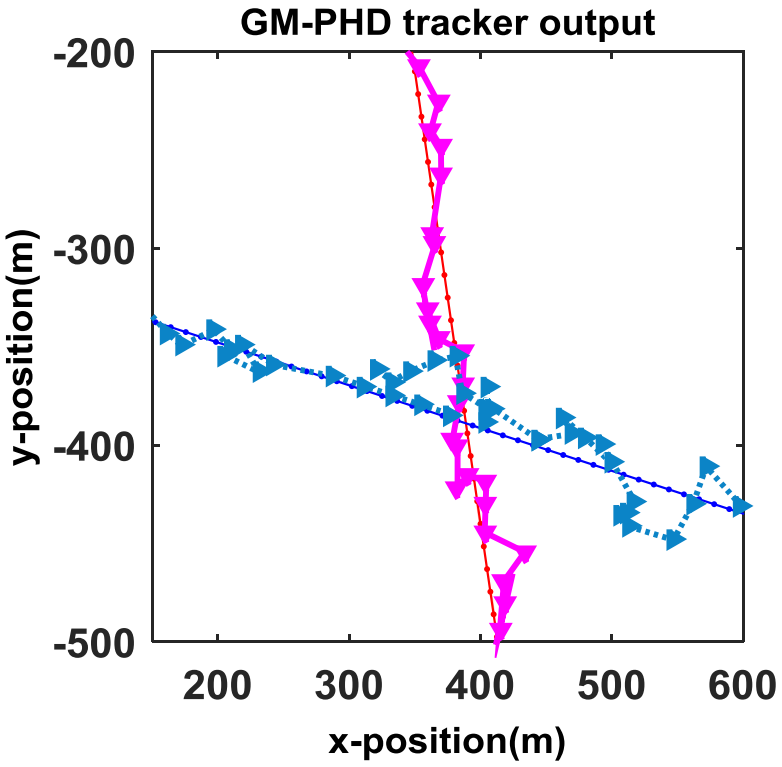


Figure 4.13 Trajectories given by the GM-PHD tracker. The results from time step 40 to step 60 is presented.

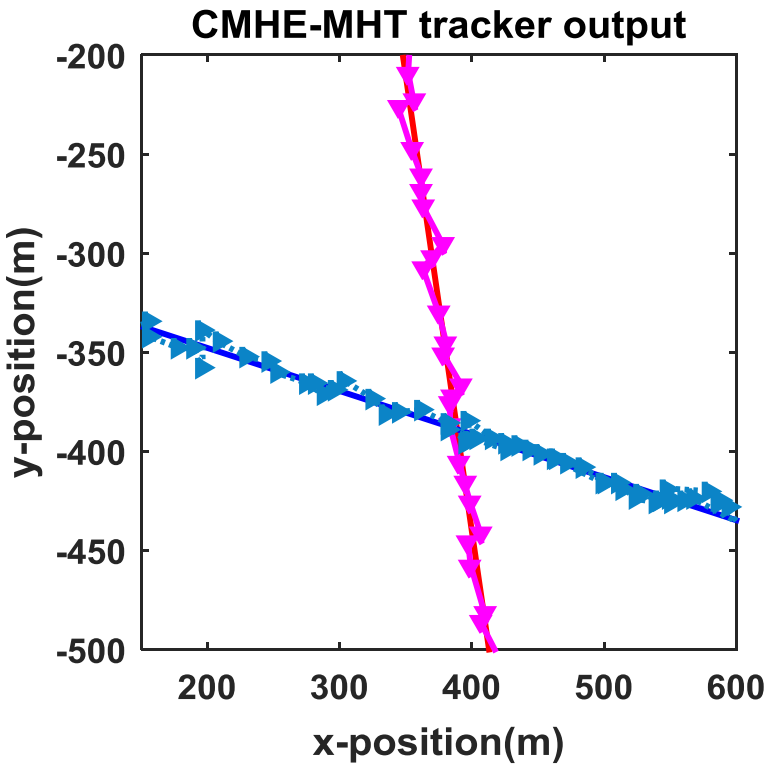


Figure 4.14 (Correct) trajectories by the CMHE-MHT tracker. The results from time step 40 to step 60 is presented.

In order to further assess the proposed multi-target tracking algorithm, two different measures of performance are used. The first one is a measure of the *cardinality* estimation, i.e., how well the algorithms estimate the number of targets. The name comes from set theory [130], where the cardinality of a set is the number of elements in the set. The cardinality measure used is the root-mean square error (RMSE) shown below:

$$e^{RMSE}(k) = \sqrt{\frac{1}{M}\sum_{l=0}^{M}(\widehat{N}_{k/k}^{l} - N_k^{\text{true}})^2} \tag{4.23}$$

where M is the number of Monte Carlo simulation, $\widehat{N}_{k/k}^{l}$ is the estimate of cardinality at time k for the l^{th} iteration, and N_k^{true} is the true number of targets at time step k. In the scenario considered in this section, the estimate of cardinality could either mean the detected (estimated) targets which includes potential tracks, or the number of actual confirmed targets/tracks. Due to the nature of MHT and the relevant benefit of

the *target maintenance* logic considered in this Chapter, the $\widehat{N}_{k/k}^{l}$ is considered as the number of actual confirmed targets.

In Figure 4.15, the cardinality error over time is shown, where the error is in relation to the true number of targets. For CMHE-MHT, the cardinality is the number of confirmed target tracks. A track is only presented if its probability of existence supersedes the threshold *Ct*.

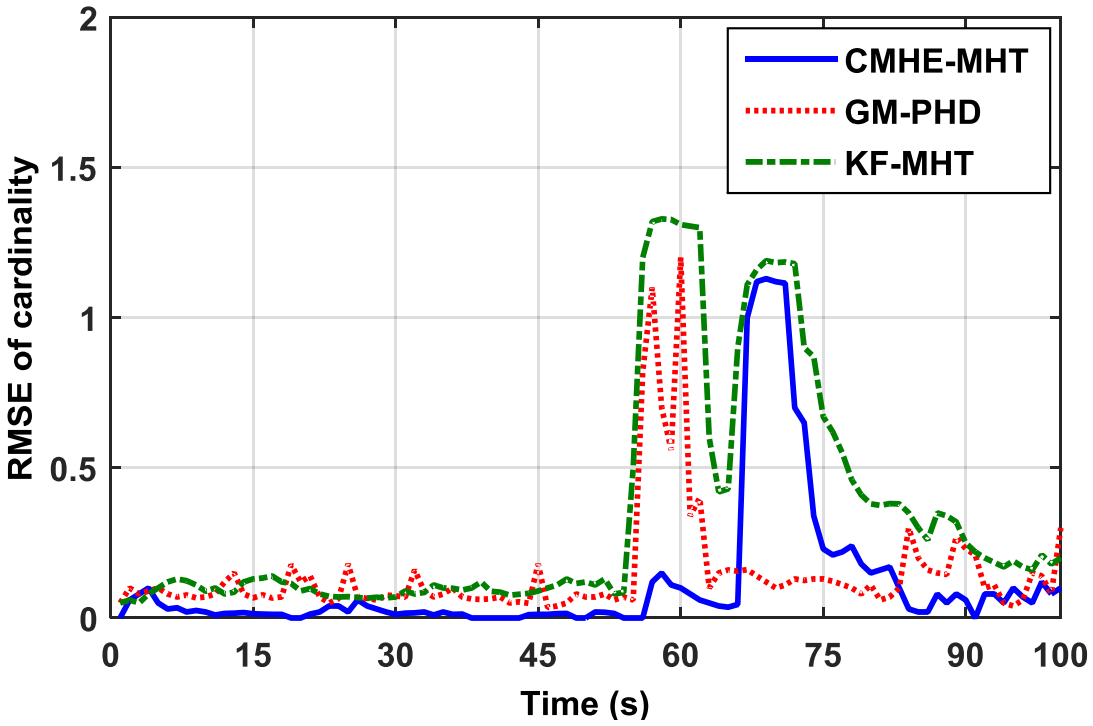


Figure 4.15 RMSE of cardinality estimation error for CMHE-MHT, GM-PHD and KF-MHT

A number of interesting conclusions can be drawn from Figure 4.15. First, the CMHE-MHT shows an overall lower number of error estimation representing a stable estimate of cardinality (once the cardinality estimate is equal to the true track number). This is due to the implementation of the road information and the extra target maintenance logic with *m*-best *N* Scan Pruning which provides a robust algorithm and helps reducing the number of false cardinality targets to be confirmed. The GM-PHD on the other hand is relatively sensitive to any possible new targets. It turns out that it over estimates the number of true new tracks in this scenario. The GM-PHD however

has relatively fast response to the changes in number of potential tracks. Most of the false detected targets are discarded very quickly.

The second thing shown from Figure 4.15 is that the CMHE-MHT is slower in adapting to new target appearance. In this scenario, a new target appears at k=66 and the RMSE of cardinality estimation error is higher than 1 for about 5 time steps which is due to the nature of target maintenance logic. The GM-PHD, on the other hand, is fast in adapting to changes in the cardinality. After the new target appearance, GM-PHD almost immediately estimates the number of target to 3 with a lower RMSE estimation error.

The third observation from Figure 4.15 is the behaviour of CMHE-MHT and GM-PHD, when two targets cross each other at time k=55, i.e., where tracking occlusion happens and the true visible number of targets decrease. The CMHE-MHT produces a stable cardinality estimation shown lower and smoother estimation error curve during the occlusion process while the GM-PHD and KF-MHT shows an increase estimation error. The reason for this is also due to the implementation of extra road information in CMHE-MHT which discards the less possible association results that are off road network. The GM-PHD and MHT on the other hand do not use the road information and thus resulting in generating a number of false tracks.

Overall the proposed CMHE-MHT is relatively slower in confirming and reducing the cardinality, but on the other hand keeps the tracking continuity over short periods of target occlusion. No method is uniformly better than the other, but for autonomous vehicle tracking applications where targets are frequently occluded by others, a slower response to a cardinality decrease has the benefit of allowing for track continuity but not removing the target track. Extra death model or new target detection and reconnection method could be introduced in MHT framework to allow for a fast response to cardinality reduction/appearance [131] as suggested in the future work.

On top of the cardinality performance, a measure is also required on how well the algorithms estimate the target states especially position error. This is measured by the position (R)MSE in previous scenarios in Chapter 3 and 4. However, since the scenario in this section is more challenging with unknown number of estimated targets, the position (R)MSE is not suitable for a straight-forward measure. Instead, the recently developed Optimal Sub-pattern Assignment metric (OSPA) [130] is used. The OSPA is proposed for evaluating the performance of MTT algorithms, which considers not only the estimation performance but also association accuracy. The OSPA metric computes the distance between two sets of tracks by adding the error between target labels (or target indices) to the spatial distance. Let Y be the set of true target states and X be the set of target estimates, with cardinalities *n* and *m*, respectively. The OSPA measure d_p^c is then defined as:

$$\bar{d}_p^c(\mathrm{X},\mathrm{Y}) = \left(\frac{1}{n} \left(\min_{\pi \in \prod n} \sum_{i=1}^{m} d^c\left(\mathrm{x}_i, \mathrm{y}_{\pi(i)}\right)^p + c^p(n-m) \right) \right)^{1/p} \tag{4.24}$$

if $m \leq n$ and $\bar{d}_p^c(\mathrm{X},\mathrm{Y}) = \bar{d}_p^c(\mathrm{Y},\mathrm{X})$ otherwise. Here $d^c(\mathrm{x},\mathrm{y}) \triangleq \min(c, d(\mathrm{x},\mathrm{y}))$ is the distance d between x and y, cut-off at c. Further, $\prod n$ is the set of all possible permutations of Y. In this thesis, the d is set as the Euclidean distance [130]. In practice, $\bar{d}_p^c$ performs an optimal assignment of target estimates to true target states which is calculated by *Hungarian* algorithm in this Chapter.

The average OSPA performance measure of KF-MHT, GM-PHD and CMHE-MHT is presented in Figure 4.16. For OSPA, p = 1 and c = 200 m. The results are based on 100 Monte Carlo simulations.

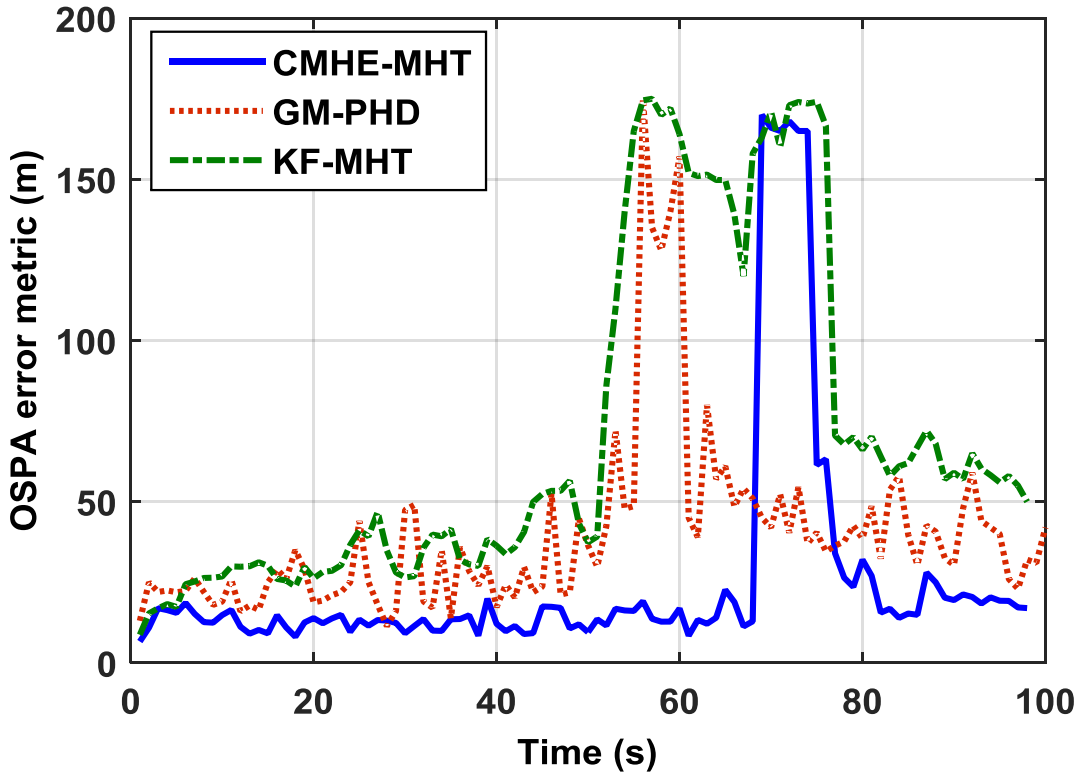


Figure 4.16. OSPA performance for CMHE-MHT, GM-PHD and KF-MHT

The overall performance of different filters including both cardinality estimation and the state estimation is shown in Figure 4.16 namely OSPA. A low value indicates good performance. Some conclusions can be drawn. The measure captures the cardinality estimation performance, where the OSPA measure increase dramatically around time step k=53 and k=66 which indicates a less accurate cardinality estimate. The figures clearly show that the CMHE-MHT aided with road information yields better performance. It is more stable than the others by observing the variation of the OSPA distance over time, which presents the smoothest OSPA results. The CMHE-MHT only shows a high OSPA value around k=66 where the cardinality estimate is very bad. Besides, the overall OSPA performance is very stable due to the extra road information and target maintenance strategy. The CMHE-MHT still shows a lower OSPA even when the cardinality estimates of different filters are very similar which indicates a better state estimate due to the road information. The GM-PHD on the other hand shows a relatively faster response to the change of the number of cardinality. Due these different natures, which behavior that is better between CMHE-MHT and GM-PHD is a matter of application requirement.

The CMHE-MHT algorithm performance is also more stable than KF-MHT which is concluded by the variation of the OSPA distance over time. This is because of the more accurate state estimation performance for constrained MHE which also affects the accuracy of new target detection and data association. In original KF-MHT, road width constraint is not considered which makes the predicted target more likely to associate with clutter and thus generate false new targets. At time k=66, the new target appears which makes OSPA increase significantly, however in MHE-MHT the faulty association hypotheses will soon be discarded by the correct one which has a higher hypothesis probability.

The OSPA performance of different filters gets worse after time step k=80 comparing with the beginning 20 steps because the ego vehicle is getting further to the targets with higher nonlinearity and lower measurement accuracy.

Due to the implementation complexity of MHT based filters, the tradeoff exists between the performance of the MHE-MHT filter and the associated computational cost and memory. Additional improvement on the performance would require more memory and increased computational cost e.g. extended horizon length for MHE and N-scan pruning. On the other hand, the MHE filter employs optimization based algorithm hence can be used for non-linear measurement models without further modification. The computational cost heavily relies on the efficiency of the optimization toolbox. In this study the *fmincon* optimization toolbox in MATLAB software is used. It is obvious that in order to tracking a large number of moving objects, a number of nonlinear MHE algorithms are involved (working in parallel) which would make the overall MHE-MHT tracking algorithm more computational expensive.

The main purpose of this chapter is not to select a better MTT algorithm between MHT and other filters. A more detailed performance comparison above between generic MHT and GM-PHD algorithm can be found in of recent reviews [131],[132]. However from this study the conclusion is drawn that the road information based constraints play an important role in improving MTT performance of the autonomous vehicle.

4.7 Summary

This chapter proposes a novel constrained MHE-MHT algorithm for MTT problems with the aid of road constraints considering multiple aspects of autonomous vehicle applications. The key idea is to use road knowledge from maps and geological information systems in different layers of the MTT structure. Besides, the contribution also comes from the combination of MHE and MHT. The external road information is employed in not only the state estimation process by the constrained MHE filter (as mentioned in Chapter 3) but also other data association process by projected state prediction. An improved merging and pruning technique and target maintenance logic are designed for the proposed algorithm to manage the hypothesis generation and track multiple targets efficiently and accurately.

A number of simulation scenarios have been set up to test the proposed algorithms to track an unknown and time-varying number of targets under detection uncertainty and false alarms with both linear and nonlinear measurement models. The performance of the proposed algorithm has also been compared with standard MHT and recently proposed GM-PHD algorithms. By using qualitative and quantitative results it was shown that the proposed framework significantly improved the tracking results in both state estimation and data association aspects.

Although the proposed MHE-MHT algorithm has proven its efficiency for autonomous vehicle tracking scenarios by accommodating the road constraint information, it does not fully take into account the domain knowledge introduced by environmental conditions. In most of the current model based MTT algorithms, the targets are considered moving independently without having interacting behaviors with other targets or physical environment. However in realistic tracking scenarios, the vehicle's maneuver is more complex and influenced by factors such as the intended directive/speed, other moving vehicles on the road, environment construction and road marks/rules. Such information could be modelled as potential force based interaction and implemented in the MTT structure. To overcome this problem and so as to achieve more accurate MTT results, the proposed MHE-MHT is further improved in Chapter 5 incorporating both the road constraint and target interaction information.

Chapter 5

Environmental Interaction Modelling and Target Tracking

5.1 Introduction

As mentioned in previous chapters, the motion of the ground vehicles is often affected by its operational environment. This information could be taken as domain knowledge and exploited in the development of tracking algorithms in order to enhance tracking quality and continuity. The most apparent domain knowledge for ground vehicle tracking is the road constraint information such as the constrained region imposed by a road map. The studies on the road network-aided ground vehicle tracking have been reported in [105], [133-136]. In these papers, the road network is taken as physical constraint information. Although comprehensive studies have been made for dealing with constraint information, limitations still exist. In Chapter 3 and 4, the MTT assisted by road map inequality constraints have been solved by using the proposed MHE-MHT framework. However for a realistic tracking scenario, in addition to physical road constraints, there are other interactions between the target and its surrounding environment which need to be considered. For instance, the driver behaviours are affected by the surrounding environment and tend to obey the traffic rules. Drivers typically try to keep away from the road boundary while following the

road/lane centre and speed limit. They also anticipate potential collision risks with incoming cars and make avoidance manoeuvres whenever necessary.

An accurate dynamic model reflecting the aforementioned realistic movement of a vehicle is vital to obtain good tracking performance, especially when limited or even no measurements are available. However, most of the current vehicle dynamic models for target tracking [30] predict the target's location from its past trajectory without fully taking into account the environmental interaction information. Recently, a social force model [137, 138] has been applied to model the interactions between pedestrians and environmental objects (building and walls) by using forces introduced by a potential field. These forces reflect different motion behaviours, for example, targets may be attracted to other objects or pushed away from them. However, the applications of the social force model are limited to pedestrian tracking in the context of surveillance rather than vehicle tracking.

With this background, a new vehicle dynamic modelling approach is proposed and its application to the MTT problem in this chapter. The proposed modelling extends the traditional methods by incorporating the environmental information into the noisy control input of a dynamic model. The interaction between the target and the environment is modelled by virtual forces constructed by the target state, target dynamics and environment information. Compared with existing social force model used for pedestrian tracking [137, 138], the proposed model is more suitable for ground vehicle tracking involving much faster manoeuvres as it utilises the entire vehicle dynamic states (e.g. position and speed) and the predicted future position rather than using the current position information only.

Among various estimation algorithms [107, 112], the optimisation-based MHE [37, 106, 117] has a promising capability of being able to accommodate different types of constraints as mention in previous chapters. Thus, in this chapter a domain knowledge-aided MHE method (denoted as DMHE) by using the aforementioned vehicle dynamic model is proposed, which incorporates both the physical environmental constraints (as mentioned in Chapter 3 and 4) and interaction information into the tracking process in a comprehensive manner. The DMHE is further combined with the improved MHT structure developed in Chapter 4, denoted as the DMHE-MHT, to deal with data association problems with miss detection and false alarm considering realistic MTT scenarios. Note that although miss detection and false alarms frequently occur in a cluttered environment, they have not been fully

considered in most domain knowledge-aided tracking works [105], [133-136]; only miss-detection is considered in [105, 134]. Different from MHE-MHT developed in Chapter 4, DMHE-MHT exploits the constraint information in both the state and measurement. The measurements are pre-projected into the constraint region to obtain more effective measurement values so as to improve the data association accuracy.

This remaining part of this chapter is structured as follows. The literature review on the associated problems in target interaction and domain knowledge aided MTT is presented in section 5.2. The domain knowledge dependent dynamic and measurement model are proposed in section 5.3. Section 5.4 explains the DMHE based target tracking algorithm, as well as its extension by combining with MHT for solving MTT problem. In order to verify the benefit and efficiency of the proposed algorithm, numerical simulation results are presented in Section V. Finally, conclusions are given in Section 5.5.

5.2 Background

In this section, different traffic models and relative target interaction behaviour in MTT are reviewed.

Various dynamic models can be generally divided into three categories: macroscopic, mesoscopic and microscopic models [139], [140]. In macroscopic models, the dynamics of the whole group of moving objects is described as an aggregate flow. Mesoscopic models determine the state of the system by the position or velocity distribution of each entity on the basis of aggregate relationships. Microscopic models refer to entities individually. In this case, the dynamics of every individual is considered by incorporating the social behaviour of each target taking into account the interaction between the target and environmental moving/stationary objects.

This chapter focuses on incorporating the environmental information in a target tracking problem using the concept of the microscopic model. Examples of microscopic models include car-following model [139], cellular automata (CLA) model [140], optimisation-based models [144] and force-based models [137]. Car-following model mentioned in [139] is particularly applied in the traffic modelling, in this model, a driver is assumed to adjust the velocity and acceleration of the vehicle according to the conditions ahead. The acceleration of a vehicle-driver unit is related

to motivational or perceived stimuli such as desired speed, speed difference and distance to the predecessor. However the application field of the car-following model is limited which is only used for modelling the traffic with respect to the following vehicles. Besides, it is assumed that a vehicle's velocity is only affected by its leader, which is not realistic considering other environmental elements, such as road boundary, will also affect its state. Optimal velocity (OV) models [141] may be interpreted as a technical variant of the car following approach, where the acceleration is determined by the difference between the velocity of the vehicle $v_i(t)$ and an optimal velocity v_{opt}.

In CLA model [140], moving objects are represented as cells and the moving areas are divided into a number of girds. A set of general logic rules are applied to state which particular cell will be occupied by a moving object, in this way, the dynamic of every individual is modelled. This model is discrete in space so that the precise estimation of the state of a moving object (position and velocity) could not be obtained. In order to obtain an accurate state estimation, a large number of cells and rules are needed, which will increase the computational complexity.

In [142-144], the optimization based technique are applied to model the air traffic. In these works, a function whose variables are the control parameters of different aircrafts is defined to be optimized. Constraints are also incorporated into the optimization problem e.g., speed limit of aircrafts and the constraint that two aircrafts could not conflict with each other. In this case the optimization problem becomes a constrained one. Stochastic optimization is applied in CLAs considering the uncertainties exist in the control model (such as the effects of wind, sensor noise, control noise, etc.). Different stochastic based optimization techniques are applied in [142-144], such as the Monte Carlo Markov Chain (MCMC) based sampling method applied in [142] and [143], and the Sequential Monte Carlo (SMC) optimization applied in [144].

Force based models are based on the assumption that the dynamic of every individual is affected by different sources of 'potential forces'. The concept of potential field is also presented in [138] for robot navigation problem, the potential field of different objects (either obstruction or attraction in the environment) are estimated and combined within a certain area. And the robot is navigated according to certain algorithms to reach the minimum point of the combined potential field.

However, in the robot navigation problem as in [138], the movement of the robot is only dependent on the potential field corresponding to the surrounding environment. No considerations are taken for the interaction between a robot and other moving objects, and it is also thought that there is no `desired velocity' for a robot in the robot navigation problem as we model the movement of vehicles/pedestrians. Comparing with other interaction models, the force-based models have a great advantage of incorporating the environmental information as different sources of forces deterministically in a continuous model as described in the following details. The force-based model could be applied for modelling vehicles in MTT in comprehensive behaviours which are not only limited to car-following scenarios or optimal velocity model. It considers comprehensive aspects for affecting the dynamics of a moving target for a more realistic modelling; Different from the CLA model which could only describe the object movement in discrete space represented as cells, force based model could describe the vehicle movement in continuous space for a more accurate state representation; It is also easy to implement and needs much less computational time compared with the optimization based methods especially considering multiple moving targets.

5.3 Environmental Information aided Dynamic, Measurement and Road Model

5.3.1 Environmental Information aided Dynamic Model:

First let us review the general dynamic model for target tracking problem:

$$x_{k+1} = f(x_k) + \omega_k \tag{5.1}$$

where x_k represents the state vector, which usually includes the position and velocity for tracking problem. ω_k is generally known as the process noise and more specifically considered as noisy acceleration components that controls the dynamic evolution of x_k and follows a certain type of distribution to represent uncertainty of a driver's behaviour. $f(\cdot)$ represents the system dynamic function which reflects a desired target dynamic type representing the state transition between consecutive time steps. According to [30], in most of the target tracking problems, the control term ω_k

is modelled as a Gaussian distribution with zero mean and constant covariance matrix representing target movement uncertainty irrespective to the surrounding environment. However, in realistic tracking scenarios, targets' movements are affected by the surrounding environment (e.g. road boundary, road centreline or speed limit). In other words, the vehicle noisy control input ω_k from uncertain driver behaviours is related with the environment.

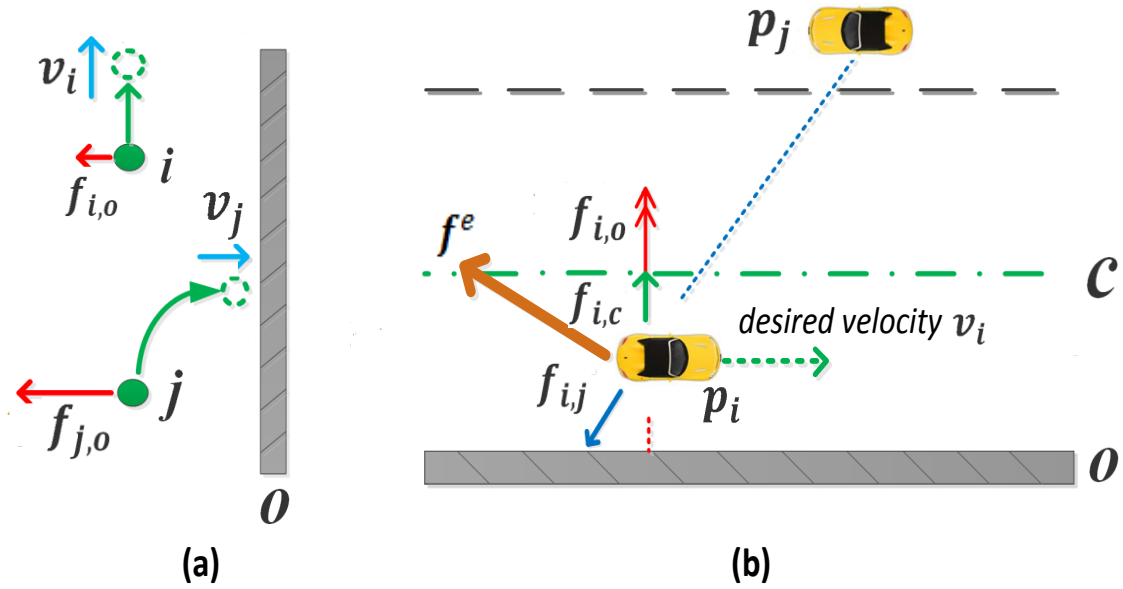


Figure 5.1 The influence of the environment on a moving target by forces: (a) different repulsive forces $\boldsymbol{f}_{i,o}$ and $\boldsymbol{f}_{j,o}$ on objects i and j with different dynamics between T=t (when objects position are marked as green circles) and T=t+Δt (where objects position are marked as dash circles) (b) p_i receives interaction force $\boldsymbol{f}_{i,j}$ from another vehicle, attractive force $\boldsymbol{f}_{i,c}$ from the centreline and repulsive force $\boldsymbol{f}_{i,o}$ from the road boundary.

Therefore, this section proposes a new vehicle dynamic modelling approach which incorporates environmental information into the vehicle control input, inspired by the social force model [137, 145]. In the original social force model, pedestrians are assumed moving with low and constant velocity in a short time interval and force is considered to be only related to the relative distance between pedestrian's current position and other environmental objects. Compared with the human tracking scenario,

our problem exhibits much more complex vehicle movements with high velocity. In this case, the force (control) term needs to consider not only position but velocity information and the desired dynamics of the vehicle. As illustrated in Figure 5.1(a), the object *j* is assumed to perform a turning manoeuvre should receive a higher repulsive force than the object *i* because it will get closer to the boundary. Besides, forces should also relate to the magnitude of the velocity; for instance, if the velocity v_j of the object *j* towards the road boundary becomes larger, a larger repulsive force should be imposed on the object.

In the proposed dynamic model, both repulsive and attractive effects from the environment are considered where the repulsive (or attractive) force is modelled as a monotonously decreasing (or increasing) exponential function. According to the current state $\mathrm{x_k}$ (including both position and velocity states) of the vehicle *i*, the predicted position $\mathrm{x}_i^{predict}$ is first calculated from the dynamic model determined by $f(\mathrm{x}_k)$. In this way, the entire state and dynamic model information are incorporated. Then, the relative Euclidean distance $d_{ij}^{prediction}$ between $\mathrm{x}_i^{predict}$ and position of the object *j* (e.g. road boundary, road centreline or other vehicles) is estimated. The repulsive/attractive force between target *i* and object *j* can then be represented as:

$$\boldsymbol{f}_{i,j}^{repulsive} = A \cdot \exp\left(\frac{-d_{ij}^{prediction}}{B}\right)\boldsymbol{n}_{ji} \tag{5.2}$$

$$\boldsymbol{f}_{i,j}^{attractive} = A \cdot \left(1 - \exp\left(\frac{-d_{ij}^{prediction}}{B}\right)\right)\boldsymbol{n}_{ij} \tag{5.3}$$

where *A* and *B* are positive constants representing the magnitude and range of the force, respectively. $\boldsymbol{n}_{ij}$ is the normalised vector pointing form *i* to *j*.

As shown in Figure 5.1(b), it is assumed that there exist different forces acting on ego vehicle *i* generated by the surrounding environmental objects, such as the repulsive force $\boldsymbol{f}_{\boldsymbol{i,o}}$ from road boundary *o*, attractive force $\boldsymbol{f}_{\boldsymbol{i,c}}$ to the centreline *c* and the repulsive force $\boldsymbol{f}_{\boldsymbol{i,j}}$ from another moving vehicle *j* to avoid a collision**,**

These forces are summed to a net environmental force $\mathrm{f^e}$ acting on the vehicle *i*, which can be incorporated into the dynamic model (5.1) shown below in (5.4):

$$\mathrm{x}_{k+1} = f(\mathrm{x}_k) + I(a_k^e) + \omega_k \tag{5.4}$$

where $a_k^e = \frac{f_k^e}{m}$ represents the acceleration introduced by the environmental force. $I(a_k^e)$ is the function representing the influence of the acceleration a_k^e on the vehicle dynamic model, which has different forms according to different dynamic function $f(\cdot)$ as shown in (5.1).

5.3.2 Environmental Information aided Measurement Model:

In order to incorporate the road constraint information, in Chapter 4 the road inequality constraints is accommodated in the predicted state. Besides, the state vector, the environmental information is also considered in the measurement vectors.

For the model based tracking problem, usually measurements are associated with a measurement model which can be generally represented as:

$$\mathrm{z}_k = h(\mathrm{x}_k) + v_k \tag{5.5}$$

where z_k is a measurement vector, $h(\mathrm{x}_k)$ is the measurement function and v_k is zero mean Gaussian noise of the measurement with the covariance R.

It is assumed that the ground vehicles only move within the road network region. This matches with a realistic scenario where road boundaries are considered as physical constrains and all drivers are supposed to move within the constraint region. In addition to the physical constraints, the behaviour of a vehicle is affected by the environment following the Highway Code and traffic rules. As a result, when moving on the road, the vehicle not only keeps away from the road border but also tends to follow the centre line of the road according to the heading direction.

Due to the limited tracking sensor's capability, the received measurements usually contain noises as in (5.5), which make them not always stay on the road network and far away from the ground truth values. Such noisy measurements are usually known as false alarms in MTT which make data association process really difficult with lots of tracking ambiguity problems. Especially in MHT based approaches, where a number of candidate hypotheses are generated at each step. The 'hard decision' is only made when more measurements are received. In this case, the prior decisions could be evaluated and corrected so as to achieve a higher probability for correct data association. Such off road noisy measurements could result in lots of redundant hypotheses with low probabilities which are most likely discussed later.

To this end, a pre-processing approach is used in this paper to project the raw measurements onto the constrained surface (road network) at each time step so as to decrease the uncertainty from the false alarms. The pre-processing approach could also bring down the number of unnecessary MHT hypotheses and as a result substantial amount of computation time could be reduced for real time tracking applications.

Assuming that that target vehicles are traveling on linear road following the centre line, the raw measurement data z_k could then be projected by the following linear equality constraint (5.6) to reflect more effective measurements:

$$C\tilde{\mathrm{z}}_k = d_k \tag{5.6}$$

where C is a full-rank constraint matrix and d_k is the constraint vector. $\tilde{\mathrm{z}}_k$ is the projected (constrained) Cartesian measurement. Following [106], the expression of deriving constrained measurement $\tilde{\mathrm{z}}_k$ by directly projecting the unconstrained Cartesian coordinate measurement z_k onto the constraint surface is by solving the problem:

$$min_{\tilde{\mathrm{z}}_k}(\tilde{\mathrm{z}}_k - \mathrm{z}_k)\, W(\tilde{\mathrm{z}}_k - \mathrm{z}_k) \qquad s.t \qquad C\tilde{\mathrm{z}}_k = d_k \tag{5.7}$$

where W is a symmetric positive definite weighting matrix. In this work, it is chosen as $W = R$ following the *mean square method*, where R is a measurement error covariance matrix of the original measurements. The solution of this problem is then given by:

$$\tilde{\mathrm{z}}_k = \mathrm{z}_k - R^{-1}C^{T}(CR^{-1}C^{T})^{-1}(C\mathrm{z}_k - d_k) \tag{5.8}$$

According to [106], the projected measurement error covariance matrix $\tilde{R}$ can be expressed as:

$$\tilde{R} = (I - R^{-1}C^{T}(CR^{-1}C^{T})^{-1}C)R(I - R^{-1}C^{T}(CR^{-1}C^{T})^{-1}C)^{T} \tag{5.9}$$

In this way, the measurement model is modified as:

$$\tilde{\mathrm{z}}_k = \begin{bmatrix} x_k \\ y_k \end{bmatrix} + \tilde{v}_k \tag{5.10}$$

where $\tilde{z}_k$ is the projected measurement, $\begin{bmatrix} x_k \\ y_k \end{bmatrix}$ represents the target position and $\tilde{v}_k$ is the measurement noise for the projected measurement with zeros mean and covariance $\tilde{R}$.

For nonlinear road constraint $g(\tilde{z}_k) = d_k$, the development of the constrained measurement (5.10) as given above is still valid with a linearisation process. A first order Taylor series expansion of the constraint equation around $\tilde{z}_k$to is used to obtain:

$$g(\tilde{z}_k) = d_k \approx g(\tilde{z}_k) + g'(\tilde{z}_k)(z_k - \tilde{z}_k) \tag{5.11}$$

which indicates that:

$$g'(\tilde{z}_k)\tilde{z}_k \approx d_k - g(\tilde{z}_k) + g'(\tilde{z}_k)\tilde{z}_k \tag{5.12}$$

An approximated nonlinear constraint is now formed that is equivalent to the linear constraint $C\tilde{z}_k = d_k$ where C is replaced with $g'(\tilde{y}_k)$ and d_k is replaced with $d_k - g(\tilde{z}_k) + g'(\tilde{z}_k)\tilde{z}_k$.

5.3.3 State Dependent Road Model Transition

In a realistic tracking scenario, in addition to multiple target data association and state estimation problems, each vehicle may undergo different road segments with different environmental conditions. Thus, one single model might not be able to accurately describe various movement types. E.g. the vehicle may have different dynamic model such as CV, CT and CA model when moving on different road segments incorporated with different environmental constraints.

Multiple state models have been exploited for target tracking in [136], [127] namely the interacting multiple model methods (IMM). These approaches assume a Markov jump model with constant state transition probabilities. However, the manoeuvring type of a certain target is actually state dependent and environmentally related. E.g. a vehicle may slow down and then turn when it approaches to a road junction. To its end, the non-Markov jump model approaches [146] is required. The transition probabilities between different models are not constant but modelled in a state dependent way related to surrounding environmental conditions.

Considering that multiple state models are involved, before performing the DMHE-MHT algorithm, it is required to associate the vehicle to different road

segments so that the corresponding information, such as road width, speed limits, can be taken in to account as system constraints. In this chapter, it is assumed that each vehicle is moving on the road structure restricted by road boundaries. The road structure is constructed by different segments and associated with different constraints which are known from a prior digital road map. A vehicle may jump from one to another road segment depending on the vehicle state (positions, velocities) and the road map information. The chosen road segment is then applied for DMHE-MHT implementation to track each specific target.

5.4 MHE based Target Tracking with Environmental Information

Based on the domain knowledge aided dynamic modelling and measurements as mentioned in the previous section, the MHE based optimization scheme is applied for the state estimation, which is detailed as follows:

5.4.1 Domain Knowledge aided MHE (DMHE)

Although the aforementioned MHE method could incorporate the constraint information for the state estimation as discussed in Chapter 3, it cannot exploit the environmental information in a comprehensive way:

i. The interaction between the target and surrounding environment (e.g. a vehicle keeps away from stationary/moving environmental objects, such as road boundary, another vehicle, etc.) is not considered in the original MHE framework
ii. Domain knowledge is not considered in the measurement model

To this end, a new framework of the MHE which fully exploits the domain knowledge (denoted as DMHE for short) is proposed. Both the proposed state model (5.4), which considers the interaction information and the projected measurement model (5.10) are exploited to construct a new MHE optimization function for estimating the state, which is illustrated as:

$$\min_{\{x_{k-N},\{\omega_j\}_{j=k-N}^{k-1}\}} \sum_{j=k-N+1}^{k} \left(\left\| x_j - f(x_{j-1}) - I\left(a_j^e\right) \right\|_{Q^{-1}}^2 + \left\| \tilde{z}_j - h(x_j) \right\|_{\tilde{R}^{-1}}^2 \right)$$

$$+\Gamma_{k-N}(x_{k-N}), \qquad \text{for } \{x_{k-N}, .., x_k\} \in C_x. \qquad (5.13)$$

Compared with the constrained MHE function in (3.14), besides the road constraint based information the domain knowledge is better exploited from two folds:

i. a new $I\left(a_j^e\right)$ term is introduced, which is related to the environmental force modelling the interaction as mentioned previously. In this way, the interaction information is considered in the new MHE process.
ii. the projected measurements $\tilde{z}_j$ and associated error covariance $\tilde{R}$ are exploited to model measurement information in a more accurate way

By solving the DMHE cost function (5.13), the optimised estimated state at time *k* considering a linear dynamic system with estimated initial state $\hat{x}_{k-N}^*$ and the optimised process noise sequence $\left\{\widehat{\omega}_j^*\right\}_{j=k-N}^{k-1}$ in the horizon length *N* as:

$$\hat{x}_k^* := x\left(k; \hat{x}_{k-N}^*, \left\{\widehat{\omega}_j^*\right\}_{j=k-N}^{k-1}, \left\{f_j^{e,*}\right\}_{j=k-N}^{k-1}\right) =$$

$$F^k \hat{x}_{k-N}^* + \sum_{j=k-N}^{k-1} F^{k-j-1} \left(\widehat{\omega}_j^* + I\left(a_j^{e,*}\right)\right) \qquad (5.14)$$

Note that, $a_j^{e,*}$ is a function of $\hat{x}_{k-N}^*$ and $\left\{\widehat{\omega}_j^*\right\}_{j=k-N}^{k-1}$, according to the force terms defined as in (2) and (3). The covariance required for the arrival cost computation as in equation (3.16) is modified by considering the influence of the term $I(a_k^e)$ at time instance *k* by:

$$P_{k+1} = Q + \left(F + \nabla_{\hat{x}_k^*} I(a_k^e)\right) P_k \left(F + \nabla_{\hat{x}_k^*} I(a_k^e)\right)^T$$

$$-(F + \nabla_{\hat{x}_k^*} I(a_k^e)) P_k H^T (R + H P_k H^T)^{-1} H P_k (F + \nabla_{\hat{x}_k^*} I(a_k^e))^T \qquad (5.15)$$

where $\nabla_{\hat{x}_k^*} I(a_k^e)$ represents the gradient of the term $I(a_k^e)$ with respect to $\hat{x}_k^*$ at time *k*.

5.4.2 DMHE based MHT (DMHE-MHT)

The proposed DMHE algorithm is further extended to address the data association problem by incorporating it into a multiple hypothesis tracking (MHT) structure, which constructs a DMHE-MHT framework for MTT in a more complicated scenario with both miss detections and false alarms. The detail of the improved MHT structure has been explained in Chapter 4. The flow diagram of the DMHE-MHT algorithm is presented in Figure 5.2.

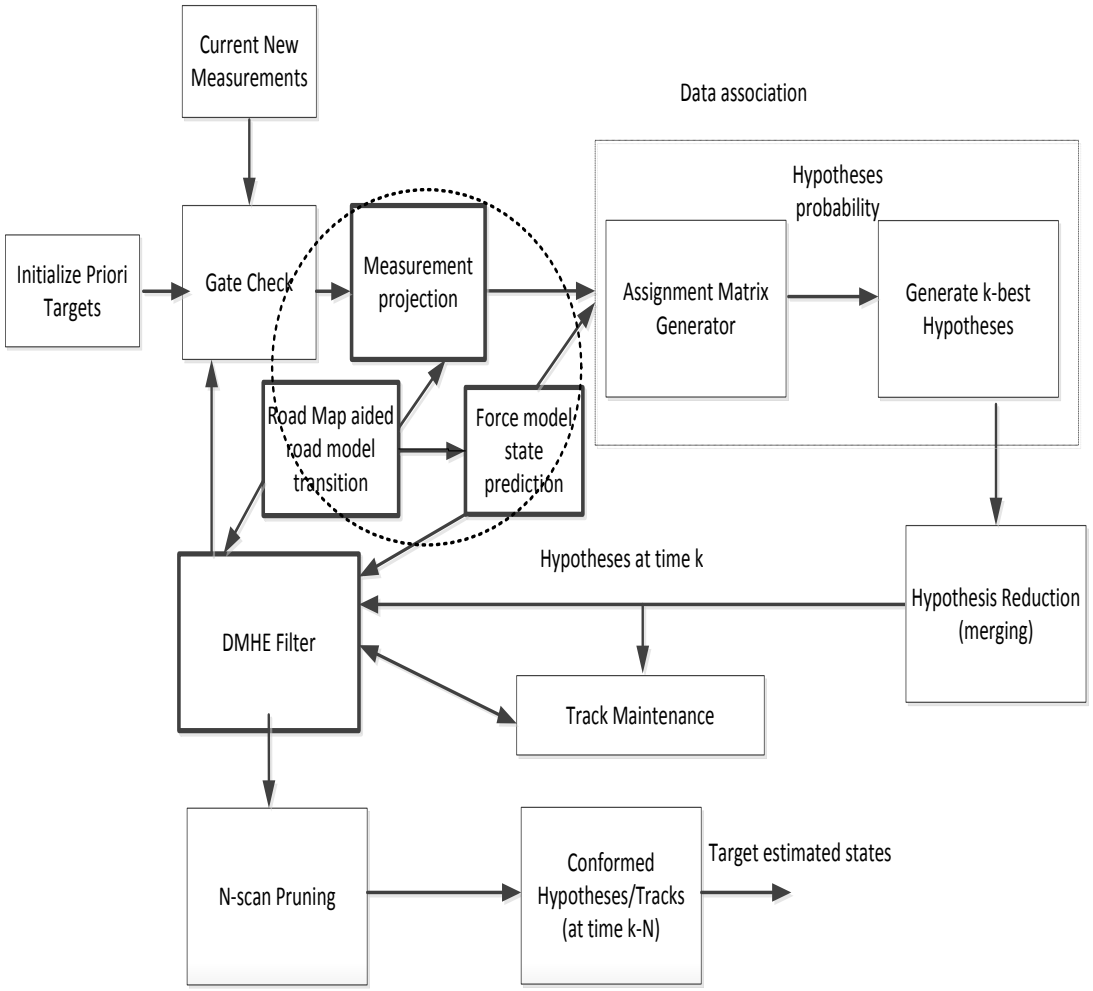


Figure 5.2 Flow diagram of DMHE-MHT algorithm.

Comparing with MHE-MHT, the structure is more completed by incorporating the domain knowledge from both road physical constraint in the state and measurement as

well as interaction behaviour between targets and the environment. This is achieved by three blocks as shown in Figure 5.2: measurement projection, road map aided road model transition and force model state prediction. First the road map aided road model transition process considering a state-dependent process to determine which road the target is moving on with which specific constrains as discussed above. The candidate measurements are then projected to the road using the process shown in section 5.3.2 considering road constraints. The projected measurement $\tilde{\mathrm{z}}$ and related covariance $\tilde{R}$ are calculated and utilised in the data association process later on.

Once state dependent road model transition process is solved, the corresponding target environment interaction force such as (5.2) and (5.3) can be determined and calculated considering the specific road segment. The interaction force based state dynamic prediction is then calculated which is used in both data association and MHE process when missed measurement issue happens. Different from the constrained state prediction process (4.12) mentioned in Chapter 4, where the predicted state $\hat{\mathrm{x}}_{k|k-1}$ is only constrained by the physical road boundary, in this chapter the interaction information is also accommodated in the state prediction shown below in (5.16):

$$\hat{\mathrm{x}}_{k|k-1} = \tilde{\mathrm{x}}_{k|k-1} + I(a_k^e) \tag{5.16}$$

where $\tilde{\mathrm{x}}_{k|k-1}$ is the constrained prediction calculated by equation (4.12). $a_k^e = \frac{f_k^e}{m}$ as discussed above represents the acceleration introduced by the environmental force. $I(a_k^e)$ is the function representing the interaction influence of the acceleration a_k^e on the vehicle according to different dynamic model $f(\cdot)$.

In this thesis, the missed measurement issue is presumed as a one-step state prediction under the MHE framework. In this case, both road constraint and environmental interaction information are vital to reflect the realistic movement. In addition to the state estimation process, the force model state prediction is also utilised in *gate check* and *data association* process in the DMHE-MHT with the same form as mention in Chapter 4. In order to avoid wasting the computational time in generating a lot of low certainty MHT hypotheses (generated by noisy false alarms) which will be most likely to be discarded from the hypotheses tree later, the projected measurement $\tilde{\mathrm{z}}$, constrained measurement error covariance $\tilde{R}$ and forced based state prediction are used for calculating a more convergent data association result.

5.5 Numerical Simulation Results

In this section, two simulation examples are presented in the context of ground vehicle tracking. The first example is single target tracking, aiming at illustrating the proposed DMHE with both linear and nonlinear inequality road constraint. The second one is a complex multiple vehicle tracking scenario incorporating road inequality constraints from real world map data for the DMHE-MHT. The details of implementation with fragments of the code is provided in the Appendix C.

5.5.1 Single Target Tracking

The proposed DMHE algorithm is evaluated by single target tracking scenario for both linear (position) and nonlinear (bearing/range) measurement models with road boundary constraints. The first one is a linear trajectory, considering a single carriageway with road width of 4 meters and an angle of 45 degrees anticlockwise to the horizontal axis. The vehicle dynamics is described by a constant velocity model with the noisy acceleration:

$$\mathrm{x}_{k+1} = \begin{bmatrix} 1 & T & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & T \\ 0 & 0 & 0 & 1 \end{bmatrix} \mathrm{x}_k + \begin{bmatrix} T^2/2 & 0 \\ T & 0 \\ 0 & T^2/2 \\ 0 & T \end{bmatrix} \omega_k \tag{5.16}$$

where the state vector $\mathrm{x}_k = [x_k, \dot{x}_k, y_k, \dot{y}_k]^T$ consists of the vehicle position and velocity in x and y directions, and $T = 1$ is the sampling interval, ω_k is a two-dimensional Gaussian process noise with zero mean and covariance matrix Q = $diag\{5,2\}$ in a local coordinate as discussed in Chapter 4 where $diag\{.\}$ represents a diagonal matrix. This covariance represents higher motion uncertainty along the centre line direction and smaller uncertainty orthogonal to the road. The vehicle measurement model is a linear matrix in x and y potion with a Gaussian measurement noise v_k and covariance matrix R = $diag\{20/\sqrt{2},\ 20/\sqrt{2}\}$ in a global Cartesian coordinate as:

$$\mathrm{z}_k = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \mathrm{x}_k + v_k. \tag{5.17}$$

The vehicle has a centre line direction velocity of 10m/s with no initial lateral velocity and the initial state is $x_0 = [0, 7.0711, 1502.83, -7.0711]^T$.

The movement of the target is constrained by road boundaries and supposed to follow the centre line of the road. Different environmental forces are considered including lateral forces orthogonal to the road as:

- Repulsive force generated from lower road boundary

$$\boldsymbol{f}_{i,j}^{rep1} = A \cdot \exp\left(\frac{-\boldsymbol{d_{lb}}}{B}\right)\boldsymbol{n}_{ji} \tag{5.18}$$

- Repulsive force generated from upper road boundary

$$\boldsymbol{f}_{i,j}^{rep2} = A \cdot \exp\left(\frac{-\boldsymbol{d_{ub}}}{B}\right)\boldsymbol{n}_{ji} \tag{5.19}$$

- Attractive force to centre line of the road

$$\boldsymbol{f}_{i,j}^{att} = A \cdot \left(1 - \exp\left(\frac{-\boldsymbol{d_{center}}}{B}\right)\right)\boldsymbol{n}_{ij} \tag{5.20}$$

where *i* and *j* represents the target and the environment (road boundary, centre line, and speed limit where applicable), respectively. $\boldsymbol{d_{lb}}$ and $\boldsymbol{d_{ub}}$ represent the Euclidean distance between lower and upper boundary of the road and the predicted vehicle position $x_i^{predict}$ calculated from the dynamic model (11) base on the current location, respectively. Similarly, $\boldsymbol{d_{center}}$ represents the distance between centre line and predicted vehicle position. Note that the closer (further) the vehicle gets to the road boundaries (away from the centre line), the larger the repulsive (attractive) force will be generated.

Beside above lateral forces, a velocity-based breaking (repulsive) force is also considered along the centre line direction so as to present the road speed limit:

$$\boldsymbol{f}_{i,j}^{rep3} = -A \cdot \exp\left(\frac{-(\boldsymbol{v_{limit}} - v_{heading})}{B}\right)\boldsymbol{v} \tag{5.21}$$

where $v_{heading} = \sqrt{\dot{x}_k^2 + \dot{y}_k^2}$ is the speed of the vehicle towards heading direction. And similar to $\boldsymbol{n}_{ij}$, $\boldsymbol{v}$ represents a unit velocity vector. The speed limit $\boldsymbol{v_{limit}}$ is defined as a specific speed value in the heading direction which is different for each road section. The breaking force has the same manner as the repulsive force

(5.18) and (5.19) and the vehicle always tends to follow the speed limit. When the vehicle's heading velocity is within the speed limit ($v_{heading}<v_{limit}$) only a small repulsive force will be affected. However the repulsive effect grows exponentially when the vehicle exceeds the speed limit, as illustrated in Figure 5.3:

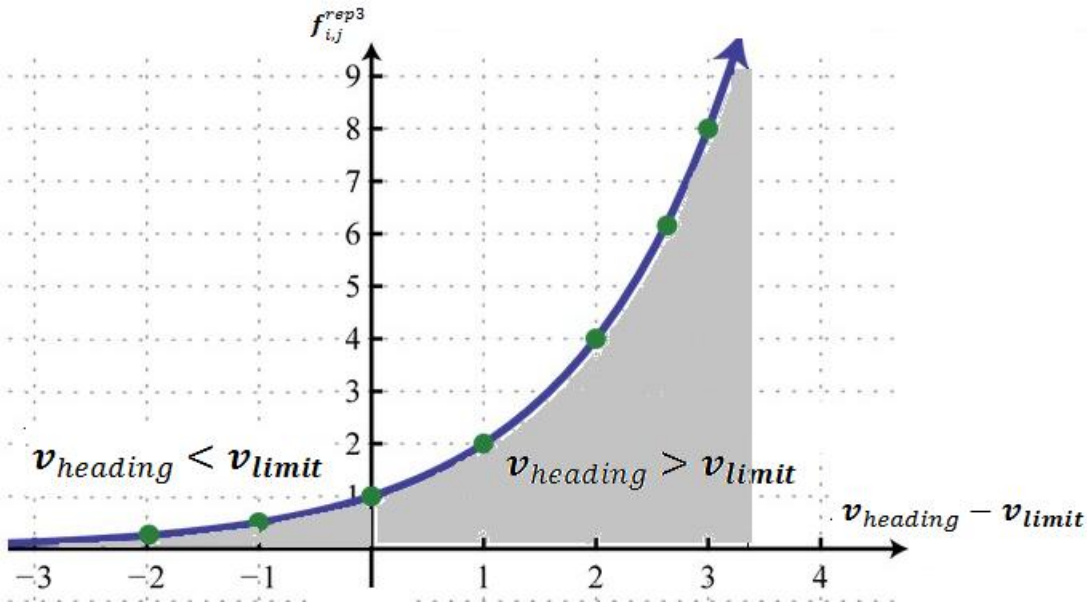


Figure 5.3 Force generated from the speed limit.

To evaluate the performance, four different tracking models are compared: i) general MHE without considering any environmental information (MHE), ii) force based MHE without considering physical constraints (FMHE), iii) general MHE with inequality physical constraints (road boundaries) (CMHE), and iv) the proposed DMHE approach. In Table 5.1, the performance of different models is compared in terms of RMSE in three different aspects: i) position RMSE, ii) centre line direction position RMSE and iii) orthogonal position RMSE to the road with a horizon length of N=4. It is shown that road physical constraint is of great importance when comparing the CMHE with the MHE and the DMHE with the DMHE, especially in orthogonal direction where road boundary is considered. In addition to physical constraints, environmental forces further improve the estimation accuracy. Both the FMHE and the DMHE have shown a significant improvement for target's position estimate compared with their relative MHE and CMHE.

Table 5.1 Estimation performance comparison of MHE, FMHE, CMHE, and DMHE

RMSE(m)	MHE	FMHE	CMHE	DMHE
Position(m)	2.6506	2.4206	2.4877	2.3216
Centre line direction(m)	3.1298	2.9606	3.1181	2.9414
Orthogonal position(m)	2.8932	2.5160	1.6295	1.4588

In the second scenario, a vehicle is simulated to move along the quarter of a circular road with an angular velocity of 0.1 rad/s along the road centreline for 15 seconds. Small noises are added to the simulated vehicle position to represent the disturbance of the vehicle movement. The road has a width of 4 meters and is defined by two arc boundaries of r_1=96 m and r_2=100 m, respectively, centred at the origin of a Cartesian coordinate system, as shown in Figure 5.4. The speed limit of this road segment for the vehicle to keep is assumed to be 30 miles/hour (13.4m/s).

Regarding the range and bearing measurement model in (5.22), it is assumed that a radar sensor is positioned at the origin. The corresponding measurement noise v_k follows a Gaussian distribution with zero mean and covariance $R = diag\{36, 10^{-2}\}$.

$$\mathrm{z}_k = \begin{bmatrix} r_k \\ \theta_k \end{bmatrix} = \begin{bmatrix} \sqrt{{x_k}^2 + {y_k}^2} \\ arctan\left(\frac{y_k}{x_k}\right) \end{bmatrix} + v_k \tag{5.22}$$

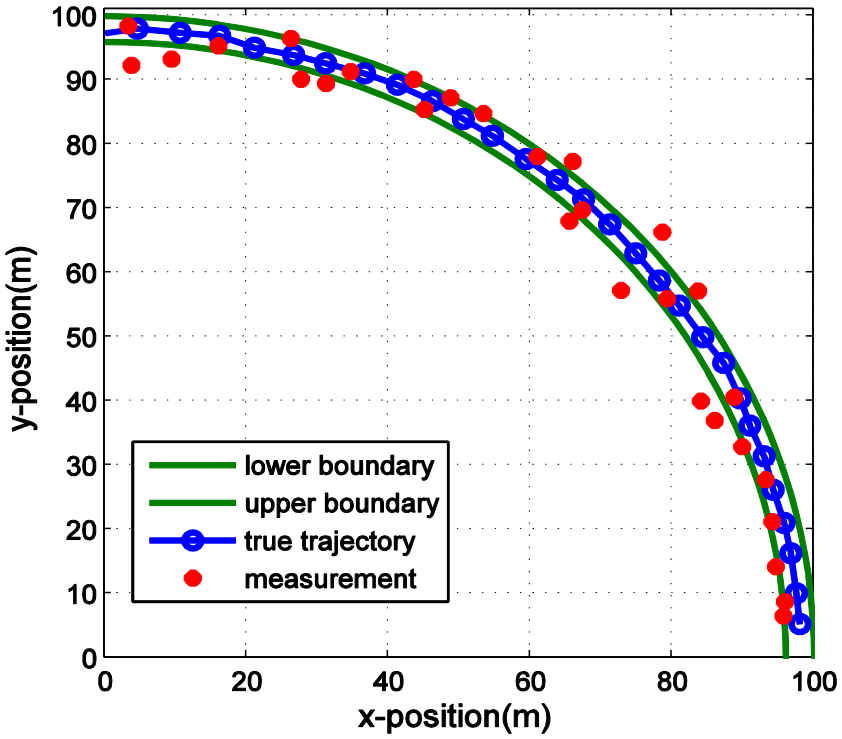


Figure 5.4 The simulated circular road tracking scenario.

Three algorithms are chosen for comparison for this simulated scenario including the EKF, the constrained MHE (CMHE) which considers the road boundary constraint and the proposed DMHE. The system dynamic model for tracking is the same as the previous scenario. The reason a constant velocity model is still used here is to emphasise the benefit of using domain knowledge in the target tracking even with a poor dynamic model. Although better nonlinear models (e.g. a constant turning model) could be used, by using a relatively less accurate dynamic model, the benefit of the additional force-based interaction information could be emphasised especially when comparing the DMHE with the CMHE. For the EKF and the CMHE, the system dynamic model for tracking is the same as the previous scenario. For the proposed DMHE method, additional interactions between the target and environment are considered by using two forces: i) road repulsive forces generated by the road upper and lower boundary and ii) force acting in the opposite of movement tangential direction to prevent the vehicle from exceeding the speed limit. For a fair comparison,

all the algorithms are set to have the same initial condition with mean $x_0 = [75,10,0,10]^T$ and covariance $P_0 = diag\{10,1,10,1\}$.

Firstly, a sample tracking performance of three different algorithms is illustrated in Figure. 5.5. It can be observed that the estimation result of the EKF is outside the road boundary. The performance is improved in the CMHE with the tracking results being projected on the road boundary. However, it is still quite different from the true trajectory. The most accurate and reasonable tracking result is obtained by the DMHE. Next, numerical evaluations are performed on three algorithms using the root mean square errors (RMSEs) through a hundred Monte Carlo simulations for the same scenario. Figure 5.6 presents the averaged RMSE time history of the estimated position of each filter (the sampling interval is 0.5s). It can be seen that the DMHE approach achieves the minimum RMSEs during the majority of times. Besides, the averaged RMSEs for the whole target trajectory by different methods are presented in Table 5.2. Again, the DMHE achieves the most accurate tracking performance. In comparison to the EKF and the CMHE, the averaged RMSE for position estimation by the DMHE is improved by 66.8% and 27.7%, respectively.

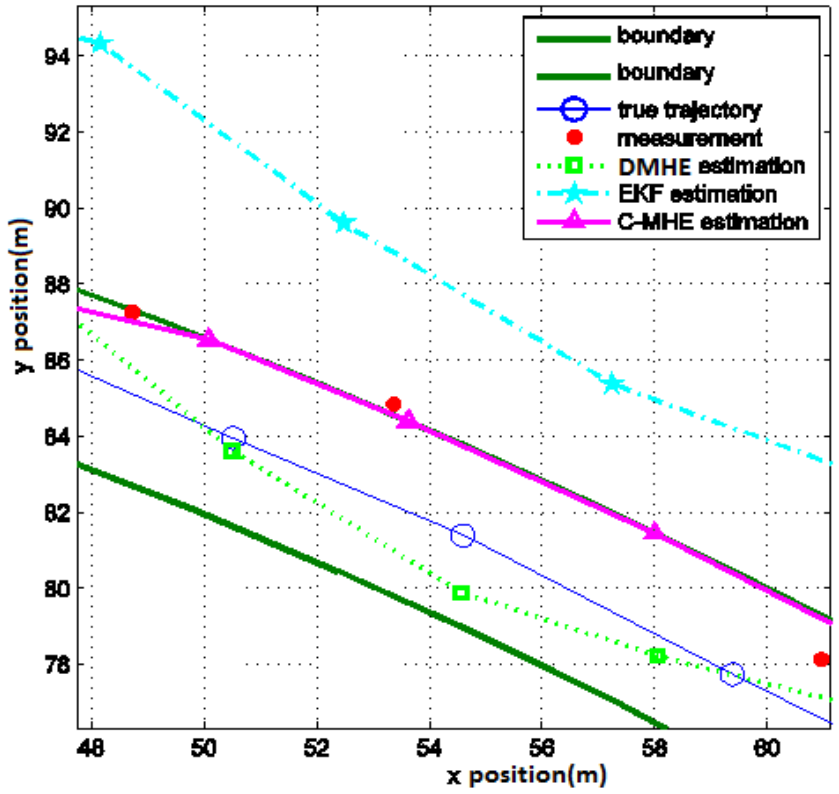


Figure 5.5 True and estimated results for EKF, CMHE and DMHE.

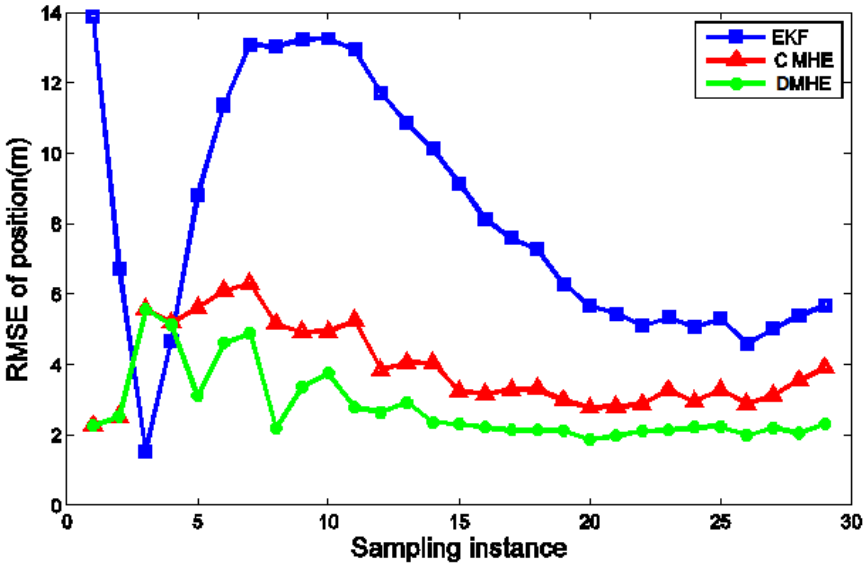


Figure 5.6 RMSE of estimated position of EKF, CMHE and DMHE.

Table 5.2 Averaged RMSEs for EKF, CMHE and DMHE.

	EKF	CMHE	DMHE
RMSE(m)	8.8261	4.0494	2.9281

5.5.2 Multiple Target Tracking

Simulation scenario: The performance of the DMHE-MHT is compared against the MHT, and the CMHE-MHT for multiple target tracking. A more realistic environment is considered. Three vehicles are simulated to move in a realistic region (near Loughborough town in the UK, and the region's geographic information is obtained from the GIS). As shown in Figure 5.7, a road intersection scenario is considered with a rectangular region of surveillance, with an unknown and time varying number of targets observed in a clutter environment. The vehicle dynamics is described the same as (21). The two-dimensional Gaussian process noise has covariance matrix Q of 25 m/s^2. Initially, two targets start moving in the environment: vehicle 1 (shown as the red point) heads to the southwest direction with an initial speed along road one of 12 m/s, it then crosses the intersection and travel on road 3; vehicle 2 (shown as the black point) starts from road 4 heading to the northwest direction with an initial speed along the road network of 8 m/s, it then crosses the intersection and travel on road 2. A new vehicle 3 starts to move three seconds later from road 2 with initial speed of 8 m/s heading to southeast direction and then change its direction at the intersection heading to northeast on road 1. As shown in Figure 5.7, tracking ambiguity occurs during the process around the intersection and on road 1 and 2, which makes the problem challenging.

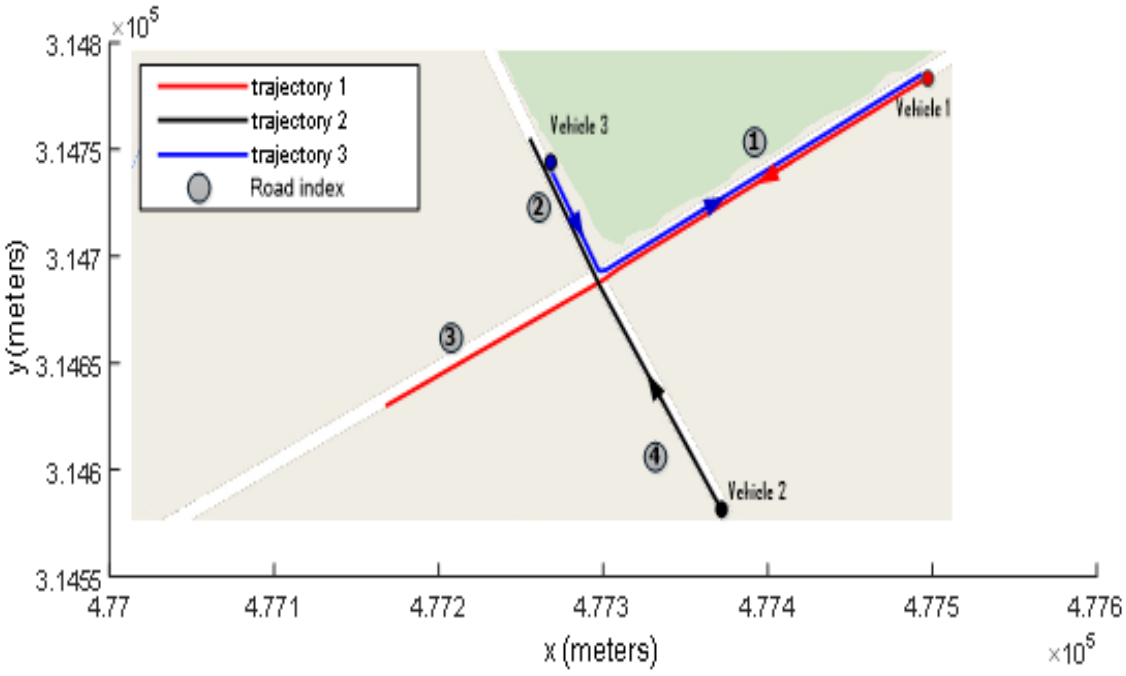


Figure 5.7 Multiple target tracking scenario.

The target initial covariance is defined as $P_0 = diag\{100,25,100,25\}$ for all three targets. Each target is detected with a probability of $P_d = 0.98$. Regarding the range and bearing measurement model in (28), it is assumed that a radar sensor positioned at the bottom right corner. The corresponding measurement noise v_k follows a Gaussian distribution with zero mean and covariance $R = diag\{25, 2.5^{-3}\}$. The detected measurements are immersed in a high clutter environment that can be modelled as a Poisson distribution with clutter density of $\beta_{FA} = 7.3 * 10^{-5}$ (false alarms/area/scan) over the $1.375 * 10^5 m^2$ region (i.e., clutter returns over the region of interest).

Domain knowledge exploitation: The speed limits of the main road (road one and road three along the east-west direction) and side road (road two and road four along the north-south direction) are 40 miles/hour (17.9m/s) and 30 miles/hour, respectively. And the road constraints are applied to constrain the vehicle positions and measurements.

In addition to physical constraints, different target interactions with the environment are considered including interaction between: i) the vehicle and road boundary, ii) the vehicle speed and speed limit, and iii) vehicle in the minor road (2

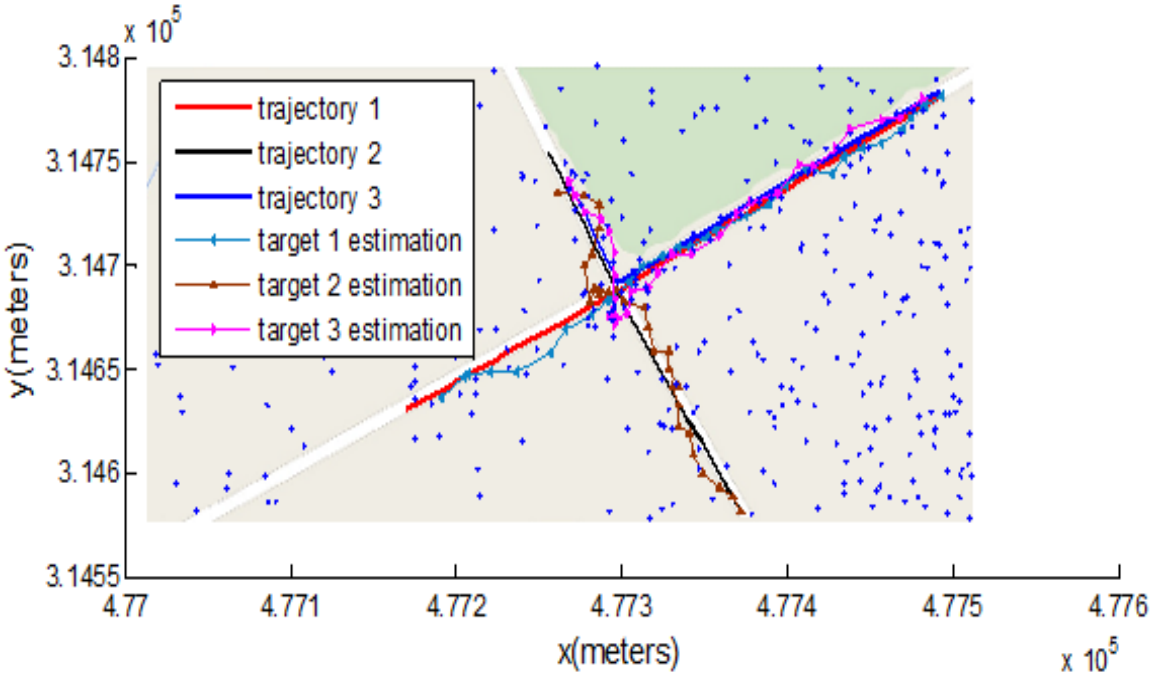
and 4) and the junction (the vehicle in the minor road will slow down when it approaches the junction). Besides, the interactions between moving vehicles are also considered. These interactions are represented by forces, which is defined below:

$$\boldsymbol{f}_{i,j}^{rep4} = \begin{cases} A \cdot \exp\left(\frac{-\boldsymbol{d}_{ij}}{B}\right)\boldsymbol{n}_{ji}, & \text{if } \boldsymbol{d}_{ij} \leq D_t \\ \boldsymbol{f}_{i,j}^{rep4} = 0, & \text{otherwise} \end{cases} \tag{5.23}$$

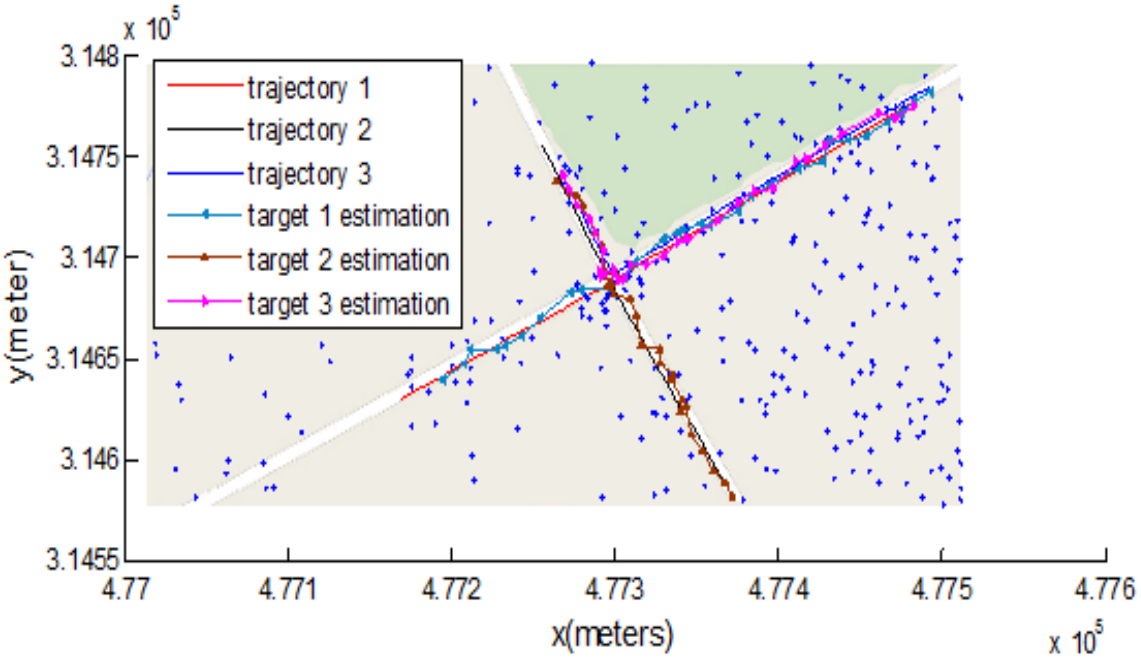
where $\boldsymbol{d}_{ij}$ represents the relative distance between vehicle i and vehicle j in a Cartesian coordinate. A threshold value D_t is defined for interaction force so that repulsive behaviour is activated only if the relative distance $\boldsymbol{d}_{ij}$ is less than D_t.

Parameters setting for the DMHE-MHT: The lifetime threshold is defined as 5 in the MHT implementation, which means any new target can only be confirmed if successfully detected in 5 consecutive time steps. Similarly, tracking any existing target will be terminated after miss detection of 5 sequential time steps. The horizon length used in the MHE is set as 4 and so as for N-scan pruning. Since only a small number of targets are considered in this study, at each time step, 3 new hypotheses generated from one existing parent hypothesis are kept so as to reduce the computational cost.

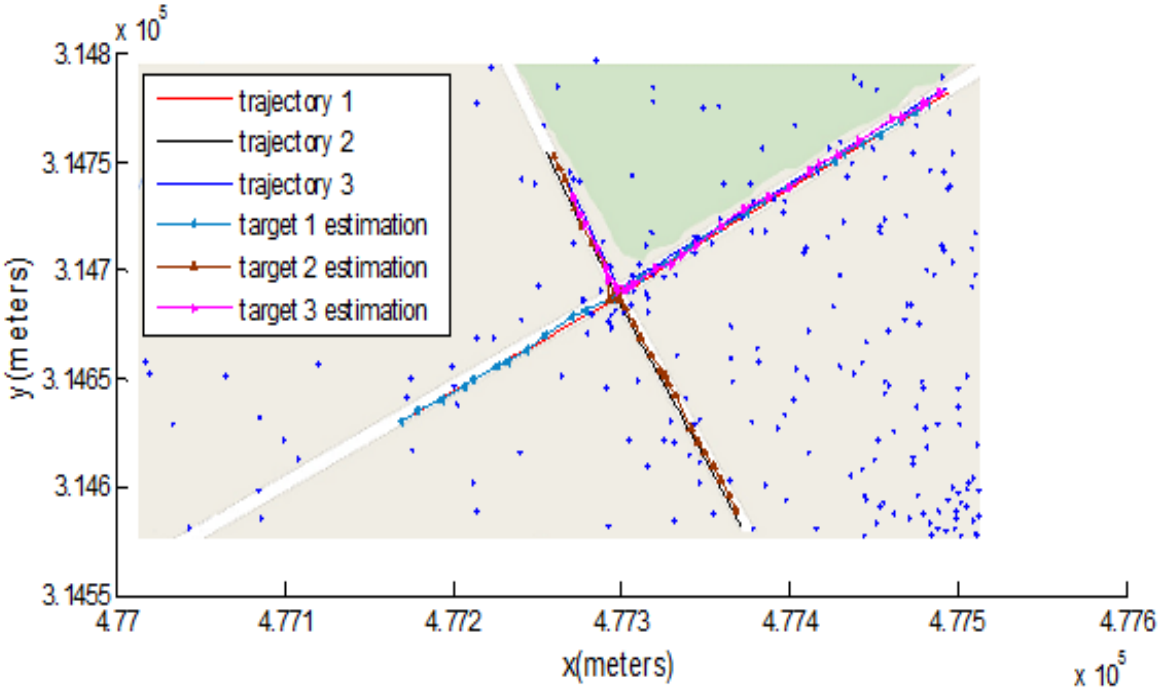
The position estimates are shown in Figure 5.8 and it can be shown that the road constraint and force based interaction play a significant part for improving the tracking accuracy. By comparing Figure 5.8 (a) and (b) it is shown that map-based road boundary constraints improves the overall tracking results significantly. Due to the inequality state constraints, the vehicle positions are constrained within the road. The results are getting even better after introducing the force-based interaction information. In this case, the estimated vehicle trajectories are not only limited within the road boundaries but also get closer to the real trajectories.


x 10^5
3.148
3.1475
3.147
3.1465
3.146
3.1455
y(meters)
trajectory 1
trajectory 2
trajectory 3
target 1 estimation
target 2 estimation
target 3 estimation
4.77
4.771
4.772
4.773
4.774
4.775
4.776
x(meters)
x 10^5

(a)


x 10^5
3.148
3.1475
3.147
3.1465
3.146
3.1455
y(meter)
trajectory 1
trajectory 2
trajectory 3
target 1 estimation
target 2 estimation
target 3 estimation
4.77
4.771
4.772
4.773
4.774
4.775
4.776
x(meters)
x 10^5

(b)



(c)

Figure 5.8 Multiple target tracking using (a) EKF-MHT; (b) CMHE-MHT with road constraint ; (c) DMHE-MHT with force interaction model and road constraint .

For further comparing different algorithms, 50 trials of Monte-Carlo simulations are performed. The performances of different algorithms are measured using the root mean-square error (RMSE). As shown in Table 5.3, the DMHE-MHT gives the best tracking results for all three targets by introducing both road boundary constraint and force based interaction. A more remarkable performance improvement is obtained for target 3 as it has the most interactions with the road and other incoming vehicles.

Table 5.3 Averaged RMSEs for three vehicles by different approaches.

	EKF-MHT	CMHE-MHT	DMHE-MHT
Overall RMSE position (m)	8.9004	5.6353	5.0077
RMSE for Target 1(m)	6.9271	5.4747	5.1271
RMSE for Target 2 (m)	8.7000	5.3629	4.8760
RMSE for Target 3(m)	11.0740	6.0683	5.0200

The EKF-MHT, the CMHE-MHT and the proposed DMHE-MHT are also compared using the OSPA [130]. The OSPA is proposed for evaluating the performance of MTT algorithms, which considers not only the estimation performance but also association accuracy. The OSPA metric computes the distance between two sets of tracks by adding the error between target labels (or target indices) to the spatial distance. As can be seen in Figure 5.9, the DMHE-MHT has the smallest OSPA value, which represents the smallest estimation error and least amount of incorrect data association. Besides, the proposed DMHE-MHT algorithm performance is more stable than the others by observing the variation of the OSPA distance over time, which presents the smoothest OSPA results.

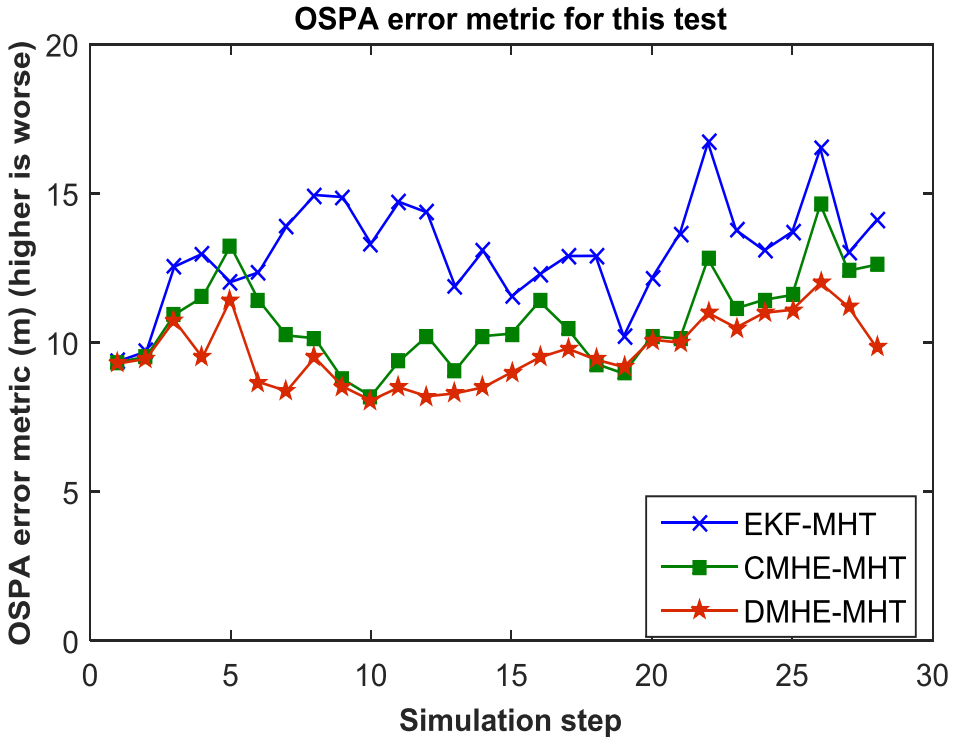


Figure 5.9 OSPA(m) for different algorithms.

Table 5.4 Computational cost for different approaches.

	EKF-MHT	CMHE-MHT	DMHE-MHT
Mean computation time (s)	16.464987	25.443732	24.135973

The computation time of tracking algorithms is compared as shown in Table 5. Each algorithm is run on a 2.4 GHz PC for a hundred Monte Carlo simulations. The original MHT, using the EKF for state estimation considering no extra environmental information, shows the fastest computation time as expected. Comparing with the EKF, the MHE requires a higher computation cost due to the nature of optimisation based on the quadratic programming. However, it still shows an acceptable computational load for a real time application. Note that the computational cost for the MHE heavily relies on the efficiency of the optimisation method. In this study, the optimisation toolbox in the MATLAB software is used. There is only a slight computation time difference between the CMHE-MHT and the DMHE-MHT while

the DMHE-MHT actually shows a better result. This is because after introducing the interaction information and using the improved MHT data association process, the relatively poor data association branches with a low probability are trimmed from the whole MHT hypotheses tree; thus less time is wasted on the unnecessary data association process.

5.6 Summary

This chapter has proposed a new model-based ground vehicle tracking method considering domain knowledge in a comprehensive way. The main contribution comes from the use of interaction models with target tracking. In particular, the physical road constraint together with a force-based dynamic model representing interactions between the target and the environment is used in the DMHE target tracking approach. This DMHE is further extended to the DMHE-MHT to deal with target association ambiguity, noisy measurements and multiple road model transition in multiple target tracking. By comparing the DMHE based approach with traditional constrained state estimation methods using numerical simulation studies, it was shown that a significant improvement can be obtained in terms of target position estimate. Besides, the simulation results also showed that the proposed DMHE-MHT algorithm provides the most accurate tracking performance and robustness for an unknown and time varying number of targets observed in clutter environment using real road map constraint information and force-based target interaction information. To further verify the benefit and effectiveness of the proposed algorithm, real world experiments with actual sensor measurements will be considered as future work.

Chapter 6

Path Planning in Dynamic and Partially Known Environment with Trajectory Prediction

6.1 Introduction

Recently, various advanced driver assistant systems (ADASs) have successfully incorporated into the automotive industry to realise a fully or semi-autonomous vehicle. Most ADASs are used to assist human drivers with the main purpose of improving driving safety while reducing poses new aims to reduce the amount of traffic accidents which are mostly caused by human errors. Here, one of the most important research challenges is to execute autonomous driving and collision avoidance in a partially known dynamic environment.

This chapter particularly focuses on path planning for autonomous vehicle, one of the fundamental tasks of autonomous robots and vehicles [147]. In autonomous navigation tasks [148], the path planning is concerned with finding a path or a trajectory that leads the vehicle from its current state to a desired final state while avoiding collisions with obstacles in the environment. For this purpose, usually a given map of the environment is used together with knowledge of the kinematic and dynamic models of the robot [147]. However, in order to achieve collision-free autonomous driving in a realistic environment, the path planning approach must be

able to deal with moving objects considering motion uncertainties that were not considered on the map when the path was generated, but are detected at the time of execution using sensors and target tracking system (namely, perception information).

In order to get the perception information, in this thesis, sensing and MTT algorithms are developed to estimate the states (e.g. position, velocity and acceleration) and relevant motion uncertainty of the surrounding moving objects which have been discussed in Chapters 3 and 4. In order to achieve a more effective and accurate sensor information and reduce the perception uncertainty, the domain knowledge such as the operation environment, the rules of the road and interactions between the environmental objects are also accommodated in the developed MTT algorithms in Chapter 5. This perception information will then be used in the path planning system to achieve early warning and collision avoidance in this chapter.

Based on the availability of environmental information, path planning problems can be roughly classified into two levels [149]. The higher level global planning primarily concerns environments where workspace information about stationary obstacles and map are known in advance. Path planning in this case is the problem of finding a geometric feasible path from a known initial position to a given goal. Unlike path planning problems in traditional indoor mobile robots assuming deterministic motion and full knowledge of the environment, the outdoor autonomous vehicle motion planning problems are more complicated due to the vehicle kinematic and dynamic constraints while avoiding both stationary and moving objects; global planning is not suitable to deal with dynamic environment. In this case, the lower level local planning approaches need to be implemented for autonomous vehicle which concerns a partially known dynamic environment.

In recent years, sampling based planning methods [150] have proved great success in path planning problems. These methods are probabilistically complete, in other words able to find a feasible path relatively quickly, even in high dimensional configuration spaces. Among them, the RRT algorithm, in particular, handles high dimensional kinodynamic systems including differential constraints effectively while avoiding the state explosion problems [81] that are often found in classical methods such as A*. These features make the RRT particularly suitable for autonomous vehicle path planning problems.

One major problem of standard RRT is that it only tries to find a feasible solution as quickly as possible while does not use a metric to measure the optimality of the

trajectory between the initial state and the other nodes. Usually, the algorithm is not terminated after finding the first feasible solution. The remaining computation time is used to optimize the trajectory with respect to a cost function. In this case, the RRT* [151] is introduced, which addresses the optimality problem of the RRT-based planning algorithms. RRT* achieves this by incrementally rewiring the tree as lower cost trajectories become available with the addition of new nodes to the tree. Under certain assumptions, this algorithm converges to the optimal solution as the number of samples reaches infinity [81]. RRT* does not guarantee upper bounds for the computation time of the optimal solution which makes it not suitable for real time applications.

Other approaches have also been investigated to improve the standard RRT in other aspects. In [152], continuous cost functions are incorporated in RRT for producing higher-quality paths in a global planning problem. Similarly, in [100] a dynamic model based cost function is used to expand and prune the nodes of RRT tree. Methods for incorporating dynamic environments have also been investigated. Solutions include extending the configuration space with a time dimension ($\mathcal{C} - \mathcal{T}$ space), in which the obstacles are static [154], as well as pruning and rebuilding the tree when changes occur [155] [156]. Other methods [153] use RRT to generate feasible path without guaranteeing the optimal solution and then subsequently optimise the generated path based on different algorithms such as close-loop control methods. In [157], a reachable set approach is proposed for better choices of vertices to expand RRT. Meanwhile in [158], the anytime algorithm approach is used for real time RRT with bounded solution time. The RRT keeps being improved until a new trajectory is required by the robot, which can happen at any time.

On the other hand, methods for incorporating moving objects have also been investigated. Among them, model predictive control (MPC) methods have been taken for solving collision avoidance problem in local planning [75]. MPC is increasingly being applied to autonomous vehicle applications because it naturally combines path planning with on-line stability and convergence guarantees [87]. Furthermore, MPC has the advantage of considering future position of the moving objects and suitable for situations where multiple objects with complex dynamic models are involved. However, one big issue of this optimisation based method is that it may get stuck in a local minimum during execution and MPC itself is not capable of solving path planning problem.

Some approaches have been developed to overcome these problems. In [159], MPC type approaches are used to navigate vehicle in unknown environments by combining with some mapping algorithms. In [160], MPC is used to search the control space deterministically by projecting the vehicle forward along fixed set of elemental paths such as lines, circles etc. In [161], the sampling based MPC (SBMPC) algorithm is introduced for expanding the configuration area for local path planning. When expanding a vertex, different from traditional RRT based sampling method, a random control input is used for sampling. This method can efficiently solve the local minimization problem which is often seen in original MPC based planning. However, the results only consider in planning in static environments.

More recently, in [100], a dynamic model of the robot and MPC cost function is used in RRT to expand and prune the nodes of the tree. However in this paper, MPC was used as a pruning process in assistance of the RRT algorithm for a higher level trajectory generation. In this case, the rigorous guarantees of constraint satisfaction normally provided in MPC approaches are harder to show. Neither were the other relative benefits such as dealing with kinodynamic constraints and solving planning in a dynamic environment. Similarly, in [149] cost functions are incorporated in RRT to make the algorithm able to produce optimized paths for a global planning problem.

In this chapter, the main focus is on collision avoidance in autonomous vehicle path planning. An autonomous vehicle path planning framework is developed, which is able to quickly generate a feasible path in the configuration area as well as integrate information from both digital map (which contains the road map constraints and stationary obstacles) and previously developed MTT system (which provides state and the covariance estimation for surrounding moving objects)

In particular, a modified Rapidly-Exploring Random trees (RRTs) is proposed in this framework as a solution to the find a feasible trajectory through partially known environment. The planned trajectory from RRT is updated by seeding the tree using the best trajectory from the previous iteration, and using a pruning mechanism for selecting the best current planned trajectory. Instead of planning an entire trajectory from initial point to the goal region as in the RRT, the modified RRT is used to plan only a short segment trajectory at each iteration. When planned, an optimization based MPC approach is used for executing the planned trajectory aiming towards the nodes calculated by the modified RRT. In order to deal with dynamic environment, the on-board sensor is utilized to estimate the position and velocity of moving vehicles using

the proposed MTT structure which incorporates the domain knowledge as discussed in previous chapters. When planning, instead of only relying on the estimated position, the motion uncertainty of other moving vehicles is also considered which is presented as an uncertain region defined by state mean under a specified confidence level represented by the covariance.

Furthermore, to achieve a collision free manoeuvre, the motion uncertainty of the moving vehicles from current to the multiple step ahead are predicted in a stochastic way using the KF. The predicted prior error covariance for each vehicle is used to represent the motion uncertainty along the predicted trajectory mean and used in the MPC optimization function. The planning problem then becomes to optimize a trajectory cost function through a dynamic potential field.

The reset of this chapter is organised as follow: The overall logic and structure of the proposed algorithm is explained in Section 6.2. Section 6.3 explains how the modified RRT is used for trajectory generation. The MPC optimization is formulated and the trajectory prediction method is described in Section 6.4. In Section 6.5, the algorithm is demonstrated in a simulated road intersection environment with moving vehicles. Finally, concluding remarks are made in Section 6.6.

6.2 System Overview

6.2.1 System Architecture

In order to achieve collision-free autonomous driving in partially known environment with both stationary obstacles and moving objects, several realtime systems must be interoperated, usually including environmental perception, localization, planning and control (as discussed in Chapter 1).

As shown in Figure 6.1, the system architecture considered in this chapter is simplified into two layers with three main components, sensing and extra domain knowledge, environmental perception and path planning. The raw data provided by different perception sensors such as cameras, radar and LIDAR are used in the proposed MTT algorithm (as mention in previous chapters) to understand the surrounding environment of the ego vehicle. Extra domain knowledge is also considered to achieve better tracking performance. Estimation of the states and covariances of surrounding moving objects provided by the MTT system is essential

for the path planning layer to achieve an early warning and collision-free behaviour. Meanwhile, finding the ego vehicle's pose or configuration in the surrounding environment namely localization is achieved by on-board sensors such as GPS, an inertial measurement unit (IMU), odometer and etc. This chapter assumes that the ego vehicle has perfect position information from GPS and a predefined road map without localization problem.

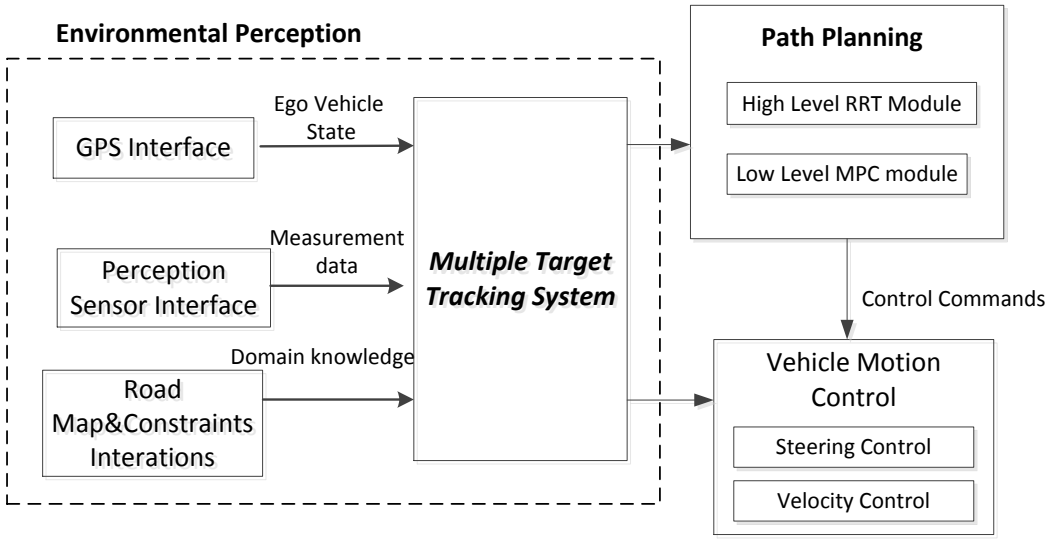


Figure 6.1 System Architecture

For path planning, a two level hierarchical scheme is adopted as shown in Figure 6.2 for details. The requirement for the algorithm is to minimise the probability of collision, while performing local path planning through a partially known environment considering road constraints and moving objects.

The high level path planning module generates the directions and way-points by using a modified RRT algorithm. Different trajectories are classified according to the nodes in the RRT 'tree' while only the 'best trajectory' is collected by a pruning process. The low level MPC motion control computes the control commands for the autonomous vehicle attempting to follow the 'best trajectory' under the vehicle's dynamic and kinematic constraints meanwhile plans a feasible collision free path for a certain horizon period of time in future. At each iteration, the planner uses sensor (perception) information to update the estimates of the states of dynamic targets in the configuration area. By using the perception information, the MPC approach is taken

to accommodate the motion uncertainty and prediction of the moving objects so as to prevent collision in the near future. When the short segment trajectory is executed, a new one is calculated online and the entire process is repeated.

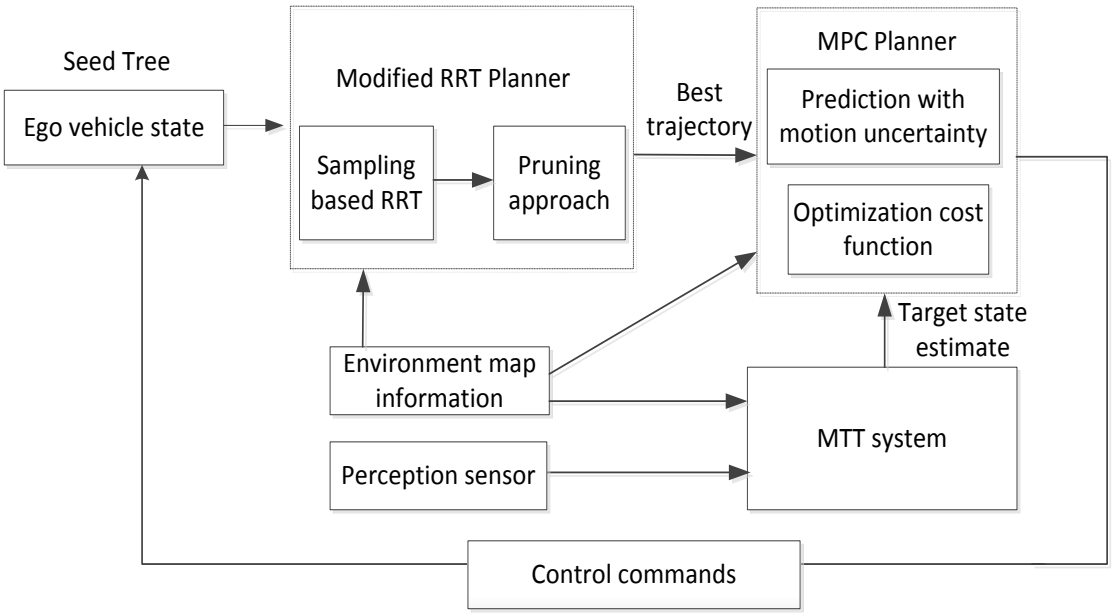


Figure 6.2 Overview of path planning structure

6.2.2 Kinematic Vehicle Model:

The ego vehicle is modelled as a simplified non-holonomic car-like model with second order dynamics due to its relative lower cost and convenience for applications. It is assumed that the vehicle is a rigid body and has non-deforming wheels. Due to the presence of non-holonomic constraints in its kinematic model, it is also assumed that the vehicle moves without slipping. Under these assumptions, the nonlinear mode of an Acker-man steered vehicle [162] is shown in Figure 6.3.

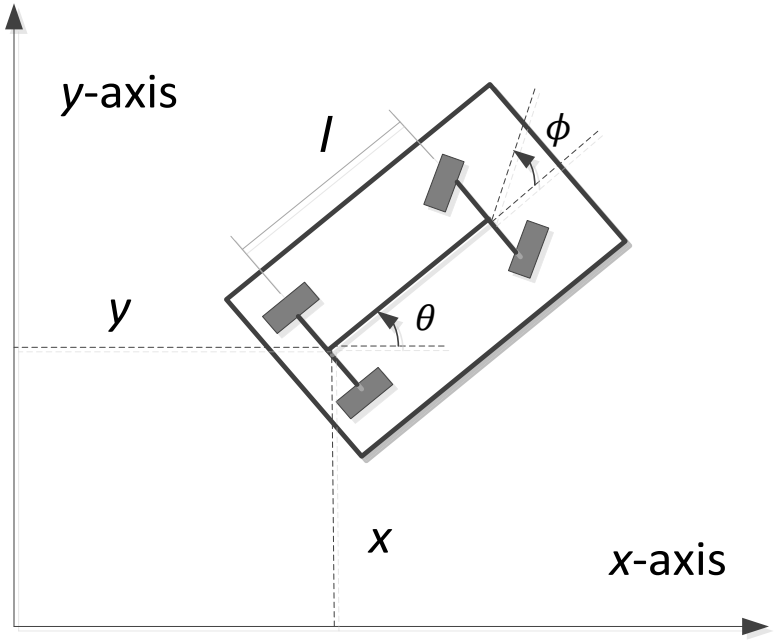


Figure 6.3 A simplified ego vehicle model for planning

In this model, the state $q = (x, y, \theta, \phi)$ of the vehicle is a four-dimensional vector consisting of the Cartesian coordinates of the centre point of the rear axle (x, y); the heading (orientation) angle θ with respect to the horizontal x axis and its steering angle ϕ. T is the sampling time and l is the longitudinal wheel separation as shown in Figure 6.3. The system control input $u = (u_1, u_2)$ is a two-dimensional vector consisting of the vehicle heading velocity u_1 and the steering rate u_2. This gives the following non-linear model shown in (6.1)

$$q(k+1) = f(q(k), u(k)) =$$

$$\begin{bmatrix} x(k+1) \\ y(k+1) \\ \theta(k+1) \\ \phi(k+1) \end{bmatrix} = \begin{bmatrix} x(k) + T \cdot u_1(k) \cdot cos\,\theta(k) \\ y(k) + T \cdot u_1(k) \cdot sin\,\theta(k) \\ \theta(k) + T \cdot u_1(k) \cdot \frac{tan\,\phi(k)}{l} \\ \phi(k) + u_2(k) \cdot T \end{bmatrix} \tag{6.1}$$

where the non-holonomic constraint can be expressed as:

$$\dot{x}(k)\, sin\,\theta(k) - \dot{y}(k)\, cos\,\theta(k) = 0 \tag{6.2}$$

Note that when $\phi(k) = \pm\frac{\pi}{2}$, the model becomes singular. This corresponds to the situation where the front wheel heading is orthogonal to the car longitudinal axis. In practice, the range of the steering angle ϕ is restricted to prevent this singular case. This is also considered in the simulation section in this chapter.

6.3 Trajectory Generation

In this chapter, a 2D configuration space is considered. $q(k) \in \mathrm{X}$ and $u(k) \in \mathrm{U}$, where $\mathrm{X} \in \mathbb{R}^{\mathrm{d}}$ and $\mathrm{U} \in \mathbb{R}^{m}$, denote the ego vehicle state space and the control input space respectively at step *k*. It is assumed that time is discretized into stages of equal duration, and that applying a control input $u(k)$ at step *k* brings the ego vehicle from state $q(k)$ to state $q(k+1)$, according to (6.1). Let $\mathrm{X}_{\mathrm{obs}}$ denote the obstacle region, and $\mathrm{X}_{\mathrm{free}} = \mathrm{X} \backslash \mathrm{X}_{\mathrm{obs}}$ define the obstacle-free space. Bounds on the control input, and constraints of various conditions, such as stationary and dynamic obstacles avoidance and the physical rules of the environment, can be captured with a set of constraints imposed on the states and the inputs, $q(k) \in \mathrm{X}_{\mathrm{free}}$, $u(k) \in \mathrm{U}$. The situation considered in this chapter is a partially known environment with a given environmental map representing the stationary obstacles and road boundaries.

If given a start sate $q_0 \in \mathrm{X}$ where the vehicle initializes at time *k*=0 and a goal region $\mathrm{X}_{\mathrm{goal}} \in \mathrm{X}$ where the ego vehicle aims to go. The motion planning problem is then to find a series of feasible path states $q(k) \in \mathrm{X}_{\mathrm{free}}$ and control inputs $u(k)$ from an initial state to the goal region that obeys the system constraints.

6.3.1 Modified RRT Planning

The standard RRT algorithm grows and maintains a tree where each node of the tree is a point (state) in the workspace, presented in Figure 6.4. The area explored by the algorithm is the area occupied by the tree. Initially the algorithm starts with a tree which has source as the only node. At each iteration the tree is expanded by selecting a random state and expanding the tree towards that state. Expansion is done by extending the closest node in the tree towards the selected random state by a small step. The algorithm runs till some expansion takes the tree near enough to the goal. The size of the step is an algorithm parameter. Small values result in slow expansion, but finer paths or paths which can take fine turns. The tree expansion may be made

biased towards the direction of the goal by selecting goal as the random state with some probability. The high level logic for standard RRT can be found in appendix.

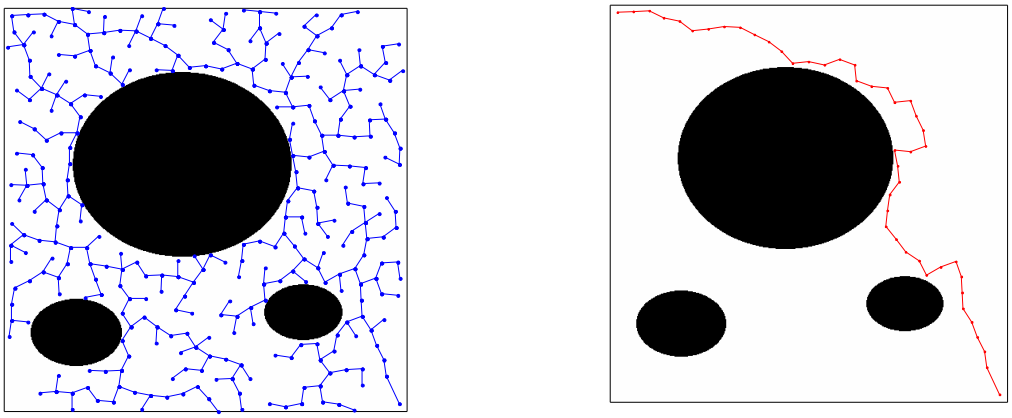


Figure 6.4 Illustration of standard RRT

One major problem of standard RRT is that it is not feasible for real time vehicle planning in partially known dynamic environment. For the dynamic environment with limited computational time and moving objects, the ego vehicle needs to plan for the next couple of steps while making a control decision for the current movement. This means that the predefined goal region is only suitable for high level planning but not short segment aiming direction for the ego vehicle. Instead of expanding the RRT tree for searching in the entire configuration area to find a feasible path, the limited time should be used to focus on computation for desired areas of the configuration space in autonomous vehicle path planning problem.

The modified RRT algorithm in this chapter as shown in Table 6.1 aims to solve the path planning problem by using an on-line updating strategy. Here different from the standard RRT where the extending process is repeated indefinitely till a new vertex reaches the goal region, the modified algorithm focus on keeping a desired direction using only limited steps. In other words, instead of searching for a complete feasible path from the initial position to the goal region, only a small portion of the planned trajectory is executed by the MPC scheme, while the planner is restarted to plan a new trajectory on-line. To facilitate this, the stopping condition in line 3 is changed. When the ego vehicle needs a new trajectory, or when certain maximum number of vertexes (*Nmax*) have been satisfied, the modified RRT is stopped. The *Nmax* is adjusted dynamically depending on the current state of the ego vehicle

(position, forward velocity and orientation angle) and the environmental constraints such as road structure and stationary obstacles given by the road map.

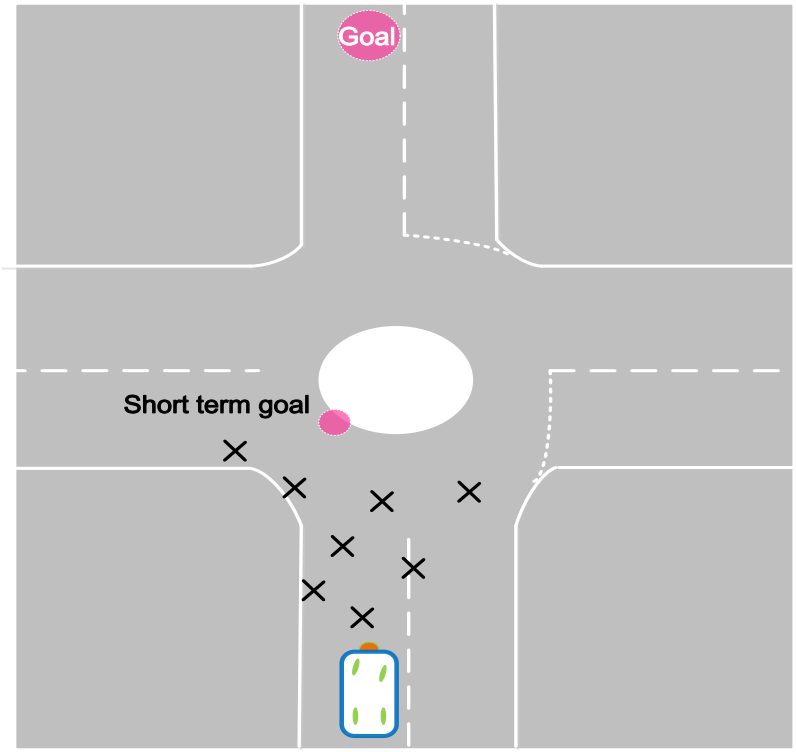


Figure 6.5 Modified RRT on-line planning strategy

As illustrated in Figure 6.5, the path planning strategy is implemented in a roundabout scenario. The ego vehicle needs to drive through the roundabout with road structure given by a known map. Since the location of stationary obstacles is known by the map, instead of generating trajectories cross the whole scenario from the start point to the final goal, the planning problem can be separated into several segments. E.g. we can take the location of the edge of the roundabout as a short term goal and the path planning is only carried out within the desired areas between the short term goal and ego vehicle's current location. The length of the short segment trajectory can be determined by *Nmax* which is calculated based on the ego vehicle dynamic constraints (how far it could move in each step) and environmental constraints (the location of stationary obstacles/road structure). For more complex planning scenarios with more obstacles, a smaller *Nmax* needs to be utilized to generate more precise path so as to avoid collisions. Besides, improvement on the performance would mean an increased computational cost e.g. a smaller *Nmax*

requires more planning iterations repeated overall. Therefore, the tradeoff needs to be considered between *Nmax* and the associated computational cost requirement depending on scenarios.

Table 6.1 Modified RRT

Modified RRT for autonomous vehicle planning
RRTmain(Tree)
1. Tree = q_{start}
2. $q_{new} = q_{start}$
3. while (Nnodes<Nmax) do
4. q_{target}= **SampleTarget**()
5. $q_{nearest}$= **NearestVertex**(Tree,q_{target})
6. q_{new}= **ExtendTowards**($q_{nearest}$,q_{target})
7. if q_{new} valid (environmental constraints)
8. **Tree.add**(q_{new})
9. else
10. continue
11. end if
12. end while
13. Trajectorys=**Separate_Tree_to_Trajectories**(Tree.add)
14. Best_trajectory=**Prun**(Trajectorys)
15. **MPC_Trajectory_Execution**(Best_trajectory)

6.3.2 Pruning and Trajectory Selection

Once the RRT 'tree' is built, all the possible trajectories (branches) are scored and pruned based on a minimal cost criteria and only the 'best' trajectory is chosen by using a specific pruning strategy.

At each iteration, the entire tree is classified into different trajectories (branches) by line 13 in Table 6.1. A pruning process in line 14 is designed to pick the best trajectory based on a cost function which is similar to RRT*. The main idea for the pruning strategy is to allow potentially large areas of state space to be ignored if a provably better trajectory has already been found. There are several different

optimality criteria of choosing one possible trajectory over another [163] including: minimum distance, minimum wheel rotation, minimum control effort, optimal surveillance rate and etc. The cost function in this chapter is defined in a rewarding distance such as A* algorithm [164] based on each trajectory length and the minimum distance from the ending node (of each trajectory) to the goal region. As illustrated in Figure 6.6, an RRT 'tree' is generated towards the short term goal (according to Figure 6.5) which includes 6 trajectories in total. Base on the minimal cost criteria, the red trajectory is chosen as the 'best'.

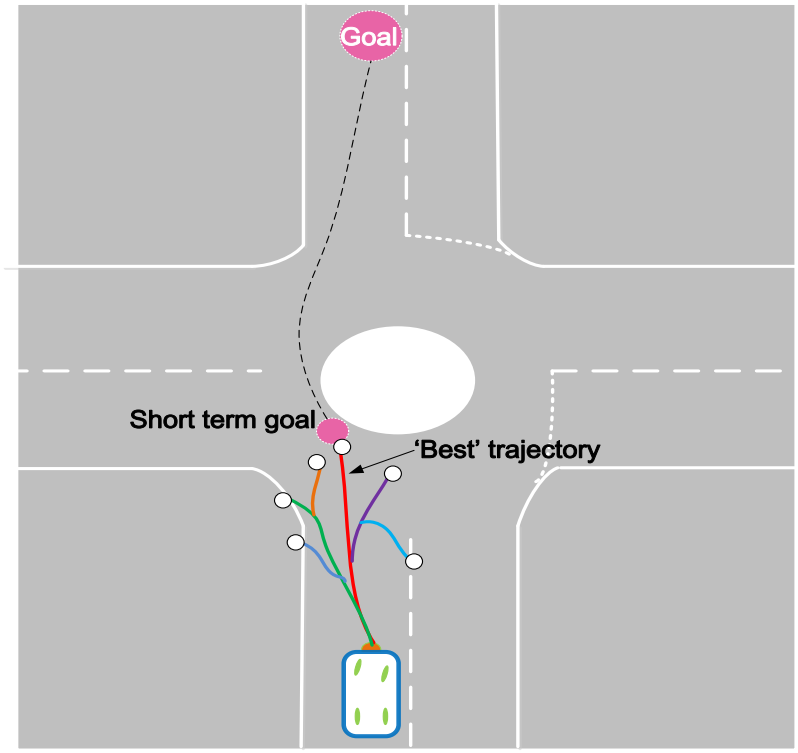


Figure 6.6 Modified RRT pruning and trajectory selection

Once the 'best' trajectory is chosen, a MPC approach is then used for executing the short segment trajectory as shown in line 15 in Table 6.1, the details is explained in Section 6.4. When the ego vehicle reaches the short term goal or when certain number of the nodes has been extended, the current iteration of RRT is finished. The new iteration RRT tree is continued from the last node of the RRT 'best' trajectory.

6.4 MPC for Collision Avoidance Path Planning

As discussed above, it is not enough just to use the RRT for autonomous vehicle path planning in a partially known dynamic environment where moving objects are involved. In order to achieve a collision-free maneuver, the autonomous vehicle should not be directed to completely follow the reference given by the high level module which is assumed to be collision-free and achievable by the vehicle in a static environment. The main reasons are that the path generated from high level module for autonomous vehicle is not guaranteed to be collision-free, since the high level RRT module only considers the map information for road structure and stationary obstacles; even if the moving objects are considered in the high level module, the predicted situation from the world in the near future may differ from the actual situation when moving objects change their trajectories during execution which may result in invalid paths.

In this case, a MPC strategy needs to be used to optimise a cost function over a limited receding time-horizon which will safely bring the ego vehicle closer to the goal region. The details of the cost function formulation are explained in the following.

6.4.1 On-board Target Tracking

Firstly, in order to deal with moving objects in a dynamic environment, a MTT algorithm is implemented as discussed in previous chapters. The vehicles are tracked by range and bearing sensors modelled as (6.3), where $x_{G,k}$ and $y_{G,k}$ are the global coordinates of the sensor in *x* and *y* direction while $x_{L,k}$ and $y_{L,k}$ represent the global coordinates of the ego vehicle platform. H_G is the heading of the sensor relative to the platform.v_k is a two-dimensional Gaussian zero-mean measurement noise. The details of the proposed MTT algorithm can be found in Chapter 4.

$$z_k = \begin{bmatrix} \sqrt{(x_{L,k}-x_{G,k})^2+(y_{L,k}-y_{G,k})^2} \\ arctan\left(\frac{(y_{L,k}-y_{G,k})}{(x_{L,k}-x_{G,k})}\right)-H_G \end{bmatrix} + v_k \tag{6.3}$$

Comparing with traditional target tracking algorithms, the proposed method has the advantage of incorporating additional domain knowledge for ground vehicle

tracking such as road weight constraint, speed limit and target interaction information. As a result, the tracking performance is improved by achieving a more accurate state estimation. This is very important to achieve collision avoidance behaviour in autonomous vehicle path planning problems especially when multiple moving targets are involved in. The estimated posterior state and error covariance for each target will be used for trajectory prediction and further more in MPC based motion planning process.

6.4.2 Trajectory Prediction with Motion Uncertainty

In order to plan a collision free trajectory, the motion planning algorithm needs the ability of 'looking ahead' for several steps based on the current states of the dynamic environment. The trajectory prediction process in this chapter relies on the target vehicle current state and kinematic motion models. It is assumed that the driver's behaviour is not significant changed during the prediction period from k to $k+N$. The future motion is then predicted using vehicle's dynamic models such as CV, CA and CT. Considering a discrete time linear system, in each prediction step of the KF, the state vector x and error covariance matrix P are estimated using:

$$\hat{\mathrm{x}}_{k|k-1} = A\hat{\mathrm{x}}_{k-1|k-1} + Bu_{k-1} \tag{6.4}$$

$$\widehat{\mathrm{P}}_{\mathrm{k|k-1}} = A\mathrm{P}_{\mathrm{k-1|k-1}}A^T + Q \tag{6.5}$$

where A is the process model and B is the control input matrix. $\mathrm{x}_k = [x_k, y_k, \dot{x}_k, \dot{y}_k]^T$ is the state vector which contains position and velocity for the target vehicle in x and y axis respectively. $\hat{\mathrm{x}}_{\mathrm{k|k-1}}$ is the predicted state, $\widehat{\mathrm{P}}_{\mathrm{k|k-1}}$ is the predicted error covariance which represents the motion uncertainty. The predicting process is performed for a horizon of N steps. At each iteration, the predicted state and error covariance generated from last time step are used as prior information.

However due to the road boundary considered in this thesis, the road inequality constraints such as equation (4.4) in Chapter 4 are also accommodated in the predicted state. In this case, the constrained state/covariance prediction is considered in this chapter as shown in equation (4.12) and (4.13) where the detail of the explanation can be found in Section 4.5 in Chapter 4.

By repeating the KF Gaussian prediction process, one can obtain a mean and error covariance matrix of the state of the target vehicle for future time steps, which can be transformed into a mean trajectory with associated uncertainty (Gaussian distribution at each time step). In Figure 6.7, each ellipse of the target vehicle is drawn by the predicted error covariance matrix with a certain level of confidence and a predicted mean shown by a mass point in center.

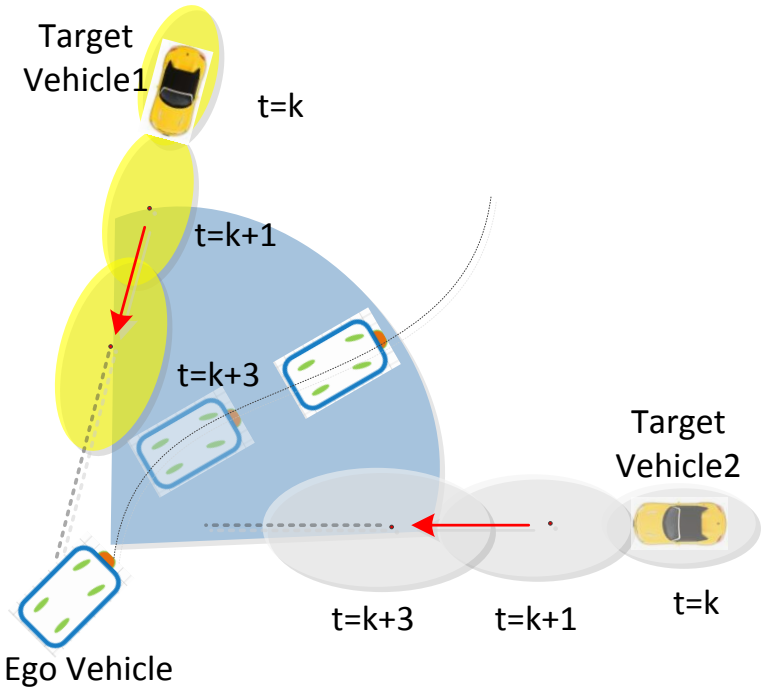


Figure 6.7 Trajectory prediction for a target vehicle with a constant velocity motion model and Gaussian noise. Ellipses (predicted error covariance) represent prediction uncertainty.

6.4.3 Dynamic Cost Function

Based on the 'best trajectory' provided by the modified RRT algorithm as well as the dynamics of the vehicle and the *a priori* road map of the environment, the planning problem is then formulated as optimizing a trajectory cost function for a limited time horizon. At each time iteration, the on-line MTT algorithm uses sensor information to update estimates of the position and velocity as well as the error covariance of the target vehicles, and the relevant predictions for the time horizon are also calculated. Using these estimates and predictions, and the short term goal provided by the 'best

trajectory' from the modified RRT, a feasible collision-free path over the next few seconds is generated.

The cost function in the MPC problem formulation includes with four different aspects: 1) a penalty cost associated with the ego vehicle position avoiding stationary obstacles in the environment given by the road map; 2) a penalty cost associated with the ego vehicle position avoiding other target vehicles in the region of interest; 3) a rewarding cost associated with the ego vehicle moving towards the short term goal; 4) a penalty cost avoiding excessive control inputs considering the energy.

1) Cost related to stationary obstacle: A road map including all stationary obstacles is assumed to be given as prior information e.g. road structure constraints. For the RRT layer, the road structure information is represented by occupancy grids, which indicate if a grid is occupied by the road structure. The feasible path is only generated in the obstacle-free region $\mathrm{X_{free}}$. In addition to the occupancy grids, in the MPC cost function, the road network structure such as the roundabout shown in Figure 6.6 is considered as a stationary obstacle. Higher costs are assigned to the position closer to the road structure areas.

The Euclidean distance between the stationary obstacle and the ego vehicle is used to calculate the cost function as given in (6.6).

$$c_s\big(q(k)\big) = \omega_s s\ (q(k)_{1:2} - L_s) \tag{6.6}$$

where ω_s is a constant determining how much the ego vehicle should keep away from the stationary obstacle, $q(k)_{1:2}$ represents the current 2-D location of the ego vehicle. L_s represents the location of the centre of the stationary obstacle. $s(.)$ depends on the shape of the stationary obstacle, e.g. rectangle, circle, ellipse etc.

2) Cost related to moving target vehicles of interest: the cost function (6.7) is designed as a form of normal distribution for a target vehicle. It is assumed that the target vehicles are considered as a mass point without a geometric size. As explained before, the motion uncertainty of each target vehicle is represented by its error covariance centred at the mean position. In this case, the ellipse defined by the error covariance then can provide a safe margin such that the ego vehicle dose not collide with other moving vehicles. However, in order to generate a collision-free path in the near future over the horizon of *N* steps, at current time *k*, the predicted position and

covariance for the target vehicle at the future time $k+i$ (where $i=0,\dots,N-1$) are calculated:

$$c_m\left(q(k+i),\widehat{\mathrm{P}}_{k+i|k+i-1},\hat{\mathrm{x}}_{k+i|k+i-1}\right)= \tag{6.7}$$

$$\omega_m \exp\left(-\frac{1}{2}\left[q(k+i)_{1:2}-\hat{\mathrm{x}}_{k+i|k+i-1_{1:2}}\right]^T \widehat{\mathrm{P}}_{k+i|k+i-1}{}^{-1}\left[q(k+i)_{1:2}-\hat{\mathrm{x}}_{k+i|k+i-1_{1:2}}\right]\right)$$

where ω_m is a normalizing constant for target vehicle. $\hat{\mathrm{x}}_{k+i|k+i-1}$ and $\widehat{\mathrm{P}}_{k+i|k+i-1}$ represents the predicted position mean and relative error covariance for the target vehicle respectively at future time $k+i$ which are calculated from the MTT algorithm estimation from last step at $k-1$. The predicted position $q(k+i)$ of the ego vehicle is calculated, following the ego vehicle dynamic equation $f(q(k),u(k))$ in (6.1), which is based on its current position $q(k)$ and a horizon of the predicted control inputs $u(k),\dots,u(k+N-1)$ calculated by the MPC optimization. In this case, a horizon time of prediction information will be considered in the MPC cost function at each time step so as to achieve collision avoidance in path planning. The details will be explained in Section 6.4.4.

Figure 6.8 shows a potential field around a target vehicle which is represented by the Gaussian form cost function c_m. The target vehicle is located at (8m, 8m) with a process noise covariance matrix $Q=diag\{5,2\}\,(m^2)$ which means the vehicle has more moving uncertainty in *x* axis. The contour plot shows the spreads of the cost function. The ego vehicle should try to move around the potential field or at least towards the lower parts of the cost function, i.e. the dark blue areas, and avoid the red area.

Figure 6.9 shows a potential field of the target vehicle with a prediction horizon of *N*=2. The target vehicle is moving along the *x* axis with velocity (5m/s, 0m/s). In this case, the mixture Gaussian function (c_m, *N*=2) for the ego vehicle of its current state (8m,8m) and the one step-ahead prediction is plotted. Comparing with Figure 6.8, the high cost potential field area occupies a larger area where the ego vehicle should not be in order to avoid collision to the target vehicle.

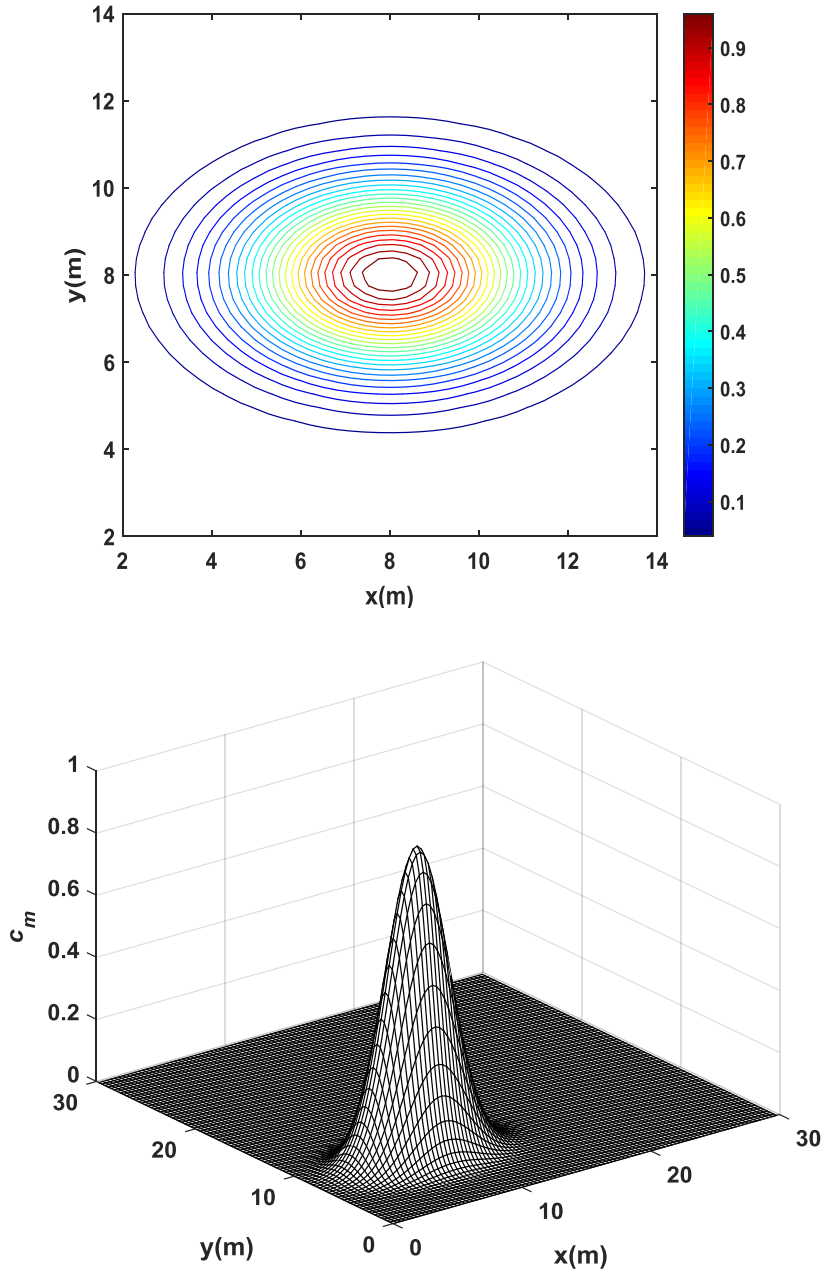


Figure 6.8 The contour plot (upper) and 3-D shaded surface plot (lower) for the cost function c_m with N=1.

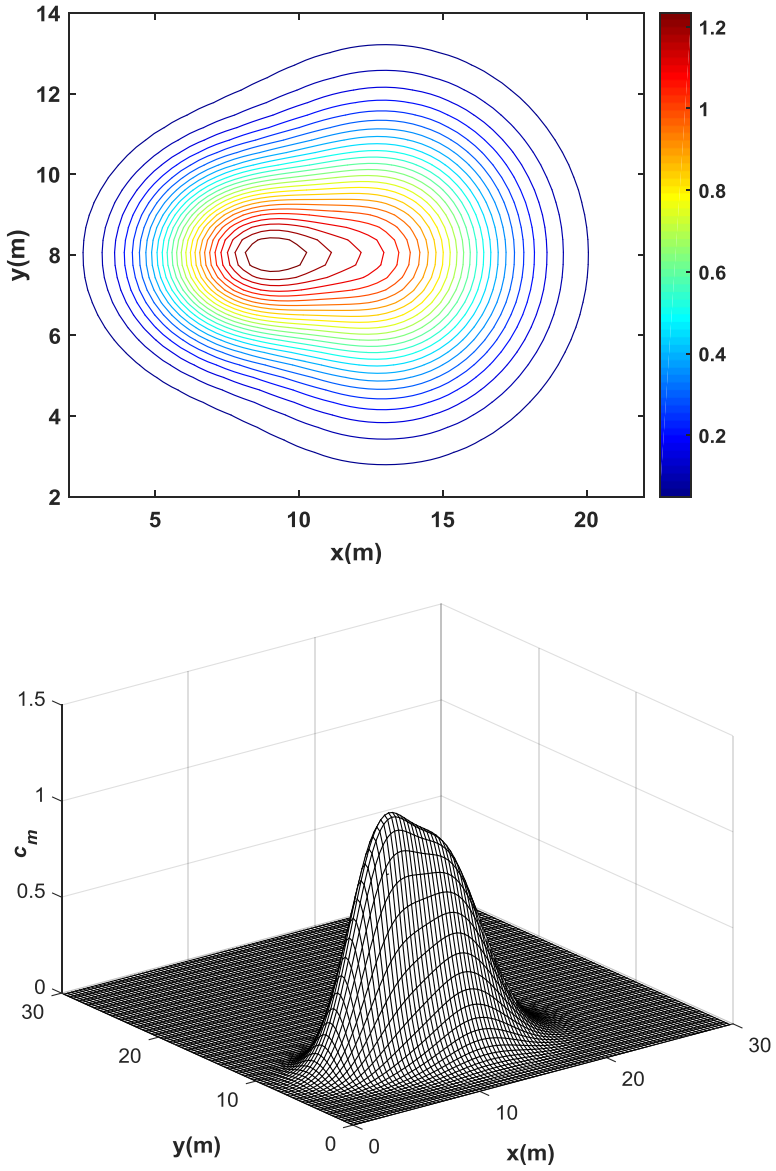


Figure 6.9 The contour plot (upper) and 3-D shaded surface plot (lower) for the cost function c_m with N=2.

3) Cost rewarding moving towards the short term goal: this cost function (6.8) rewards the ego vehicle for moving towards the short term goal of the 'best' trajectory given by the modified RRT layer. The function penalizes if the ego vehicle dose not move forward towards the desired goal. This is important for a path planning problem since the ego vehicle needs to not only avoid stationary obstacle and moving objects but eventually move to the desired location.

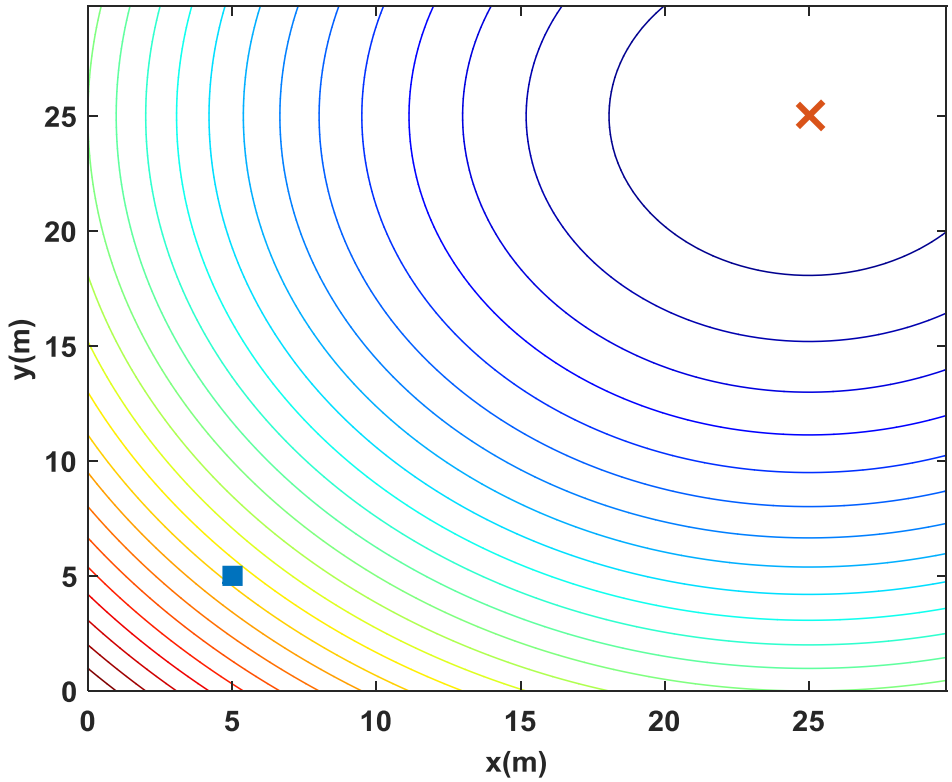


Figure 6.10 The contour plot for cost function c_g with ego vehicle position (square) and short term goal (cross)

Similar to the cost function (6.7) related to moving target vehicle, the cost function (6.8) is defined based on the Euclidean distance between the ego vehicle's current position and the short term goal. It is set up as an exponential function which means that larger distances are penalized much, while it is close to the same value for short distances (getting close enough to the short term goal), i.e. it does not change much if the ego vehicle is only 1 or 2 meters away from the end point. As shown in Figure 6.10, the contour plot shows the spreads of the cost function. The ego vehicle should try to move from its initial position (high cost red area) towards the short term goal (low cost dark blue area).

$$c_g\big(q(k), r(k)\big) = \omega_{g,1} exp\big(\omega_{g,2} |r(k) - q(k)_{1:2}|\big) \tag{6.8}$$

where $\omega_{g,1}$ and $\omega_{g,2}$ are the positive scaling constants and $r(k)$ is the desired short term goal of the 'best' trajectory defined by the modified RRT algorithm. The

selection of the short term goal is the end point of the 'best' trajectory which leads the ego vehicle to move towards the desired direction.

4) Cost penalizing control input considering the energy cost: is the cost function rewards the ego vehicle for generating an energy efficient path as given in (6.9)

$$c_u\big(u(k)\big) = \omega_u {u_2}^4 \tag{6.9}$$

where ω_u is a scaling constant and u_2 is the steering rate control input. The function (6.9) is defined to penalize sharp turns, which reflects the last moment emergency maneuver for collision avoidance. By implementing the prediction information of other moving vehicles, the proposed algorithm should be able to make early warning for possible collisions. In this case, the last moment sharp turns should be avoided and smoother trajectory will be generated. The reason u_2 is raised to the fourth-order is to keep the term closer to zero for any general turns while enlarge the effect of emergency turns.

6.4.4 MPC Minimization with Trajectory Prediction

To achieve the collision free path planning in a dynamic environment with moving vehicles, the discrete time cost function for the ego vehicle to traverse the dynamic potential function is solved by a MPC approach.

The problem can be posed as follows. At current time *k*, the current state $q(k)$ of the ego vehicle is given considering the state transition equation shown in (6.1). The short term goal $r(k)$ of the 'best' trajectory is defined by the modified RRT algorithms. The potential filed of stationary obstacles $c_s\big(q(k)\big)$ is assumed known from a *prior* road map. The behavior of the moving target vehicle is estimated by the proposed on-line MTT algorithm and its future motion is predicted by (6.4) and (6.5) represented by predicted state $\hat{\mathrm{x}}_{k+i|k+i-1}$and predicted error covariance $\widehat{\mathrm{P}}_{k+i|k+i-1}$. At time *k*, the planning occurs considering a horizon of *N* steps prediction information of moving target vehicles. The planning problem then becomes to determine a horizon of control input $u^*_{k:k+N-1}$, which minimizes the cost of the potential function $J(u^*_{k:k+N-1})$ as given:

$$u^*_{k:k+N-1} = arg^{min}_{u^*} J = \tag{6.10}$$

$$\sum_{i=0}^{N-1} \Bigg[c_g\big(q(k+i), r(k)\big) + c_s\big(q(k+i)\big) + c_u\big(u(k+i)\big) + \sum_{j=1}^{n} \Big(c_m\big(q(k+i), \hat{\mathrm{P}}^j_{k+i|k+i-1}, \hat{\mathrm{x}}^j_{k+i|k+i-1}\big)\Big) \Bigg]$$

where $q(k)$ is the ego vehicle's state at time k; c_g, c_s, c_g and c_m represent each individual cost function as explained above. n is the number of detected moving target vehicles. It is assumed that the short term goal $r(k)$ is not changed during the prediction horizon meaning the ego vehicle is heading to the same direction during the horizon of *N* steps. The optimization cost function (6.10) is solved by *fmincon* solver in MATLAB which is able to find the minimum of solution of a constrained nonlinear multivariable function. The details of MPC constraints considered in (6.10) is explained in Section 6.4.5. $u^*_{k:k+N-1}$ determines a horizon of optimized discrete control input sequence from k to $k+N-1$ so as to achieve collision avoidance but also keep moving towards the goal position. The implementation details of the optimization cost function is provided in Appendix D.

However instead of applying the full sequence of control input $u^*_{k:k+N-1}$, only the first control u^*_k at time *k* is directly implemented in the ego vehicle state transition function $q(k+1) = f(q(k), u^*_k)$ in (6.1) considering the ego vehicle's current state $q(k)$. Note that, the next step state $q(k+1)$ of the ego vehicle is executed considering not only the current information but a horizon of predicted information. For autonomous vehicle scenarios with moving objects, the ability of 'look-ahead' is extremely important to achieve collision avoidance in path planning. The above process is repeated at each time instant until the ego vehicle reaches the desired short term goal.

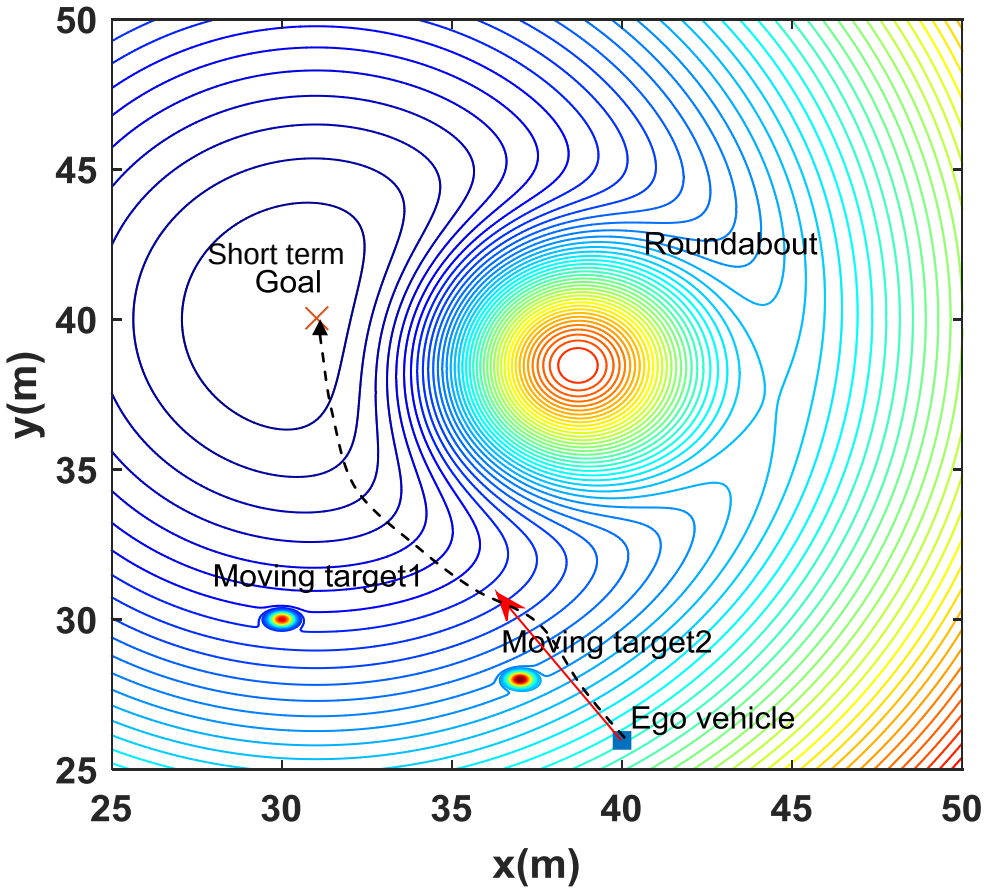


Figure 6.11 The contour plot for a scenario with two moving target vehicles and a stationary roundabout. The ego vehicle should move from current position (square) towards short term goal (cross) while avoid collisions.

Figure 6.11 illustrates an example of a roundabout scenario with two moving target vehicles. A cost function is formed considering the stationary roundabout and two moving target vehicles with N=1. The ego vehicle starts from the current position at (40m, 26m) and heading to the short term goal (32m, 40m). The ego vehicle would run into the moving target vehicle 2 while moving towards the low cost goal area. Therefor it is important to take into account the predicted motion uncertainty of other moving objects while planning the trajectory.

6.4.5 Constraints analysis for MPC

Besides the non-holonomic constraint shown in (6.1), there are other types of constraints encountered in the MPC cost function. The first one deals with constraints

imposed on the control variables $u(k)$, and the second deals with ego vehicle state $q(k)$.

According to the kinematics and dynamics of vehicle model, the constraints, imposed on velocity and steering rate, for path planning problem are specified as:

$$\begin{bmatrix} -I \\ I \end{bmatrix} * (u_k^*) \leq \begin{bmatrix} -ub \\ lb \end{bmatrix} \tag{6.11}$$

where ub and lb represents the upper bound and lower bound of the control input $u_k = (u_1, u_2)$ respectively. Since u_k is a two-dimensional vector consisting of the vehicle heading velocity u_1 and the steering rate u_2, different set of value for ub and lb are defined. The constraints for u_1 represent the limitations for ego vehicle velocity. E.g. maximum speed limit ub given by the road domain knowledge; and positive lower speed limit lb if the vehicle is only moving forward. The constraints for u_2 represent the physical limitation of the turning rate (rad/s) in clockwise (ub) and anti-clockwise (lb).

For the ego vehicle state $q(k)$ according to (6.1), the road boundary constrains as mention in Chapter 4 are also considered on the ego vehicle position state $q(k)_{1:2}$ so as to guarantee that the planned trajectory is executed in the feasible area:

$$a_k \leq C_k(q(k)_{1:2}) \leq b_k \tag{6.12}$$

where C_k is the constraint function. a_k and b_k are the known vectors representing the lower and upper road boundary individually. The details can be found in Section 4.4.2. Other ego vehicle state constraints for heading angle θ and the steering angle ϕ are defined in the similar manner as (6.11).

6.5 Numerical Simulation Results

In this section, the performance of the proposed algorithm is tested with numerical simulations. The algorithm is applied on a road intersection scenario incorporating road map based environmental domain knowledge as well as moving target vehicles. The objectives are to test the collision avoidance performance and robustness of the proposed algorithm.

Simulation scenario setup: The simulation setup is to navigate the ego vehicle starting from position (25m, 5m) to a goal point at (25m, 40m). During the process,

the ego car needs to cross a road interaction with a region of 50m*50m with two incoming vehicles from side roads. The situation considered in this section is a partially known environment with a given environmental map representing the road structure. Any off-road region is taken as infeasible area and thus feasible trajectories are only generated within road region $\mathrm{X}_{\mathrm{free}}$. The path planning problem is then to find a series of feasible path states $q(k) \in \mathrm{X}_{\mathrm{free}}$ and control inputs $u(k)$ from an initial state to the goal region that obeys the system constraints.

MTT: In order to obtain the perception information for moving target vehicles such as state and error covariance, the proposed MTT algorithm is implemented for the ego vehicle. The ego vehicle is equipped with on-board sensors with measurements modelled by equation (6.3).

As shown in Figure 6.12, a road intersection scenario with a rectangular region of surveillance in a clutter environment is considered. Vehicle 1 starts from state (0m, 27m, 12m/s, 0m/s) moving towards the straight right while Vehicle 2 starts from (50m, 23m, -12m/s, 0m/s) moving towards the straight left. Each target vehicle is moving on a single carriage way starting from a position of the middle of the road. Each road is assumed to have a total width of 4 meters and the vehicle's trajectory is limited within the road width constraint. All the vehicles have the same initial speed of 12m/s in a straight direction along the road network and try to follow the road centre line. The road speed limit is set as 13m/s.

The tracking scenario set up is similar to Section 4.6.1 in Chapter 4. The vehicle dynamics is defined as the CV model shown in (3.23) under the global coordinate $\mathcal{G}$ with the state vector $\mathrm{x}_k = [x_k, \dot{x}_k, y_k, \dot{y}_k]^{\,T}$. The sensor sampling time interval is $\mathrm{T_s} = 0.1\mathrm{s}$. Given a road model r, the process noise ω_k^r is defined under the road local coordinate $\mathcal{L}_r$ which is a two-dimensional Gaussian process noise with zero mean and covariance matrix $\sigma_\omega^r = diag\{5,2\}$. This covariance represents higher motion uncertainty along the centre line direction and smaller uncertainty orthogonal to the road. Considering the road inequality constraint defined in (4.7) with state $\mathrm{x}_k^r = [x_k^r, \dot{x}_k^r, y_k^r, \dot{y}_k^r]^T$ under $\mathcal{L}_r$, the global dynamic function can be written below which is similar to (4.21):

$$\mathrm{x}_{k+1} = T_r^g(F \cdot T_g^r(\mathrm{x}_k) + G \cdot \omega_k^r) \tag{6.13}$$

As mentioned, a nonlinear measurement model (6.3) in the global Cartesian coordinate is considered with Gaussian noise v_k and the corresponding covariance matrix R = $diag\{8, 10^{-3}\}$.

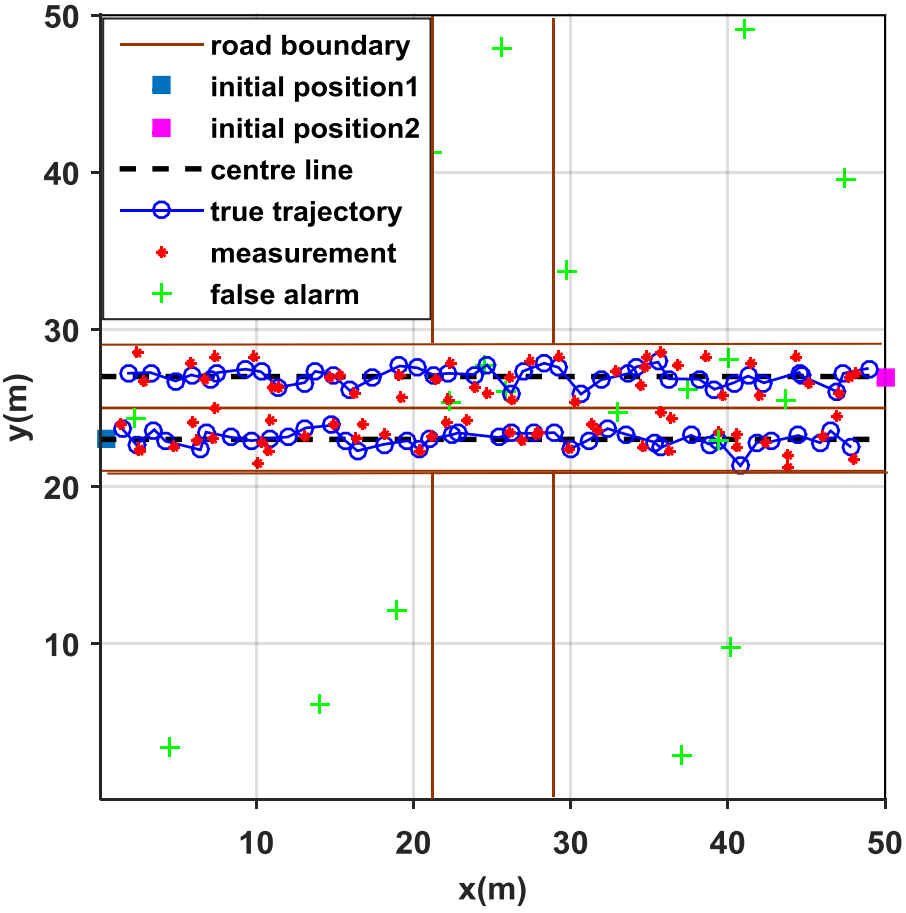


Figure 6.12 MTT in road intersection scenario

Each target is detected with a probability of $P_d = 0.98$. The detected measurements are immersed in a clutter environment that can be modelled as a Poisson distribution with clutter density of $\beta_{FA} = 8 \times 10^{-2}$ (false alarms/area/scan) over the $250m^2$ region (i.e., 20 clutters return over the surveillance region at each scan). New target density is $\beta_{NT} = 2 * 10^{-5}$ and the gate region $P_G = 0.97$.

Parameters setting for Path Planning: The ego vehicle state transition model (6.1) is employed for the simulation with the following constraints and simulation parameters. The constraint for ego vehicle velocity control input u_1 is $[0, 13](m/s)$ corresponding to the road speed limit and the vehicle only move in the single direction. The constraint for steering rate u_2 is $\left[-\frac{\pi}{8}, \frac{\pi}{8}\right](rad/s)$ so as to avoid the

vehicle moving to the opposite wrong direction. The *x* axis position constraint for ego vehicle state $q(k)$ is limited within the road boundary $[29, 29](m)$. The heading (orientation) angle θ with respect to the horizontal x axis is constrained as $\left[-\frac{\pi}{2}:\frac{\pi}{2}\right]$ (rad) meaning it only moves forward. The time interval for the MPC is also set as $\mathrm{T} = 0.1\mathrm{s}$ according to the sensor sampling time interval.

The weights for cost functions are design parameters which determine the ego vehicle's behaviour. The weights ω_m (for moving target vehicle) and $\omega_{g,1}$(for moving towards the goal) specify the relative importance of achieving moving object avoidance and reaching the goal. In this simulation, ω_m=1 is set as and $\omega_{g,1} = 10^{-5}$ which represents that collision avoidance is important, but the ego vehicle also need to keep moving toward the goal position. The choice of the weights will affect the shape of the trajectory that the vehicle will move towards the goal position. Similarly, the weight $\omega_s > 0$ defines how much the ego vehicle should keep away from the stationary obstacle which is chosen individually according to the stationary obstacle function $s(.)$ for different shape of the stationary obstacle. $\omega_u > 0$ is scaling factors that are used to weight the importance minimizing the use of control energy while satisfying the other requires such as achieving the goal and collision avoidance. In this scenario, ω_u is chosen as $\omega_{g,1}/100$ to generate a smooth trajectory and avoid following the wrong direction. Compared to a heuristic algorithm such as Vector Field Histogram [165], the process of parameter selection is relatively simple because each paramater has a clear physical meaning with obvious consequences. The choice of the cost function and relevant weights for evaluating the quality of the plans is scenario dependent.

The main objective for this simulation is to test if the ego vehicle can pass through the incoming vehicle by changing its steering rate and forward velocity while avoiding any collision risks. Different horizon lengths for the moving vehicle motion prediction are considered including *N*=1, 4 and 8.

As shown in Figures 6.13 to 6.15, three different planning trajectories are compared. When the prediction horizon is *N*=1, it means the planner has no prediction ability and as a result it presents a rough trajectory with sharp turn and braking so as to avoid the coming vehicles e.g. the ego vehicle has to make two emergency manoeuvres during t=2 to t=3 sec so as to avoid the coming vehicle. Comparing the position of the ego vehicle at the same time e.g. t=1, 2, 3s using different prediction

horizons, it is clear that when increasing a prediction step to N=4 and N=8, the ego vehicle exploits a prediction information about the near future position of the coming vehicles and decide to maintain its velocity and heading direction while crossing the road intersection (t=1s to t=2s) from energy saving point of view. As a result, both trajectories are smooth and relatively straight without aggressive and sudden change of maneuver.

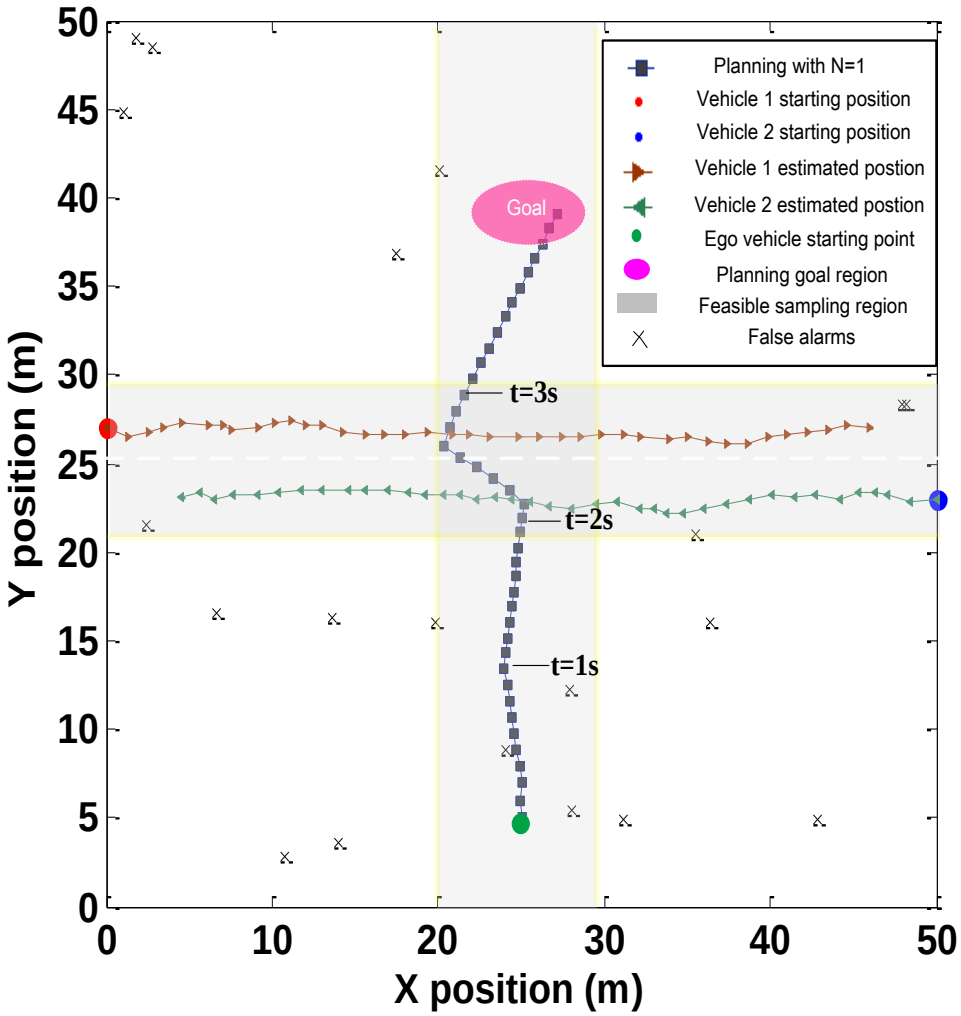


Figure 6.13 Collision avoidance based planning using prediction horizon length N=1

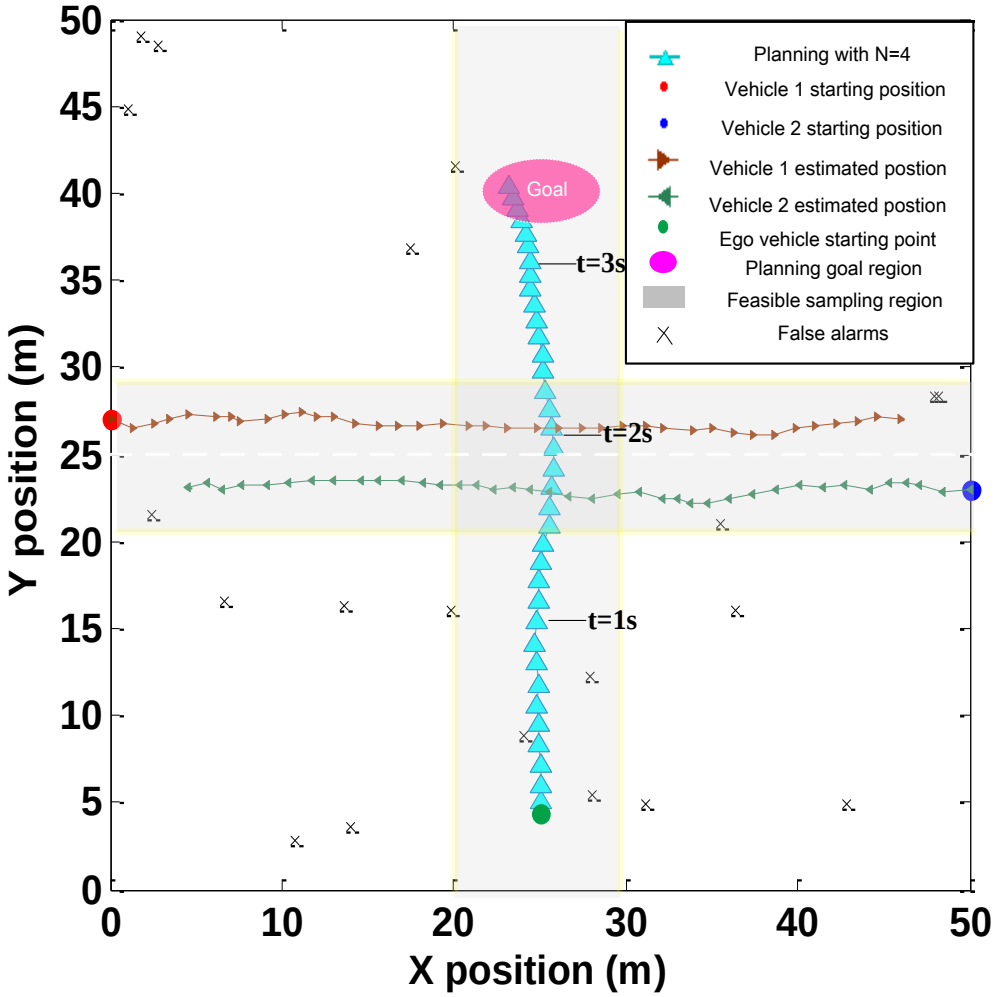


Figure 6.14 Collision avoidance based planning using prediction horizon length N=4

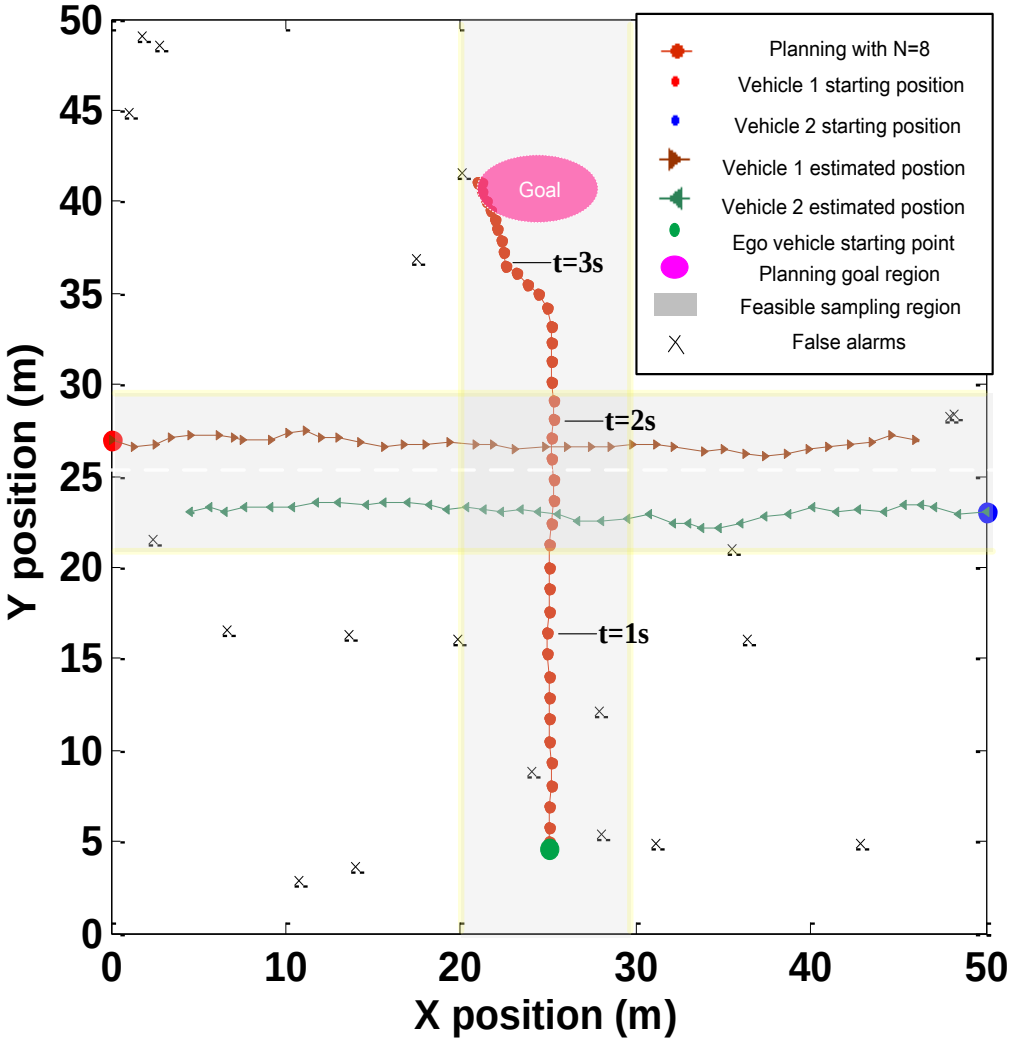
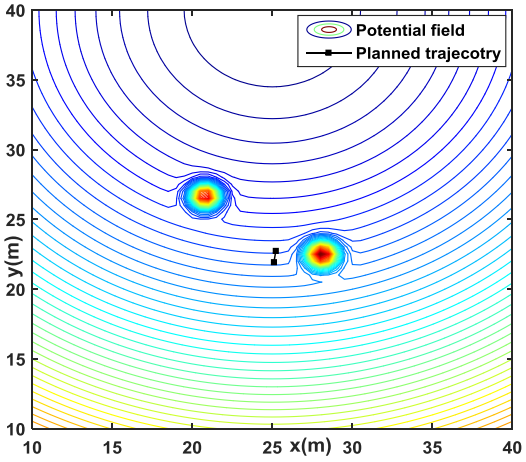
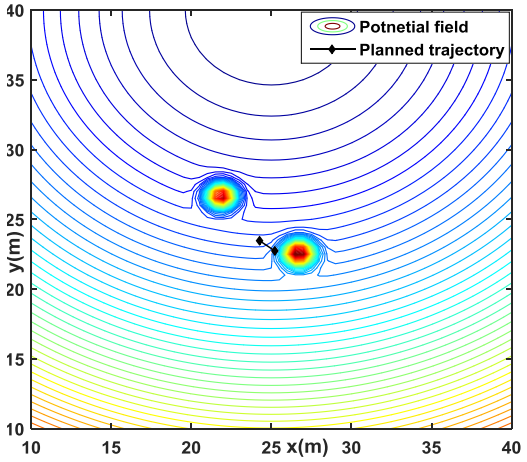


Figure 6.15 Collision avoidance based planning using prediction horizon length N=8

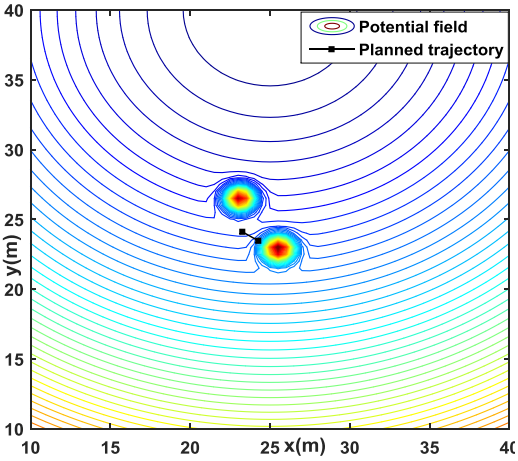
The details of how the ego vehicle avoids coming vehicles are also shown in Figures 6.16 to 18. Figures 6.16 (a) to (f) show the result for the trajectory planning with prediction horizon N=1 from time step 20 to 25; Figures 6.17 (a) to (f) show the result for the trajectory planning with prediction horizon N=4 from time step 16 to 21; Figure 6.18 (a) to (f) show the result for the trajectory planning with prediction horizon N=8 from time step 12 to 17;



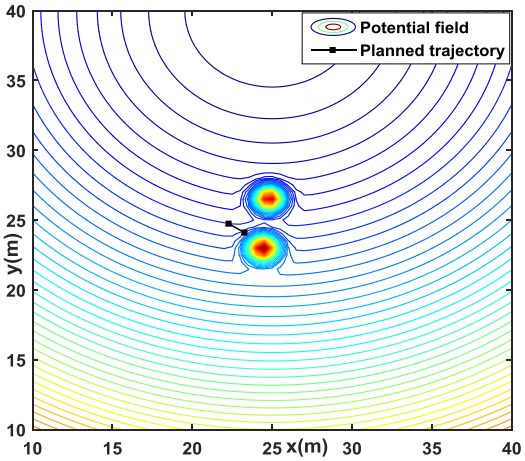
(a) Time step 20



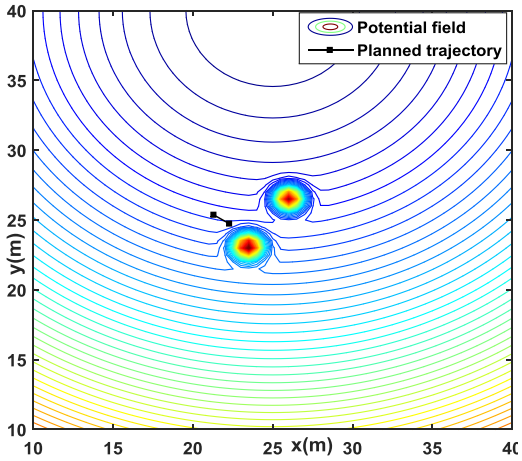
(b) Time step 21



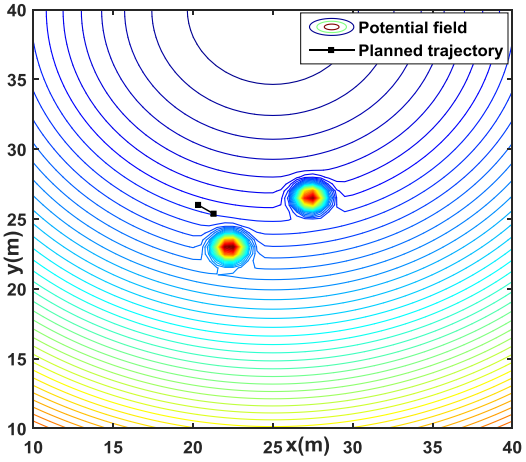
(c) Time step 22



(d) Time step 23

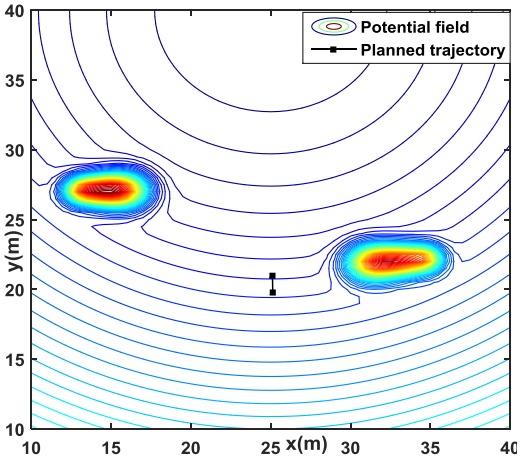


(e) Time step 24

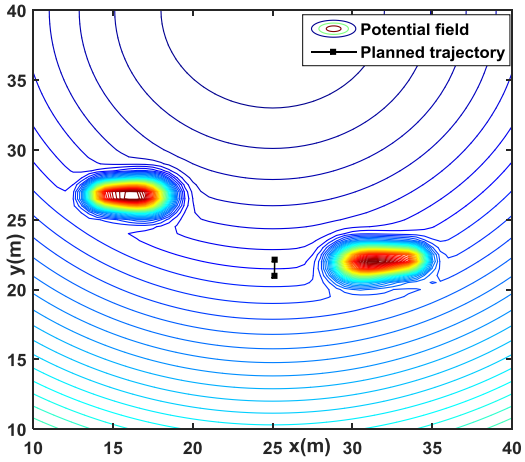


(f) Time step 25

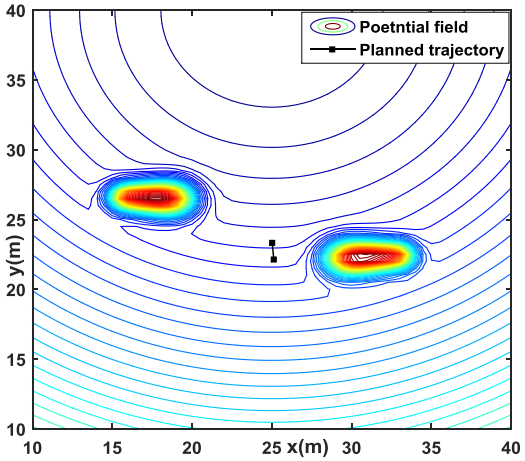
Figure 6.16 Trajectory planning with prediction horizon *N*=1 at time step 20 to 25



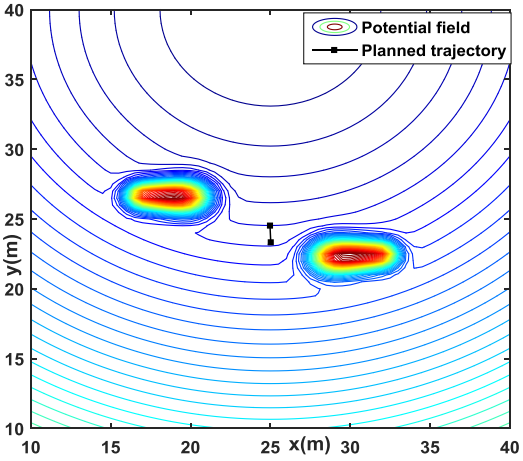
(a) Time step 16



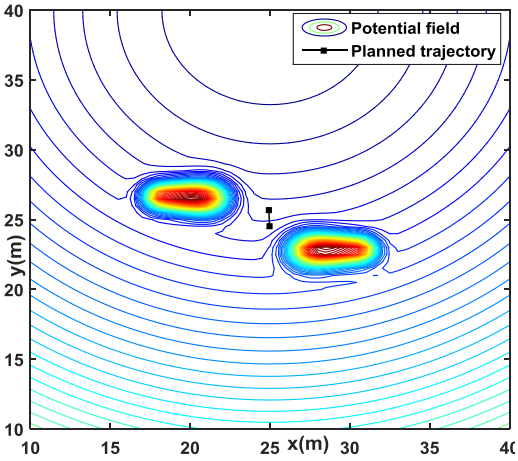
(b) Time step 17



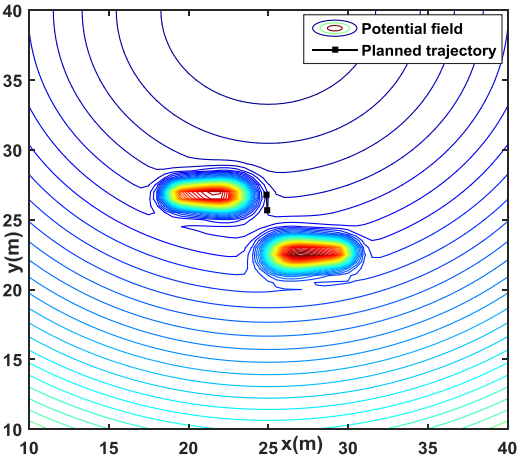
(c) Time step 18



(d) Time step 19

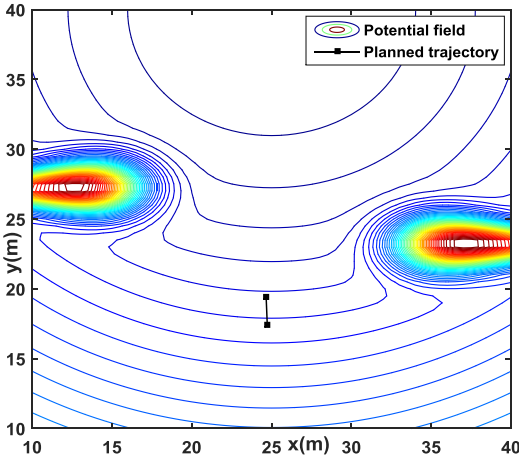


(e) Time step 20

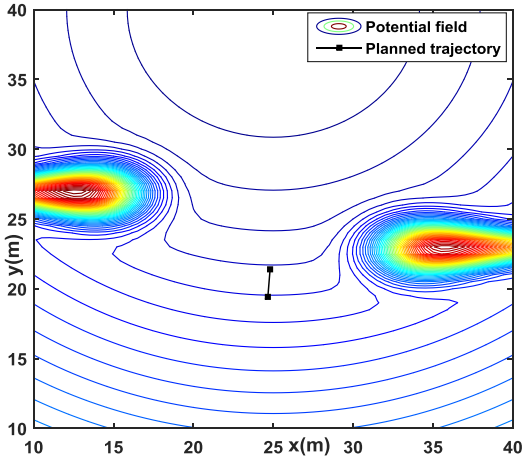


(f) Time step 21

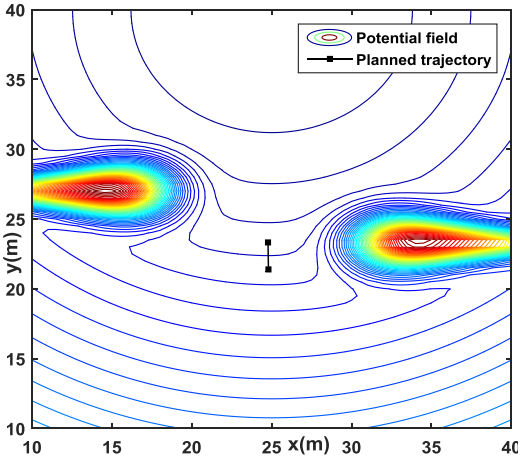
Figure 6.17 Trajectory planning with prediction horizon N=4 at time step 16 to 21



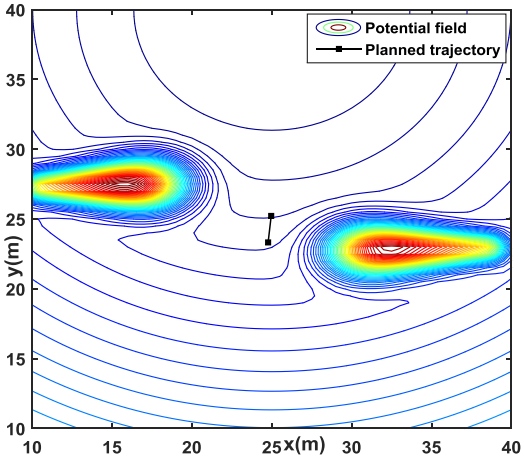
(a) Time step 13



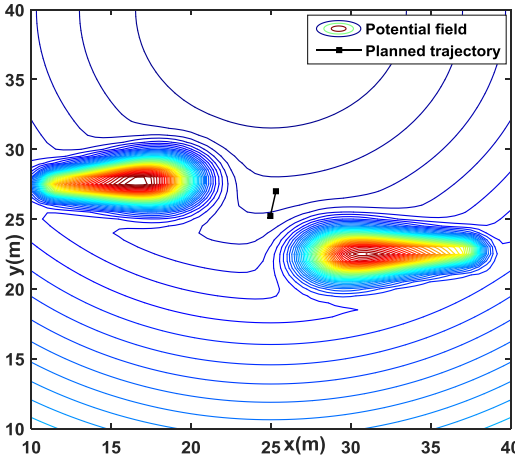
(b) Time step 14



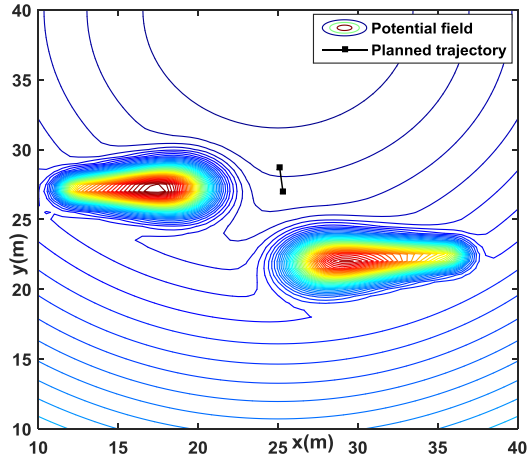
(c) Time step 15



(d) Time step 16



(e) Time step 17



(f) Time step 18

Figure 6.18 Trajectory planning with prediction horizon *N*=8 at time step 13 to 18

As shown in Figure 6.16, when no trajectory prediction (N=1) is used, the potential field represents only the current motion uncertainty of the coming vehicles. The ego vehicle becomes conscious of the coming vehicle very late at time step 21 and as a result the vehicle has to take a sudden turn to the left.

Situations are improved when the ego vehicle has a longer window of prediction ability such as N=4 and 8, it becomes conscious of the coming vehicle at a much earlier stage. Since a horizon of prediction is used, the potential fields of the moving target vehicles shown in Figures 6.17 and 18 are the mixture Gaussian of a horizon of predicted error covariance as discussed in equation (6.8). The actual position of the moving target vehicle is located close to the tail end of each of the moving target potential field. The ego vehicle actually keeps a long distance (safety range) away from the moving targets. In this case, the ego vehicle has longer prediction time and space to avoid the coming vehicles and thus the maneuver then becomes smoother with small turning angle meanwhile maintaining a relatively stable velocity.

A distance to collision criterion is also used in this study to compare the effect of using different prediction horizon length. However only the target vehicle coming from the right side is considered. The Euclidean distance between the ego vehicle and the coming vehicle is calculated. Since the ego vehicle and target vehicle have the same speed limit (13m/s) and start moving about the similar distance (about 25m) from the center of the road intersection, they are expected to meet each other in the cross road area at about t=2s during the simulation unless further collision avoidance maneuver is made by the ego vehicle.

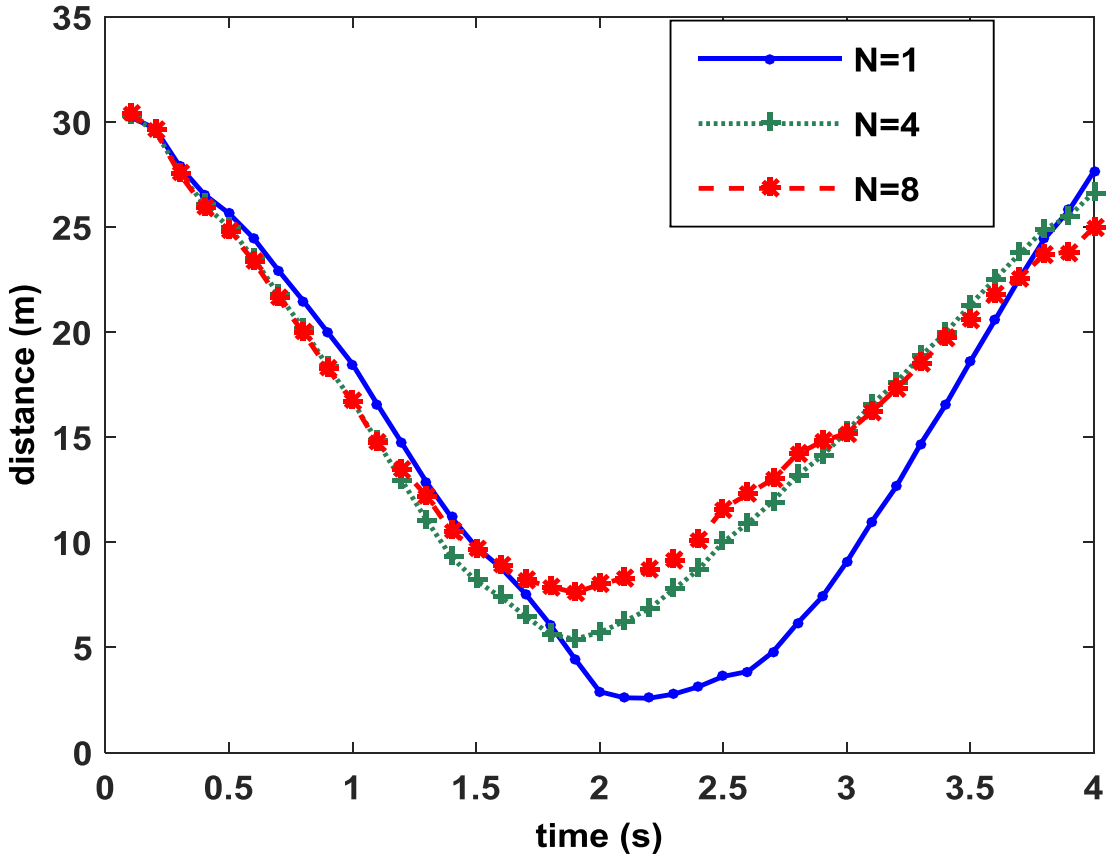


Figure 6.19 Distance between the ego and the target vehicle

As shown in Figure 6.19, without any prediction ability, the distance between two vehicles becomes quite close during the potential collision period at around t=2s, the minimum distance is 2.2 meters. The relative distance is increased when a longer prediction window is considered with the minimum distance higher than five meters. There the improvement is less significant when using a horizon length of eight instead of four. This is because in this scenario it is assumed that the target vehicle is not aware of the ego vehicle and only the ego vehicle has the ability to predict and change its maneuver for collision avoidance. The maximum maneuver rate of the ego vehicle is also limited due to the vehicle control input constraints.

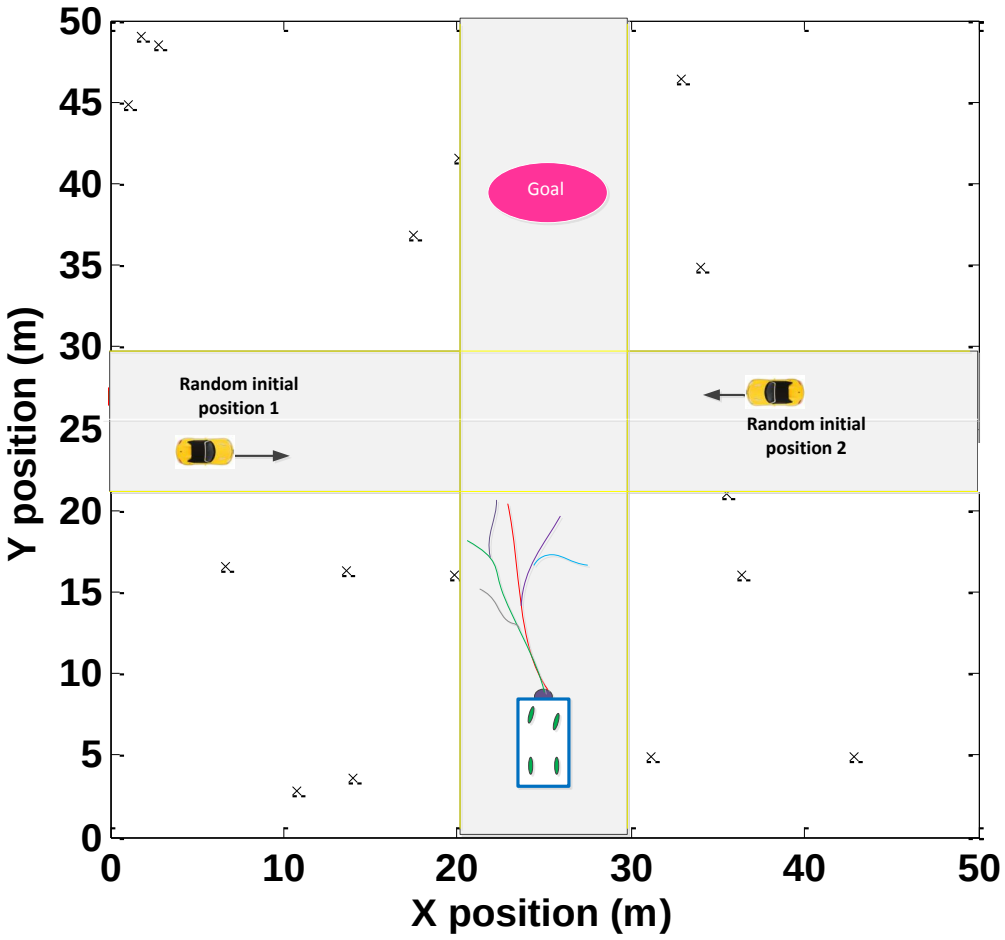


Figure 6.20 MC simulation for collision avoidance based planning

To further analyse the efficiency of the collision avoidance function, 200 trials of MonteCarlo simulations are performed for different planning scenarios shown in Figure 6.20. It is assumed that the ego vehicle has a geometric collision region of circular shape with a radius of 3 meters. Any coming vehicle moving within this area will be regarded as collision. The target vehicles are set with random initial positions within the side road region and moving towards the other side of the road following the road center line. The ego vehicle starts from the same initial position towards the goal region however different 'best trajectory' and corresponding short term goals are generated at each trial using the sampling based modified RRT algorithm. Comparison of the proposed method with different prediction horizon is shown below in Table 6.2:

Table 6.2: Comparison of collision probability using 200 Monte Carlo simulations

Horizon length	N=1	N=4	N=8
Collision rate (%)	94%	2%	0%

As shown in Table 6.2, the planner with no predication ability presents a high collision percentage while a proper prediction window can greatly enhance the safety of the planned trajectory. When a length of four step prediction window is used, the 98% planning can produce a collision free trajectory while is number is even increased to 100% by using an eight step prediction window.

In other to analysis the trajectory generation performance by the modified RRT, a more complex roundabout scenario is considered as shown in Figure 6.21, where the ego vehicle (bottom) attempts to cross the roundabout while avoiding the coming vehicles. Similar to the first scenario, the ego vehicle is equipped with on-board sensors which can scan the entire planning region of $80\text{m} \times 80\text{m}$. The measurement model noise parameters are considered in the similar fashion as the above scenario. Each target is detected with a probability of $P_d = 0.98$ and the detected measurements are immersed in clutters that can be modelled as a Poisson distribution with clutter density of β_{FA}=$7.8 * 10^{-2}$ over the $640\ m^2$ region (20 clutters return over the region of interest). Its task is to navigate from the starting position (35m, 5m) towards the second exit with the goal region located at (35m, 75m). Three target vehicles start from the initial position shown in Figure 6.21 approaching the roundabout with an initial velocity of 12m/s. The environmental constraints such as give way marks, speed limit and road boundary are incorporated in proposed MTT algorithm as discussed in Chapter 4 and 5. The roundabout is defined as a stationary obstacle cost function .The MPC prediction horizon in this case is set as *N*=5.

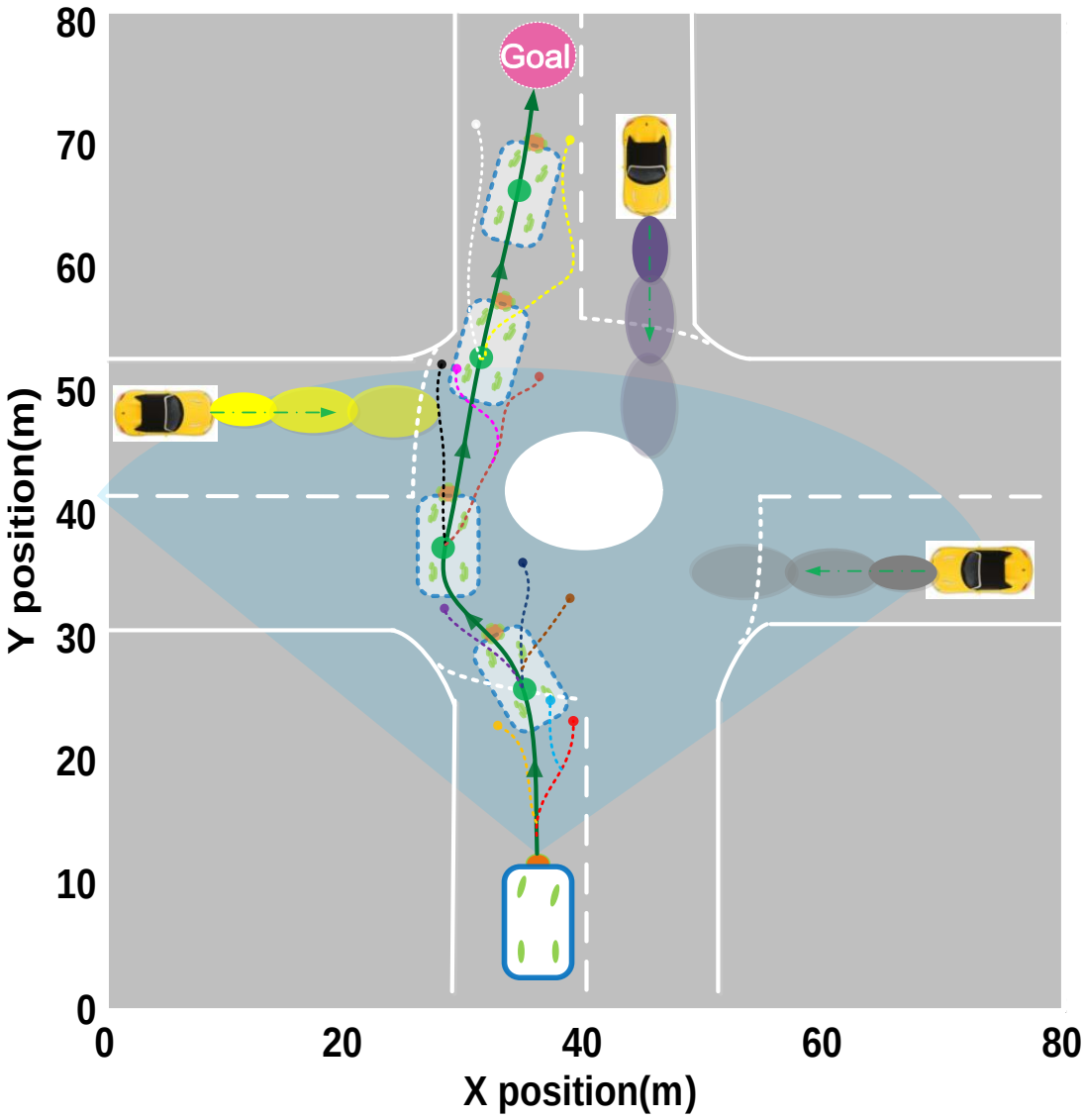


Figure 6.21 An example of the simulation result. The trajectories generated by the modified RRT are shown with different colours and the short term goals are shown as green nodes. Only the vertex node of each trajectory is presented to avoid clutter of the graph. As a result, the green trajectory is the planned trajectory with the least cost.

As shown in Figure 6.21, a feasible collision free path is successfully found by the proposed algorithm. Although MPC and its modifications have been used for obstacle avoidance technique, they are not are able to make path planning in a partially known environment without an overall planner. The modified RRT explores the environment by the short term goals (vertices) represented by green circles. In the first iteration, four possible trajectories are generated while only the best green path is reserved and executed. When the ego vehicle reached the first short term goal (green circle), a new

tree and corresponding trajectories are generated based the ego vehicle's current position for the next iteration.

In the proposed algorithm, instead of executing the whole trajectory, only a short segment of the trajectory is executed each time while the modified RRT is used to find feasible paths. This gives it a great benefit of avoiding the local minimum problem that is often found in traditional MPC path planning problem [169]. In order to compare the proposed algorithm against traditional MPC for solving local minimum in path planning, the simulation scenario shown in Figure 6.21 is carried out for 50 Monte Carlo simulations with step time T=1.

Table 6.3 Planning robustness: MPC and Proposed Algorithm

	MPC	Proposed method
Stuck in local minimum	36%	0%

As shown in Table 6.3, the proposed method is better than traditional MPC for solving local minimum in partially known dynamic environment. The proposed method has the ability to create kinematic feasible trajectory in a complex environment without getting stuck in a local minimum while the traditional MPC has a high rate of getting stuck in a local minimum during the planning.

6.6 Summary

This chapter present a path planning framework for autonomous vehicle in dynamic environments that takes into account the motion uncertainty of moving objects perceived by an on-board target tracking system. The main contribution comes from the use of uncertainty information to inform path planning. The modified RRT is proposed in this framework as a solution to find a feasible trajectory through partially known environment. Instead of executing a whole trajectory, when planned, an optimization based MPC approach is used for executing only a short segment of the planned trajectory with a limited time horizon while a new iteration of the RRT is computed. In order to deal with a dynamic environment, an on-board sensor based target tracking system is implemented to estimate the states and error covariance of moving vehicles. To achieve collision free manoeuvres, the future motions of the

moving target vehicles are predicted in a stochastic way using the KF. The predicted prior error covariance for each vehicle is used to present the motion uncertainty along the predicted trajectory. These predictions and their associated uncertainty together with other factors are incorporated in the cost function of MPC. The algorithm is demonstrated in a simulated road intersection environment with moving vehicles and different horizon steps. Our results show that the proposed method can generate a smooth and accurate trajectory taking into account domain knowledge and constraints while efficiently achieving collision avoidance with moving vehicles.

This chapter focus on experimental implementation of the proposed path planning framework and the contextual information aided target tracking algorithms which are developed in previous chapters. Potential future researches are suggested in Chapter 7 to make the whole autonomous vehicle path planning system more mature.

Chapter 7

Conclusions and Future Work

This chapter summarises the main contributions of this thesis and concludes with suggestions of possible future work.

7.1 Summary

This thesis considers the problem of improving the automation level of intelligent vehicles in both situational awareness and decision making layers. This includes two specific sub-systems: model based target tracking in the environmental perception module and motion planning in the path planning module. The focus is to utilise additional domain knowledge such as road constraints and interaction between the vehicle and its operational environment to assist both of these two systems. In summary, the overall contributions are threefold:

- Develop a rigorous Bayesian MTT approach for pooling road constraint information and sensor measurement data to provide better situational awareness of ground vehicles.
- Propose a dynamic modelling approach considering the target interaction information. The proposed modelling approach is incorporated into the proposed MTT strategy to accommodating different types of domain knowledge in a comprehensive manner so as to improve target tracking performance.

- Develop an autonomous vehicle motion planning strategy for partially known dynamic uncertain environment. The strategy aims to achieve a collision free planning in dynamic environment by making use of the estimated and predicted position and their associated uncertainty provided by the situational awareness algorithms.

7.2 Conclusions

In this thesis, to address the challenge of MTT using external domain knowledge information from the world model, a rigorous Bayesian framework is developed for pooling road constraint information and sensor measurement data to provide a better multiple ground targets state estimation.

Among various state estimation algorithms, MHE is of particular interest. This is because by applying on-line optimization in MHE, not only nonlinear dynamic systems but also additional state constraints in target tracking problems can be naturally handled. Besides, the unique moving horizon property of MHE provides a natural benefit for ground target tracking especially in cluttered environment with noisy measurements and occlusion problems. A new single target tracking strategy by using the constrained MHE approach is proposed in Chapter 3. External road information is incorporated in CMHE filters as road boundary inequality constraints in both linear and nonlinear forms. The proposed MHE algorithm is demonstrated by single target tracking scenarios consisting of both linear and nonlinear measurement models and both linear or nonlinear inequality constraints. Simulation results show that the constrained MHE can produce better estimation accuracy comparing with traditional state estimation algorithms without taking into account constraint information. Furthermore, to validate the performance of the proposed algorithm, the proposed algorithm is compared with a number of state-of-art algorithms. Results show that the constrained MHE can produce high estimation accuracy with an acceptable computational load.

In order to extend the target tracking method developed in Chapter 3 for solving more complicated multiple target issues, an improved MTT framework is developed in Chapter 4 by combining MHT with the constrained MHE namely MHE-MHT where the constrained MHE replaces the traditional KF. Comparing with traditional MHT, the new MHE-MHT framework inherits the advantages from MHE which

makes it suitable for system with nonlinear measurement and capable to systematically deal with state constraint based environmental information such as road width and speed limit in not only the state estimation but also the data association layer. The proposed MHE-MHT algorithm is demonstrated by a multiple ground vehicle tracking scenario considering road constraints with an unknown and time varying number of targets observed in cluttered environments using nonlinear measurements. The performance of the proposed algorithm is also compared with standard MHT and recently proposed GM-PHD algorithms. By using qualitative and quantitative analysis, it is clearly shown that the proposed framework significantly improves the tracking results in terms of both the state estimation and data association aspects.

Although the proposed MHE-MHT algorithm has proved its efficiency for autonomous vehicle tracking scenarios by accommodating road constraint information, it does not fully take into account the domain knowledge introduced by environmental conditions. To further improve situational awareness for autonomous vehicle, target interaction information for MTT problems is considered in Chapter 5. In a realistic ground tracking scenario, a target's movement is generally affected by its surrounding environment both stationary and moving objects, which means there are interactions between the tracked target and its surrounding environment in addition to constraint information. For example, the vehicle may be repelled away or attracted to certain objects, e.g. road centre line, in the environment and thus the surrounding environment may interact with vehicle's movement to a certain level. To address this issue, a dynamic modelling approach is proposed in Chapter 5 by considering the interaction information between a vehicle and its surrounding environment by using the 'virtual force' concept. The proposed model is then utilised in the target tracking strategy developed in Chapter 3 for an improved algorithm namely DMHE which can incorporate domain knowledge information including both environmental physical constraints and target interaction information in a comprehensive way. The DMHE is then introduced in the newly developed MHE-MHT framework namely DMHE-MHT for ground MTT with complicated environmental information. Compared with the results in Chapter 4, where limited/no interaction information is used, the proposed algorithms in this chapter shows a better tracking performance in terms of both estimation RMSE and data association accuracy.

Finally, utilising the MTT strategy developed in the previous chapters, an autonomous vehicle motion planning strategy in dynamic environments is suggested in Chapter 6. The estimated or anticipated environmental situation provided by the improved situational awareness approaches are fed to this novel decision maker. The proposed MTT technique is utilized to provide accurate on-board tracking information while taking into account the motion uncertainty of moving objects perceived by the sensor-based target tracking system. A modified Rapidly-Exploring Random trees (RRTs) is proposed in this framework as a solution to find a feasible trajectory through partially known environment. Instead of executing a whole trajectory, when planned, a Model Predictive Control (MPC) approach is used for executing only a short segment of the planned trajectory with a limited time horizon while a new iteration of the RRT is computed. In order to deal with dynamic environments, an onboard target tracking system is implemented to estimate the states and error covariances of each object of interest. The planning problem is then implemented by optimizing a trajectory cost function consisting of safety imposed by both stationary environment and moving objects and other factors. To achieve collision free manoeuvres, the future motions of the other vehicles are predicted in a stochastic way using the KF. The predicted prior error covariance for each vehicle is used to capture the motion uncertainty along the predicted trajectory and incorporated in the MPC optimization function. The algorithm is successfully demonstrated in a simulated road intersection environment with moving vehicles.

7.3 Future Work

This thesis introduced new methods to solve situational awareness and decision making problems for autonomous vehicles. Most of the work was evaluated via simulations, which leaves much scope for development in real world applications with real measurement data. As an outcome from this thesis, a number of future research challenges have been identified as follows. This section introduces these challenges along with some technical suggestions.

(1) The approach to explore an area with the constrained MHE for target tracking leaves some developments for future work. It is common in MHE implementations to assume that the a priori pdf representing the arrival cost is a multivariate Gaussian distribution. For linear unconstrained systems, the

standard KF covariance update formula can be used to express the arrival cost explicitly. However, for nonlinear or constrained systems, the initialization of the MHE with the best choice of arrival cost term is still an open issue, which also leaves the computational complexity of MHE implementation as an open challenge. The existing methods of recursively updating the arrival cost based on functional approximations of nonlinearities by truncated Taylor series can lead to unpredictable behaviour e.g. the covariance used to weight the arrival cost may diverge and thus fail to be a reliable measure of the quality of the knowledge of the state. This gives the motivation to investigate methods such as sampling based nonlinear filters [167] to properly parameterize the arrival cost.

(2) The novel approach of MHE-MHT target tracking structure also leaves room for future work. Although the algorithm successfully tracks multiple vehicles in a challenging environment, the essential structural complexity of the MHT algorithm makes the proposed approach suffer from the additional computational effort. Possible future work would include developing a detection and track management system which can reduce the total size of measurements by using multiclustering methods. In this case, a parallel process can be implemented for target detection before the data association process is done. Other possible approaches of improving the computational efficiency would include investigating MHE based state estimation into other MTT structures such as PHD and JPDA structures. Further work would be required for performance comparison. The introduction of multi-core processors could be used to boost processing speed by utilizing the concepts of parallel processing. Facilitating real-time implementation of the proposed MTT algorithms by modifying the algorithms for parallel processing can be a future research direction.

(3) Another area of interest is how to simplify and estimate the parameters used in the DMHE-MHT algorithm. Since the proposed domain knowledge aided dynamic model could incorporate a number of interaction forces which are defined by functions with individual parameter properties. The overall number of parameters considered in DMHE-MHT makes it difficult to be used in real world applications. To solve this problem, some optimization algorithms could be used to determine the optimal parameter specifications for the force model.

Such optimization method could be accommodated in the MHE optimization function for direct parameter estimation iteratively. The MHE cost function needs to be modified for incorporating both the state and parameter vectors. Heuristic global optimal search techniques such like differential evolution (DE) [168] could be applied to solve the complex optimization problems.

(4) For path planning layer, more detailed dynamic/kinematic models for autonomous vehicle could be considered for more accurate trajectory planning. Such as the vehicle model described by the derivative of lateral and longitudinal movement [148]. The characterization of different types of moving obstacles could also be considered for real world vehicle motion planning experiments such as the IMM approach. Besides, the prediction of obstacles future position is directly relevant to obstacles behaviour. If the motion of vehicles is known stochastically, the overall probability of collision for a potential trajectory may also be computed based on the expected behavior of other obstacles. The prediction information could then be used in MPC using a receding horizon structure for a better future motion planning.

Reference

[1] Gasser, T. M., & Westhoff, D. (2012, July). BASt-study: Definitions of automation and legal issues in Germany. In *Presentation at the Road Vehicle Automation Workshop at the*.

[2] Hebert, M. H., Thorpe, C. E., & Stentz, A. (Eds.). (2012). *Intelligent unmanned ground vehicles: autonomous navigation research at Carnegie Mellon* (Vol. 388). Springer Science & Business Media.

[3] Foyle, D. C., Andre, A. D., & Hooey, B. L. (2005). Situation awareness in an augmented reality cockpit: Design, viewpoints and cognitive glue. In*Proceedings of the 11th International Conference on Human Computer Interaction* (pp. 3-9).

[4] Endsley, M. R. (1988, October). Design and evaluation for situation awareness enhancement. In *Proceedings of the Human Factors and Ergonomics Society Annual Meeting* (Vol. 32, No. 2, pp. 97-101). SAGE Publications.

[5] Hone, G., Martin, L., & Ayres, R. (2006). Awareness–does the acronym "SA" still have any value. In *Proceedings of the 11th International Command and Control Research and Technology Symposium: Coalition Command and Control in the Networked Era. held at Cambridge UK. Washington*.

[6] Darms, M., Baker, C. R., Rybski, P. E., & Urmson, C. (2008). Vehicle detection and tracking for the urban challenge.

[7] Ferguson, D., Darms, M., Urmson, C., & Kolski, S. (2008, June). Detection, prediction, and avoidance of dynamic obstacles in urban environments. In*Intelligent Vehicles Symposium, 2008 IEEE* (pp. 1149-1154). IEEE.

[8] Kalman, R. E. (1960). A new approach to linear filtering and prediction problems. *Journal of basic Engineering*, *82*(1), 35-45.

[9] Schmidt, S. F. (1981). The Kalman filter-Its recognition and development for aerospace applications. *Journal of Guidance, Control, and Dynamics*, *4*(1), 4-7.

[10] Kocak, D. M., da Vitoria Lobo, N., & Widder, E. A. (1999). Computer vision techniques for quantifying, tracking, and identifying bioluminescent plankton.*IEEE Journal of Oceanic Engineering*, *24*(1), 81-95.

[11] Blackman, S., & Popoli, R. (1999). Design and analysis of modern tracking systems(Book). *Norwood, MA: Artech House, 1999.*

[12] Thrun, S., Fox, D., Burgard, W., & Dellaert, F. (2001). Robust Monte Carlo localization for mobile robots. *Artificial intelligence*, *128*(1), 99-141.

[13] Petrovskaya, A., & Thrun, S. (2009). Model based vehicle detection and tracking for autonomous urban driving. *Autonomous Robots*, *26*(2-3), 123-139.

[14] Stiller, C., León, F. P., & Kruse, M. (2011). Information fusion for automotive applications–An overview. *Information Fusion*, *12*(4), 244-252.

[15] Pulford, G. W. (2005). Taxonomy of multiple target tracking methods. *IEE Proceedings-Radar, Sonar and Navigation*, *152*(5), 291-304.

[16] Bishop, R. (2000). Intelligent vehicle applications worldwide. *IEEE Intelligent Systems and Their Applications*, *15*(1), 78-81.

[17] Berndt, H., Wender, S., & Dietmayer, K. (2007, June). Driver braking behavior during intersection approaches and implications for warning strategies for driver assistant systems. In *2007 IEEE Intelligent Vehicles Symposium* (pp. 245-251). IEEE.

[18] den Camp, O. O., Ranjbar, A., Uittenbogaard, J., Rosen, E., & Buijssen, S. H. H. M. (2014). Overview of main accident scenarios in car-to-cyclist accidents for use in AEB-system test protocol. In *Proceedings of International Cycling Safety Conference*.

[19] Coelingh, E., Eidehall, A., & Bengtsson, M. (2010, September). Collision warning with full auto brake and pedestrian detection-a practical example of automatic emergency braking. In *Intelligent Transportation Systems (ITSC), 2010 13th International IEEE Conference on* (pp. 155-160). IEEE.

[20] Udugu, K., Saddala, V. R., & Lingan, S. (2016). *Active and Passive Safety: An Overview on Establishing Safety Assessment Standards in India* (No. 2016-28-0244). SAE Technical Paper.

[21] Designateddrivertk. (2016). *Designated Driver*. From http://designateddriver.tk/advanced-driver-assistance-system/, Retrieved 22 June, 2016

[22] Colin Jeffrey, (Oct7 2015). From http://www.gizmag.com/peugeot-citroen-autonomous-car-paris-bordeaux/39738/, Retrieved 22 June, 2016

[23] Coué, C., Fraichard, T., Bessiere, P., & Mazer, E. (2003, September). Using Bayesian programming for multi-sensor multi-target tracking in automotive applications. In *Robotics and Automation, 2003. Proceedings. ICRA'03. IEEE International Conference on* (Vol. 2, pp. 2104-2109). IEEE.

[24] Folster, F., & Rohling, H. (2006, October). Lateral velocity estimation based on automotive radar sensors. In *2006 CIE International Conference on Radar*(pp. 1-4). IEEE.

[25] Hebert, T., Valavanis, K., & Kolluru, R. (1998). A real-time, hierarchical, sensor-based robotic system architecture. *Journal of Intelligent and Robotic Systems*, *21*(1), 1-27.

[26] Hall, D. L., & Llinas, J. (1997). An introduction to multisensor data fusion.*Proceedings of the IEEE*, *85*(1), 6-23.

[27] Lu, M., Wevers, K., & Van Der Heijden, R. (2005). Technical feasibility of advanced driver assistance systems (ADAS) for road traffic safety.*Transportation Planning and Technology*, *28*(3), 167-187.

[28] Bernais, B., Lotz, A., & Pu, H. (2016, March). Design and implementation of a traffic light assistance system based on C2X and TSI messages. In *AmE 2016-Automotive meets Electronics; 7th GMM-Symposium* (pp. 1-6). VDE.

[29] Veres, S. M., Molnar, L., Lincoln, N. K., & Morice, C. P. (2011). Autonomous vehicle control systems—a review of decision making. *Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering*, *225*(2), 155-195.

[30] Li, X. R., & Jilkov, V. P. (2003). Survey of maneuvering target tracking. Part I. Dynamic models. *IEEE Transactions on aerospace and electronic systems*, *39*(4), 1333-1364.

[31] Gordon, N. J., Salmond, D. J., & Smith, A. F. (1993, April). Novel approach to nonlinear/non-Gaussian Bayesian state estimation. In *IEE Proceedings F-Radar and Signal Processing* (Vol. 140, No. 2, pp. 107-113). IET.

[32] Wan, E. A., & Van Der Merwe, R. (2000). The unscented Kalman filter for nonlinear estimation. In *Adaptive Systems for Signal Processing, Communications, and Control Symposium 2000. AS-SPCC. The IEEE 2000*(pp. 153-158). Ieee.

[33] Arulampalam, M. S., Maskell, S., Gordon, N., & Clapp, T. (2002). A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking. *IEEE Transactions on signal processing*, *50*(2), 174-188.

[34] Bar-Shalom, Y., & Li, X. R. (1995). Multitarget-multisensor tracking: principles and techniques. *Storrs, CT: University of Connecticut, 1995.*

[35] Julier, S. J., & Uhlmann, J. K. (2004). Unscented filtering and nonlinear estimation. *Proceedings of the IEEE*, *92*(3), 401-422.

[36] Rao, C. V., Rawlings, J. B., & Lee, J. H. (2001). Constrained linear state estimation—a moving horizon approach. *Automatica*, *37*(10), 1619-1628.

[37] Rao, C. V., Rawlings, J. B., & Mayne, D. Q. (2003). Constrained state estimation for nonlinear discrete-time systems: Stability and moving horizon approximations. *IEEE transactions on automatic control*, *48*(2), 246-258.

[38] Rao, C. V. (2000). *Moving horizon strategies for the constrained monitoring and control of nonlinear discrete-time systems* (Doctoral dissertation, UNIVERSITY OF WISCONSIN {MADISON).

[39] Allgöwer, F., Badgwell, T. A., Qin, J. S., Rawlings, J. B., & Wright, S. J. (1999). Nonlinear predictive control and moving horizon estimation—an introductory overview. In *Advances in control* (pp. 391-449). Springer London.

[40] Russo, L. P., & Young, R. E. (1999, June). Moving-horizon state estimation applied to an industrial polymerization process. In *American Control Conference, 1999. Proceedings of the 1999* (Vol. 2, pp. 1129-1133). IEEE.

[41] Bemporad, A., Mignone, D., & Morari, M. (1999). Moving horizon estimation for hybrid systems and fault detection. In *American Control Conference* (Vol. 4, pp. 2471-2475). IEEE.

[42] Tyler, M. L., Asano, K., & Morari, M. (2000). Application of moving horizon estimation based fault detection to cold tandem steel mill. *International Journal of Control*, *73*(5), 427-438.

[43] Dong, J., & Verhaegen, M. (2009). Subspace based fault detection and identification for LTI systems. *IFAC Proceedings Volumes*, *42*(8), 330-335.

[44] Thomas, Y. A. (1975). Linear quadratic optimal estimation and control with receding horizon. *Electronics Letters*, *1*(11), 19-21.

[45] Ferrari-Trecate, G., Mignone, D., & Morari, M. (2002). Moving horizon estimation for hybrid systems. *IEEE Transactions on Automatic Control*,*47*(10), 1663-1676.

[46] De Maesschalck, R., Jouan-Rimbaud, D., & Massart, D. L. (2000). The mahalanobis distance. *Chemometrics and intelligent laboratory systems*,*50*(1), 1-18.

[47] Bar-Shalom, Y. (1978). Tracking methods in a multitarget environment. *IEEE Transactions on automatic control*, *23*(4), 618-626.

[48] Blackman, S. S. (2004). Multiple hypothesis tracking for multiple target tracking. *IEEE Aerospace and Electronic Systems Magazine*, *19*(1), 5-18.

[49] Bar-Shalom, Y., Daum, F., & Huang, J. (2009). The probabilistic data association filter. *IEEE Control Systems*, *29*(6), 82-100.

[50] Frank, O., Nieto, J., Guivant, J., & Scheding, S. (2003, October). Multiple target tracking using sequential Monte Carlo methods and statistical data association. In *Intelligent Robots and Systems, 2003.(IROS 2003). Proceedings. 2003 IEEE/RSJ International Conference on* (Vol. 3, pp. 2718-2723). IEEE.

[51] Fortmann, T., Bar-Shalom, Y., & Scheffe, M. (1983). Sonar tracking of multiple targets using joint probabilistic data association. *IEEE journal of Oceanic Engineering*, *8*(3), 173-184.

[52] Fisher, J. L., & Casasent, D. P. (1989). Fast JPDA multitarget tracking algorithm. *Applied optics*, *28*(2), 371-376.

[53] Zhou, B., & Bose, N. K. (1993). Multitarget tracking in clutter: fast algorithms for data association. *IEEE Transactions on aerospace and electronic systems*, *29*(2), 352-363.

[54] Kuhn, H. W. (1955). The Hungarian method for the assignment problem.*Naval research logistics quarterly*, *2*(1-2), 83-97.

[55] Munkres, J. (1957). Algorithms for the assignment and transportation problems. *Journal of the society for industrial and applied mathematics*, *5*(1), 32-38.

[56] Malkoff, D. B. (1997, July). Evaluation of the Jonker-Volgenant-Castanon (JVC) assignment algorithm for track association. In *AeroSense'97* (pp. 228-239). International Society for Optics and Photonics.

[57] Bertsekas, D. P. (1988). The auction algorithm: A distributed relaxation method for the assignment problem. *Annals of operations research*, *14*(1), 105-123.

[58] Cox, I. J., & Hingorani, S. L. (1996). An efficient implementation of Reid's multiple hypothesis tracking algorithm and its evaluation for the purpose of visual tracking. *IEEE Transactions on pattern analysis and machine intelligence*, *18*(2), 138-150.

[59] Streit, R. L., & Luginbuhl, T. E. (1995). *Probabilistic multi-hypothesis tracking* (No. NUWC-NPT-TR-10428). NAVAL UNDERWATER SYSTEMS CENTER NEWPORT RI.

[60] Dempster, A. P., Laird, N. M., & Rubin, D. B. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the royal statistical society. Series B (methodological)*, 1-38.

[61] Efe, M., Ruan, Y., & Willett, P. (2002, July). The pedestrian PMHT. In*Information Fusion, 2002. Proceedings of the Fifth International Conference on* (Vol. 2, pp. 838-845). IEEE.

[62] Hue, C., Le Cadre, J. P., & Pérez, P. (2002). Tracking multiple objects with particle filtering. *IEEE Transactions on Aerospace and Electronic Systems*,*38*(3), 791-812.

[63] Hue, C., Le Cadre, J. P., & Perez, P. (2002). Sequential Monte Carlo methods for multiple target tracking and data fusion. *IEEE Transactions on signal processing*, *50*(2), 309-325.

[64] Stiller, C., León, F. P., & Kruse, M. (2011). Information fusion for automotive applications–An overview. *Information Fusion*, *12*(4), 244-252.

[65] Mahler, R. P. (2007). *Statistical multisource-multitarget information fusion*. Artech House, Inc..

[66] Vo, B. N., & Ma, W. K. (2006). The Gaussian mixture probability hypothesis density filter. *IEEE Transactions on signal processing*, *54*(11), 4091-4104.

[67] Vo, B. T., Vo, B. N., & Cantoni, A. (2009). The cardinality balanced multi-target multi-Bernoulli filter and its implementations. *IEEE Transactions on Signal Processing*, *57*(2), 409-423.

[68] Ulmke, M., Erdinc, O., & Willett, P. (2007, July). Gaussian mixture cardinalized PHD filter for ground moving target tracking. In *Information Fusion, 2007 10th International Conference on* (pp. 1-8). IEEE.

[69] Choset, H. M. (2005). *Principles of robot motion: theory, algorithms, and implementation*. MIT press.

[70] De Berg, M., Van Kreveld, M., Overmars, M., & Schwarzkopf, O. C. (2000). Computational geometry. In *Computational geometry* (pp. 1-17). Springer Berlin Heidelberg.

[71] Latombe, J. C. (2012). *Robot motion planning* (Vol. 124). Springer Science & Business Media.

[72] LaValle, S. M. (2006). *Planning algorithms*. Cambridge university press.

[73] Raja, P., & Pugazhenthi, S. (2012). Optimal path planning of mobile robots: A review. *International Journal of Physical Sciences*, *7*(9), 1314-1320.

[74] Zhang, S., Deng, W., Zhao, Q., Sun, H., & Litkouhi, B. (2013, October). Dynamic trajectory planning for vehicle autonomous driving. In *2013 IEEE International Conference on Systems, Man, and Cybernetics* (pp. 4161-4166). IEEE.

[75] Hoy, M., Matveev, A. S., & Savkin, A. V. (2015). Algorithms for collision-free navigation of mobile robots in complex cluttered environments: a survey.*Robotica*, *33*(03), 463-497.

[76] Dadkhah, N., & Mettler, B. (2012). Survey of motion planning literature in the presence of uncertainty: considerations for UAV guidance. *Journal of Intelligent & Robotic Systems*, *65*(1-4), 233-246.

[77] LaValle, S. M., Branicky, M. S., & Lindemann, S. R. (2004). On the relationship between classical grid search and probabilistic roadmaps. *The International Journal of Robotics Research*, *23*(7-8), 673-692.

[78] Barraquand, J., & Latombe, J. C. (1991). Robot motion planning: A distributed representation approach. *The International Journal of Robotics Research*, *10*(6), 628-649.

[79] Bessiere, P., Ahuactzin, J. M., Talbi, E. G., & Mazer, E. (1993, July). The "Ariadne's clew" algorithm: global planning with local methods. In *Intelligent Robots and Systems' 93, IROS'93. Proceedings of the 1993 IEEE/RSJ International Conference on* (Vol. 2, pp. 1373-1380). IEEE.

[80] Kavraki, L. E., Svestka, P., Latombe, J. C., & Overmars, M. H. (1996). Probabilistic roadmaps for path planning in high-dimensional configuration spaces. *IEEE transactions on Robotics and Automation*, *12*(4), 566-580.

[81] Kuffner, J. J., & LaValle, S. M. (2000). RRT-connect: An efficient approach to single-query path planning. In *Robotics and Automation, 2000. Proceedings. ICRA'00. IEEE International Conference on* (Vol. 2, pp. 995-1001). IEEE.

[82] en Muestreo, P. D. M. B., & Compendio, U. (2008). Sampling-based motion planning: a survey. *Computación y Sistemas*, *12*(1), 5-24.

[83] LaValle, S. M., & Kuffner Jr, J. J. (2000). Rapidly-exploring random trees: Progress and prospects.

[84] Khatib, O. (1986). Real-time obstacle avoidance for manipulators and mobile robots. *The international journal of robotics research*, *5*(1), 90-98.

[85] Simmons, R. (1996, April). The curvature-velocity method for local obstacle avoidance. In *Robotics and Automation, 1996. Proceedings., 1996 IEEE International Conference on* (Vol. 4, pp. 3375-3382). IEEE.

[86] Mayne, D. Q., & Raković, S. (2003). Model predictive control of constrained piecewise affine discrete-time systems. *International Journal of Robust and Nonlinear Control*, *13*(3-4), 261-279.

[87] Richards, A., & How, J. P. (2007). Robust distributed model predictive control. *International Journal of control*, *80*(9), 1517-1531.

[88] Sathyaraj, B. M., Jain, L. C., Finn, A., & Drake, S. (2008). Multiple UAVs path planning algorithms: a comparative study. *Fuzzy Optimization and Decision Making*, *7*(3), 257-267.

[89] Koenig, S., & Likhachev, M. (2005). Fast replanning for navigation in unknown terrain. *IEEE Transactions on Robotics*, *21*(3), 354-363.

[90] Garrido, S., Moreno, L., Abderrahim, M., & Martin, F. (2006, October). Path planning for mobile robot navigation using voronoi diagram and fast marching. In *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems* (pp. 2376-2381). IEEE.

[91] Škrjanc, I., & Klančar, G. (2010). Optimal cooperative collision avoidance between multiple robots based on Bernstein–Bézier curves. *Robotics and Autonomous systems*, *58*(1), 1-9.

[92] Lau, B., Sprunk, C., & Burgard, W. (2009, October). Kinodynamic motion planning for mobile robots using splines. In *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems* (pp. 2427-2433). IEEE.

[93] Qu, Z., Wang, J., & Plaisted, C. E. (2004). A new analytical solution to mobile robot trajectory generation in the presence of moving obstacles. *IEEE Transactions on Robotics*, *20*(6), 978-993.

[94] Ge, S. S., & Cui, Y. J. (2000). New potential functions for mobile robot path planning. *IEEE Transactions on robotics and automation*, *16*(5), 615-620.

[95] Wang, Y., & Chirikjian, G. S. (2000). A new potential field method for robot path planning. In *Robotics and Automation, 2000. Proceedings. ICRA'00. IEEE International Conference on* (Vol. 2, pp. 977-982). IEEE.

[96] Abichandani, P., Ford, G., Benson, H. Y., & Kam, M. (2012, May). Mathematical programming for multi-vehicle motion planning problems. In*Robotics and Automation (ICRA), 2012 IEEE International Conference on*(pp. 3315-3322). IEEE.

[97] Besada-Portas, E., de la Torre, L., Jesus, M., & de Andrés-Toro, B. (2010). Evolutionary trajectory planner for multiple UAVs in realistic scenarios. *IEEE Transactions on Robotics*, *26*(4), 619-634.

[98] Zheng, C., Li, L., Xu, F., Sun, F., & Ding, M. (2005). Evolutionary route planner for unmanned air vehicles. *IEEE Transactions on Robotics*, *21*(4), 609-620.

[99] Kurniawati, H., Du, Y., Hsu, D., & Lee, W. S. (2010). Motion planning under uncertainty for robotic tasks with long time horizons. *The International Journal of Robotics Research*, 0278364910386986.

[100] Brooks, A., Kaupp, T., & Makarenko, A. (2009, May). Randomised MPC-based motion-planning for mobile robot obstacle avoidance. In *Robotics and Automation, 2009. ICRA'09. IEEE International Conference on* (pp. 3962-3967). IEEE.

[101] Krishnamurthy, P., & Khorrami, F. (2005, June). GODZILA: A low-resource algorithm for path planning in unknown environments. In *Proceedings of the American Control Conference* (Vol. 1, p. 110).

[102] Yang, K., Gan, S. K., & Sukkarieh, S. (2010). An efficient path planning and control algorithm for RUAV's in unknown and cluttered environments. *Journal of Intelligent and Robotic Systems*, *57*(1-4), 101-122.

[103] Yu, H., Sharma, R., Beard, R. W., & Taylor, C. N. (2011, June). Observability-based local path planning and collision avoidance for micro air vehicles using bearing-only measurements. In *Proceedings of the 2011 American Control Conference* (pp. 4649-4654). IEEE.

[104] Gordon, N., Ristic, B., & Arulampalam, S. (2004). Beyond the kalman filter: Particle filters for tracking applications. *Artech House, London*.

[105] Ulmke, M., & Koch, W. (2006). Road-map assisted ground moving target tracking. *IEEE Transactions on Aerospace and Electronic Systems*, *42*(4), 1264-1274.

[106] Simon, D., & Chia, T. L. (2002). Kalman filtering with state equality constraints. *IEEE transactions on Aerospace and Electronic Systems*, *38*(1), 128-136.

[107] Simon, D. (2010). Kalman filtering with state constraints: a survey of linear and nonlinear algorithms. *IET Control Theory & Applications*, *4*(8), 1303-1318.

[108] Gill, P. E., Murray, W., & Wright, M. H. (1981). Practical optimization.

[109] Yang, C., & Blasch, E. (2009). Kalman filtering with nonlinear state constraints. *IEEE Transactions on Aerospace and Electronic Systems*,*45*(1), 70-84.

[110] Kolås, S., Foss, B. A., & Schei, T. S. (2009). Constrained nonlinear state estimation based on the UKF approach. *Computers & Chemical Engineering*,*33*(8), 1386-1401.

[111] Bell, B. M., Burke, J. V., & Pillonetto, G. (2009). An inequality constrained nonlinear Kalman–Bucy smoother by interior point likelihood maximization.*Automatica*, *45*(1), 25-33.

[112] Straka, O., Duník, J., & Šimandl, M. (2012). Truncation nonlinear filters for state estimation with nonlinear inequality constraints. *Automatica*, *48*(2), 273-286.

[113] Shao, X., Huang, B., & Lee, J. M. (2010). Constrained Bayesian state estimation–A comparative study and a new particle filter based approach.*Journal of Process Control*, *20*(2), 143-157.

[114] Rao, C. V., Rawlings, J. B., & Mayne, D. Q. (2003). Constrained state estimation for nonlinear discrete-time systems: Stability and moving horizon approximations. *IEEE transactions on automatic control*, *48*(2), 246-258.

[115] Yang, C., Bakich, M., & Blasch, E. (2005, July). Nonlinear constrained tracking of targets on roads. In *2005 7th International Conference on Information Fusion* (Vol. 1, pp. 8-pp). IEEE.

[116] Rao, C. V., Rawlings, J. B., & Lee, J. H. (2001). Constrained linear state estimation—a moving horizon approach. *Automatica*, *37*(10), 1619-1628.

[117] Ding, R., Yu, M., & Chen, W. H. (2015). A Multiple Target Tracking Strategy Using Moving Horizon Estimation Approach. In *24th International Technical Conference on the Enhanced Safety of Vehicles (ESV)* (No. 15-0390).

[118] Allgöwer, F., Badgwell, T. A., Qin, J. S., Rawlings, J. B., & Wright, S. J. (1999). Nonlinear predictive control and moving horizon estimation—an introductory overview. In *Advances in control* (pp. 391-449). Springer London.

[119] Arbuckle, J. L. (1996). Full information estimation in the presence of incomplete data. *Advanced structural equation modeling: Issues and techniques*, *243*, 277.

[120] Julier, S. J., & LaViola, J. J. (2007). On Kalman filtering with nonlinear equality constraints. *IEEE Transactions on Signal Processing*, *55*(6), 2774-2784.

[121] NøRgaard, M., Poulsen, N. K., & Ravn, O. (2000). New developments in state estimation for nonlinear systems. *Automatica*, *36*(11), 1627-1638.

[122] Lopez-Negrete, R., Patwardhan, S. C., & Biegler, L. T. (2011). Constrained particle filter approach to approximate the arrival cost in moving horizon estimation. *Journal of Process Control*, *21*(6), 909-919.

[123] Qu, C. C., & Hahn, J. (2009). Computation of arrival cost for moving horizon estimation via unscented Kalman filtering. *Journal of Process Control*, *19*(2), 358-363.

[124] Cox, I. J. (1991). Blanche-an experiment in guidance and navigation of an autonomous robot vehicle. *IEEE Transactions on robotics and automation*,*7*(2), 193-204.

[125] Leonard, J. J., & Durrant-Whyte, H. F. (1991). Mobile robot localization by tracking geometric beacons. *IEEE Transactions on robotics and Automation*,*7*(3), 376-382.

[126] Fox, D., Burgard, W., Dellaert, F., & Thrun, S. (1999). Monte carlo localization: Efficient position estimation for mobile robots. *AAAI/IAAI*, *1999*, 343-349.

[127] Liu, C., Li, B., & Chen, W. H. (2013). Road network based vehicle navigation using an improved IMM particle filter. *IFAC Proceedings Volumes*, *46*(10), 193-198.

[128] Panta, K., Clark, D. E., & Vo, B. N. (2009). Data association and track management for the Gaussian mixture probability hypothesis density filter.*IEEE Transactions on Aerospace and Electronic Systems*, *45*(3), 1003-1016.

[129] Maher, R. (2007, April). A survey of PHD filter and CPHD filter implementations. In *Defense and Security Symposium* (pp. 65670O-65670O). International Society for Optics and Photonics.

[130] Schuhmacher, D., Vo, B. T., & Vo, B. N. (2008). A consistent metric for performance evaluation of multi-object filters. *IEEE Transactions on Signal Processing*, *56*(8), 3447-3457.

[131] Svensson, D., Wintenby, J., & Svensson, L. (2009, July). Performance evaluation of MHT and GM-CPHD in a ground target tracking scenario. In*Information Fusion, 2009. FUSION'09. 12th International Conference on* (pp. 300-307). IEEE.

[132] Lamard, L., Chapuis, R., & Boyer, J. P. (2012, June). A comparison of two different tracking algorithms is provided for real application. In *Intelligent Vehicles Symposium (IV), 2012 IEEE* (pp. 414-419). IEEE.

[133] Arulampalam, M. S., Gordon, N., Orton, M., & Ristic, B. (2002, July). A variable structure multiple model particle filter for GMTI tracking. In*Information Fusion, 2002. Proceedings of the Fifth International Conference on* (Vol. 2, pp. 927-934). IEEE.

[134] Cheng, Y., & Singh, T. (2007). Efficient particle filtering for road-constrained target tracking. *IEEE Transactions on Aerospace and Electronic Systems*,*43*(4), 1454-1469.

[135] Oh, H., Shin, H. S., Kim, S., Tsourdos, A., & White, B. A. (2013). Airborne behaviour monitoring using Gaussian processes with map information. *IET Radar, Sonar & Navigation*, *7*(4), 393-400.

[136] Yu, M., Liu, C., Chen, W. H., & Chambers, J. (2014, June). A Bayesian framework with an auxiliary particle filter for GMTI-based ground vehicle tracking aided by domain knowledge. In *SPIE Defense+ Security* (pp. 90911I-90911I). International Society for Optics and Photonics.

[137] Helbing, D., & Molnar, P. (1995). Social force model for pedestrian dynamics.*Physical review E*, *51*(5), 4282.

[138] Bang, G., & Kweon, I. S. (2013). Multi-target tracking using social force model in discrete-continuous optimisation framework. *Electronics Letters*,*49*(21), 1331-1333.

[139] Bellomo, N., & Dogbe, C. (2011). On the modeling of traffic and crowds: A survey of models, speculations, and perspectives. *SIAM review*, *53*(3), 409-463.

[140] Kumar, P., Merzouki, R., Conrard, B., Coelen, V., & Bouamama, B. O. (2014). Multilevel modeling of the traffic dynamic. *IEEE Transactions on Intelligent Transportation Systems*, *15*(3), 1066-1082.

[141] Nakayama, A., Sugiyama, Y., & Hasebe, K. (2001). Effect of looking at the car that follows in an optimal velocity model of traffic flow. *Physical Review E*, *65*(1), 016112.

[142] Lecchini, A., Glover, W., Lygeros, J., & Maciejowski, J. (2005, March). Air-Traffic Control in Approach Sectors: simulation examples and optimisation. In *International Workshop on Hybrid Systems: Computation and Control* (pp. 433-448). Springer Berlin Heidelberg.

[143] Lecchini, A., Glover, W., Lygeros, J., & Maciejowski, J. (2005). Air Traffic Control with an expected value criterion. *IFAC Proceedings Volumes*, *38*(1), 13-18.

[144] Pellegrini, S., Ess, A., Schindler, K., & Van Gool, L. (2009, September). You'll never walk alone: Modeling social behavior for multi-target tracking. In*2009 IEEE 12th International Conference on Computer Vision* (pp. 261-268). IEEE.

[145] Helbing, D., Farkas, I., & Vicsek, T. (2000). Simulating dynamical features of escape panic. *Nature*, *407*(6803), 487-490.

[146] Yu, M., Oh, H., & Chen, W. H. (2016). An improved multiple model particle filtering approach for manoeuvring target tracking using airborne GMTI with geographic information. *Aerospace Science and Technology*, *52*, 62-69.

[147] Thrun, S., Burgard, W., & Fox, D. (2005). *Probabilistic robotics*. MIT press.

[148] Levinson, J., Askeland, J., Becker, J., Dolson, J., Held, D., Kammel, S., ... & Sokolsky, M. (2011, June). Towards fully autonomous driving: Systems and algorithms. In *Intelligent Vehicles Symposium (IV), 2011 IEEE* (pp. 163-168). IEEE.

[149] Goerzen, C., Kong, Z., & Mettler, B. (2010). A survey of motion planning algorithms from the perspective of autonomous UAV guidance. *Journal of Intelligent and Robotic Systems*, *57*(1-4), 65-100.

[150] Karaman, S., & Frazzoli, E. (2011). Sampling-based algorithms for optimal motion planning. *The International Journal of Robotics Research*, *30*(7), 846-894.

[151] Kuffner, J. J., & LaValle, S. M. (2000). RRT-connect: An efficient approach to single-query path planning. In *Robotics and Automation, 2000. Proceedings. ICRA'00. IEEE International Conference on* (Vol. 2, pp. 995-1001). IEEE.

[152] Urmson, C., & Simmons, R. G. (2003, October). Approaches for heuristically biasing RRT growth. In *IROS* (Vol. 2, pp. 1178-1183).

[153] Van Den Berg, J., Abbeel, P., & Goldberg, K. (2011). LQG-MP: Optimized path planning for robots with motion uncertainty and imperfect state information. *The International Journal of Robotics Research*, *30*(7), 895-913.

[154] Brock, O., & Kavraki, L. E. (2001, May). Decomposition-based motion planning: A framework for real-time motion planning in high-dimensional configuration spaces. In *Robotics and Automation, 2001. Proceedings 2001 ICRA. IEEE International Conference on* (Vol. 2, pp. 1469-1474). IEEE.

[155] Utgo, P. E. (1994). An improved algorithm for incremental induction of decision trees. In *Proceedings of the Eleventh International Conference on Machine Learning* (pp. 318-325).

[156] Svenstrup, M., Bak, T., & Andersen, H. J. (2010, October). Trajectory planning for robots in dynamic human environments. In *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on* (pp. 4293-4298). IEEE.

[157] Shkolnik, A., Walter, M., & Tedrake, R. (2009, May). Reachability-guided sampling for planning under differential constraints. In *Robotics and Automation, 2009. ICRA'09. IEEE International Conference on* (pp. 2859-2865). IEEE.

[158] Ferguson, D., & Stentz, A. (2006, October). Anytime rrts. In *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems* (pp. 5369-5375). IEEE.

[159] Shim, D., Chung, H., Kim, H. J., & Sastry, S. (2005, August). Autonomous exploration in unknown urban environments for unmanned aerial vehicles. In*Proc. AIAA GN&C Conference*.

[160] Kelly, A., Stentz, A., Amidi, O., Bode, M., Bradley, D., Diaz-Calderon, A., ... & Rander, P. (2006). Toward reliable off road autonomous vehicles operating in challenging environments. *The International Journal of Robotics Research*,*25*(5-6), 449-483.

[161] Caldwell, C. V., Dunlap, D. D., & Collins, E. G. (2010, September). Motion planning for an autonomous underwater vehicle via sampling based model predictive control. In *OCEANS 2010 MTS/IEEE SEATTLE* (pp. 1-6). IEEE.

[162] Nagy, B., & Kelly, A. (2001). Trajectory generation for car-like robots using cubic curvature polynomials. *Field and Service Robots*, *11*.

[163] Gasparetto, A., & Zanotto, V. (2007). A new method for smooth trajectory planning of robot manipulators. *Mechanism and machine theory*, *42*(4), 455-471.

[164] Yao, J., Lin, C., Xie, X., Wang, A. J., & Hung, C. C. (2010, April). Path planning for virtual human motion using improved A* star algorithm. In*Information Technology: New Generations (ITNG), 2010 Seventh International Conference on* (pp. 1154-1158). IEEE.

[165] Borenstein, J., & Koren, Y. (1991). The vector field histogram-fast obstacle avoidance for mobile robots. *IEEE Transactions on Robotics and Automation*, *7*(3), 278-288.

[166] Kim, J. O., & Khosla, P. K. (1992). Real-time obstacle avoidance using harmonic potential functions. *IEEE Transactions on Robotics and Automation*, *8*(3), 338-349.

[167] Ungarala, S. (2009). Computing arrival cost parameters in moving horizon estimation using sampling based filters. *Journal of Process Control*, *19*(9), 1576-1588.

[168] Storn, R., & Price, K. (1997). Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces. *Journal of global optimization*, *11*(4), 341-359.

[169] Hoy, M. (2014). Methods for collision-free navigation of multiple mobile robots in unknown cluttered environments. *arXiv preprint arXiv:1401.6775*.

Appendix A

Reference: [107]

Table 2 Filter results for the nonlinear pendulum example

Filter type	RMS estimation error (Q, $\tilde{Q}$)	RMS constraint error (Q, $\tilde{Q}$)
unconstrained	0.0411, 0.0253	0.1167, 0.0417
perfect measurement	0.0316, 0.0905	0.0660, 0.0658
estimate projection	0.0288, 0.0207	0.0035, 0.0003
MHE, horizon size 2	0.0105, 0.0067	0.0033, 0.0008
MHE, horizon size 4	0.0089, 0.0067	0.0044, 0.0007
system projection	N/A, 0.0250	N/A, 0.0241
PDF truncation	0.0288, 0.0207	0.0035, 0.0003
2nd order constraint	0.0288, 0.0204	0.0001, 0.0000
SCKF	0.0270, 0.0235	0.0000, 0.0000
unconstrained UKF	0.0400, 0.0237	0.1147, 0.0377
projected UKF	0.0280, 0.0192	0.0046, 0.0007
equality constrained UKF	0.0261, 0.0173	0.0033, 0.0004
two-step UKF	0.0286, 0.0199	0.0005, 0.0000

Two numbers in each cell indicate the errors that are obtained when Q and $\tilde{Q}$ respectively are used in the filter. The numbers shown are RMS errors averaged over 100 Monte Carlo simulations

Appendix B

Code for MHE-MHT main structure:

```
for t = 1 : nStep

Z = meas{t};
cellHypoSeed = {cellHypo{head:rear}};     % generate new hypoes in MHT
cellTargSeed = {cellTarg{head:rear}};

if isempty(Z)
 error('There in no measurement at step %d.', t);
 else
 [cellHypoNew] = GenHypo(cellHypoSeed, cellTargSeed,F, Q, G, M,Z, H, R, Pd,
                          densNew, densClt,pg);    % data association in MHT

 R_cellHypoNew = Reduce_Hypo(cellHypoNew, M, N, t);   % Hypothesis merging

cellTargNew = MHE_MHT_Update(cellTargSeed, R_cellHypoNew,M, Z, H, R, F, Q,
G, maxLifePoint,lasttime_index,t, numTarget,nStep); % MHE and MHT combination

 cellHypo = [cellHypo, R_cellHypoNew];
 cellTarg = [cellTarg, cellTargNew];
 lasttime_index=[];

 for i=1:M
   lasttime_index(i) = cell2mat(R_cellHypoNew{i}(3));
 end
      if t < N
        head = rear + 1;
        rear = rear + M;
      else
        [finalcellEstm,cellEstm, cellHypo, cellTarg] = Prune(finalcellEstm,cellEstm,
              cellHypo, cellTarg, M, N, t, maxLifePoint); % m best N scan prunning
      end
  end
 last=[];
end
```

Code for MHE_MHT_Update function:

```
function cellTargNew = MHE_MHT_Update(cellTargSeed, R_cellHypoNew,...
    M, Z, H, R,A,Q,G, maxLifePoint,lasttime_index,t,numTarget,nStep)

cellTargNew = cell(1, size(R_cellHypoNew, 2)); %%%%3 empty cells
index=[];
global time_new;
global X;
global Xmhe;
global Wmhe;
global Vmhe;
global xhat0;
global dk;
global horizon;
global Pnow;
global X0;
global nx;%number of states
global nm; %number of measurements (noise) v
global nw; %number of process noise w
global nmw;% total number of combined noise
for i = 1 : size(R_cellHypoNew, 2)
     asso = R_cellHypoNew{i}{1}  ;
     index(i) = R_cellHypoNew{i}{4}  ;
     a=find(index(i)==lasttime_index);
      %%%
     saving=i+M;
     time_new(saving,:)= time_new(a,:);
        nExistedTarg = size(cellTargSeed{a}, 2);
        maxTargIdx = max(cellfun(@(v) v{1}, cellTargSeed{a}));
        nNewTarg = sum(asso > maxTargIdx);
        aCase = cell(1, nExistedTarg+nNewTarg);

        for k = 1 : nExistedTarg
            aTarg = cellTargSeed{a}{k}; % one target
            idx = aTarg{1}; % the index of aTarg
            lifePoint = aTarg{2};
            XX = aTarg{3};
            P = aTarg{4};
               %% Case one permanent disappeared target (Already
disappeared previously)
           if lifePoint == 0 % a disappeared target
                aCase{k} = aTarg;
                continue; % just pass it
           end

                %% Case two updating to existing target or target
temporary missing at current time step (or already missing from
previous time step but not yet permanantly disappeared )
            flg = find(asso == idx);
                     %% Case two.one: temporary missing
               if isempty(flg) % there is no meas asso with aTarg
                lifePoint = lifePoint - 1;
                     if lifePoint>0
                   saving=i+M;
                   %%%%%%%%%%%%%%%%%%%%%%%%%%
                   tt=t-time_new{a,idx};
                   %%%%%%%%%%%%%%%%%%%%%%%%%%
                   if tt<=horizon
```

```
                    Pnow{saving,idx}=diag([R(1,1), R(2,2), Q(1,1),
Q(2,2)]);
                        if tt==1
                        dk{saving,idx}(:,tt)=H*A*XX;  %%presuming that
the predicted state are measurement
                        %X0=zeros((2*nx+nm+nw),1);
                        xhat0{saving,idx}=XX  ;   %%%not changed

X{saving,idx}=[zeros(nmw,1);xhat0{saving,idx};zeros(nx,1)];
                        Xmhe{saving,idx}(:,tt)=A*XX;
                        else
                        dk{saving,idx}=dk{a,idx};
                        dk{saving,idx}(:,tt)=H*A*XX;  %%presuming that
the predicted state are measurement
                        X{saving,idx}= [X{a,idx}(1:(tt-
1)*nmw);zeros(nmw,1);X{a,idx}((tt-1)*nmw+1:(tt-
1)*nmw+nx);zeros(nx,1)];
                        xhat0{saving,idx}=X{a,idx}((tt-1)*nmw+1:(tt-
1)*nmw+nx);
                        Xmhe{saving,idx}= Xmhe{a,idx};
                        Xmhe{saving,idx}(:,tt)=A*XX;
                        end
                    else
                    xhat0{saving,idx}=Xmhe{a,idx}(:,tt-horizon);  %%
now xhat0 will not be affected by the non-detected measurement
                    Pnow{saving,idx} = A * Pnow{a,idx}* A' + G*Q*G';
                    dk{saving,idx}(:,1:horizon-
1)=dk{a,idx}(:,2:horizon);
                    dk{saving,idx}(:,horizon)=H*A*XX;

X{saving,idx}=[X{a,idx}(nmw+1:horizon*nmw);zeros(nmw,1);xhat0{saving,
idx};X{a,idx}((end-(nmw-1)):end)];
                    Xmhe{saving,idx}= Xmhe{a,idx};
                    Xmhe{saving,idx}(:,tt)=A*XX;
                    end
                      end
                   XX=A*XX;
                 %% need to be discussed as either predicted update or
measurement update
                    P = A * P * A' + G*Q*G';
             else
               %% target associated with measurement
                     %% Case two.two: target detected
               %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%
                 aMeas= Z(:, flg);
       % MHE estimation%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
       % cellTargSeed{a}{k} :: here at each time step t: 'a'
represeting
       % the number of hypothesis (target cell) and 'k' represeting
the each target in
       % that target cell
              if idx==1

[Xmhe,Wmhe,Vmhe]=C_MHT_MHE(t,i,a,idx,R,Q,H,XX,aMeas,M);
              elseif idx==2

[Xmhe,Wmhe,Vmhe]=C2_MHT_MHE(t,i,a,idx,R,Q,H,XX,aMeas,M);
              else

[Xmhe,Wmhe,Vmhe]=C3_MHT_MHE(t,i,a,idx,R,Q,H,XX,aMeas,M);
```

```
                end
                 saving=i+M;
                 ttt=t-time_new{a,idx};
                 XX=Xmhe{saving,idx}(:,ttt);
                 %%%%%%%%%%%%%%%%
                  P = A * P * A' + G*Q*G';
                  S = R + H*P*H';
                  K = P*H'/S;
                  P = P - K*S*K';
                        if lifePoint < maxLifePoint
                             lifePoint = lifePoint + 1;
                        end
              end
                aTarg{2} = lifePoint;
                aTarg{3} = XX;   %%%%%%%%%%%%%%%%%%it can either be
measurement updated or not updated%%%
                aTarg{4} = P;
                aCase{k} = aTarg;
          end
          for k = 1 : nNewTarg
              idx = maxTargIdx + k; %%%starts from the 5th target (the
first new target)
              flg = find(asso == idx);
              aMeas = Z(:, flg);
              time_new{saving,idx}=t;
              % initialize a new target
              aTarg = cell(1, 4);
              aTarg{1} = idx;
              aTarg{2} = 1;
              aTarg{3} = [aMeas(1), aMeas(2),  0,0]';
              aTarg{4} = diag([R(1,1), R(2,2), Q(1,1), Q(2,2)]);
              aCase{idx} = aTarg;
          end

      cellTargNew{i} = aCase;
```

Appendix C

Code for force_model function:

```
function [cellTargSeed,
socialcellTargSeed]=force_model(t,cellTargSeed,parameter,roads,F,Q,G,
T)

for i = 1 : size(cellTargSeed, 2)
 forcecell = cellTargSeed{i};
 %%known as three targets only
for j = 1 : size( forcecell, 2)
if forcecell{j}{2} == 0
  continue; %Pass control to next iteration of for or while loop
end
 index=setdiff([1:size( forcecell, 2)],j);
 %%define all the parameters
 x_socialforce= forcecell{j}{3};  %x
 rmode=forcecell{j}{5}(end);  %mode
 predicted_x=F * x_socialforce;
 %% road repulsive force
predicted_distance_to_boundary1=(roads(rmode).boundary(1,:)*[predicte
d_x(2);predicted_x(1);1])/sqrt(roads(rmode).boundary(1,1)^2+roads(rmo
de).boundary(1,2)^2);
predicted_distance_to_boundary2=(roads(rmode).boundary(2,:)*[predicte
d_x(2);predicted_x(1);1])/sqrt(roads(rmode).boundary(2,1)^2+roads(rmo
de).boundary(2,2)^2);
if (roads(rmode).angle<=pi/2)
boundary1_force=parameter(1)*exp(-
parameter(2)*predicted_distance_to_boundary1)*[-
sin(roads(rmode).angle);cos(roads(rmode).angle)];
boundary2_force=parameter(1)*exp(parameter(2)*predicted_distance_to_b
oundary2)*[sin(roads(rmode).angle);-cos(roads(rmode).angle)];
else
boundary1_force=parameter(1)*exp(-
parameter(2)*predicted_distance_to_boundary1)*[sin(roads(rmode).angle
);-cos(roads(rmode).angle)];
boundary2_force=parameter(1)*exp(parameter(2)*predicted_distance_to_b
oundary2)*[-sin(roads(rmode).angle);cos(roads(rmode).angle)];
end
 % velocity repulsive force
```

```
along_road_velocities=[cos(roads(rmode).angle),sin(roads(rmode).angle
)]*predicted_x(3:4);
directions=sign(along_road_velocities);
amplitudes=abs(along_road_velocities);
velocity_force=-
directions.*(parameter(5)*exp(parameter(6)*(amplitudes-
roads(rmode).speedlimitation)))*[cos(roads(rmode).angle);sin(roads(rm
ode).angle)];
 %% giveway_force
 giveway_force=zeros(2,1);
 if((rmode==2)|(rmode==4))

along_road_distances=abs([cos(roads(rmode).angle),sin(roads(rmode).an
gle)]*(x_socialforce(1:2)-roads(rmode).startpoint'));
 if(along_road_distances<=16)&(directions==-1);
 %giveway_force=parameter(7)*(1-exp(-
parameter(8)*amplitudes))*[cos(roads(rmode).angle);sin(roads(rmode).a
ngle)];
  giveway_force=parameter(7)*(1-exp(-
parameter(8)*amplitudes))*[cos(roads(rmode).angle);sin(roads(rmode).a
ngle)];
 end
 end
 %% interaction force
 for ii=1:length(index)
 othervehicle(:,ii)= F*forcecell{index(ii)}{3};
 distances(ii)=sqrt((predicted_x(1)-
othervehicle(1,ii))^2+(predicted_x(2)-othervehicle(2,ii))^2);
 vector_with_othervehicles(:,ii)=(predicted_x(1:2)-
othervehicle(1:2,ii))/distances(ii);
 vehicle_force(:,ii)=zeros(2,1);
 if distances(ii)<10
 vehicle_force(:,ii)=parameter(3)*exp(-
parameter(4)*distances(ii))*vector_with_othervehicles(ii);
 end
 end
if (sum(abs(boundary1_force))==Inf)
    boundary1_force=[0;0];
end
if (sum(abs(boundary2_force))==Inf)
    boundary2_force=[0;0];
end
if (sum(abs(velocity_force))==Inf)
    velocity_force=[0;0];
end
if (sum(abs(giveway_force))==Inf)
    giveway_force=[0;0];
end

total_force=boundary1_force+boundary2_force+velocity_force+giveway_fo
rce;

 for ii=1:length(index)

   if (sum(abs(vehicle_force(:,ii)))==Inf)
    vehicle_force(:,ii)=[0;0];
   end
 total_force=total_force+vehicle_force(:,ii);
 end
 predicted_x=predicted_x+ G * (total_force);
 forcecell{j}{3}= predicted_x;
```

```
 forcecell{j}{6}= [forcecell{j}{6},total_force];
 %%%calculate jacobain F for social force
 P = forcecell{j}{4};
 P = F* P * F' + G*Q*G';
 forcecell{j}{4}=P;
 %%%%
 cellTargSeed{i}{j}{6}=[forcecell{j}{6}];
 %%%%
end
 socialcellTargSeed{i} =  forcecell;
end
```

Appendix D

Standard RRT algorithm

Standard RRT algorithm

Standard RRT [43]

RRTmain*(Tree)*

1. $Tree = q_{start}$
2. $q_{new} = q_{start}$
3. *while Distance* (q_{new}, q_{goal}) < *Error-tolerance*
4. q_{target} = ***SampleTarget****()*
5. $q_{nearest}$ = ***NearestVertex****(Tree,*q_{target}*)*
6. q_{new} = ***ExtendTowards****(*$q_{nearest}$,q_{target}*)*
7. *if* q_{new} *valid*
8. ***Tree.add****(*q_{new}*)*
9. *else*
10. *continue*
11. *end if*
12. *end while*

SampleTarget*()*

13. *p = RandomReal([0.0, 1.0])*
14. *if (p < goal-sampling-prob)*
15. *return* q_{goal}
16. *else*
17. *return* ***RandomConfiguration****()*
18. *end if*

Publications

1. Runxiao Ding, and Wen-Hua Chen. "Moving object tracking in support of unmanned vehicle opeartion." *Automation and Computing (ICAC), 2013 19th International Conference on*. IEEE, 2013.
2. Runxiao Ding, Miao Yu, and Wen-Hua Chen. "A Multiple Target Tracking Strategy Using Moving Horizon Estimation Approach." *24th International Technical Conference on the Enhanced Safety of Vehicles (ESV)*. No. 15-0390. 2015.
3. Miao Yu, Runxiao Ding and Wen-hua Chen. "New environmental dependent modelling with Gaussian particle filtering based implementation for ground vehicle tracking." *Sensor Signal Processing for Defence (SSPD), 2016 International Conference on*. IEEE, 2016.
4. Runxiao Ding, Hyondong Oh and Wen-Hua Chen. "Autonomous vehicle path planning in dynamic environment with trajectory prediction, *Journal of Intelligent & Robotic Systems*, to be submitted.
5. Runxiao Ding, Miao Yu, Hyondong Oh and Wen-Hua Chen. "New Multiple Target Tracking Strategy Using Domain Knowledge and Optimization." *Transactions on Systems, Man and Cybernetics: Systems (SMC)*, Minor correction, under review.