Autonomous road vehicles localization using satellites, lane markings and vision

Zui Tao

To cite this version:

Zui Tao. Autonomous road vehicles localization using satellites, lane markings and vision. Robotics [cs.RO]. Laboratoire Heudiasyc, 2016. English. NNT : . tel-01529732v1

HAL Id: tel-01529732

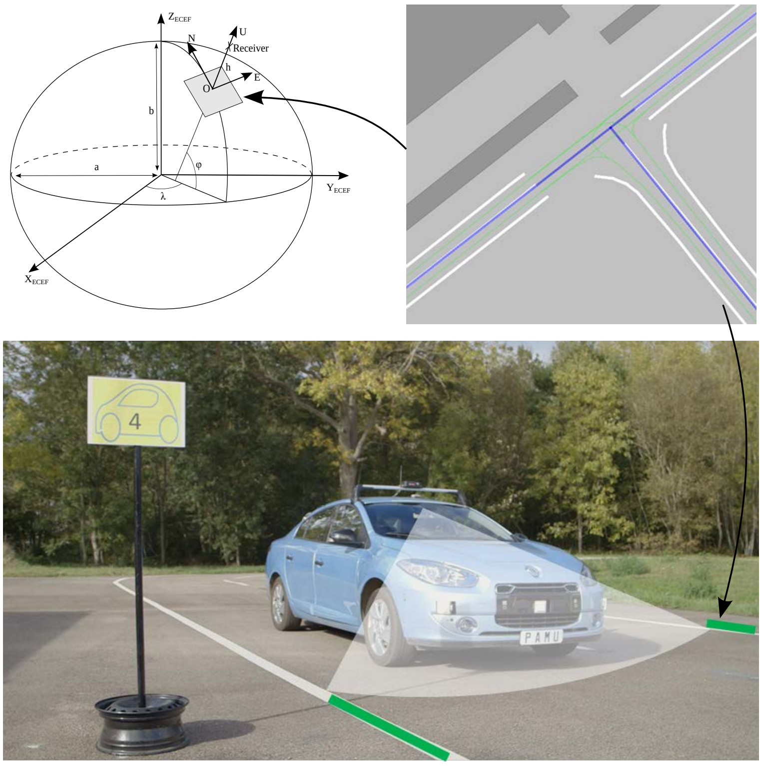
https://hal.science/tel-01529732v1

Submitted on 31 May 2017 (v1), last revised 24 Feb 2017 (v2)

by Zui TAO

Autonomous road vehicles localization using satellites, lane markings and vision

Thesis submitted of the requirements for the degree
of Doctor of Philosophy of UTC



Defense date : the Twenty-second of February, 2016
Area of specialization : Information Technologies and Systems

UNIVERSITY OF TECHNOLOGY OF COMPIÈGNE
THESIS

Submitted of the requirements for the degree of

Doctor of Philosophy
Area of specialization: Information Technologies and Systems
by
Zui TAO

Autonomous Road Vehicles Localization Using Satellites, Lane markings and Vision

Heudiasyc Laboratory, UMR UTC/CNRS 7253

Defence date: the Twenty-second of February, 2016

Thesis Committee:

Reviewers:	David BETAILLE	IFSTTAR
	Roland CHAPUIS	Université Blaise Pascal
Examiners:	Vincent FREMONT	Université de Technologie de Compègne
	Patrick RIVES	INRIA
	Javier IBANEZ-GUZMAN	Renault S.A.
Supervisors:	Philippe BONNIFAIT	Université de Technologie de Compiègne

Acknowledgements

At first, I would like to acknowledge my thesis supervisor, Philippe Bonnifait, for his patient guidance, continuous encouragement, and confidence in me. Thanks for the time he has given to our constructive discussions along my PhD journey. His enthusiasm in research and serious academic attitudes have also influenced me a lot.

Thank you to Clémont Zinoune, Arthur de Miranda Neto, Vincent Frémont, and Javier Ibanez-Guzman for their help at the beginning of my thesis. My special thanks to Javier for giving me access to experimental data sets within project PAMU.

I would like to acknowledge my committee members : Roland Chapuis, David Bétaille, Vincent Frémont, Patrick Rives, Javier Ibanez-Guzman, and Philippe Bonnifait, for their support in completing the research documented herein.

My gratitude for my colleagues : Khaoula, Yue, Kun, Zhicheng, Nicole, Gilles, Danilo, Julien, Bin, Adam, Carine, Frank, Hafida, Alia, Chunlei, Phuc, Dingfu. Thanks for all of you for making a comfortable environment and offering scientific support. I would like to send specific acknowledgements to Hoda Dandach, Qiaochu Li, and Bihao Wang for the useful and fun-filled discussions with them on my research and life.

I am very thankful to Stéphanne Bonnet, Thierry Monglchongon, Gérald Dherbomez for providing support during data collection, development, and testing. It was great working with such a nice group of people.

I am also grateful that China Scholarship Council provided me financial support which helped me to do research with a free mind. Thanks to the Sappart project commitee who has provided me support in attending scientific conferences with the European GNSS specialists which helps me to broaden my research horizon.

My thanks to Lanting Yu who decided to finish her studies at Compiègne for living with me. I appreciate your love and companionship.

Thank you to my sister Ye Tao for her support and encouragement despite the long distance. My special acknowledgements to my nephews Linxin and Linze who bring me so much happiness. Last but not least, to my father Lijun Tao and mother Cuiping Cheng, thank you for your endless and unconditional love throughout my life. This thesis is dedicated to you.

Abstract

Estimating the pose (position and attitude) in real-time is a key function for road autonomous vehicles. This thesis aims at studying vehicle localization performance using low cost automotive sensors. Three kinds of sensors are considered : dead reckoning (DR) sensors that already exist in modern vehicles, mono-frequency GNSS (Global navigation satellite system) receivers with patch antennas and a front-looking lane detection camera. Highly accurate maps enhanced with road features are also key components for autonomous vehicle navigation. In this work, a lane marking map with decimeter-level accuracy is considered. The localization problem is studied in a local East-North-Up (ENU) working frame. Indeed, the localization outputs are used in real-time as inputs to a path planner and a motion generator to make a valet vehicle able to drive autonomously at low speed with nobody on-board the car.

The use of a lane detection camera makes possible to exploit lane marking information stored in the georeferenced map. A lane marking detection module detects the vehicle's host lane and provides the lateral distance between the detected lane marking and the vehicle. The camera is also able to identify the type of the detected lane markings (e.g., solid or dashed). Since the camera gives relative measurements, the important step is to link the measures with the vehicle's state. A refined camera observation model is proposed. It expresses the camera metric measurements as a function of the vehicle's state vector and the parameters of the detected lane markings.

However, the use of a camera alone has some limitations. For example, lane markings can be missing in some parts of the navigation area and the camera sometimes fails to detect the lane markings in particular at cross-roads. GNSS, which is mandatory for cold start initialization, can be used also continuously in the multi-sensor localization system as done often when GNSS compensates for the DR drift. GNSS positioning errors can't be modeled as white noises in particular with low cost mono-frequency receivers working in a standalone way, due to the unknown delays when the satellites signals cross the atmosphere and real-time satellites orbits errors. GNSS can also be affected by strong biases which are mainly due to multipath effect. This thesis studies GNSS biases shaping models that are used in the localization solver by augmenting the state vector. An abrupt bias due to multipath is seen as an outlier that has to be rejected by the filter. Depending on the information flows between the GNSS receiver and the other components of the localization system, data-fusion architectures are commonly referred to as loosely coupled (GNSS fixes and velocities) and tightly

coupled (raw pseudoranges and Dopplers for the satellites in view). This thesis investigates both approaches. In particular, a road-invariant approach is proposed to handle a refined modeling of the GNSS error in the loosely coupled approach since the camera can only improve the localization performance in the lateral direction of the road.

Finally, this research discusses some map-matching issues for instance when the uncertainty domain of the vehicle state becomes large if the camera is blind. It is challenging in this case to distinguish between different lanes when the camera retrieves lane marking measurements.

As many outdoor experiments have been carried out with equipped vehicles, every problem addressed in this thesis is evaluated with real data. The different studied approaches that perform the data fusion of DR, GNSS, camera and lane marking map are compared and several conclusions are drawn on the fusion architecture choice.

Résumé

L'estimation de la pose (position et l'attitude) en temps réel est une fonction clé pour les véhicules autonomes routiers. Cette thèse vise à étudier des systèmes de localisation pour ces véhicules en utilisant des capteurs automobiles à faible coût. Trois types de capteurs sont considérés : des capteurs à l'estime qui existent déjà dans les automobiles modernes, des récepteurs GNSS mono-fréquence avec antenne patch et une caméra de détection de la voie regardant vers l'avant. Les cartes très précises sont également des composants clés pour la navigation des véhicules autonomes. Dans ce travail, une carte de marquage de voies avec une précision de l'ordre du décimètre est considérée. Le problème de la localisation est étudié dans un repère de travail local Est-Nord-Haut. En effet, les sorties du système de localisation sont utilisées en temps réel comme entrées dans un planificateur de trajectoire et un contrôleur de mouvement pour faire en sorte qu'un véhicule soit capable d'évoluer au volant de façon autonome à faible vitesse avec personne à bord. Ceci permet de développer des applications de voiturier autonome aussi appelées « valet de parking ».

L'utilisation d'une caméra de détection de voie rend possible l'exploitation des informations de marquage de voie stockées dans une carte géoréférencée. Un module de détection de marquage détecte la voie hôte du véhicule et fournit la distance latérale entre le marquage de voie détecté et le véhicule. La caméra est également capable d'identifier le type des marquages détectés au sol (par exemple, de type continu ou pointillé). Comme la caméra donne des mesures relatives, une étape importante consiste à relier les mesures à l'état du véhicule. Un modèle d'observation raffiné de la caméra est proposé. Il exprime les mesures métriques de la caméra en fonction du vecteur d'état du véhicule et des paramètres des marquages au sol détectés.

Cependant, l'utilisation seule d'une caméra a des limites. Par exemple, les marquages des voies peuvent être absents dans certaines parties de la zone de navigation et la caméra ne parvient pas toujours à détecter les marquages au sol, en particulier, dans les zones d'intersection. Un récepteur GNSS, qui est obligatoire pour le démarrage à froid, peut également être utilisé en continu dans le système de localisation multi-capteur du fait qu'il permet de compenser la dérive de l'estime. Les erreurs de positionnement GNSS ne peuvent pas être modélisées simplement comme des bruits blancs, en particulier avec des récepteurs mono-fréquence à faible coût travaillant de manière autonome, en raison des perturbations atmosphériques sur les signaux des satellites et les erreurs d'orbites. Un récepteur GNSS peut également être affecté par de fortes perturbations locales qui sont principalement dues aux multi-trajets. Cette

thèse étudie des modèles formeurs de biais d'erreur GNSS qui sont utilisés dans le solveur de localisation en augmentant le vecteur d'état. Une variation brutale due à multi-trajet est considérée comme une valeur aberrante qui doit être rejetée par le filtre.

Selon le flux d'informations entre le récepteur GNSS et les autres composants du système de localisation, les architectures de fusion de données sont communément appelées « couplage lâche » (positions et vitesses GNSS) ou « couplage serré » (pseudo-distance et Doppler sur les satellites en vue). Cette thèse étudie les deux approches. En particulier, une approche invariante selon la route est proposée pour gérer une modélisation raffinée de l'erreur GNSS dans l'approche par couplage lâche puisque la caméra ne peut améliorer la performance de localisation que dans la direction latérale de la route.

Enfin, cette recherche traite de certains problèmes d'ambigüité de map matching par exemple lorsque le domaine d'incertitude de l'état du véhicule devient grand si la caméra est aveugle. Il est en effet difficile dans ce cas de distinguer les différentes voies lorsque la caméra récupère à nouveau des mesures de marquage de voie.

Comme de nombreuses expériences ont été effectuées avec des véhicules équipés, chaque problème abordé dans cette thèse est évalué avec des données réelles. Les différentes approches étudiées qui effectuent la fusion de données des différents capteurs et de la carte contenant les marquages de voie sont comparées entre elles et plusieurs conclusions sont tirées sur le choix de l'architecture de fusion.

Contents

List of Figures

Chapter 1

General Introduction

Autonomous road vehicles (ARVs) have existed as prototypes and demonstration vehicles since the 1970s. Their widespread use promises increased comfort, safety, reduced traffic congestion, energy conservation and pollution reductions [83]. In November 2007, the DARPA (Defense Advanced Research Projects Agency) Urban Challenge hold in Victoria City in the USA. In this event, autonomous vehicles have interacted with both manned and unmanned vehicles in an urban environment. It required teams to build autonomous vehicles capable of driving in traffic (in fact, the artificial traffic environment was with no pedestrians or bicyclists and had no traffic lights), performing complex maneuvers such as merging, passing, parking and negotiating intersections. Over fifty vehicles took the challenge and six of them completed the race. Since then, many efforts have been done to improve algorithm performance and operate autonomous car on public roads. Successful demonstrations have been carried out.

Usually, an ARV needs to perform four kinds of tasks: localization, perception, path planning, and control. Localization primarily refers to determining the vehicle's position, velocity and attitude in a map that includes the vehicle's goals. The perception problem is the task of making sense of vehicle's surroundings from sensing data. This task generally consists of detection and tracking of both dynamic objects and road environmental static features. Path planning is the task of determining and executing a series of actions to achieve the vehicle's goals. Successful planning implies a clear understanding of the effects of the vehicle's actions on itself and its environment. Control systems enable the vehicle to react in real time to changes in the operational environment by following the planned moves.

Usually, expensive and redundant sensor suits are used to achieve a high level of performance such that the system can be trusted to replace human operation. One typical representative of ARVs is the Google's self-driving car. The vehicle carries a variety of sensors which include: four radars (mounted on the front and rear bumpers, that allow the car to "see" far enough to be able to deal with fast traffic on freeways), a 3D lidar (mounted on the top of the vehicle for object detection), a camera (positioned near the rear-view mirror, that detects traffic lights), a RTK-GPS, a high-grade Inertial Measurement Unit (IMU) assisted with a wheel encoder which determines the vehicle's location and keeps track of its movements. Although the cost of the sensors is going down as robotic cars become more ubiquitous, it still remains as a barrier to their adoption in series automobiles. Low cost approaches integrating market-ready sensors make sense in promoting the development of ARVs. The European V-Charge project [44] is one of the approach that tries to address this problem. In this work, the vehicle is equipped with sonars, cameras, standard GPS receiver, odometers, accelerometers and gyroscopes to achieve autonomous driving.

In addition to the sensor suit, another crucial element for autonomous driving is the map. In fact, most successful demonstrations make use of maps [80][134]. Digital road maps were originally developed for route planning to find global topological paths on the maps. For this kind of application, the road network is generally represented by a planar model and a set of connected road centerlines. Advanced Driving Assistance System (ADAS) and autonomous driving rely on the development of more accurate and detailed maps [20]. Advanced systems such as lane-level navigation and autonomous driving need to plan a route more precisely. The task includes not only the choice of which road to take, but also which lane to drive in. The map is designed to represent lanes, lane markings, buildings, traffic signs, etc. As the accuracy of enhanced maps increases, the requirements on localization performance are more and more exigent, because localization is an essential prerequisite for relating the vehicle state with the map. The localization is not only an important input for the planning system but also helps in perception tasks. Recently, map-aided perception is intensively considered for autonomous driving because prior knowledge makes the processing and understanding of perception data easier. [81] outlines a map-based approach to control the region of interest for detecting the position and orientation of traffic light with in camera images. In [108][33], map coupled with localization is used to help in lane marking detection. [74] exploits prior map for dynamic scene understanding. Due to the existence of detailed map, localization and perception are linked as interdependent tasks.

This thesis aims at working out a localization system which provides in real-time an accurate pose estimate which is used as a feedback to make navigate autonomously a full-size valet car (Figure 1.1). In fact, the pose is used by a motion planner that enables the car to follow a predetermined path defined by the mission to achieve. We focus on map-aided methods to explore the feasibility of using low-cost automotive sensors to achieve this goal.

The term “low-cost”has a two-fold meaning. Firstly, expensive sensors such as high-end IMU, Differential GNSS and 3D laser scanners are not adopted. Instead, MEMS inertial navigation sensors, L1-GNSS and automotive smart cameras are preferred. Secondly, sensors that already exist on modern automobile for some ADAS or safety functions are reused for localization with minor or non modification. These are for instance, wheel speed sensors or yaw rate gyros.

The combination of inertial navigation systems (INS) and GNSS is ideal for vehicle localization. INS and dead reckoning (DR) provide high frequency pose estimation which is accurate for a short-term horizon. GNSS is an efficient mean to correct

(a) *Robotic car named IRIS of the Heudiasyc laboratory* **(b)** *Autonomous valet parking vehicle from Renault PAMU (Urban Mobility Application Platform) project*

Figure 1.1: *The equipped experimental vehicles used in this thesis.*

this drift but it suffers from nonwhite biases and strong multipath in urban canyons. A INS/GNSS coupled system generally provides position estimates with high availability but with an accuracy of several meters which is not enough for autonomous driving.

Accurate and geo-referenced features maps help in improving localization when adapted exteroceptive sensors are integrated. However, the need for an accurate map introduces limitations on the way the localization system can be used since it can only work in a surveyed navigation area. In addition, the performance of perception sensors (like cameras) in complex outdoors environments is still challenging. Therefore, a good solution has to consider all the factors. Firstly, the pose estimation must be accurate enough to fulfill the requirements of motion planning with high availability. Secondly, it should be not too complex to build the map and easy to keep it up-to-date. Finally, sensors which detect map features should be robust and close-to-market. In this thesis, we focus on a lane marking map and a lane detection camera used initially for a Lane Departure Warning System (LDWS).

More specifically, we seek the solutions which give GNSS/DR-like availability and vision/map-like accuracy. For this goal, the following questions need to be addressed:

1. To build an accurate camera observation model and elaborate sensor error models that will be used in the localization solver, especially for GNSS.
2. To design localization solvers and state observers that fulfill system observability conditions.
3. To study different data fusion schemes (e.g., loosely and tightly coupled in terms of GNSS information) and different implementations.
4. To check the consistency of the state observers since reliable confidence is needed for monitoring the localization system.

Manuscript organization The manuscript is organized as follows.

Chapter 2 covers the considered technologies for localization which include dead reckoning, GNSS and vision/map coupled methods. The objective of this chapter is to give a global view of available information sources for this thesis. In particular, a camera observation model is proposed. The outdoor experimental data used in this thesis is reported. Sensor performances are examined using the recorded data, which give important insights in the design of the localization system.

In Chapter 3, sensor error models and data fusion filters are reminded. A loosely coupled data fusion scheme is addressed. Particularly, the shaping models for GNSS errors are investigated using experimental data. A widely-used autoregressive error model is adopted. Experimental results are analyzed to validate the proposed camera observation model and shaping models.

Chapters 4 expands the loosely coupling scheme by an enhanced GNSS error modeling and a different implementation of the filter. The enhanced model is based on an observability analysis which relies on algebraic concepts of observability. Considering that the camera measures the lateral distance, an enhanced modeling of the GNSS error in the lateral direction is proposed. Then, the problem is addressed in a road-oriented frame. A road invariant extended Kalman filter (EKF) is designed to conserve the observability of the proposed state vector.

One important thing to note is the localization task becomes very challenging when the vehicle travels into an urban environment. Performance of both aforementioned loosely coupling approaches degrades greatly. The lower availability of the camera measurement in this area aggravates this situation. Therefore, in Chapter 5, we switch our focus to a scheme that tightly couples the raw measures of the GNSS receiver. One prominent advantage of the tightly coupled approach is that the GNSS receiver can still provide measurements for the localization system even when the satellites in view are less than 4. In addition, the satellite raw measurements contaminated by multipath are easier to be detected and excluded.

Chapter 6 compares the performance of the approaches proposed in Chapter 3, 4 and 5 in terms of accuracy and filter consistency. A multi-hypothesis approach is studied to handle map matching ambiguities.

Chapter 7 concludes the thesis by highlighting the contributions and by providing perspectives for further work.

Chapter 2

Considered Information Sources for Autonomous Road Vehicle Localization

Contents

2.1 Introduction

Sensors and maps are essential information sources for vehicle navigation. Automotive manufacturers are currently giving much interest to in-vehicle sensors and sensing-based services because of their proved benefits in accidents avoidance and higher driving efficiency. Furthermore, autonomous driving hinges on advanced sensors and maps for more challenging tasks such as object detection and tracking, scene understanding and highly accurate self-localization. In this chapter, sensing technologies and maps that help in the vehicle localization task are highlighted. Sensor measurements and their basic observation models are studied.

An automotive localization and navigation system for a human driver typically uses a GNSS device to get its position data which is then correlated with a digital road map for route planning. For a driverless vehicle, GNSS sensors also help other sensory data matching with the enhanced map. The versatility and low cost of GNSS positioning makes it a candidate for autonomous vehicle navigation systems. However, in urban canyons or tunnels, GNSSs can loss signal or suffer from multipath. During short-term outages of satellite, data from vehicle proprioceptive sensors like gyrometers, accelerometers and wheel speed sensors can be used to dead reckon the vehicle pose with an appropriate vehicle kinematic model. In the context of this thesis, those proprioceptive sensors are referred to as dead reckoning (DR) sensors. DR sensors usually have greater reliability and availability than GNSSs but their estimation errors are accumulated unboundedly in long-term. For enhancing the positioning performance, another widely adopted method is using surround sensors (camera/radar/lidar) to extract features in view and map match them with a prior map. We refer to this method as vision/map method. This method usually requires a coarse estimation of the vehicle pose to ensure a fast and accurate map matching with the enhanced map of the features.

For autonomous vehicle navigation or other safety-critical applications, there is no single system that fulfills the reliability and availability for now. The system is always designed in a multisensor fusion framework, i.e. fusion of DR sensors, GNSSs and vision/map sensors which often have CAN (controller area network) bus interfaces. The main objective of this chapter is to outline this three data sources for the localization system. Several useful coordinate systems are firstly introduced. Since the localization is a state estimation problem, along with the introduction of the state-of-the-art technologies, the sensor measurements, vehicle kinematic model and sensor observation models are developed. At the end of this chapter, the experimental platform is presented and experimental data which are used in the next chapters are investigated.

2.2 Reference coordinate systems

Definition of reference coordinate systems is central to vehicle localization problem. Under the multisensor fusion framework, different sub-systems produce information in different reference coordinate systems (RCSs). For example, GNSS receivers record data in latitude-longitude using the WGS84 (World Geodetic System 1984) datum by default. The lane detection camera measures the lane markings in its body frame and a road map is often defined in a local navigation coordinate system. In this section, the useful RCSs are introduced according to three different scales, i.e., global, local and body.

2.2.1 Global coordinates systems

To formulate the mathematics of the global satellite navigation problem, it is necessary to choose a global coordinate reference system in which the states of both the satellites and the receiver can be represented. In this formulation, it is typical to describe satellite and receiver states in terms of position and velocity vectors measured in a Cartesian coordinate system. The two main Cartesian coordinate systems are either inertial or rotating.

The Earth-Centered Inertial Coordinate System The origin of the Earth-Centered Inertial (ECI) coordinate system is defined as coinciding with the Earth's center of mass, hence the name Earth-Centered. The x axis is permanently fixed in a direction relative to the celestial sphere. Its z axis is coincident with Earth's geographical North Polar axis, and its $x-y$ plane coincides with the equatorial plane. The y axis is right hand pointing. The Cartesian coordinates are used to express a location using the ECI system. Comparing with the frames which rotate in inertial space, the equations that describes orbital motion of objects in space are simpler in ECI system. Therefore it is convenient to employ the ECI system to describe the motion of the satellites using Newton's laws. However, it is inconvenient for the Earth surface application, considering that the Earth is rotating with respect to this system. Therefore, another global Cartesian coordinate system ECEF is defined as follows.

The Earth-Centered Earth-Fixed Coordinate System As the ECI system, the Earth-Centered Earth-Fixed Coordinate System (ECEF) also has its $x-y$ plane coincident with the Earth's equatorial plane, and the z axis is defined to be normal to the equatorial plane in the direction of the geographical North Pole. The origin is also defined as the Earth's center of mass. However, its x and y axis rotate with the Earth instead of pointing in fixed directions in inertial space. The x axis points in the direction of 0° latitude, and the y axis points in the direction of 90°E longitude. ECEF system represents positions as an X, Y and Z Cartesian coordinates. A point fixed on the surface of the Earth doesn't change its coordinates in ECEF.

It is typical to transform the Cartesian coordinates to latitude (ϕ), longitude (λ), and height (h). A World Geodetic System is defined to describe the Earth and to carry out this transformation.

World Geodetic System The World Geodetic System (WGS) is a standard for use in geodesy, cartography, and navigation [95]. It comprises a standard coordinate system for the Earth, a standard spheroidal reference surface for raw altitude data, and a gravitational equipotential surface that defines the nominal sea level. The latest revision is WGS84, established in 1984 and last revised in 2004. It is the standard physical model of the Earth used for GPS applications. One part of WGS84 is a detailed model of the Earth's gravitational irregularities. Such information is necessary to derive accurate satellite ephemeris information. However, we are concerned here with estimating the latitude, longitude, and height of a GPS receiver. For this purpose, WGS84 provides an ellipsoidal model of the Earth's shape, as shown in Figure 2.1. In this model, cross-sections of the Earth parallel to the equatorial plane are circular. The equatorial cross-section of the Earth has radius

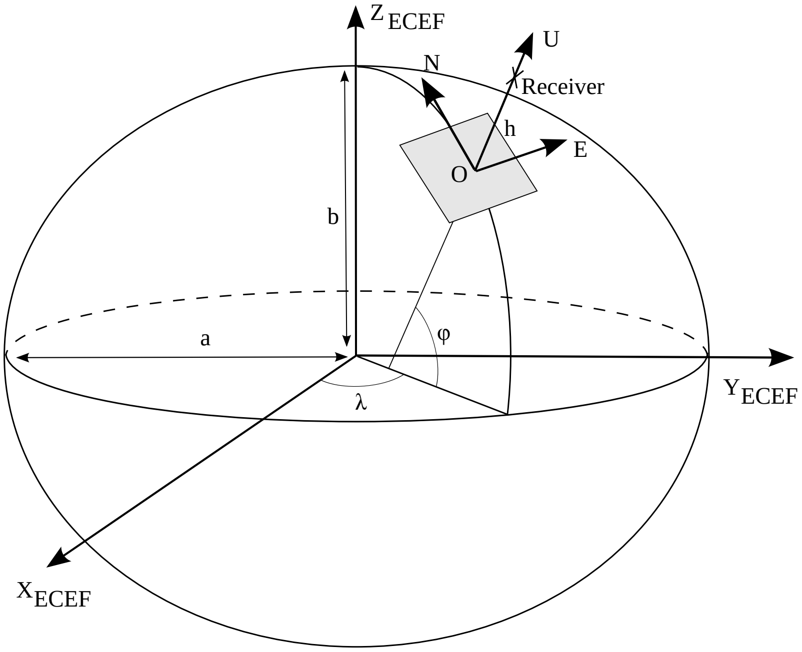


Figure 2.1: *Ellipsoidal model of Earth*

6378.137 km, which is the mean equatorial radius of the Earth. In the WGS84 Earth model, cross-sections of the Earth normal to the equatorial plane are ellipsoidal. In an ellipsoidal cross-section containing the z-axis, the major axis coincides with the equatorial diameter of the Earth. Therefore, the semi-major axis, a, has the same value as the mean equatorial radius given previously. The minor axis of the ellipsoidal cross-section shown in Figure 2.1 corresponds to the polar diameter of the Earth, and the semiminor axis, b, in WGS 84 is taken to be 6,356.7523142 km. Thus, the eccentricity of the Earth ellipsoid, e, can be determined by

$$e = \sqrt{1 - \frac{b^2}{a^2}}$$

WGS84 takes $e^2 = 0.00669437999014$. Locations are defined in terms of the position on a globe using latitude and longitude values. The longitude λ is the angle made by the projection of the user in the equatorial plane and the x-axis of ECEF frame. The longitude ranges from -180° to 180° and positive longitudes correspond to degrees East. The latitude ϕ is the angle made by the ellipsoid normal at the receiver position with the Equator plane and ranges from -90° to 90° with positive values on the northern hemisphere. The height h is the shortest distance between the ellipsoid and the user. It is positive if the user is above the ellipsoid.

The output of a GPS receiver is by default latitude, longitude and height in the WGS84 coordinate system. When testing other navigation instruments, such as inertial navigation system (INS) with DR sensors, it is often helpful to transform

WGS84 coordinates to local navigation coordinates.

2.2.2 Local coordinate systems

East, north, up coordinate system A local East, North, Up (ENU) RCS is a Cartesian coordinate system defined to be tangent to the Earth ellipsoid at a defined origin. The North axis is tangent to the meridian that contains the ENU origin and in the North direction. The East axis is normal to the North axis and is in the positive longitudes direction. The Up axis is chosen so that the ENU is a right-handed coordinate system (Figure 2.1). The ENU system is also known as local navigation frame. It is widely used in the navigation tasks since it is convenient for the computation of motion on short distances.

The localization problem is defined in the local ENU frame. GNSS receivers usually record data in (λ, φ, h) using the WGS84 system. A conversion from a WGS84 Datum to ECEF coordinates (X, Y, Z) is used as an intermediate step in converting to the ENU coordinates (x, y, z). For this thesis, the conversions among WGS84, ECEF and ENU coordinate systems are performed by the algorithm developed in the CityVIP project [2].

Simplified 2D local navigation system When the navigation area is flat enough, the 3D ENU frame can be simplified into 2D coordinates by only considering East and North axis. The frame R_O in Figure 2.2 is the simplified 2D frame with x_O pointing east and y_O pointing north. The localization problem is simplified into a 2D problem in this thesis.

Road-oriented frame A road-oriented frame is defined to have the same origin as the local ENU frame and its x-axis pointing to the direction of the road in which the vehicle is traveling. Its use will be discussed in Chapter 3.

Now, after having defined coordinate systems at the global and local scales, two more frames of body scale need to be defined, i.e., the body frames of the vehicle and the vision sensor (the vision sensor adopted in this thesis is a lane detection camera with more details in Section 2.6).

2.2.3 Mobile coordinate systems

Vehicle body frame The vehicle body frame, denoted R_M in Figure 2.2, is vehicle-carried and is directly defined on the body of the mobile vehicle. Its origin is defined at the center of the rear axis. x_M is the longitudinal axis pointing forward and y_M is such that z_M is upwards. It is the frame used to control the vehicle and has to be localized in R_O.

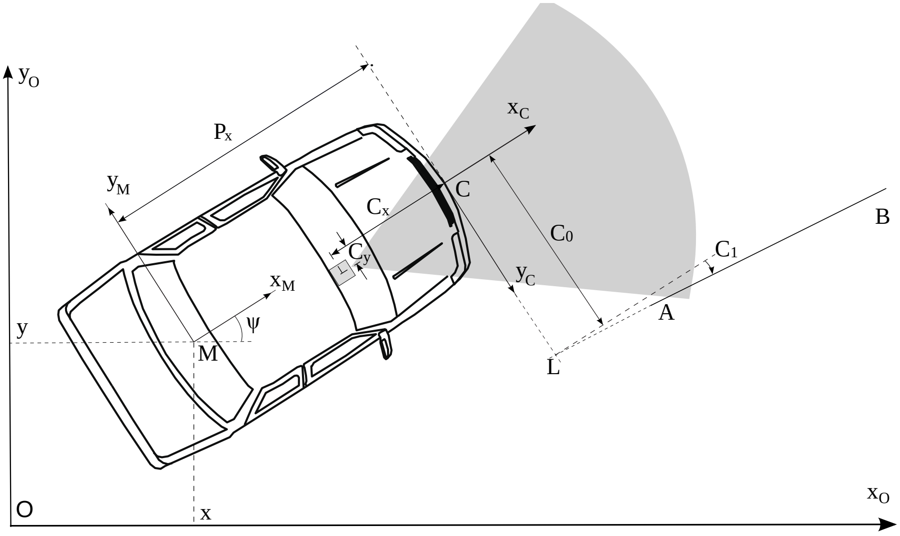


Figure 2.2: *Mobile frames. [AB] is a lane marking segment*

Camera body frame The origin of the camera coordinate system is located at the front of the vehicle after its calibration (Figure 2.2). In order to stay consistent with the camera system conventions used in this work, z_C is downward. The Lane Departure Warning camera is often installed behind the windscreen with a position offset (C_x,C_y). Every detected lane marking is expressed in R_C. P_x is a translation from point M up to the bumper.

The localization problem is studied in the local ENU frame, hence the transformation between these coordinate systems are defined as following.

Let denote ${}^M\mathbf{T}_C$ the 2D homogeneous transformation matrix mapping R_C in R_M:

$$ {}^M\mathbf{T}_C = \begin{bmatrix} 1 & 0 & P_x \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2.1} $$

Let define $\mathbf{q} = (x, y, \psi)^T$ the vehicle's pose which includes position (x, y) and heading angle ψ. The transformation ${}^O\mathbf{T}_M$ maps R_M in R_O:

$$ {}^O\mathbf{T}_M = \begin{bmatrix} \cos\psi & -\sin\psi & x \\ \sin\psi & \cos\psi & y \\ 0 & 0 & 1 \end{bmatrix} \tag{2.2} $$

Solving the localization problem means finding the pose vector "$\mathbf{q}$" in R_O. We begin with presenting the dead reckoning method.

2.3 Dead reckoning

In navigation, dead reckoning (DR) is the process of calculating the current pose of the vehicle by using a previously determined pose and estimating that pose based upon known or estimated speeds over an elapsed time course. It is originally marine technology and now has been used in many fields, such as pedestrian dead reckoning, air navigation, autonomous navigation in robotics and automotive navigation.

DR can be implemented with different configurations. DR with inertial navigation systems (INSs) is widely used in localization and navigation on ground vehicles, spacecrafts, submarine vehicles and ships [132]. For automotive navigation system, the vehicle is often equipped with sensors that know the wheel diameter and record wheel rotations and steering direction. These sensors are present in a modern automobile for many purposes and can be read by the navigation system from the controller-area network (CAN) bus. Then the DR is developed with an appropriate vehicle kinematic model.

2.3.1 Inertial navigation systems

An inertial navigation system is a navigation unit that uses motion sensors and rotation sensors to continuously calculate via dead reckoning the position, orientation, and velocity of a moving object without the need of external references. The inertial measurement unit (IMU) is the main component of INS. IMUs typically contain three orthogonal rate-gyrometers and three orthogonal accelerometers, measuring angular velocity and linear acceleration respectively. It actuates as a sensor. INS uses an IMU to form a self-contained navigation system which tracks the position and orientation in three dimensions by processing signals form IMU. In the context of this writing, the term IMU is used to refer to expensive high-end IMUs.

INS usually works with high frequency. Most INS data rates exceed 50 Hz with some reaching into several hundreds of hertz. It is able to accurately measure the rapid changes of angular rotation rates and linear accelerations in short term. There are mainly two categories of INS: stable platform system and strapdown systems [132].

In stable platform type systems, the inertial sensors are mounted on a platform which is isolated from any external rotation motion. The platform is held in alignment with the navigation frame. In strapdown systems, the inertial sensors are mounted rigidly onto the device and therefore output quantities measured in the body frame rather than the navigation frame.

Stable platform and strapdown systems are both based on the same underlying principles. Strapdown systems have reduced mechanical complexity and tend to be physically smaller than stable platform systems. These benefits are achieved at the cost of increased computational complexity. As the cost of computation has decreased, strapdown systems using micro-machined electromechanical systems

(MEMS) devices have become the dominant type of INS. The widely used inertial sensors are gyrometers and accelerometers which are presented in the following.

2.3.1.1 Gyrometers

Gyrometers (gyros) measure the angular velocity of a system in one direction. There are many types of gyros, like mechanical, optical and MEMS gyros. Mechanical and optical gyros often require parts with high-precision tolerances and intricate assembly techniques and high part counts. As a result they remain expensive. In contrast MEMS sensors built using silicon micro-machining techniques have low part counts and are relatively cheap to manufacture. The major disadvantage of MEMS gyros is that they are currently far less accurate than optical devices. In [132], the errors which arise in MEMS gyros are examined. The main sources are outlined as follows.

White noise The output of a MEMS gyro is perturbed by some thermomechanical noise which fluctuates at a rate much greater than the sampling rate of the sensor. As a result the samples obtained from the sensor are perturbed by a white noise sequence, which is simply a sequence of zero-mean uncorrelated random variables. In this case, each random variable is identically distributed and has a finite variance. The noise introduces a zero-mean random walk error into the integrated signal that grows proportionally to the square root of time.

Constant bias This is the offset of the output from the true value. A constant bias error of ε, when integrated, cause an angular error which grows linearly with time. The bias wanders over time due to flicker noise in the electronics, the effects of which are usually observed at low frequencies in electronic components. At high frequencies flicker noise tends to be overshadowed by white noise. Bias fluctuations which arise due to flicker noise are usually modeled as a random walk. In reality, bias fluctuations do not behave as random walks (in this case, the uncertainty in the bias would grow unboundedly as the time-span increases). In practice the bias is constrained to be within some range, and therefore the random walk model is only a good approximation to the true process for short periods of time. Temperature fluctuations due to changes in the environment and sensor heating also induce movement in the bias.

Calibration errors This term refers to errors in the scale factors, alignments, and linearities of the gyros. Such errors tend to produce bias errors that are only observed whilst the device is turning. It is usually possible to measure and correct calibration errors.

Remark 1. The relative importance of each error source varies across different gyros. For MEMS gyros, white noise, and uncorrected bias errors either due to uncompensated temperature fluctuations or an error in the initial bias estimation are usually the most important sources of error.

More details about the notions of “white noise” and “random walk” will be introduced in Chapter 3.

2.3.2 Wheel speed sensors

Angular encoders are attached to the wheels of modern automobiles. these sensors basically measure the wheel speeds. They use a magnet and a coil of wire to generate a signal. The rotation of the wheel induces a magnetic field around the sensors. Wheel encoders detect angular rotation by measuring small increments of movement by observing this signal that varies as the wheel rotates. This sensor can become inaccurate at slow speeds. A wheel speed sensor only measures velocity in the forward direction. The integration of Wheel speed sensors in the navigation system applies to non-holonomic constraints on the lateral and vertical directions. These non-holonomic constraints are effective only when the vehicle operates on a flat road and when no side slip occurs [61].

In Europe, every new vehicle must be equipped with an Electric Stability Program (ESP) since January 2012, and Anti-lock Braking System (ABS) has been part of standard equipment in earlier times. These two systems contain a wealth of proprioceptive sensors which are capable of measuring vehicle rotation and individual wheel speeds. The sensors communicate with the electronic control units via CAN bus. These sensors are used as dead reckoning sensors.

2.3.3 Vehicle kinematic model

The performance of a positioning system depends not only on the filter structure but also on the appropriate choice of the process model for a filter. The roles of the process models in positioning systems have been investigated in [68].

Figure 2.3 illustrates the geometry of a car-like vehicle in a turning maneuver. ρ is the curvature radius at point M. e is the vehicle half-track. The simplest model consists in representing the vehicle by a single oriented point. Assuming that the vehicle has a 2D motion and that its speed and yaw rate can be measured, it conducts to the classical unicycle model.

$$\begin{cases} \dot{x} = v \cdot cos\psi \\ \dot{y} = v \cdot sin\psi \\ \dot{\psi} = \omega \end{cases} \tag{2.3}$$

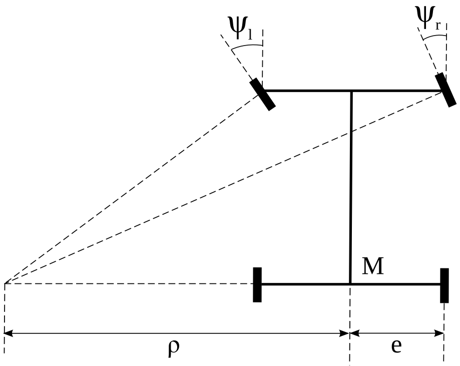


Figure 2.3: *Geometry of car in a turning maneuver*

where v and ω are the linear and angular velocity.

The available sensor measurements are $\left(v_{fl}^m, v_{fr}^m, v_{rl}^m, v_{rr}^m, \psi_L^m, \psi_R^m, \omega^m\right)$. where v_{fl}, v_{fr}, v_{rl}, and v_{rr} are the linear speeds of front left, front right, rear left and rear right wheels respectively. ω is the yaw rate of the vehicle. ψ_L and ψ_R are the steering angle of the front left and front right wheel. $(^m)$ means a measured value of a physical quantity. According to the Ackermann steering geometry, the sensor measurements satisfy the following relationship:

$$\begin{cases} v_{rl} = v - \omega.e \\ v_{rr} = v + \omega.e \\ v_{fl} \cdot cos\left(\psi_l\right) = v - \omega.e \\ v_{fr} \cdot cos\left(\psi_r\right) = v + \omega.e \end{cases} \tag{2.4}$$

The input for equation (2.3) is the linear velocity v and angular velocity ω. Therefore those measurements are redundant. Bonnifait, *et al.* presented data fusion of $(v_{fl}, v_{fr}, v_{rl}, v_{rr}, \psi_l, \psi_r)$ for outdoor localization based on Extended Kalman filter [22]. In this work, the yaw rate was not directly measured. The redundancy are used to reduce the effects of the unknown disturbances. In this thesis, considering that the yaw rate is directly measured by the gyro, only part of the measurements are used considering these low cost sensors have complex error characteristics. Our experimental vehicle is a front-wheel drive and front-wheel steering car. Under the assumption that the rear wheels don't slip, the speed vector is collinear with the x_M axis. The measured linear velocity v^m is calculated by:

$$v^m = \left(v_{rl}^m + v_{rr}^m\right)/2 \tag{2.5}$$

The measured yaw rate is a function of the real yaw rate of the vehicle and a bias

ε_ω, which is given by:

$$\omega^m = \omega + \varepsilon_\omega \tag{2.6}$$

An estimate of the pose $\mathbf{q}$ of the vehicle is provided by integrating these measurements from a known initial pose. Dead reckoning can be highly accurate over short time. However, a significant disadvantage of dead reckoning is that the errors of the process are cumulative, since new positions are calculated solely from previous positions. Therefore, the error in the pose grows with time. To overcome this, the navigation system needs to integrate sensors which can directly observe vehicle pose or part of it. GNSS receiver is an ideal choice.

2.4 Global navigation satellite systems

The term "global navigation satellite system" refers to a constellation of satellites providing signals from space transmitting positioning and timing data with global coverage. The basic operational idea of a GNSS is that receivers measure the time of arrival of satellite signals and compare it to the transmission time to calculate the signals propagation time. The propagation time is used to estimated the distances from the GNSS receiver to the satellites. The estimated distance is the so-called pseudorange. From the pseudorange estimates, GNSS receivers calculate position by means of multilateration which relies on multiple satellite measurements to produce a position fix. The accuracy of the position estimates is dependent on both the accuracy of the pseudorange measurements and the geometry of the satellites used in the multilateration.

At this time, there are several examples of GNSS which are fully or partially in operation: the USA's NAVSTAR Global Positioning System (GPS), Russia's Globalanaya Navigatsionnaya Sputnikovaya Sistema (GLONASS), Europe's Galileo positioning system and China's Beidou navigation satellite system (BDS).

The U.S. Department of Defense began the GPS project in 1973. It became fully operational in 1995. The GPS was initially developed to fulfill U.S. military needs. Now, it is a dual-use system that can be accessed by both military and civilian users. GLONASS was developed contemporaneously with GPS, but suffered from incomplete coverage of the globe until the mid-2000s. Galileo positioning system is currently being created by the European Union and the European Space Agency. The aim of this project is to provide an alternative high-precision global positioning system for European nations. Galileo will start offering first services from 2016. Full completion of the 30-satellite Galileo system is expected by 2020. China's BDS is still under construction and will be a global satellite navigation system consisting of 35 satellites. BDS became operational in China in December 2011, and began offering services to customers in Asia-Pacific region in December 2012. It is planned to begin serving global customers upon its completion in 2020.

Those GNSSs share some similarity: high availability, continuous, all-weather and near-real-time microwave technique with signals through the Earth's atmosphere. Since GPS is the only GNSS which is fully operational and widely used in the last decades, we overview the GNSS by taking GPS as an example.

2.4.1 Overview of the GPS

The GPS system is built around three main elements, i.e., space, control and user segments.

Space segment In order to provide a continuous global positioning capacity, a constellation with a sufficient number of satellites must be developed. The GPS space segment consists of a constellation of satellites in medium Earth orbit transmitting radio signals to users. The satellites are equally distributed over six orbital planes. This arrangement ensures a minimum visibility of 4 satellites at 15 degrees minimum elevation from virtually any point on Earth. In June 2011, the Air Force successfully completed a GPS constellation expansion known as the "Expandable 24" which may increase GPS performance. GPS now operates as a 27-slot constellation with improved coverage in most parts of the world (Figure 2.4). More details about the orbits, coverage and performance of the GPS satellite constellation are documented in the GPS Performance Standards [1].

The satellites provide a platform for equipment used to operate the system such as computers, atomic clocks and radio transceivers. All operating satellites are numbered with a space vehicle identifier (SV ID) and pseudorandom noise (PRN) number which uniquely identifies the ranging codes that a satellite uses. One to one correspondence between SV IDs and PRN numbers is fixed but the SV ID/PRN number of a satellite may be changed.

The main function of the space segment are to transmit radio navigation signals and the navigation message sent by the control segment. By using atomic clocks, the satellites are able to transmit signals with highly stable frequency. The GPS signal structure is detailed in section 2.4.2.1.

Control segment The GPS control segment consists of a global network of ground facilities that track the GPS satellites, monitor their transmissions, perform analyses, and send commands and data to the constellation. The current operational control segment includes a master control station (MCS), an alternate master control station, 12 command and control antennas, and 16 monitoring sites. The locations of these facilities are show in Figure 2.5.

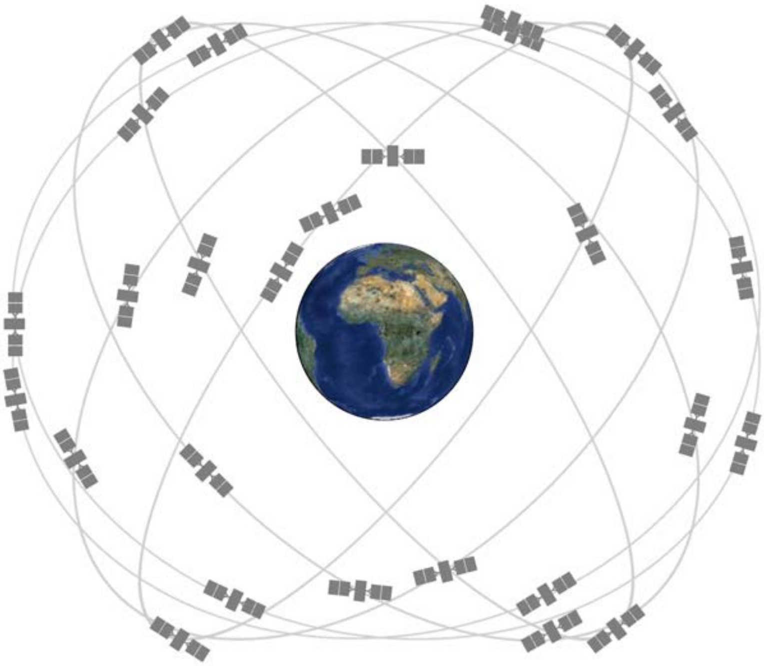


Figure 2.4: *GPS constellation. Source: [60]*

The ground antennas are used to communicate with the GPS satellites. Monitor stations track the GPS satellites as they pass overhead and collect atmospheric data, range/carrier measurements, and navigation signals. Monitor stations channel their observations back to the master control station in Colorado. The MCSs generate and upload navigation messages and ensure the health and accuracy of the satellite constellations.

User segment The GPS User Segment consists on L-band radio receivers and antennas which receive GPS signals, determine pseudoranges and other observables, and solve the navigation equations in order to obtain their coordinates and provide a very accurate time.

A GPS receiver is a device capable of determining the user position, velocity and precise time (PVT) by processing the signal broadcast by satellites. Receivers can be categorized by their type in different ways and under different criteria. For instance, receivers can be stand-alone, or may benefit from corrections or measurements provided by augmentation system or by receivers in the vicinity. Moreover receivers might be generic all purpose receivers or can be built specifically having the application in mind: navigation, accurate positioning or timing, surveying, etc. Low cost single-frequency receiver (L1-GPS) is often adopted for an application in the field of automobile. According to the GNSS Market Report [1] provided by European GNSS

---

[1] http://www.navipedia.net/index.php/GNSS_Market_Report

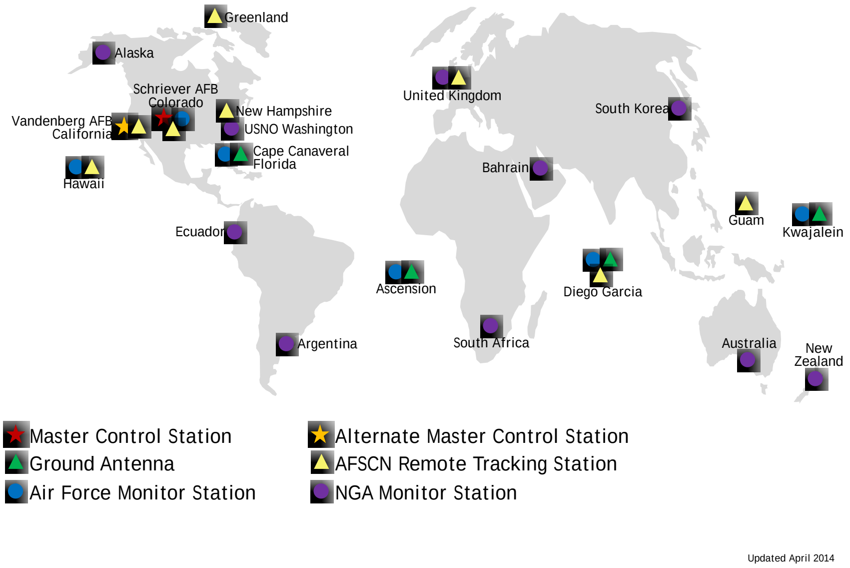


Figure 2.5: *The GPS Control Segment map. Source: [59]*

agency, the automotive industry agrees on the future view of a sensors fusion with GNSS as a core component, integrated into a car to provide enhanced positioning capabilities to be used with various innovative applications.

2.4.2 Generation of GPS observations

The GPS is a satellite-based positioning system. In order to decide user's position and velocity, information on satellites position, clock time and time of transmission of the signal etc. must be available to the user receiver. In practice, 6-12 satellites are visible simultaneously, depending on constellation geometry and elevation cut-off angle. GPS satellites broadcast continuously on 2 frequencies in the L-band. By using atomic clocks, the satellites are able to transmit signals with highly stable frequency.

2.4.2.1 Structure of GPS signal

GPS signals include ranging signals and navigation messages. The original GPS design contains two ranging codes: the coarse/acquisition (C/A) code and the restricted precision (P) code. Selective Availability (SA) is a purposeful degradation of the GPS signal by the U.S. Government that can be imposed to restrict the full accuracy of the GPS system to authorized military users. SA is part of the SPS, which was formally implemented on 25 March 1990. The degradation was accomplished primarily through dithering of the satellite clock. On May 1, 2000, SA was officially removed [121]. Hereafter, legacy GPS signal is reminded in terms of carriers, coarse acquisition code, precision code and navigation message.

Carriers The GPS satellites transmit signals on two carrier frequencies in L band, L1 and L2 [2]. They are centered at 1575.42 and 1227.6 MHz respectively. The atomic clocks aboard the satellite produces the fundamental L-band frequency (10.23 MHz). The L1 and L2 carrier frequencies are generated by multiplying the fundamental frequency by 154 and 120, respectively.

GPS supplies two services, Standard Positioning Service (SPS) and Precise Positioning Service (PPS). The SPS is a positioning and timing service provided on GPS L1 frequency and available to all GPS users. The L1 carrier contains a coarse acquisition code (C/A code) and a navigation message. The PPS is reserved for highly accurate military positioning, velocity, and timing service broadcast at the L1 and L2 frequencies. Both frequencies contain a precision code (P code) ranging signal with an encrypted navigation data message that is reserved for authorized users. Both C/A and P code are pseudo random code. The PRN codes and the navigation message are described as following.

Coarse acquisition code The coarse acquisition code modulates the L1 carrier with a chipping rate of 1.023 MHz and a sequence length of 1023 bits, using a bi-phase shift keying (BPSK) modulation technique.

The satellites transmit on the same L band frequency. Appropriate code division multiplexing techniques allow differentiating between the satellites. C/A code is designed to have high autocorrelation peak and low cross correlation peaks. Therefore, it can provide a wide dynamic range for signal acquisition. The C/A code is the basis for civilian GPS use.

Precision code Precision code is reserved for PPS. It consists of a unique sequence of 2.3547×10^{14} bits at 10.23 MHz. If the P code were allowed to continue without being reset, each P code would continue without repetition for more than 38 weeks. In practice, this overall period has been divided. Each of operational satellites or ground transmitters gets a one-week period code. Thus, an emitted P code will not overlap with that of any other satellite or ground transmitter. The P code is intended for military users and can be encrypted. When it's encrypted, it is called Y code.

The P code is difficult to acquire because of its length. But it has the advantage of not being ambiguous. A receiver correlator must be timed to within one P code chip (roughly 0.1 μs) and clocked in synchronicity in order to correlate at all. Many military receivers start by acquiring the C/A code and then move on to P code.

---

[2] As defined by the IEEE, the L band is the 1 to 2 GHz range of the radio spectrum

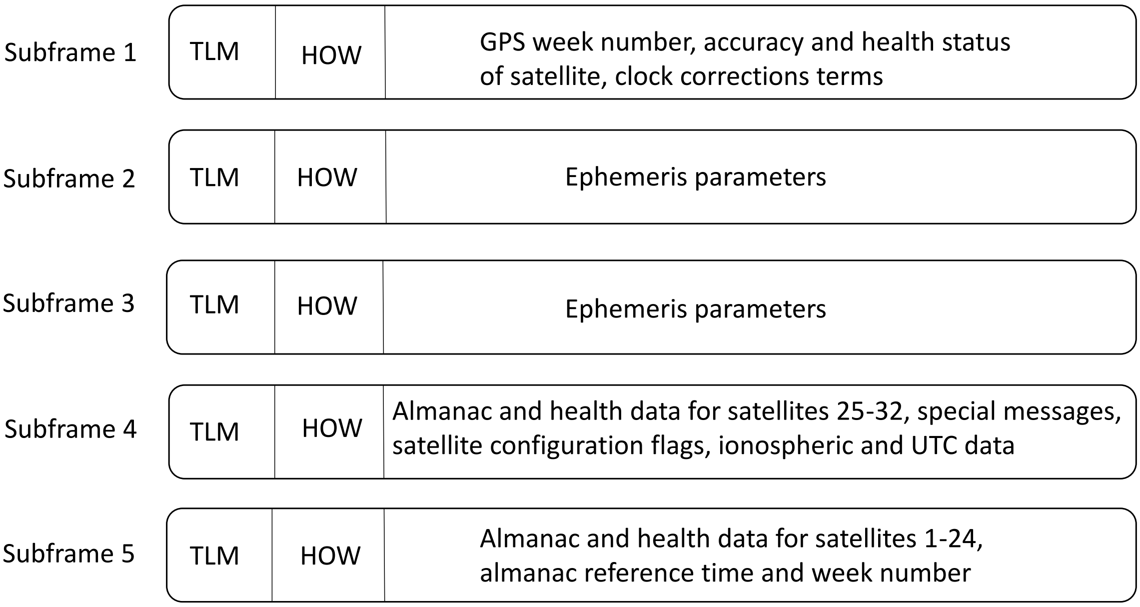


Figure 2.6: *Navigation message format. Source: [111].*

Navigation message As indicated in section 2.4.1, the navigation message is uploaded to each satellite by the GPS Control Segment for later broadcast to the user. It is modulated on both carriers. The navigation message has a cycle time of 30 seconds at 50 bps bit rate. So, the total navigation data set is 1500 bits and it is divided into 5 subframes of 6 seconds duration. Each subframe contains ten words. As shown in Figure 2.6, the first two words of each subframe are the telemetry (TLM) word and the C/A to P code handover word (HOW). The TLM word contains a synchronization pattern which is necessary for the receiver to access the navigation data. The HOW provides time information (seconds of the GPS week), allowing the receiver to acquire the week-long P code segment.

The first subframe contains the GPS week number, the satellite clock correction terms, and the satellite accuracy and health. The second and third subframes contain necessary ephemeris parameters for computation of satellite coordinates. The fourth and fifth subframes contain the almanac data with clock and ephemeris parameters for all available GPS satellites. They include also ionospheric correction parameters, UTC data, and particular alphanumeric information for authorized users. The ionospheric data from subframe 4 can be used to reduce the ionospheric effect, especially for L1-GPS receiver. However, subframes 4 and 5 are not repeated every 30 seconds. They consist of 25 pages that appear subsequently. The total information content is available after 12.5 minutes.

Now we summarize the process of generating GPS signals. The signals are driven by atomic clocks. The fundamental frequency is 10.23 MHz. Multiplying this frequency by 154 and 120, the L1 and L2 carriers are created. The navigation message is added to the C/A or P code respectively. Then the coded message is modulated to one

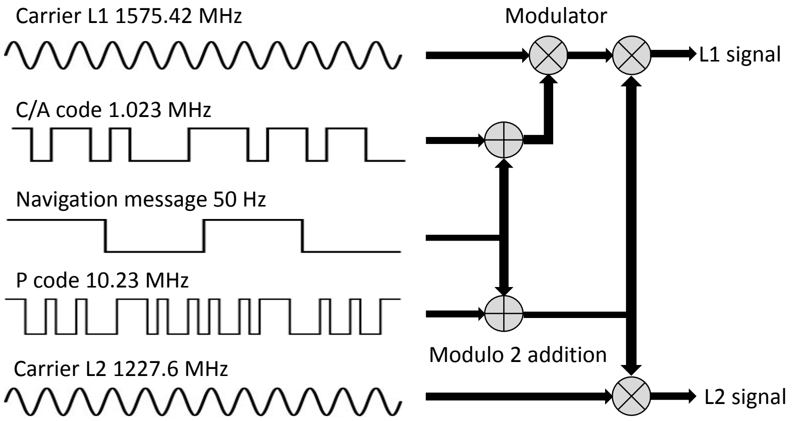


Figure 2.7: *GPS basic signals on L1/L2 carriers*

of carriers as depicted in Figure 2.7. The C/A code is only modulated onto the L1 carrier, while the P code is modulated onto both the L1 and L2 carriers.

New navigation signals designed for civilian use are being added to the satellite constellation, which are L2C, L5 and L1C[3]. However, users must upgrade their equipment to benefit from the new signals. The L1 C/A signal will continue broadcasting in the future.

2.4.2.2 Measure of time of flight

One of the fundamental observables of GPS system is the signal travel time between the center of the satellite transmitting antenna and the center of the receiver antenna. It is performed by the GPS receiver and then scaled into a range measurement using the signal propagation velocity. The conceptions which permit to construct the measure of time of flying on the PRN code are reminded.

In order to determine the time of flight, the receiver keeps a copy of the code sequence which is phase-shifted in time, i.e. the code must be generated within the receiver channel using the same algorithm that is utilized in the satellite. The replica code is correlated with the received code signal. Two sequences are shifted step by step in phase until maximum correlation is achieved. This process happens in one of the two tracking loops, which are equipped for each channel of the receiver:

- Delay lock loop (DLL): it aims at tracking and estimating the current misalignment between the locally generated PRN code replica and the incoming signal, within the tracking loops. In order to keep the local replica as “matched” as possible with the incoming signal, the DLL provides a correction of the current observed delay and this correction is applied to the local replica code generators.

---

[3] http://www.gps.gov/systems/gps/modernization/civilsignals/

- Phase Lock Loop (PLL): it tracks and estimates the current misalignment between the prompt correlator and the incoming signal phase, within the tracking loops. The demodulated satellite carrier phase signal is aligned with the phase signal of the receiver's oscillator. The carrier reconstruction technique only works on L2 when the P code is available, or for authorized users with access to the Y code. All GPS receivers must lock onto and track the signal's carrier to measure pseudoranges. A GPS receiver cannot distinguish one cycle of a carrier from another. The best a receiver can do is to measure the fractional phase and keep track of changes to the phase; the initial phase is ambiguous by an integer number of cycles [77]. To use the carrier phase as an observable for positioning, this unknown number of cycles must be estimated.

t_e is the system time when the signal left the satellite, and t_r is the system time when the signal would have reached the user receiver in the absence of errors. Thus, gradually shifting the signal, the receiver is able to estimate the time of flight $\triangle t$ as:

$$\triangle t = t_r - t_e$$

GPS is a one-way ranging system which means that a clock reading at the transmitter antenna is compared with a clock reading at the receiver antenna. Therefore, satellite and receiver clock offsets need to be taken into consideration. In the following sections, the index u denotes a receiver dependent parameter and the index s a satellite dependent parameter.

Effect related to clock offsets The satellites contain atomic clocks that control all on-board timing operations, including broadcast signal generation. Although these clocks are highly stable, the clock correction fields in the navigation data message are sized such that the deviation between satellite time and GPS time may be as large as 1 ms. The MCS determines and transmits clock correction parameters to the satellites for rebroadcast in the navigation message. However, some residual error remains. This residual clock error dt_s results in ranging errors that typically vary from 0.3-4m [70], depending on the type of satellite and age of the broadcast data. The clock offset of the receiver is denoted dt_u. For a given time t, time measurements of satellite and receiver clocks are:

$$t_s\left(t\right) = t + dt_s\left(t\right) \tag{2.7}$$

$$t_u\left(t\right) = t + dt_u\left(t\right) \tag{2.8}$$

These two time scales will help to establish the relationship between the true time of flight and the measured one.

Measured time of flight The satellite transmits the signal in its own time reference at time t_e, hence:

$$t_s\left(t_e\right) = t_e + dt_s\left(t_e\right) \tag{2.9}$$

The receiver performs the measurement at time t_r in its own time reference:

$$t_u\left(t_r\right) = t_r + dt_u\left(t_r\right) \tag{2.10}$$

The measured time of flight is:

$$\triangle t_m = t_u\left(t_r\right) - t_s\left(t_e\right) \tag{2.11}$$

Inserting equation (2.9) and (2.10) into equation (2.11), one obtains the measure time of flight in the form:

$$\triangle t_m = t_r - t_e + dt_u\left(t_r\right) - dt_s\left(t_e\right) \tag{2.12}$$

2.4.2.3 Pseudorange of the C/A code

The pseudorange ρ measured by the receiver is equivalent to the distance traveled by the GPS signal during its measured time of flight. Let c denotes the signal speed:

$$\rho = c\left(t_r - t_e\right) + c\left(dt_u\left(t_r\right) - dt_s\left(t_e\right)\right) \tag{2.13}$$

Let R denote the geometric distance between the satellite and the receiver such that:

$$R = c\left(t_r - t_e\right) = \left\| {}^{ECEF(t_r)}\mathbf{x}_u\left(t_r\right) - {}^{ECEF(t_r)}\mathbf{x}_s\left(t_e\right)\right\| \tag{2.14}$$

Where $\mathbf{x}_u$ is the receiver position vector in the ECEF frame related to the time of reception; $\mathbf{x}_s$ is the satellite position in the ECEF frame also related to the time of reception of the GPS signal. The pseudorange model is simplified to:

$$\rho = R + c\left(dt_u\left(t_r\right) - dt_s\left(t_e\right)\right) \tag{2.15}$$

Equation (2.15) is true only in the case that a signal evolves in a vacuum. However, the GPS signal has to pass through the Earth's atmosphere. The PRN code component of the satellite signal experiences delays as it propagates through the atmosphere, which elongates the pseudorange. The signal can also be delayed or advanced by other effects which are detailed in the following.

Estimation errors of broadcast ephemeris To resolve the receiver position from the pseudorange measurements, one must know satellite position and satellite clock offsets which are provided in the navigation message. The broadcast ephemeris are generated in two steps [111]. First, a reference ephemeris is generated based on several days of observations from the monitor stations. Second, corrections to the reference ephemeris are predicted by processing the discrepancies between the reference ephemeris and the current observations at the monitor stations. Estimates of ephemeris for all satellites are computed and uploaded to the satellites with other navigation data message parameters for broadcast to the user. For each instant, it is possible to estimate the satellite position ${}^{ECEF}\mathbf{x}_s$ by interpolation. The typical magnitude of the residual error is in the range of 1-6m [70].

The residual satellite position error can be divided into components in 3 directions: the radial direction (from the satellite toward the Earth center), the along-track direction (the instantaneous direction of travel of the satellite), and cross-track direction (perpendicular to the along-track and radial direction). Although the along-track and cross-track components are more difficult for the Control Segment to observer since its monitoring stations are on the surface of the Earth, these error components don't project significantly onto the satellite-to-receiver line of sight (LOS) vector. The radial error is the smallest component and it can be well monitored by the Control Segment. The effective pseudorange error δ_{X_s} due to ephemeris prediction errors is on the order of 0.8m [70].

Atmospheric effects The atmosphere is a dispersive medium. Its effects are smallest when the satellite is directly overhead since the path through the atmosphere is the shortest. There are two layers of the atmosphere which primarily affect the GPS signal: the ionosphere and troposphere. Their effects are recalled as follows.

Ionospheric effects The ionosphere is a dispersive medium located primarily in the region of the atmosphere between about 70 km and 1000 km above the Earth's surface. Within this region, ultraviolet rays from the sun ionize a portion of gas molecules and release free electrons. These free electrons influence electromagnetic wave propagation, including the GPS satellite signal broadcasts. The crossing of ionosphere causes a slower group velocity, which elongates pseudorange measurement of the PRN code.

Tropospheric delay The troposphere is the lower part of the atmosphere that is non-dispersive for frequencies up to 15 GHz. Within this medium, the phase and group velocities associated with the GPS carrier and signal information (PRN code and navigation data) on both L1 and L2 are equally delayed with respect to free space propagation.

The atmospheric effects are the main error sources in determining the receiver position. They are related to the environment between the receiver and the satellite transmitter and are therefore impossible to be accurately estimated by the Control Segment.

Relativistic effects Relativistic effects are due to the fact that there is a difference in speed between the satellite and the receiver. It results in a dilatation of time and a frequency offset. The relativistic effects can reach a maximum of 70ns [111]. Correcting the satellite clock for this relativistic effect results in a more accurate estimation of the time of transmission by the user.

Complete model of the pseudorange of the C/A code When taking into account the aforementioned degradation, the complete model of the pseudorange is presented as follows.

$$\rho = R + c\left(dt_u\left(t_r\right) - dt_s\left(t_e\right)\right) + \delta_{X_s} + \delta_{iono} + \delta_{tropo} + \delta_{rel}$$

where δ_{X_s} is the error due to the use of broadcast ephemeris, δ_{iono} and δ_{tropo} represent respectively the range equivalents of the delays caused by the ionospheric and tropospheric effects, and δ_{rel} represents the relativistic effects.

With the navigation message, the GPS receiver is able to know or estimate part of the degradation. $dt_s\left(t_e\right)$, δ_{X_s}, δ_{iono} and δ_{rel} are able to be estimated by using classical atmosphere models implemented in an open source software like the GPSTk [56]. Their estimations are denoted as $\hat{dt}_s\left(t_e\right)$, $\hat{\delta}_{X_s}$, $\hat{\delta}_{iono}$ and $\hat{\delta}_{rel}$ respectively.

The pseudorange measurement after applying these corrections is denoted ρ_c.

$$\rho_c = \rho + \hat{dt}_s\left(t_e\right) - \hat{\delta}_{X_s} - \hat{\delta}_{iono} - \hat{\delta}_{rel} = R + c \cdot dt_u\left(t_r\right) + \varepsilon_{pr} + \beta \tag{2.16}$$

where ε_{pr} is the residual error and β is the measurement noise.

2.4.2.4 Doppler shift

The Doppler shift is produced by relative motion between the satellite and the receiver. Doppler measurement is available in all GPS receivers as necessary to compensate the Doppler shift to perform the pseudorange measurements. The Doppler shift measurement has the advantage of being insensitive to atmospheric effects and accurate [90]. At the receiver antenna, the received frequency, f_r, can be approximated by the classical Doppler equation as follows [70]:

$$f_r = f_e\left(1 - \frac{\mathbf{v}_r \bullet \mathbf{u}_{los}}{c}\right) \tag{2.17}$$

where f_e is the satellite signal's emission frequency, $\mathbf{v}_r$ is the satellite-to-user relative velocity vector, $\mathbf{u}_{los}$ is the unit vector pointing along the line of sight from the user to the satellite, and c is the speed of propagation. $\bullet$ represents the dot product. $\mathbf{v}_r \bullet \mathbf{u}_{los}$ leads to the radical component of the relative velocity vector along the line of sight to the satellite.

Let $\mathbf{v}_s$ and $\mathbf{v}_u$ denote the velocity of the satellite and of the user respectively, we get:

$$\mathbf{v}_r = \mathbf{v}_s - \mathbf{v}_u \tag{2.18}$$

The Doppler offset due to the relative motion is obtained form these relation as

$$\triangle f = f_r - f_e = -f_e \cdot \frac{(\mathbf{v}_s - \mathbf{v}_u) \bullet u_{los}}{c} \tag{2.19}$$

The Doppler measurement is biased by clock drifts. Suppose $\dot{dt}_u$ is the receiver clock drift and $\dot{dt}_s$ the satellite clock drift. Let denote δf the bias on the Doppler shift due to the clocks drifts:

$$\delta f = f_e\left(\dot{dt}_u - \dot{dt}_s\right) \tag{2.20}$$

$$\triangle f = -f_e \cdot \frac{(\mathbf{v}_s - \mathbf{v}_u) \bullet \mathbf{u}_{los}}{c} - \delta f \tag{2.21}$$

Let $\dot{\rho}$ denotes the Doppler measurement:

$$\dot{\rho} = -c \cdot \frac{\triangle f}{f_e} \tag{2.22}$$

This allows to write the Doppler measurement in a form similar to pseudorange:

$$\dot{\rho} = (\mathbf{v}_s - \mathbf{v}_u) \bullet \mathbf{u}_{los} + c\left(\dot{dt}_u - \dot{dt}_s\right)$$

At the GPS L1 frequency, the maximum Doppler frequency for a stationary user on the Earth is approximately 4 kHz, corresponding to a maximum line-of-sight velocity of approximately 800 m/s.

Since the satellite clock drift is estimated by the control segment and can be reconstructed from the broadcast navigation message, the corrected Doppler shift $\dot{\rho}_c$ is presented as follows:

$$\dot{\rho}_c = \dot{\rho} - c \cdot \hat{\dot{dt}}_s$$

$$\dot{\rho}_c = (\mathbf{v}_s - \mathbf{v}_u) \bullet \mathbf{u}_{los} + c \cdot \dot{dt}_u \tag{2.23}$$

If we take now into consideration the variation of the pseudorange error ε_{pr}, equation (2.23) becomes:

$$\dot{\rho}_c = (\mathbf{v}_s - \mathbf{v}_u) \bullet \mathbf{u}_{los} + c \cdot \dot{dt}_u + \dot{\varepsilon_{pr}} + \beta_d \tag{2.24}$$

where β_d is the measurement noise.

2.4.2.5 GPS position fixes

By convention, the receiver provides positions corresponding to its antenna. In the early 1980s, the National Marine Electronics Association (NMEA) developed the NMEA 0183 standard for an interface standard to permit reliable data communication among electronic marine instruments [76]. This standard is adopted by most GPS manufacturers. The data includes the complete PVT (position, velocity, time) solution computed by the GPS receiver. Every GPS fix (latitude, longitude and altitude) is converted in the working frame. If the GPS antenna is located at point M (see Figure 2.8), we have:

$$\begin{cases} x_{GPS} & = x \\ y_{GPS} & = y \end{cases} \tag{2.25}$$

In practice, the antenna is often translated with a lever arm (see Figure 2.8).

The homogeneous matrix expressing the frame M in the local frame is:

$${}^{O}\mathbf{T}_M = \begin{bmatrix} \cos\psi & -\sin\psi & x \\ \sin\psi & \cos\psi & y \\ 0 & 0 & 1 \end{bmatrix}$$

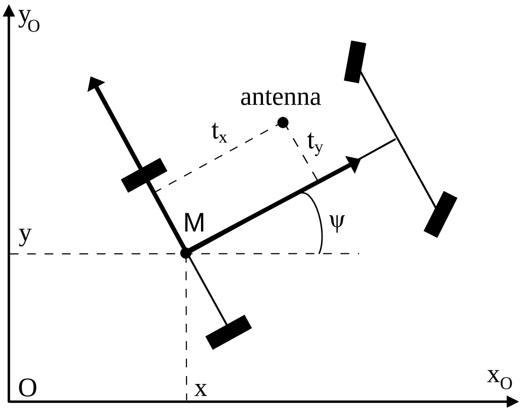


Figure 2.8: *GPS antenna coordinates*

The antenna position is the body frame is:

$$^{M}\mathbf{g} = \begin{bmatrix} t_x \\ t_y \\ 1 \end{bmatrix}$$

The antenna position in the local frame is:

$$^{O}\mathbf{G} = {}^{O}T_M . {}^{M}\mathbf{g}$$

Which gives:

$$\begin{cases} x_{GPS} = \cos\psi \cdot t_x - \sin\psi \cdot t_y + x \\ y_{GPS} = \sin\psi \cdot t_x + \cos\psi \cdot t_y + y \end{cases} \tag{2.26}$$

For a low-cost GPS receiver operating in standalone mode, only L1 frequency C/A code measurements are available to measure pseudoranges. They provide positioning service with several meters accuracy. In order to improve accuracy, differential technique is often used.

2.4.3 Differential GPS

Differential GPS (DGPS) is an enhancement to GPS that provides improved positioning or timing performance. It uses one or a network of fixed, ground-based reference stations whose positions are accurately known. Each station is equipped with at least one GPS receiver, and broadcast the difference between its GPS observations and internally computed observations. The user's receiver can correct their GPS observations according to the broadcast data. The DGPS techniques may be categorized in different ways: as absolute or relative differential positioning; as local area, regional area, or wide area; and as code based or carrier based [70]. In this thesis, we refer to DGPS as code based.

The GPS error sources discussed in section 2.4.2 are highly correlated over space and time. The underlying premise of DGPS is that any two receivers that are relatively close together will experience similar errors. Satellite clock errors vary slowly with time and it is regardless of the location of the user. The pseudorange error, induced by broadcast ephemeris, depends the LOS between the satellite and the receiver. The LOSs of different receivers in close proximity to the same satellite are very similar. Therefore, the difference in pseudorange errors of close receivers is very small. In addition, the broadcast ephemeris errors change very slowly over time.

A similar system that transmits corrections from orbiting satellites instead of ground-based transmitters is called a Wide-Area DGPS (WADGPS) or Satellite Based Augmentation System (SBAS). Such systems use accurately-surveyed monitoring stations to produce differential error corrections and integrity messages which are broadcast to the users using geostationary earth orbiting satellites. Various SBASs are implemented, such as the Wide Area Aviation System (WAAS) operated by the United States Federal Aviation Administration and the European Geostationary Navigation Overlay Service (EGNOS) operated by European Satellite Services Provider. SBASs reach more users but ground-based systems enable quicker message transmission and more accurate differential corrections.

By the early 1990s, researchers had shown that is was possible to improve on the accuracy of DGPS by transmitting carrier-phase data to user receivers. This technique permits real-time highly accurate positioning even if the receivers are moving, thus is named real-time kinematic (RTK).

2.4.4 RTK-GPS

RTK-GPS is a differential GPS technique used to enhance positioning performance. Its origin dates back to the mid-1990s. It uses measurements of the phase of the signal's carrier wave, and relies on a single reference station or interpolated virtual station to provide real-time corrections, providing up to centimeter-level accuracy. The key feature enabling the high accuracy afforded by RTK operation is the ability to determine the carrier-phase integer ambiguities while the user's receiver is in motion.

A RTK base station covers a service area spreading about 10 or 20 kilometers. The systematic errors grows with the distance between the base station and the user's receiver (baseline length). In [125], virtual reference stations (VRS) simulate a local reference station nearby the user receiver. It allows a more precise modeling of distance-dependent systematic errors and performing RTK positioning in reference station networks with distances of up to 40 km. In 1985, the Radio Technical Commission for Maritime Services (RTCM) suggested a standard format for coding and transmitting such correction, i.e., the RTCM format, which remains the mostly widely used.

In RTK applications, a discontinuity of the integer number of cycles may occur because of GPS signals blockage, communications outages or constellation changes. Consequently, the integer counter is reinitialized. The speed with which ambiguities can be fixed again depends on several factors including the number of satellites tracked, satellite-receiver geometry, use of pseudorange data in addition to carrier phase, observation noise, and use of dual-frequency observations [77]. Under good conditions, fix times can be shorter than one minute and optimally, less than 10 seconds.

As one of the most accurate positioning technologies, the RTK GNSS system is employed as ground truth system in this thesis.

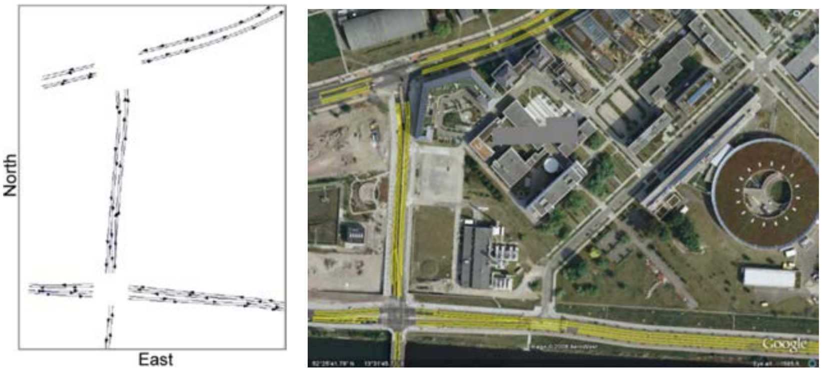
2.4.5 Precise point positioning

Precise point positioning (PPP) uses ambiguous carrier-phase measurements but only from the user's receiver. It needs highly accurate satellite orbit and clock information such as that provided by the international GNSS Service. PPP is able to provide few centimeter-level accuracy in static mode and decimeter-level accuracy in kinematic mode. The convergence period required from a cold start to a decimeter-level position solution, is typically about 30 minutes under normal conditions. There is also the need to have uninterrupted GNSS signal availability for PPP. Loss of tracking lock on a minimum number of satellites requires processing filter reinitialization. The results are poorer than decimeter-level in accuracy until filter reconvergence. PPP is a viable option wherever accurate positioning is required and station infrastructure is not available but open sky conditions of visibility are required in practice.

2.5 Digital maps

2.5.1 Enhanced maps for autonomous driving

Most car navigation maps represent roads with polylines which are defined by a series of points and connected by segments. OpenStreetMap (OSM) is a typical representative. There are three basic primitives in its data model: nodes, ways and relations [17]. Nodes are points in space. A node can be used as a junction between two ways or just a change in direction of a way. Ways are ordered list of nodes. They can describe linear features such as roads and waterways. They can also be closed



(a) *Emap of the same area* **(b)** *Emap of the test site superimposed on the Google Maps image of the area*

Figure 2.9: *Enhanced lane-level map. Source: [19]*

to form areas such as buildings. Relations are used to model features that can't be described using a single node or way such as branching roads, long distance routes or turn restrictions at junctions. Those typical car navigation maps provide mostly topological information but with too limited geometric information and accuracy that are not enough for lane-level navigation.

Bétaille *et al.* suggested an enhanced map model which provides a comprehensive description of the road [19]. Each carriageway and each lane are described (cf. Figure 2.9a). Clothoids are exploited to follow the shape of the lane centerline. The authors removed the crossroads considering they allow too many possible trajectories. In addition to the geometry description, every road lane is topologically connected to the rest of lanes. The proposed map model is applied in [122] for lane-level navigation.

At the DARPA Urban Challenge final event, teams were provided a map in Route Network Definition File (RNDF) format [35]. With this format, roads are split into segments comprising one or more lanes. The lanes are approximated by a set of waypoints as well as an optional width parameter. Some of the waypoints are labeled as entry or exit waypoints. In addition to lanes, zones are defined to represent a freely driveable area approximated by a bounding polygon. Figure 2.10 shows a scenario of RNDF map. As we have mentioned, the map is not only useful for path planning but also helps in environment perception. Huang et al. propose a multi-sensor lane finding system in the context of DARPA Urban Challenge. The RNDF was treated as a strong prior on the type and number of lanes, and a weak prior on their position and geometry. The lane detection and tracking system guided their vehicle through a 90 km course at speeds up to 40 km/h as reported in [58].

RNDF was created specifically for the artificial road network for Urban Challenge. It was enhanced to model real world road networks in [34]. The enhanced format

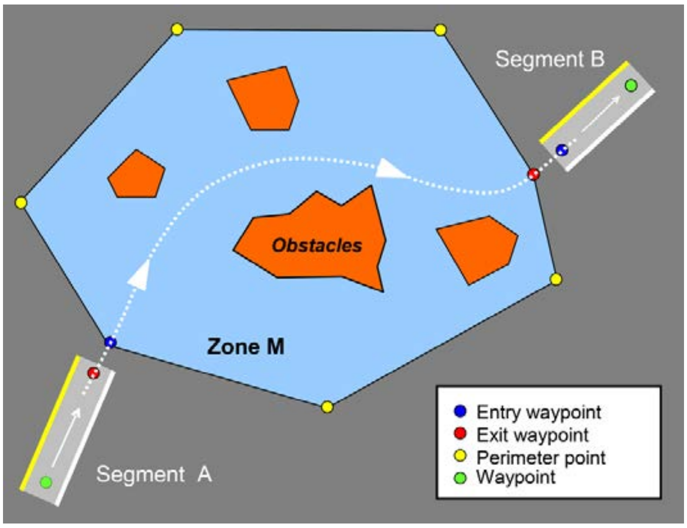


Figure 2.10: *A scenario of RNDF map. The yellow, red and blue dots that create the polygonal zone are perimeter points. Some of which are also entry and exit points to adjacent segments. Connections between lanes are thereby represented. The dotted line is a possible travel path. According to [35].*

is called RNDFGraph. The authors claimed to have tested the map in autonomous driving for hundreds of kilometers under real traffic conditions.

Recently Bender *et al.* presented lanelets [16] as a map format for autonomous vehicles. A lanelet describes a lane segment which is characterized by its left and right bound 2.11. The bounds are polylines and therefore allow for an approximation of lane geometries. Bidirectional lanes are modeled as two lanelets and shared bounds are represented by just one way element. Traffic rules like merge and cross behavior are introduced on lanelets. The proposed map was employed for autonomous driving on the 104 km long Bertha Benz Memorial Route in 2013 [134]. About two thousand lanelets were annotated manually to map the test route.

In addition to the topology and geometry information in the navigation map for autonomous vehicle, physical property of the road surface such as reflectivity is also proposed to be mapped. In [82], the resulting map is an overhead image of the road surface, where the image brightness corresponds to the infrared reflectivity. The map is a 2D grid which assigns to each x-y location in the environment an infrared reflectivity value. [124] stores location-tagged images as a map database. This map is used to perform visual localization which means identifying the location of a query image by comparison with images in the map.

In conclusion, we refer to maps for ARV of two types:

- Road network map which uses the waypoints of road and lane to define the metric and topological relationships of the road network. Traffic rules are also assigned to each road and lane.

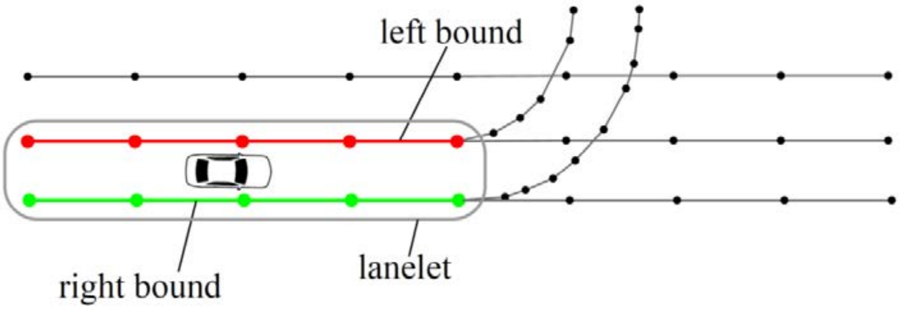


Figure 2.11: *This image highlights a single lanelet. The roles of the bounds specify the driving direction. Black/green/red dots, bounds and lanelets correspond to nodes, ways, and relations defined in OSM respectively. The authors propose to use JOSM (an OSM editor) [17] to create and edit the lanelet map. The image is according to [35].*

- Environment map which is mainly feature-based such as maps of lane marking, traffic sign and road surface reflecticity.

Recently, the generation of enhanced maps has achieved significant progress. In the following, some mapping techniques are briefly introduced.

2.5.2 Mapping techniques for enhanced maps

Nowadays, the main method for building a detailed map is mobile mapping. Mobile mapping is the process of collecting geospatial data from a vehicle, typically equipped with a range of video cameras, radars, Lidar or any number of remote sensing systems. The mobile vehicle is accurately localized mainly using a high performance localization system (i.e. RTK-GPS coupled with IMU and vehicle odometry). Post processing to refine the mobile mapping platform's trajectory is often needed. All sensors data are time-stamped and fused to georeferencing the road features such as lane markings and traffic signs.

The enhanced map in [19] is built by a mobile mapping method. The equipped vehicle is localized at decimeter-level by post processing and needs to drive as close as possible to the lane centerlines. Later approaches resort to vision sensors and coarse prior map to simplify the mapping process [**?**][54]. Vision sensors are used to capture road features, such as lane markings. All roads have different types of lane markings which are painted in a symmetrical manner and follow national standards [3]. Lane markings have similar properties with lane centerlines. They can be simplified to polylines or clothoids having a limited number of shape points. The vehicle doesn't need to follow the centerline of the road during the mobile mapping of the lane markings. Coarse prior maps, such as Google Maps (Figure 2.9b) and OSM, provide strong priors on the existence of roads and intersections. [**?**] proposes to use OSM and accurately mobile mapped lane markings with Lidar to generate lane-level maps. Similarly, [54] exploits the road network defined in

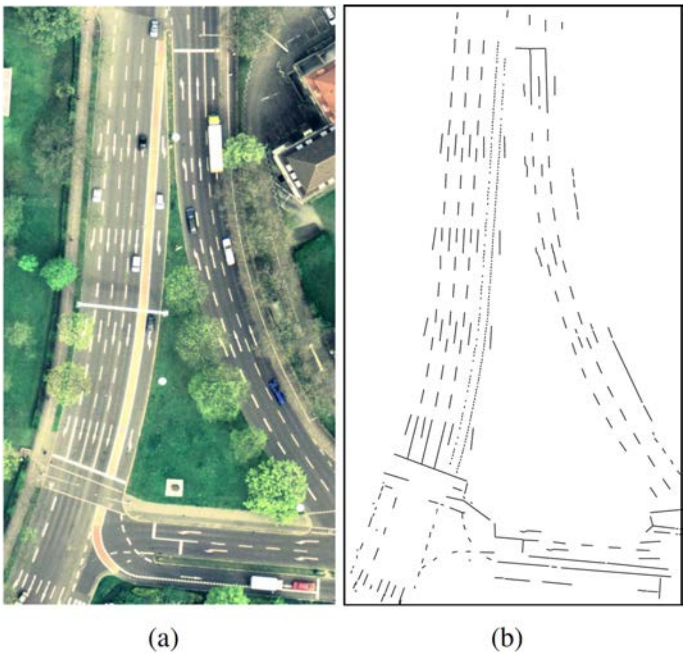

(a) (b)

Figure 2.12: *An example of the aerial image and the corresponding feature (lane marking) map used in [101]. The aerial images have a resolution of 10 cm per pixel. The resulting map is 2-dimensional. All detected lane markings are assumed to lie in one ground plane.*

OSM to generate road orthographic images and a lane-level map with conventional low-cost automotive sensors. Particularly, they proposed a method for building the lane graph in the intersections according to the traffic rules. Both approaches are automatic.

Mobile mapping for large areas is very time-consuming. An alternative is the use of aerial images and visual tools to obtain the mapping points. [102] describes a method of automated feature map generation from aerial images. Figure 2.12 displays the aerial image and the generated lane marking map. This map was used for vehicle localization and most of the absolute position errors are within 1 m as reported in [101].

A new trend of mapping is crowdsourcing. OpenStreetMap is probably the most prominent crowdsourcing project for geospatial data based on the ideas of the open source initiative. It has grown to over 2 million registered users who collect data using GPS devices, manual survey, aerial photography and other free sources [97]. However, it is effective mainly for building maps of road network. As automobile's sensing ability increases, crowdsourcing sensory data from customer road vehicles is a very promising low-cost solution to build enhanced maps for autonomous driving at world-wide scale [116].

2.5.3 Map matching

The vehicle is related to the maps through a map matching process. Considering the different kinds of maps, we classify the map matching in the literature into two categories as follows.

Map matching in car navigation system for human drivers In this context, the general purpose of a map matching algorithm is to identify the correct road segment on which the vehicle is traveling and to determine the vehicle location on that segment [104]. Usually, determining which road the vehicle is traveling on is enough for the path planning task of the navigation system. Since the direct overlay of positional data doesn't reconcile with a digital map, advanced map matching algorithms have been developed using different techniques such as topological analysis of spatial road network data, probabilistic theory, Kalman filter [72], fuzzy logic [103], belief theory [91] and interval analysis [94]. Topological reasoning and a weighting scheme are performed to match a position with a road segment. The weighting scheme usually relies on proximity of orientation and distance. These methods share the same idea, i.e., projecting the vehicle position onto the centerline of the roads or the lanes. The position is often provided by a GNSS sensor or coupled with a dead reckoning system. Those algorithms mainly deal with the road ambiguity problems in at cross roads. The road map accuracy dominates the positioning accuracy. Therefore, map matching can make an estimation of the positioning bias (even if the map has an overall shift, the degraded accuracy will not influence the route guidance too much).

Map matching in the context of robotics In this context, a vision sensor creates a local map of the surrounding environment. Then the local map is correlated with the global feature map to localize the vehicle. This strategy has been often used in indoor positioning [123][10], since there is no GNSS signal. The correlation is also made by a map matching procedure. When used in outdoor localization for autonomous vehicles, this matching process is usually performed using the iterative closest point (ICP) algorithm [18][133][128]. An initial estimate of the relative pose is often required [82][101], in order to prevent the result from converging to a local minimum. In this thesis, we refer to the map matching in this context as data association to distinguish it with the map matching in car navigation system for a human driver. For autonomous vehicle, the data association is usually performed to estimate the vehicle pose in a specific lane.

2.5.4 Lane marking map used in this thesis

The lane markings of the test area have been surveyed by Viametris [4] which is a French company providing mobile mapping solutions. The mobile mapping system

---

[4] http://www.viametris.com/

Figure 2.13: *Mobile mapping platform utilized by Viametris*

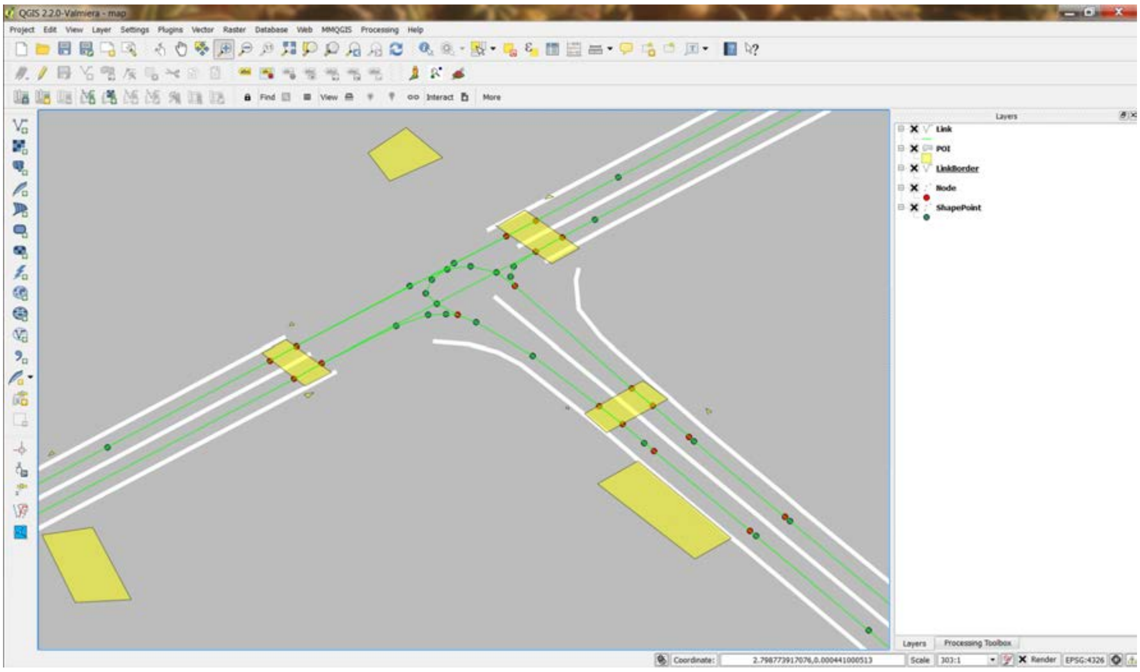
is equipped with a RIEGL VQ450 lidar with millimeter-level accuracy. The mobile mapping platform (Figure 2.13) itself is localized using post-processed kinematic (PPK). The absolute error of the collected lane marking data is estimated to be 5-10cm.

The lane markings are stored in a map which is managed as a geographical database with the open source SpatiaLite library. SpatiaLite extends the SQLite core to support spatial SQL capabilities. The map database can be managed with the Quantum GIS (QGIS) software which is a cross-platform free and open-source desktop geographic information system (GIS) application that provides data viewing, editing and analysis. As shown in Figure 2.14, the map contains three main layers as follows:

- *links.* This layer contains the lane centerlines (green polylines). These are described in terms of connectivity by a set of nodes (red points) and in geometry by a set of shape points (green points).
- *points of interest.* These points or areas represent every road contextual static features (yellow areas). They can be for example pedestrian crossings, traffic signs, bumpers, battery charging areas and driver pick-up points.
- *link border.* This map layer contains the road lane markings (white segments).

The types of the lane markings are also defined in the map. Table 2.1 lists the lane marking types. Figure 2.15 shows the specifications of dashed markings in France.

In order to facilitate the map matching process, another layer is added to the map: a *road* layer. This layer contains the carriageway centerlines expressed by a set of polylines which are also described in terms of connectivity by a set of nodes and



Coordinate: 2.798773917076,0.000441000513
Scale 303:1
EPSG:4326

Figure 2.14: *The lane marking map with QGIS interface*

type	solid	dashed							*pavement*	*barrier*	*other*
	T0	*T1*	*T'1*	*T2*	*T'2*	*T3*	*T'3*	*T4*			
index	1	2	3	4	5	6	7	8	9	10	11-15

Table 2.1: *Lane marking types in the map*

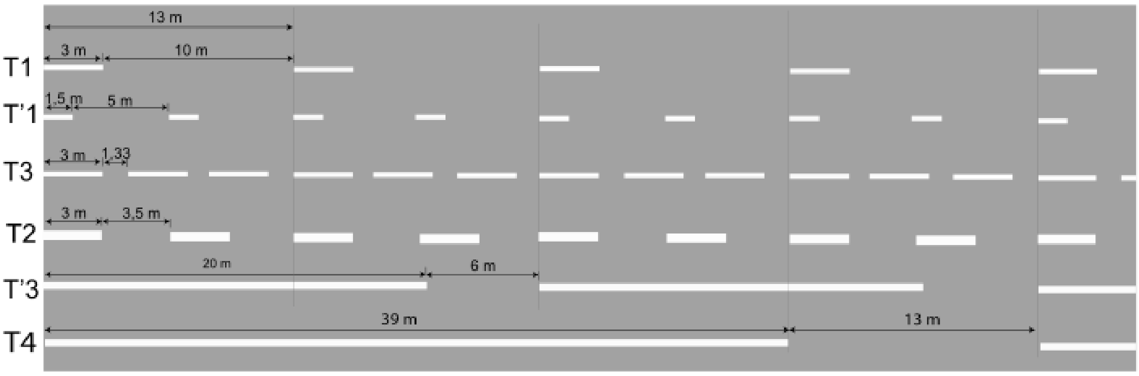


Figure 2.15: *Types of longitudinal dashed markings on pavements in France*

in geometry by a set of shape points as the *links*. For each carriageway, denoted R^i with $i = 1, 2, ..., n_r$ (n_r being the total number of roads), the number of lanes and *link*s contained in this carriageway is defined. The *road* layer has been added manually using QGIS.

The map structure used in this thesis is illustrated in Figure 2.16. The *link*s related to each carriageway are found by sampling the *road* polyline. The lane markings (*link border*s) on both sides are spatially related to each lane polyline (*link*).

There are two main purposes of defining the *road* layer. The first one is to facilitate the definition of a road-oriented frame (see section 2.2.2). The road-oriented frame will be used in Chapter 4. The second purpose is to assist the definition of topologically relations between lanes which are not defined in the original map. For example, if the vehicle is initially localized in lane 1 of *road* m, when the positioning uncertainty grows, the vehicle can be aware of that there is lane 2 which is spatially adjacent to lane 1 and the vehicle could possibly change from lane 1 to lane 2. The definition of this topological relation is realized by relating them to the same *road* segment. This topological relation is used to monitor the ambiguities between lanes in Chapter 6 when performing a multi-hypothesis lane-level localization. In addition, the lanes belonging to different carriageways inherit the topological relations of *road*s. For example the vehicle on lane 1 of *road* m-1 is possible to travel to lane 1 or 2 of *road* m at the next moment. But it is not possible to travel directly to lane 1 or 2 of *road* m+1 even if the two lanes (*link*s) are spatially very close.

Figure 2.17 shows the complete map expressed in the working frame R_O. The blue, green and white lines of the zoomed part represent respectively the carriageway centerlines, the lane centerlines and the lane markings.

2.6 Lane detection camera

For an autonomous vehicle, the environment perceiving tasks can vary from basic functions like lane marking detection, pedestrian and moving object recognition, to a sophisticated high dimensional world representation for traffic scene understanding. For example in [47], the authors proposed a probabilistic generative model to reason with the 3D scene layout as well as the location and orientation of objects in the scene. The scene topology, geometry and traffic activities are then inferred from video sequences.

The involved sensing technologies for environment perception mainly consist of camera, radar (radio detection and ranging), ultrasonic, and lidar (light detection and ranging). Cameras are appealing sensors as they are cheap and readily available in modern automobiles. Figure 2.18 displays four main application scenarios of automotive cameras: front, rear, surround and interior view [15]. In April 2014,

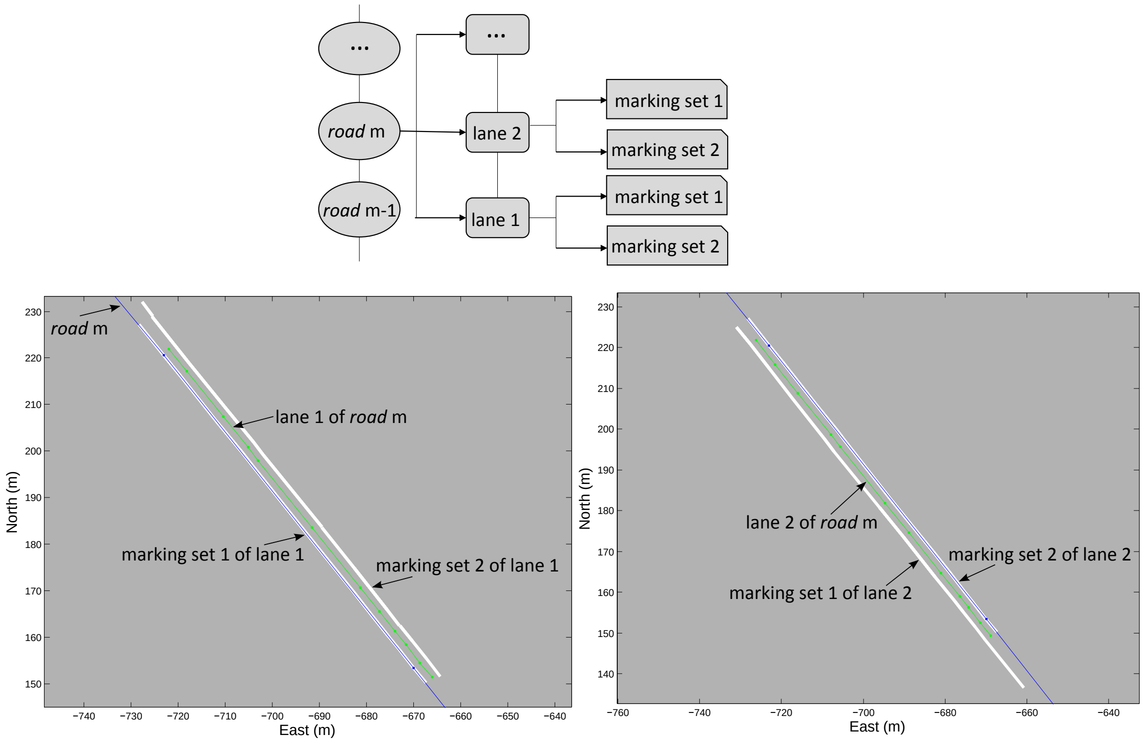


Figure 2.16: *Relationships between different layers of the map. Roads are connected through their common nodes. White lines are lane markings. Green lines represent the lane centerlines. Blue lines are carriageway centerlines.*

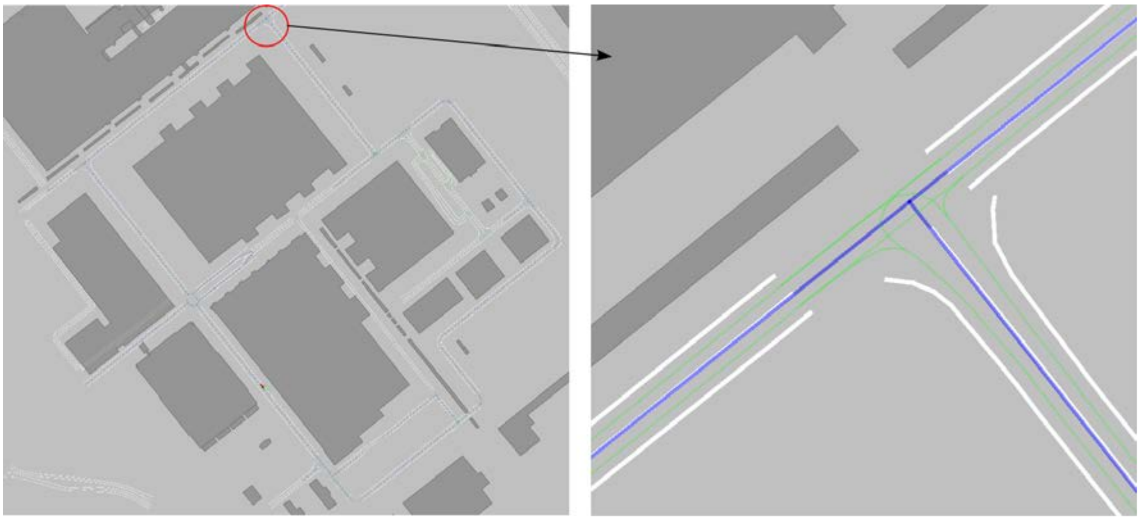


Figure 2.17: *Map used in this research. The different layers are plotted on the right image. The bounded areas in gray are buildings imported from OpenStreetMap to show the urban condition of the navigation area. The visualization of the map is managed using QGraphicsScene from the Qt library (http://www.qt.io/).*

(a) *Front view with lane and object detection for longitudinal control, lane keeping, collision mitigation, collision avoidance, etc. Image: [46].*

(b) *Rear view for parking aid and collision avoidance*

(c) *Surround view for parking and maneuvering aid. Image: [55]*

(d) *Interior view for driver drowsiness monitoring*

Figure 2.18: *Application areas of automotive cameras*

the US Department of Transportation's National High Traffic Safety Administration (NHTSA) issued a proposed rule to require all light vehicle to have rear view cameras in the US by the year 2018.

Among those applications, lane detection is one the most important functions which has been extensively studied for more than twenty years. An up-to-date survey of lane detection can be found in [12]. Lane detection supports various applications such as lane departure warning, lane keeping assist and lane centering in autonomous driving systems. Lane detection is also vital for understanding the traffic context. In [23], the lane detection and tracking system outputs vehicle position in the lane, lane position on the road, number of lanes, lane marking type and road coordinates in the image plane. The lane detection also helps in road sign recognition by offering road coordinates cues. Those outputs are used to feed a traffic context interpretation block which is important for vehicle's path planning.

Most of existing lane detection approaches follow a common strategy composed of different stages: perception modality, feature extraction, and model fitting. One important difficulty for a lane detection system is the rich set of conditions that have to be taken care of, requiring development of many different algorithms and sub-systems. According to [12], the main sources for condition diversity are:

- Lane and road appearance diversity: lane markings are typically 0.1 meter wide with white or yellow color, but many other variants exist such as circular reflectors, cat's-eyes, lanes with special colors and with variable width.
- Image clarity issues: near vehicles can create severe occlusions. And shadows may create misleading edges and texture on the road.
- Poor visibility conditions: the system should operate, or at least identify the condition and lower its confidence, under rain, fog, haze and night conditions. Each such condition requires another algorithmic treatment at some processing level.

Vision based imaging takes a leading role in the task of lane and road detection. According to [12], there are two main reasons. Firstly, visual data is certainly the main modality in use when human drivers are involved. Therefore, lane markings and road boundaries are designed so that a human driver is able to see them in all driving conditions. Secondly, cameras currently are the cheapest and most robust modality for automotive applications. Lidar can be used to detect host vehicle pitch and road angles in order to improve image to world correspondence. The major obvious drawback of the lidar modality is the relatively high cost of such a sensor. The current high cost prevents such sensors from becoming wide-spread commodities for automotive applications. Stereo imaging poses a greater processing challenge compared to lidar system, with increased probability of errors. Radar sensors are useful in rural areas for resolving road boundaries [85]. However, it lacks the resolving power to observe lane marking [88].

One of the most basic systems supported by lane detection is lane departure warning system (LDWS). The LDWS assists the driver by providing a warning that the vehicle is about to depart the road lane. Usually, the LDWS uses a forward looking video camera to detected lane markings of host lane (see Figure 2.19), and issue auditory or visual warning to the driver to indicate lane departure. For LDWS to reduce crashes, it must operate at a certain level of performance under varying conditions.

For a LDWS, in order to be useful, the false alarm rate should be very low, as high rates irritate drivers and lead to system rejection. The exact amount of false alarms acceptable by drivers is still a subject for research. Some available systems report few false alarms per hour. For closed-loop autonomous driving features, errors should be even several orders of magnitude lower. The robustness of the state-of-the-art processing algorithms is still far from satisfactory and much further development is required. Current vision based algorithm call for the use of many assumptions on

Figure 2.19: *LDWS camera utilized in this thesis. The camera is forward looking and located behind the windscreen.*

road and lane nature and lack in adaptive power compared to the average human driver.

2.6.1 Observation model

The LDWS module detects lane boundaries, finds the road curvature, measures position of the vehicle relative to the lanes and provides indications of unintentional deviation from the roadway in the form of an audible rumble strip sound. The system can detect various types of lane markings: solid, dashed. In the absence of lane markings, the system can utilize road edges and curbs. Figure 2.20 shows a two-lane scene, the vehicle is in the right lane. We see that the LDWS camera used can detect and box up the pedestrians, vehicles, especially the lane markings on both sides which are expressed by the green lines. It is important to note that the camera used in this thesis only detects the host lane.

Lane markings are fitted with three different models: linear model, parabolic model, and a cubic model. Take the cubic model as an example. The lane marking equation in the camera coordinate system can be expressed by:

$$y = C_3 \cdot x^3 + C_2 \cdot x^2 + C_1 \cdot x + C_0 \tag{2.27}$$

Table 2.2 displays the lane marking attributes measured by the lane detection camera.

- C_{type} consists of unknown, invalid, none, solid, dashed, botts' dots, road edge and double.

Figure 2.20: *Illustrative scenario of lane marking detection at Compiègne*

Parameter name	Definition
C_{type}	Lane marking type
$C_{quality}$	Quality of the measurement
C_{degree}	Model degree
C_0	Position parameter
C_1	Heading angle parameter
C_2	Curvature parameter
C_3	Curvature derivative parameter

Table 2.2: *Typical lanes marking attributes*

- $C_{quality}$ indicates the quality of the lane marking detection, $C_{quality} \in (1, 2, 3)$.
- C_0 is the physical distance between lane mark and camera along the lateral direction at $x = 0$ (Figure 2.2). The camera extrapolate C_0 from the lane marking in its field of view.
- C_1 is the heading angle of the detected lane marking in the camera frame at $x = 0$.
- C_2 is the curvature of the fitted lane marking at $x = 0$. C_3 is the lane curvature derivative at $x = 0$.

Since the lane marking map is represented here by polylines, the parameters that are considered for vehicle localization are the lateral distance C_0 and the heading C_1 (see Figure 2.2). Let us establish the observation model between the camera measurement and the pose vector q.

2.6.1.1 Camera distance observation model

In Figure 2.2, let L denote the lane marking detection located at ordinate C_0 in R_C. The homogeneous coordinates of point L are ${}^{C}L = [0, C_0, 1]^T$. ${}^{O}L = {}^{O}T_M \cdot {}^{M}T_C \cdot {}^{C}L$ are the homogeneous coordinates of point L in frame R_O:

$$
{}^{O}L = \begin{bmatrix} P_x \cdot \cos\psi + C_0 \cdot \sin\psi + x \\ P_x \cdot \sin\psi - C_0 \cdot \cos\psi + y \\ 1 \end{bmatrix} \tag{2.28}
$$

where P_x is a translation from point M to the bumper.

With equation (2.28), we can directly get the Cartesian coordinates (x_L, y_L) of L in R_O:

$$
\begin{bmatrix} x_L \\ y_L \end{bmatrix} = \begin{bmatrix} P_x \cdot \cos\psi + C_0 \cdot \sin\psi + x \\ P_x \cdot \sin\psi - C_0 \cdot \cos\psi + y \end{bmatrix} \tag{2.29}
$$

In Figure 2.2, $[AB]$ represents the detected lane marking segment. It is extracted from the map by data association process. The coordinates of point A and B are (x_A, y_A) and (x_B, y_B) in R_O. Vector $V = (x_{AB}, y_{AB})^T$ is defined with $x_{AB} = x_B - x_A$ and $y_{AB} = y_B - y_A$. Point L on segment $[AB]$ meets:

$$
\begin{cases} x_L = x_A + \lambda \cdot x_{AB} \\ y_L = y_A + \lambda \cdot y_{AB} \end{cases} \tag{2.30}
$$

Plugging equation (2.29) into equation(2.30), one can get:

$$
\begin{cases} P_x \cdot \cos\psi + C_0 \cdot \sin\psi + x = x_A + \lambda \cdot x_{AB} \\ P_x \cdot \sin\psi - C_0 \cdot \cos\psi + y = y_A + \lambda \cdot y_{AB} \end{cases} \tag{2.31}
$$

In order to derive the camera observation model, different cases arise.

When $x_{AB} \cdot y_{AB} \neq 0$, we get

$$
\frac{P_x \cdot \cos\psi + C_0 \cdot \sin\psi + x - x_A}{x_{AB}} = \frac{P_x \cdot \sin\psi - C_0 \cdot \cos\psi + y - y_A}{y_{AB}}
$$

Since $x_{AB} \cdot \cos\psi + y_{AB} \cdot \sin\psi \neq 0$ (the detected line is not perpendicular to the vehicle), we can derive

$$
C_0 = \frac{(P_x \cdot \sin\psi + y - y_A) \cdot x_{AB} - (P_x \cdot \cos\psi + x - x_A) \cdot y_{AB}}{x_{AB} \cdot cos\psi + y_{AB} \cdot sin\psi} \tag{2.32}
$$

When $x_{AB} = 0$, we get $C_0 = \frac{P_x \cdot cos\psi + x - x_A}{-sin\psi}$. When $y_{AB} = 0$, we can get $C_0 = \frac{P_x \cdot \sin\psi + y - y_A}{cos\psi}$. Equation (2.32) holds in both cases.

2.6.1.2 Camera heading observation model

In Figure 2.2, let α be the slope angle of segment [AB], with $\alpha \in [-\pi/2, \pi/2]$. We get:

$$C_1 = \begin{cases} \alpha - \psi, & |\alpha - \psi| \leq \frac{\pi}{2} \\ \alpha - \psi + \pi \cdot \operatorname{sign}(\psi), & |\alpha - \psi| > \frac{\pi}{2} \end{cases} \tag{2.33}$$

with $\psi \in [-\pi, \pi]$.

2.6.2 Sensitivity analysis

In practice, the camera is attached to the body of the vehicle which is itself a suspended mass. So, the 2D-world hypothesis can be easily violated. However, when a camera is correctly installed behind the windshield, the camera roll and yaw angles are very small and can be easily compensated by calibration and image warping. But when the vehicle crosses a speed bumper, both the altitude and the tilt angle of the camera frame undergo changes (see Figure 2.21). The same issue appears when the slope of the road changes or when the vehicle accelerates or breaks. Moreover, the height of the camera also depends on the load of the vehicle. Finally, some calibration errors related to the camera intrinsic parameters may occur.

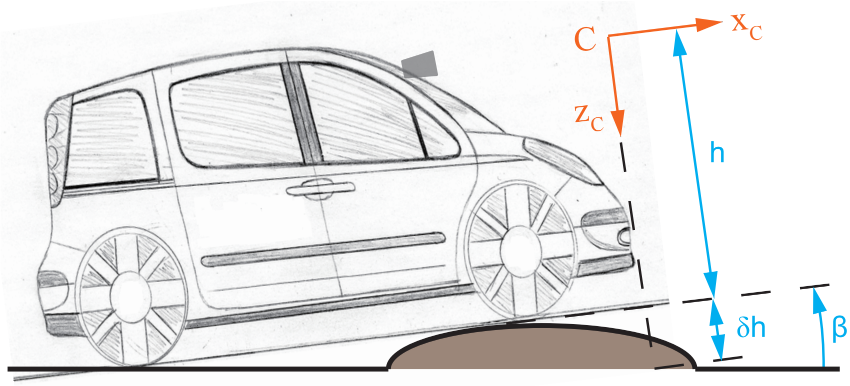


Figure 2.21: *Parameter variation while climbing a speed bumper*

So, it is crucial to evaluate the sensitivity of the parameters involved in the development of lane marking measures particularly the tilt angle and the height with respect to the road.

2.6.2.1 Nominal model with the camera parallel to the road

In the camera frame R_C, the equation of a detected lane marking can be written as (by considering a first order Taylor's expansion of a clothoid in the camera frame [73]):

$$\begin{cases} y = C_1 \cdot x + C_0 \\ z = h \end{cases} \tag{2.34}$$

Where C_0 and C_1 are respectively the lateral distance and the heading of the detected lane marking and h is the height of the camera w.r.t. the road.

In the image plane, using perspective projection, we have:

$$\begin{cases} u = f \cdot \frac{y}{x} + u_0 \\ v = f \cdot \frac{z}{x} + v_0 \end{cases} \tag{2.35}$$

with f the focal length, and $[u_0, v_0]^T$ the coordinates of the principal point.

By plugging equation (2.34) into equation (2.35), we get:

$$\frac{u - u_0}{f} = C_1 + \frac{C_0}{x} \tag{2.36}$$

and

$$x = \frac{h \cdot f}{v - v_0} \tag{2.37}$$

By plugging equation (2.36) into equation (2.37), we get:

$$u - \frac{C_0}{h} \cdot v + \frac{C_0 \cdot v_0}{h} - u_0 - C_1 \cdot f = 0 \tag{2.38}$$

The equation of the detected lane marking [73][28], expressed in image plane is:

$$a_p \cdot u + b_p \cdot v + c_p = 0 \tag{2.39}$$

where a_p, b_p and c_p are the line coefficients. By comparing equation (2.38) and equation (2.39), we get:

$$\begin{cases} C_0 = -\frac{h \cdot b_p}{a_p} \\ C_1 = -\frac{a_p \cdot u_0 + b_p \cdot v_0 + c_p}{a_p \cdot f} \end{cases} \tag{2.40}$$

2.6.2.2 The camera is tilted downward

Let suppose that the camera is tilted downward (or upward) with an angle β. In the camera frame R_C, the equation of a detected lane marking can be written as:

$$\left\{ \begin{array}{c} y = C_1 \cdot (x \cdot cos\beta - z \cdot sin\beta) + C_0 \\ x \cdot sin\beta + z \cdot cos\beta = h \end{array} \right. \tag{2.41}$$

We get:

$$\left\{ \begin{array}{c} C_0 = -\frac{h \cdot b_p \cdot f + (a_p \cdot u_0 + b_p \cdot v_0 + c_p - f \cdot b_p \cdot tan\beta) \cdot h \cdot sin\beta}{a_p \cdot f \cdot cos\beta} \\ C_1 = -\frac{a_p \cdot u_0 + b_p \cdot v_0 + c_p - f \cdot b_p \cdot tan\beta}{a_p \cdot f} \end{array} \right. \tag{2.42}$$

2.6.2.3 Sensitivity analysis

In this subsection, β, f , h , C_0 and C_1 stand for the nominal values of these five parameters, and $\beta + \delta\beta$, $f + \delta f$, $h + \delta h$, $C_0 + \delta C_0$ and $C_1 + \delta C_1$ are their true values. We suppose that the range of C_1 is $[-0.3, 0.3]$ rad and the range of C_0 is $[-127, 128]$ meters. Let study the sensitivity to β, f and h.

- Sensitivity to β

$$\left\{ \begin{array}{c} \frac{\delta C_0}{C_0} = \frac{a_p \cdot C_1 \cdot [tan(\beta + \delta\beta) - tan\beta] - b_p \cdot [sec(\beta + \delta\beta) - sec(\beta)]}{a_p \cdot C_1 \cdot tan\beta - b_p \cdot sec\beta} \\ \frac{\delta C_1}{C_1} = \frac{b_p \cdot [tan(\beta + \delta\beta) - tan\beta]}{a_p C_1} \end{array} \right. \tag{2.43}$$

Let suppose that a straight line in the image plane meets: $\frac{b_p}{a_p} = -1$. In order to see the sensitivity clearly, an example is given in Table 2.3.

$\frac{\delta\beta}{\beta}$	0.4	1	2
$\frac{\delta C_0}{C_0}$	0.014	0.037	0.083
$\frac{\delta C_1}{C_1}$	0.118	0.296	0.602

Table 2.3: *Sensitivity analysis with* $h = 1.5\,m$, $C_1 = 0.3\,rad$, $\beta = 5\,degree$

As shown by Table 2.3, C_1 is very sensitive to an error in β, but C_0 is not.

- Sensitivity to f

$$\left\{ \begin{array}{c} \frac{\delta C_0}{C_0} = \frac{b_p \cdot tan\beta \cdot sin\beta - C_1 \cdot a_p}{b_p - C_1 \cdot a_p \cdot sin\beta} \cdot \frac{\delta f}{f + \delta f} \\ \frac{\delta C_1}{C_1} = \frac{b_p \cdot tan\beta - a_p \cdot C_1}{a_p \cdot C_1} \cdot \frac{\delta f}{f + \delta f} \end{array} \right. \tag{2.44}$$

Taking the relationship $\frac{b_p}{a_p} = -1$, and the ranges of C_0 and β into consideration, one can easily find that C_0 is not sensitive to an error in f, but C_1 could be very sensitive to it.

- Sensitivity to h

$$\begin{cases} \frac{\delta C_0}{C_0} = -\frac{\delta h}{h} \\ \frac{\delta C_1}{C_1} = 0 \end{cases} \tag{2.45}$$

In this case, the error on parameter C_0 is directly related to the height variation.

As a conclusion:

- a C_0 error is proportional to an error in h but δh has a variation limited to few centimeters in normal driving conditions;
- C_1 is sensitive to errors on f and β. The influence of the intrinsic parameters of the camera is limited but the tilt angle is a crucial parameter even in normal driving conditions.

So, we believe that C_0 is an accurate measure while C_1 should be used with caution. A way to exploit it accurately could be to use an IMU providing an estimate of β or to estimate vanishing points using computer vision[14] which are not in the scope of this thesis.

2.7 Experimental setup and collected data

2.7.1 Experimental setup

The experimental vehicles used to evaluate the developed algorithm are equipped Renault Fluence ZE as depicted in Figure 1.1. Figure 1.1a presents the experimental vehicle IRIS of Heudiasyc laborotary. Figure 1.1b illustrates the experimental vehicle of the PAMU project on a test road where environmental factors are more controllable than on public roads. Both vehicles have been used to collect data for validating the algorithms proposed in this thesis. This section presents the data collected by the PAMU experimental vehicle. It was equipped with an IMU Oxford RT3000 which provided ground truth data at 100Hz rate. A CAN-bus gateway was used to access to the wheel speed sensors and the yaw rate gyro. The measured speeds $[v^m, w^m]$ from the CAN bus were available at 50Hz. A MobilEye EyeQ2 camera was installed behind the windscreen to detect the lane markings at 10Hz. A low-cost u-blox 6T GPS receiver with a patch antenna was used. It provided position measurements at 5Hz with raw data.

Outdoor experiments were carried out near Paris France in May 2013. Five passages were performed on the same road. In two of them, the ground truth system failed when passing through a challenging urban area. Therefore, only three of tests with

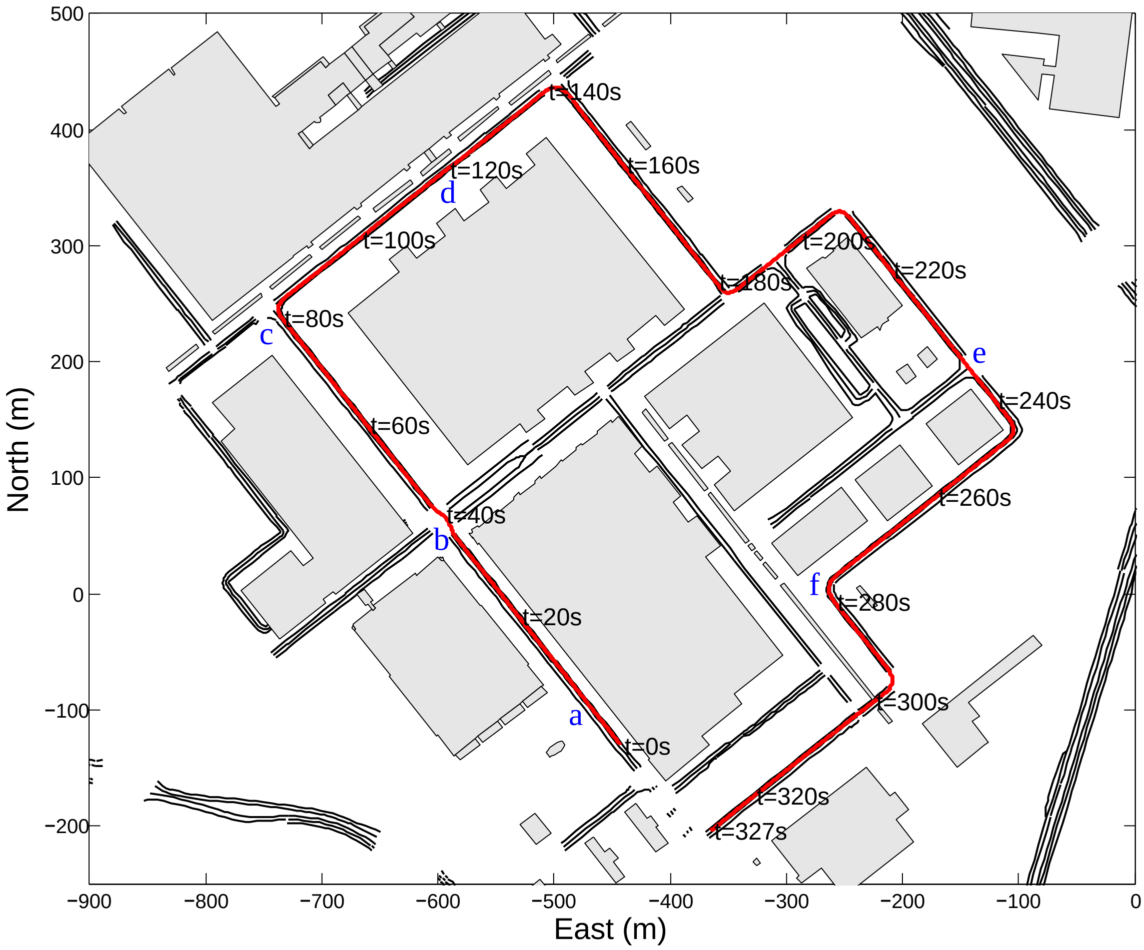
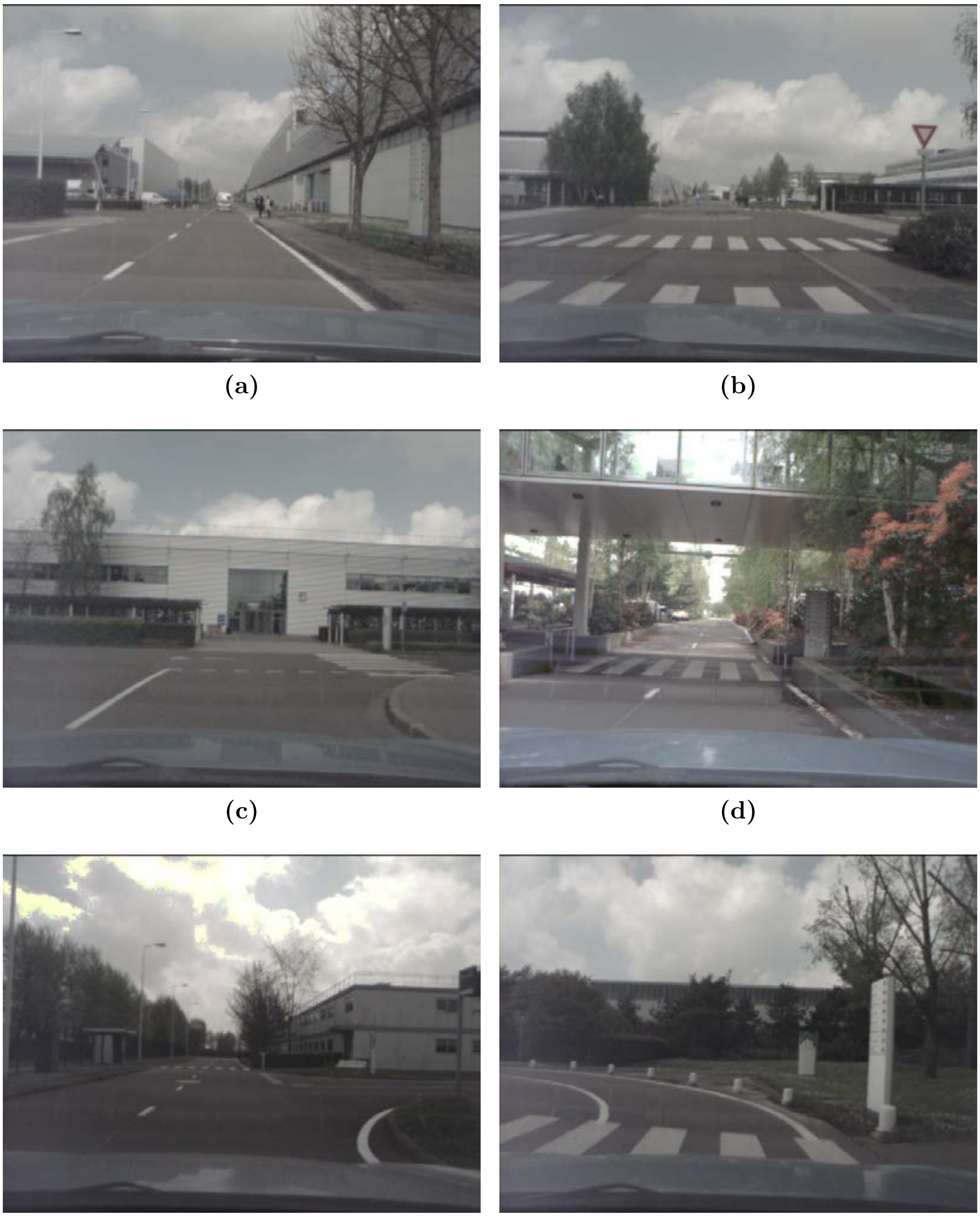


Figure 2.22: *Test scene and the reference trajectory (in red color) in the local ENU frame. The start/end points and traveling direction are indicated by the tagged time.*

validated ground truth are used to analyze the proposed methods in this thesis. The first two tests were performed in the morning. The third in the afternoon.

The traveling distance for each test was about 2km. Taking the first test as an example, Figure 2.22 depicts the experimental trajectory. The red line stands for the ground truth. The boxes in gray are buildings. Blacks lines are mapped lane markings.

The vehicle starts at t = 0s and stops at t = 327s. Between t = 80s and t = 140s, the vehicle passes in a strong urban canyon of 300 meters and the u-blox receiver suffers from multipath around t = 123s when the vehicle is traveling in an urban canyon environment. Figure 2.23 displays several typical scenarios recorded by the experimental vehicle.



(a) **(b)**

(c) **(d)**

(e) **(f)**

Figure 2.23: *Scenarios recorded by a webcam mounted on the experimental vehicle. The roads of the trials consist of two lanes with mainly dashed lane markings in the center of the road and solid lane markings on both sides, as in (a). In (b), the vehicle passes through a roundabout with rare lane marking feature. (c) shows a typical scenario of corner turning, the lane marking detection always fails during these moments. (d) displays the most challenging area for our experiments where multipath effect and camera failure happen at the same time. In (e), the vehicle arrives at a fork where the camera fails. In (f), the camera fails to detect the highly curved lane markings. Their positions in the map are tagged in Figure 2.22.*

2.7.2 Dead reckoning sensors

Thanks to a CAN bus gateway, the DR sensors are those already embedded in the ABS and ESP systems of the car (we have not added any IMU nor encoders). They are sampled at 100 Hz.

Figure 2.24 shows the measured vehicle linear speed v^m and yaw rate ω^m of the three tests. The maximum speed was about 8 m/s.

2.7.3 Data of L1-GPS

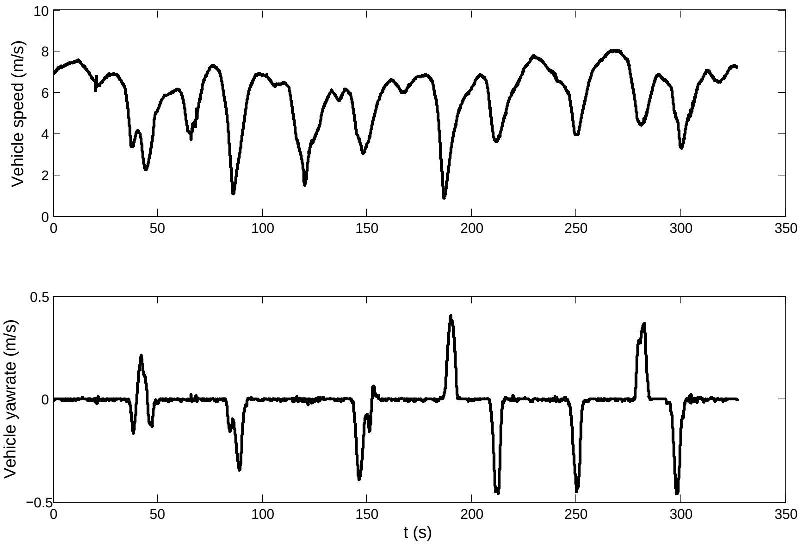
Figure 2.25a displays the satellites in view for the first two tests and figure 2.25b for the third test. Clearly the GPS satellites geometries are quite different between the morning and the afternoon.

Figures 2.26, 2.27 and 2.28 show the GPS positioning errors. The biggest horizontal positioning error is more than 7m which happened in test 3 when the vehicle passed through the challenging urban area.

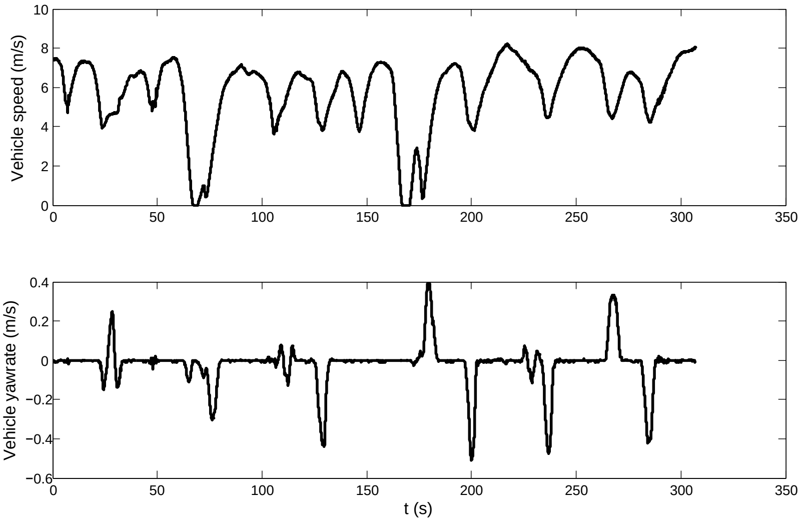
2.7.4 Lane detections of the camera

Figure 2.29 gives the camera measurement C_0. As the vehicle was travelling in the center of the lane, C_0 is typically around 1.5 m. Although the MobilEye's frequency is bigger than the u-blox GPS receiver, one can notice that the continuity of the camera measurement is worse than GPS. The non-availability of the camera measurement mainly occurs in highly curved areas, such as roundabouts and corners.

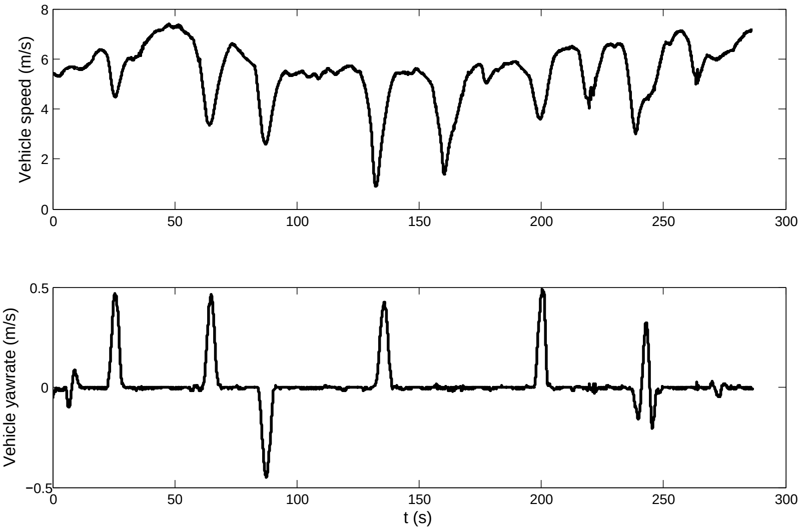
The camera measurement plays an important role in improving the localization performance, so its measurement error is investigated here. Suppose the lane marking is perfectly extracted from the image and the lane marking segment in the map is well matched. The error sources are mainly caused by the violation of the assumption that the camera is parallel with the road, calibration errors of camera parameter and



(a) *Test 1*

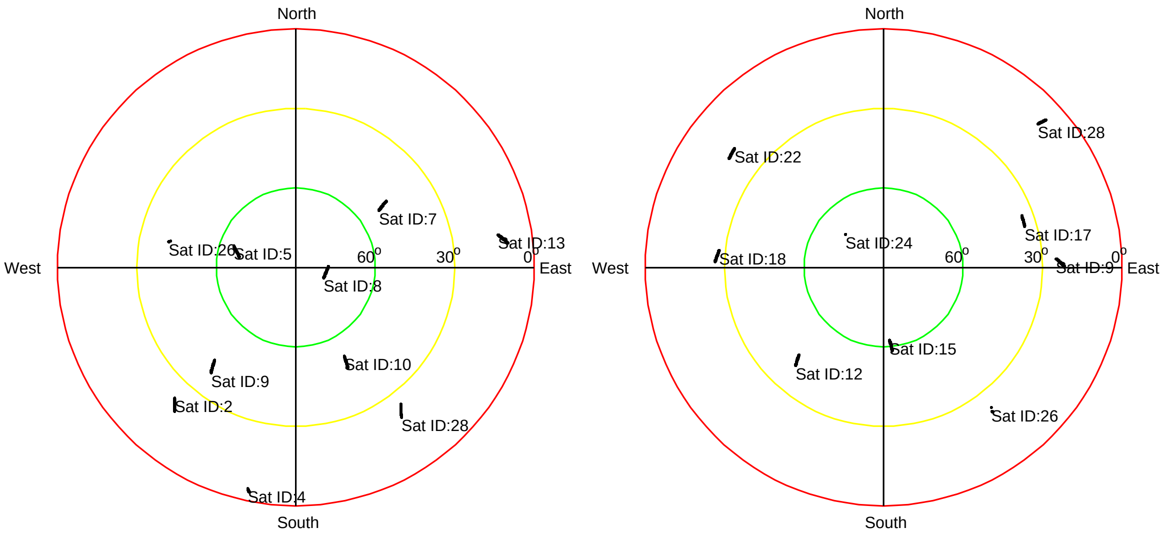


(b) *Test 2*



(c) *Test 3*

Figure 2.24: *Vehicle speed and yaw rate*



(a) *Satellites in view around 10 o'clock UTC in the morning*

(b) *Satellites in view around 14 o'clock UTC in the afternoon*

Figure 2.25: *Azimuth-Elevation plot of the tracked GPS satellites during the experiments*

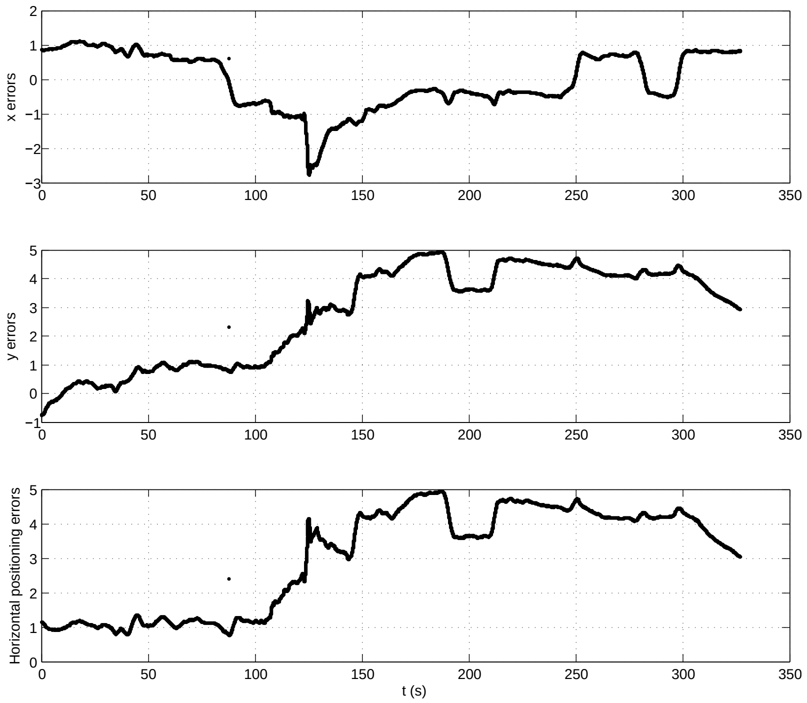


Figure 2.26: *GPS positioning errors in the ENU frame of test 1*

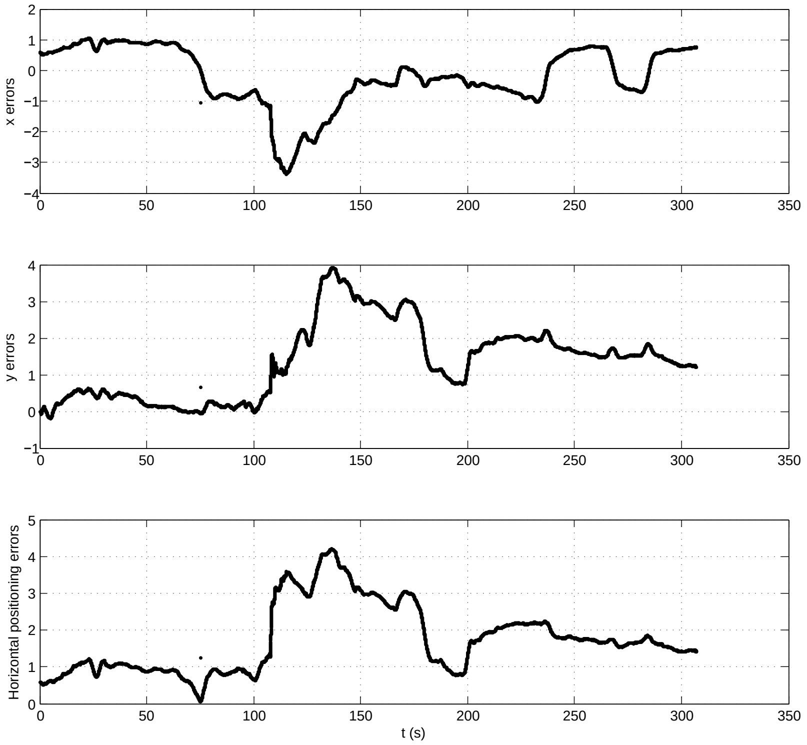


Figure 2.27: *GPS positioning errors in the ENU frame of test 2*

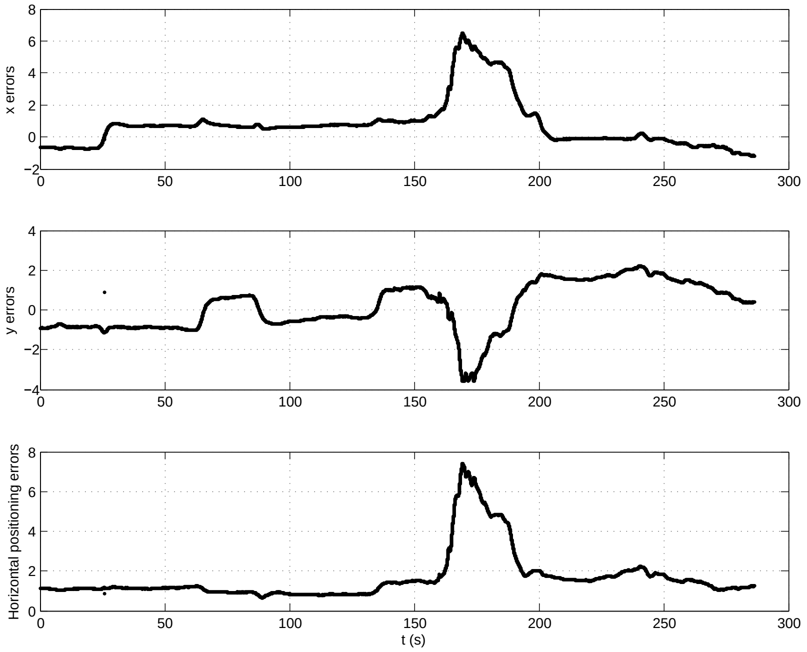
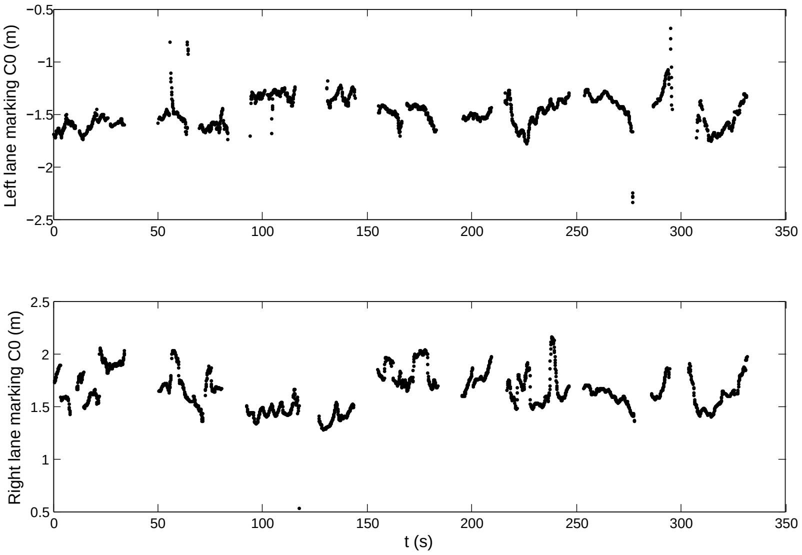
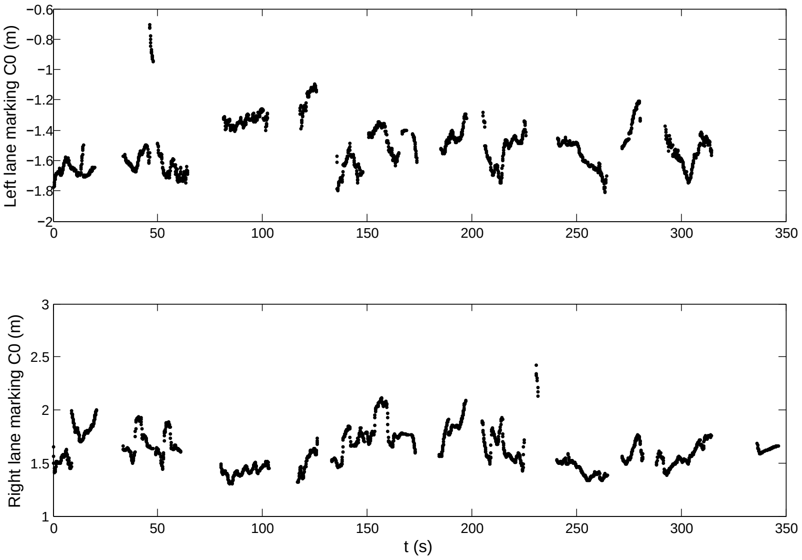


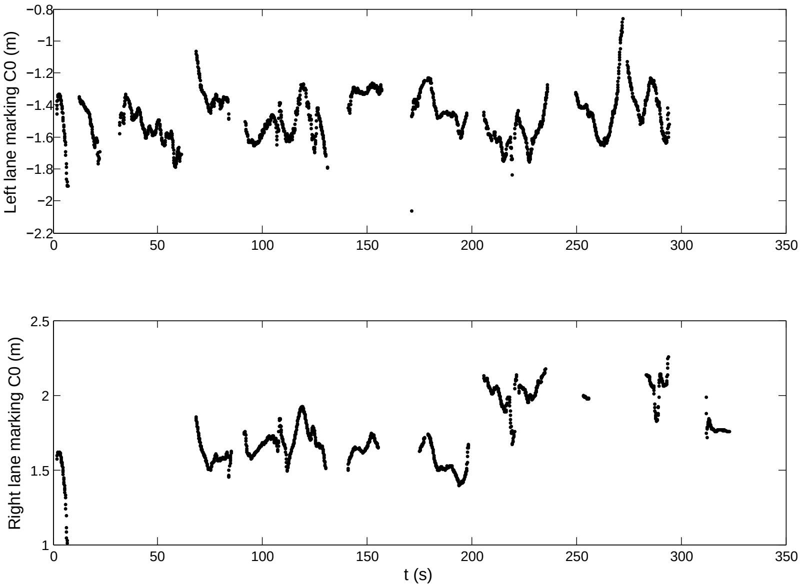
Figure 2.28: *GPS positioning errors in the ENU frame of test 3*



(a) *Test 1*



(b) *Test 2*



(c) *Test 3*

Figure 2.29: *Camera's C_0 measurements with high quality ($C_{quality} \geq 2$)*

		camera				total
		none (2)	*solid (3)*	*dashed (4)*	*double (7)*	
map	*solid*	15	757	81	45	898
	dashed	172	791	5243	0	6206
	pavement	100	3090	244	66	3500
	barrier	3	27	16	12	58
	other	55	342	651	0	1048

Table 2.4: *Camera detection of lane marking type with map reference*

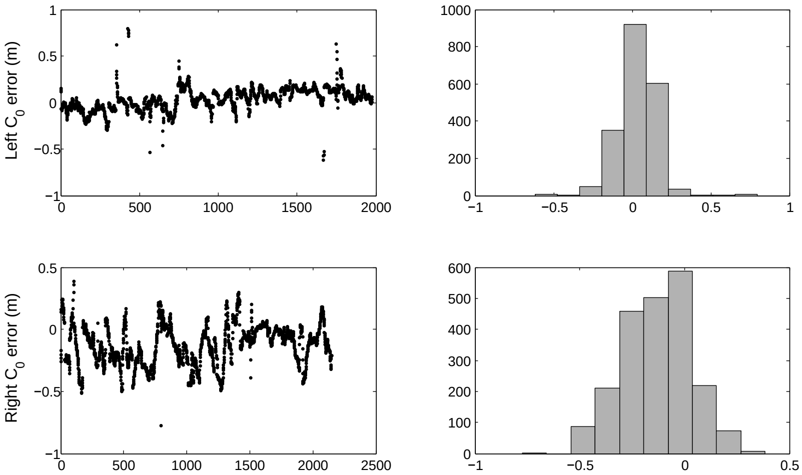
errors of the lane marking map. Here, the referenced pose estimated by the IMU Oxford RT3000 is used to match the detected lane marking. The reference value of C_0 is calculated by substituting the pose and the parameters of the matched marking into equation (2.32). Errors of the C_0 observation is shown in Figure 2.30. Its distribution approximates quite well a zero mean distribution. The variance is less than 0.05 m^2.

Figure 2.31 gives the type of detected lane marking classified by the camera. The number (2, 3, 4 and 7) which represents different lane marking type is defined in Table 2.4. The vehicle traveled mostly on a two-lane roadway with dashed marking in the center of the road and solid marking on both sides, so the left detection is mainly dashed and the right detection is mainly solid.

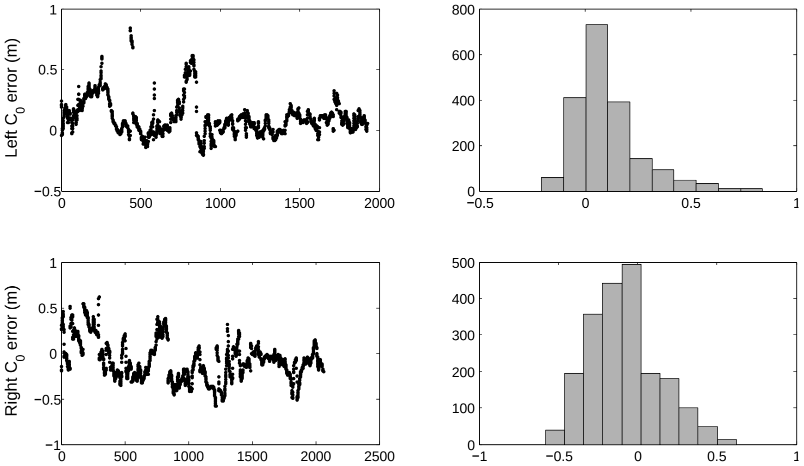
The lane marking type detection is not a location based measurement, but it is helpful to disambiguate in which lane the vehicle is, especially for a multilane road. Table 2.4 shows the performance of the lane type detection where 11710 samples are studied.

2.8 Conclusion

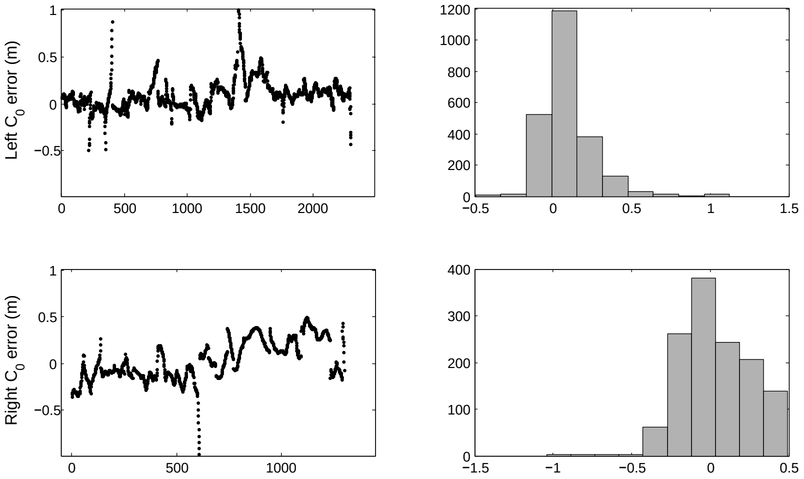
In this chapter, several technologies for estimating the vehicle localization have been presented and discussed. In order to achieve a low-cost solution, we will consider the data fusion problem of DR sensors with L1-GPS, a lane marking detection camera and a lane marking map. A new camera observation model has been be presented and we suggest to use only the lateral position information based on a sensitivity analysis. Outdoor experiment results carried out with an experimental vehicle have been presented. The camera measurement, coupled with the lane marking map, has a better accuracy than L1-GPS. Therefore, it is possible to improve the overall localization performance when the system exploits the camera information. A GPS receiver being a major component for vehicle navigation, we propose to use it as much as possible. It provides different kind of information at different level



(a) *Test 1. Left C_0 error: mean = 0.03m; variance = 0.01m^2. Right C_0 error: mean = -0.1m; variance = 0.03m^2.*

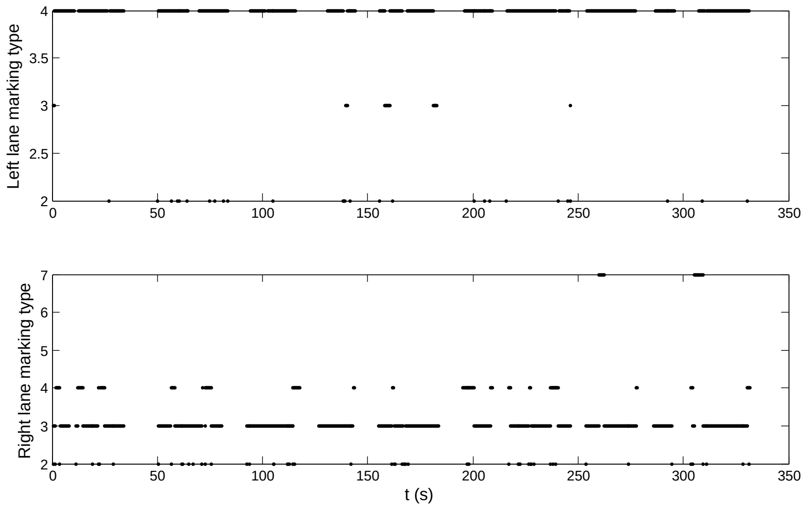


(b) *Test 2. Left C_0 error: mean = 0.1m; variance = 0.02m^2. Right C_0 error: mean = -0.08m; variance = 0.05m^2.*

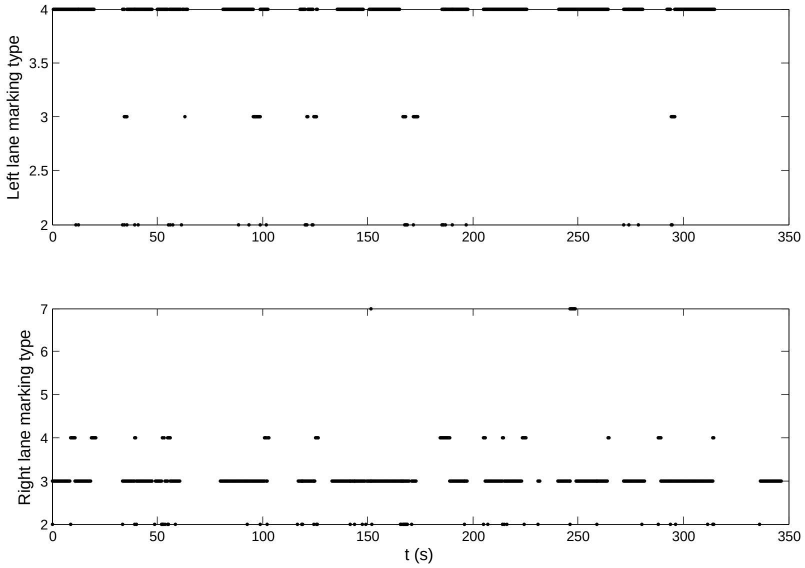


(c) *Test 3. Left C_0 error: mean = 0.09m; variance = 0.02m^2. Right C_0 error: mean = 0.03m; variance = 0.05m^2.*

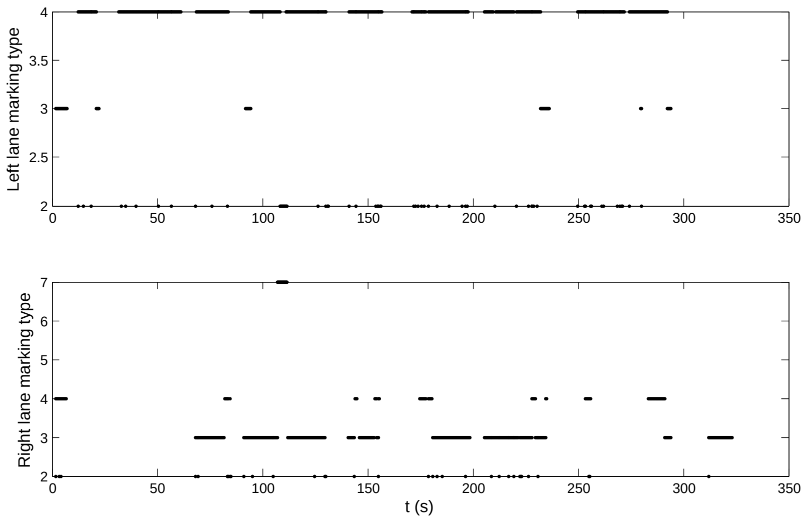
Figure 2.30: *Camera C_0 measurement errors. The ground truth of C_0 is calculated by inserting the vehicle pose estimated by RTK-GPS into equation (2.32)*



(a) *Test 1*



(b) *Test 2*



(c) *Test 3*

Figure 2.31: *Lane markings type provided by Mobileye camera. 2,3,4 and 7 represent "none", "solid", "dashed", "double" respectively. The types that Mobileye is able to detect is listed in table 2.4*

depending on the way the data fusion is handled, such as position fixes or pseudoranges and Dopplers. Depending on at what level the data is exchanged between the GPS receiver and other sensors, the localization system architectures are commonly referred to as loosely coupled, tightly coupled and ultra tightly coupled. In the following chapter, we starts by studying a loosely coupling scheme.

Chapter 3

GNSS Map-aided Localization with a Loosely Coupled Scheme

Contents

3.1 Introduction

The estimation of the pose of the vehicle in the working frame R_O with enough accuracy, availability and reliability for autonomous navigation needs the data fusion of several sources of information particularly when using low cost sensors. The problem that is addressed in this chapter is a loosely coupling data fusion scheme which refers to the fusion of L1-GNSS position fixes with other sensors as illustrated by Figure 3.1 (DR sensors, a forward looking camera and georeferenced lane markings). In chapter 2, the measurements of the DR sensors (gyro and wheel speed sensors), GNSS and lane detection camera have been studied and their corresponding observation models linking the measurements with the vehicle state have been developed.

GNSS/DR data fusion has been widely studied in vehicle navigation area [22][13][45][117]. GNSS and DR systems are complementary in many aspects. DR systems are accurate in short-term if they are well calibrated, but the estimation error grows

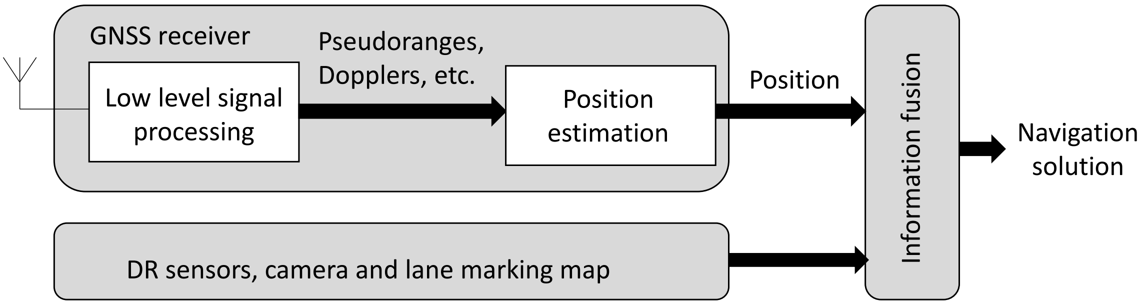


Figure 3.1: *Loosely coupled scheme*

unboundedly over time. GNSS is worldwide and provides information that can compensate the drift. However, GNSS signals are subject to outages and errors. DR systems provide data with continuity at a high frequency for highly dynamic vehicles. They can bridge short-term GNSS outages.

A loosely coupled GNSS/DR system is also known as cascaded filtering which means that the estimates of the GNSS receiver and the unfiltered data of DR sensors are fed into the navigation filter [13]. GNSS/DR sensor fusion is a nonlinear filtering problem. The integration is often performed using Extended Kalman filtering (EKF) [9] or Unscented Kalman Filtering (UKF) [67]. Generally, both EKF and UKF work well for the integration of typical MEMS-based dead reckoning with GNSS and have similar performance [92]. UKF is more robust to INS calibration errors and large initialization errors [86][32][64] and the EKF is more computationally efficient [115].

This loosely coupled option is often chosen because of its simplicity as position fixes are available in any standard GNSS unit. Although GNSS receiver manufacturers provide more and more reliable solutions with the development of satellite technology with different constellations, for a standalone solution (i.e. a solution that doesn't exploit other sensors or differential corrections) it is still hard to compensate for the errors mentioned in chapter 2. The remaining errors after having applied correction models make that L1-GNSS is not accurate enough to localize a vehicle within a lane. In addition, prediction mechanisms in case of outages and multipath can induce large errors which make GNSS receivers alone unavailable for the navigation task. Recently, the integration of highly detailed maps coupled with camera is becoming increasingly popular. Thanks to this configuration, it is possible to model GNSS errors and to compensate them in the filter. Since we use L1-GNSS technology, this chapter will therefore pay attention to the modeling of L1-GNSS errors.

In section 3.2, previous work related to loosely coupling GNSS with DR, camera and digital map is surveyed. Section 3.3 reminds several useful mathematical concepts for describing random sensor errors. The localization solver is realized by an EKF which takes sensor errors into consideration by augmenting the state space. Thus section 3.5 reminds Kalman filtering with shaping filter. Modeling of GNSS errors is discussed in section 3.4. Outdoor experiments are described in section 2.7. Section 3.6 explains the practical implementation of the EKF. Section 3.7 presents the results and section 3.8 draws conclusion of the loosely coupled data fusion strategy.

3.2 Related work

Detailed and highly accurate maps provide prior information to the navigation tasks and give the possibility to get a low cost self-driving capability in terms of on-board hardware equipment. With this in mind, this section investigates hereafter several earlier and recent approaches which use accurate digital maps coupled with camera measurements to improve the positioning performance.

When using GNSS as a sensor in a loosely coupled fusion architecture, it provides position and velocity information. Some simple GNSS/DR coupling solutions model the errors of GNSS position fixes as white noises. In reality, a GNSS receiver is affected by systematic noise that is caused by atmospheric effects which tend to change slowly with time. In addition, the time-correlation of the errors in position are made worse by the effects of the filter implemented in the receiver. If these correlations are not considered, it can cause a strong degradation of the data fusion filter. The random part of GNSS fixes can be modeled by some random process. The modeling and estimation of GNSS position fix biases have been discussed in many publications as [75][29][89] for instance.

A GNSS fix observation is commonly expressed as:

$$\begin{cases} x_{GNSS} = x + \varepsilon_x + \beta_x \\ y_{GNSS} = y + \varepsilon_y + \beta_y \end{cases} \tag{3.1}$$

where ε_x and ε_y are biases on the x and y axes in the ENU frame. β_x and β_y are white noises.

Now let us consider the measurements done by cameras. A camera observation model is in general a non linear expression:

$$\mathbf{y}_{cam} = g_{cam}\left(\mathbf{x}, \mathbf{l}\right) + \boldsymbol{\beta}_{cam} \tag{3.2}$$

where $\mathbf{y}_{cam}$ is the measurement vector. $\mathbf{x}$ is the state vector of the vehicle. $\mathbf{l}$ is the vector of the lane marking parameters extracted from the digital map. $\boldsymbol{\beta}_{cam}$ is a measurement noise that is supposed centered if the camera is well calibrated. The observation function $g_{cam}\left(\cdot\right)$ changes a lot depending on the approach. The observation model of the camera/map measurement used in this thesis has been already presented in chapter 2. It can take explicitly into account the camera position in the body frame.

Different map models have been considered in the literature. In [75], a map made of a grid of rectangular facets representing the roadsides is used. For each facet, its origin, orientation, length and width are defined in the local navigation frame. By using a front view camera, an image processing algorithm is then implemented to give precise lateral position and orientation of the vehicle with respect to the roadside. $\mathbf{l}$ corresponds to the parameters of nearest facet. This work shows the possibility and the advantage of integrating lane boundary information which refers often to white lane marking detections. In [89], the map is modeled by GNSS waypoints designating lane centers, stop lines and lane markings. Dashed, solid and unstructured lane types are denoted for each lane marking. In this approach, $\mathbf{y}_{cam}$ contains the perpendicular distance of the camera from each of the lane boundaries, the camera heading with respect to the lane, the lane width, the lane type and the distance to the stop line. The authors use a particle filter to augment GNSS/DR solution with camera measurements coupled with a surveyed map. They give a demonstration of the stability of the localization solutions used to feedback the controller of a full-size autonomous ground vehicle. Recently, more works with similar ideas but different configurations have been carried out, such as in [66][52][79]. In [66], a front view camera is used to detect lane markings and zebra lines which have been charted in the map. In [52], two lateral cameras are used to provide an assessment of the lateral distances between the vehicle and the ego-lane markings or borders. This information is then coupled with an accurate digital map of the road markings. However, the modeling of GNSS errors is not considered. Particularly, the authors of [79] propose a localization method with GNSS/DR error estimation based on lane detection of curved lane models. A longitudinal measurement from a curve matching is proposed. The curved parameter of the lane ahead is fitted using the waypoint map by supposing that the waypoints and the lane markings are parallel. The fitted waypoint curve is then transformed into the vehicle frame. However, the impact of the uncertainty of the estimated vehicle heading on this transformation (which introduces errors on the longitudinal measurement) is not discussed.

In regard to the modeling of ε_x and ε_y, in [75] and [79], the GNSS fixes biases are considered constant between two samples:

$$\begin{cases} \varepsilon_{x,k} = \varepsilon_{x,k-1} \\ \varepsilon_{y,k} = \varepsilon_{y,k-1} \end{cases} \tag{3.3}$$

The authors of [75] proposed a bias management strategy to decide whether the bias has changed.

In [89][66], the GNSS biases are modeling as:

$$\begin{cases} \varepsilon_{x,k} = e^{-dt/\tau_x} \cdot \varepsilon_{x,k-1} + \nu_{x,k-1} \\ \varepsilon_{y,k} = e^{-dt/\tau_y} \cdot \varepsilon_{y,k-1} + \nu_{y,k-1} \end{cases} \tag{3.4}$$

where τ_x and τ_y are the bias's autocorrelation time constants. dt is the elapsed time. ν_x and ν_y are zero-mean white noise processes. This modeling of GNSS biases was first suggested in [13]. For a loosely coupled approach which takes the modeling of the biases into account, the vehicle dynamic equations should be augmented by equation (3.3) or (3.4).

In the following, let us start with theories and techniques for modeling sensor errors.

3.3 Random processes for modeling sensor errors

Random processes can represent the evolution of the uncertainties of knowledge about physical systems [50]. For a localization system, the uncertainties come from two aspects: uncertainties in the vehicle dynamic processes and random errors of observations made by sensors. For a GNSS/DR coupled system, the ability of the DR system to bridge GNSS outages depends on the DR sensor quality. The errors can be classified into deterministic and random errors. Deterministic errors (also known as systematic) can be removed by calibration. Random errors are often modeled using random processes such as random constant, random walk or autoregressive process. This is common for inertial sensors [13][93] and GNSS [89].

In this section, an overview of different random processes for stochastic modeling of sensor errors is presented. Let $\{b_t, t \in T\}$ be a random process. T may be discrete or continuous. The stochastic process can be characterized by specifying the n-order joint probability density function $f_n\left(\cdot\right)$ of the random variables $b_{t_1}, \dots, b_{t_n}$:

$$f_n\left(b_{t_1}, \dots, b_{t_n}\right) \tag{3.5}$$

for any finite set $\{t_1, \dots, t_n\} \in T$ [65].

When modeling a stochastic error using a random process or a signal, the process is often assumed to be stationary, i.e. its statistical quantities are invariant over time. The strict stationarity requires the joint density functions to be invariant to a time translation τ:

$$f_n\left(b_{t_1}, \dots, b_{t_n}\right) = f_n\left(b_{t_1+\tau}, \dots, b_{t_n+\tau}\right) \tag{3.6}$$

The process is strictly stationary of order k, if equation (3.6) holds for $n \leq k$ only. The densities of a random process at the first and second order can answer many important questions about the process. In practice, a less demanding form which requires equation (3.6) holds for $n \leq 2$ is often used. The random process under this condition is called weakly stationary or covariance stationary. In the following sections, we refer to stationary process as a covariance stationary one.

3.3.1 Autocorrelation function

A stationary process can be completely defined by its autocorrelation function specifications which describes the general dependence of the data values at one time with the values at other times [24]. Autocorrelation is the similarity between observations as a function of the time lag between them. It is a mathematical tool for finding repeating patterns such as the presence of a periodic signal obscured by noise. The autocorrelation of b_t is defined as:

$$\bar{R}_{bb}\left(t_1, t_2\right) = E\left[b_{t_1} \cdot b_{t_2}\right] \tag{3.7}$$

where $E\left[\cdot\right]$ is the mathematical expectation operator. If the random signal is stationary, the expectation is not dependent on t. Therefore, equation (3.7) depends only on the difference $\tau = t_2 - t_1$ and the autocorrelation can be rewritten as:

$$\bar{R}_{bb}\left(\tau\right) = E\left[b_t \cdot b_{t+\tau}\right] \tag{3.8}$$

where $\bar{R}_{bb}\left(\tau\right)$ is the autocorrelation function of b_t, t an arbitrary time and τ the time lag. Note that the autocorrelation function is the ensemble average of the product of b_t and $b_{t+\tau}$. If ergodicity applies, it is often easier to compute $\bar{R}_{bb}\left(\tau\right)$ as a time average instead of an ensemble average. The autocorrelation function in the form of time average, denoted $R_{bb}\left(\tau\right)$, is defined as:

$$R_{bb}\left(\tau\right) = \int_{-\infty}^{\infty} b_t \cdot b_{t+\tau} dt \tag{3.9}$$

When dealing with discrete time signals, the autocorrelation sequence is computed instead of the autocorrelation function. The autocorrelation sequence $R_{bb}\left(m\right)$ is defined as

$$R_{bb}\left(m\right) = mean\left[\sum_{k=-\infty}^{\infty} b_k \cdot b_{k+m}\right] \tag{3.10}$$

where k is the sampling sequence number and m is the sampling lag.

Equation (3.10) assumes an infinite data record for calculating the autocorrelation. In practice, only a block of N signal samples is available which is only a segment of the signal. If the assumption that the signal is stationary and ergodic holds, for a block of N signal samples, the autocorrelation sequence expressed by equation (3.10) can be replaced by the sample autocorrelation [98]. For a time series of b_k,

$k = 0, 1, 2, 3, ..., N - 1$, the sample autocorrelation $\hat{R}_{bb}\left(m\right)$ can be estimated by:

$$\hat{R}_{bb}\left(m\right) = \frac{1}{N}\sum_{k=0}^{N-1} b_k \cdot b_{k+m} \tag{3.11}$$

The value of the sample autocorrelation at lag $m = 0$ is given as:

$$\hat{R}_{bb}\left(0\right) = \frac{1}{N}\sum_{k=0}^{N-1} b_k^2 = \sigma_b^2 + \mu_b^2 \tag{3.12}$$

where σ_b and μ_b are the standard deviation and mean value of b, respectively.

It is important to note that, if the random signal is not ergodic, it is necessary to distinguish between the ensemble average autocorrelation function and the time average version. If the autocorrelation function decreases rapidly with τ, the random process changes rapidly with time. Conversely, if the autocorrelation function decreases slowly with τ, the process is more predictable.

For stationary processes, there is the well known Wiener-Khinchine relation defined as:

$$S_{bb}\left(j\omega\right) = F\left[R_{bb}\left(\tau\right)\right] = \int_{-\infty}^{+\infty} R_{bb}\left(\tau\right) \cdot e^{-j\omega\tau} d\tau \tag{3.13}$$

where $F\left[\cdot\right]$ indicates Fourier transform. S_{bb} is called the power spectral density (PSD) function of the random process. The PSD function of the discrete signal is

$$S_{bb}\left(e^{j\omega}\right) = \sum_{m=-\infty}^{\infty} R_{bb}\left(m\right) \cdot e^{-j\omega m} \tag{3.14}$$

The PSD describes how the power (or variance) of a time series of measurements is distributed in the frequency domain. In turn, the autocorrelations are also determined, respectively, by the inverse Fourier transform of the PSD, such that

$$R_{bb}\left(\tau\right) = \frac{1}{2\pi}\int_{-\infty}^{+\infty} S_{bb}\left(j\omega\right) \cdot e^{j\omega\tau} d\omega \tag{3.15}$$

$$R_{bb}\left(m\right) = \frac{1}{2\pi}\int_{-\pi}^{\pi} S_{bb}\left(e^{j\omega}\right) \cdot e^{j\omega m} d\omega \tag{3.16}$$

3.3.2 White noise

In signal processing, white noise is a stationary random signal with a constant PSD. In discrete time, white noise is a discrete signal whose samples are regarded as a sequence of serially uncorrelated random variables with zero mean and finite variance. A white random sequence $\{w_n, n = 1, 2, ...\}$ is a Markov sequence for which the probability density function $p(\cdot)$ respects:

$$p(w_k|w_l) = p(w_k) \ \ (k > l) \tag{3.17}$$

Denoting the white noise PSD amplitude by A_w, one gets:

$$S_{ww}(j\omega) = S_{ww}\left(e^{j\omega}\right) = A_w \tag{3.18}$$

The corresponding autocorrelation function for continuous and discrete white noise signal are respectively:

$$R_{ww}(\tau) = \frac{A_w}{2\pi}\int_{-\infty}^{+\infty} e^{j\omega\tau} d\omega = A_w \cdot \delta(\tau) \tag{3.19}$$

$$R_{ww}(m) = \frac{A_w}{2\pi}\int_{-\pi}^{\pi} e^{j\omega m} d\omega = A_w \cdot \delta(m) \tag{3.20}$$

where $\delta(\cdot)$ is the delta function (Dirac-delta function $\delta(\tau)$ for continuous time and unit impulse function $\delta(m)$ for discrete time) and is defined as:

$$\delta(\tau) = \begin{cases} 0 & \tau \neq 0 \\ \infty & \tau = 0 \end{cases} ; \int_{-\varepsilon}^{\varepsilon} \delta(\tau) d\tau = 1 \quad \forall \varepsilon > 0 \tag{3.21}$$

$$\delta(m) = \begin{cases} 0 & m \neq 0 \\ 1 & m = 0 \end{cases} \tag{3.22}$$

The autocorrelation of a white noise process indicates zero correlation for all lag values except at lag = 0 since it involves a δ function. Therefore, a white noise process is called sometimes a pure random process. The autocorrelation function and PSD of a white noise process are sketched in Figure 3.2.

Taking into account the above definition of the Dirac-delta function, the variance of a continuous white noise process is infinite. This implies that such a process is only a theoretical concept (the process is not physically realizable). However, white noise can be used successfully to approximate some physical processes with respect

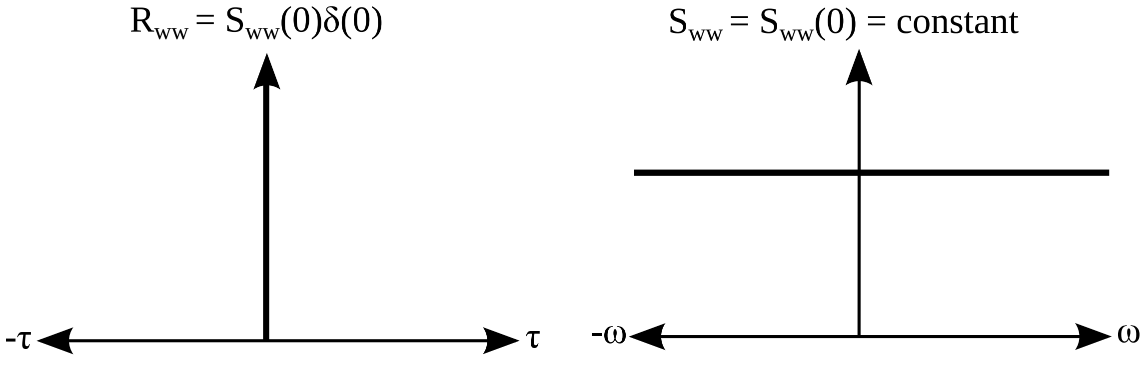


Figure 3.2: *Autocorrelation function and PSD of a white noise process*

to the bandwidth of the system. Random signals can often be modeled as filtered or shaped white noises. It means that one can filter a white noise source to achieve a colored noise source that is correlated in the time domain and band-limited in the frequency domain [129]. One can achieve other time-correlated random processes by passing a white noise through shaping filters which are discussed as follows.

3.3.3 Autoregressive process

An autoregressive (AR) process is a representation of a special kind of random process. The name autoregressive comes from the fact that each signal sample is regressed on the previous values of itself. The AR process can be described using a pole-zero transfer system ($H(z)$) as follows:

$$\mathrm{H}(z) = \frac{\mathrm{O}(z)}{\mathrm{I}(z)} = \frac{\beta_0}{1 + \sum_{n=1}^{p} a_n z^{-n}} \tag{3.23}$$

where $\mathrm{I}(z)$ is the z-transform of the input I_k, $\mathrm{O}(z)$ is the z-transform of the output O_k. β_0^2 represents the estimated variance of the white noise input to the AR model; p is the AR order.

Applying the inverse z-transform to equation (3.23), the AR sequential process in the time domain is given by:

$$\mathrm{O}_k = -\sum_{n=1}^{p} a_n \mathrm{O}_{k-n} + \beta_0 \mathrm{I}_k \tag{3.24}$$

There are different methods to estimate the parameters fitting an AR model to the input data. In [93], three different methods for the estimation of AR model parameters are investigated: the Yule-Walker method, the covariance method and the Burg's method. The Yule-Walker method determines first the sample autocorrelation sequence of the input signal (centered GNSS residual errors in our case). Then, the AR model parameters are optimally computed by solving a set of linear normal equations in a least-square sense. However, the Yule-Walker method performs adequately only for very long data records [63] and it may introduce a large bias in the estimated AR coefficients, since it does not guarantee a stable solution. The covariance method is similar to the Yule-Walker method in minimizing the prediction error in the least-squares sense. Burg's method was introduced to overcome most of the drawbacks of the other modeling techniques by providing both stable and high resolution, especially for short data records [25]. Burg's method tries to make the maximum use of the data by defining both a forward and a backward prediction error terms. As we have a limited amount of data, the Burg's method is adopted here to estimate the AR coefficients and the variance of the input white noise of the autoregressive process.

3.3.4 Random walk

The random walk process results when a random sequence is integrated. A random walk took its name by analogy with a person walking with a fixed step length (distance) in arbitrary directions. The differential equation is:

$$\dot{b}_t = w \tag{3.25}$$

where w is a purely random (white) signal. The equivalent discrete process is:

$$b_{k+1} = b_k + w_k \tag{3.26}$$

Thus, for a very large number of data samples, equation (3.26) is simply:

$$b_{k+1} = \sum_{i=1}^{k} w_i \tag{3.27}$$

Using equation (3.26), the mean of a random walk process is computed by:

$$\mu_b = E\left[b_{k+1}\right] = E\left[\sum_{i=1}^{k} w_i\right] = \sum_{i=1}^{k} E\left[w_i\right] = 0 \tag{3.28}$$

Taking into account that w_i are uncorrelated sequences, the variance is computed as:

$$\sigma_b^2 = E\left[b_{k+1}^2\right] - \mu_b^2 = E\left[b_{k+1}^2\right] = E\left[\left(\sum_{i=1}^{k} w_i\right)^2\right] = \sum_{i=1}^{k} E\left[w_i^2\right] = k\sigma_w^2 \tag{3.29}$$

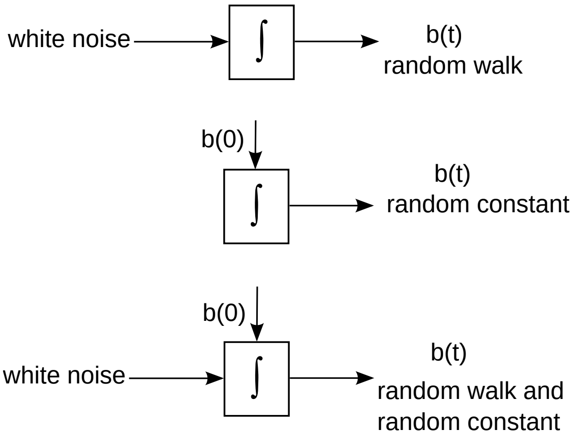


Figure 3.3: *Block diagrams of random constant and random walk processes. According to [48]*

Therefore, the random process is not stationary since its variance is changing linearly with the number of samples and hence the characteristics of the autocorrelation cannot be used to completely define the process. However, it can be considered stationary within small time intervals [6].

3.3.5 Random constant

The random constant model is also called random bias. It is both constant over time and unpredictable for any particular experiment [71]. In this case, the sensor error b_t is defined by the following differential equation:

$$\dot{b}_t = 0 \tag{3.30}$$

The discrete form of the above equation is represented by the difference equation:

$$b_{k+1} = b_k \tag{3.31}$$

Substituting equation (3.31) into equation (3.11) results in:

$$R_{bb}(m) = E\left[b_k^2\right] = R_{bb}(0) = constant \tag{3.32}$$

In the physical world, very few sensor biases remain absolutely constant. They often change slowly with time. A remedy is to insert some process noise into each of the state variables [129]. Therefore, the random constant model is often used is the one of the third diagram of Figure 3.3.

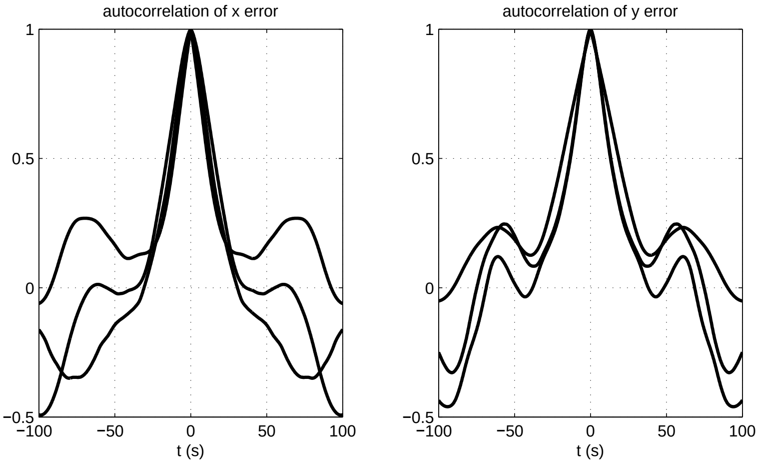


Figure 3.4: *Autocorrelation of three different sequences*

3.4 GNSS bias modeling

Outdoor tests have been made to collect GNSS data (a mono-frequency u-blox 6T receiver) for a vehicle driving in normal conditions. The GNSS errors are calculated in comparison to a ground truth system (see section 2.7) with centimeter-level accuracy. Autocorrelations of 3 different sequences of the L1-GNSS receiver (1000 samples each at 5 Hz) are shown in figure 3.4. The mean value of the GNSS fix errors is removed for each sequence. As the shape is clearly different from a Delta-Dirac at zero, errors are colored. Moreover, for short correlation time (smaller than 30 seconds), the different curves superimpose quite well which indicates a quite repeatable behavior which can be modeled.

Therefore, the correlation of the L1-GNSS error can be modeled by a zero-mean white noise passing through a shaping filter to yield an output statistically similar to the error under consideration. Let $[\varepsilon_x, \varepsilon_y]^T$ denotes the vector of GNSS errors. The problem is to find a structure for the filter and then to estimate its parameters and the variance of the driving noise. Moreover, GNSS errors are quite non-stationary (even if they have a correlation from day to day at a given place since the repetition of the satellites geometry is approximately 24 hours). So, the error model has to be quite robust with respect to this non-stationarity.

3.4.1 Autoregressive process for modeling GNSS biases

In [89][66], AR processes are used to model GNSS biases. Here, we suggest using the same AR structure to model GNSS bias correlation. Therefore, in this case, equation (3.24) becomes:

$$\begin{cases} \varepsilon_{x,k} = -\sum_{n=1}^{p} a_{x,n}\varepsilon_{x,k-n} + w_{x,k} \\ \varepsilon_{y,k} = -\sum_{n=1}^{p} a_{y,n}\varepsilon_{y,k-n} + w_{y,k} \end{cases} \tag{3.33}$$

where ε_x and ε_y are time-correlated errors on GNSS position fixes; w_x and w_y are the input white noises. Please note that this modeling supposes that the errors are not inter-correlated.

The following section discusses the determination of the AR model coefficients (a_x and a_y), the input white noises and the AR order p.

3.4.2 Discussion on the AR order

For a autoregressive model, the choice of the order p is important. In this section, the criteria for this choice is discussed.

There is rarely a direct physical motivation for choosing the AR model. The choice of p has to be based on real data analysis [78]. There has been much work on ways of making this choice with particular emphasis on automatic model selection criteria such as Akaike information criteria (AIC) [7] and the Bayes information criterion (BIC) [110]. Another method is to select an order which gives the optimal Akaike's final prediction error (FPE) [8]. A related work in modeling GNSS bias using this strategy can be found in [36].

Normalized AIC is defined as

$$\mathrm{AIC} = \ln\left(\det\left(\frac{1}{N}\sum_{t=1}^{N}\mathbf{e}\left(t,\hat{\theta}_N\right)\left(\mathbf{e}\left(t,\hat{\theta}_N\right)\right)^T\right)\right) + \frac{2n_p}{N} \tag{3.34}$$

where N is the number of samples in the estimation data set. $\hat{\theta}_N$ represents the estimated parameters. $\mathbf{e}\left(t,\hat{\theta}_N\right) = \boldsymbol{\varepsilon}(t) - \hat{\boldsymbol{\varepsilon}}\left(t|\hat{\theta}_N\right)$ is the prediction error. n_p is the number of estimated parameters. Normalized BIC is defined as

$$\mathrm{BIC} = \ln\left(\det\left(\frac{1}{N}\sum_{t=1}^{N}\mathbf{e}\left(t,\hat{\theta}_N\right)\left(\mathbf{e}\left(t,\hat{\theta}_N\right)\right)^T\right)\right) + \frac{n_p \ln(N)}{N} \tag{3.35}$$

Akaike's final prediction error (FPE) is defined as

$$\mathrm{FPE} = \det\left(\frac{1}{N}\sum_{t=1}^{N}\mathbf{e}\left(t,\hat{\theta}_N\right)\left(\mathbf{e}\left(t,\hat{\theta}_N\right)\right)^T\right)\left(\frac{N+n_p}{N-n_p}\right) \tag{3.36}$$

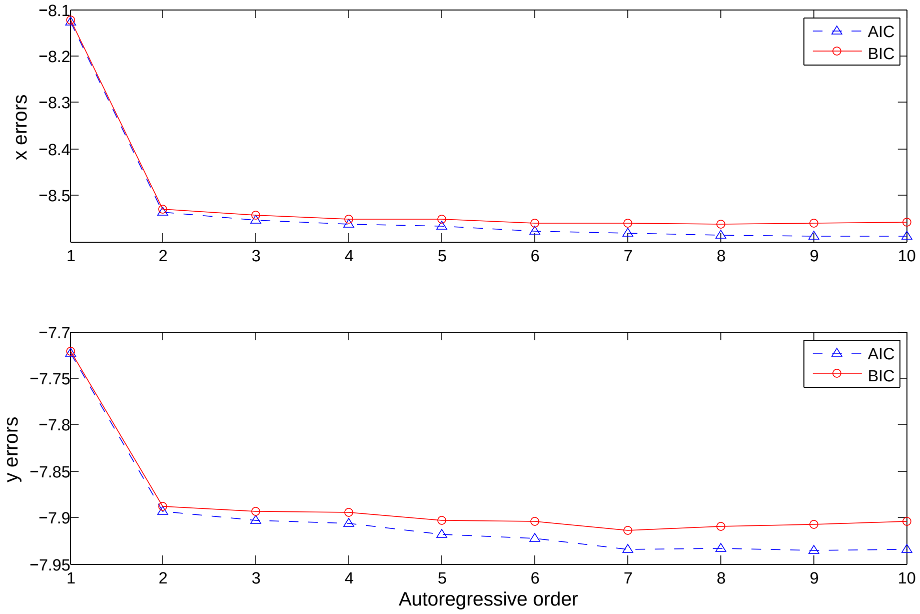


Figure 3.5: *Variation of AIC and BIC criteria*

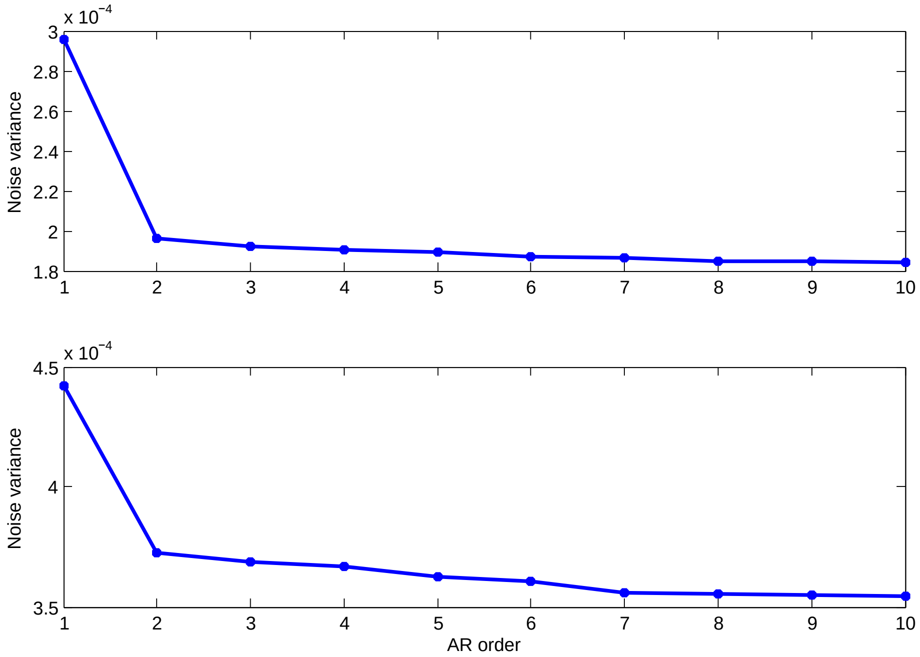


Figure 3.6: *Variation of FPEs*

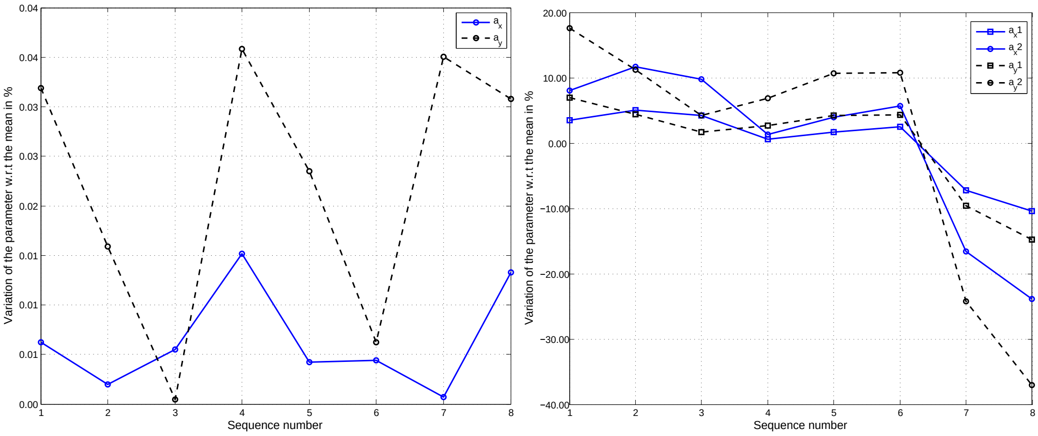


Figure 3.7: *AR coefficients estimation variation (Burg's method)*

Figures 3.5 and 3.6 show the variation of normalized AIC and BIC and FPE over order 1 to 10. One can notice that there is no significant change when the order is over 2. AIC and BIC are smaller for $p = 2$ than for $p = 1$. It means a reduced prediction error which is worth than the added variability by estimating an additional AR parameter (the higher the AR order, the more the estimated process will represent the property due to signal noise).

Considering the complexity and time efficiency of integrating the AR model into a shaping filter, the study is limited up to order 2. The AR parameters are calculated with Burg's method. Figure 3.7 shows the variation of the AR1 (first-order AR process) and AR2 (second-order AR process) coefficients respectively. One can notice that the biggest variation of the AR1 coefficients is not more than 0.05%. However, the variation of AR2 coefficients reaches nearly 40%. Even if the sequences which we have used to identify the parameters are probably too short to estimate the AR2 parameters adequately, it is clear that the stability of an AR1 model is better. In addition, the FPEs of AR1 and AR2 are of the same order of magnitude (10^{-4}). Our strategy is so to use a lower order filter but more robust to the non-stationarity which means, in other words, an AR process of order 1.

In order to validate the shaping filter, we reverse it in offline processing by using the following discrete time difference system:

$$\begin{cases} w_{x,k} = \varepsilon_{x,k} + a_x \varepsilon_{x,k-1} \\ w_{y,k} = \varepsilon_{y,k} + a_y \varepsilon_{y,k-1} \end{cases} \tag{3.37}$$

The autocorrelations of w_x and w_y are given in Figure 3.8. They approximate quite well a Delta-Dirac function. Thus, the first order AR model is good enough to get a white driving noise that one can integrate in a Kalman filter.

Remark 2. Generally, for an autoregressive model, the more model parameters, the

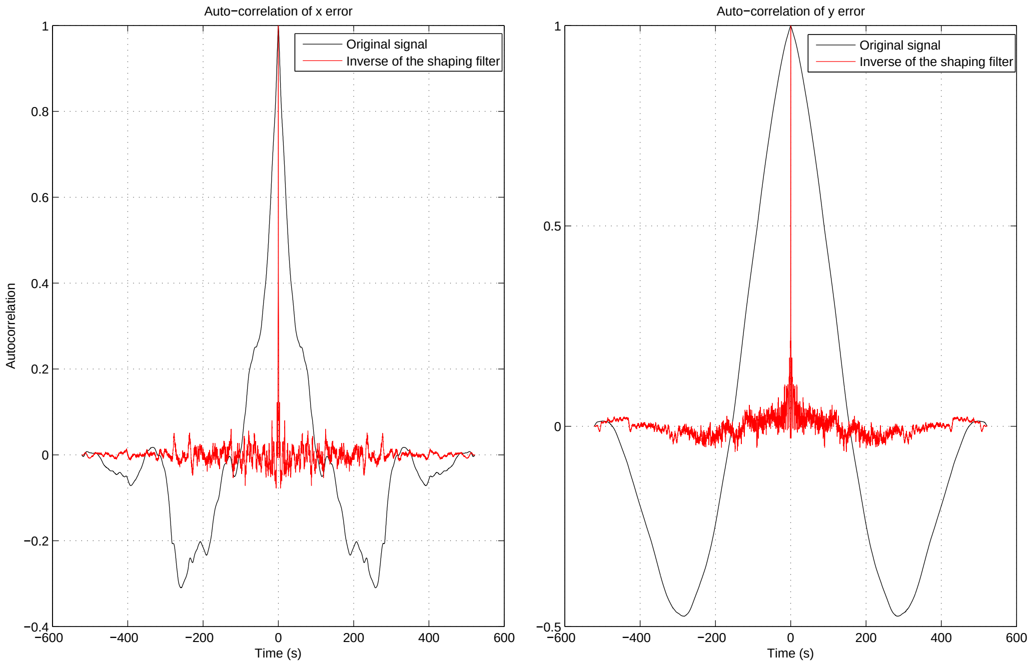


Figure 3.8: *Autocorrelation of experimental signal and the inverse of the shaping filter*

better the fitting result which means smaller fitting residual errors (see Figure 3.6). However, with the growth of the parameters, more information is needed. Thus, higher AR order results in greater parameter estimation errors or variations and less reliability of the prediction model for a given information (see Figure 3.7).

In terms of a dynamic localization problem, the GNSS bias should be estimated online. In the following section, we introduce how to integrate the sensor bias model into a Kalman filter style state observer.

3.5 Kalman filtering with shaping filters

Hereafter, the Kalman filter, shaping filter and Extended Kalman filter (EKF) are reminded. EKF is the filtering technique used in this thesis. For details about the UKF, please refer to [67][127].

3.5.1 Kalman filter reminder

A Kalman filter is an algorithm that uses a series of measurements observed over time, containing statistical noise and other inaccuracies, and produces estimates of unknown variables that tend to be more precise than those based on a single measurement alone.

A linear state model is first considered. There is a process model that expresses the transformation of the process state over time. This can be usually represented as a linear stochastic difference equation:

$$\mathbf{x}_k = \mathbf{A}\mathbf{x}_{k-1} + \mathbf{B}\mathbf{u}_{k-1} + \boldsymbol{\alpha}_{k-1} \tag{3.38}$$

where $\mathbf{A}$ is the state transition matrix, $\mathbf{B}$ is the input transition matrix, $\mathbf{u}$ is the input vector, $\boldsymbol{\alpha}$ is the process noise vector. The actual state transformation model is partially known: process models therefore incorporate some notion of random motion or uncertainty.

In addition, there is some form of measurement model that describes the relationship between the process state and the measurements. This can be represented by a linear expression:

$$\mathbf{y}_k = \mathbf{C}\mathbf{x}_k + \boldsymbol{\beta}_k \tag{3.39}$$

where $\mathbf{C}$ is the measurement matrix, $\boldsymbol{\beta}$ is the measurement noise vector.

For stochastic estimation from noisy sensor measurements, the Kalman filter [69] describes a recursive solution to the discrete-data linear filtering problem.

The process and observation noises are supposed centered which means that the process and observation models are accurate on average. It is assumed that their respective covariance matrices $\mathbf{Q}$ and $\mathbf{R}$ are known. We have:

$$\mathrm{E}\left(\boldsymbol{\alpha}_k\right) = 0,\ \mathrm{Var}\left(\boldsymbol{\alpha}_k\right) = \mathbf{Q}$$

$$\mathrm{E}\left(\boldsymbol{\beta}_k\right) = 0,\ \mathrm{Var}\left(\boldsymbol{\beta}_k\right) = \mathbf{R}$$

Often, $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ are supposed to be independent, but it is not mandatory. Please note also that the Kalman filter does not require the noises to be Gaussian distributed.

$\mathbf{P}$ is the covariance matrix of the estimation error. It is a measure of the estimated accuracy of the state estimate $\hat{\mathbf{x}}$.

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

Let denote by the subscript $_{k|k}$ the posterior estimates (for both state vector and covariance matrices) at time k given observations up to and including time k. The subscript $_{k|k-1}$ corresponds to prior estimates at time k given estimates and input vector at time $k-1$. One gets $\hat{\mathbf{x}}_{k|k}$ the state estimate, $\hat{\mathbf{x}}_{k|k-1}$ the one-step prediction, $\mathbf{P}_{k|k}$ the covariance matrix of the updated estimation error and $\mathbf{P}_{k|k-1}$ the covariance matrix of the one-step prediction error.

The algorithm works in a two-steps. In the prediction step, the Kalman filter produces predicted estimates of the current state variables, along with their uncertainties. It is calculated by:

$$\begin{aligned} \hat{\mathbf{x}}_{k|k-1} &= \mathbf{A}\hat{\mathbf{x}}_{k-1|k-1} + \mathbf{B}\mathbf{u}_{k-1} \\ \mathbf{P}_{k|k-1} &= \mathbf{A}\mathbf{P}_{k-1|k-1}\mathbf{A}^T + \mathbf{Q} \end{aligned} \tag{3.40}$$

Once the outcome of the next measurement is observed, the innovation $\boldsymbol{\mu}_k$ and its covariance matrix $\mathbf{S}$ are calculated by:

$$\boldsymbol{\mu}_k = \boldsymbol{y}_k - \mathbf{C}\hat{\mathbf{x}}_{k|k-1} \tag{3.41}$$

and

$$\mathbf{S}_k = \mathbf{C}\mathbf{P}_{k|k-1}\mathbf{C}^T + \mathbf{R} \tag{3.42}$$

The predictions $\hat{\mathbf{x}}_{k|k-1}$ and $\mathbf{P}_{k|k-1}$ are updated using a weighted average with $\boldsymbol{\mu}_k$ and $\mathbf{S}_k$, with more weight being given to estimates with higher certainty. The Kalman gain is defined by:

$$\mathbf{K}_k = \mathbf{P}_{k|k-1}\mathbf{C}^T\mathbf{S}_k^{-1} \tag{3.43}$$

The Kalman filter is a state observer for which the gain $\mathbf{K}$ is calculated so that the estimation is optimal under the principle of orthogonality [87]. Then, the state update is calculated by:

$$\begin{aligned} \hat{\mathbf{x}}_{k|k} &= \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k\boldsymbol{\mu}_k \\ \mathbf{P}_{k|k} &= \left(\mathbf{I} - \mathbf{K_k}\mathbf{C}\right)\mathbf{P}_{k|k-1}\left(\mathbf{I} - \mathbf{K_k}\mathbf{C}\right)^T + \mathbf{K}_k\mathbf{R}\mathbf{K}_k^T \end{aligned} \tag{3.44}$$

3.5.2 Kalman filtering with shaping filters for colored or biased measurement errors

In the classical Kalman filtering formalism, the state equation and sensor measurement are driven by additive, zero-mean, white and uncorrelated errors. As it is difficult to estimate the driving white noises $\boldsymbol{\alpha}$ for the system model, the matrix $\mathbf{Q}$ is often considered as a weight of the prediction from the state equations relatively to the measurements. Correspondingly, the measurement uncertainties are critical to a good filter operation [71]. Once a model of the colored measurements errors is obtained, a Kalman filter with shaping filters has to be implemented.

Let suppose that the sensor errors can be approximately modeled as the output of a white noise passing through a certain shaping filter providing a time-correlated noise. The parameters of such a shaping filter have been estimated through the minimization of the differences between the output of the shaping filter and the actual noise sequence of the sensor output in a least-square sense. The residual component is determined from a sequence of data measurements after removing its mean value. By using a shaping filter, one can obtain a noise source with desired characteristics from a wide-band source. If the residual error is perfectly modeled by the shaping filter, the measurement becomes noiseless and the observation model becomes (with a first order AR model):

$$\begin{cases} \mathbf{y}_k = \mathbf{C}\mathbf{x}_k + \boldsymbol{\varepsilon}_k \\ \boldsymbol{\varepsilon}_k = \mathbf{A}^{'}\boldsymbol{\varepsilon}_{k-1} + \boldsymbol{\alpha}_{\varepsilon,k} \end{cases} \tag{3.45}$$

with $\boldsymbol{\varepsilon}_k = \mathbf{A}^{'}\boldsymbol{\varepsilon}_{k-1} + \boldsymbol{\alpha}_{\varepsilon,k}$ the shaping model for the colored noise and α_ε the driving white noise.

In [13], the authors propose to model the measurement noise as a sum of a colored noise with a white noise $\boldsymbol{\beta}^{'}$ that accounts for other unmodeled measurement errors, considering that the shaping model can not perfectly represent the noise:

$$\begin{cases} \mathbf{y}_k = \mathbf{C}\mathbf{x}_k + \boldsymbol{\varepsilon}_k + \boldsymbol{\beta}^{'}_k \\ \boldsymbol{\varepsilon}_k = \mathbf{A}^{'}\boldsymbol{\varepsilon}_{k-1} + \boldsymbol{\alpha}_{\varepsilon,k} \end{cases} \tag{3.46}$$

For a GNSS fix measurement, equation (3.46) becomes:

$$\mathbf{y}_{GPS} = \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} \varepsilon_x \\ \varepsilon_y \end{pmatrix} + \begin{pmatrix} \beta^{'}_x \\ \beta^{'}_y \end{pmatrix} \tag{3.47}$$

and

$$\begin{cases} \varepsilon_{x,k} = -a_x\varepsilon_{x,k-1} + w_{x,k} \\ \varepsilon_{y,k} = -a_y\varepsilon_{y,k-1} + w_{y,k} \end{cases} \tag{3.48}$$

The design of the first order autoregressive GNSS bias expressed by equation (3.48) has been discussed in section 3.4. In this thesis, the noise $\left(\beta^{'}_x, \beta^{'}_y\right)$ is modeled to have the same variance as the origin GNSS errors. This is the choice made in [96].

The gyro drift is modeled by a random bias:

$$\begin{cases} \omega_k^m = \dot{\psi}_k + \varepsilon_{\omega,k} + w_{\omega,k} \\ \varepsilon_{\omega,k} = \varepsilon_{\omega,k-1} + w_{\varepsilon_\omega,k} \end{cases} \tag{3.49}$$

where w_ω and $w_{\varepsilon_\omega,k}$ are unobservable white noise.

As we will see later, the augmented state vector in our case will be:

$$\mathbf{x} = [x, y, \psi, \varepsilon_\omega, \varepsilon_x, \varepsilon_y]^T \tag{3.50}$$

$[x, y, \psi]$ is the state of interest and $[\varepsilon_\omega, \varepsilon_x, \varepsilon_y]$ is the state augmentation.

3.5.3 Extended Kalman filter

The GNSS/DR/camera multisensor fusion problem is a nonlinear filtering problem. Let us now consider a nonlinear system represented by the following equations:

$$\begin{cases} \mathbf{x}_k = f\left(\mathbf{x}_{k-1}, \mathbf{u}_{k-1}\right) + \boldsymbol{\alpha}_{k-1} \\ \mathbf{y}_k = g\left(\mathbf{x}_k\right) + \boldsymbol{\beta}_k \end{cases} \tag{3.51}$$

The EKF exploits techniques from multivariate Taylor Series expansions to linearize a model around a working point. The linearization can be performed around the previous estimate or prediction:

$$\begin{aligned} f\left(\mathbf{x}_{k-1}, \mathbf{u}_{k-1}\right) \quad = \quad & f\left(\hat{\mathbf{x}}_{k-1|k-1}, \mathbf{u}_{k-1}\right) + \left[\tfrac{\partial f}{\partial \mathbf{x}}\left(\hat{\mathbf{x}}_{k-1|k-1}, \mathbf{u}_{k-1}\right)\right]\left(\mathbf{x}_{k-1} - \hat{\mathbf{x}}_{k-1|k-1}\right) + \\ & o_2\left(\mathbf{x}_{k-1} - \hat{\mathbf{x}}_{k-1|k-1}\right) \\ g\left(\mathbf{x}_k\right) = g\left(\hat{\mathbf{x}}_{k|k-1}\right) + & \left[\tfrac{\partial f}{\partial \mathbf{x}}\left(\hat{\mathbf{x}}_{k|k-1}, \mathbf{u}_k\right)\right]\left(\mathbf{x}_k - \hat{\mathbf{x}}_{k|k-1}\right) + o_2\left(\mathbf{x}_k - \hat{\mathbf{x}}_{k|k-1}\right) \end{aligned}$$

where $o_2(\cdot)$ is a function for which:

$$\lim_{x \to 0} \frac{o_2(x)}{x^2} = 0$$

The linearization is good if the prediction error $\mathbf{x}_k - \hat{\mathbf{x}}_{k|k-1}$ and the estimation error $\mathbf{x}_{k-1} - \hat{\mathbf{x}}_{k-1|k-1}$ are small enough.

Let denote

$$
\begin{aligned}
&\mathbf{A}_{k-1} = \tfrac{\partial f}{\partial \mathbf{x}}\left(\hat{\mathbf{x}}_{k-1|k-1}, \mathbf{u}_{k-1}\right)\\
&\check{\mathbf{u}}_k = f\left(\hat{\mathbf{x}}_{k-1|k-1}, \mathbf{u}_{k-1}\right) - \mathbf{A}_{k-1}\hat{\mathbf{x}}_{k-1|k-1}\\
&\mathbf{C}_k = \tfrac{\partial g}{\partial \mathbf{x}}\left(\hat{\mathbf{x}}_{k|k-1}\right)\\
&\mathbf{z}_k = \mathbf{y}_k - g\left(\hat{\mathbf{x}}_{k|k-1}\right) + \mathbf{C}_k\hat{\mathbf{x}}_{k|k-1}
\end{aligned}
$$

Then the nonlinear system is transformed into a linear system:

$$
\begin{cases}
\mathbf{x}_k = \mathbf{A}_{k-1}\mathbf{x}_{k-1} + \check{\mathbf{u}}_k + \boldsymbol{\alpha}_{k-1}\\
\mathbf{z}_k = \mathbf{C}_k\mathbf{x}_k + \boldsymbol{\beta}_k
\end{cases}
\tag{3.52}
$$

The prediction step is

$$
\begin{aligned}
&\hat{\mathbf{x}}_{k|k-1} = \mathbf{A}_{k-1}\hat{\mathbf{x}}_{k-1|k-1} + \check{\mathbf{u}}_k = f\left(\mathbf{x}_{k-1|k-1}, \mathbf{u}_{k-1}\right)\\
&\mathbf{P}_{k|k-1} = \mathbf{A}_{k-1}\mathbf{P}_{k-1|k-1}\mathbf{A}_{k-1}^T + \mathbf{Q}
\end{aligned}
\tag{3.53}
$$

The update step is

$$
\begin{aligned}
&\hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k\boldsymbol{\mu}_k\\
&\mathbf{P}_{k|k} = \left(\mathbf{I} - \mathbf{K}_k\mathbf{C}_k\right)\mathbf{P}_{k|k-1}\left(\mathbf{I} - \mathbf{K}_k\mathbf{C}_k\right)^T + \mathbf{K}_k\mathbf{R}\mathbf{K}_k^T
\end{aligned}
\tag{3.54}
$$

with

$$
\begin{aligned}
&\boldsymbol{\mu}_k = \boldsymbol{z}_k - \mathbf{C}_k\hat{\mathbf{x}}_{k|k-1} = \mathbf{y}_k - g\left(\hat{\mathbf{x}}_{k|k-1}\right)\\
&\mathbf{K}_k = \mathbf{P}_{k|k-1}\mathbf{C}_k^T\left(\mathbf{C}_k\mathbf{P}_{k|k-1}\mathbf{C}_k^T + \mathbf{R}\right)
\end{aligned}
\tag{3.55}
$$

3.5.4 Extended Kalman filter with measured input

For some systems, the inputs are measured. The measurements noises on the inputs have to be considered.

A nonlinear system with measured input is expressed by:

$$
\begin{cases}
\mathbf{x}_k = f\left(\mathbf{x}_{k-1}, \mathbf{u}_{k-1}^m\right) + \boldsymbol{\alpha}_{k-1}\\
\mathbf{u}_{k-1}^m = \mathbf{u}_{k-1} + \boldsymbol{\gamma}_{k-1}\\
\mathbf{y}_k = g\left(\mathbf{x}_k\right) + \boldsymbol{\beta}_k
\end{cases}
\tag{3.56}
$$

In this case, the system input $\mathbf{u}_{k-1}$ is measured with a measurement noise $\boldsymbol{\gamma}_{k-1}$, with known covariance matrix $\text{Cov}\left(\boldsymbol{\gamma}_{k-1}\right) = \mathbf{N}$. The measured input vector $\mathbf{u}_{k-1}^m$ is applied to the system. $\boldsymbol{\alpha}$, $\boldsymbol{\beta}$, and $\boldsymbol{\gamma}$ are supposed to be white, centered, uncorrelated with each other and independent of the state. Performing a Taylor extension to the first order of function $f(\cdot)$ around $\hat{\mathbf{x}}_{k-1|k-1}$ and measured input $\mathbf{u}^m_{k-1}$

$$\begin{aligned} f\left(\mathbf{x}_{k-1},\mathbf{u}^m_{k-1}\right) &= f\left(\hat{\mathbf{x}}_{k-1|k-1},\mathbf{u}^m_{k-1}\right)+\left[\frac{\partial f}{\partial \mathbf{x}}\left(\hat{\mathbf{x}}_{k-1|k-1},\mathbf{u}^m_{k-1}\right)\right]\left(\mathbf{x}_{k-1}-\hat{\mathbf{x}}_{k-1|k-1}\right)+\\ &\quad\left[\frac{\partial f}{\partial \mathbf{u}}\left(\hat{\mathbf{x}}_{k-1|k-1},\mathbf{u}^m_{k-1}\right)\right]\boldsymbol{\gamma}_{k-1}+o_2\left(\mathbf{x}_{k-1}-\hat{\mathbf{x}}_{k-1|k-1},\boldsymbol{\gamma}_{k-1}\right)\\ g\left(\mathbf{x}_k\right) = g\left(\hat{\mathbf{x}}_{k|k-1}\right) &+ \left[\frac{\partial f}{\partial \mathbf{x}}\left(\hat{\mathbf{x}}_{k|k-1},\mathbf{u}_k\right)\right]\left(\mathbf{x}_k-\hat{\mathbf{x}}_{k|k-1}\right)+o_2\left(\mathbf{x}_k-\hat{\mathbf{x}}_{k|k-1}\right)\end{aligned}$$

Let denote

$$\begin{aligned} \mathbf{A}_{k-1} &= \frac{\partial f}{\partial \mathbf{x}}\left(\hat{\mathbf{x}}_{k-1|k-1},\mathbf{u}_{k-1}\right)\\ \mathbf{B}_{k-1} &= \frac{\partial f}{\partial \mathbf{u}}\left(\hat{\mathbf{x}}_{k-1|k-1},\mathbf{u}^m_{k-1}\right)\\ \mathbf{C}_k &= \frac{\partial g}{\partial \mathbf{x}}\left(\hat{\mathbf{x}}_{k|k-1}\right)\end{aligned}$$

The nonlinear equation (3.56) is linearized as:

$$\begin{cases} \mathbf{x}_k = \mathbf{A}_{k-1}\mathbf{x}_{k-1}+\check{\mathbf{u}}_k+\mathbf{B}_{k-1}\boldsymbol{\gamma}_{k-1}+\boldsymbol{\alpha}_{k-1}\\ \mathbf{z}_k = \mathbf{C}_k\mathbf{x}_k+\boldsymbol{\beta}_k\end{cases} \tag{3.57}$$

and then a linear Kalman filter can be applied with a slightly different model noise and, in the way as before, we can use directly the non linear functions of the original state space.

Initialize states and covariance by $\mathbf{x}=\mathbf{x}_0$, $\mathbf{P}=\mathbf{P}_0$. For $k=1,2,....$, the EKF with measured input is expressed by algorithm 3.1.

Algorithm 3.1 Extended Kalman filter with measured input

```
$\mathbf{u}^m_{k-1} \leftarrow$ measured inputs
$\hat{\mathbf{x}}_{k|k-1} = f\left(\mathbf{x}_{k-1|k-1},\mathbf{u}^m_{k-1}\right)$
$\mathbf{A}_{k-1} = \frac{\partial f}{\partial \mathbf{x}}\left(\hat{\mathbf{x}}_{k-1|k-1},\mathbf{u}_{k-1}\right), \mathbf{B}_{k-1} = \frac{\partial f}{\partial \mathbf{u}}\left(\hat{\mathbf{x}}_{k-1|k-1},\mathbf{u}^m_{k-1}\right)$
$\mathbf{P}_{k|k-1} = \mathbf{A}_{k-1}\mathbf{P}_{k-1|k-1}\mathbf{A}^T_{k-1}+\mathbf{B}_{k-1}\mathbf{N}\mathbf{B}^T_{k-1}+\mathbf{Q}$
if Output Measurements are available then
    $\mathbf{y}_k \leftarrow$ sensor measurements
    $\mathbf{C}_k = \frac{\partial g}{\partial \mathbf{x}}\left(\hat{\mathbf{x}}_{k|k-1}\right), \boldsymbol{\mu}_k = \mathbf{y}_k - g\left(\hat{\mathbf{x}}_{k|k-1}\right)$
    $\mathbf{K}_k = \mathbf{P}_{k|k-1}\mathbf{C}^T_k\left(\mathbf{C}_k\mathbf{P}_{k|k-1}\mathbf{C}^T_k+\mathbf{R}\right)$
    $\hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k|k-1}+\mathbf{K}_k\boldsymbol{\mu}_k$
    $\mathbf{P}_{k|k} = \left(\mathbf{I}-\mathbf{K}_k\mathbf{C}_k\right)\mathbf{P}_{k|k-1}\left(\mathbf{I}-\mathbf{K}_k\mathbf{C}_k\right)^T+\mathbf{K}_k\mathbf{R}\mathbf{K}^T_k$
end if
Go to 1
```

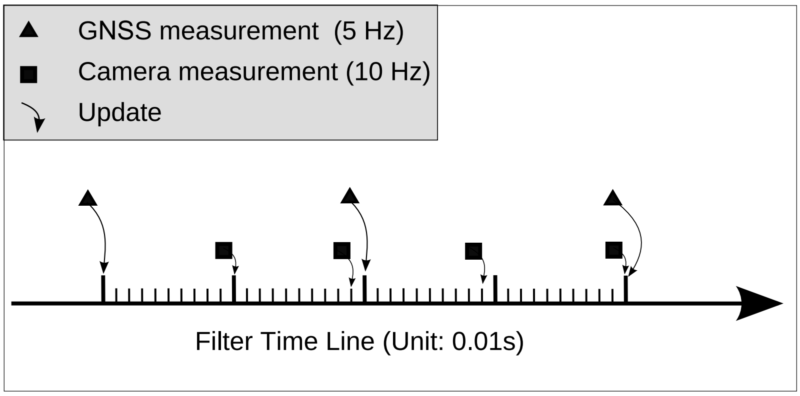


Figure 3.9: *Filter time line*

3.6 Localization solver

The localization solver which merges all the sensors information and the lane marking map is realized by an EKF with measured input and shaping filters for GNSS errors and gyro biases. The wheel speed sensor measurement (v^m) and gyro measurement (ω^m) from the CAN bus are used as measured input.

The filter is implemented as a discrete EKF triggered by the proprioceptive sensors (typically at a sampling period of 0.01s). GNSS and camera measurements are used as soon as they are available in a asynchronous way (see Figure 3.9 for an illustration).

Figure 3.10 displays the structure of the localization solver: $\hat{\mathbf{x}}$ is the estimated state and $\mathbf{P}$ is the estimated covariance matrix. Depending on the step, they can correspond to predictions or updates but, in terms of implementation, there are the same variables. As said before, the localization solver works at the frequency of the DR sensors (from ABS and ESP systems). A unicycle model is used to dead reckon the vehicle pose. When a GNSS fix is available, it is first validated or rejected by an innovation gating then, if validated, it is used to update the vehicle state. When a camera C_0 measurement is available, a map matching process is performed to find which lane marking in the map database is detected. Only when the detected lane marking is map matched, the C_0 observation model is employed. Then, the vehicle state is updated by C_0 with an innovation gating similar to the one used for GNSS. More details about prediction, GNSS/camera corrections and innovation gating techniques are as follows.

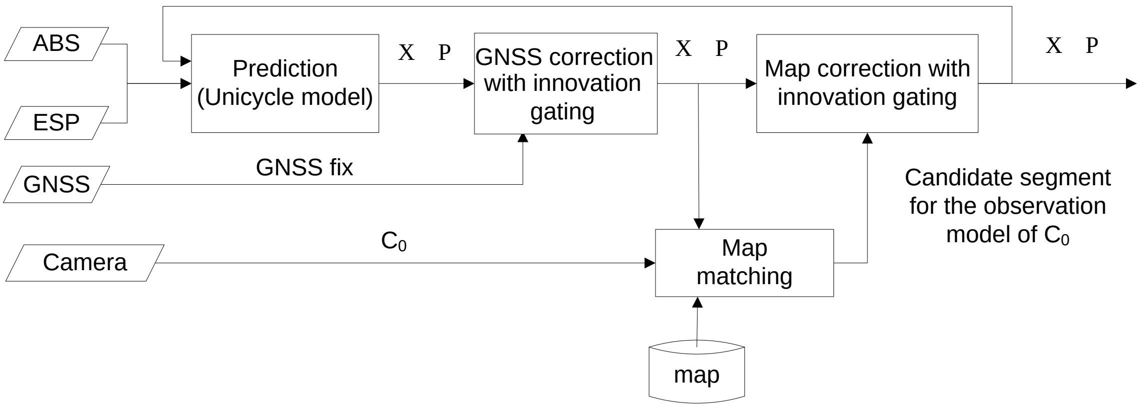


Figure 3.10: *Diagram of the localization solver*

3.6.1 Prediction stage

The continuous state evolution model is built around a unicycle model with the shaping models:

$$\begin{cases} \dot{x} = v^m \text{cos}\psi \\ \dot{y} = v^m \text{sin}\psi \\ \dot{\psi} = \omega^m - \varepsilon_\omega \\ \dot{\varepsilon_x} = -\varepsilon_x/\tau_x \\ \dot{\varepsilon_y} = -\varepsilon_y/\tau_y \\ \dot{\varepsilon_\omega} = 0 \end{cases} \tag{3.58}$$

A CAN-bus gateway is used to access to the wheel speed sensors and to the yaw rate gyro at 100 Hz. The prediction can be decomposed into two parts: prediction of the state vector and prediction of the associated covariance matrix of the state vector.

At instant k, the prediction of the state is realized by the following discrete equation:

$$\begin{cases} x_{k|k-1} = x_{k-1|k-1} + T_e v^m_{k-1} \text{cos}\psi_{k-1|k-1} \\ y_{k|k-1} = y_{k-1|k-1} + T_e v^m_{k-1} \text{sin}\psi_{k-1|k-1} \\ \psi_{k|k-1} = \psi_{k-1|k-1} + T_e(\omega^m_{k-1} - \varepsilon_{\omega,k-1|k-1}) \\ \varepsilon_{x,k|k-1} = a_x \varepsilon_{x,k-1|k-1} \\ \varepsilon_{y,k|k-1} = a_y \varepsilon_{y,k-1|k-1} \\ \varepsilon_{\omega,k|k-1} = \varepsilon_{\omega,k-1|k-1} \end{cases} \tag{3.59}$$

$$\mathbf{P}_{k|k-1} = \mathbf{A}_{k-1}\mathbf{P}_{k-1|k-1}\mathbf{A}^T_{k-1} + \mathbf{B}_{k-1}\mathbf{N}\mathbf{B}^T_{k-1} + \mathbf{Q} \tag{3.60}$$

The Jacobian matrix $\mathbf{A}_{k-1}$, consisting of the partial derivatives of equation (5.10) with respect to the state vector, is given by:

$$\mathbf{A}_{k-1} = \begin{bmatrix} 1 & 0 & -T_e v_{k-1} \sin\psi_{k-1|k-1} & 0 & 0 & 0 \\ 0 & 1 & -T_e v_{k-1} \cos\psi_{k-1|k-1} & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & -Te \\ 0 & 0 & 0 & a_x & 0 & 0 \\ 0 & 0 & 0 & 0 & a_y & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.61}$$

$\mathbf{B}_{k-1}$ is the Jacobian matrix of equation (5.10) with respect to the input vector $\left[v_{k-1}^m, \omega_{k-1}^m\right]^T$:

$$\mathbf{B}_{k-1} = \begin{bmatrix} T_e \cdot cos\psi_{k-1|k-1} & 0 \\ T_e \cdot \sin\psi_{k-1|k-1} & 0 \\ 0 & T_e \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \tag{3.62}$$

3.6.2 GNSS correction

The GNSS receiver provides position of the vehicle at $5Hz$. If the GNSS information is available at instant k, the state vector is updated once. By considering GNSS biases and the lever arm of the antenna is the body frame, the observation model expressed by equation (2.26) changes to:

$$\begin{cases} x_{GNSS} = \cos\psi \cdot t_x - \sin\psi \cdot t_y + x + \varepsilon_x + \beta_x^{'} \\ y_{GNSS} = \sin\psi \cdot t_x + \cos\psi \cdot t_y + y + \varepsilon_y + \beta_y^{'} \end{cases} \tag{3.63}$$

By denoting the GNSS measurement as $\mathbf{y}_{GNSS} = \left[x_{GNSS}, y_{GNSS}\right]^T$, the update equations of the state vector and covariance matrix are:

$$\begin{cases} \mathbf{x}_{k|k} = \mathbf{x}_{k|k-1} + \mathbf{K}_{GNSS}\left[\mathbf{y}_{GNSS} - g_{GNSS}\left(\mathbf{X}_{k|k-1}\right)\right] \\ \mathbf{P}_{k|k} = \left[\mathbf{I} - \mathbf{K}_{GNSS}\mathbf{C}_{GNSS}\right]\mathbf{P}_{k|k-1}\left[\mathbf{I} - \mathbf{K}_{GNSS}\mathbf{C}_{GNSS}\right]^T + \mathbf{K}_{GNSS}\mathbf{R}_{GNSS}\mathbf{K}_{GNSS}^T \end{cases} \tag{3.64}$$

with the GNSS observation matrix:

$$\mathbf{C}_{GNSS} = \begin{bmatrix} 1 & 0 & -t_x \cdot \sin\psi_{k|k-1} - t_y \cdot \cos\psi_{k|k-1} & 1 & 0 & 0 \\ 0 & 1 & t_x \cdot \cos\psi_{k|k-1} - t_y \cdot \sin\psi_{k|k-1} & 0 & 1 & 0 \end{bmatrix} \tag{3.65}$$

The covariance matrix $\mathbf{R}_{GNSS}$ of measurement noise $\left(\beta_x^{'}, \beta_y^{'}\right)$ is defined as follows

$$\mathbf{R}_{GNSS} = \text{diag}\left[\sigma^2_{x_{GNSS}}, \sigma^2_{y_{GNSS}}\right] \tag{3.66}$$

$\sigma^2_{x_{GNSS}}$ and $\sigma^2_{y_{GNSS}}$ are available from the GNSS receiver computation.
The Kalman gain is:

$$\mathbf{K}_{GNSS} = \mathbf{P}_{k|k-1}\mathbf{C}^T_{GNSS}\left[\mathbf{C}_{GNSS}\mathbf{P}_{k|k-1}\mathbf{C}^T_{GNSS} + \mathbf{R}_{GNSS}\right]^{-1} \tag{3.67}$$

3.6.3 Camera correction

Map matching consists to determine which is the lane marking segment that the camera has detected. In this thesis, the marking type given by the video camera is also used to map-match the detected lane marking segment. In a first stage, a set S of candidate segments is selected based on the following conditions:

- The lane marking type is consistent with the one estimated by the camera.
- The orientation of the segment is close to the heading of the vehicle.
- The distance d between the point L (equation (2.28)) and the candidate segment is less than the width of the road.

In a second stage, the segment s which has the minimal distance is chosen as the map matching result:

$$\text{map matched segment} = \arg\min_{s\in S}\{d\} \tag{3.68}$$

The left and right side camera measurements are used sequentially. Camera observation function $g_{cam}(\cdot)$ is defined by equation (2.32). Denoting $\mathbf{y}_{cam} = [C_0]^T$, the update equations of the state vector and covariance matrix are:

$$\begin{cases} \mathbf{x}_{k|k} = \mathbf{x}_{k|k-1} + \mathbf{K}_{cam}\left[\mathbf{y}_{cam} - g_{cam}\left(\mathbf{X}_{k|k-1}\right)\right] \\ \mathbf{P}_{k|k} = \left(\mathbf{I} - \mathbf{K}_{cam}\mathbf{C}_{cam}\right)\mathbf{P}_{k|k-1}\left(\mathbf{I} - \mathbf{K}_{cam}\mathbf{C}_{cam}\right)^T + \mathbf{K}_{cam}\mathbf{R}_{cam}\mathbf{K}^T_{cam} \end{cases} \tag{3.69}$$

with the observation matrix:

$$\mathbf{C}_{cam} = \frac{\partial g_{cam}}{\partial \mathbf{x}} |_{\mathbf{x}=\mathbf{x}_{k|k-1}} \tag{3.70}$$

Kalman gain:

$$\mathbf{K}_{cam} = \mathbf{P}_{k|k-1}\mathbf{C}^T_{cam}\left(\mathbf{C}_{cam}\mathbf{P}_{k|k-1}\mathbf{C}^T_{cam} + \mathbf{R}_{cam}\right)^{-1} \tag{3.71}$$

with $\mathbf{R}_{cam} = Var\left(\boldsymbol{\beta}_{cam}\right)$.

3.6.4 Innovation gating

As one can see in Figure 3.10, before using a GNSS fix to update the state vector, an innovation gating strategy is adopted. Here, the innovation is evaluated based on the Mahalanobis distance:

$$D_{GNSS} = \left(\mathbf{y}_{GNSS} - g_{GNSS}\left(\mathbf{x}_{k|k-1}\right)\right)^T \left(\mathbf{C}_{GNSS}\mathbf{P}_{k|k-1}\mathbf{C}_{GNSS}^T + \mathbf{R}_{GNSS}\right)^{-1} \left(\mathbf{y}_{GNSS} - g_{GNSS}\left(\mathbf{x}_{k|k-1}\right)\right) \tag{3.72}$$

Only when the Mahalanobis distance is less than a rejection threshold, the GNSS fix can be used to update the state vector and the covariance matrix. For the camera measurements, the same strategy is adopted. The corresponding Mahalanobis distance is :

$$D_{cam} = \left(\mathbf{y}_{cam} - g_{cam}\left(\mathbf{x}_{k|k-1}\right)\right)^T \left(\mathbf{C}_{cam}\mathbf{P}_{k|k-1} \cdot \mathbf{C}_{cam}^T + \mathbf{R}_{cam}\right)^{-1} \left(\mathbf{y}_{cam} - g_{cam}\left(\mathbf{x}_{k|k-1}\right)\right) \tag{3.73}$$

These integrity tests are useful to exclude GNSS outliers, wrong lane marking detections and mismatches.

3.7 Experimental results

For the initialization of the state vector and the covariance matrix, there are two modes:

- Cold start mode:
  - Initialize all the elements with 0.
  - Wait for a GNSS fix with valid track angle.
  - Initialize the position and heading $\mathbf{x} = [x_{GNSS,1}, y_{GNSS,1}, \psi_1, 0, 0, 0, 0, 0]^T$.
  - Start the EKF which needs some time to converge.
- Warm start mode: initialize all the elements of the EKF with the corresponding values stored by the system in the pose file when it stopped working the last time.
  - The EKF is immediately operational and works well if the vehicle has not been moved and if the stopped duration was short (because of the GNSS error modeling)

In this thesis, a warm start mode is adopted. The vehicle is initialized with the reference value (provided by the ground truth system) and an enlarged covariance matrix. The localization solver has been tested using data replay. Figure 3.11 displays the trajectory of the EKF estimates. The estimated trajectory is in 4 different colors which indicate different sensor fusion modes. There are 4 modes:

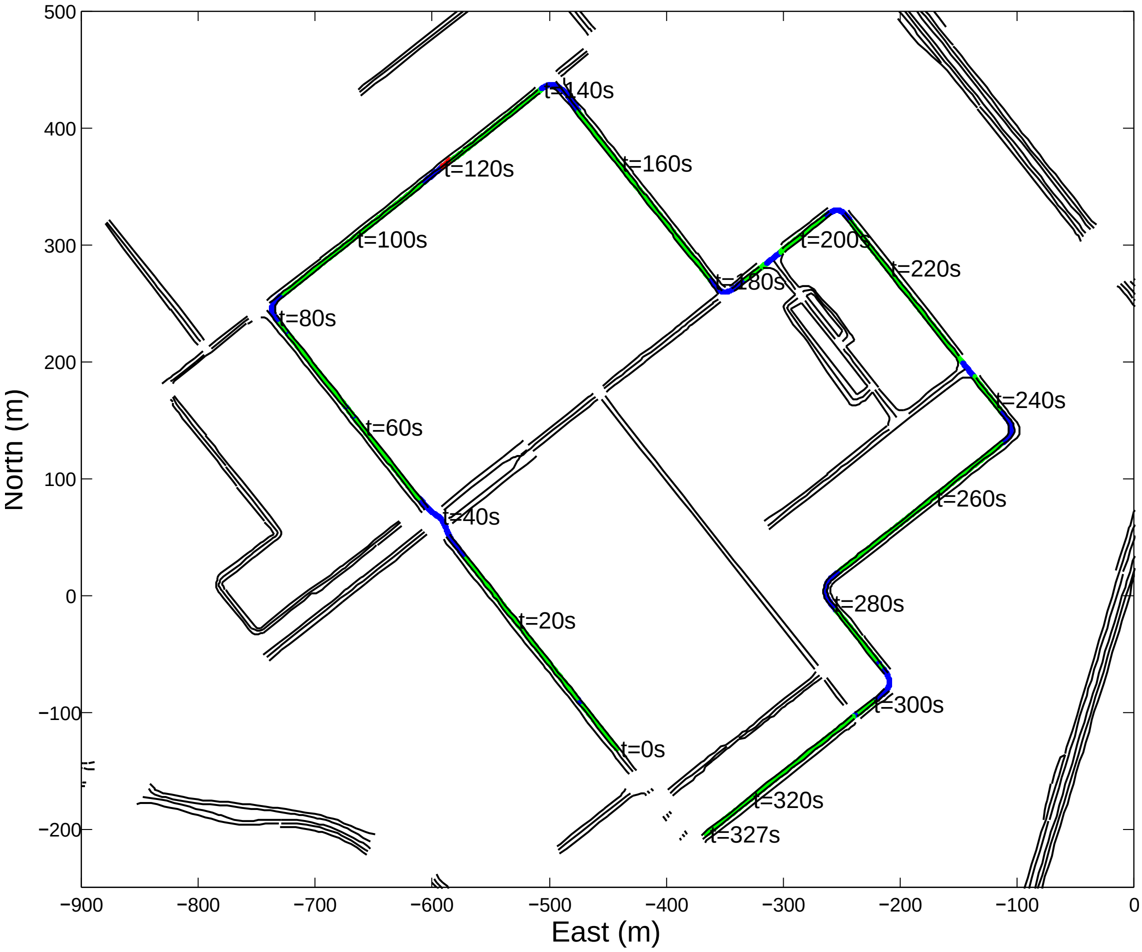


Figure 3.11: *Different sensor fusion modes for an experiment*

- Mode 1 (in red): dead reckoning.
- Mode 2 (in yellow): DR coupled with camera and lane marking map.
- Mode 3 (in blue): DR coupled with GNSS.
- Mode 4 (in green): DR coupled with GNSS, camera and lane marking map.

Take mode 2 as an example, data from DR sensors is used for prediction. Camera measurement C_0 coupled with lane marking map is used to update vehicle state. GNSS measurements are not available or rejected by the localization solver. Considering output frequency of the solver is higher than the camera update, if $\mathbf{y}_{cam,k-1}$ is used in the solver, instances between $\mathbf{y}_{cam,k-1}$ and $\mathbf{y}_{cam,k}$ are considered as working in a mode with camera measurements available. The same is done for deciding the GNSS availability. Mode 3 usually happens around the road corners where lane marking detection fails.

Figure 3.12 shows the obtained lateral and longitudinal positioning errors (PE). Figure 3.13 displays the heading error. The maximum of the heading error is smaller

	Lateral PE (m)		Longitudinal PE (m)	
	I	II	I	II
mean	1.30	0.07	1.55	-0.32
std. dev.	1.12	0.29	1.18	0.32
median	0.96	0.10	1.31	0.30
95th percentile	3.20	0.68	3.88	0.88
max	6.78	1.83	4.69	1.50

Table 3.1: *Error statistics: I: u-blox; II: ENU EKF*

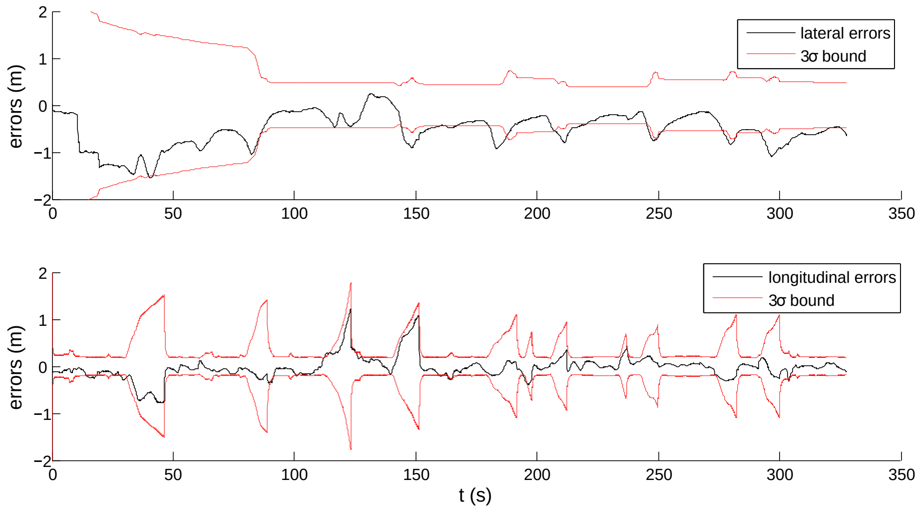
than 5 degree. The red line is the estimated $\pm 3\sigma$ bound. Three tests with the same filter tuning produce comparable results which demonstrates the stability of the localization solver.

Table 3.1 compares several accuracy metrics of the localization solver with a standalone u-blox. As expected, the lateral localization is highly improved, because the camera observation is relative to the lateral distance between the vehicle and the detected lane marking. The longitudinal accuracy is also improved by the camera system due to the heading variation of the path. The 95 percentiles of lateral and longitudinal positioning errors are both less than 1 m.

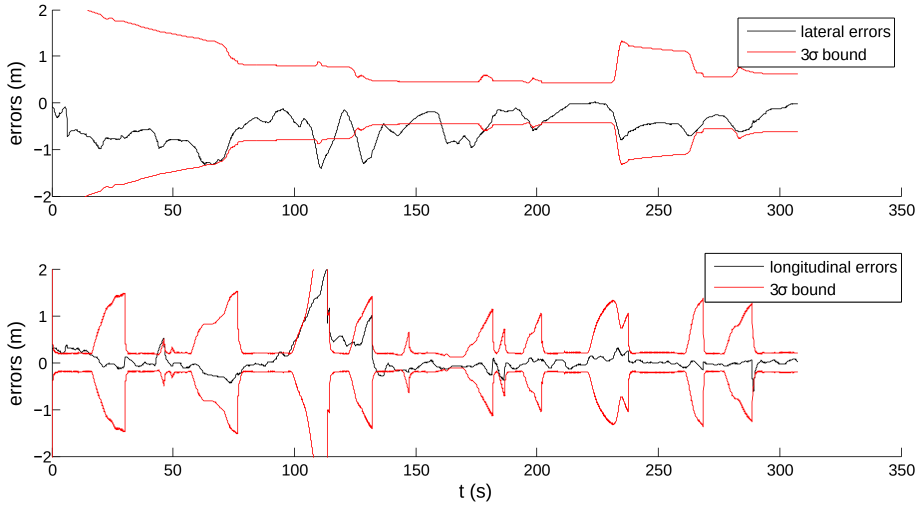
In addition to the accuracy, consistency with the estimated bound is also an important issue for a state observer. One can get a first investigation by looking at Figures 3.12 and 3.13 in which most errors are well bounded. We will give more details about consistency in Chapter 6 by comparing this filter with two other approaches.

Figure 3.14 displays several special cases where the vehicle arrives at intersections or when GNSS multipath happens. Figure 3.14a shows the vehicle trajectory around a corner. The white lines are the lane markings in the digital map. The black line is the reference trajectory and the cyan line is standalone GNSS. One can notice that, standalone GNSS has large but stable biases. Around t = 79s, the vehicle begins a corner turning and loses camera measurement. The system is working in a DR/GNSS mode. Thanks to the shaping filtered GNSS biases, the vehicle is always localized in the correct lane.

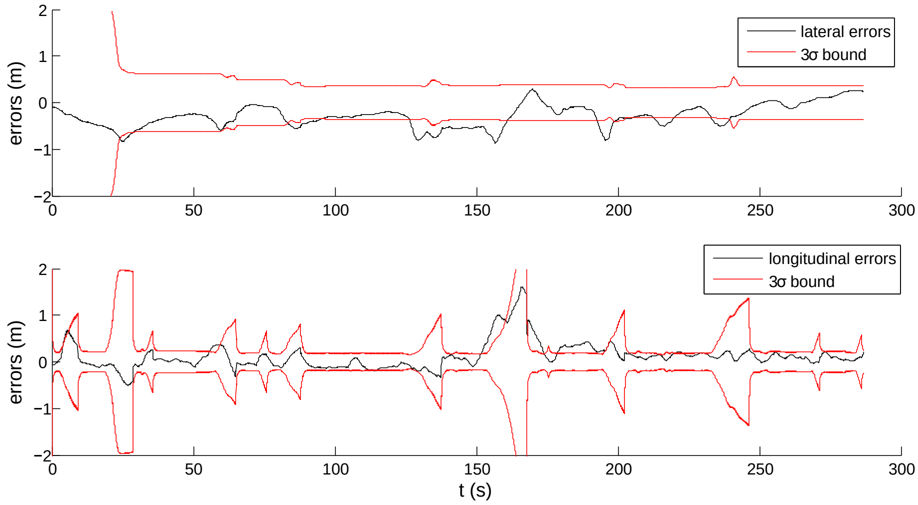
During t = 111s-122s, the vehicle enters the urban canyon environment. Figure 3.14b depicts the estimated trajectory. The localization system loses the camera measurement around t = 112s (The MobilEye is not able to output lane marking estimation with $C_{quality} > 1$). The fusion mode changes from mode 4 to mode 3. Around t = 119s, in addition to the camera measurement outage, the error of GNSS changes greatly and the GNSS measurement is not used any more. The fusion mode changes to mode 1. Around t = 122s, the camera retrieves lane marking detection, the fusion mode changes from mode 1 to mode 2 and then to mode 4. One can notice



(a) *Test 1*

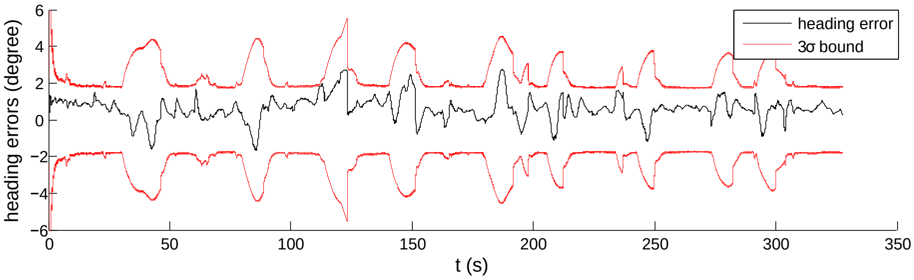


(b) *Test 2*

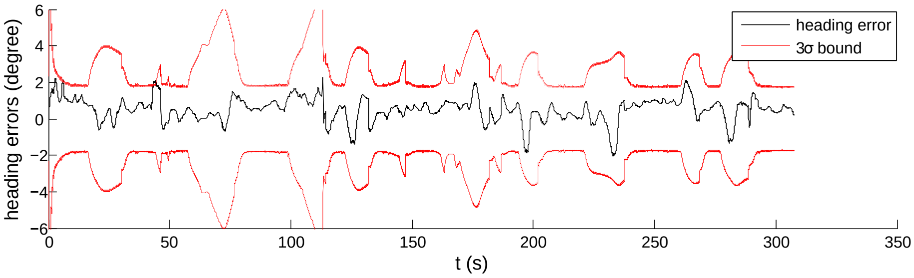


(c) *Test 3*

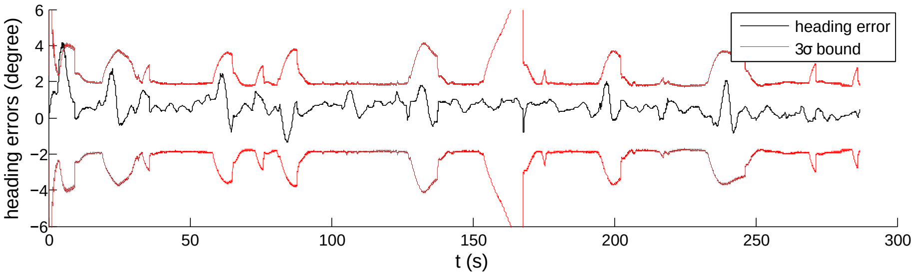
Figure 3.12: *Lateral and longitudinal positioning errors*



(a) *Test 1*

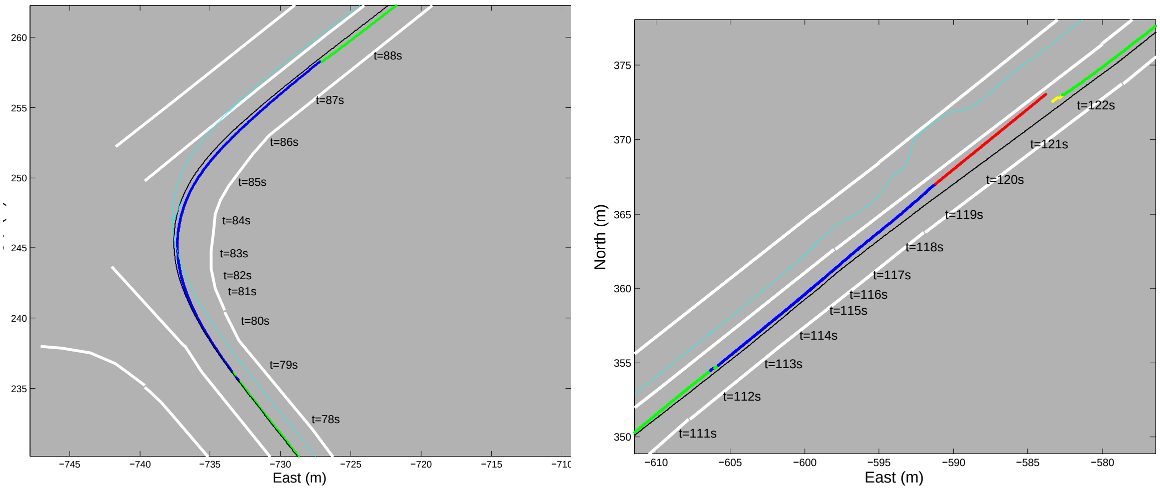


(b) *Test 2*

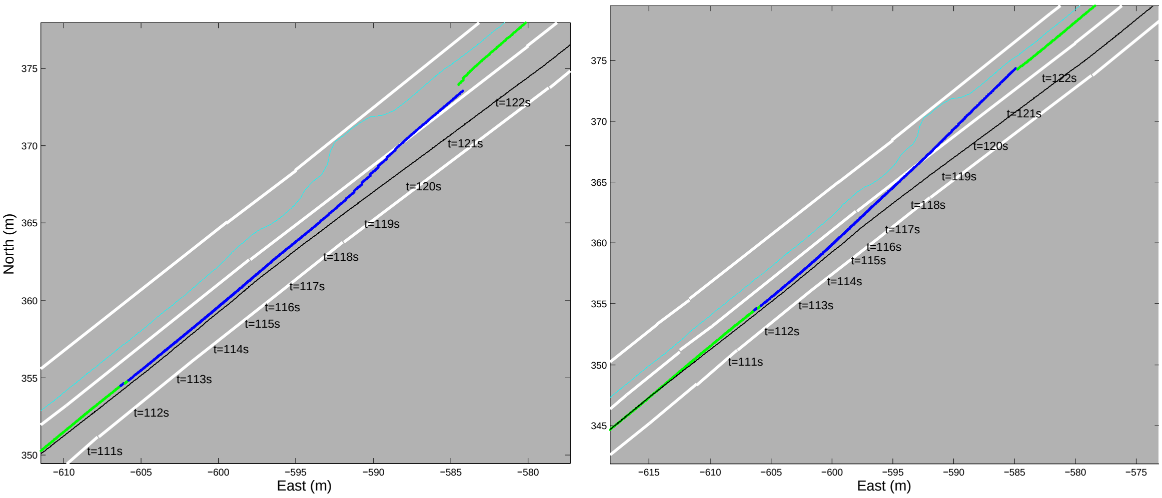


(c) *Test 3*

Figure 3.13: *Heading errors.The max of the heading error doesn't exceed 5 degree.*



(a) *Performance during turning a corner* **(b)** *The vehicle is localized in the correct lane*



(c) *The vehicle is localized in an incorrect lane* **(d)** *EKF without shaping filter*

Figure 3.14: *Different scenarios of the localization results. The colors indicates 4 different working modes of the localization solver. Mode 1 (in red): dead reckoning; mode 2 (in yellow): DR coupled with camera and map yellow; mode 3 (in blue): DR coupled with GNSS; mode 4 (in green): DR coupled with GNSS, camera and map. White lines are lane markings. Black line is the true trajectory. GNSS trajectory is in cyan. The traveling direction is indicated by the tagged time.*

that there is an abrupt but accurate lateral correction when the camera measurement is available. That is due to the high accuracy of the camera measurement.

Figure 3.14c shows the result with a different threshold for the GNSS measurements. One can notice that, during t = 111s-122s, the GNSS measurement is always used. This results that the vehicle is localized at the wrong lane at t = 122s when the camera measurements are mismatched. This indicates that an appropriate threshold has to be chosen for GNSS measurements. This choice is generally difficult, considering the GNSS bias due to multipath effect may change slowly and our bias model doesn't take it into consideration. Remedies for this critical situation should be improving the availability of the lane marking measurement or reducing the drift rate of the dead reckoning system but will not be discussed in this thesis which focuses on a low-cost solution.

The result always converges toward the biased GNSS position when there is no camera measurement for a long term. In this case a wrong map matching is possible. A possible solution is to use the lane type information provided by the camera to disambiguate which lane the vehicle is in when camera retrieves measurement with a high position uncertainty. But as indicated in Table 2.4 in Chapter 2, the lane marking type information is not reliable enough for determining the lane with only one camera detection. We will discuss a solution using lane marking type information in Chapter 6.

For comparison purpose, Figure 3.14d plots the estimated trajectory by a localization solver without the shaping filter. When the system loses camera measurement, the vehicle position converges faster to the biased GNSS fix than the result in Figure 3.14b and 3.14c.

Hereafter we look at the estimated biases using the shaping filters both for GNSS and for the gyro.

Take test 1 as an example, Figure 3.15 demonstrates the estimated biases compared with their reference value calculated by using a RTK-GNSS receiver. Figure 3.16 displays the errors of bias estimate with $\pm 3\sigma$. Around t = 122s, the GNSS bias changes rapidly and camera measurement is not available. Therefore, it occurs a large estimation error of GNSS bias. Generally, estimated biases have similar dynamic than the true biases and the estimation error is well bounded.

Figure 3.17 shows the estimated gyro bias and its 3σ error bound. The gyro bias converges to -0.003 rad/s. Since there is no ground truth for the gyro bias, we cannot decide whether this bias is well estimated or its estimation error is well bounded. Nevertheless we can conclude that the proposed gyro bias model is observable as shown by the convergence of the error bound.

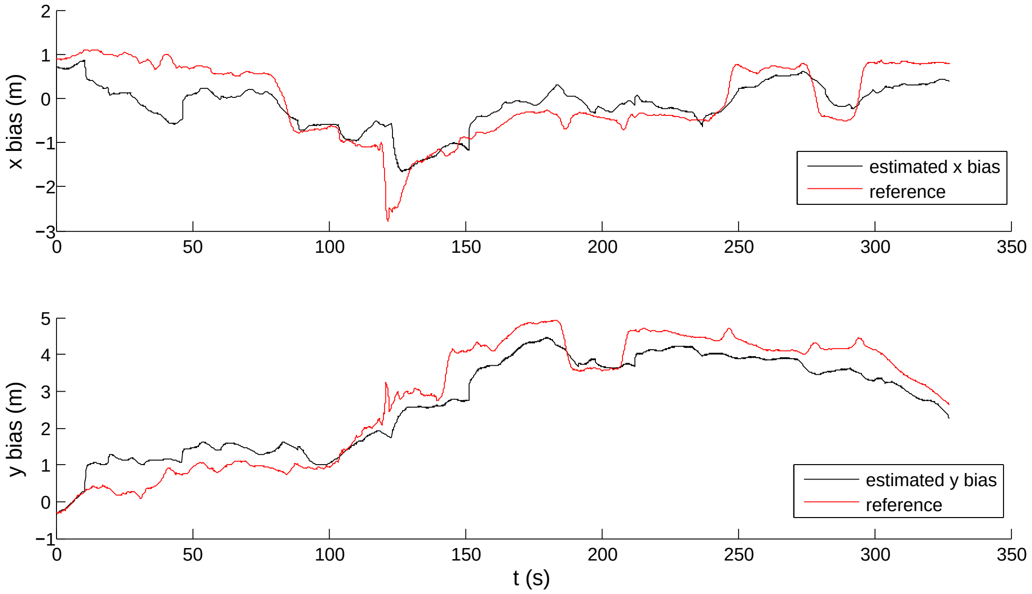


Figure 3.15: *Estimated biases compared to the reference*

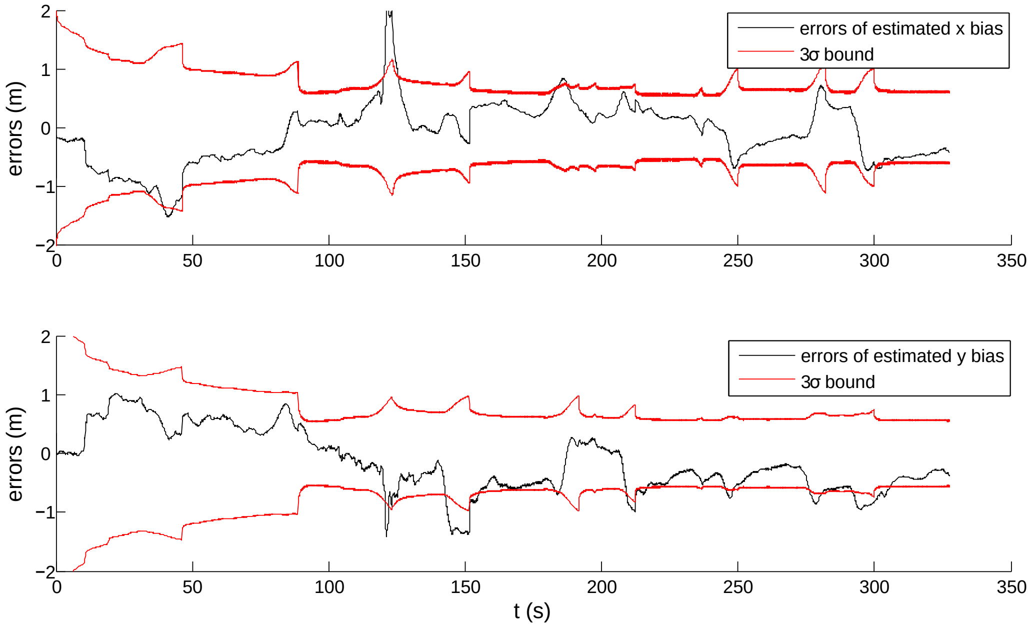


Figure 3.16: *Estimated GNSS bias errors and error bounds*

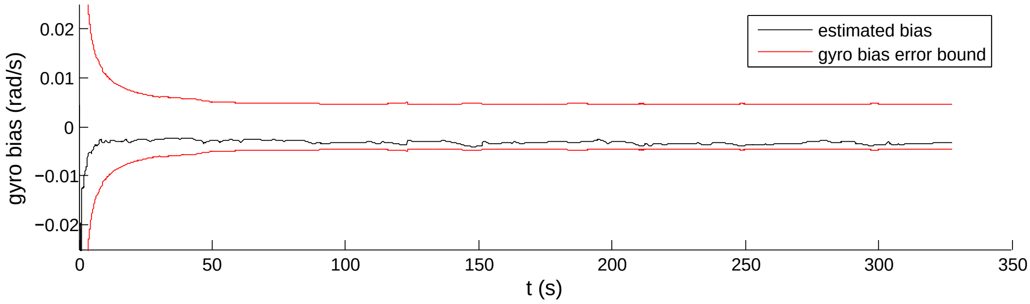


Figure 3.17: *Estimated gyro bias and gyro bias error bounds*

3.8 Conclusion

This chapter focused on loosely coupling GNSS with DR sensors, a lane detection camera and a digital map containing the lanes markings. The main efforts have been dedicated to the modeling of the non-white biases of the GNSS position fixes. A widely used autoregressive model has been adopted. Thanks to the camera measurement, the filter is able to estimate the biases using Kalman filtering with shaping filter. A contribution is the integration and the validation of the camera observation model proposed in Chapter 2.

The localization solver is done by an EKF with measured input. The goal of this system is to achieve a localization system with high availability and camera-like accuracy to make the vehicle able to navigate autonomously. The experimental results demonstrate the feasibility of fusing the proposed camera observation model and shaping filtering to achieve a position estimation with decimeter-level accuracy (median value) in both lateral and longitudinal direction. The results inspire us to perform an observability analysis and to seek to enhanced state space models.

Chapter 4

Enhanced GNSS errors modeling and design of a road-invariant filter

Contents

4.1 Introduction

Chapter 3 has demonstrated the effectiveness of the camera measurements (exploited in a Kalman filter thanks to a refined observation model that takes into account the exact position of the camera in the body frame) to increase the accuracy of the localization process. GNSS fixes errors were handled thanks to autoregressive models in both directions of the ENU frame.

Camera measurements are related to the local geometry of the road. When using a smart camera as a MobilEye, the corrections are essentially in the lateral direction. As shown in Figure 4.1, vehicles 0 and 1 have the same heading and lateral position in the lane, but different longitudinal positions. C_0 measurements are the same for both vehicles (C_0 relies only on vehicle lateral position in the lane for a given vehicle heading). The observability of the GNSS error along the road axis is therefore intuitively weak whereas it is high in the cross-track direction. The localization problem possesses some “state invariance” with respect to road or lane rotations. However, a state modeling in the ENU frame doesn’t reflect this invariance, because the x-y axes rarely correspond to lateral-longitudinal directions. Therefore, this

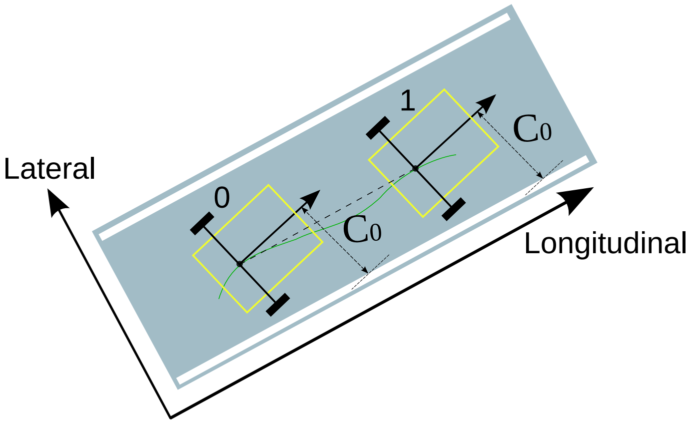


Figure 4.1: *C_0 is a lateral measurement*

chapter is dedicated to study an enhanced model of the GNSS biases for this kind of localization problem with lateral vision information.

As seen before, localization is a pose estimation problem. The estimation of an enhanced model is feasible only when the system has suitable observability characteristics [114]. The GNSS biases can be modeled by different models which are more or less complex. So, observability conditions can be used to validate the choice of a proposed enhanced model. Once the model has been chosen, the next step is to design a localization solver. In our particular case, a particular problem is to find an effective working frame which keeps the state invariance with respect to road rotations. We will see that a working frame aligned with the road provides an interesting solution.

This chapter is organized as follows. Section 4.2 focuses on observability conditions to validate a proposed enhanced modeling. Section 4.3 describes a road invariant EKF algorithm that works in a frame kept aligned with the road. Section 4.4 presents experimental results. Section 4.5 concludes this chapter.

4.2 Design of an enhanced modeling of the errors based on observability conditions

4.2.1 Proposed enhanced modeling in a road-oriented frame

The working frame in which a localization solver is implemented plays an important role in terms of error modeling and estimation performance. A local ENU frame is often used, but in order to develop an enhanced modeling of GNSS bias in the lateral direction, we propose to consider a modeling of the system in a road-oriented Cartesian frame. A road-oriented frame is defined to have the same origin as the

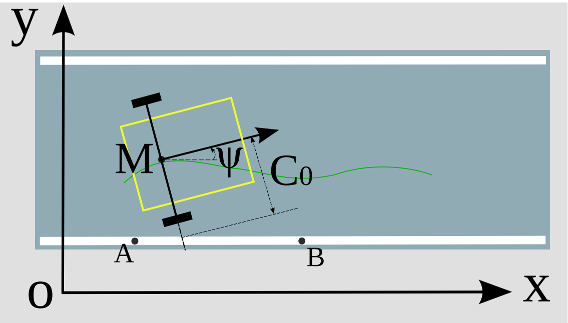


Figure 4.2: *Road-oriented frame*

local ENU frame but with its x-axis pointing to the direction of the road in which the vehicle is traveling (cf. Figure 4.2).

In Chapter 3, two kinds of modeling of GNSS bias have been mentioned: a random bias model defined by equation (3.3) and a first order autoregressive (or Markov) model defined by equation (3.4). In [13], the combination of these two models is used to model a gyro bias. Here, the same model is used to enhance the modeling of GNSS bias, especially in the lateral direction.

Working in the road frame, an enhanced model can be defined as:

$$\varepsilon_y = \varepsilon_{y1} + \varepsilon_{y2} \tag{4.1}$$

with

$$\dot{\varepsilon}_{y1} = -\varepsilon_{y1}/\tau_1$$

$$\dot{\varepsilon}_{y2} = 0$$

The physical meaning of this modeling is as follows: the autoregressive bias corresponds to the filtering effect of the GNSS receiver and the random constant corresponds to the bias caused by the slow varying error sources described in Chapter 2.

In the longitudinal direction, only autoregressive modeling is used considering its weak observability. However, the error is also decomposed into two parts:

$$\varepsilon_x = \varepsilon_{x1} + \varepsilon_{x2} \tag{4.2}$$

with

$$\dot{\varepsilon}_{x1} = -\varepsilon_{x1}/\tau_1$$

$$\dot{\varepsilon}_{x2} = -\varepsilon_{x2}/\tau_2$$

The reason for this will be explained in Section 4.3.

The state vector becomes:

$$\mathbf{x} = [x, y, \psi, \varepsilon_\omega, \varepsilon_{x1}, \varepsilon_{x2}, \varepsilon_{y1}, \varepsilon_{y2}]^T \tag{4.3}$$

where (x, y, ψ) is the 2D pose of the vehicle in the road frame; ε_ω denotes the gyro bias; $(\varepsilon_{x1}, \varepsilon_{x2}, \varepsilon_{y1}, \varepsilon_{y2})$ are GNSS errors on x and y in the road-oriented frame.

The evolution model of the state vector is different than the one of equation (3.58):

$$\begin{cases} \dot{x} = v \cdot cos\psi \\ \dot{y} = v \cdot sin\psi \\ \dot{\psi} = \omega - \varepsilon_\omega \\ \dot{\varepsilon_\omega} = 0 \\ \dot{\varepsilon}_{x1} = -\varepsilon_{x1}/\tau_1 \\ \dot{\varepsilon}_{x2} = -\varepsilon_{x2}/\tau_2 \\ \dot{\varepsilon}_{y1} = -\varepsilon_{y1}/\tau_1 \\ \dot{\varepsilon}_{y2} = 0 \end{cases} \tag{4.4}$$

In this model, first order autoregressive models with time constant τ_1 and τ_2 are used to model the non-whiteness of the GNSS errors. The error in the x-direction is split into two components (ε_{x1} and ε_{x2}) with different decorrelation time constants in order to manage the frame transformation when the road changes (detailed explanation in Section 4.3.1).

The time constant of ε_{y1} is supposed to be the same as the one of ε_{x1} in order to simplify the tuning with only one parameter.

v is the linear velocity measured by the wheel speed sensors and ω is the angular velocity measured by the yaw rate gyro.

ε_{x2} is modeled as an autoregressive process to guarantee the observability of the filter, but it has a larger time constant since it is transformed from ε_{y2}, when a road frame change is performed (and vice versa). This modeling is similar to the modeling of accelerometers biases as done in [13] because accelerometers usually have two kinds of biases: a first order Markov process with large time constant and a faster changing Markov process [105].

The last equation of the model associated with ε_{y2} plays an important role in our localizer. Indeed, it is a random constant model as used in [75] which is well adapted to estimate quickly the lateral bias of the GNSS fix in the road frame [13].

The exteroceptive sensors that are considered are a GNSS receiver providing position fixes and a front-looking camera that detects lane markings. In order to study the structural properties of the modeling with equations easy to handle, let us suppose that the camera and the GNSS antenna coincide with point M, the origin of the body frame (cf. Figure 4.2).

Moreover, we consider at this stage that there is only one lane marking, locally represented by a line $[AB]$. The observation model in this case is given by:

$$C_0 = (y - y_A) / \cos\psi \tag{4.5}$$

C_0 is the lateral distance measured by the camera in the body frame [119]. y_A is the ordinate of points A and B in the road-oriented frame.

The GNSS fixes with their shaping errors are linked to the state by the following model:

$$\begin{cases} x_{GNSS} = x + \varepsilon_{x1} + \varepsilon_{x2} \\ y_{GNSS} = y + \varepsilon_{y1} + \varepsilon_{y2} \end{cases} \tag{4.6}$$

4.2.2 Observability concepts

In control theory, observability is a condition checking of how internal states of a system can be inferred by knowledge of its external outputs. The observability was first introduced by Rudolf E. Kalman for linear dynamic systems. Observability is a necessary condition for any filtering algorithm to converge to an unbiased state estimate. The localization system in this thesis is a nonlinear system.

Considering a nonlinear system:

$$\Sigma : \begin{array}{l} \dot{\mathbf{x}} = f(\mathbf{x}, \mathbf{u}), \mathbf{x} \in M \subseteq \mathbb{R}^n \\ \mathbf{y} = g(\mathbf{x}), \mathbf{y} \in \mathbb{R}^m \end{array}$$

with $\mathbf{x}$ the state vector, $\mathbf{u}$ the input vector, and $\mathbf{y}$ the measurement vector.

In order to describe its observability, the following definitions express first the concepts propose by Hermann and Krener in [57], which is a geometrical approach to observability analysis.

Definition 1. Indistinguishable states

Two states are said to be indistinguishable, if for every admissible input $\mathbf{u}(t)$, $t_0 \leq t \leq t_{end}$, identical outputs result:

$$\mathbf{y}(t; \mathbf{x}_0) \equiv \mathbf{y}(t; \mathbf{x}_1) \text{ for } t_0 \leq t \leq t_{end}$$

The set of all points that are indistinguishable from $\mathbf{x}_0$ is denoted $I(\mathbf{x}_0)$.

For example, in Figure 4.1, if one uses only the camera, the two initial states fulfill $C_0(t;\mathbf{x}_0) \equiv C_0(t;\mathbf{x}_1)$ for $t_0 \leq t \leq t_{end}$, thus they are indistinguishable with respect to C_0.

Definition 2. Observability

$\mathbf{x}_0$ of the system Σ is observable, if $I(\mathbf{x}_0) = \mathbf{x}_0$. Σ is observable, if $I(\mathbf{x}) = \mathbf{x}$ for all $\mathbf{x} \in M$.

Observability is sometimes called global observability. Reconstruction of $\mathbf{x}$ for measurement data may be possible for certain inputs $\mathbf{u}(t)$, $t_0 \leq t \leq t_{end}$ only.

Definition 3. Local observability

System Σ is locally observable, if $I_N(\mathbf{x}) = \mathbf{x}$ for every open neighborhood N of $\mathbf{x} \in M$.

Local observability is a stronger property than global observability.

Definition 4. Local weak observability

The state $\mathbf{x}_0$ of the system Σ is locally weakly observable, if there is some neighborhood ν of $\mathbf{x}_0$, where $I_N(\mathbf{x}_0) \cap \nu = \mathbf{x}_0$, for all solutions $\mathbf{x}(t)$ completely in any neighborhood N of $\mathbf{x}_0$. System Σ is locally weakly observable, if this property holds for all $\mathbf{x} \in M$.

The most classical method of studying the local weak observability relies on the study of a rank condition after linearization and Lie derivatives computation [57].

There is another approach that relies on the differential algebraic concept which is largely the work of Ritt [107]. The differential algebra has been introduced in control theory through the works of Fliess, Glad and Ljung [41][49][40][84]. The algebraic observability can be expressed as follows (see [113]):

Definition 5. Algebraic observability

The state of a system with known internal dynamics is said to be observable if, and only if, there is an algebraic equation linking the state vector to the measured output $\mathbf{y}$ and input $\mathbf{u}$ and a finite number of their time derivatives.

The following is an example.

Example. The two-dimensional single input, single output system

$$\begin{cases} \dot{x}_1 = 0 \\ \dot{x}_2 = u + x_2 \\ y = x_1 + x_2 \end{cases} \tag{4.7}$$

is observable since $x_2 = \dot{y} - u$ and $x_1 = y - \dot{y} + u$.

Algebraic observability is therefore a different way to study observability. It has also the advantage to provide a closed form to build a state observer as soon as the derivatives of the inputs and outputs can be estimated with a good quality. It is equivalent to local generic observability [37].

In the following sections, we study the observability of the different components of the state in the algebraic framework considering it provides local observability.

4.2.3 Observability analysis

The state space given by equation (3.50) is the basic modeling on which a state observer can be built to estimate the pose of the vehicle and to merge the information from different sensors. The last five components of the state vector with their respective modeling act as shaping filters in the observer. The question that needs to be answered is: are all the components of the state observable when using the exteroceptive measurements $\mathbf{y} = [x_{GNSS}, y_{GNSS}, C_0]$ and the proprioceptive ones $\mathbf{u} = [v, w]$? If a state component $\mathbf{x}_1$ is observable and another state component $\mathbf{x}_2$ can be expressed by a algebraic function of $\mathbf{x}_1$, $\mathbf{y}$, $\mathbf{u}$ and their derivatives, $\mathbf{x}_2$ is also observable. Observability of the vehicle heading and gyro bias are first studied by defining $\psi \in (-0.3, 0.3)$ rad in a road oriented frame (according to the range of C_1 provided by MobilEye), then vehicle position and GNSS biases by the following.

Equations (4.4) and (4.5) are analytic but not algebraic. A solution is to define an equivalent algebraic system (with algebraic functions) by introducing auxiliary state variables [27].

Let us define two auxiliary state variables:

$$\begin{cases} x_1 = \cos\psi \\ x_2 = \sin\psi \end{cases} \tag{4.8}$$

Since

$$\dot{x}_1 = -\dot{\psi} \cdot \sin\psi = -\left(\omega - \varepsilon_\omega\right) x_2$$

and

$$\dot{x}_2 = \dot{\psi} \cdot \cos\psi = \left(\omega - \varepsilon_\omega\right) x_1$$

equations (4.4) and (4.5) are complemented with two additional equations and $\dot{\psi}$ is removed:

$$\begin{cases} \dot{x} = v \cdot x_1 \\ \dot{y} = v \cdot x_2 \\ \dot{x}_1 = -x_2 \left(\omega - \varepsilon_\omega\right) \\ \dot{x}_2 = x_1 \left(\omega - \varepsilon_\omega\right) \\ \dot{\varepsilon_\omega} = 0 \\ \dot{\varepsilon}_{x1} = -\varepsilon_{x1}/\tau_1 \\ \dot{\varepsilon}_{x2} = -\varepsilon_{x2}/\tau_2 \\ \dot{\varepsilon}_{y1} = -\varepsilon_{y1}/\tau_1 \\ \dot{\varepsilon}_{y2} = 0 \end{cases} \tag{4.9}$$

and

$$C_0 = \left(y - y_A\right)/x_1 \tag{4.10}$$

All the functions are now rational.

4.2.3.1 Observability of the vehicle heading ψ

By taking the derivative of equation (4.10), we have:

$$\dot{y} = \dot{C}_0 \cdot x_1 - \dot{\psi} \cdot C_0 \cdot x_2 \tag{4.11}$$

By plugging $\dot{y} = v \cdot x_2$ and $\dot{\psi} = \omega - \varepsilon_\omega$:

$$v \cdot x_2 = \dot{C}_0 \cdot x_1 - \left(\omega - \varepsilon_\omega\right) \cdot C_0 \cdot x_2 \tag{4.12}$$

By taking now the derivative of equation (4.12), we have:

$$\left[\dot{v} + 2\dot{C}_0 \left(\omega - \varepsilon_\omega\right) + \dot{\omega} \cdot C_0\right] x_2 = \left[\ddot{C}_0 - C_0 \left(\omega - \varepsilon_\omega\right)^2 - v \left(\omega - \varepsilon_\omega\right)\right] x_1 \tag{4.13}$$

If ψ is identically null (the vehicle is traveling parallel to the lane marking) then $\dot{\psi} = 0$ and we have $\varepsilon_\omega = \omega$ (the gyro bias is then observable). Moreover, in this case, we have $\dot{C}_0$ that is identically null and so one can observe that ψ is identically null.

Now suppose that ψ is not null. So, $x_2 \neq 0$. From equation (4.12), we can work out:

$$\varepsilon_\omega = \frac{v \cdot x_2 - \dot{C}_0 \cdot x_1}{C_0 \cdot x_2} + \omega \tag{4.14}$$

C_0 is physically non null because the lane markings are on the sides of the lane.

By plugging equation (4.14) into equation (4.13) and by noting that $x_1 = (1 - x_2^2)^{1/2}$, one gets an implicit algebraic function :

$$\left(\dot{v} \cdot C_0 - v \cdot \dot{C_0} + \dot{\omega} \cdot C_0^2\right) x_2^3 + \left(\ddot{C_0} \cdot C_0 - \dot{C_0}^2\right) \left(1 - x_2^2\right)^{3/2} + \left(2\dot{C_0}^2 - \ddot{C_0} \cdot C_0\right) \left(1 - x_2^2\right)^{1/2} - v \cdot \dot{C_0} \cdot x_2 = 0 \quad (4.15)$$

This expression is quite complicated. We have solved it using the symbolic solver of Matlab

Let us denote:

$$f = p_1 x_2^3 + p_2 \left(1 - x_2^2\right)^{3/2} + p_3 \left(1 - x_2^2\right)^{1/2} + p_4 \cdot x_2 \qquad (4.16)$$

with

$$\begin{cases} p_1 = \dot{v} \cdot C_0 - v \cdot \dot{C_0} + \dot{\omega} \cdot C_0^2 \\ p_2 = \ddot{C_0} \cdot C_0 - \dot{C_0}^2 \\ p_3 = 2\dot{C_0}^2 - \ddot{C_0} \cdot C_0 \\ p_4 = -v \cdot \dot{C_0} \end{cases}$$

One can get an explicit algebraic function which express x_2 with p_1, p_2, p_3 and p_4 by using the following Matlab code:

```
syms x2;
syms p1 p2 p3 p4;
f = p1*x2^3+p2*(sqrt(1-x2^2))^3+p3*sqrt(1-x2^2)+p4*x2;
explicit_function=solve(f,'x2');
```

The result "*explicit_function*" is the explicit function which expresses x_2 as a function of p_1, p_2, p_3 and p_4. Thus, we are able to find a (complicated) algebraic equation that is not reported in this document.

Another solution is to use the implicit function theorem on simulated typical scenario. This is what is done in the following section.

Theorem 1. *Implicit function theorem*

Let $\mathbf{f}\colon \mathbb{R}^{n+m} \to \mathbb{R}^m$ *be a continuously differentiable function, and let* $\mathbb{R}^{n+m}$ *have coordinates* $(\mathbf{x}, \mathbf{y})$. *Fix a point* $(\mathbf{a}, \mathbf{b}) = (a_1, \ldots, a_n, b_1, \ldots, b_m)$ *with* $\mathbf{f}(\mathbf{a}, \mathbf{b}) = \mathbf{c}$, *where* $\mathbf{c} \in \mathbb{R}^m$. *If the matrix* $[(\partial \mathbf{f}_i / \partial \mathbf{y}_i)(\mathbf{a}, \mathbf{b})]$ *is invertible, then there exists an open set* U *containing* $\mathbf{a}$, *an open set* V *containing* $\mathbf{b}$, *and a unique continuously differentiable function* $\mathbf{g}\colon U \to V$ *such that*

$\{(\mathbf{x}, g(\mathbf{x})) | \mathbf{x} \in U\} = \{(\mathbf{x}, \mathbf{y}) \in U \times V | f(\mathbf{x}, \mathbf{y}) = \mathbf{c}\}$.

Proposition. *As equation (4.15) is non-singular when the vehicle linear or rotational speeds are not null or when the vehicle accelerates or decelerates, it is possible to get, around a working point, a function Φ_{x_2} of C_0, $\dot{C}_0$, $\ddot{C}_0$, v, $\dot{v}$, ω and $\dot{\omega}$ such as:*

$$x_2 = \Phi_{x_2}\left(C_0, \dot{C}_0, \ddot{C}_0, v, \dot{v}, \omega, \dot{\omega}\right) \tag{4.17}$$

Proof. Here, we give some demonstration elements using the implicit function theorem which gives a sufficient condition to ensure that there is such an algebraic function.

Let be ϕ a real-valued function defined on a domain D such that

$$\begin{aligned}\phi\left(C_0, \dot{C}_0, \ddot{C}_0, v, \dot{v}, \omega, \dot{\omega}, x_2\right) = \;& \left(\dot{v}\cdot C_0 - v\cdot\dot{C}_0 + \dot{\omega}\cdot C_0^2\right)x_2^3 + \left(\ddot{C}_0\cdot C_0 - \dot{C}_0{}^2\right)\left(1-x_2^2\right)^{3/2} \\ & + \left(2\dot{C}_0{}^2 - \ddot{C}_0\cdot C_0\right)\left(1-x_2^2\right)^{1/2} - v\cdot\dot{C}_0\cdot x_2\end{aligned} \tag{4.18}$$

Let take a point $\left(C_0^O, \dot{C}_0{}^O, \ddot{C}_0{}^O, v^O, \dot{v}^O, \omega^O, \dot{\omega}^O, x_2^O\right) \in D$, which satisfies equation (4.15)

$$\phi\left(C_0^O, \dot{C}_0{}^O, \ddot{C}_0{}^O, v^O, \dot{v}^O, \omega^O, \dot{\omega}^O, x_2^O\right) = 0 \tag{4.19}$$

According to the implicit function theorem, if

$$\frac{\partial\phi\left(C_0^O, \dot{C}_0{}^O, \ddot{C}_0{}^O, v^O, \dot{v}^O, \omega^O, \dot{\omega}^O, x_2^O\right)}{\partial x_2} \neq 0 \tag{4.20}$$

then there exists a neighborhood $V_0\left(C_0^O, \dot{C}_0{}^O, \ddot{C}_0{}^O, v^O, \dot{v}^O, \omega^O, \dot{\omega}^O\right) \subset D$, an open set W_0 containing x_2^O and a real valued function $\Phi_{x_2} : V_0 \to W_0$ such that for any $\left(\left(C_0, \dot{C}_0, \ddot{C}_0, v, \dot{v}, \omega, \dot{\omega}\right), x_2\right) \in V_0 \times W_0$, if $\phi\left(C_0, \dot{C}_0, \ddot{C}_0, v, \dot{v}, \omega, \dot{\omega}, x_2\right) = 0$, then $x_2 = \Phi_{x_2}\left(C_0, \dot{C}_0, \ddot{C}_0, v, \dot{v}, \omega, \dot{\omega}\right)$. If this function exists for some neighborhoods of x_2, the observability is local and weak. Otherwise if equation (4.20) holds for all the open sets containing x_2, the observability is local. □

The derivative of ϕ with respect to x_2 is another function of x_2 (which is easy to calculate here). In order to do the observability analysis, we propose to do a realistic simulation.

Let define a vehicle trajectory as an explicit function with time t (a sine function

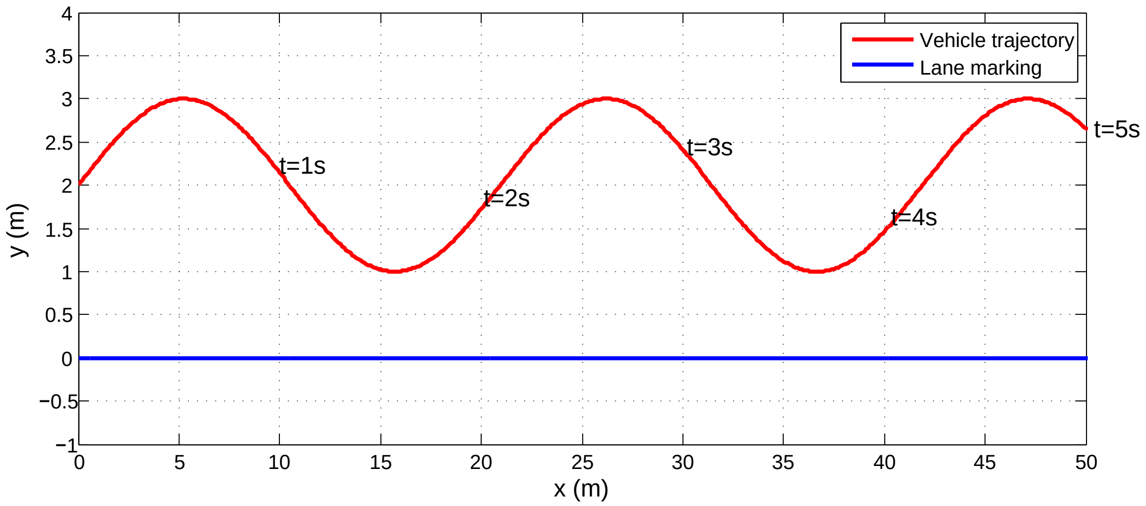


Figure 4.3: *Simulated vehicle trajectory and lane marking*

with a variable speed here):

$$\begin{cases} x = 10t \\ y = \sin 3t + 2 \\ \psi = \arctan\left(0.3\cos 3t\right) \\ v = 10\sqrt{1 + 0.09\cos^2 3t} \\ \omega = -0.9\sin 3t / \left(1 + 0.09\cos^2 3t\right) \\ C_0 = -\left(\sin 3t + 2\right) / \cos\left(\arctan\left(0.3\cos 3t\right)\right) \end{cases} \tag{4.21}$$

With the lane marking equation as $y = 0$.

Figure 4.3 shows the simulated vehicle trajectory and the lane marking. The trajectory is designed according to the capability of the camera to detect the lane marking in its field of view. Along the trajectory, every parameter of equation 4.20 is known. Every sample of the trajectory is the point around which the analysis is conducted.

Figure 4.4 shows the values of the derivative of equation (4.19) with respect to x_2. One can remark that, only when $x_2 = 0$, this derivative equals 0. Otherwise, the implicit theorem condition is verified since the derivative is different from zero.

In other words, the system observability is globally verified for this trajectory (since there exists an algebraic function to express x_2) except on a finite number of points where $x_2 = 0$. As shown in Figure 4.4, equation (4.20) holds for all the open sets of $[-\sin(0.3), 0[\cup]0, \sin(0.3)]$. Therefore, we can say that

$$x_2 = \Phi_{x_2}\left(C_0, \dot{C}_0, \ddot{C}_0, v, \dot{v}, \omega, \dot{\omega}\right) \tag{4.22}$$

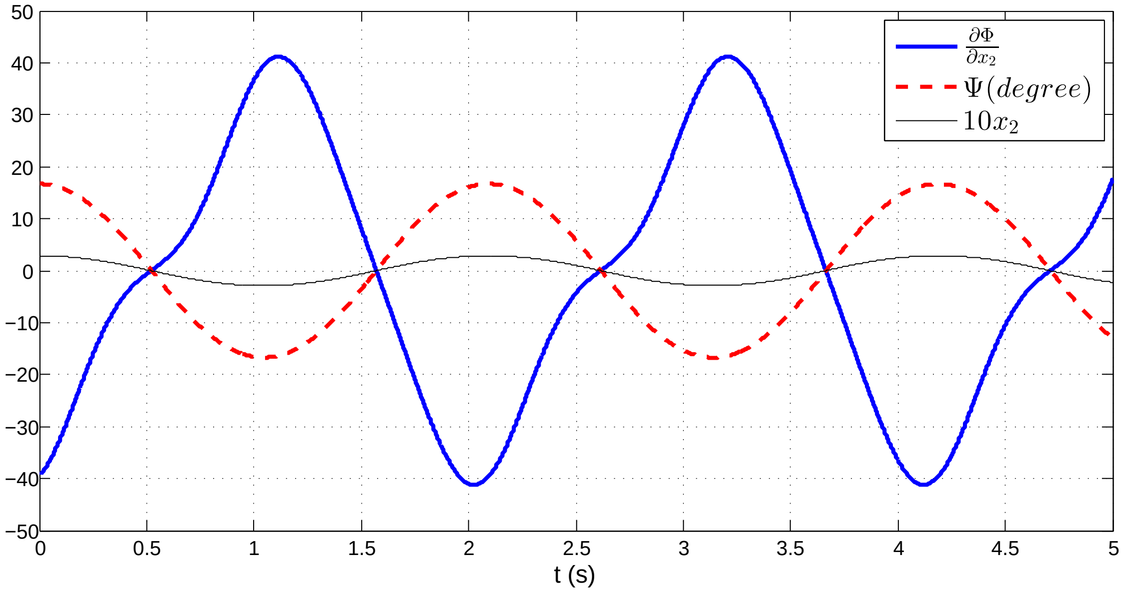


Figure 4.4: *Derivative of simulated data*

is generically observable with $x_2 \in [-\sin(0.3), 0[\cup]0, \sin(0.3)]$.

$$x_1 = \sqrt{1 - x_2^2} = \Phi_{x_1}\left(C_0, \dot{C}_0, \ddot{C}_0, v, \dot{v}, \omega, \dot{\omega}\right) \tag{4.23}$$

is observable.

We could repeat this procedure for many different simulated trajectories but we think that the reported trajectory is relevant for real vehicles driving on linear roads.

ψ is determined by $\psi = \arcsin(x_2)$. Thus ψ is observable in $[-0.3, 0[\cup]0, 0.3]$.

There is a particular situation when the trajectory is strictly parallel to the road. In this case, ψ is identically null and then $\dot{C}_0$ is identically null. One can therefore observe that ψ is identically null and so ψ is observable.

Note: If C_1 would be accurate enough and used to update the state vector, the observability analysis of ψ would be simplified.

4.2.3.2 Observability of the gyro bias ε_ω

By using equation (4.23) and (4.22) in equation (4.14), a function $\Phi_{\varepsilon_\omega}$ which gives ε_ω is obtained:

$$\varepsilon_\omega = \Phi_{\varepsilon_\omega}\left(C_0, \dot{C}_0, \ddot{C}_0, v, \dot{v}, \omega, \dot{\omega}\right) \tag{4.24}$$

ε_ω is therefore observable.

4.2.3.3 Observability of the vehicle position (x, y) and of GNSS biases

By taking the derivative of equation (4.6), we have:

$$\dot{x}_{GNSS} = v \cdot x_1 - \varepsilon_{x1}/\tau_1 - \varepsilon_{x2}/\tau_2 \tag{4.25}$$

By doing the same with equation (4.25):

$$\ddot{x}_{GNSS} = \dot{v} \cdot x_1 - v \cdot x_2 \cdot (\omega - \varepsilon_\omega) + \varepsilon_{x1}/\tau_1^2 + \varepsilon_{x2}/\tau_2^2 \tag{4.26}$$

Since ψ is observable and as we have a linear system with two unknowns and two equations, we get:

$$\varepsilon_{xi} = \Phi_{\varepsilon_{xi}} \left(C_0, \dot{C}_0, \ddot{C}_0, \dot{x}_{GNSS}, \ddot{x}_{GNSS}, v, \dot{v}, \omega, \dot{\omega} \right) \; i = 1, 2 \tag{4.27}$$

So ε_{x1} and ε_{x2} are observable. If $\tau_1 = \tau_2$, ε_{x1} and ε_{x2} are not solvable by equations (4.6) and (4.25). However, $(\varepsilon_{x1} + \varepsilon_{x2})$ is observable.

With $x = x_{GNSS} - (\varepsilon_{x1} + \varepsilon_{x2})$ and equation (4.27), x is observable.

With $y = y_A + C_0 \cdot x_1$ and equation (4.23), y is observable.

The expression of ε_{y1} is given as follows:

$$\dot{y}_{GNSS} = v \cdot x_2 - \varepsilon_{y1}/\tau_1 \tag{4.28}$$

$$\varepsilon_{y1} = \left(v \cdot x_2 - \dot{y}_{GNSS} \right) \tau_1 \tag{4.29}$$

So ε_{y1} is observable.

With $\varepsilon_{y2} = y_{GNSS} - y - \varepsilon_{y1}$, we can derive that ε_{y2} is observable.

4.2.3.4 Observability conclusion

So far, we have proved that every element in the state vector X can be expressed by an algebraic function of components of Y and U and a finite number of their derivatives. We can consequently conclude that the state vector with its associated state space is observable as long as the vehicle moves or accelerates.

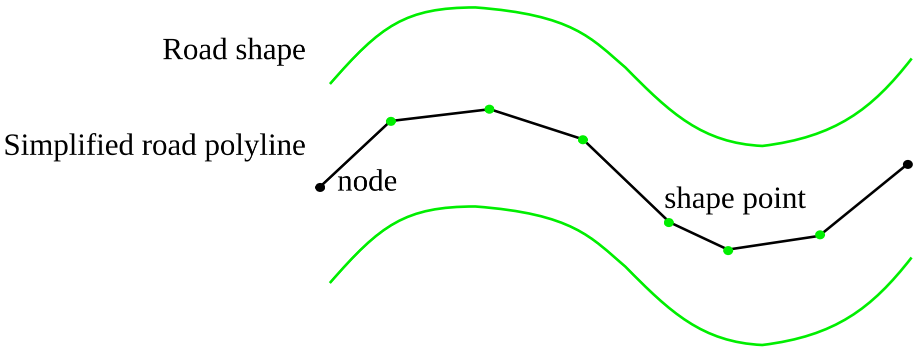


Figure 4.5: *The geometrical transformation can be performed along segments of the simplified road polyline (in black). Nodes (black points) are the start and end points of the polyline. Shape points (green points) define geometry between two nodes.*

4.3 Road-invariant Extended Kalman filter

The observability of the state has been demonstrated in a road-oriented frame. However, in reality, the orientation of the road changes as the vehicle moves from one road to another or when the road orientation changes. In this thesis, orientations are predefined for each road and the geometrical transformations are performed from road to road considering the roads are quite straight in the experiments. As indicated in Figure 4.5, the road can be curved. In this case, one can perform the geometrical transformation along each segment of the simplified road polyline very frequently. In this case, the road frame follows the orientation of the segment between two adjacent nodes or shape points.

We are now building an EKF that estimates the pose vector of the vehicle from one road to another, in a sequential way. This idea is inspired form the Invariant EKF proposed by Bonnabel et al. [21].

4.3.1 Geometrical transformation from one road to another

R_O denotes the local ENU frame and R_i is the working road-oriented frame, with its x-axis pointing to the direction of road i. When the vehicle passes from road i to road j, the working frame changes from R_i to R_j.

Let ${}^{j}\mathbf{x}$ denote the state vector in frame R_j:

$$ {}^{j}\mathbf{x} = \left[{}^{j}x, {}^{j}y, {}^{j}\psi, {}^{j}\varepsilon_{\omega}, {}^{j}\varepsilon_{x1}, {}^{j}\varepsilon_{x2}, {}^{j}\varepsilon_{y1}, {}^{j}\varepsilon_{y2} \right]^{T} \tag{4.30} $$

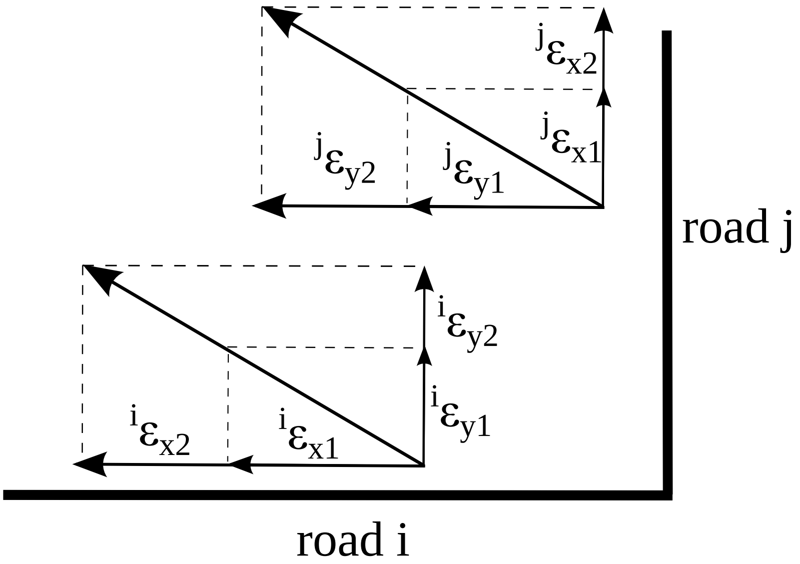


Figure 4.6: *Geometrical transformation of the GNSS biases.*

The transformation from ${}^i\mathbf{x}$ to ${}^j\mathbf{x}$ involves rotations and is given by equation (4.31):

$$\begin{cases} {}^jx = {}^ix \cdot \cos\alpha + {}^iy \cdot \sin\alpha \\ {}^jy = -{}^ix \cdot \sin\alpha + {}^iy \cdot \cos\alpha \\ {}^j\psi = {}^i\psi - \alpha \\ {}^j\varepsilon_\omega = {}^i\varepsilon_\omega \\ {}^j\varepsilon_{x1} = {}^i\varepsilon_{x1} \cdot \cos\alpha + {}^i\varepsilon_{y1} \cdot \sin\alpha \\ {}^j\varepsilon_{x2} = {}^i\varepsilon_{x2} \cdot \cos\alpha + {}^i\varepsilon_{y2} \cdot \sin\alpha \\ {}^j\varepsilon_{y1} = -{}^i\varepsilon_{x1} \cdot \sin\alpha + {}^i\varepsilon_{y1} \cdot \cos\alpha \\ {}^j\varepsilon_{y2} = -{}^i\varepsilon_{x2} \cdot \sin\alpha + {}^i\varepsilon_{y2} \cdot \cos\alpha \end{cases} \tag{4.31}$$

where $\alpha = \theta_j - \theta_i$, θ_i and θ_j are respectively the orientation of road i and road j in R_O (cf. Figure 4.7). For an efficient implementation, the orientation of each road can be stored in the map.

Figure 4.6 illustrates how the estimated biases change when the two successive roads are orthogonal.

Let ${}^i\mathbf{P}$ denote the covariance matrix estimated by the EKF in the working frame R_i. The transformation from $({}^i\mathbf{x},{}^i\mathbf{P})$ to $({}^j\mathbf{x},{}^j\mathbf{P})$ is described by the function given in Algorithm 4.1, where sα and cα denote sinα and cosα respectively. The road directions being deterministic, the new covariance matrix is simply the result of a linear transformation of a random vector.

Now, the reason why the bias on ${}^ix$ has been modeled by two components becomes clear. We aim at modeling the lateral bias on ${}^iy$ by an autoregressive process plus

Algorithm 4.1 Function $R_i2R_j\_State$

```
Input: ${}^i\mathbf{x}, {}^i\mathbf{P}, \theta_i, \theta_j$
    $\alpha = \theta_j - \theta_i$
    ${}^j\mathbf{H}_i = \begin{bmatrix} c\alpha & s\alpha & 0 & 0 & 0 & 0 & 0 & 0 \\ -s\alpha & c\alpha & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & c\alpha & 0 & s\alpha & 0 \\ 0 & 0 & 0 & 0 & 0 & c\alpha & 0 & s\alpha \\ 0 & 0 & 0 & 0 & -s\alpha & 0 & c\alpha & 0 \\ 0 & 0 & 0 & 0 & 0 & -s\alpha & 0 & c\alpha \end{bmatrix}$
    ${}^jb_i = [0, 0, -\alpha, 0, 0, 0, 0, 0]^T$
    ${}^j\mathbf{x} = {}^j\mathbf{H}_i {}^i\mathbf{x} + {}^jb_i$
    ${}^j\mathbf{P} = {}^j\mathbf{H}_i {}^i\mathbf{P} \left({}^j\mathbf{H}_i\right)^T$
Output: ${}^j\mathbf{x}, {}^j\mathbf{P}$
```

a random constant to get a better estimation process (we have seen that these two components are observable thanks to the camera measurements, when working in the road-oriented frame). When the vehicle pose is converted from one road frame to another, doing the inverse transformation has to give the same estimate. Mathematically, it means that matrix ${}^j\mathbf{H}_i$ has to be squared such that ${}^j\mathbf{H}_i \cdot {}^i\mathbf{H}_j = \mathbf{I}$ (Identity matrix).

As we have

$$
{}^i\mathbf{H}_j = \begin{bmatrix} c\alpha & -s\alpha & 0 & 0 & 0 & 0 & 0 & 0 \\ s\alpha & c\alpha & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & c\alpha & 0 & -s\alpha & 0 \\ 0 & 0 & 0 & 0 & 0 & c\alpha & 0 & -s\alpha \\ 0 & 0 & 0 & 0 & s\alpha & 0 & c\alpha & 0 \\ 0 & 0 & 0 & 0 & 0 & s\alpha & 0 & c\alpha \end{bmatrix}
$$

It is straightforward to check that our proposal verifies ${}^j\mathbf{H}_i \cdot {}^i\mathbf{H}_j = \mathbf{I}$ since the transformation matrix involves rotations (when multiplying a rotation matrix by its inverse, one gets the identity).

Now, let us proof that if one models the bias on ${}^ix$ by only one component, when the frame changes from R_i to R_j, there is no way to find a bijective transformation. The question can be expressed as follows.

Let define

$$^{j}\varepsilon_{y} = {}^{j}\varepsilon_{y1} + {}^{j}\varepsilon_{y2}$$

When the road frame changes from road i to road j, we have:

$$\begin{cases} ^{j}\varepsilon_{x} = {}^{i}\varepsilon_{x}\text{cos}\alpha - {}^{i}\varepsilon_{y}\text{sin}\alpha \\ ^{j}\varepsilon_{y} = {}^{i}\varepsilon_{x}\text{sin}\alpha + {}^{i}\varepsilon_{y}\text{cos}\alpha \end{cases} \tag{4.32}$$

In order to find the transformation matrix, we need now to split $^{j}\varepsilon_{y}$ into two components such that:

$$\begin{aligned} &^{j}\varepsilon_{y1} = h\left(\alpha\right) {}^{j}\varepsilon_{y} \\ &^{j}\varepsilon_{y2} = \left(1 - h\left(\alpha\right)\right) {}^{j}\varepsilon_{y} \end{aligned}$$

where $h\left(\cdot\right)$ is an unknown function of α.

The transformation matrix would be:

$$^{j}\mathbf{H}_{i}^{*} = \begin{bmatrix} \text{cos}\alpha & -\text{sin}\alpha & \text{-sin}\alpha \\ h\left(\alpha\right)\text{sin}\alpha & h\left(\alpha\right)\text{cos}\alpha & h\left(\alpha\right)\text{cos}\alpha \\ \left(1 - h\left(\alpha\right)\right)\text{sin}\alpha & \left(1 - h\left(\alpha\right)\right)\text{cos}\alpha & \left(1 - h\left(\alpha\right)\right)\text{cos}\alpha \end{bmatrix}$$

Thus, the inverse is obtained by setting $\alpha \to -\alpha$ which gives

$$^{i}\mathbf{H}_{j}^{*} = \begin{bmatrix} \text{cos}\alpha & \text{sin}\alpha & \text{sin}\alpha \\ -h\left(-\alpha\right)\text{sin}\alpha & h\left(-\alpha\right)\text{cos}\alpha & h\left(-\alpha\right)\text{cos}\alpha \\ \left(h\left(-\alpha\right) - 1\right)\text{sin}\alpha & \left(1 - h\left(-\alpha\right)\right)\text{cos}\alpha & \left(1 - h\left(-\alpha\right)\right)\text{cos}\alpha \end{bmatrix}$$

The product is

$$^{j}\mathbf{H}_{i}^{*} \cdot {}^{i}\mathbf{H}_{j}^{*} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & h\left(\alpha\right) & h\left(\alpha\right) \\ 0 & 1 - h\left(-\alpha\right) & 1 - h\left(-\alpha\right) \end{bmatrix} \neq I \tag{4.33}$$

Equation (4.33) proves that it is not possible to find a reversible transformation between two frames by modeling the bias component as $[\varepsilon_{x}, \varepsilon_{y1}, \varepsilon_{y2}]$ in a Cartesian coordinate system.

The reversible transformation guarantees that the estimated GNSS bias properties in the ENU frame will not change along with the transformation.

4.3.2 ENU Transformation

Since the output of the filter has to be given in the ENU frame R_O, algorithm 4.2 describes the state transformation, where $\left({}^{O}\mathbf{x}, {}^{O}\mathbf{P}\right)$ denotes the Kalman filter estimates expressed in the ENU frame.

```
Algorithm 4.2 Function $R_i 2 R_O\_State$
Input: ${}^{i}\mathbf{x}, {}^{i}\mathbf{P}, \theta_i$
    ${}^{O}\mathbf{H}_i = \begin{bmatrix} c\theta_i & -s\theta_i & 0 & 0 & 0 & 0 & 0 & 0 \\ s\theta_i & c\theta_i & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & c\theta_i & 0 & -s\theta_i & 0 \\ 0 & 0 & 0 & 0 & 0 & c\theta_i & 0 & -s\theta_i \\ 0 & 0 & 0 & 0 & s\theta_i & 0 & c\theta_i & 0 \\ 0 & 0 & 0 & 0 & 0 & s\theta_i & 0 & c\theta_i \end{bmatrix}$
    ${}^{O}b_i = [0, 0, \theta_i, 0, 0, 0, 0, 0]^T$
    ${}^{O}\mathbf{x} = {}^{O}\mathbf{H}_i {}^{i}\mathbf{x} + {}^{O}b_i$
    ${}^{O}\mathbf{P} = {}^{O}\mathbf{H}_i {}^{i}\mathbf{P} \left({}^{O}\mathbf{H}_i\right)^T$
Output: ${}^{O}\mathbf{x}, {}^{O}\mathbf{P}$
```

4.3.3 Road-invariant EKF implementation

The filter is described in algorithm 4.3. ${}^{i}A$ and ${}^{i}B$ denote the coordinates of the detected lane marking $[AB]$ in R_i. In this algorithm, just one detected lane is considered but it can be easily extended to multi-lane detections.

The filter is triggered by DR sensor measurements. GNSS and camera measurements are transformed into the road frame to perform the update by functions $R_O 2 R_i$ and $R_O 2 R_i\_Point$. Line 12 to 15 of algorithm 4.3 monitors the change of the road orientation. If it changes, the vehicle state is transformed in the new road frame by function $R_i 2 R_j\_State$. Finally, the Kalman estimates are transformed back to ENU frame by the function $R_i 2 R_O\_State$, *to provide the system output.*

Hereafter, the main stages and functions used in the road-invariant EKF are described.

Algorithm 4.3 One loop of the Road-invariant Extended Kalman filter

```
$\mathbf{u}$ =Get(DR sensors measurements)
$({}^i\mathbf{x}, {}^i\mathbf{P})$=Predict$({}^i\mathbf{x}, {}^i\mathbf{P}, \mathbf{u})$
if GNSS data is available then
    $\left({}^O\mathbf{y}_{GNSS}, {}^O\mathbf{R}_{GNSS}\right)$ = Get(GNSS fix)
    $({}^i\mathbf{y}_{GNSS}, {}^i\mathbf{R}_{GNSS})$ $=R_O 2R_i \left({}^O\mathbf{y}_{GNSS}, {}^O\mathbf{R}_{GNSS}, \theta_i\right)$
    $({}^i\mathbf{x}, {}^i\mathbf{P})$ =Update_GNSS$({}^i\mathbf{x}, {}^i\mathbf{P}, {}^i\mathbf{y}_{GNSS}, {}^i\mathbf{R}_{GNSS})$
end if
if Camera data is available then
    $C_0$ =Get(camera measurement)
    $\left({}^O\mathbf{x}, {}^O\mathbf{P}\right)$ $=R_i 2R_O\_State({}^i\mathbf{x}, {}^i\mathbf{P}, \theta_i)$
    $\left(j, {}^O A, {}^O B\right)$=Map_Match$\left({}^O\mathbf{x}, {}^O\mathbf{P}, C_0, \text{Map}\right)$
    if $\theta_i \neq \theta_j$ then
        $({}^i\mathbf{x}, {}^i\mathbf{P})$ $=R_i 2R_j\_State({}^i\mathbf{x}, {}^i\mathbf{P}, \theta_i, \theta_j)$
        $\theta_i = \theta_j$
    end if
    ${}^i A = R_O 2R_i\_Point\left({}^O A, \theta_i\right)$
    ${}^i B = R_O 2R_i\_Point\left({}^O B, \theta_i\right)$
    $({}^i\mathbf{x}, {}^i\mathbf{P})$ =Update_Camera$({}^i\mathbf{x}, {}^i\mathbf{P}, C_0, {}^i A, {}^i B)$
end if
$\left({}^O\mathbf{x}, {}^O\mathbf{P}\right)$ $=R_i 2R_O\_State({}^i\mathbf{x}, {}^i\mathbf{P}, \theta_i)$        ▷ /*System output*/
                                              ▷ /*See Algorithms 4.1, 4.2, 4.4 and 4.5*/
                                              ▷ /*The Kalman filter functions are expressed in the text*/
```

4.3.3.1 Prediction

The proprioceptive sensors are available during a sampling period T_e, the function Predict($^i\mathbf{x}, ^i\mathbf{P}, \mathbf{u}$) consists in computing:

$$^i\mathbf{x} = f\left(^i\mathbf{x}, \mathbf{u}\right) \iff \begin{cases} ^ix =^i x + T_e \cdot v^m \cdot \cos\left(^i\psi\right) \\ ^iy =^i x + T_e \cdot v^m \cdot \sin\left(^i\psi\right) \\ ^i\psi = {^i\psi} + T_e \cdot \left(\omega^m - {^i\varepsilon_\omega}\right) \\ ^i\varepsilon_\omega = {^i\varepsilon_\omega} \\ ^i\varepsilon_{x1} = a_1 \cdot {^i\varepsilon_{x1}} \\ ^i\varepsilon_{x2} = a_2 \cdot {^i\varepsilon_{x2}} \\ ^i\varepsilon_{y1} = a_1 \cdot {^i\varepsilon_{y1}} \\ ^i\varepsilon_{y2} = {^i\varepsilon_{y2}} \end{cases} \tag{4.34}$$

and

$$\begin{cases} ^i\mathbf{P} = \mathbf{A}^i\mathbf{P}\mathbf{A}^T + \mathbf{B}\mathbf{N}\mathbf{B}^T + \mathbf{Q} \\ \mathbf{A} = \frac{\partial f\left(^i\mathbf{x},\mathbf{u}\right)}{\partial^i\mathbf{x}}, \mathbf{B} = \frac{\partial f\left(^i\mathbf{x},\mathbf{u}\right)}{\partial\mathbf{u}} \end{cases} \tag{4.35}$$

with $a_1 = e^{-Te/\tau_1}$ and $a_2 = e^{-Te/\tau_2}$. v^m and ω^m denote the measured inputs, $\mathbf{u} = [v^m, \omega^m]^T$. The measurement noises on v and ω are supposed to be zero-mean independent white noises, $\mathbf{N}$ denotes their covariance matrix. $\mathbf{Q}$ is the covariance matrix of the process noise.

4.3.3.2 GNSS update

When GNSS data is available, the measurement vector $^O\mathbf{y}_{GNSS} = \left(^Ox_{GNSS}, ^O y_{GNSS}\right)$ in the ENU frame R_O is transformed to the working road-oriented frame R_i (see Figure 4.7) by Algorithm 4.4. The covariance matrix $^O\mathbf{R}_{GNSS}$ given by the receiver is also converted.

$$^O\mathbf{R}_{GNSS} = \begin{bmatrix} \sigma_e^2 & \sigma_{e,n}^2 \\ \sigma_{n,e}^2 & \sigma_n^2 \end{bmatrix} \tag{4.36}$$

Algorithm 4.4 shows the transformation.

Then, a classical Kalman update step is performed to update ($^i\mathbf{x}, ^i\mathbf{P}$) with an innovation gating to reject the GNSS fix outliers (e.g. multipath on close buildings).

Algorithm 4.4 Function $R_O 2 R_i$

```
Input: ${}^{O}Y_{GNSS}, {}^{O}R_{GNSS}, \theta_i$
$ {}^{i}T_O = \begin{bmatrix} cos\theta_i & sin\theta_i \\ -sin\theta_i & cos\theta_i \end{bmatrix} $
$ {}^{i}Y_{GNSS} = {}^{i}T_O \cdot^{O} Y_{GNSS} $
$ {}^{i}R_{GNSS} = {}^{i}T_O \cdot^{O} R_{GNSS} ({}^{i}T_O)^T $
Output: ${}^{i}Y_{GNSS}, {}^{i}R_{GNSS}$
```

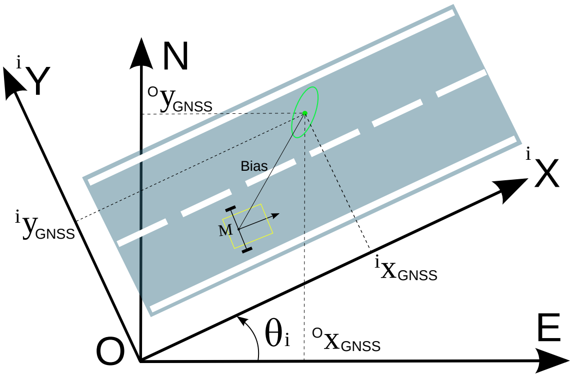


Figure 4.7: *GNSS fix error and frame transformation*

For an accurate data fusion, the level arm of the antenna with respect to the body frame has to be taken into account:

$$\begin{cases} x_{GNSS} = \cos\psi \cdot t_x - \sin\psi \cdot t_y + x + \varepsilon_{x1} + \varepsilon_{x2} + \beta_x^{'} \\ y_{GNSS} = \sin\psi \cdot t_x + \cos\psi \cdot t_y + y + \varepsilon_{y1} + \varepsilon_{y2} + \beta_y^{'} \end{cases} \tag{4.37}$$

4.3.3.3 Camera update

In order to update the filter with a camera measurement, the map is used as it contains the coordinates of the lane markings. Map matching is therefore done only when camera measurements are available. Its goal is to find the road that matches with the detected lane marking. Since the map is defined in R_O, the first step consists in converting the estimated pose into ENU frame. Map matching is then performed firstly to find the closest road segment by a point-to-semgent map matching (see the structure of the map in 2.5), and the closest lane marking is chosen as the detected lane marking as described in section 3.6.3. At this moment, the algorithm checks if the vehicle is in a different road than the one previously matched. In this case, the road working frame is modified and the state with its covariance matrix is converted. It just remains to get the coordinates of the lane marking in the road frame which is described by Algorithm 4.5 and to update the state and covariance matrix by applying an estimation stage of the EKF. In practice, the location of the camera in the body frame is taken into account to get an accurate correction as in section 3.6.3.

Algorithm 4.5 Function $R_O 2R_i$_Point

```
Input: $^{O}A, \theta_i$
$^{i}T_O = \begin{bmatrix} cos\theta_i & sin\theta_i \\ -sin\theta_i & cos\theta_i \end{bmatrix}$
$^{i}A = {}^{i}T_O \cdot {}^{O} A$
Output: $^{i}A$
```

Remark 3. Please note that, in the road-invariant EKF, the GNSS error modeling in lateral and longitudinal directions is not symmetric. If the GNSS bias is modeled as in Chapter 3 or in [119], there is no need to implement the EKF in the road frame. In the published paper [119], the GNSS was modeled as the combination of a random bias and an autoregressive bias in both lateral and longitudinal direction. In this case, the observability is obtained when the road orientation changes. In other words, if the vehicle drives in a very long straight road for a long time, there is no way to distinguish between the random bias and the autoregressive in the longitudinal direction. The road-invariant EKF proposed in this chapter is better because the GNSS bias modeling is guaranteed to be fully observable in any situation.

Localization results are presented in the following.

4.4 Results

In this section, some simulation results are first presented. The sensor measurements and the vehicle trajectory ground truth is defined by equation (4.21). The objective of this study is to give a better understanding of the system observability. Then, we will discuss real experimental results.

4.4.1 Simulation results

The gyro and GNSS errors (biases and colored noises) have been simulated. They are denoted as $\left(\varepsilon_{\omega}^{truth}, \varepsilon_{x1}^{truth}, \varepsilon_{x2}^{truth}, \varepsilon_{y1}^{truth}, \varepsilon_{y2}^{truth}\right)$. Figure 4.8 gives the simulated gyro bias and its estimation error with a 3σ bound. The gyro bias is well estimated. Figure 4.9a shows the simulated GNSS biases (in black) and their estimation (in red). One can notice that the autoregressive biases $(\varepsilon_{x1} + \varepsilon_{x2})$ and ε_{y1} are well estimated. In addition, the random constant bias ε_{y2} converges well to the true value. Figure 4.9b shows the bias estimation error with 3σ bounds.

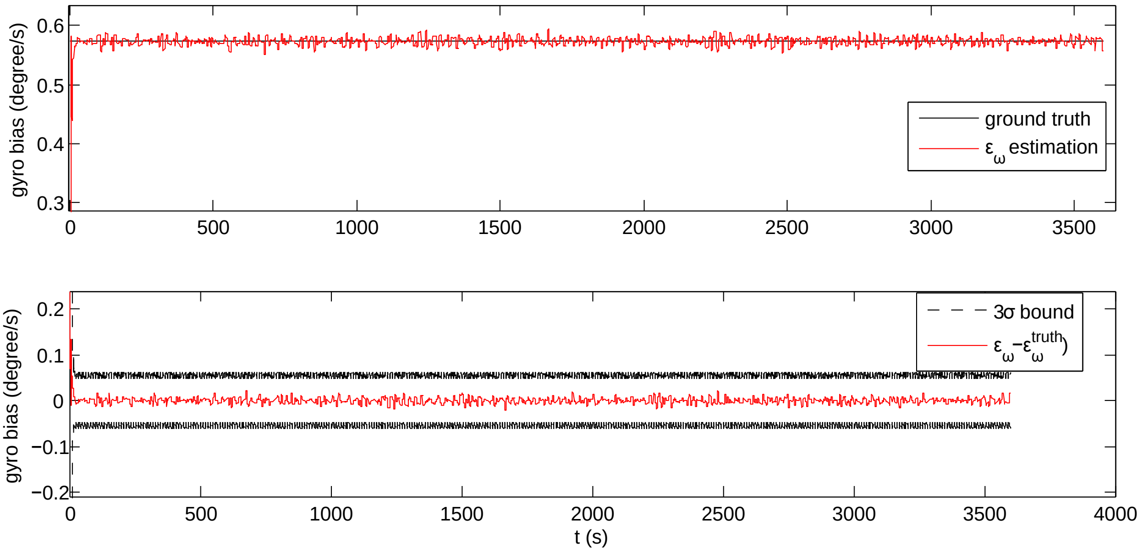


Figure 4.8: *Estimated gyro bias compared with ground truth*

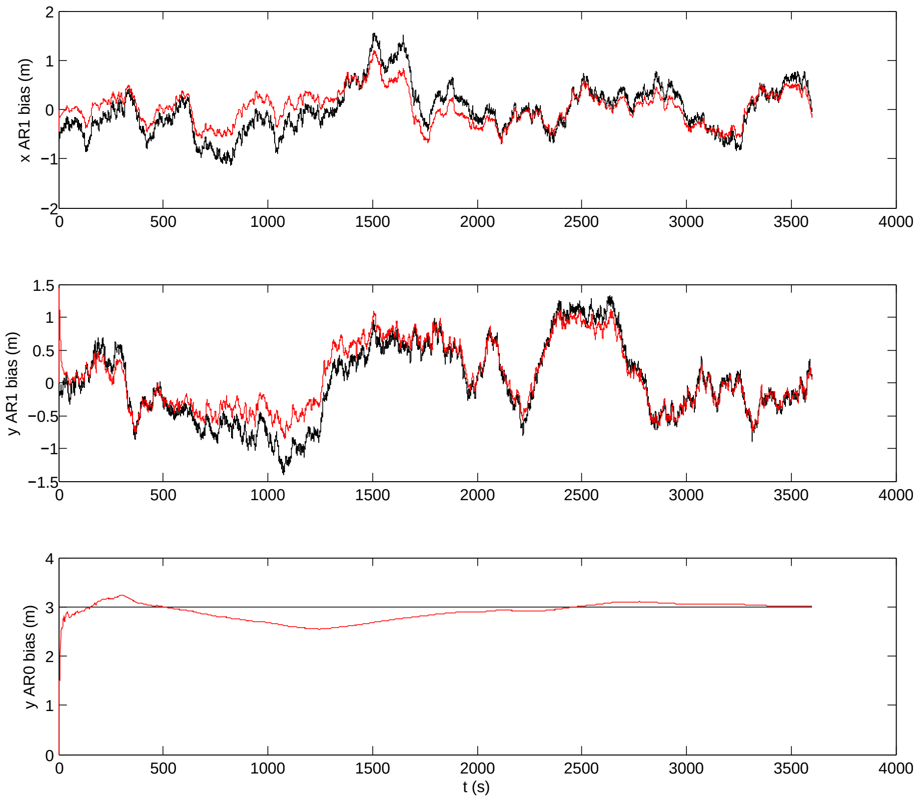
These simulation results support well the observability analysis done in section 4.2. If a GNSS random constant bias is added in the x (longitudinal) direction, Figures 4.10a and 4.10b confirm that it is not observable since the filter is not able to make the estimation error to converge towards zero. In other words, it means that the enhanced modeling of GNSS bias is only observable in y (lateral) direction. This result supports the proposition of the road-invariant EKF.

4.4.2 Outdoor experimental results

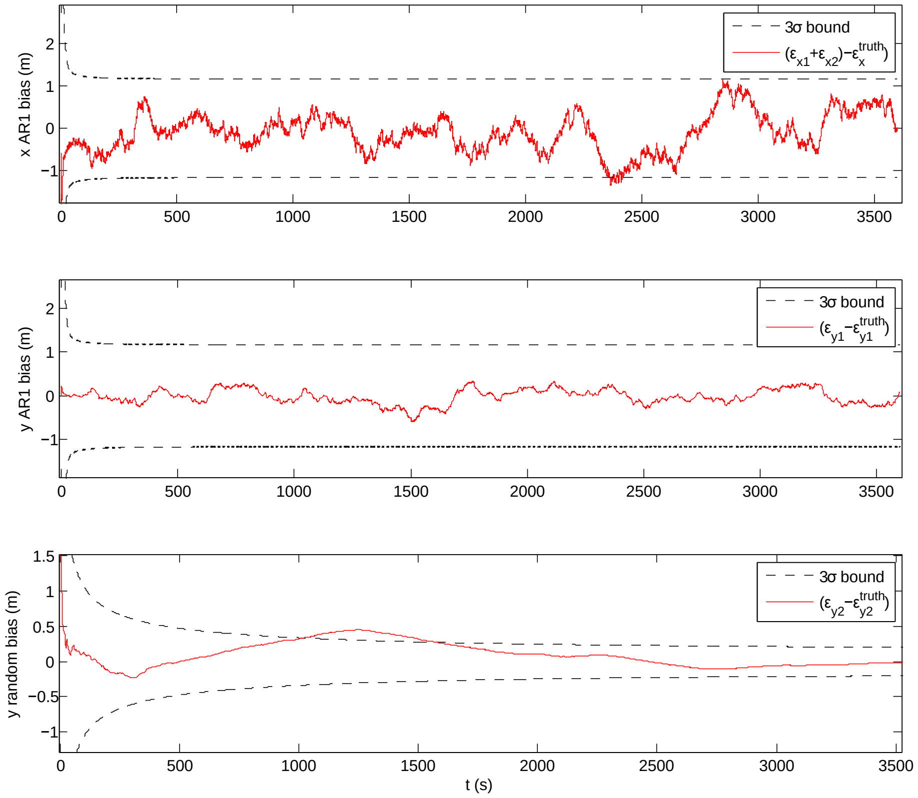
The method proposed in this chapter is compared with the loosely coupled EKF implemented in the ENU frame R_O, in which the GNSS biases on x and y are modeled only by first order autoregressive processes to make sure that the observability is kept (see chapter 3). As described in Algorithm 4.3, the output of the localization solver is converted into frame R_O. So the localization performance of the two approaches can be studied in R_O. Here, the results have been carried out by using data replay and both filters have been implemented in C++ to test them with exactly the same framework and with the same tuning.

In 4.11, the green bar indicates the orientation of the x-axis and the red bar the y-axis. The blue lines represent polylines of the ways and the white lines are the lane markings. The black ellipse represents the estimated confidence domain (3σ).

Table 4.1 gives global performance metrics of three tests obtained by the u-blox receiver alone, the EKF in the ENU frame (ENU EKF) and the road invariant EKF. Lateral and longitudinal positioning errors (PEs) are analyzed and compared. One

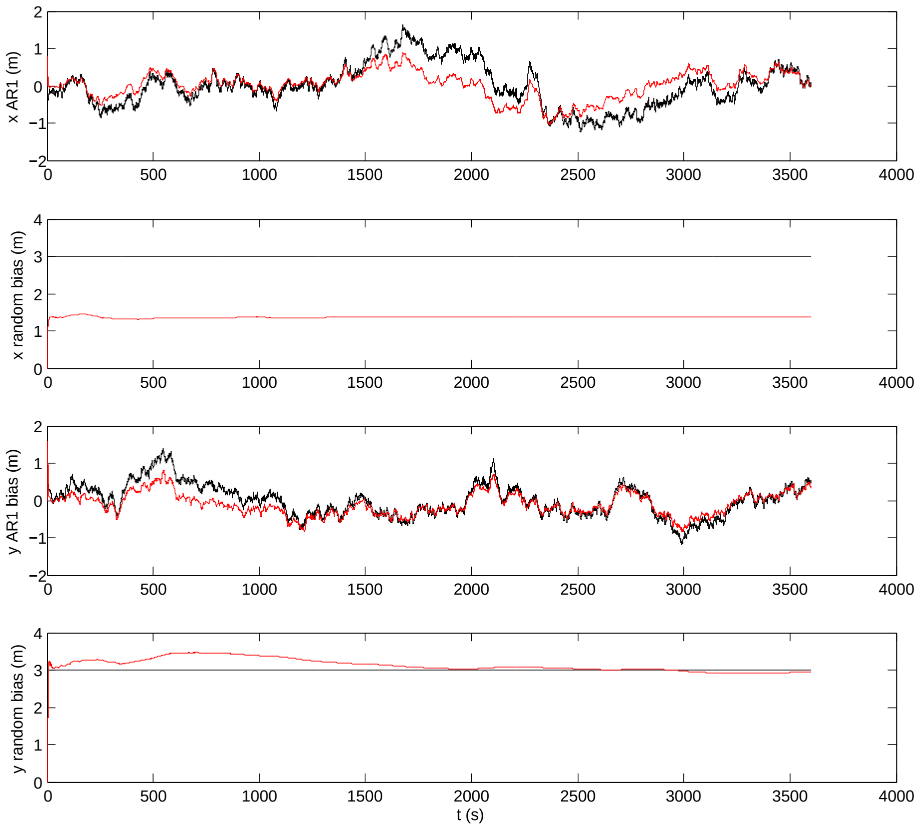


(a) *Estimated GNSS biases*

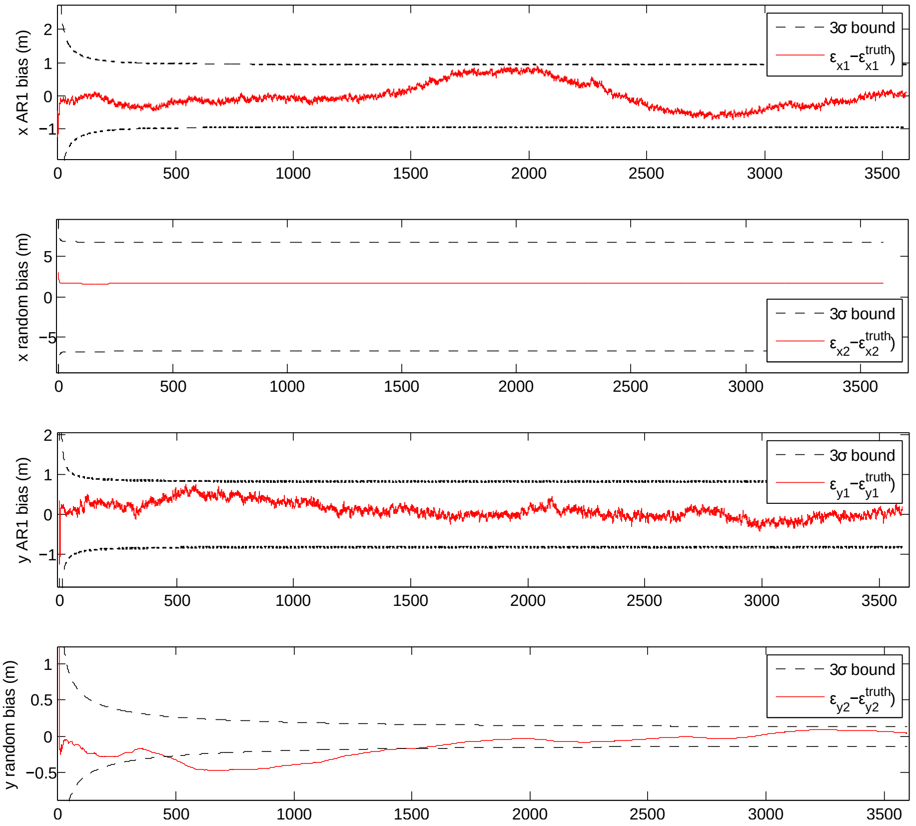


(b) *Estimation errors of GNSS biases with error bound*

Figure 4.9: *Simulation results*



(a) *Estimated GNSS biases*



(b) *Estimation errors of GNSS biases with error bound*

Figure 4.10: *Simulation results with an additional random constant model in the longitudinal direction*

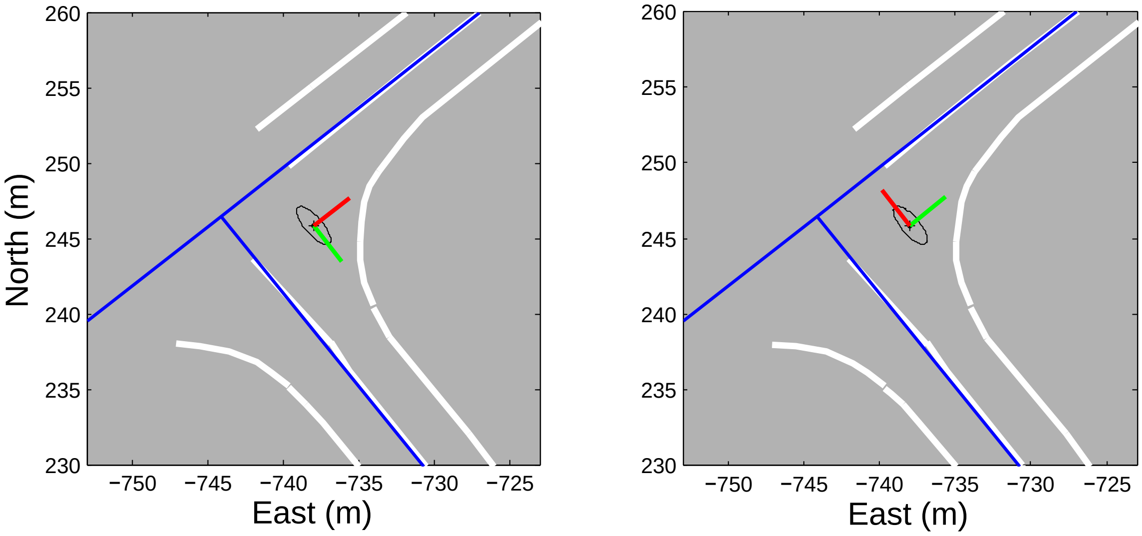


Figure 4.11: *Illustration of a road frame change from one road to another during a real experiment. The vehicle is passing from the southeast toward road to the northeast toward one.*

	Lateral PE (m)			Longitudinal PE (m)		
	I	II	III	I	II	III
mean	1.30	0.07	0.04	1.55	-0.32	-0.19
std. dev.	1.12	0.29	0.26	1.18	0.32	0.29
median	0.96	0.10	0.09	1.31	0.30	0.24
95th percentile	3.20	0.68	0.55	3.88	0.88	0.73
max	6.78	1.83	1.37	4.69	1.50	1.36

Table 4.1: *Error statistics. (PE: positioning error; I: u-blox; II: ENU EKF; III: road invariant EKF)*

can notice that the localization accuracy is highly improved by using the road invariant EKF with 95% lateral positioning error less than 0.55m. Table 4.2 gives the relative improvement reached by the road invariant EKF with respect to the EKF in the ENU frame, in terms of median, 95th percentile and maximum of the lateral and longitudinal positioning error. Figure 4.12 displays the cumulative distribution of the absolute positioning errors by the road invariant and ENU EKFs of the three tests. The road invariant EKF gives better estimation on both lateral and longitudinal directions.

Hereafter, only the results of the first test are taken into consideration to do a more deep analysis. Figure 4.13 shows changes of lateral and longitudinal positioning errors over time with $\pm 3\sigma$ bounds by the road invariant EKF. One can notice that,

	median	95th percentile	max
Lateral Positioning	10%	19%	25%
Longitudinal Positioning	20%	17%	9%

Table 4.2: *Improvement by road-invariant EKF compared to the ENU EKF*

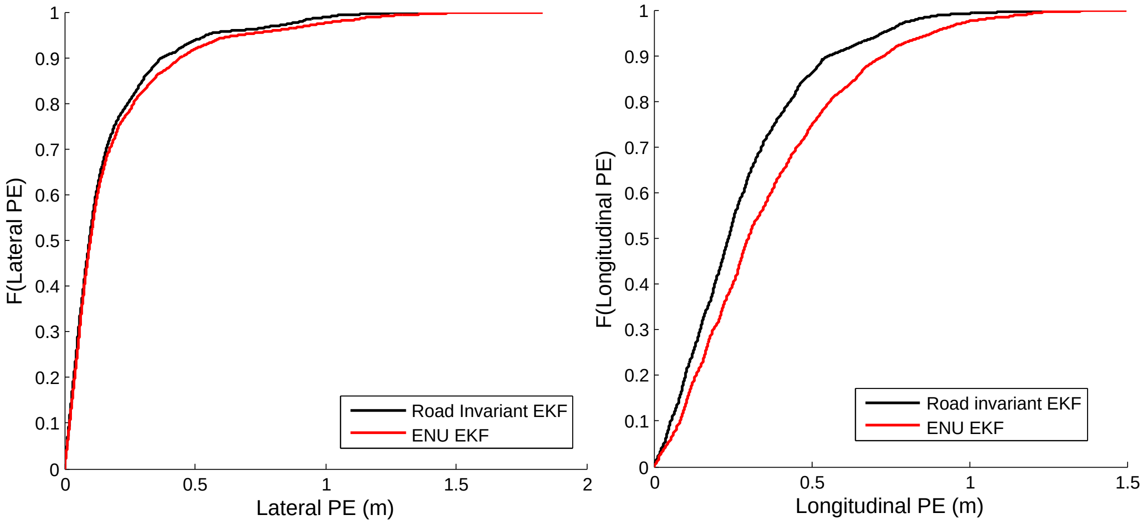


Figure 4.12: *Cumulative distribution function (CDF) of the positioning lateral and longitudinal errors*

the uncertainty on lateral position increases greatly when there are no lane marking measurements. The road invariant EKF remains consistent (99.7% probability level) for all but 5.8% of the data points. The consistence failure rate of ENU EKF is much higher since it reaches 24.4%. Figure 4.14 shows the positioning errors over time by the two approaches.

4.4.3 Convergence analysis of the GNSS biases

Figure 4.15 depicts the estimated standard deviation of ε_{x1}, ε_{x2}, ε_{y1}, and ε_{y2} in R_O. It has to be noticed that they converge towards constants, except when there is no lane marking detection. This is an experimental checking of the observability. By looking at the biases estimates (Figure 4.16), one can see that they remain bounded and in the order of magnitude of usual L1-GNSS errors.

Figure 4.17 shows that the sums of the estimate biases in both direction of R_O match very well the bias computed with the ground truth equipment.

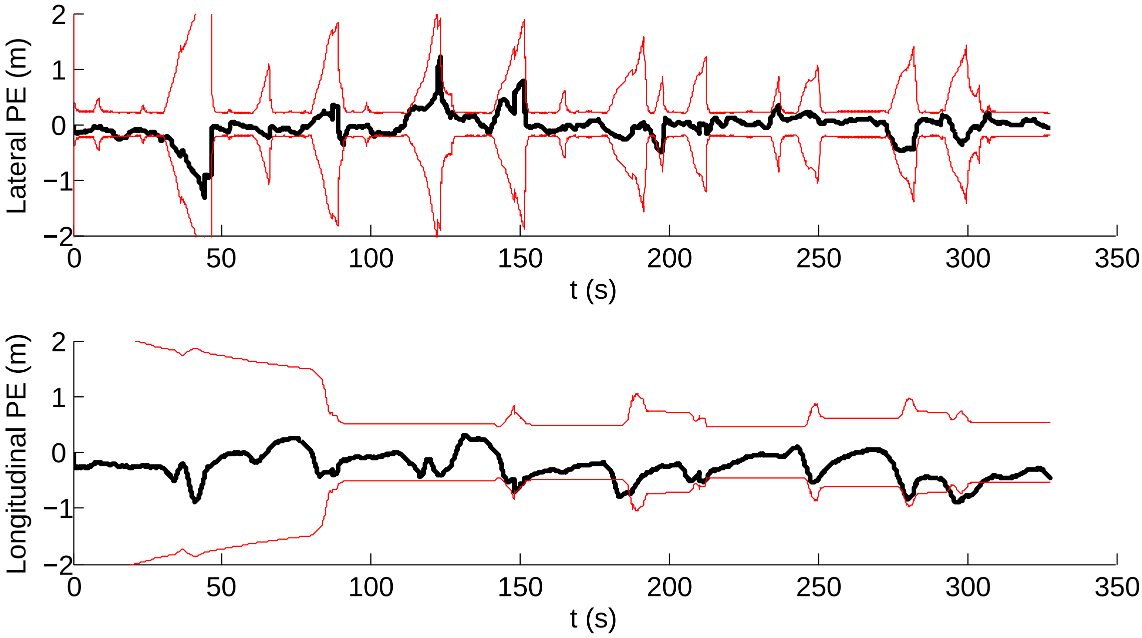


Figure 4.13: *Lateral and longitudinal PE with the road invariant EKF*

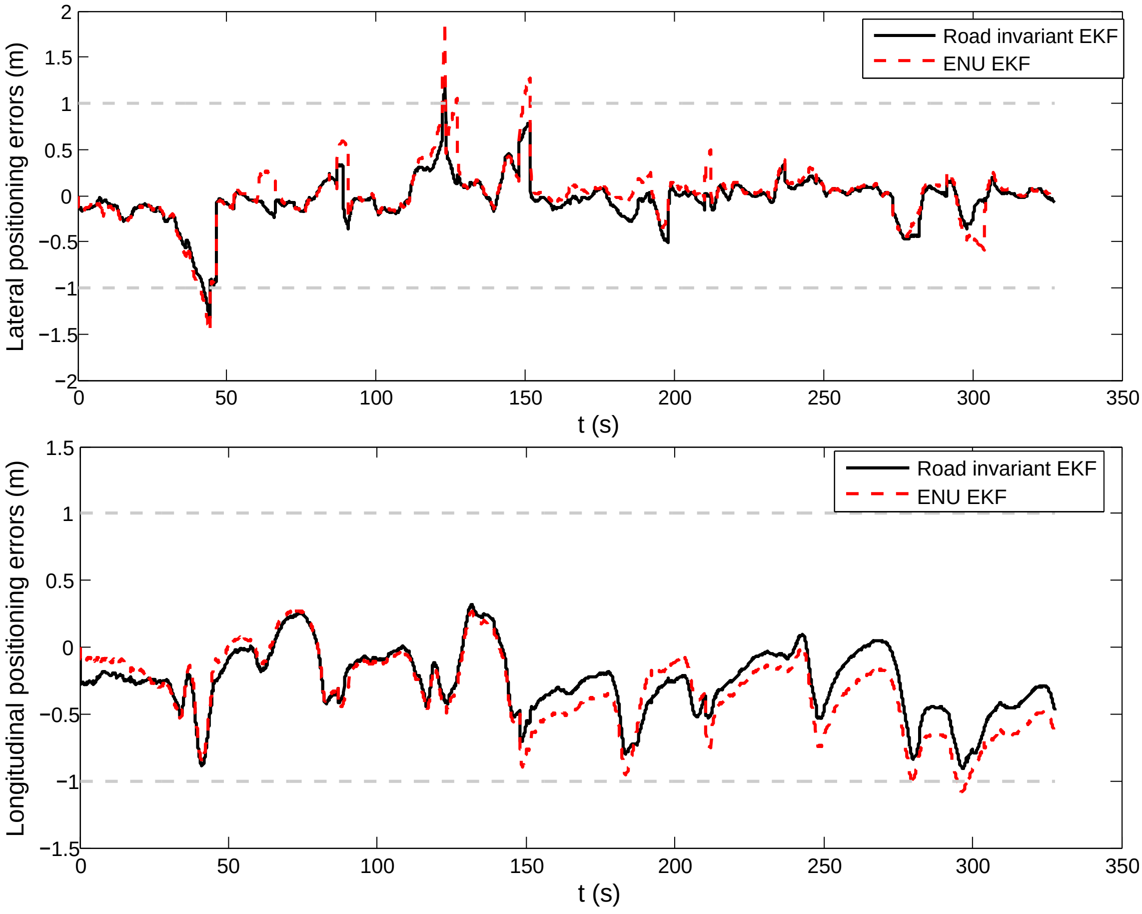


Figure 4.14: *Lateral/longitudinal PE by road invariant and ENU EKFs*

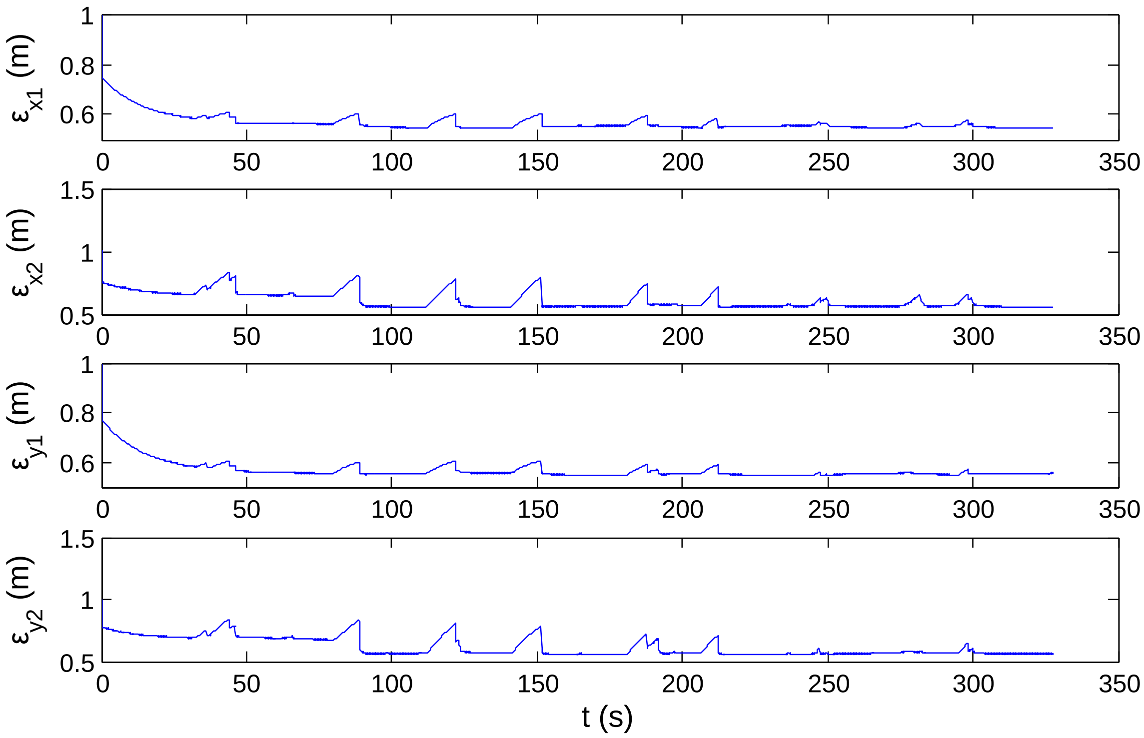


Figure 4.15: *Estimated standard deviation of the GNSS biases*

In Figure 4.18, the CDFs of GNSS biases estimation errors when using the road invariant EKF and the ENU EKF are compared. Road invariant EKF generally produces a smaller estimation error. This clearly indicates that the GNSS errors are better estimated by the road invariant EKF.

4.4.4 Robustness of the Road Invariant EKF

4.4.4.1 Robustness to GNSS multipath

In Figure 4.19, the yellow box indicates a period during which a GNSS multipath happens because of a urban canyon. For instance, during the time interval t = 120-128s of test 1, the GNSS receiver suffers from multipath effect. The green and blue points indicate lane marking detections on left and right sides respectively. One can notice that unfortunately there is no lane marking measurements when the multipath effect begins. The lateral positioning error of the road invariant EKF reaches 1.2m, while ENU EKF reaches 1.83m, and the lateral PE of the road invariant EKF is overall smaller than the one of the ENU EKF during the multipath effect. The camera retrieves a lane marking measurement at the right side of the lane at t = 122.2s. The lateral PE of the road invariant EKF is quickly reduced to less than 0.5m.

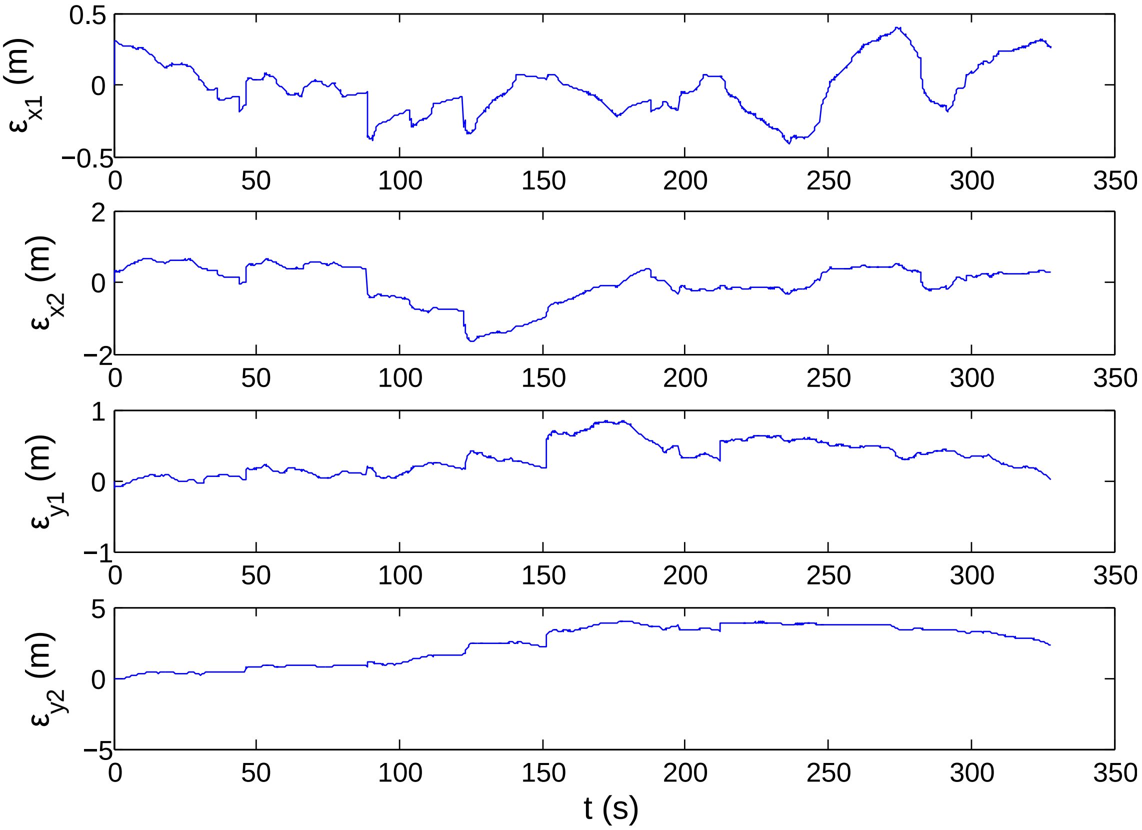


Figure 4.16: *Estimated biases in* R_O

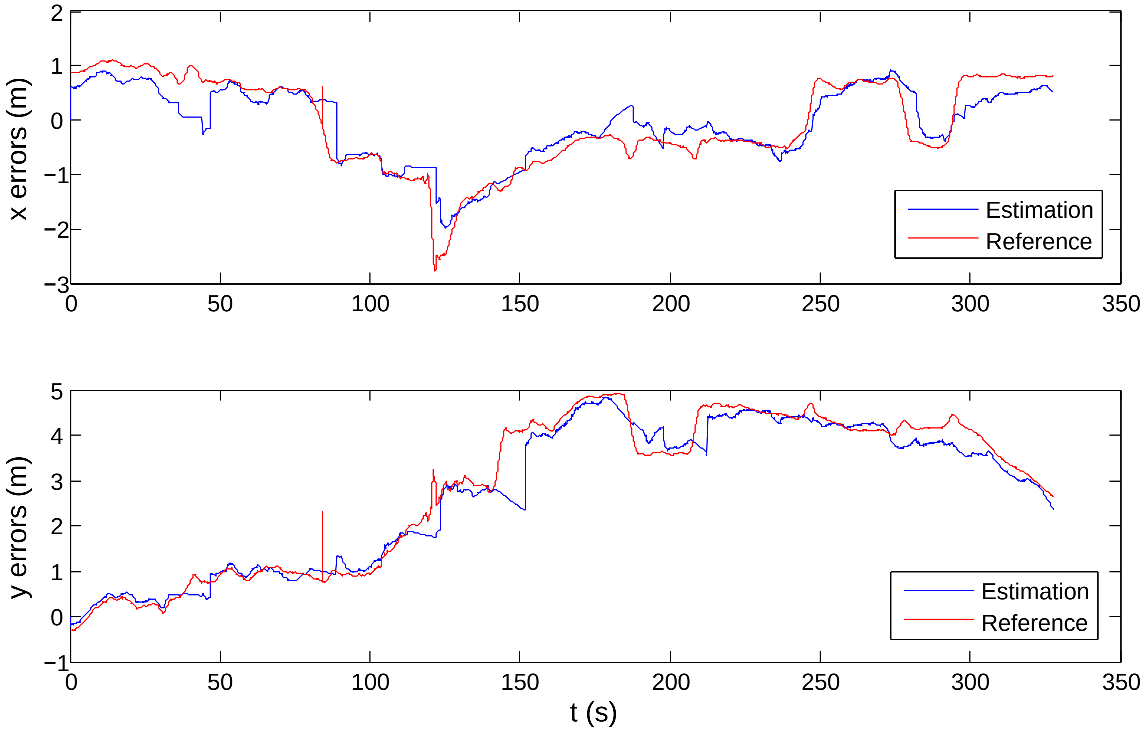
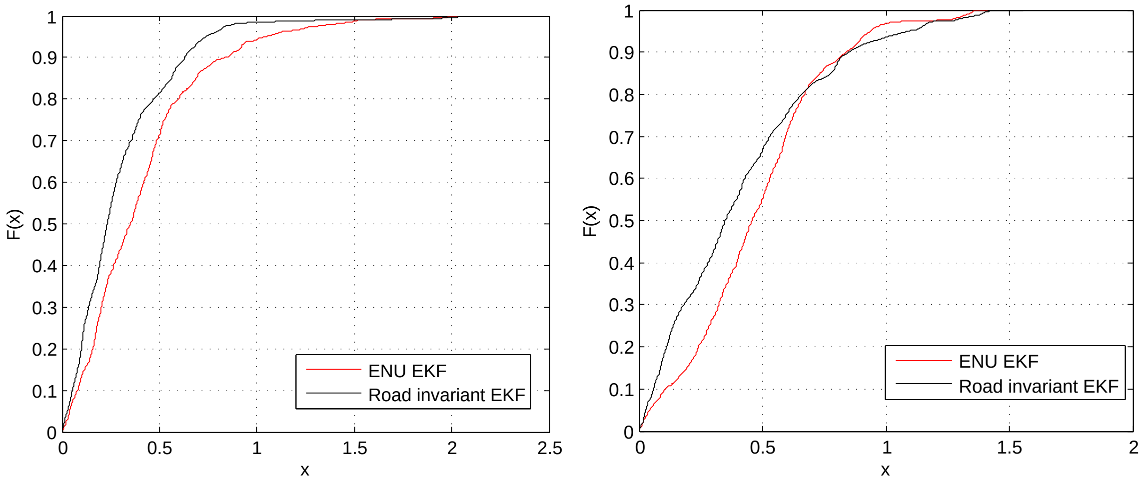


Figure 4.17: *Estimated biases compared to the reference*



(a) *GNSS bias estimation error on x* **(b)** *GNSS bias estimation error on y*

Figure 4.18: *Comparison of the CDFs of GNSS biases estimation errors when using the road invariant EKF and the ENU EKF described in chapter 3.*

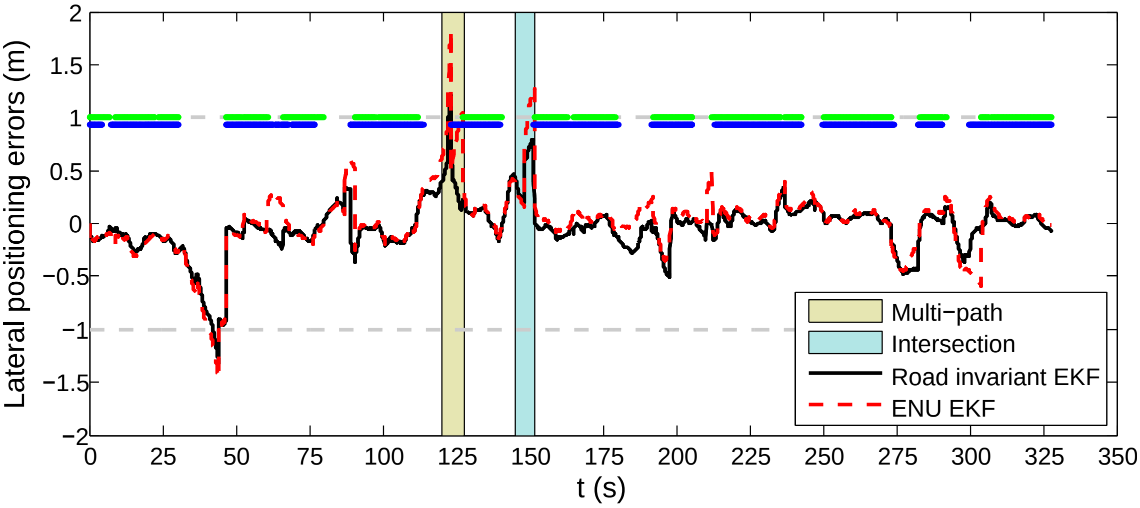


Figure 4.19: *Lateral positioning error during multipath and intersection*

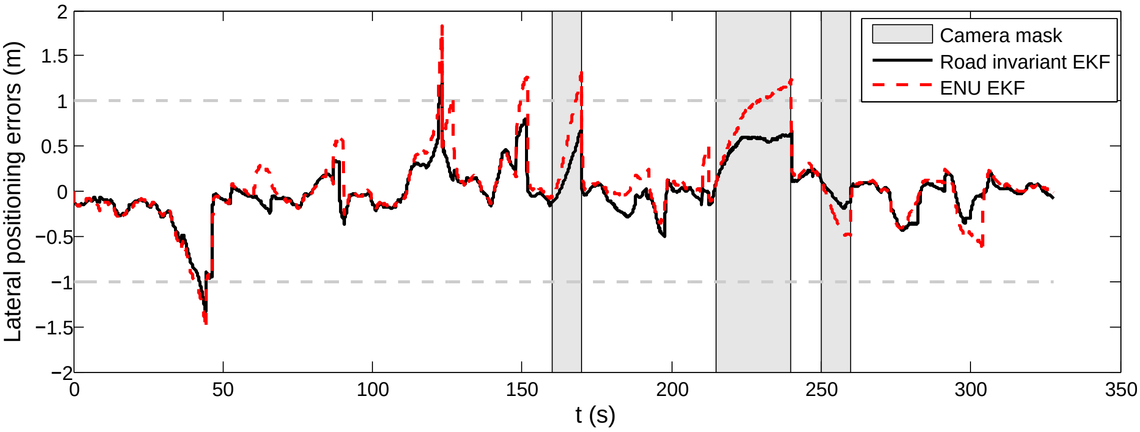


Figure 4.20: *Lateral positioning errors with camera measurement masks*

4.4.4.2 Robustness to outages of the camera lane marking measurements

Now, let us see what happens in an intersection without any camera detection.

In Figure 4.19, the blue box indicates a period during which the vehicle is passing through an intersection. There is no lane marking detection in this kind of situation. One can notice that the lateral positioning performance of the road invariant EKF is much better during this period and the lateral errors remain bounded by 1m.

In order to further validate the robustness to outages of lane marking measurements, we have simulated camera outages at t = 160-170s, 215-240s and 250-260s of test 1 (the gray boxes in Figure 4.20) when the vehicle was driving straightly (cf. Figure 2.22). Results show that the lateral positioning of the road invariant EKF is again much better than the one of the ENU EKF. The positioning errors are less than 0.5m which indicates a lane level positioning capability.

4.5 Conclusion

In this chapter, we have proposed an enhanced state modeling which handles well the time-correlated errors and bias of L1-GNSS in a road frame. A road invariant EKF algorithm has been proposed and tested. In particular, the proposed state space model is observable and a bijective transformation between roads guarantees the continuity of the Kalman filter estimates. Observability analysis has been used to design the modeling of the system. The method has been tested in simulation and validated with real outdoor data. It works much better than a localization solver implemented in the ENU frame in terms of accuracy and consistency because it handles a more refined modeling of the errors. The roads of the test area being quite orthogonal with each others, we haven't tested yet the behavior of the road invariant EKF on roads with less varying geometries such as sinusoidal roads. This analysis remains to be done in future work.

It has been noticed that a large estimation error occurs during t = 120-128s for both loosely coupled approaches in constraint environments with multipath effects. In the next chapter, a tightly coupling approach is studied to improve the performance especially in this case of degraded environment for the propagation of the signals of the satellites.

Chapter 5

GNSS map-aided localization with a tightly coupled scheme

Contents

5.1 Introduction

Two loosely coupled schemes using GNSS position fixes have been studied in Chapter 3 and 4. The localization system works well most of the time. The integration of the camera measurement makes possible the estimation of GNSS biases and the bias shaping filter bridges well the camera measurement outages in the considered scenarios. However, when a camera measurement outage and GNSS multipath effect happen at the same time (which is a case we encountered in our experiments several times), the localization system has troubles to keep a lane-level accuracy before the camera measurement is retrieved. It happens particularly in urban areas where less than 4 reliable satellites are in view. The GNSS position fix is either not available nor consistent if the receiver is tuned to be very sensitive which makes the engine compute positions with satellites suffering from multipath. In this last case, if the localization solver fails to reject the GNSS fix contaminated by unpredictable multipath errors, the shaping filter can no more whiten the GNSS bias. Thus, the estimated confidence of the localization solver output can be inconsistent with respect to ground truth.

Considering the aforementioned problems, this chapter looks at a tightly coupled approach which employs raw pseudoranges and Dopplers from each satellite in view.

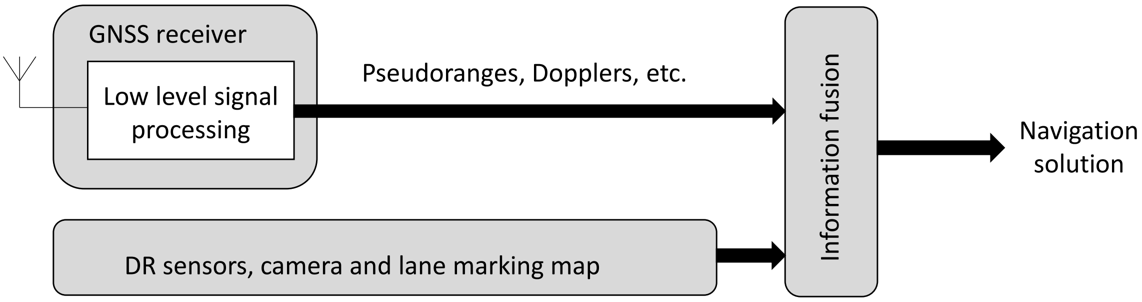


Figure 5.1: *Tightly coupled scheme*

In this case, GNSS measurement updates of the localization estimates can be performed even when the number of visible satellites are only three or fewer. This can improve the availability of the navigation system in degraded GNSS environments. In addition, as the filter processes GNSS signals directly, it increases the chance of optimal filtering performance and integrity monitoring possibilities. On the other hand, the need to handle raw GNSS data (e.g. calculation of satellite clock corrections, satellite positions and velocities from ephemeris data) complicates the design process of a tightly coupled system [130]. For a tightly coupled scheme, the accuracy of the range measurements plays an important role on the correctness of the position estimate. The overall position accuracy can be improved if the measurement errors (in particular at the pseudorange level) are accurately estimated. In a similar way to the shaping filter studied in Chapter 3, it is expected to estimate the GNSS bias at the pseudorange level by integrating the camera measurement.

This chapter is organized as follows. In section 5.2, works related to tightly coupling GNSS with pseudorange bias estimation are reviewed. Section 5.3 presents the system modeling of our approach. Section 5.4 presents experimental results and section 5.5 concludes this chapter.

5.2 Related work and problem statement

Before 2000, the SA (Selective Availability) was the greatest source of positioning error for civilian GPS users with an horizontal positioning error up to 100m typically. Many earlier works on tightly coupling GPS addressed this issue of inferring the SA systematic error with digital maps. SA made it difficult to determine which road a car was on, particularly in areas where several ways run in parallel. This inaccuracy could sometimes give erroneous turn information for navigation systems. Although SA has been disabled since 2000, the idea of using digital map to compensate for GPS errors (especially for errors on pseudoranges) is still instructive for our work. In [120], the authors combine raw GPS pseudorange measurements, road centerline geometry and DTM (Digital Terrain Model) data to improve the positioning quality. A raw vehicle position is first computed using all satellites available plus height aiding. The effect of SA is viewed as a slowly varying bias in short term. It can be subtracted from position measurements provided that a measure of the bias can be made at regular intervals which is the fundamental principle of DGPS. Therefore, road geometry-derived corrections for pseudoranges are computed by comparing the raw position and the position snapped onto the road segments. Then, a GPS fix is computed using the corrected pseudoranges. The authors call this strategy virtual DGPS. However, the along track error cannot be resolved for a straight road. An alternative way of exploiting digital maps is to use them as a heading sensor by considering that map precision is often better than map accuracy [42]. Map matching with a digital map gives then useful feedback to correct GNSS errors. That is mainly because standalone GNSS accuracy was lower than the one of maps. GNSS accuracy was so highly improved after the removal of SA. This led to a situation where the positioning accuracy had the same level of the one as the geometry of the digital map (e.g. several meters accuracy). New emerging strategies seeks to integrate vision measurements and enhanced maps.

Positioning accuracy is highly dependent on the accuracy of the pseudoranges. When integrating with vision/map coupled measurements, a pseudorange observation is often modeled as:

$$\rho^i = R^i + c \cdot dt_u + \varepsilon^i + \beta^i \tag{5.1}$$

where ε^i is an observable bias and β^i is an unobservable noise for satellite i.

In [126], the authors tightly couple pseudoranges and Dopplers with vision measurements and DR sensors. ε^i represents a common-mode error (the atmospheric error and the ephemeris error coming from the computed satellites positions) and β^i is mainly caused by thermal noise and multipath. ε^i is removed by differential GPS technology. The authors conclude that a single visual feature measurement at 1 Hz is able to achieve a submeter-level accuracy. In [106], the authors propose a method for collaboratively sharing georeferenced vision measurements and GNSS pseudoranges to calibrate pseudorange errors for each satellite in view. The advantage is that this method needs no stationary reference receiver. ε^i is also modeled as a common-mode error. Sharing of lane-boundary measurements can make ε^i fully solvable (in the least-squares sense) for networked GNSS receivers. For a single-user estimation problem (which is our case) and with a snapshot solver (epoch by epoch - which is not our case since we do state filtering), the authors also prove that the lane-boundary measurement makes it possible to estimate only the transverse part of ε^i in the road. In order to make the pseudorange errors observable, the proposed collaborative method estimates the correlated biases ε^i among different users that are on different roads and can observe different part of the errors.

Under a single-user framework, the number of unknowns (pseudorange bias) grows linearly with the number of measurements. It results that measurements are not enough to solve the problem as in the collaborative way. So, we model ε^i as a time correlated bias and try to estimate it in a dynamic filter as suggested in [13][30]:

$$\begin{cases} \varepsilon^i_{k+1} = \lambda \varepsilon^i_k + w_\varepsilon \\ \lambda = e^{-T_e/\tau_{pr}} \end{cases} \tag{5.2}$$

In [13], the pseudorange bias is modeled as a stationary first-order autoregressive process driven by zero-mean white noise w_ε. A time constant $\tau_{pr} = 600$s is suggested and used in [89]. T_e is the sampling period. The dimension of the state of the filter is increased by the number of visible satellites because each pseudorange measurement noise requires a first-order shaping filter.

5.3 Tightly coupling L1-GNSS

In this section, we use a pseudorange observation model similar to equation (5.1). ε^i is considered as specific to every satellite in view since the propagation paths of GNSS signals and orbit errors are different between satellites. The basic idea is to enhance the estimation of the pseudorange bias of the satellites in view by integrating lane marking measurements.

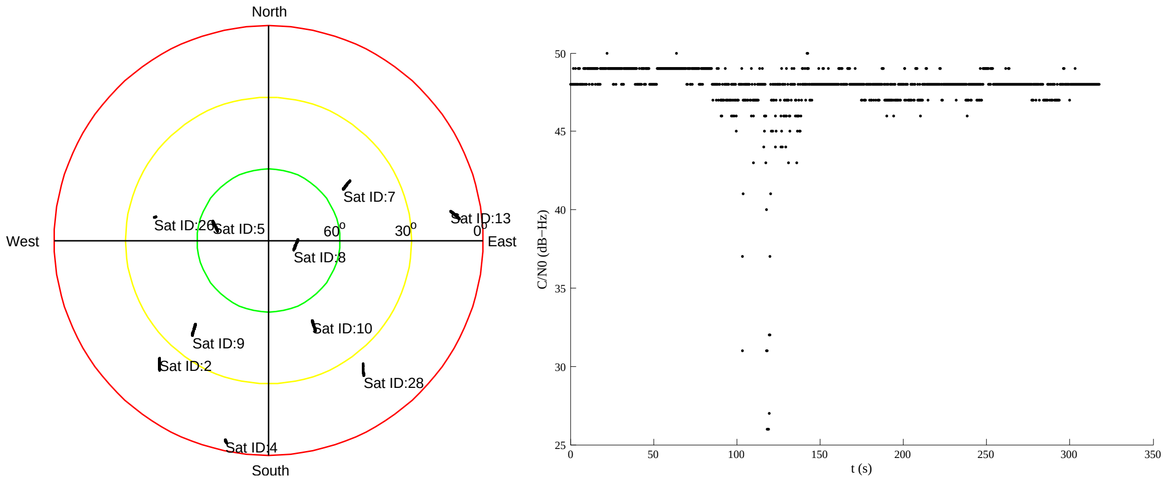
5.3.1 GNSS Measurements Used in a Tightly Coupled Scheme

A L1-GNSS receiver tracks all the satellites in view. We have used a u-blox 6T receiver with a patch antenna as mentioned in Chapter 2. It provides measurements and navigation information with respect to each GNSS satellite in view as follows:

- Position of the satellite at the time of emission
- Velocity of the satellite at the time of emission
- Pseudorange C/A on L1
- Doppler on L1
- Signal-to-noise ratio (C/N_0)
- User range accuracy (URA)

These raw data are not directly provided by the receiver. GNSS receiver raw data and corrections are computed using the GPSTK library (see Appendix B). The GNSS signal and the basic observation models have been introduced in Chapter 2. Hereafter, we recall briefly how a GNSS receiver gets the above information.

The GNSS receiver uses ephemerid data to calculate the position and velocity of the satellites. The ephemerid is broadcast in real-time through navigation messages and contains the orbit of each satellite refreshed periodically by the control segment.



(a) *Satellites configuration with their trajectories during test 1* **(b)** *C/N_0 variation over time for satellite number 8*

Figure 5.2: *Illustration of satellites configuration and C/N_0 variation during a test*

The GNSS receiver keeps a delay lock loop (DLL) for code tracking and a phase lock loop (PLL) for carrier phase tracking (see Chapter 2). The measurements produced by the DLL are the code pseudoranges. The Doppler shift is computed by tracking the frequency of the received signal and its value is immune to cycle slips [11]. C/N_0 refers to the ratio of the carrier power with the noise power per unit bandwidth. Higher C/N_0 results in less noisy pseudorange and Doppler measurements [26]. URA is a 1-sigma estimate of the user range error in the navigation data for the transmitting satellite. It includes all errors for which the space and control segments are responsible but errors introduced by the user or the transmission media are not included [31].

When a satellite appears at the horizon, the C/N_0 is in the range of 30-40 dB-Hz and increases up to a maximum of slightly over 50 dB-Hz once the satellite is at 50-60 degrees elevation angle or higher. As shown in Figure 2.25a, there were a total of 10 satellites in view over the outdoor experimental area for test 1. Take the highest satellite (satellite 8) as an example, Figure 5.2b displays the variation of its C/N_0 value over time. Around t = 120s, the vehicle passes through an urban area and so the C/N_0 degrades rapidly during this period. Low C/N_0 can also arise because of unintentional interference, deliberate jamming or weak signals [51].

5.3.2 System Modeling

In this section, we present a strategy for tightly coupling the GNSS raw measurements (pseudoranges and Dopplers) with the lane marking measurements. When calculating the pseudorange and the Doppler, the position of the GNSS antenna with respect to the body frame has to be taken into account:

$$\begin{cases} x_a = x + \cos\psi \cdot t_x - \sin\psi \cdot t_y \\ y_a = y + \sin\psi \cdot t_x + \cos\psi \cdot t_y \\ z_a = z + t_z \end{cases} \tag{5.3}$$

where $[x, y, z]^T$ is the 3D position of the vehicle, ψ is its heading, $[t_x, t_y, t_z]^T$ is the position of the receiver antenna in R_M and $\mathbf{x}_a = [x_a, y_a, z_a]^T$ is the position of the receiver antenna in R_O. For a given satellite i, its position vector $\mathbf{x}_s^i = [x^i, y^i, z^i]^T$ at the emission time and in the working frame is reconstructed from the received navigation message. The corresponding pseudorange is:

$$\rho^i = \sqrt{(x_a - x_s^i)^2 + (y_a - z_s^i)^2 + (z_a - z_s^i)^2} + c \cdot dt_u + \varepsilon^i + \beta^i \tag{5.4}$$

where ε represents the sum of the residual (non-white) errors of the pseudorange. β^i is the measurement noise.

In land vehicle applications, if the altitude changes slowly, it is often considered constant and sometimes assumed to be known by using altitude maps [100]. In this thesis, we look for a planar model of the system in order to have an efficient state observer in terms of computation with a reduced number of state components. z_a is considered as known from the Up coordinate of the lane segment (*link*) stored in the map:

$$z_a = h_a + z_{lane} \tag{5.5}$$

where h_a is the height of the GNSS antenna with respect the road, which is a constant. z_{lane} is the Up coordinate of the lane in which the vehicle is located. z_{lane} is inferred using the estimated 2D position. Figure 5.3 shows the Up coordinates mapped for each node or shape point of lane polylines.

The main sources of range errors are (cf. Figure 5.4):

- Atmospheric pseudorange residual errors ε_{pr}
- Inaccurate satellite position estimates due to the use of real-time navigation messages ε_{Sat}
- Error of the mapped altitude of the road ε_z

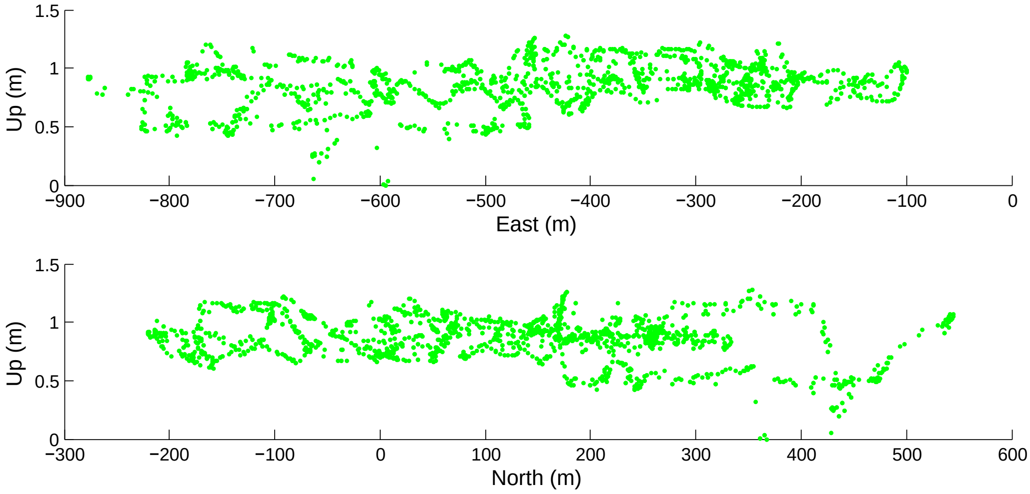


Figure 5.3: *The Up coordinates of nodes and shape points of the lane centerlines defined in the lane marking map.*

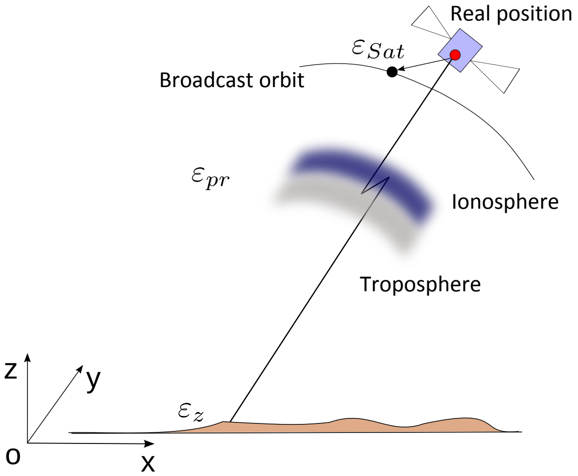


Figure 5.4: *pseudorange error sources*

We propose to estimate the range-error parameter ε^i which results from the combination of ε_{Sat}, ε_{pr} and ε_z. A first order auto-regressive shaping filter is used for satellite i as :

$$\dot{\varepsilon}^i = -\varepsilon^i/\tau_\varepsilon + w^\varepsilon \tag{5.6}$$

where w^ε is the driving noise, τ_ε is the correlation time.

The Doppler shift is caused by the relative motion of the satellite with respect to the receiver antenna. By defining the line-of-sight vector u^i_{los} of satellite i as

$$u^i_{los} = \left(\mathbf{x}_a - \mathbf{x}^i_s\right)/R^i \tag{5.7}$$

the Doppler shift is linked to the GNSS antenna through the following equation (cf. equation (2.24)):

$$\dot{\rho}^i = \left(\mathbf{v}_r - \mathbf{v}^i_s\right) \bullet u^i_{los} + c\delta\dot{t}_u - \varepsilon^i/\tau_\varepsilon + \beta^i_d \tag{5.8}$$

where $\mathbf{v}_r = [v^m \cdot \cos\psi, v^m \cdot \sin\psi, 0]^T$ is the velocity vector of the receiver and $\mathbf{v}^i_s = \left[\dot{x}^i, \dot{y}^i, \dot{z}^i\right]^T$ is the one of satellite i. $\bullet$ denotes the dot product. β^i_d is the measurement noise. $\delta\dot{t}_u$ is the drift of the clock of the receiver. The impact of the vehicle angular speed onto the Doppler effect, which is through the antenna lever arm, is neglected.

For efficient numerical computation, the filter estimates range equivalent values of the receiver clock parameters:

$$d = c \cdot dt_u \quad \dot{d} = c\cdot\dot{dt}_u$$

With n satellites in view, the full state space in the local ENU frame becomes:

$$\mathbf{x} = \left[x, y, \psi, \varepsilon_\omega, d, \dot{d}, \varepsilon^1, \ \cdots \ , \varepsilon^n\right]^T \tag{5.9}$$

With a 2D unicycle kinematic model, a discrete evolution model is given by:

$$\mathbf{x}_k = f\left(\mathbf{x}_{k-1}, \mathbf{u}_k^m\right) \Leftrightarrow \begin{cases} x_k = x_{k-1} + T_e \cdot v_k^m \cdot \cos\psi_{k-1} \\ y_k = y_{k-1} + T_e \cdot v_k^m \cdot \sin\psi_{k-1} \\ \psi_k = \psi_{k-1} + T_e \cdot (w_k^m - \varepsilon_{\omega,k-1}) \\ \varepsilon_{\omega,k} = \varepsilon_{\omega,k-1} + \alpha_k^{\omega} \\ d_k = d_{k-1} + T_e \cdot \dot{d}_{k-1} + \alpha_k^d \\ \dot{d}_k = \dot{d}_{k-1} + \alpha_k^{\dot{d}} \\ \varepsilon_k^1 = a \cdot \varepsilon_{k-1}^1 + \alpha_k^{\varepsilon 1} \\ \vdots \\ \varepsilon_k^n = a \cdot \varepsilon_{k-1}^n + \alpha_k^{\varepsilon n} \end{cases} \tag{5.10}$$

where $\mathbf{u}^m = [v^m, \omega^m]^T$ denotes the measured input vector and T_e the sampling period.

5.3.3 Localization solver for tightly coupling GNSS measurements

5.3.3.1 GNSS measurement validation

At time instant k, when the GNSS measurements are available, a validation step is performed on the measurements of every satellite to avoid the use of badly tracked satellites and to reject multipath signals.

The antenna is less sensitive to signals arriving at low elevation angles. Low elevation angle signals can be affected by undesired reflections (multipath) and their effect (as they arrive delayed, by the detour, as compared to the direct path signal) on the receiver tracking loops and eventual pseudorange measurements should be minimized. Let us describe now a robustification strategy to handle GNSS measurement outliers.

For a Doppler measurement, the validation process is based on the following strategy:

- C/N_0 gating: check that C/N_0 is high enough (e.g. 38 dB-Hz).
- Elevation mask: the elevation angle of satellite i is calculated using $\mathbf{x}_s^i$ and the current estimate $\mathbf{x}$. The elevation mask angle is usually set to 15 degrees.
- Innovation gating: check the Normalized Innovation Squared [43].

As a Doppler is in general more accurate than a pseudorange, a pseudorange is used only when its corresponding Doppler measurement is valid. For a pseudorange measurement, an innovation gating step is also added after having used the Doppler.

5.3.3.2 Filter implementation

By denoting $\mathbf{A}_k = \frac{\partial f\left(\mathbf{x}_{k-1}, \mathbf{u}_k^m\right)}{\partial \mathbf{x}}$, $\mathbf{B}_k = \frac{\partial f\left(\mathbf{x}_{k-1}, \mathbf{u}_k^m\right)}{\partial \mathbf{u}^m}$, the tightly coupling filter is described by Algorithm 5.1 in which the Dopplers are used at first. Additionally a Mahalanobis distance threshold is set to reject the mismatch before using C_0 to update the state vector. The process is time-triggered with the CAN bus data which has the highest rate. The latency of the GNSS receiver is neglected, in the current implementation, since the filter runs in post-processing on recorded data with PPS time stamping. The camera latency is also neglected since it provides essentially lateral corrections: neglecting its latency has almost no impact on the accuracy.

Algorithm 5.1 An iteration stage of the filter

```
In out: $\mathbf{x}$, $\mathbf{P}$
$\mathbf{u}^m = [v^m, \omega^m]^T$ =Get (DR measurements)
$\mathbf{x} = f(\mathbf{x}, \mathbf{u}^m)$
$\mathbf{P} = \mathbf{A}_k \cdot \mathbf{P} \cdot \mathbf{A}_k^T + \mathbf{B}_k \cdot \mathbf{N} \cdot \mathbf{B}_k^T + \mathbf{Q}$
if New GNSS data is available then
    $[\rho^{1,\dots,n}, \dot{\rho}^{1,\cdots n}]$ =Get (GNSS measurements)
    Good_Doppler=$\varnothing$   Good_Pr=$\varnothing$
    for $j = 1, \dots, n$ do// n is the number of satellites in view
        if ($\dot{\rho}^j$ is valid) then // Please refer to Section 5.3.3
            Add($\dot{\rho}^j$) to the Good_Doppler list
        end if
    end for
    $[\mathbf{x}, \mathbf{P}]$ =Update ($\mathbf{x}$,$\mathbf{P}$,Good_Doppler)
    for $j = 1, \dots, n$ do
        if ($\dot{\rho}^j$ is valid ) and (innovation < Threshold) then
            Add($\rho^j$) to the Good_Pr list
        end if
    end for
    $[\mathbf{x}, \mathbf{P}]$ = Update ($\mathbf{x}$,$\mathbf{P}$,Good_Pr)
end if
if New camera measurements are available then
    $[C_0]$ =Get ( camera measurements )
    $[AB]$=map_match ($\mathbf{x}, C_0, map$)
    if ($[AB]$ is consistent with the vehicle state) then
        $[\mathbf{x}, \mathbf{P}]$ = Update ($\mathbf{x}, \mathbf{P}, C_0, [AB]$)
    end if
end if
```

It is important to note that the filter uses a measured linear velocity v^m in the Doppler observation model (see equation (5.8)). Therefore, the estimation process

Model noises variances	Measurement noises variances
$Var\left(\alpha^{\omega}\right) = 5 \times 10^{-10}$	$Var\left(\beta_c\right) \sim N\left(0, 0.16\right)$
$Var\left(\alpha^{d}\right) = 1 \times 10^{-3}$	$Var\left(\beta\right) = S \cdot 10^{\frac{-C/N_0^i}{10}}$
$Var\left(\alpha^{\dot{d}}\right) = 1 \times 10^{-4}$	$Var\left(\beta_d\right) = 0.05$
$Var\left(\alpha^{\varepsilon}\right) = 1 \times 10^{-4}$	$Var\left(\gamma^{v}\right) = 1 \times 10^{-4}$
	$Var\left(\gamma^{\omega}\right) \sim N\left(0, 2.5 \times 10^{-3}\right)$

Table 5.1: *Tuning of the EKF-CN parameters (International System Units)*

has correlated noises and the EKF needs to be reformulated as an EKF with correlated noises (EKF-CN) [99]. When using Doppler measurements to update the state vector, the Kalman gain $\mathbf{K}$ is calculated as:

$$\mathbf{K} = \left(\mathbf{P} \cdot \mathbf{H}^T + \mathbf{S}\right) / \left(\mathbf{H} \cdot \mathbf{P} \cdot \mathbf{H}^T + \mathbf{D} \cdot \mathbf{N} \cdot \mathbf{D}^T + \mathbf{R} + \mathbf{H} \cdot \mathbf{S} + \mathbf{S}^T \cdot \mathbf{H}^T\right) \tag{5.11}$$

with $\mathbf{S} = \mathbf{B}_v \cdot \mathbf{N} \cdot \mathbf{D}^T$, $\mathbf{N}_v = Var\left(\gamma^v\right)$, $\mathbf{D} = \frac{\partial \dot{\rho}^i}{\partial v^m}$, $\mathbf{H} = \frac{\partial \dot{\rho}^i}{\partial \mathbf{x}}$ and $\mathbf{B}_v = \frac{\partial f\left(\mathbf{x}_{k-1}, \mathbf{u}_k^m\right)}{\partial v^m}$. The update of the covariance is

$$\mathbf{P} = \mathbf{P} - \mathbf{K}\left(\mathbf{H} \cdot \mathbf{P} + \mathbf{S}^T\right) \tag{5.12}$$

The prediction and update by camera and pseudorange follow the EKF with measured input presented before (Algorithm 3.1, page 82).

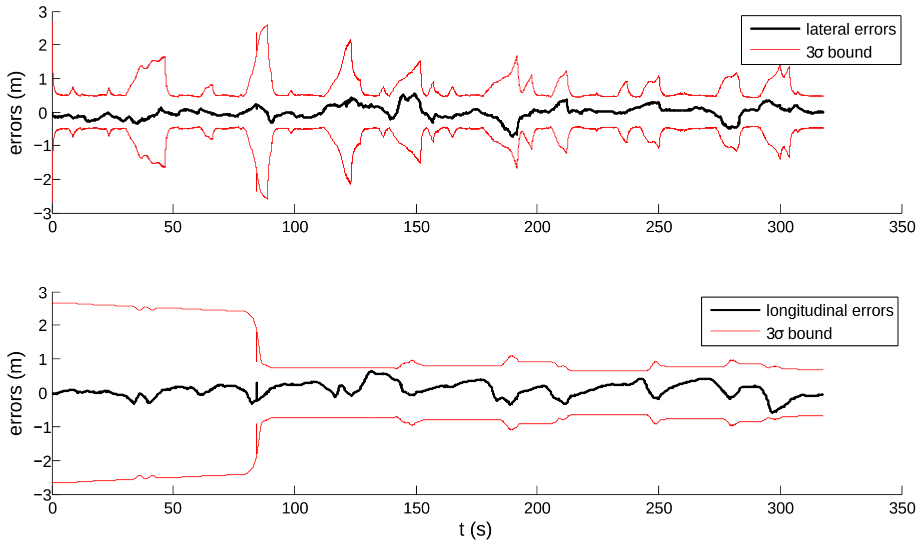
5.4 Experimental Results

The measurement noise β^i in equation 5.4 is supposed to be white and zero mean ($\beta^i \sim N\left(0, \sigma_i^2\right)$). As it is not stationary as indicated in [5], its variance is modeled by Wieser's model in which the measured carrier-to-noise density ratio C/N_0 is used as a quality indicator:

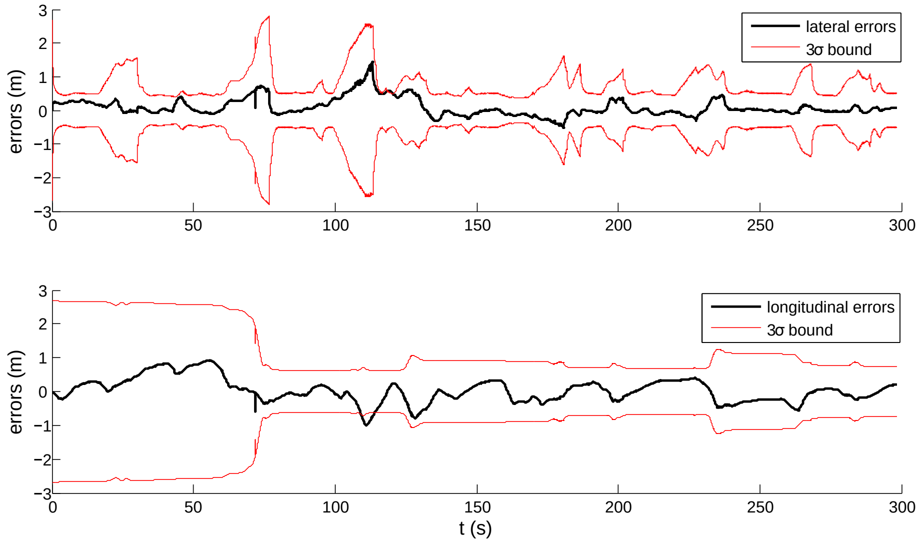
$$\sigma_i^2 = S \cdot 10^{\frac{-C/N_0^i}{10}} \tag{5.13}$$

where $S = 60000 m^2 Hz$.

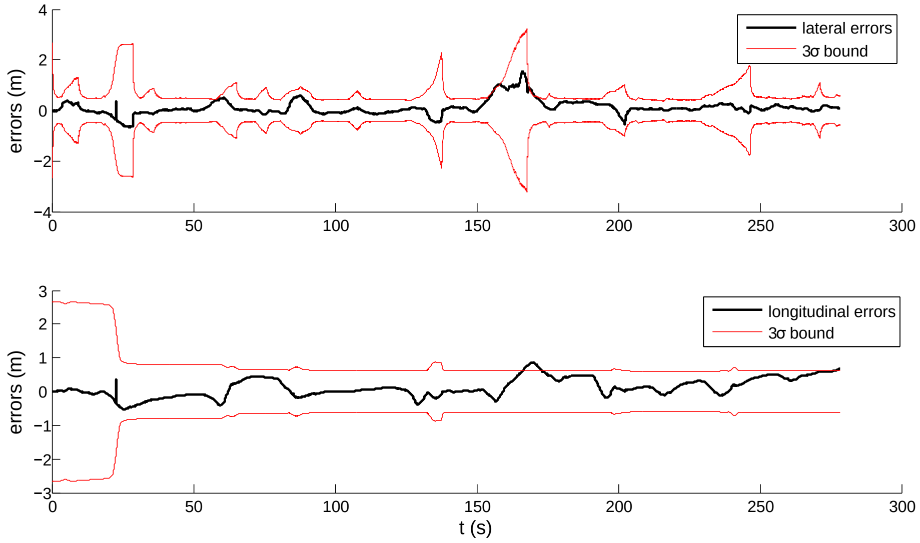
Table 5.1 specifies all the process and measurement noises.



(a) *Test 1*

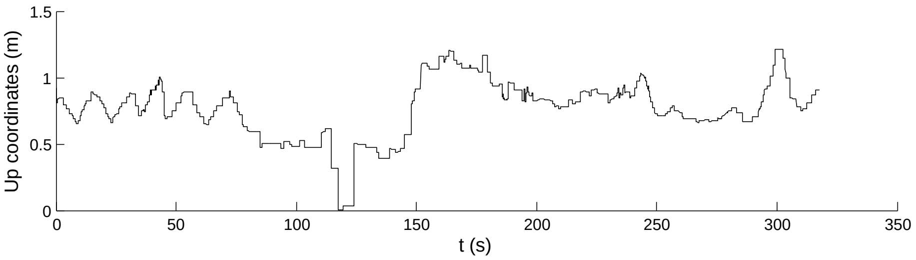


(b) *Test 2*

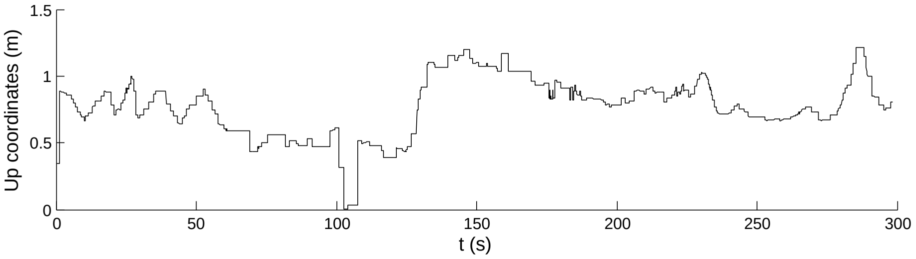


(c) *Test 3*

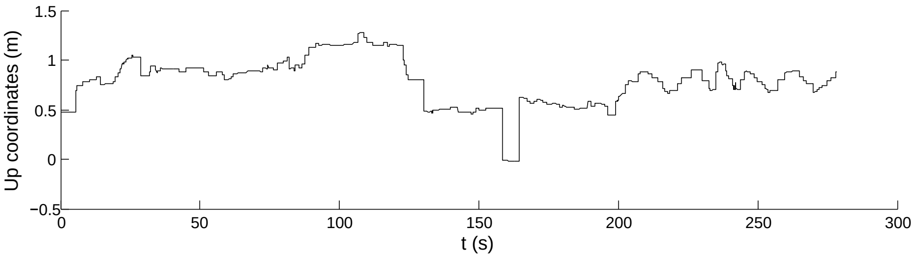
Figure 5.5: *Longitudinal/lateral positioning errors by tightly coupling*



(a) *Test 1*



(b) *Test 2*



(c) *Test 3*

Figure 5.6: *Up coordinates of the experimental vehicle obtained from the map*

Positioning errors are first studied. Figure 5.5 shows the lateral and longitudinal positioning errors with estimated 3σ bounds.

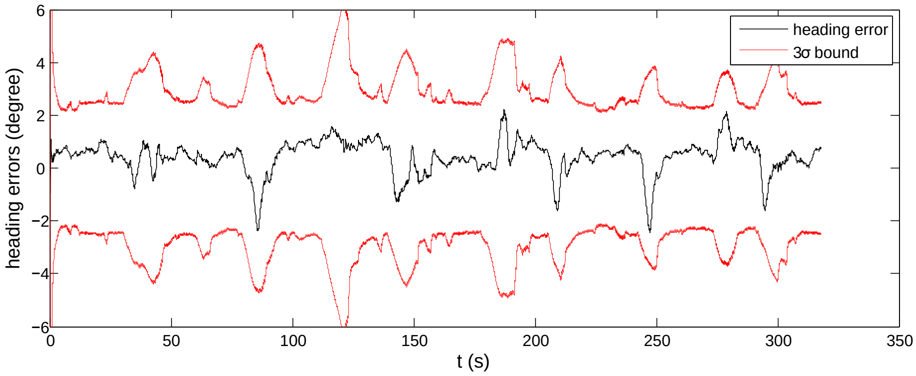
Figure 5.6 shows the altitudes of the GNSS antenna computed using the map.

Figure 5.7 displays the variation of heading errors. The biggest heading error is less than 3 degrees.

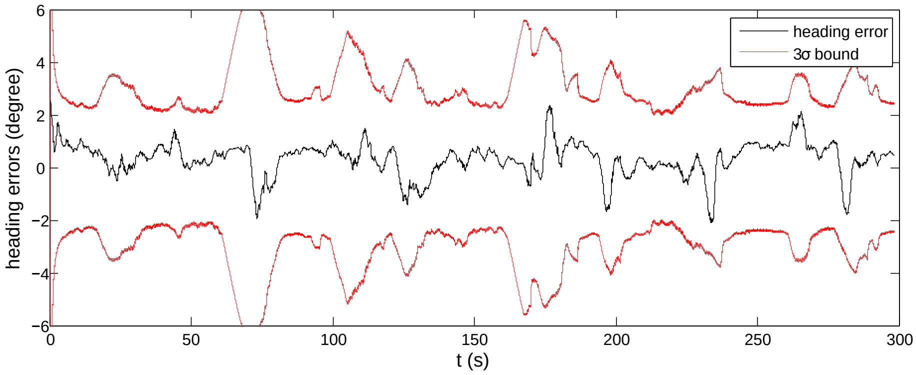
Figure 5.8 displays the estimated receiver clock bias and the corresponding 3σ bounds of the estimation errors.

Figure 5.9 gives the estimated receiver clock bias drift and the 3σ bounds of the estimation errors. Both clock bias and clock bias drift have converging uncertainties.

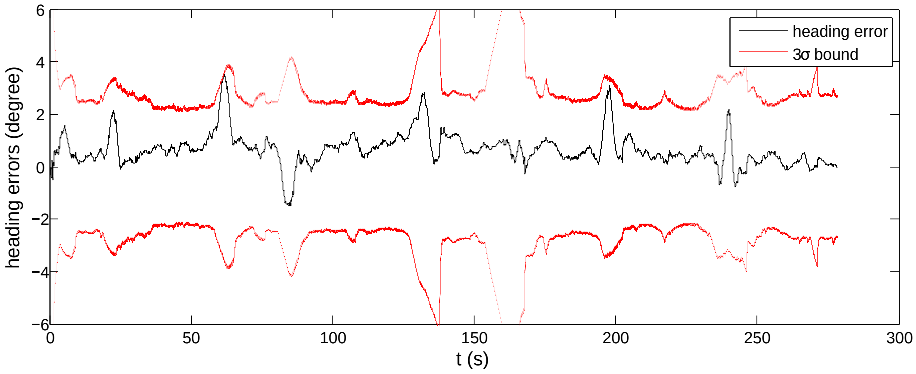
The advantage of tightly coupling GNSS is that the localization solver utilizes only the raw measurements with high quality. As described in algorithm 5.1, the satellites



(a) *Test 1*

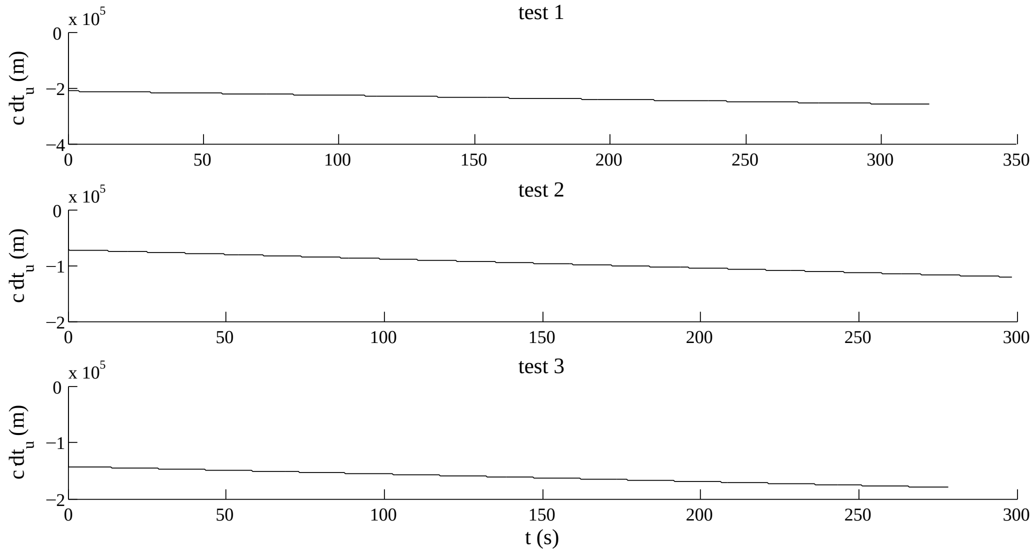


(b) *Test 2*

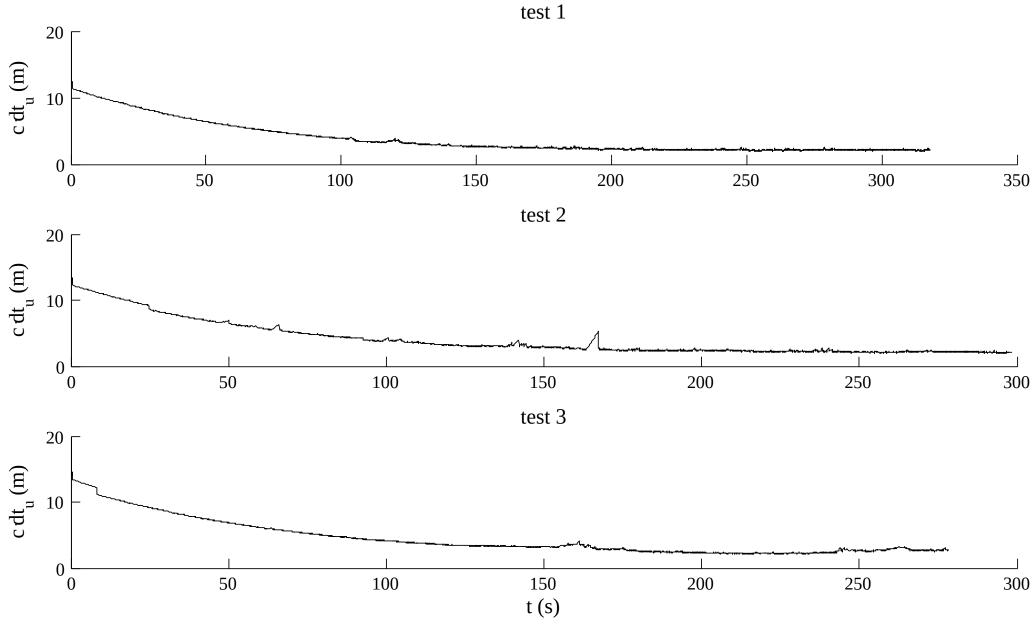


(c) *Test 3*

Figure 5.7: *Heading errors*

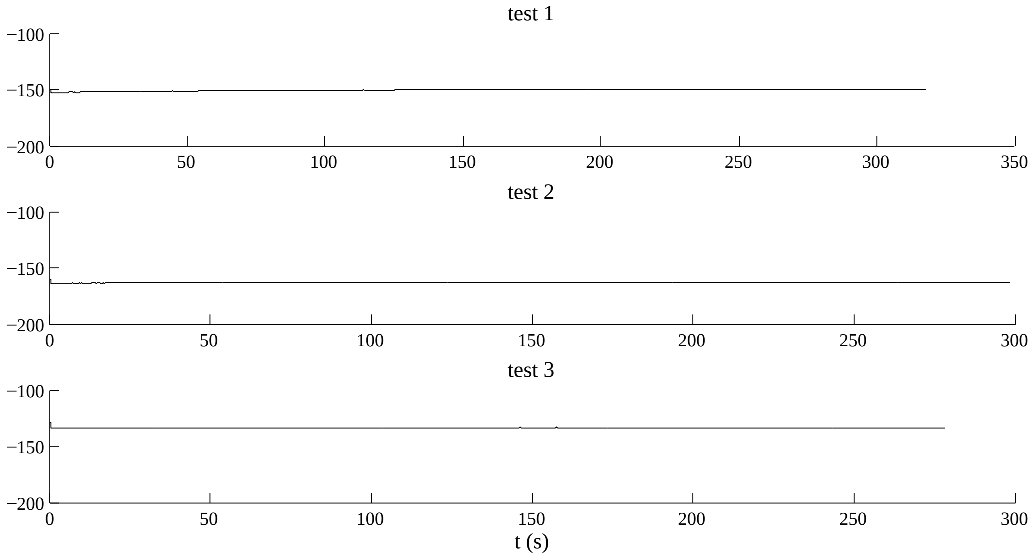


(a) *Estimated receiver clock bias times the speed of light (meters)*

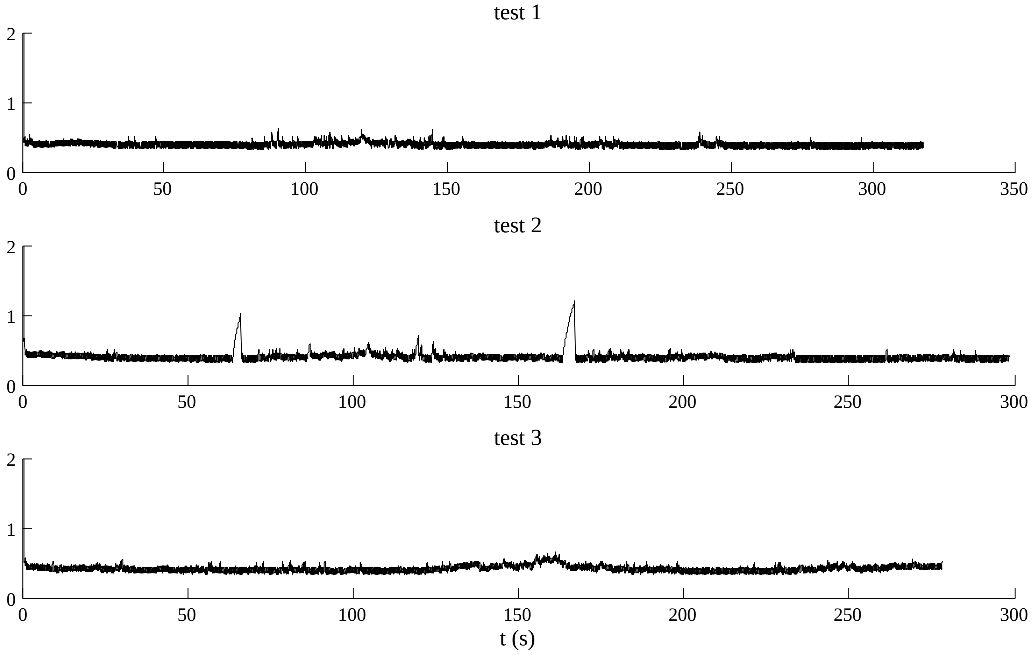


(b) *Estimated* 3σ *bound (meters)*

Figure 5.8: *Receiver clock bias estimations*

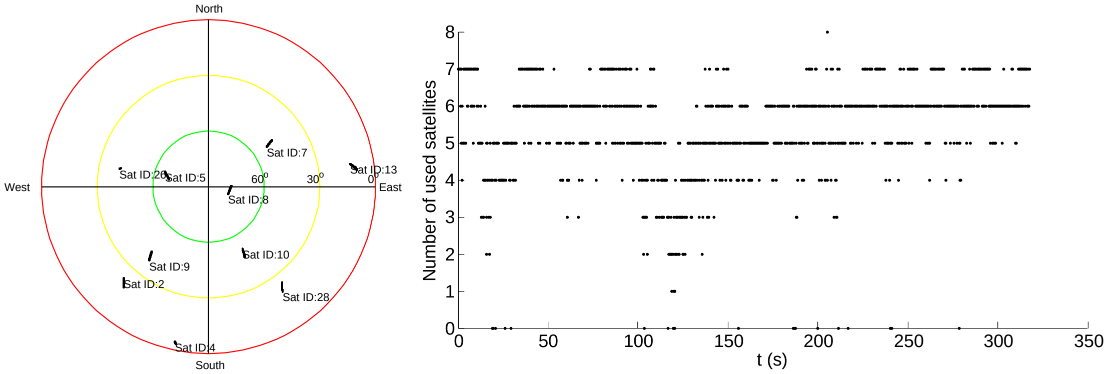


(a) *Estimated receiver clock bias drift times the speed of light (meters per second)*



(b) *Estimated 3σ bound (meters per second)*

Figure 5.9: *Receiver clock bias drift estimations*



(a) *Azimuth-Elevation plot of GNSS satellites during Test 2*

(b) *The number of used satellites*

Figure 5.10: *Satellite in view and the number of used satellites for test 1*

in view are selected according to elevation and C/N_0 and validated by innovation gating one by one. Figure 5.10 shows the satellite in view and the number of used satellites. There are totally 10 satellites in view, but the highest number of satellites used at the same time is 8. This is because measurements of satellite 4 and 13 are always rejected, due to their elevation angle under 15 degrees. One can also notice that the number of used satellite reduces rapidly when the vehicle enters the urban area (t = 112 -123s).

For comparison purpose, we have performed a tightly coupling EKF without using the camera measurement. Figure 5.11 presents the cumulative distribution functions of the horizontal positioning errors. The tightly coupling EKF with camera gives the most accurate positioning. One can notice that the positioning performance is highly improved compared with standalone GNSS, even without using the camera measurement. This verifies the autoregressive modeling of pseudorange bias proposed in [13]. Hereafter, we look at the estimated pseudorange biases.

For each subplot of Figure 5.12, the abscissa expresses time in seconds, the ordinate gives every estimated bias ε^i in meters. The bias on every pseudorange is initialized at zero. Measurements of satellites 4 and 13 are not used. Moreover, the measurements of satellite 26 became available only after 100s and are lost after 200s. The estimated biases are quite smooth, in the order of few meters and stay bounded during all the trial.

A limit of the presented results is that it lacks a comparison with a ground truth value of pseudorange biases. In order to remedy this issue, we suggest to investigate firstly the convergence of the estimated error covariance and the consistency of the position estimation later. Figure 3.16 displays the estimated 3σ bounds for each satellite in view. The error bounds converge toward constants which indicates an observable behavior.

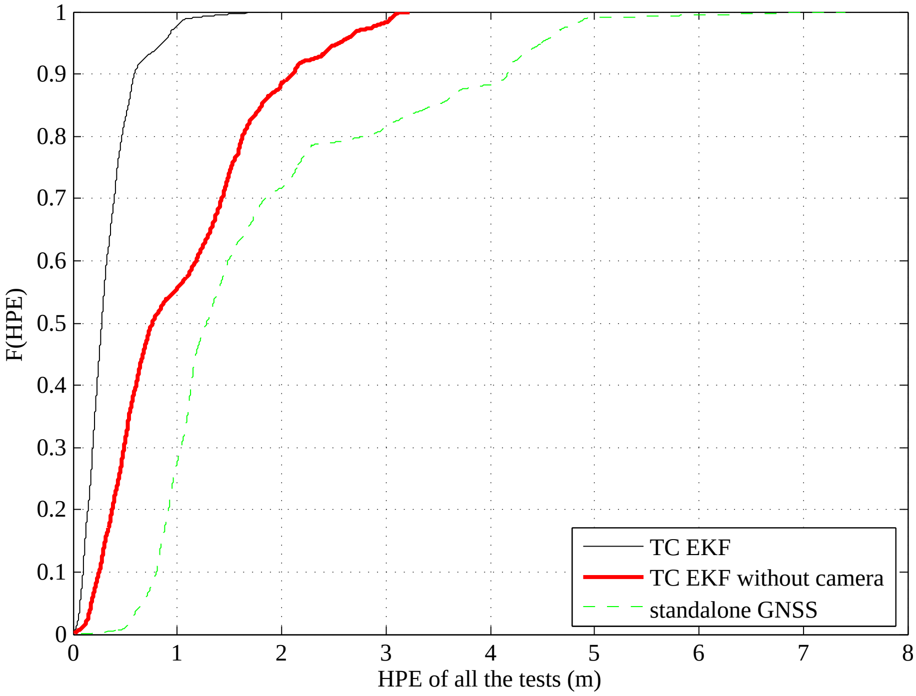


Figure 5.11: *Plots of cumulative distribution functions (CDF) of the HPEs*

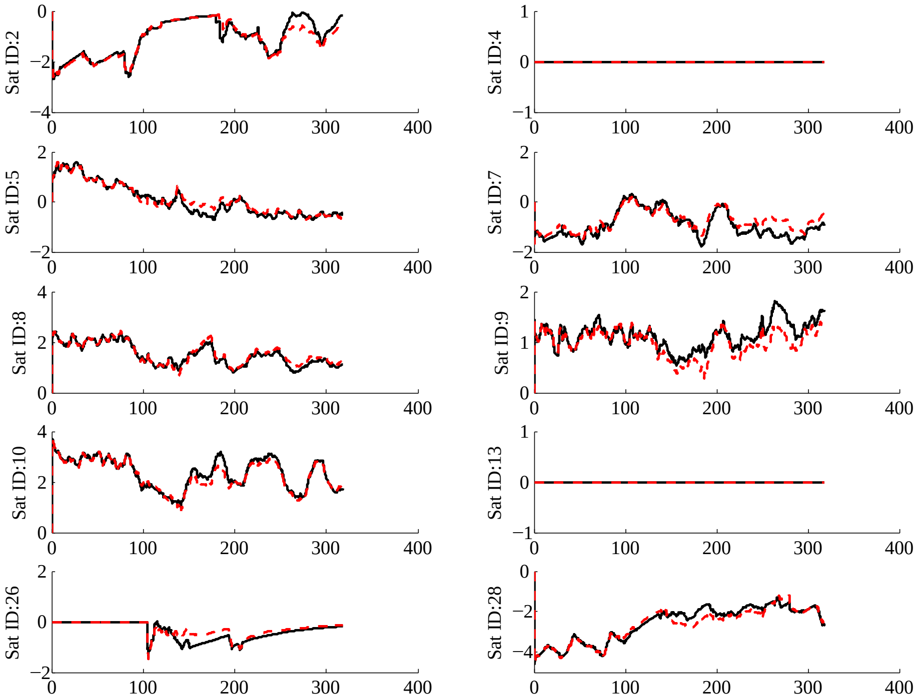


Figure 5.12: *Estimated biases on pseudoranges. The red line for TC EKF without camera. The black line for TC EKF.*

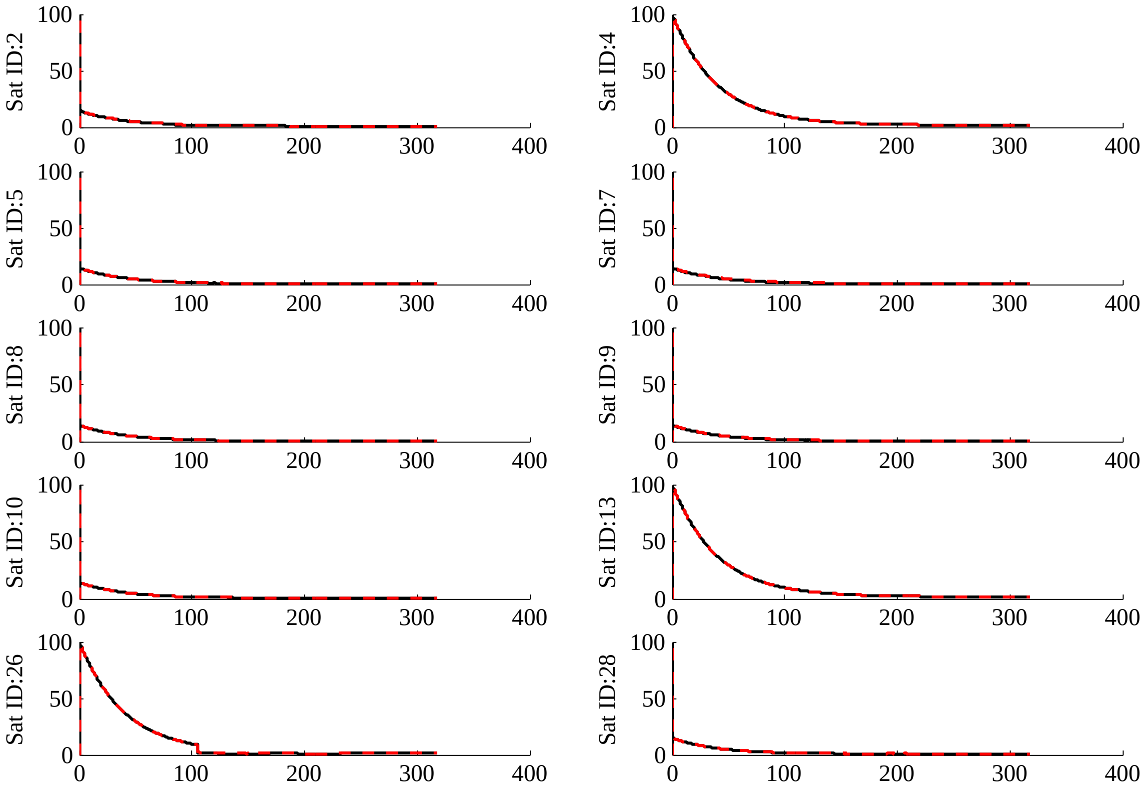


Figure 5.13: *The estimated 3σ bounds of the error of estimated pseudorange bias*

We can conclude that modeling and estimating a time correlated bias on pseudorange of each satellite is feasible even without the camera. The integration of lane marking measurements enhance their estimation as shown in Figures 5.11 and 5.12.

5.5 Conclusion

In this chapter, we have studied a method which merges raw GNSS measurements (pseudoranges and Dopplers) with lane marking measurements and DR sensors. In order to reduce the dimension of the state vector and to have a fast response to the velocity change, the linear and angular velocity are used as measured input. The measured linear velocity appears in the Doppler observation model, so the EKF filter is reformulated as an EKF with correlated noise for Doppler. The vehicle is assumed to travel on a flat surface thanks to the use of a navigation frame close to the navigation area. The up dimension of the vehicle is considered as known by the map. A first order autoregressive model is applied to estimate pseudorange bias of each satellite. The state vector is significantly augmented compared to a loosely coupled approach but stays observable thanks to the time-correlated model. Outdoor experiments data has been used to validate the proposed method.

Chapter 6

Discussion and integrity considerations

Contents

6.1 Introduction

In the last three chapters, three variants of the localization solver have been studied. This chapter is dedicated to a detailed comparison of the performance of the different filters according to several specific metrics. The good choice of performance metrics is usually critical for assessing a system and evaluating its application. For vehicle localization systems, accuracy is an important metric which usually refers to statistical figures of merit of the position or velocity error. In this thesis, we adopt the definition of accuracy as defined in the SaPPART (Satellite positioning performance assessment for road transport) project [4]. The metrics are built from statistical distributions of the errors with respect to a ground truth. For instance, the median value and the 95^{th} percentile of the horizontal positioning error distribution can be chosen for accessing the horizontal positioning accuracy. Since the performance of GNSS-based systems has a strong dependence on time and operational conditions, the performance such as accuracy cannot be determined by a single measurement.

The sensor fusion being made by probabilistic estimators in this thesis, we propose to evaluate the consistency of the filters by checking whether the real covariance of the estimation error is well estimated.

Another metric is availability which refers to the percentage of time during which a pose estimate is delivered with the required performance (e.g., required accuracy and consistency). For a given localization system, the higher the performance requirement, the lower its availability. Since confidence bounds act often as decision variables for qualifying availability, a localization system has to be not too pessimistic (otherwise the provided localization information will be declared unavailable in situations where it is, in fact, good enough for the current navigation task). So, a good localization system is a system that provides adequate confidence information.

Finally, the localization performance of the proposed approach is strongly influenced by map matching and data association with the lane marking map. As we have seen in Chapter 3, the vehicle pose estimate can be matched with an incorrect lane when ambiguity between lanes appears. This can be seen as a loss of integrity since the information provided to the client applications is misleading. Therefore, the system should be designed to handle such ambiguous situations and able to inform the client applications of this issue, in a way similar to RTK GPS when the phase ambiguity resolution is not fixed (this is called floating RTK).

In the following sections, we first compare the accuracy and consistency of the methods that have been studied in the last three chapters. Then, we discuss a multi-hypothesis solution to enhance the capability of handling data association ambiguities with the lane marking map.

6.2 Accuracy and consistency comparisons

6.2.1 Accuracy

Figure 6.1 depicts the cumulative distribution of HPEs of the aforementioned three methods: loosely coupled ENU EKF (LC ENU EKF), tightly coupled EKF (TC EKF) and loosely coupled road invariant EKF (LC RI EKF). Table 6.0a-6.0c lists accuracy metrics (standard deviation, maximum, median, and 95th percentile) for the three methods respectively.

Let us take a look at each test. For test 2 and test 3 the TC EKF and RI EKF have similar HPE distributions. For test 1, the TC EKF is much better than the other two methods. The improvement due to TC EKF mainly happens in the urban canyon area (see Figure 6.2). The loosely coupled method relies heavily on the GNSS fixes computed by the standalone GNSS receiver. When the GNSS receiver suffers from large errors during a long time, the loosely coupled solvers outputs are closer to the biased GNSS fixes. Inversely, the TC EKF uses raw satellite measurements rather than filtered GNSS solutions. Errors are modeled at the pseudorange level and measurements of each satellite are validated by an innovation gating against the current pose estimate. Intuitively, the TC EKF can be more powerful on excluding faulty satellite measurements as it uses information from the other sensors and from

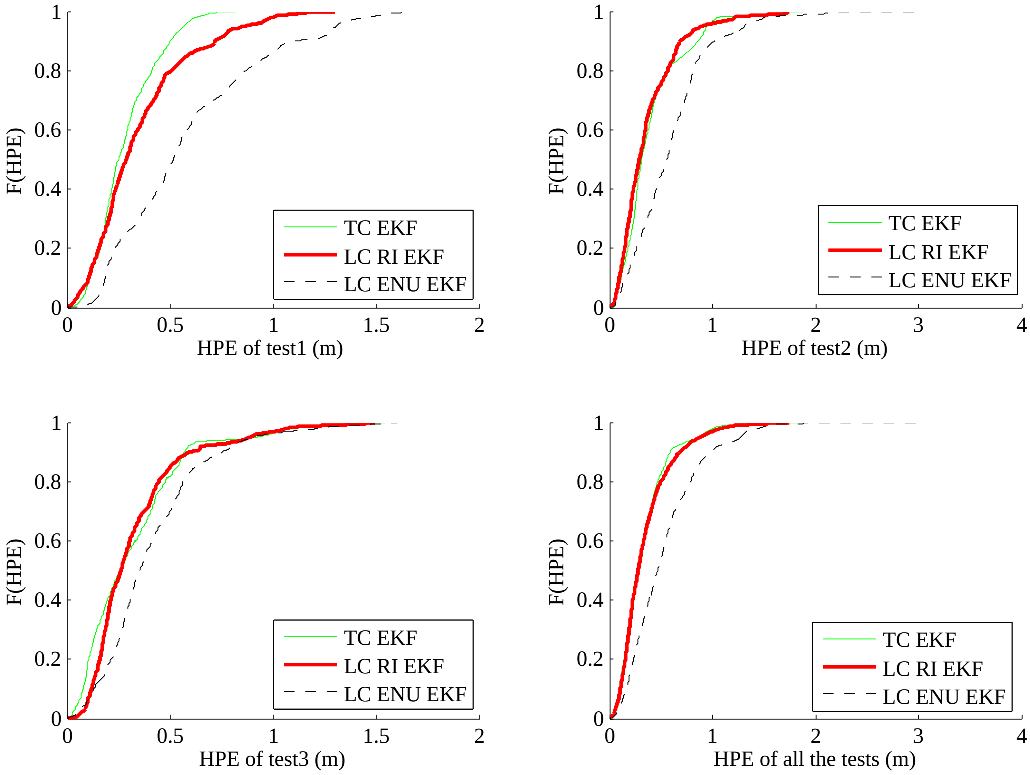


Figure 6.1: *Plots of the cumulative distribution function (CDF) of the HPE produced by the different methods*

(a) *LC ENU EKF*

	test 1	test 2	test 3	global
std. dev. (m^2)	0.35	0.38	0.26	0.35
max (m)	1.64	3.04	1.61	3.04
median (m)	0.52	0.56	0.35	0.47
95th percentile (m)	1.31	1.31	0.91	1.28

(b) *LC RI EKF*

	test 1	test 2	test 3	global
std. dev. (m^2)	0.24	0.29	0.24	0.26
max (m)	1.30	1.74	1.49	1.74
median (m)	0.29	0.29	0.26	0.29
95th percentile (m)	0.86	0.90	0.89	0.89

(c) *TC EKF*

	test 1	test 2	test 3	global
std. dev. (m^2)	0.15	0.29	0.27	0.24
max (m)	0.82	1.87	1.61	1.87
median (m)	0.25	0.31	0.35	0.28
95th percentile (m)	0.57	0.95	0.91	0.87

Table 6.1: *HPE statistics of the three methods*

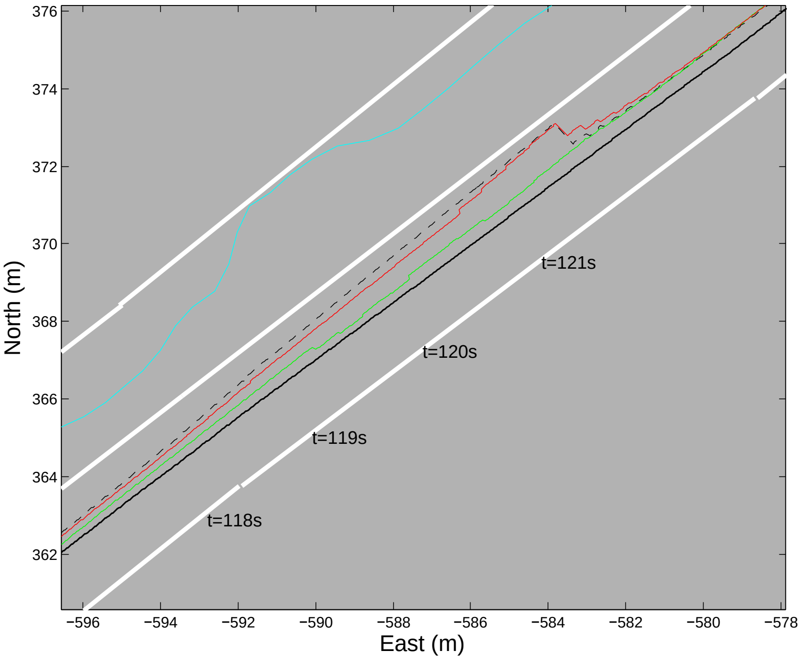


Figure 6.2: *Zoom view of test 1 in the urban canyon. LC ENU EKF, LC RI EKF, TC EKF, standalone GPS and ground truth trajectories are respectively expressed by dashed black, red, green, cyan and solid black lines. White lines are lane markings. Clearly, the TC EKF provides the best estimate.*

the map. Result also demonstrates that the estimated trajectory (in green) keeps close to the ground truth even in the area where heavy multipath happens.

At t = 122s, the camera retrieved lane marking measurements. The three methods have then similar performance as their trajectories are nearly overlapped. It hints that the camera update dominates the localization performance when all the information sources are available.

The accuracy performance is consistent among the different tests as shown in Figure 6.1 and Table 6.1. Globally speaking, TC EKF and LC RI EKF give comparable positioning accuracy as demonstrated in right-below sub-figure of Figure 6.1.

6.2.2 Consistency

Filter consistency can be checked by constructing the normalized estimation error squared δ^2 which has to follow a Chi2 distribution under Gaussian assumption:

$$\delta^2 = (\mathbf{x} - \hat{\mathbf{x}})^T P^{-1} (\mathbf{x} - \hat{\mathbf{x}}) \tag{6.1}$$

The normalized horizontal positioning error (HPE) squared e^2 is defined by the 2D position components of the state vector:

$$e^2 = \begin{pmatrix} e_x \\ e_y \end{pmatrix}^T P_{HPE}^{-1} \begin{pmatrix} e_x \\ e_y \end{pmatrix} \tag{6.2}$$

where $P_{HPE} = \begin{pmatrix} \sigma_x^2 & \sigma_{xy}^2 \\ \sigma_{xy}^2 & \sigma_y^2 \end{pmatrix}$ and σ_x, σ_y and σ_{xy} are estimated by the filter. $e_x = \hat{x} - x_{ref}$ and $e_y = \hat{y} - y_{ref}$, with (x_{ref}, y_{ref}) the ground truth of the vehicle horizontal position and $(\hat{x}, \hat{y})$ the estimated position.

If the model assumptions and the tuning filter are correct, then equation (6.2) follows a χ^2 distribution with two degrees of freedom (denoted $\chi^2\,(2)$ in the following).

In order to study the consistency of the horizontal positioning error, one usually looks at the percentage of samples exceeding a determined threshold given by a $\chi^2\,(2)$ distribution.

$$\begin{pmatrix} e_x \\ e_y \end{pmatrix}^T P_{HPE}^{-1} \begin{pmatrix} e_x \\ e_y \end{pmatrix} > k^2 \tag{6.3}$$

where k^2 is the predefined threshold linked to the chosen risk.

It is easy to check that equation (6.3) is equivalent to the following equation:

$$\sqrt{e_x^2 + e_y^2} > k\sqrt{\frac{1}{\mathbf{u_e}^T P_{HPE}^{-1} \mathbf{u_e}}} \tag{6.4}$$

where $\mathbf{u_e} = \begin{pmatrix} e_x \\ e_y \end{pmatrix} / \sqrt{e_x^2 + e_y^2}$ is the unit vector supporting the horizontal positioning error .

Let us define σ_{HPE} as the standard deviation along the horizontal positioning error vector:

$$\sigma_{HPE} = \sqrt{\frac{1}{\mathbf{u_e}^T P_{HPE}^{-1} \mathbf{u_e}}} \tag{6.5}$$

Figure 6.3 illustrates the definition of $k\sigma_{HPE}$, where the equation of the ellipse is

$$\begin{pmatrix} x - \hat{x} \\ y - \hat{y} \end{pmatrix}^T P_{HPE}^{-1} \begin{pmatrix} x - \hat{x} \\ y - \hat{y} \end{pmatrix} = k^2$$

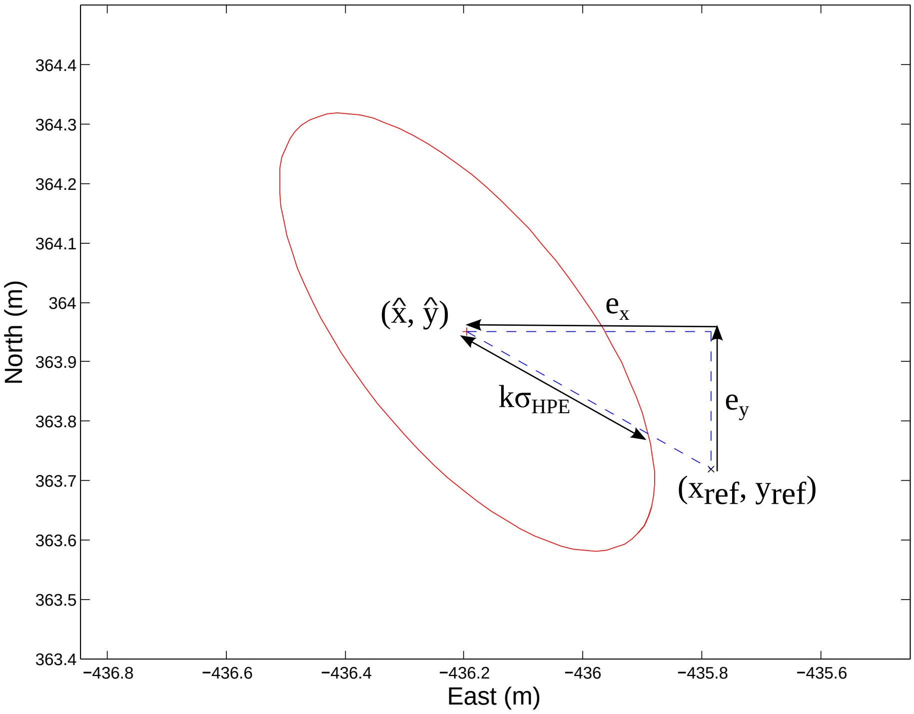


Figure 6.3: *$(\hat{x}, \hat{y})$ is the estimated position. (x_{ref}, y_{ref}) is the ground truth of the vehicle horizontal position. Here, the reference is located outside of the confidence domaine*

Before looking at the consistency of the localization solver, we study first the distribution of the estimated confidence domains done by the filters (here we consider $1\sigma_{HPE}$).

As confidence is in practice compared to a threshold to indicate “use” or “don’t use” to the client application, it is important, in terms of availability of the positioning information, to provide as small as possible confidence zones. We visually present this by plotting the CDFs of the σ_{HPE} (Figure 6.4). It demonstrates that the confidence domain produced by the loosely coupled methods is slightly tighter than the tightly coupled one.

We have chosen to set the consistency risk at 10^{-2} (1%) which is a common choice in robotics, but this value can be easily adjusted to any specific requirement. According to $\chi^2(2)$ distribution, $k^2 = 9.21$. In this case, the corresponding bound of the 2D estimated position is $3.035\sigma_{HPE}$, with σ_{HPE} being estimated in real-time by the solver. Therefore, consistency failure happens when the real error is beyond this $3.035\sigma_{HPE}$ bound, which means

$$\sqrt{e_x^2 + e_y^2} > 3.035\sigma_{HPE} \tag{6.6}$$

As discussed in [39], Bayesian state filtering is often overconfident, so we are interested in studying this issue by comparing the consistency of the loosely and tightly

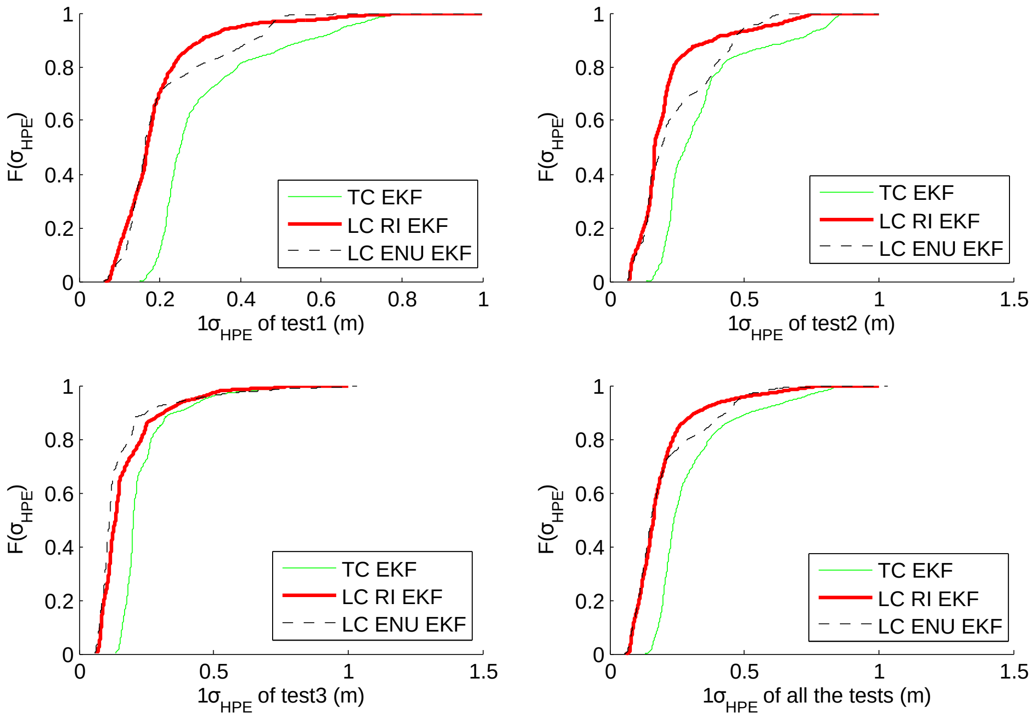


Figure 6.4: *CDFs of* $1\sigma_{HPE}$

	Consistency failure rate			
	test 1	test 2	test 3	global
LC ENU EKF	41.2%	31.8%	47.1%	39.9%
LC RI EKF	9.9%	22.9%	20.5%	17.6%
TC EKF	0	3.1%	6.1%	2.9%

Table 6.2: *Consistency failure rate of the three methods*

coupled methods.

Figure 6.5 depicts the consistency of the loosely and tightly coupled methods. The abscissa is the HPE: $\sqrt{e_x^2 + e_y^2}$. The ordinate is the $3.035\sigma_{HPE}$. Thus the points in the gray area, which satisfy $\sqrt{e_x^2 + e_y^2} > 3.035\sigma_{HPE}$, are counted to compute the percentage of samples exceeding the determined threshold 9.21.

The consistencies for each test are listed in Table 6.2. The global failure rate of the loosely coupled method is globally of 39.9% which indicates that the filter is significantly overconfident (Table 6.0a). The road invariant EKF improves the failure rate to 17.6% but remains overconfident (Table 6.0b). The tightly coupled method reduces the integrity failure rate down to 2.9% (Table 6.0c) which is the same order as the chosen risk (1%). The tightly coupled method is therefore more reliable than the two other approaches. Moreover, one can notice that the confidence domain size of the TC EKF remains lower than 2.8 meters which is important for the availability

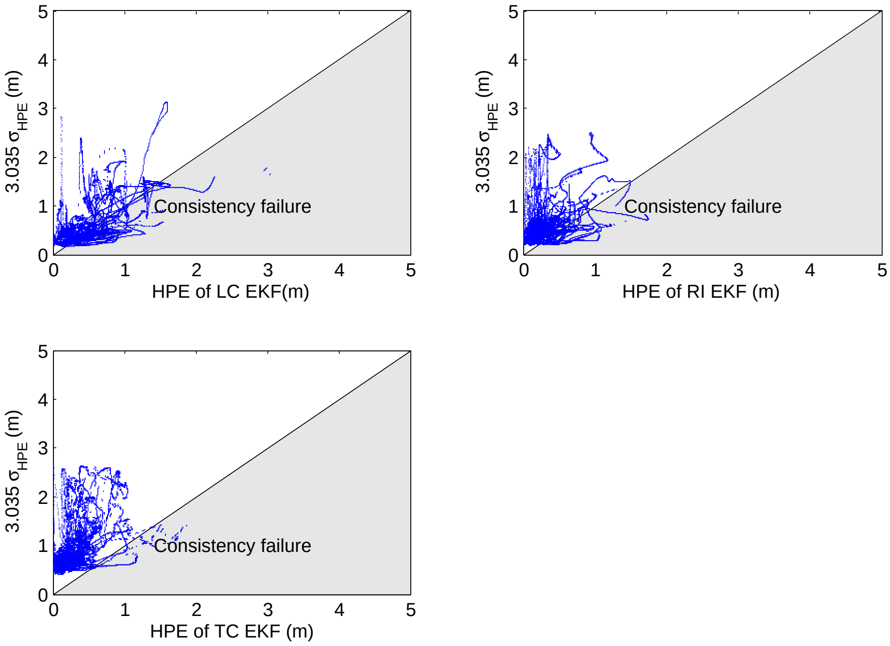


Figure 6.5: *Consistency plots for the three tests.*

of the positioning information. In other words, the TC EKF has the best consistency while keeping the σ_{HPE} quite small.

The consistency failure rate of test 3 is bigger than the other two tests. This is because the accuracy of the ground truth is degraded in the urban canyon area. Figure 6.6 demonstrates this, for t = 165-170s, the vehicle was passing through the urban canyon area along the center of the lane. The solid black line represents the trajectory of the ground truth system, with a deviation from the lane center although it was in the correct lane. Significant errors of the ground truth system with the reality has not been observed in other area. We believe that the global accuracy of the used ground truth system is in the order of tens of centimeters. The reported consistency analysis doesn't take into account of it and so the consistency failure rates are certainly lower than those indicated in this section.

6.2.3 Conclusion

To resume this consistency analysis, LC RI EKF and TC EKF have comparable confidence domains but the latter is more reliable in the sense that the ground truth is more often included in the confidence zone. Both methods perform better than the LC ENU EKF.

The analyses of the proposed algorithms rely on the three tests performed in an navigation area mainly made of straight roads. In order to demonstrate the effectiveness

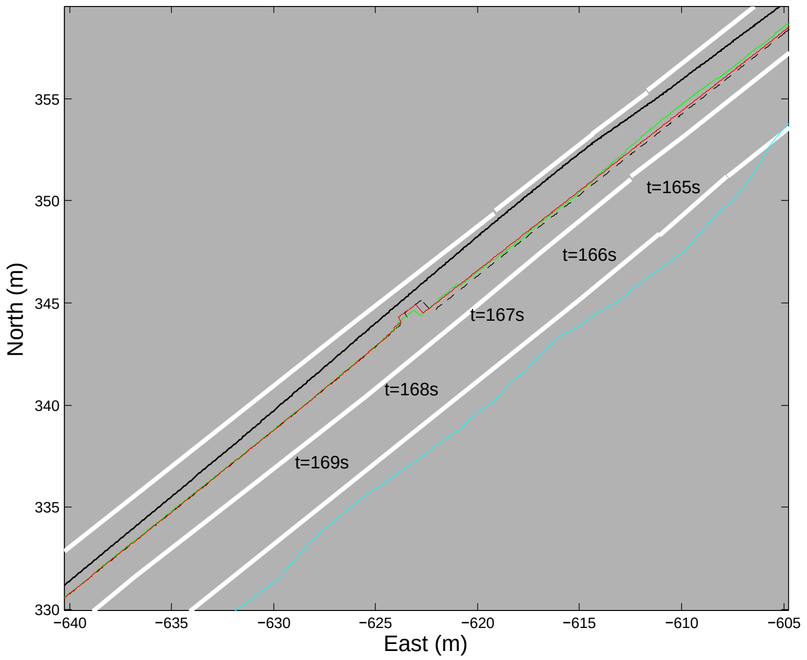


Figure 6.6: *Zoom view of test 3 in the urban canyon. White lines are lane marking. The ground truth system (in black) is 1-meter biased in this situation.*

of the proposed camera observation model and of the localization algorithms, we present other results on more curved roads in Appendix A. The lane marking map of this area has been made by mobile mapping based on our previous work presented in [118]. The localization solver implements the LC ENU EKF. Results shows that horizontal positioning performance is also highly improved compared to the standalone GNSS and sub-meter level accuracy is achieved.

Moreover, when performing the LC ENU EKF, the vehicle can encounter a situation where the camera measurement is mismatched. In this case, the GNSS bias can not be estimated correctly and the system has some difficulties in recovering from this failure. In the following section, we discuss a solution to this issue.

6.3 Handling map matching ambiguities

6.3.1 Problem statement

As we have mentioned in Chapter 3, the camera observation model relies on a data association process: a map matching must be performed to find which lane marking in the map is detected. In our approach, an off-the-shelf lane detection camera is adopted. We have seen its limitation near the road intersections where no lane marking is detected. In this case, the localization system mainly relies on fused GNSS, dead reckoning, and shaping filtered sensor errors. So data association

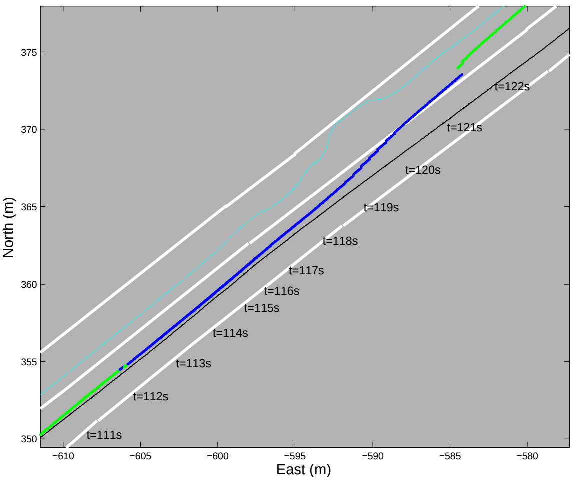
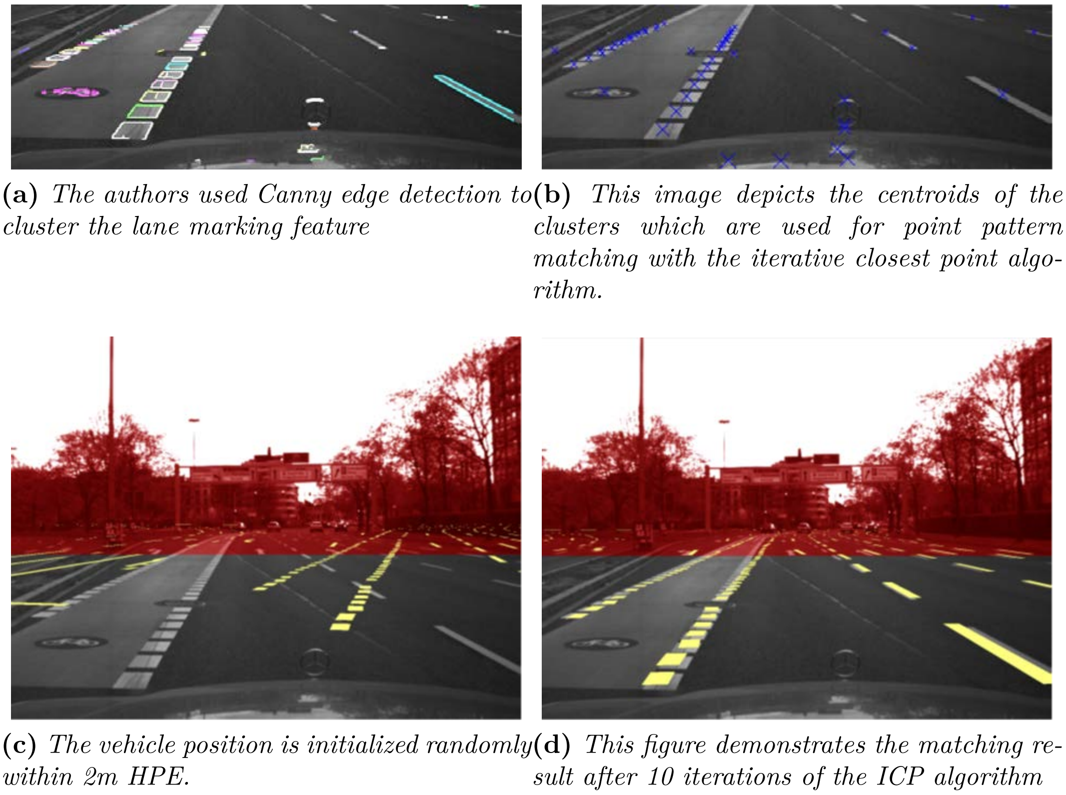


Figure 6.7: *White lines are lane markings. Black line is the true trajectory. GNSS trajectory is in cyan. Blue lines (DR coupled with GNSS) and green lines (DR coupled with GNSS, camera and map) are the estimated trajectory. Around t = 122s of test 1, the camera measurement is mismatched with the adjacent lane when the camera retrieves lane marking measurements.*

needs to be performed on the roads at the exit of crossroads. Loosely coupling the lane marking measurements with the map is then very challenging, because it is difficult to distinguish a linear model ($y = C_1x + C_0$) in the lane marking map which also contains linear segments. We have proposed to select the lane marking which minimizes the distance between the estimated lane marking point L and the considered lane marking in the map (see section 3.6.3). Mismatches can be avoided by using innovation gating. Similar strategies which search the nearest lane marking are also adopted in [75][53]. These strategies rely heavily on the current pose estimate. If the position is highly uncertain, it is hard to avoid a mismatch. With an EKF-based position tracking using lane markings, an incorrect data association can localize the vehicle in the wrong lane as indicated in Figure 6.7.

Other research approaches in the literature have focused on tightly coupling vision sensors and enhancing the feature detection algorithms such as in [101] [82][131][53]. These specially designed algorithms usually improve the availability of the feature measurements. In [53], two lateral cameras are specially equipped to detect lane markings. Detection of highly curved lane markings is greatly improved. Both in [82] and [131], a 3D lidar map augmented with ground surface reflectivities is used. The authors of [131] use a low-cost monocular camera to perform visual localization instead of the 3D lidar used in [82]. Based on the 3D map and coarse vehicle position, synthetics camera views are generated. The data association process with the camera image is then handled as an image-registration problem. The data



(a) *The authors used Canny edge detection to cluster the lane marking feature*

(b) *This image depicts the centroids of the clusters which are used for point pattern matching with the iterative closest point algorithm.*

(c) *The vehicle position is initialized randomly within 2m HPE.*

(d) *This figure demonstrates the matching result after 10 iterations of the ICP algorithm*

Figure 6.8: *This figure illustrates the data association process with a lane marking map studied in [101]. From the camera image, one can have a better perception of the vehicle's local environment. If the lane marking is well extracted from the image, there is less ambiguities between lanes which is a problem for us, since we use a lane detection camera which only outputs host lane detection. Images are from [101].*

association ambiguities are minimized in most cases. Figure 6.8 shows the data association process described in [101]. The lane marking features are firstly detected. Then the vehicle position is initialized randomly within 2m HPE to extract the lane markings in the map and transform them into the image coordinate system. The data association is turned into a point pattern matching problem which is solved by an iterative closest point algorithm.

Compared to those tightly coupled (in terms of vision sensors) approaches at the image processing level, the available information for our localization system is quite limited. The MobilEye EyeQ2 outputs only lane marking parameters of the host lane of the vehicle. Consequently, we don't have a direct observation of which lane the vehicle is currently driving in from the camera. This configuration increases the difficulty of disambiguating lane markings of different lanes when the localization confidence is low. The good aspect is that the host lane detection function is proved to be quite robust. The detection of the closest markings usually leads to less complex and more robust detection as argued in [53]. As observed in our experimental data, the host lane detection configuration prevents most false detections. Sophisticated multilane tracking algorithms can be found in [112], but it remains still challenging to perform robust multilane detection and ego-lane estimation by using a camera without a prior map.

To improve the map matching integrity with our configuration, another popular strategy is to perform multi-hypothesis (MH) tracking. Jabbour *et al.* proposed a MH approach for road tracking [62]. All the possible solutions are maintained in situations of ambiguity and removed when they become unlikely. The main advantages of a MH approach over a mono-hypothesis one has been argued as the system can inform the client applications of an ambiguity and immediately outputs a correct non-ambiguous solution without resetting the localization solver when the current mono-hypothesis solution is declared incorrect.

Inspired by the work of Jabbour, we propose thereafter a MH approach for tracking different lanes when the uncertainty is high. However, compared to tracking different roads, tracking lanes is methodologically different and generally more difficult. Here is the reason. The road tracking hypotheses are generated according to the road connectivity defined at the intersection. The weights of incorrect hypotheses decrease if the roads go in different directions with respect to the correct one when updated by the location-dependent constraint provided by the hybridized GNSS/DR. However, lanes are usually parallel and geometrically similar (e.g. the same width) with each other. Location-dependent constraints, such as geometrical distances to road features, can hardly distinguish different lanes. The vehicle may also pass from one lane to another rapidly which is not the case for road tracking. In addition, the considered task-critical application requires the localization to inform of the navigation module of an ambiguity (autonomous driving should be abandoned) and to converge to the correct solution as soon as possible. The main advantage that we have compared to the road tracking problem discussed in [62] is that the lane detection camera is able to detect the lane marking type.

6.3.2 Proposed solution

Here, we provide a preliminary attempt to gain insight into resolving the lane ambiguities when exploiting the type of the detected lane markings.

Before tracking several hypotheses, we must generate them. We suggest to generate several hypotheses when the following conditions are satisfied:

- A lane marking measurement is retrieved after a long distance without any camera measurements. The map matching is synchronized with the camera measurement because its goal is to find which lane marking is detected. The camera measurement must have a high quality.
- Likelihoods of several candidate lane markings computed using C_0 are comparable. This is a location-dependent constraint. The data structure of our

		camera			
		none	*solid*	*dashed*	*double*
map	*solid*	0.0167	0.8430	0.0902	0.0501
	dashed	0.0277	0.1275	0.8448	0
	pavement	0.0286	0.8829	0.0697	0.0188
	barrier	0.0517	0.4655	0.2759	0.2069
	other	0.0525	0.3263	0.6212	0

Table 6.3: *Camera detection of lane marking type with map reference*

map (described in Section 2.5.4) facilitates this study, because we know the adjacent lanes for a given lane and lane markings belong to them. We first compute the likelihood of the nearest lane marking, then the ones of the lane markings of the adjacent lanes. If the normalized likelihoods are in the same order of magnitude, we trigger the MH lane tracking.

For each hypothesis and under Gaussian assumption, the geometrical likelihood is computed as:

$$p\left(C_0|H^i\right) = \eta_p \cdot e^{-\frac{1}{2}\left(C_0 - g_{cam}\left(\mathbf{x}^i\right)\right)^2/\left(\mathbf{C}_{cam}\mathbf{P}^i\mathbf{C}_{cam}+\mathbf{R}_{cam}\right)} \tag{6.7}$$

with η_p a normalizing constant and $g_{cam}\left(\mathbf{x}^i\right)$ is the predicted measure.

Hypotheses are generated by duplicating the current EKF estimates. EKF estimates are $\mathbf{x}^i$, $\mathbf{P}^i$ and $\mathbf{l}^i$. $\mathbf{l}^i$ is the map matched lane marking parameters for hypothesis i. A weight ω_i showing the relative importance of the different hypotheses is initialized by equation (6.7). With conditional independence, the weights are updated as:

$$w_k^i = \eta_w \cdot w_{k-1}^i \cdot p\left(C_0, C_{type}|H^i\right) = \eta_w \cdot w_{k-1}^i \cdot p\left(C_0|H^i\right) \cdot p\left(C_{type}|H^i\right) \tag{6.8}$$

with η_w a normalizing constant. $p\left(C_{type}|H^i\right)$ is computed using Table 2.4 (cf. page 56) which gives the observed performance of the lane marking type detection on the used map, statistically. Table 6.3 presents the computed likelihoods.

For example, if the lane marking type of H^i is *dashed* (type *T1, T'1, T2, T'2, T3, T'3* or *T4*) and the camera detection is also *dashed*, the type-like likelihood $p\left(C_{type}|H^i\right)$ is computed as

$$\begin{aligned} p\left(C_{type}|H^i\right) &= likelihood\left(C_{type} = dashed|map\,reference = dashed\right) \\ &= 5243/6206 = 0.8448 \end{aligned}$$

An hypothesis is eliminated as soon as its normalized weight falls below a predefined threshold.

Hereafter, we present the results of implementing this strategy on our experimental

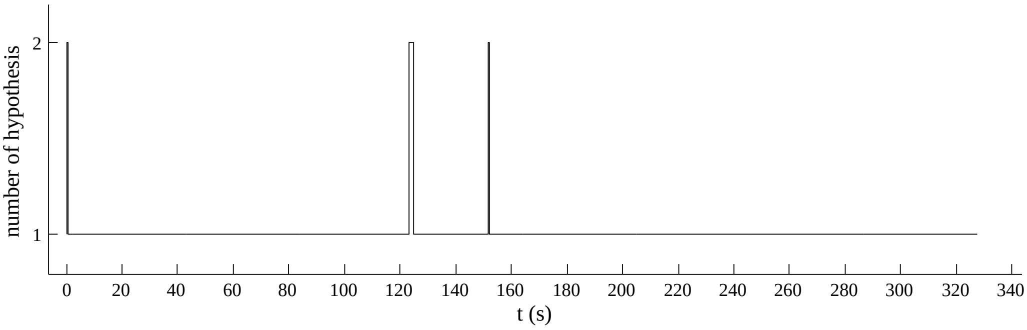


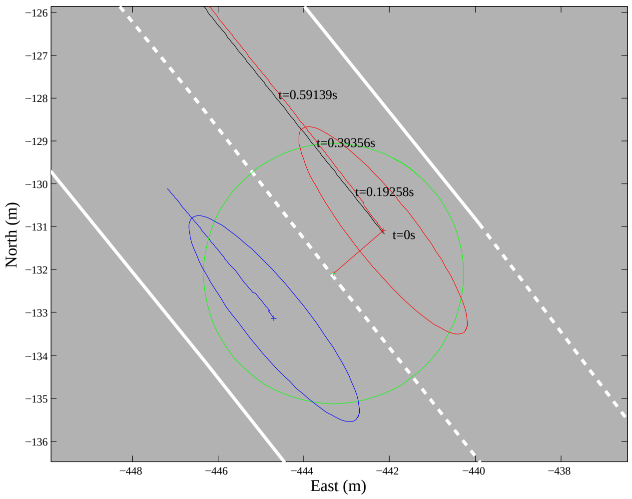
Figure 6.9: *Number of hypotheses over time*

data.

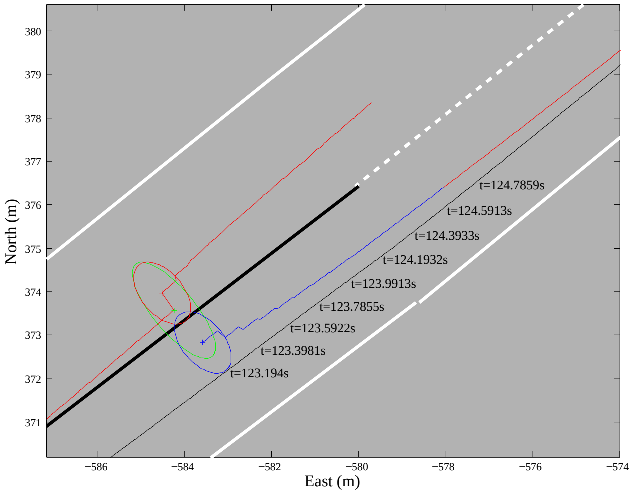
Figure 6.9 shows the variation of the number of hypotheses along time for test 1 (this is the test in which a mismatch happens). As shown in Figure 6.10a, the vehicle is initialized with a single hypothesis but with a large uncertainty (the green point and ellipse). When the camera measurements become available after a long time while they were not, the initial likelihoods computed by $p\left(C_0|H^i\right)$ are very close (Figure 6.11a) for the two lanes: 0.43 and 0.57. So, two hypotheses are generated by duplicating the pose estimates. The red ellipse and trajectory represent the hypothesis with the higher initial likelihood after the camera update, the blue one is for the hypothesis with the lower initial likelihood. Then, we use equation (6.8) to update the weight of each hypothesis. We have chosen a weight elimination threshold equals to 10^{-15} (which is very low but useful to see the evolution of the hypotheses). One can notice that the lane marking type information is a very strong constraint, since the weight changes rapidly. The correct decision is made after 0.58s of MH lane tracking.

Figure 6.10b corresponds to the area where a mismatch happened in Chapter 3 (see Figure 3.14c). The initial likelihood of the wrong lane is bigger than the one of the correct lane as displayed in Figure 6.11b. By using C_{type} to update the weights, the decision finally converges to the correct lane (Figure 6.11b). Figure 6.10c illustrates another scenario in which the uncertainty increases after the vehicle performed a corner turning (in the crossroad, the camera doesn't provide measurements). Two hypotheses are generated when the camera measurements come back. In this last situation, C_0 update is rejected for the hypothesis in blue due to the innovation gating. In this case, only $p\left(C_{type}|H^i\right)$ is available to update the weights. Figure 6.11c shows the evolution of the weights over time. The solution converges to the correct lane in less than 0.5s.

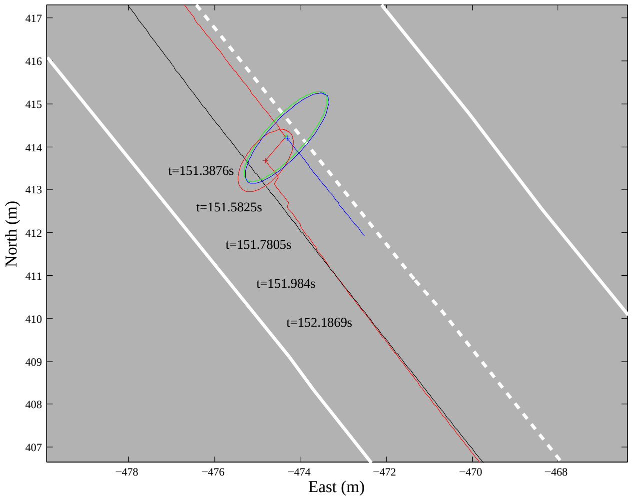
In conclusion, the multi-hypothesis strategy works well in handling lane ambiguities thanks to the lane marking type detected by the camera and the lane marking map. This is because the two lanes are distinguishable by their lane marking types in



(a) *Initialization of the filter at the beginning of the trial*

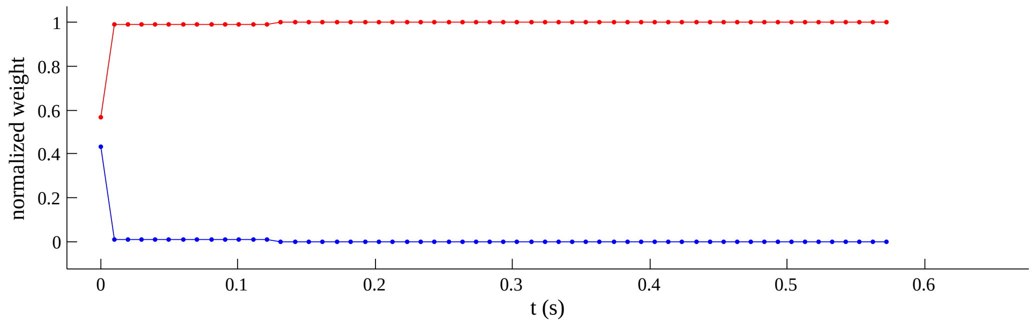


(b) *Behavior in the multipath area when the camera retrieves lane detections*

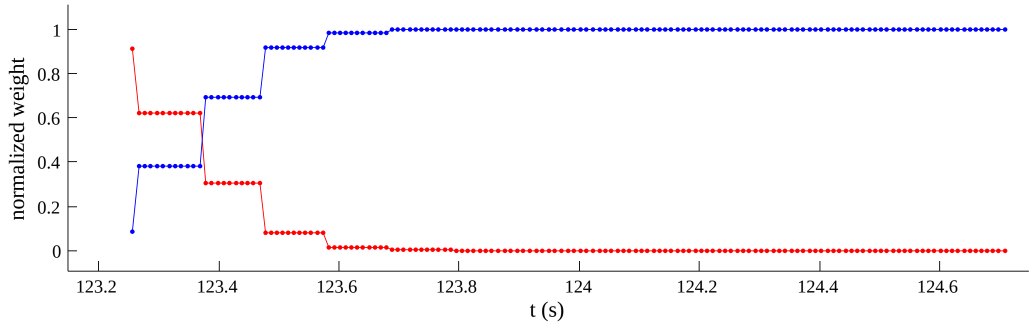


(c) *Behavior after a corner turning when the camera retrieves lane detections*

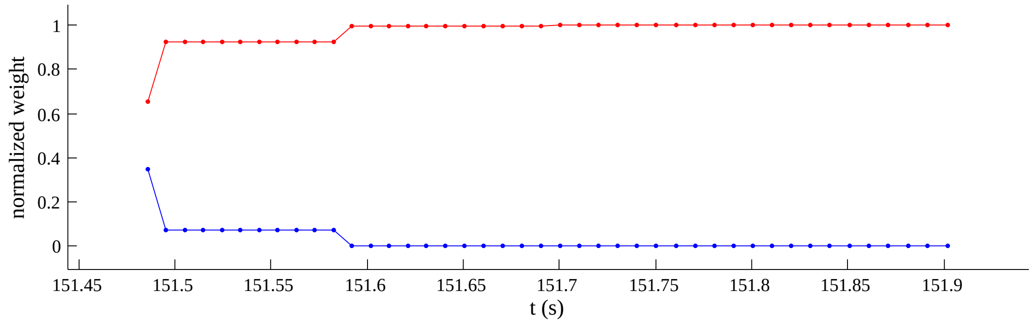
Figure 6.10: *The bold lines are lane markings of different types. The green point and the ellipse indicate the pose and the 3σ uncertainty domain where the mono-hypothesis is duplicated. The black line is the ground truth. The hypothesis with higher weight is in red. In Blue, the one with the lower weight. The red and blue ellipses are their corresponding 3σ uncertainty domains after the estimation of the first camera measurement. The filter is the ENU EKF.*



(a) *Initialization of the filter at the beginning of the trial*



(b) *Multipath area*



(c) *After the corner turning*

Figure 6.11: *Evolution of the weights. An hypothesis is killed when its weight is smaller than* 10^{-15}.

our experimental area. The method can't solve the map matching ambiguity if all the lane marking types are the same. Additionally, the vehicle didn't perform lane change maneuvers during the studied scenarios. This would introduce new issues such as merging similar hypotheses.

6.4 Conclusion

Generally speaking, the tightly coupled method has the best result in terms of accuracy and consistency. In addition, it improves the use of GNSS measurements as we have seen that less than four satellites are used sometimes. However, the design process is more complicated and the raw GNSS measurements have to be available in real-time. From our point of view, the tightly coupled method is the best when the vehicle is traveling in degraded GNSS environments, because the filter is more powerful to exclude the raw satellite measurements contaminated by multipath effect. In good GNSS environments, the LC RI EKF produces comparable accuracy but worse consistency. Another benefit of the tightly coupled method is that it can make use of the fact that the pseudorange bias of the same satellites is spatially correlated among networked GNSS receivers. The localization solver proposed in Chapter 5 can therefore be easily extended to a cooperative mode and the observability can be improved if the vehicles are traveling in different directions.

A multi-hypothesis lane tracking algorithm has also been studied in a case of ambiguity. The result shows that the data association (or map matching) ambiguity caused by low position confidence can be successfully handled by using both C_0 measurement and the lane marking type information. Although our system can not still cover all the situations especially in degraded environments for both GNSS and camera, the good performance in terms of consistency and the capability of handling map matching ambiguities enable an autonomous valet vehicle to stop autonomous navigation when the localization confidence is not high enough or when a lane ambiguity occurs.

Chapter 7

General conclusions

This dissertation has addressed an outdoor localization problem of a full-size robotized vehicle. The localization output is used as feedback information for the planning of autonomous driving. The study has been restricted to algorithms which fuse DR sensors, a L1-GNSS receiver, a lane detection camera, and a lane marking map. GNSS coupled with dead reckoning enables a localization system with high frequency and availability, but its meters level accuracy is far from the required performance for autonomous driving on constrained roads. The camera measurement coupled with the lane marking map has a better accuracy but is less available. Therefore, the objective is to fuse all the available information to get a real-time vehicle pose estimation with GNSS/DR-like availability and camera/map-like accuracy. Meanwhile, the challenge is how to work out a reliable system that fulfills high requirements with those low-cost information sources. It leads to a nonlinear state observation problem.

The classical approach to this problem is to use an extended Kalman filter according to a prediction-update scheme. Our first work have been devoted to the elaboration of a vehicle kinematic model and sensor observation models especially for the camera measurement. The lane marking map has been built by a mapping company and experimental data were collected by running an equipped vehicle on the mapped area. By studying the experimental data, we have observe that the proposed camera observation model have decimeter-level accuracy. This is a first validation of its effectiveness. Then, we have switched our focus on shaping filtering the non-white GNSS and dead reckoning errors, and the design and implementation of several localization solvers. Three localization solvers which differs with each other in error modeling (simple model and enhanced model), and fusion scheme (loosely and tightly coupled schemes in terms of GNSS) have been studied.

The first localization solver is based on a loosely coupled scheme which integrates a widely used autoregressive modeling for GNSS errors. This approach gives a first validation of the feasibility of fusing our low-cost sensors and map to improve the localization performance. The results inspire us to study how the camera measurement helps in estimating the GNSS errors and whether we can implement an enhanced error model. This motivation leads to a study of the observability property of the localization system. We have completed this study under the algebraic framework and in a road-oriented frame which simplifies the camera observation model. Hereafter, we have designed a road invariant localization solver to enable the enhanced modeling and preserve the observability of the state. Experimental results have demonstrated its practicability and effectiveness. These two strategies follow loosely coupled scheme, which means the filter exploits position outputs from the GNSS receiver without paying a lot attention to its own filter. Methods under such scheme are actually not ideal in complex environments for GNSS and camera. We have encountered this situation repeatedly along an uneven road in dense urban environment, where the GNSS receiver suffers to heavy multipath and the camera fails to detect lane marking at the same time. After passing out of this road section, the position confidence is such low that it becomes very risky to reuse the camera measurement. Therefore, we switched to study a tightly coupled scheme considering that the localization filter is more able to exclude satellite measurements contaminated by unmodeled error due to multipath effect compared to the filter implemented in the standalone GNSS receiver. The results prove that the tightly coupling method is indeed superior to the loosely coupled ones in a degraded environment, but at the same time, this approach has the same performance than the loosely coupled ones in the open-sky areas. In those areas, the position estimates delivered by the u-blox receiver are usually quite reliable. Thus it is possible to achieve good estimations of their biases.

We finally proposed a method to handle the data association ambiguities, which is based on multi-hypothesis lane tracking. The lane type information proves to be very effective in finding the correct lane in the considered experimental site.

In conclusion, our research achieves a low-cost solution to vehicle localization which meets the high demands of autonomous driving. Even if the proposed methods have been tested in many autonomous driving experiments with an equipped vehicle of the Renault PAMU project, complementary tests have to be conducted in different environments to evaluate its robustness. Working out a qualified localization system which is accurate, provides reliable confidence bounds and is able to handle map matching ambiguities still remains as a challenge. In the following sections, we provide several future directions after summarizing our contributions.

Summary of the contributions

In this thesis, we have produced the following contributions:

- Outdoor experiment datasets have been collected. The experimental vehicles are able to record and synchronize sensor data which includes data of DR sensors, lane detection camera, GNSS and RTK-GNSS. It allows to evaluate algorithms proposed in this thesis and in turn supports the practicability of the widely adopted lane marking aided localization approaches.

- In Chapter 2, we have proposed a camera observation model coupled with a lane marking map. A sensitivity analysis to the camera parameters has been performed. Then its effectiveness in improving the localization performance has been validated by sensor fusion implemented both in loosely and tightly coupled schemes.
- In Chapter 4, we have proposed an enhanced modeling of GNSS fix biases. An observability analysis has been performed in the algebraic observability framework. When integrating lane marking measurements for vehicle localization, the observability on the cross-track direction is often better than the one in along-track direction. The proposed localization solver, which we refer to as road invariant EKF, is able to enhance the GNSS bias modeling in the cross-track direction and stays observable independently of the road geometry. We have experimentally demonstrated that the road invariant EKF can be used to provide a better localization performance in terms of accuracy and consistency, compared to the localization solver which models the GNSS bias by an autoregressive process and is implemented in the local ENU frame.
- In Chapter 5, we have presented a tightly coupling approach which integrates GNSS measurements at the pseudorange and Dopplers level. The idea is to model the biases on the pseudoranges which are mainly caused by the residual atmospheric error and ephemeris error coming from the computed satellite positions. The difficulty is that the state dimension is increased by the number of visible satellites. So the time-correlated model has been adopted and the proposed lane marking measurement model has been integrated to improve the observability. Experimental results show that the estimated bias have stable and observable behavior and the localization accuracy and consistence is highly improved especially in the urban area.
- In Chapter 6, we have studied a multi-hypothesis lane tracking method which succeeds in resolving the lane marking data association ambiguities in different scenarios.

Future Directions

For the road invariant EKF, the working frame is in accordance with the heading of the road defined in the map. The roads are represented by polylines in the map. If a road is curved (e.g. a road with sinusoidal shape), each segment of the polyline has a different heading. We expect to implement the road invariant EKF which changes its working frame according to road segments in our future work. For the tightly coupling approach, more work should be done to validate the estimated pseudorange bias. For example, comparing the estimated bias between two vehicles running at the same time in parallel roads. The errors on pseudoranges can also be examined by comparing with a differential GNSS. Enhanced modeling on pseudorange errors is possible to be implemented thanks to the integration of the camera measurements. The multi-hypothesis lane tracking discussed in this thesis is preliminary, more issues such as merge of similar hypotheses should be studied. Moreover, experiments on roads with different geometries should be carried out to test the performance of the proposed lane marking aided localization system.

Next, we present several future directions to improve vehicle localization, which are inspired by the research in this thesis and still remain in the low-cost scope.

Observability improvement in the along-track direction The meter-level accuracy in the along-track direction is mainly due to the variation of the road heading. In order to improve the observability in the longitudinal direction, there are two solutions. Firstly, the observability can be directly improved if measurements of cross-track feature such as stop line, zebra-line, and traffic signs can be integrated into the localization solver. Secondly, in a networked GNSS framework, if the vehicles are traveling in different directions, the pseudorange bias becomes observable both in cross-track and along-track direction by sharing pseudoranges and lane marking detections. Tightly coupling networked GNSS has a great potential to improve the bias estimation and localization performance. In addition, a vehicle which is not equipped with the lane detection camera can also benefit from the GNSS network.

Map-aided perception and localization The lane detection camera used in this thesis only detects vehicle's host lane. When the uncertainty in the cross-track direction increases, ambiguity on which lane the vehicle is in appears. In order to disambiguate lanes, innovation gating can be risky when the lateral uncertainty is large. So we mainly rely on the type of the detected lane marking. This strategy works in this thesis because the test road is a two-lane road mainly with dashed lane marking in the center and solid lane marking on both sides. For a multilane road, the lanes can be indistinguishable by the type of the lane marking. Multilane detection and ego-lane estimation algorithm will be helpful since it provides a direct observation on which lane the vehicle is in. Multilane detection is more challenging, but it can be improved by exploiting the prior knowledge defined in the lane marking map (e.g. lane number and each lane width). It relies on the localization system to extract the prior knowledge from the feature map. This will highly improved the robustness and availability of feature detection. The other way round, the map may be not up-to-date and the camera detection can be quite confident under ideal situations. In those situations, if the extracted features are inconsistent with the detected ones, the camera detections can be used to evaluate and update the map. It means that we can not only tightly couple the GNSS receiver, but also the camera and even the map to improve the localization.This perspective is close to the simultaneous mapping and localization problem to some extend, but we emphasize the role of the prior map in enhancing the feature detection and reducing the calculation amount.

Enhanced map at world-wide scale In addition to autonomous driving, the proposed map-aided localization system in this thesis can also enable many lane-level ADAS applications which are meaningful for improving our daily driving experience in short term. Envisaging these potential good utilities, the unsolved issue is how to build and maintain such detailed and accurate maps at large scale and with low-cost. Solutions rely on mobile mapping platforms are generally expensive, because the sensor suits are high-end and it needs professional company and staff to operate them. Moreover, maps relies on those platforms can hardly have a fast update rate. With the upgrading of market-ready automotive sensing capabilities, crowd-sourcing the data recorded by consumer-grade road vehicles, such as GNSS trajectories, lane marking detections, and even camera images, to build enhanced map is a potential way. Many research institutes and technology companies have started to work on this subject. However, the literature lacks discussions regarding this aspect. From our point of view, more attention should be paid on general solutions for building enhanced maps at world-wide scale.

Bibliography

[1] *Global Positioning System Standard Positioning Service Performance Standard.* Assistant Secretary of Defense for Command, Control, Communications, and Intelligence, 2001.

[2] Cityvip french project http://www.agence-nationale-recherche.fr/documents/aap/2007/finances/predit-tsfa-resumes-2007.pdf, 2007.

[3] Manual on uniform traffic control devices http://mutcd.fhwa.dot.gov/pdfs/2009r1r2/mutcd2009r1r2edition.pdf, 2012.

[4] *SaPPART white paper: better use of Global Navigation Satellite Systems for safer and greener transport - techniques and methods TMI1.* Septemper 2015.

[5] Wieser A., Gaggl M., and Hartinger H. Improved positioning accuracy with high-sensitivity gnss receivers and snr aided integrity monitoring of pseudo-range observations. In *Proceedings of the 18th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS 2005)*, pages 1545–1554, Sept. 2005.

[6] Mohamed A.H. *Optimizing the estimation procedure in INS/GPS Integration for kinematic applications.* PhD thesis, University of Calgary, 1999.

[7] H. Akaike. Information theory and an extension of the maximum likelihood principle. In B. N. Petrov and F. Csaki, editors, *Second International Symposium on Information Theory*, pages 267–281, Budapest, 1973. Akadémiai Kiado.

[8] Hirotugu Akaike. Fitting autoregressive models for prediction. *Annals of the Institute of Statistical Mathematics*, 21(1):243–247, December 1969.

[9] Brian D. O. Anderson and John B. Moore. *Optimal Filtering (Dover Books on Engineering).* Dover Publications, January 2005.

[10] Kai Oliver Arras, José A. Castellanos, Martin Schilt, and Roland Siegwart. Feature-based multi-hypothesis localization and tracking using geometric constraints. *Robotics and Autonomous Systems*, 44:41–53, 2003.

[11] M. Bahrami and M. Ziebart. Instantaneous doppler-aided rtk positioning with single frequency receivers. In *2010 IEEE/ION Position Location and Navigation Symposium (PLANS)*, pages 70–78, May 2010.

[12] Aharon Bar Hillel, Ronen Lerner, Dan Levi, and Guy Raz. Recent progress in road and lane detection: a survey. *Machine Vision and Applications*, 25(3):727–745, 2014.

[13] Yaakov Bar-shalom, X.-Rong Li, and Thiagalingam Kirubarajan. *Estimation with Applications to Tracking and Navigation*. John Wiley & Sons, Inc., New York, NY, USA, 2002.

[14] Jean-Charles Bazin and Marc Pollefeys. 3-line ransac for orthogonal vanishing point detection. *IEEE Conf. on Int. Robots and Systems*, pages 4282–4287, 2012.

[15] Jan Becker. Me302 advanced driver assistance, 2013.

[16] P. Bender, J. Ziegler, and C. Stiller. Lanelets: Efficient map representation for autonomous driving. In *2014 IEEE Intelligent Vehicles Symposium Proceedings*, pages 420–425, June 2014.

[17] J. Bennett. *OpenStreetMap*. Packt Publishing, Limited, 2010.

[18] P. J. Besl and H. D. Mckay. A method for registration of 3-D shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2):239–256, 1992.

[19] D. Bétaille and R. Toledo-Moreo. Creating enhanced maps for lane-level vehicle navigation. *IEEE Transactions on Intelligent Transportation Systems*, 11(4):786–798, Dec 2010.

[20] V Blervaque et al. Prevent maps&adas final report. *ERTICOITS Europe Std*, 2008.

[21] S. Bonnabel, P. Martin, and E. Salaun. Invariant extended kalman filter: theory and application to a velocity-aided attitude estimation problem. In *Proceedings of the 48th IEEE Conference on Decision and Control*, pages 1297–1304, Dec 2009.

[22] P. Bonnifait, P. Bouron, P. Crubille, and D. Meizel. Data fusion of four abs sensors and GPS for an enhanced localization of car-like vehicles. In *2001 IEEE International Conference on Robotics and Automation*, volume 2, pages 1597–1602 vol.2, 2001.

[23] Pinar Boyraz, Xuebo Yang, and JohnH.L. Hansen. Computer vision systems for "context-aware" active vehicle safety and driver assistance. In John H.L. Hansen, Pinar Boyraz, Kazuya Takeda, and Huseyin Abut, editors, *Digital Signal Processing for In-Vehicle Systems and Safety*, pages 217–227. Springer New York, 2012.

[24] R.G. Brown and P.Y.C. Hwang. *Introduction to random signals and applied Kalman filtering: with MATLAB exercises and solutions*. Number v. 1 in Introduction to Random Signals and Applied Kalman Filtering: With MATLAB Exercises and Solutions. Wiley, 1997.

[25] J.P. Burg. *Maximum entropy spectral analysis*. Stanford Exploration project. Stanford University, 1975.

[26] Lepinsy Chanthalansy and Aboelmagd Noureldin. Carrier-to-noise density and ai for INS/GPS integration. *Inside GNSS*, 2009(9/10):20–29, 2009.

[27] Manolis N. Chatzis, Eleni N. Chatzi, and Andrew W. Smyth. On the observability and identifiability of nonlinear structural and mechanical systems. *Structural Control and Health Monitoring*, 22(3):574–593, 2015.

[28] Kreucher Chris, Lakshmanan Sridhar, and Kluge Karl. A driver warning system based on the lois lane detection algorithm. *Int. Vehicles Symp.*, 1:17–22, 1998.

[29] J.M. Clanton, D.M. Bevly, and A.S. Hodel. A low-cost solution for an integrated multisensor lane departure warning system. *IEEE Transactions on Intelligent Transportation Systems*, 10(1):47–59, March 2009.

[30] S. Cooper and H. Durrant-Whyte. A kalman filter model for GPS navigation of land vehicles. In *1994 IEEE/RSJ International Conference on Intelligent Robots and Systems*, volume 1, pages 157–163 vol.1, Sep 1994.

[31] Arinc R. Corporation. NAVSTAR GPS space segment. Navigation user interfaces (IDC-GPS-200C). Revision IRN-200D-001. Technical report, GPS NAVSTAR JPO, March 2006.

[32] J.L. Crassidis. Sigma-point kalman filtering for integrated GPS and inertial navigation. *IEEE Transactions on Aerospace and Electronic Systems*, 42(2):750–756, April 2006.

[33] Dixiao Cui, Jianru Xue, Shaoyi Du, and Nanning Zheng. Real-time global localization of intelligent road vehicles in lane-level via lane marking detection and shape registration. In *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4958–4964, Sept 2014.

[34] P. Czerwionka, Miao Wang, and F. Wiesel. Optimized route network graph as map reference for autonomous cars operating on german autobahn. In *2011 IEEE International Conference on Robotics and Automation*, pages 78–83, Dec 2011.

[35] DARPA. Urban Challenge Route Network Definition File (RNDF) and Mission Data File (MDF) Formats, March 2007.

[36] P. Delmas, C. Tessier, C. Debain, and R. Chapuis. Gnss bias correction for localization systems. In *2008 11th International Conference on Information Fusion*, pages 1–6, June 2008.

[37] S. Diop and Yuan Wang. Equivalence between algebraic observability and local generic observability. In *Proceedings of the 32nd IEEE Conference on Decision and Control*, pages 2864–2865 vol.3, Dec 1993.

[38] D. Douglas and T. Peucker. Algorithms for the reduction of the number of points required to represent a digitized line or its caricature. *Cartographica: The International Journal for Geographic Information and Geovisualization*, 10(2):112–122, 1973.

[39] V. Drevelle and P. Bonnifait. Localization confidence domains via set inversion on short-term trajectory. *IEEE Transactions on Robotics*, 29(5):1244–1256, Oct 2013.

[40] M. Fliess and S.T. Glad. An algebraic approach to linear and nonlinear control. In H.L. Trentelman and J.C. Willems, editors, *Essays on Control*, volume 14 of *Progress in Systems and Control Theory*, pages 223–267. Birkhäuser Boston, 1993.

[41] Michel Fliess. Automatique et corps différentiels. *Forum mathematicum*, 1(3):227–238, 1989.

[42] C. Fouque and P. Bonnifait. On the use of 2d navigable maps for enhancing ground vehicle localization. In *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1885–1890, Oct 2009.

[43] C. Fouque, P. Bonnifait, and D. Bétaille. Enhancement of global vehicle localization using navigable road maps and dead-reckoning. In *Position, Location and Navigation Symposium, 2008 IEEE/ION*, pages 1286–1291, May 2008.

[44] P. Furgale et al. Toward automated driving in cities using close-to-market sensors: An overview of the v-charge project. In *2013 IEEE Intelligent Vehicles Symposium Proceedings*, pages 809–816, June 2013.

[45] J Gao, MG Petovello, and ME Cannon. Development of precise GPS/INS/wheel speed sensor/yaw rate sensor integrated vehicular positioning system. *Proceedings of ION National Technical Meeting*, pages 780–792, 2006.

[46] Itay Gat, Meny Benady, and Amnon Shashua. A Monocular Vision Advance Warning System for the Automotive Aftermarket. In *SAE Technical Paper 2005-01-1470*. SAE, SAE, 2005.

[47] A. Geiger, M. Lauer, C. Wojek, C. Stiller, and R. Urtasun. 3d traffic scene understanding from movable platforms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(5):1012–1025, May 2014.

[48] A. Gelb. *Applied optimal estimation.* MIT Press, 1974.

[49] S.T. Glad and L. Ljung. Model structure identifiability and persistence of excitation. In *Proceedings of the 29th IEEE Conference on Decision and Control*, pages 3236–3240 vol.6, Dec 1990.

[50] M.S. Grewal and A.P. Andrews. *Kalman filtering: theory and practice using MATLAB.* [Wiley InterScience Online Books, Electronic and Electrical Engineering Collection]. Wiley, 2001.

[51] P.D. Groves. *Principles of GNSS, Inertial, and Multisensor Integrated Navigation Systems, Second Edition:*. GNSS/GPS. Artech House, 2013.

[52] D. Gruyer, R. Belaroussi, and M. Revilloud. Map-aided localization with lateral perception. In *2014 IEEE Intelligent Vehicles Symposium Proceedings*, pages 674–680, June 2014.

[53] Dominique Gruyer, Rachid Belaroussi, and Marc Revilloud. Accurate lateral positioning from map data and road marking detection. *Expert Systems with Applications*, 43:1 – 8, 2016.

[54] Chunzhao Guo, J.-I. Meguro, Y. Kojima, and T. Naito. Automatic lane-level map generation for advanced driver assistance systems using low-cost sensors. In *2014 IEEE International Conference on Robotics and Automation*, pages 3975–3982, May 2014.

[55] Fredrik Gustafsson, Umut Orguner, ThomasB. Schon, Per Skoglar, and Rickard Karlsson. Navigation and tracking of road-bound vehicles using map support. In Azim Eskandarian, editor, *Handbook of Intelligent Vehicles*, pages 397–434. Springer London, 2012.

[56] R.Benjamin Harris and RichardG. Mach. The GPSTk: an open source GPS toolkit. *GPS Solutions*, 11(2):145–150, 2007.

[57] R. Hermann and Arthur J. Krener. Nonlinear controllability and observability. *IEEE Transactions on Automatic Control*, 22(5):728–740, Oct 1977.

[58] Albert Huang, David Moore, Matthew Antone, Edwin Olson, and Seth Teller. Multi-Sensor Lane Finding in Urban Road Networks. In *Proceedings of Robotics: Science and Systems IV*, Zurich, Switzerland, June 2008.

[59] Official U.S. Government information about the Global Positioning System and related topics. Control segment.

[60] Official U.S. Government information about the Global Positioning System and related topics. Space segment.

[61] Gao J. GPS/INS/G sensors/yaw rate sensor/wheel speed sensors integrated vehicular positioning system. In *Proceedings of the 19th International Technical Meeting of the Satellite Division of The Institute of Navigation*, pages 1427–1439, Sept. 2006.

[62] Maged Jabbour, Philippe Bonnifait, and Véronique Cherfaoui. Map-matching integrity using multihypothesis road-tracking. *Journal of Intelligent Transportation Systems*, 12(4):189–201, 2008.

[63] L. B. Jackson. *Digital Filters and Signal Processing.* Kluwer Academic Publishers, 1996.

[64] Gross Jason, Gu Yu, Gururajan Srikanth, Seanor Brad, and Napolitano Marcello R. A comparison of extended kalman filter, sigma-point kalman filter, and particle filter in GPS/INS sensor fusion. *AIAA Guidance, Navigation, and Control Conference, Toronto, Canada*, 2010.

[65] A.H. Jazwinski. *Stochastic Processes and Filtering Theory.* Mathematics in science and engineering. Academic Press, 1970.

[66] Kichun Jo, Keonyup Chu, and Myoungho Sunwoo. GPS-bias correction for precise localization of autonomous vehicles. *Int. Vehicles Symp.*, pages 636–641, June 2013.

[67] Simon J. Julier and Jeffrey K. Uhlmann. A new extension of the kalman filter to nonlinear systems. pages 182–193, 1997.

[68] S.J. Julier and H.F. Durrant-Whyte. On the role of process models in autonomous land vehicle navigation systems. *IEEE Transactions on Robotics and Automation*, 19(1):1–14, Feb 2003.

[69] Rudolph Emil Kalman. A new approach to linear filtering and prediction problems. *Transactions of the ASME–Journal of Basic Engineering*, 82(Series D):35–45, 1960.

[70] E. Kaplan. *Understanding GPS - Principles and applications.* Artech House, 2nd edition edition, December 2005.

[71] Alonzo Kelly. A 3d state space formulation of a navigation kalman filter for autonomous vehicles. Technical report, 1994.

[72] Wuk Kim, Gyu-In Jee, and Janggyu Lee. Efficient use of digital road map in various positioning for its. In *IEEE Position Location and Navigation Symposium*, pages 170–176, 2000.

[73] Karl Kluge. Extracting road curvature and orientation from image edge points without perceptual grouping into features. *Int. Vehicles Symp.*, pages 109–114, 1994.

[74] Marek Kurdej, Julien Moras, Veronique Cherfaoui, and Philippe Bonnifait. Controlling remanence in evidential grids using geodata for dynamic scene perception. *International Journal of Approximate Reasoning*, 55(1, Part 3):355 – 375, 2014.

[75] J. Laneurit, R. Chapuis, and F. Chausse. Accurate vehicle positioning on a numerical map. *International Journal of Control, Automation, and Systems*, 3(1):15–31, March 2006.

[76] Richard B. Langley. Nmea 0183-a GPS receiver interface standard. In *GPS World*, volume 7, pages 54–57, July 1995.

[77] Richard B. Langley. RTK GPS. In *GPS World*, volume 9, pages 70–76, September 1998.

[78] Nhu D. Le, Adrian E. Raftery, and R. Douglas Martin. Robust bayesian model selection for autoregressive processes with additive outliers. *Journal of the American Statistical Association*, 91(433):pp. 123–131, 1996.

[79] Byung-Hyun Lee, Jong-Hwa Song, Jun-Hyuck Im, Sung-Hyuck Im, Moon-Beom Heo, and Gyu-In Jee. GPS/DR error estimation for autonomous vehicle localization. *Sensors*, 15(8):20779, 2015.

[80] J. Levinson, J. Askeland, J. Becker, J. Dolson, D. Held, S. Kammel, J.Z. Kolter, D. Langer, O. Pink, V. Pratt, M. Sokolsky, G. Stanek, D. Stavens, A. Teichman, M. Werling, and S. Thrun. Towards fully autonomous driving: Systems and algorithms. In *2011 IEEE Intelligent Vehicles Symposium Proceedings*, pages 163–168, June 2011.

[81] J. Levinson, J. Askeland, J. Dolson, and S. Thrun. Traffic light mapping, localization, and state detection for autonomous vehicles. In *2011 IEEE International Conference on Robotics and Automation,*, pages 5784–5791, May 2011.

[82] Jesse Levinson, Michael Montemerlo, and Sebastian Thrun. Map-Based Precision Vehicle Localization in Urban Environments. In *Robotics: Science and Systems*, 2007.

[83] T. Litman. *Autonomous Vehicle Implementation Predictions: Implications for Transport Planning.* 2013.

[84] L. Ljung and S.T. Glad. On global identifiability for arbitrary model parametrizations. *Automatica*, 30(2):265 – 276, 1994.

[85] Bing Ma, S. Lakshmanan, and A.O. Hero. Simultaneous detection of lane and pavement boundaries using model-based multisensor fusion. *IEEE Transactions on Intelligent Transportation Systems*, 1(3):135–147, Sep 2000.

[86] Rhudy Matthew, Gu Yu, Gross Jason, Gururajan Srikanth, and Napolitano Marcello R. Sensitivity analysis of extended and unscented kalman filters for attitude estimation. *Journal of Aerospace Information Systems*, 10(3):131 – 143, 2004.

[87] Peter S. Maybeck. *Stochastic models, estimation, and control*, volume 141 of *Mathematics in Science and Engineering.* 1979.

[88] J.C. McCall and M.M. Trivedi. Video-based lane estimation and tracking for driver assistance: survey, system, and evaluation. *IEEE Transactions on Intelligent Transportation Systems*, 7(1):20–37, March 2006.

[89] Isaac Miller, Mark Campbell, and Dan Huttenlocher. Map-aided localization in sparse global positioning system environments using vision and particle filtering. *Journal of Field Robotics*, 28(5):619–643, 2011.

[90] P. Misra and P. Enge. *Global Positioning System: Signals, Measurements, and Performance.* Ganga-Jamuna Press, Lincoln MA, 2nd edition edition, 2006.

[91] Maan E. El Najjar. A road-matching method for precise vehicle localization using belief theory and kalman filtering. *Auton. Robots*, 19:173–191, 2005.

[92] El-Sheimy Naser, Shin Eun-Hwan, and Niu Xiaoji. Kalman filter face-off: Extended vs. unscented kalman filters for integrated GPS and MEMS inertial. *Inside GNSS*, (2):48–54, March 2006.

[93] Sameh Nassar. *Improving the Inertial Navigation System (INS) Error Model for INS and INS/DGPS Applications.* PhD thesis, University of Calgary, 2003.

[94] G. Nassreddine, F. Abdallah, and T. Denoeux. Map matching algorithm using interval analysis and dempster-shafer theory. In *2009 IEEE Intelligent Vehicles Symposium*, pages 494–499, June 2009.

[95] National Imagery and Mapping Agency. Department of defense world geodetic system 1984: its definition and relationships with local geodetic systems. Technical Report TR8350.2, National Imagery and Mapping Agency, St. Louis, MO, USA, January 2000.

[96] E.M. Nebot, H. Durrant-Whyte, and S. Scheding. Frequency domain modeling of aided GPS with application to high-speed vehicle navigation systems. In *1997 IEEE International Conference on Robotics and Automation.*, volume 3, pages 1892–1897 vol.3, Apr 1997.

[97] Pascal Neis and Alexander Zipf. Analyzing the contributor activity of a volunteered geographic information project-the case of openstreetmap. *ISPRS International Journal of Geo-Information*, 1(2):146–165, 2012.

[98] S. J. Orfanidis. *Optimum Signal Processing, An Introduction.* Prentice-Hall, second edition, 1996.

[99] Bonnifait P., Garcia G., and Peyret F. A system for 3d localization of civil-engineering machines. In *Workshop "Navigation of Outdoor Autonomous Vehicles" of ICRA 1998*, May 1998.

[100] M. Phatak and S. Kohli. Position fix from three GPS satellites and altitude: a direct method. *IEEE Transactions on Aerospace and Electronic Systems*, 35(1):350–354, Jan 1999.

[101] O. Pink. Visual map matching and localization using a global feature map. In *Computer Vision and Pattern Recognition Workshops, 2008. CVPRW '08. IEEE Computer Society Conference on*, pages 1–7, June 2008.

[102] O. Pink and C. Stiller. Automated map generation from aerial images for precise vehicle localization. In *2010 International IEEE Conference on Intelligent Transportation Systems*, pages 1517–1522, Sept 2010.

[103] Mohammed A Quddus, Robert B Noland, and Washington Y Ochieng. A high accuracy fuzzy logic based map matching algorithm for road transport. *Journal of Intelligent Transportation Systems*, 10(3):103–115, 2006.

[104] Mohammed A. Quddus, Washington Y. Ochieng, and Robert B. Noland. Current map-matching algorithms for transport applications: State-of-the art and future research directions. *Transportation Research Part C: Emerging Technologies*, 15(5):312 – 328, 2007.

[105] Zanetti Renato and D'Souza Chris. Dual accelerometer usage strategy for onboard space navigation. *Journal of Guidance, Control, and Dynamics*, 35(6):1899–1902, 2012.

[106] J. Rife. Collaborative vision-integrated pseudorange error removal: Team-estimated differential GNSS corrections with no stationary reference receiver. *IEEE Transactions on Intelligent Transportation Systems*, 13(1):15–24, March 2012.

[107] J.F. Ritt. *Differential Algebra*. American Mathematical Society: Colloquium publications. American Mathematical Society, 1950.

[108] M. Schreiber, C. Knoppel, and U. Franke. Laneloc: Lane marking based localization using highly accurate maps. In *2013 IEEE Intelligent Vehicles Symposium Proceedings*, pages 449–454, June 2013.

[109] Wolfgang Schwanghart. Recursive douglas-peucker polyline simplification, 2010.

[110] Gideon Schwarz. Estimating the Dimension of a Model. *The Annals of Statistics*, 6(2):461–464, 1978.

[111] Gunter Seeber. *Satellite Geodesy*. Walter de Gruyter, 2 revised edition, 2003.

[112] Young-Woo Seo and R. Rajkumar. Tracking and estimation of ego-vehicle's state for lateral localization. In *2014 International IEEE Conference on Intelligent Transportation Systems*, pages 1251–1257, Oct 2014.

[113] H. Sert, W. Perruquetti, A. Kokosy, X. Jin, and J. Palos. Localizability of unicycle mobiles robots: An algebraic point of view. *IEEE Conf. on Int. Robots and Systems*, pages 223–228, 2012.

[114] Eduardo D. Sontag. On the observability of polynomial systems, i: Finite-time problems. *SIAM Journal on Control and Optimization*, 17(1):139–151, 1979.

[115] M. St-Pierre and D. Gingras. Comparison between the unscented kalman filter and the extended kalman filter for the position estimation module of an integrated navigation information system. In *2004 IEEE Intelligent Vehicles Symposium*, pages 831–835, June 2004.

[116] Coast Stephen. Tesla maps and the exploding future of map data, October 2015.

[117] S. Sukkarieh, E.M. Nebot, and H.F. Durrant-Whyte. A high integrity IMU/GPS navigation loop for autonomous land vehicle applications. *IEEE Transactions on Robotics and Automation*, 15(3):572–578, Jun 1999.

[118] Z. Tao, P. Bonnifait, V. Frémont, and J. Ibañez-Guzman. Lane marking aided vehicle localization. In *2013 International IEEE Conference on Intelligent Transportation Systems*, pages 1509–1515, Oct 2013.

[119] Z. Tao, P. Bonnifait, V. Frémont, and J. Ibañez-Guzman. Mapping and localization using GPS, lane markings and proprioceptive sensors. In *2013*

IEEE/RSJ International Conference on Intelligent Robots and Systems, pages 406–412, Nov 2013.

[120] George Taylor, Geoffrey Blewitt, Doerte Steup, Simon Corbett, and Adrijana Car. Road reduction filtering for GPS-GIS navigation. *Transactions in GIS*, 5(3):193–207, 2001.

[121] Office of the Press Secreta The White House. *Statement by the President Regarding the United States Decision to Stop Degrading Global Positioning System Accuracy.* White House Press Announcement, May 2000.

[122] R. Toledo-Moreo, D. Bétaille, and F. Peyret. Lane-level integrity provision for navigation and map matching with gnss, dead reckoning, and enhanced maps. *IEEE Transactions on Intelligent Transportation Systems*, 11(1):100–112, March 2010.

[123] Alessandro Corrêa Victorino, Patrick Rives, and Jean-Jacques Borrelly. Safe navigation for indoor mobile robots. part ii: Exploration, self-localization and map building. *The International Journal of Robotics Research*, 22(12):1019–1039, 2003.

[124] Kumar Vishal, C.V. Jawahar, and Visesh Chari. Accurate localization by fusing images and GPS signals. In *2015 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 17–24, June 2015.

[125] Ulrich Vollath, Alois Deking, Herbert L, and Trimble Terrasat Gmbh. Long range rtk positioning using virtual reference stations. In *Proceedings of the International Symposium on Kinematic Systems in Geodesy, Geomatics and Navigation*, 2001.

[126] Anh Vu, A. Ramanandan, Anning Chen, J.A. Farrell, and M. Barth. Real-time computer vision/DGPS-aided inertial navigation system for lane-level vehicle navigation. *Intelligent Transportation Systems, IEEE Transactions on*, 13(2):899–913, June 2012.

[127] Eric A. Wan and Rudolph Van Der Merwe. The unscented kalman filter for nonlinear estimation. pages 153–158, 2000.

[128] Lijun Wei. *Multi-sources fusion based vehicle localization in urban environments under a loosely coupled probabilistic framework.* Theses, Université de Technologie de Belfort-Montbeliard, July 2013.

[129] Greg Welch and Gary Bishop. An introduction to the kalman filter. Technical report, Chapel Hill, NC, USA, 1995.

[130] Jan Wendel and Gert F. Trommer. Tightly coupled GPS/INS integration for missile applications. *Aerospace Science and Technology*, 8(7):627 – 634, 2004.

[131] R.W. Wolcott and R.M. Eustice. Visual localization within lidar maps for automated urban driving. In *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 176–183, Sept 2014.

[132] Oliver J. Woodman. An introduction to inertial navigation. Technical Report UCAM-CL-TR-696, University of Cambridge, Computer Laboratory, August 2007.

[133] Zhengyou Zhang. Iterative point matching for registration of free-form curves and surfaces. *International Journal of Computer Vision*, 13(2):119–152, 1994.

[134] J. Ziegler et al. Making bertha drive-an autonomous journey on a historic route. *Intelligent Transportation Systems Magazine, IEEE*, 6(2):8–20, 2014.

Nomenclature

ARV	Autonomous road vehicle
BDS	Beidou navigation satellite system
BPSK	Bi-Phase Shift Keying
CAN	Controller-area network
DGPS	Differential GPS
ECEF	Earth-Centered Earth-Fixed Coordinate System
ECI	Earth-Centered Inertial
EKF	Extended Kalman filter
GLONASS	Globalanaya Navigatsionnaya Sputnikovaya Sistema
GPS	Global Positioning System
ICP	Iterative closest point
LDWS	Lane departure warning system
MCS	Master control station
MH	Multi-hypothesis
OSM	OpenStreetMap
PPP	Precise point positioning
PSD	Power spectral density
RCS	Reference coordinate system
UKF	Unscented Kalman filter
WGS	World Geodetic System
WGS84	World Geodetic System 1984

Appendix A

Mobile mapping and localization at Compiègne

Outdoor experiments have been carried out in the city of Compiègne. Figure A.1 shows the experimental vehicle.

The experimental vehicle was equipped with a NovAtel RTK-GPS receiver coupled with a SPAN-CPT IMU running at 100Hz. The system received RTCM 3.0 corrections through a 3G connection from a GPS base station Septentrio PolaRx2e@ equipped with a Zephyr Geodetic antenna. This high accuracy system (few tens of centimeter) was used during the mobile mapping process. It also provided ground truth data for the localization method. The station was located at the research center in Compiègne. It was the origin O of the local coordinate system (latitude 49.4°, longitude 2.796°, ellipsoidal height 83 m). A MobilEye camera was used to detect the lane markings. The low-cost L1-GNSS was a u-blox 4T with a patch antenna with no EGNOS correction.

The main purpose of this chapter is to demonstrate the effectiveness of the proposed camera observation model and localization algorithms on curved roads. In the following sections, we describe the main steps that are involved in the map building stage and the experimental results.

A.1 Lane marking map building

As the mobile mapping is done in post-processing, it is quite easy to synchronize the RTK-GPS positioning data with the camera measurements by interpolating the trajectory. We collect lane marking points (x_L, y_L) as follows:

$$\begin{bmatrix} x_L \\ y_L \end{bmatrix} = \begin{bmatrix} P_x \cdot \cos\psi + C_0 \cdot \sin\psi + x \\ P_x \cdot \sin\psi - C_0 \cdot \cos\psi + y \end{bmatrix} \tag{A.1}$$

Figure A.1: *Mobile mapping vehicle*

A.1.1 Clustering points by lanes

Lane marking points are graphically regrouped by polylines. Keeping in the map every mapped point can provide a huge amount of data which will not be efficient for real-time navigation particularly in the map matching stage. Therefore, the obtained clusters of marking points are simplified. In the following, we describe the key stages to do this simplification.

A.1.2 Polylines segmentation

We need to find the shape points of the simplified polylines. In other words, we need to identify the segments. Between every two adjacent shape points, there is a segment, with a certain bearing, which stands for a length of lane marking. The Douglas–Peucker's algorithm [38] is a well known algorithm for reducing the number of points in a curve that is approximated by a series of points. It's used here to find the shape points which divide the lane marking into parts with different headings. In this stage, the accuracy is controlled by choosing suitable tolerance which is the maximal euclidean distance allowed between the simplified segments and the origin points.

Let consider the example described in figure A.2. p is a set of lane marking points. By using Douglas–Peucker's algorithm, one can obtain shaping points ps_1, ps_2 and ps_3 which are three points chosen from p. For every point between ps_1 and ps_2, their Euclidean distances to segment $[ps_1; ps_2]$ are smaller than the tolerance, the same stands for the points between ps_2 and ps_3.

If we choose a very small tolerance, the segments in a polyline can be too short, and there can be too many shape points in the polyline. For this reason, we propose to choose a tolerance of decimeter-level.

A.1.3 Improving map accuracy

In order to further improve the accuracy of the polylines by reducing the errors effects due to the segmentation process, a least-squares algorithm is performed with every point between two adjacent shape points of the segmented polyline. For instance, in figure A.2, one can notice that every point between ps_1 and ps_2 is above segment $[ps_1; ps_2]$.

Let consider a polyline with n points. After $n-1$ least-squares stages, $n-1$ new straight lines are formed (blue lines in Fig. A.2). It is then necessary to retrieve the nodes (endpoints) and shape points of the new polyline. There are two cases. The intersection of two successive lines define a new shape point (e.g. $ps_2^{'}$). By convention, the new nodes are defined as the orthogonal projection of the previous

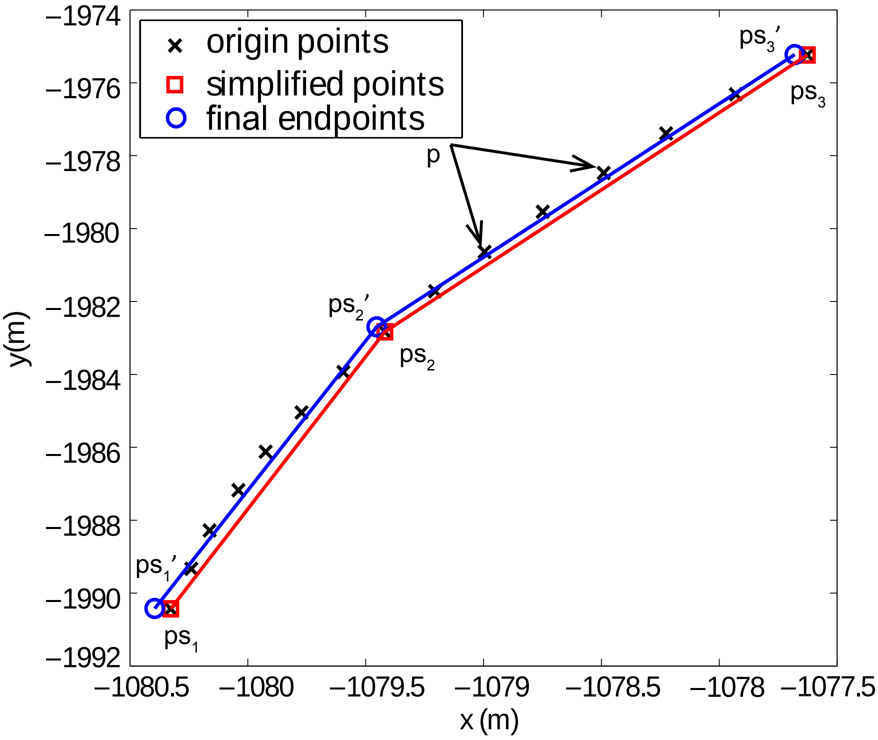


Figure A.2: *Illustration of the 2 stages lane marking fitting. The tolerance is 20 centimeters.*

node on the new line (e.g ps'_1). To come back to the example, the new polyline is $\left[ps'_1;\ ps'_2;\ ps'_3\right]$ at the end of the process. It is the result that is finally stored in the digital map.

The algorithm A.1 resumes this process.

Algorithm A.1 Polyline fitting algorithm

```
Input: set of points $p$ in the working frame $R_0$
$\left(ps_1,\ \ldots\ , ps_n\right) = dpsimplify(p, tolerance)$[109]
for $i = 2, \ldots, n$ do
    $line_{i-1}$ ←Least_squares_fitting(points between $ps_{i-1}$ and $ps_i$)
    if $i \neq n$ then
        $ps'_i$ ←intersection of $line_{i-1}$ and $line_i$
    end if
end for
$ps'_1$ ←orthogonal projection of $ps_1$ on $line_1$
$ps'_n$ ←orthogonal projection of $ps_n$ on $line_{n-1}$
Output: shape points $ps'$ of the polyline
```

A.1.4 Modifying the clustering stage

In order to perform the Douglas–Peucker's algorithm [38], it should be noticed that the set of points has to be linked by a function (i.e. each abscissa value must have only one image). It means that the points in the same cluster have to be arranged in a way that the x-coordinates are monotonically increasing or decreasing. Therefore, lane markings sets have to be divided into subsets when this condition is not verified. Figure A.3 gives an example.

As a conclusion, the clustering described in subsection A.1.1 is modified as follows:

- the points in one set have to physically belong to the same lane marking;
- the points in one set should meet the requirement of a function because of the Douglas–Peucker's algorithm;
- the points in one set should be close to each other. Indeed, when doing the collection of lane marking points, the camera can miss some lane markings, especially when the vehicle is making a turn. So, the distance between two following points in one set can be high and the resulting segment between them can be far from the real lane marking.

In practice, in order to do a good quality mapping, the lane marking points are manually divided into suitable different clusters.

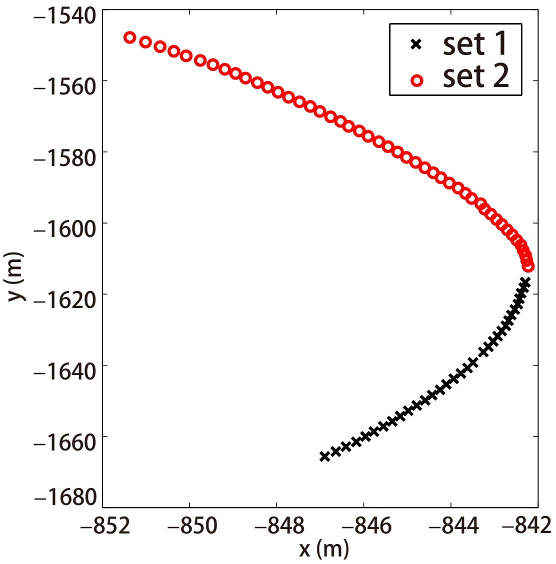


Figure A.3: *Example where consecutive points of the same lane marking are divided into two clusters*

Please note that some close lane marking detections may belong to two sides of the same physical marking. In this case, we fit both edges of the lane marking. This can occur when the marking is located between two adjacent lanes.

A.2 Experimental result

We performed two paths on the same road (one-way double lane). The length of the path is 3736 m. The vehicle passed three crossroads and a fork road. During the first path, lane marking points have been collected to build the lane marking map. The second path is used to test the localization solver (see chapter 3). Figure A.4 shows the obtained lane marking map (blue lines).

One can notice that the blue lines are quite consistent with the lane markings on the road. Though not all the lane markings are mapped (only one path is used for mobile mapping and the camera also missed some lane markings, especially at the crossroads), this map is enough to test the performance of the localization system.

A loosely coupled ENU EKF is implemented. Figure A.5 shows lateral and longitudinal positioning errors. Three methods are compared: L1-GPS (green lines), DR coupled with L1-GPS (red lines) and lane marking aided localization (black lines). The blue/green points indicate there are good lane marking detections on left/right side at the moment. Camera observations are available 45.9% of the time. One can see that, first, the data fusion of GPS with DR sensors enhances the accuracy. Second, the use of the camera with the mapped lane marking increases significantly

Figure A.4: *Map of a test site superimposed on an OpenStreetMap image*

the performance.

Table A.1 gives performance metrics of the localization solver. 95% horizontal positioning error is less than 1.25 m.

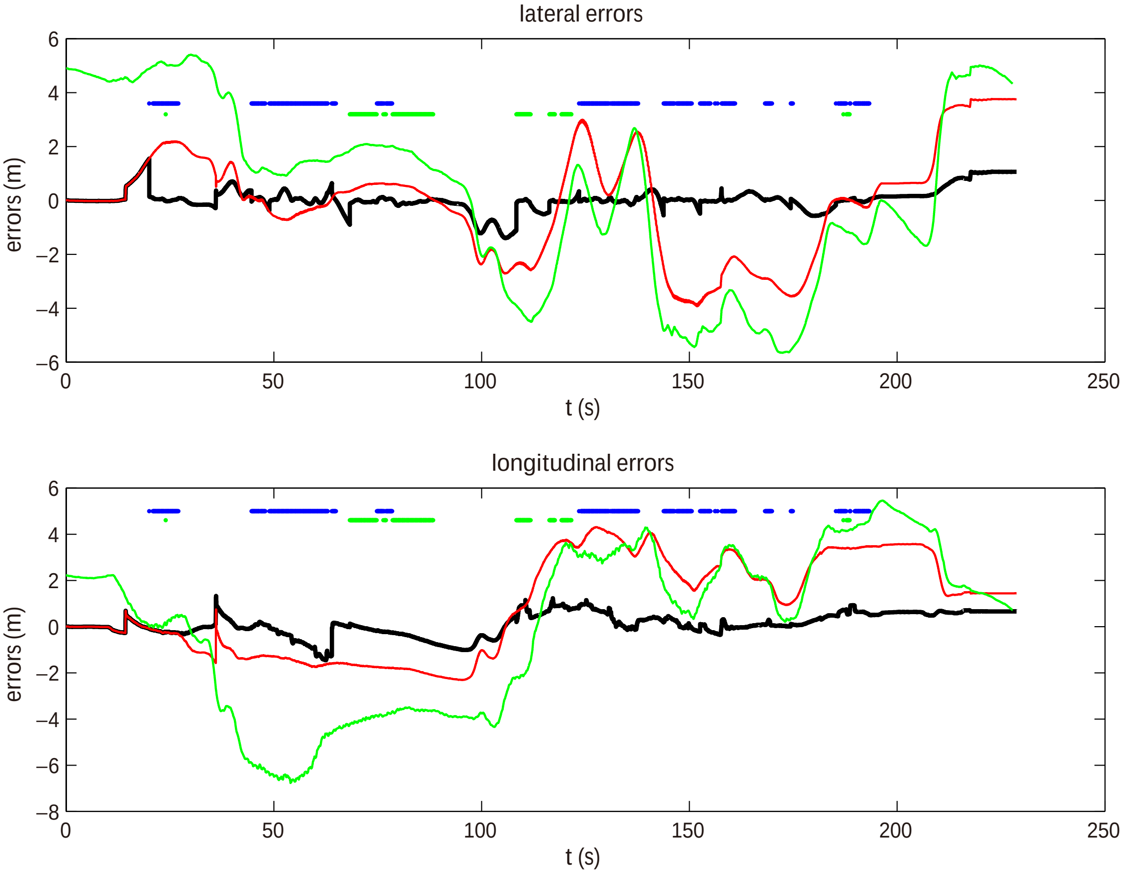


Figure A.5: *Lateral and longitudinal positioning errors*

	Horizontal PE (m)		*Lateral PE (m)*		*Longitudinal PE (m)*	
	I	*II*	*I*	*II*	*I*	*II*
mean	*2.74*	*0.54*	*1.49*	*0.26*	*1.96*	*0.39*
std. dev.	*1.23*	*0.39*	*1.26*	*0.34*	*1.16*	*0.39*
max	*4.85*	*1.56*	*3.92*	*1.56*	*4.31*	*1.46*
median	*3.23*	*0.53*	*1.09*	*0.11*	*1.74*	*0.36*
95th percentile	*4.23*	*1.25*	*3.73*	*1.06*	*3.76*	*0.94*

Table A.1: *Error statistics. (PE: positioning error; I: results without map; II: results with map)*

Appendix B

Satellite raw data acquisition

B.1 Configuring a u-blox receiver for the data acquisition

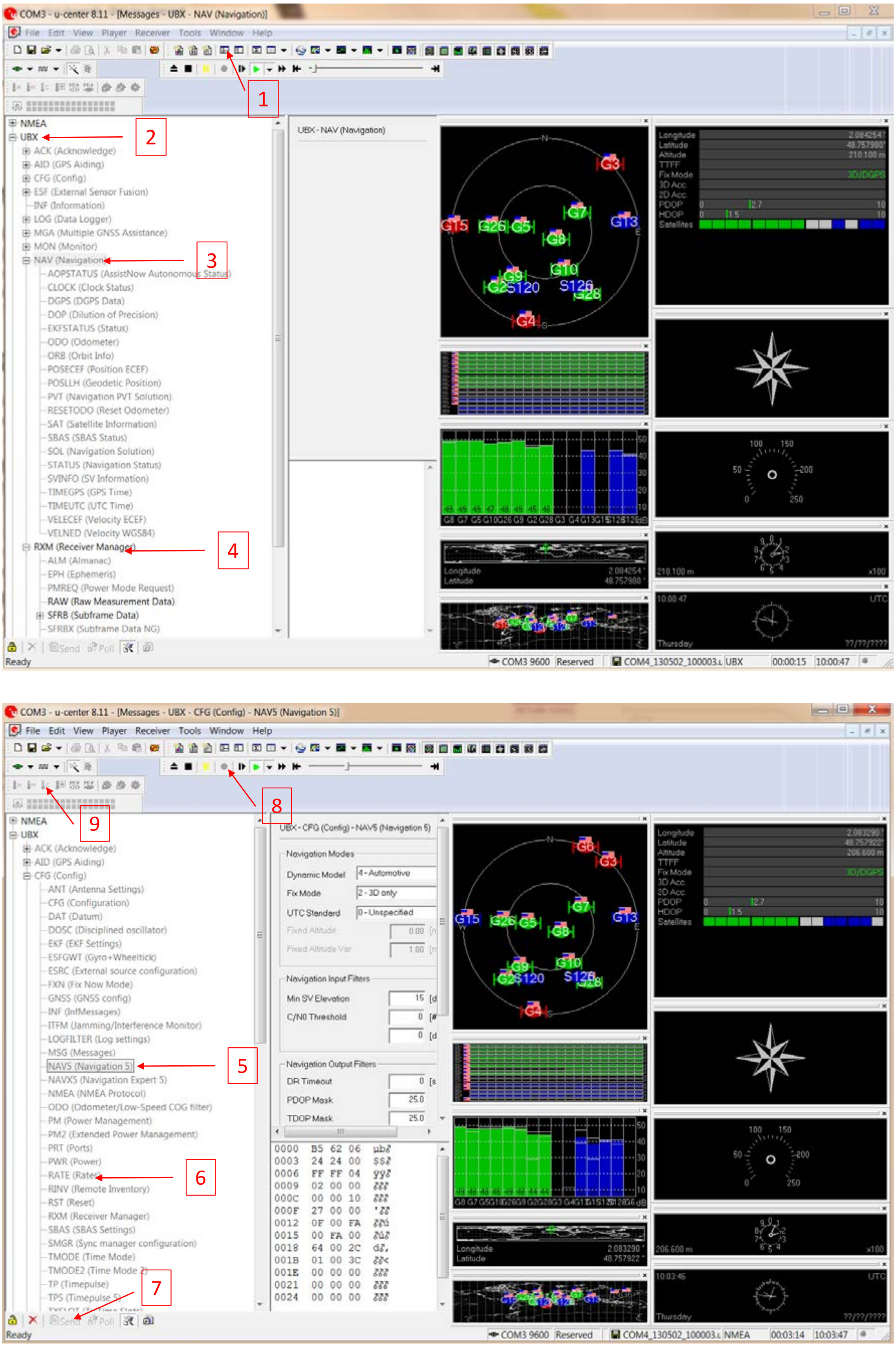
Satellite raw data is available on special receivers with RAW option enabled. Raw data features are available in u-blox 6 LEA-6T. The UBX-RXM-RAW message contains all information needed to be able to generate a RINEX file. RINEX (Receiver INdependent EXchange) is a standard data interchange format for raw satellite navigation system data. It enables storage of pseudorange, carrier phase and Doppler for GPS.

The data is recorded using the UCenter software. Once the u-blox receiver is connected, the configuration follows the steps as shown in Figure B.1:

1. open the “message view” window
2. expand “ubx”
3. right click on “nav” then click on “enable child messages”
4. right click on “rxm” then click on “enable child messages”
5. verify the navigation mode in “cfg/nav5”
6. verify the frequency in “cfg/rate”
7. click on “send” after the modifications
8. click on “record” to start the data acquisition.

B.2 Extraction of satellite raw data

In this thesis, the localization problem is studied in a local ENU frame, therefore, the u-blox data need to be transformed into satellite raw data in ENU frame. This task is divided into three steps: extraction of RINEX observation data, acquisitions of RINEX navigation data, generation of satellite raw data in ENU frame. The process is described briefly as follows.



UBX - CFG (Config) - NAV5 (Navigation 5)

Navigation Modes
Dynamic Model: 4 - Automotive
Fix Mode: 2 - 3D only
UTC Standard: 0 - Unspecified
Fixed Altitude: 0.00
Fixed Altitude Var: 1.00

Navigation Input Filters
Min SV Elevation: 15
C/N0 Threshold: 0
0

Navigation Output Filters
DR Timeout: 0
PDOP Mask: 25.0
TDOP Mask: 25.0

```
0000  B5 62 06  µbƻ
0003  24 24 00  $$ƻ
0006  FF FF 04  ÿÿƻ
0009  02 00 00  ƻƻƻ
000C  00 00 10  ƻƻƻ
000F  27 00 00  'ƻƻ
0012  0F 00 FA  ƻƻú
0015  00 FA 00  ƻúƻ
0018  64 00 2C  dƻ,
001B  01 00 3C  ƻƻ<
001E  00 00 00  ƻƻƻ
0021  00 00 00  ƻƻƻ
0024  00 00 00  ƻƻƻ
```

Figure B.1: *Configuring u-blox receiver with UCenter*

RINEX observation data TEQC (Translation, Editing and Quality Checking) is a software run by command line in DOS/Windows DOS environment:

```
teqc -ublox ubx COM4_130502_100003.ubx > observation.o
```

where *-ublox* means the GPS receiver is u-blox. *ubx* specifies the file format of u-blox. *COM4_130502_100003.ubx* is the experimental ubx file recorded by u-blox with its Ucenter software. The result is a RINEX observation file *observation.o.*

RINEX navigation data When performing the u-blox data acquisition, it is more practical to start by a cold start of the receiver (see Figure B.1, click where the number 9 refers). In this way, the receiver records the navigation data as soon as it is received. The navigation data is not provided anymore when the receiver has calculated its first position. If cold start was not made, navigation files can be recovered in another way to interpret the raw data. They are freely available on the website of *Réseau GNSS Permanent* of the IGN France:

```
ftp ://rgpdata.ign.fr//pub/data/.
```

The navigation file is named by *sitejjjs.aac.Z. site* is the acronym for the station. *jjj* is the number of the day in the year. *s* represents the session of the day (there are 24 sessions per day appointed from a to x) if the file is hourly. *aa* is the last two digits of the year. *c* represents the constellation (*n* for GPS, g for GLONASS). For *COM4_130502_100003.ubx* which is recorded at 10 O'clock 2 May 2013, the navigation file *smne122j.13n.Z* was downloaded. The principle is to download the closest navigation file in space and time to the u-blox data acquisition.

The nomenclature of the files is detailed at the following address:

```
http ://rgp.ign.fr/DONNEES/type.php.
```

The downloaded file should be decompressed using the Rinexpresso software which is available at the following address:

```
http ://rgp.ign.fr/SERVICES/outils.php.
```

The navigation file *navigation.n* is generated using Rinexpresso.

Satellite raw data in ENU frame Now, all the information is available for generating satellite raw data. The GPS Toolkit (GPSTk) is used. GPSTk is an open source project sponsored by the Applied Research Laboratories of the University of Texas. The GPSTk library provides for the reading and writing of data files in the RINEX format. It also includes the capability to defined custom extended RINEX observation types that are useful for storing intermediate quantities in RINEX-like files. A program *Rinex2ENU* is developed based on GPSTk library. The satellite raw data in ENU frame is generated by running the following command line:

```
Rinex2ENU -n smne122j.13n -o observation.o -p lat0 lon0 alt0 -g
```

where *lat0*, *lon0* and *alt0* are the latitude, longitude and altitude coordinates in the WGS84 frame. The output is a RINEX-like file which contains satellite raw data in ENU frame.

An example is given as follows:

```
1738, 381648.199, 2,
-14398090.074, -16459636.336, 8179541.498,
340.980, -2236.210, -2240.137,
23038673.983, 2408293.179, -2226.400, 45.0,
0.000, 0.000, 0.000, 0.0,
2.00, 128953.646, 0.001,
-5.305, -4.266, 11.705, 6.719
```

Each field successively represents :

- GPS week, second of week, satellite ID,
- east position, north position, up position,
- east velocity, north velocity, up velocity,
- Code pseudorange on L1, Phase measurement on L1, L1 Doppler, C/N_0,
- Code pseudorange on L2, Phase measurement on L2, L2 Doppler (the fields are null here),
- URA, satellite clock bias, satellite clock drift,
- TGD, relativity bias, ionospheric bias and tropospheric bias.