POLITECNICO DI TORINO

Master's Degree in Mechatronic Engineering

Master's Degree Thesis

Vehicle path prediction for safety enhancement of autonomous driving

Supervisors

Prof. Stefano Alberto MALAN

Eng. Daniele BREVI

Eng. Francesca PACELLA

Candidate

Andrea MANCINI

Academic Year 2020/2021

Se non credi in te stesso, nessuno lo farà per te.
KB #8 #24

Abstract

In the last few years the concept of autonomous vehicle has developed significantly and has become one of the major topics of the automotive field. Among all the functions that a self - driving vehicle must have, the ability to correctly interpret the sensors information and exploit them for trajectory prediction is fundamental. This thesis work collocates itself exactly in this framework and it is developed around two main topics. First of all, a research on motion models is conducted in order to establish which one can better describe the analysed scenario. Then, the chosen model, called CTRA model, is simulated with real sensor data with the purpose of obtaining a trajectory of the same shape of the authentic one. The second phase of the project has the goal to simulate a real - time scenario and consists in the combination of the CTRA model and the Unscented Kalman Filter for trajectory prediction purposes. Two different cases of application are examined and compared: the first analysis is conducted from the vehicle point of view, while, in the second case, the trajectory prediction is obtained thanks to the data acquired from a camera located on the side of the road. The achieved results have shown that a more accurate prediction is obtained when the correction done by the filter is based on a greater number of measured variables, but they have also revealed that the assumptions proposed by the utilized model are more suitable for short - term prediction, while they become a burden for longer prediction horizons.

Ringraziamenti

Vorrei esprimere tutta la mia gratitudine al professore Stefano Malan, a Daniele Brevi e a Francesca Pacella, i quali mi hanno dato l'opportunità di svolgere questo progetto e mi hanno sempre dimostrato una grande disponibilità, supporto e pazienza per tutte le fasi del lavoro.

Un ringraziamento speciale va alla mia famiglia, che mi è sempre stata vicina durante questi anni, incoraggiandomi a mettercela tutta e a credere in me stesso.

Grazie a Francesca per aver condiviso con me tutte le esperienze vissute durante questi cinque anni, festeggiando i successi e sostenendoci nei momenti più difficili, oltre che per avermi supportato, ma soprattutto sopportato, in tutte le fasi di questo percorso.

Un grande ringraziamento va ai miei amici Matteo, Valeria, Simone, Elisa, Davide, Letizia, Simone e Andreea, i quali fin da subito mi hanno fatto sentire parte integrante del gruppo e con cui ancora oggi condivido ogni genere di esperienze.

Grazie anche a Simone e Andrea, con i quali sono felice di avere un forte legame sia dentro che fuori dal campo da Basket.

Dedico infine un pensiero anche ai miei nonni, i quali sicuramente sarebbero fieri del percorso intrapreso e dei risultati raggiunti.

Contents

List of Figures

Chapter 1

Introduction

In the last few years the concept of autonomous vehicle has become more and more popular, leaving behind that image of a futuristic product rooted in many people's mind. Nowadays, great progress is being made in this area and several automobiles companies are implementing new autonomous features in their more recent models. Anyway, the path to reach a fully autonomous vehicle is still long and characterized by a very high complexity, since a large variety of technologies coming from several disciplines as computer science, mechanical engineering, electronics engineering, and many more must converge in this type of product. In this first chapter, an overview on the world of autonomous vehicles is presented, where their classification and main features are described. After that, the context and scenario analysed in this project are introduced, defining the major actors that operate in the studied situation. At the end of this section, the challenge and goals of the thesis are specified, followed by a brief description of the structure and topics treated in the chapters that compose this work.

1.1 Autonomous vehicles overview

The definition of autonomous vehicle describes it as a self - driving vehicle that has the capability to perceive the surrounding environment and navigate itself without human intervention [6]. Actually, this represents only the final objective of a process started in the last decades.
The current framework in this field is well depicted by SAE (Society of Automotive Engineers), which has defined six levels of driving automation, spanning from 0 (fully manual) to 5 (fully autonomous). The analysis reported in the NHTSA (National Highway Traffic Safety Administration) website [1] describes the most recent version of this classification, called J3016_201806 [20] and shown in Figure 1.1. This can be summarized in the following way:

- **Level 0 - No automation:** the vehicle is manually controlled, so the human driver has to manage all the driving tasks.
- **Level 1 - Driver assistance:** the driver and the automated system share control of the vehicle. An ADAS (Advanced Driving Assistance System) can sometimes assist the driver with actions like steering or braking/accelerating, but not simultaneously.
- **Level 2 - Partial Automation:** the automated system can take full control of the vehicle. An ADAS can actually control both steering and braking/accelerating simultaneously under some circumstances.
- **Level 3 - Conditional Automation:** in some circumstances, an ADS (Automated Driving System) can perform all aspects of the driving task.
- **Level 4 - High Automation:** in this level, an ADS can itself perform all driving tasks and monitor the driving environment; essentially, it can do all the driving in certain circumstances.
- **Level 5 - Full Automation:** no human intervention is required at all. An ADS is able to perform all the driving in all circumstances.

One fundamental aspect of the classification just described is that, in the transition from Level 2 to Level 3, the human driver has not to monitor anymore the surrounding environment. This task is now passed to the automated system and, as levels progress, it relieves the driver from the responsibility to intervene when requested.

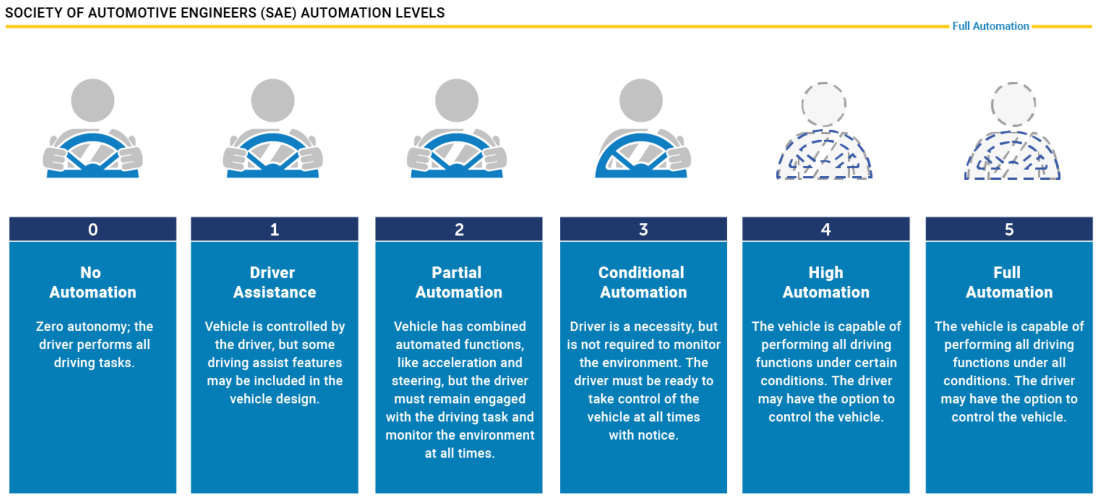


Figure 1.1: SAE Automation Levels [1]

The main capabilities that distinguish a self - driving car, as reported in [3], are the ability to sense its local environment, detect and classify different kinds of objects and interpret sensory information in order to identify appropriate navigation paths whilst obeying transportation rules. Keeping in mind these abilities, the structure of this type of vehicles can be represented by many building blocks [27], as shown in Figure 1.2, distributed in the Hardware side and the Software one.

Starting from the Hardware part, one fundamental component is represented by the sensors mounted on the vehicle, which permit to accumulate raw information about the surrounding environment.
Another peculiarity is the V2X communication, that enables the autonomous vehicle to share and receive information from other vehicles (V2V, Vehicle to Vehicle) or from a particular infrastructure (V2I, Vehicle to Infrastructure).
The last component of the Hardware side is represented by the actuators, which are responsible for controlling and moving the system, and so permit to generate actions like braking, steering, accelerating, and many more.

The Software part of the vehicle, instead, is composed by three different control systems, which covers the phases of Perception, Planning and Control.
The Perception system refers to the capacity to understand the raw information coming from the sensors and the V2X network, while the Planning phase consists in the processing of that information in order to make certain decisions and achieve some higher order goals.
The last block of the Software part of the vehicle is composed by the Control system.

It is responsible for converting the intentions and decisions taken in the previous phase into actions, which will be transformed and transmitted to the actuators in the form of inputs that will lead to the desired motions.

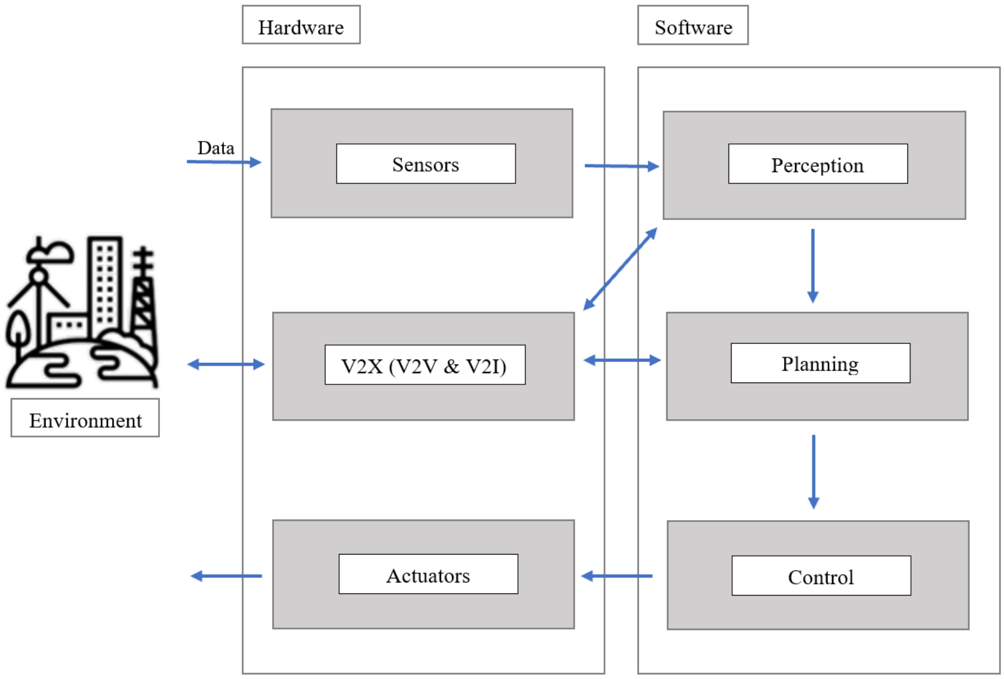


Figure 1.2: Autonomous vehicles building blocks

1.2 Context and scenario

This thesis work is built around the study of one frequent situation while driving, that is the crossing of a road intersection. This scenario is analysed from two different points of view, as showed in Figure 1.3, which are related to the vehicle and to an external camera that is able to monitor the situation in the road intersection.

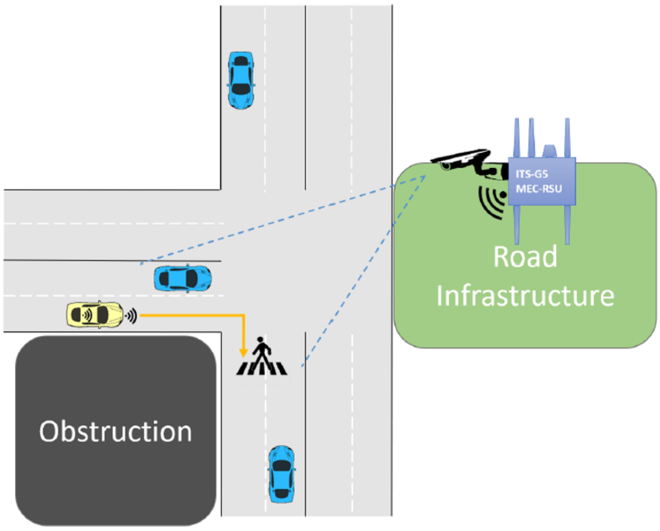


Figure 1.3: Road intersection

In order to introduce an autonomous and connected vehicle within the current automotive network, it is fundamental to be sure of its ability to understand the surrounding environment and calculate the best trajectory that permits to travel in the safest way. This thesis work, which is born from an idea proposed by LINKS Foundation, collocates itself exactly in this perspective and it is strictly related to the Perception and Planning phases described before, which allow the vehicle to receive and understand all the information coming from different sensors, leading to their processing and to the decision of what sort of action the vehicle should perform.

The analysed connected vehicle must be able to correctly monitor its own trajectory and the behaviour of the different variables described by its serial communication system, called CAN bus. Moreover, it must have the ability to adequately predict the trajectory to achieve in the next moments. However, in a public road, this particular capability turns out to be incomplete if not supported by the vision of what is happening around the self - driving vehicle. This knowledge can be provided by a road infrastructure, a camera in this case, which is able to monitor the overall situation in the analysed road intersection and provide data about longitudinal and latitudinal positions of the non - connected vehicles through the analysis of the captured images. Thanks to these information, also their trajectory can be predicted and then compared with the one of the connected vehicle. The knowledge of the future trajectory of the connected vehicle, in fact, represents a huge help, since it permits to anticipate its movements, but it must be completed by the comparison with the non - connected vehicles trajectories in order to correct in time, if necessary, the predicted movements and avoid collisions.

In this context, one of the elements introduced in the previous section assumes a particular importance. The V2X communication, in fact, covers a fundamental role in scenarios like the one analysed in this project, since permits to handle the exchange of information between the vehicle and the infrastructure. This mechanism is based on the distribution and reception of the so - called CAM (Cooperative Awareness Messages), which share information about the status and the characteristics of the generating station, and so details about time, position, vehicle type and role, and many more [2]. The CAM messages are produced periodically, and so are characterized by a precise sampling frequency. As explained in [10], one of the objectives of the V2X communication consists in keeping the vehicle up to date about all the nearby subjects that share the road, and therefore the validity and recentness of the CAM messages assume a fundamental importance.

1.3 Challenge and goals of the thesis

As introduced in the previous section, the main goals of this project are two: the first one consists in representing the autonomous and connected vehicle through a suitable mathematical model, able to efficiently exploit the sensor data in order to reproduce the trajectory as close as possible to the real one; the second one, instead, is focused on the application of the chosen model to a trajectory prediction technique, which permit to calculate the future states and movements of the vehicle before they are actually carried out.

For these reasons, the report is structured in the following way:

- ***Chapter 2 - Vehicle model:*** first of all, a state of the art concerning the chosen vehicle model is presented. This is followed by the explanation of the model implementation phase in the simulation program. At the end of the chapter, the results of the replication of the real trajectory are illustrated.
- ***Chapter 3 - Vehicle path prediction:*** the structure of this chapter is similar to that of the previous one. The state of the art of the used path prediction technique is followed by its implementation, and the final results provided by the combination of the vehicle mathematical model and the chosen technique are presented.
- ***Chapter 4 - Conclusions:*** in the last chapter the overall results are commented and the conclusions are summed up. At the end some ideas for possible future projects are proposed.

Chapter 2

Vehicle model

The second chapter of this project begins with a brief outline concerning the model analyzed before the final one, supplemented by the motivations that led to a different choice. After that, a state of the art regarding the motion models group and the final vehicle mathematical model is presented. Here, the assumptions at the base are introduced, followed by the description of the selected state space variables and by the equations that permit to perform a simulation of the trajectory. Then, the implementation phase is inserted, in which the preparatory work done on data in order to obtain a set concordant both with the state variables of the model and the simulation program is described. In the same phase, the mathematical translation and integration of the model into the simulation program are presented, followed by the last phase, in which the results of the several tests carried out in order to see if the model could simulate a trajectory with a shape similar to the real one are showed.

2.1 Single - track model

At the beginning of this thesis work the intention was to simulate the real vehicle behavior through the single - track model, or bicycle model, which is represented in Figure 2.1. This model assumes that the vehicle is able to move only in the horizontal plane, which means that the vertical dynamics is neglected, and it is composed by two main parts: the chassis, which includes also the rear wheel, and the steering system, composed only by the front wheel [26].

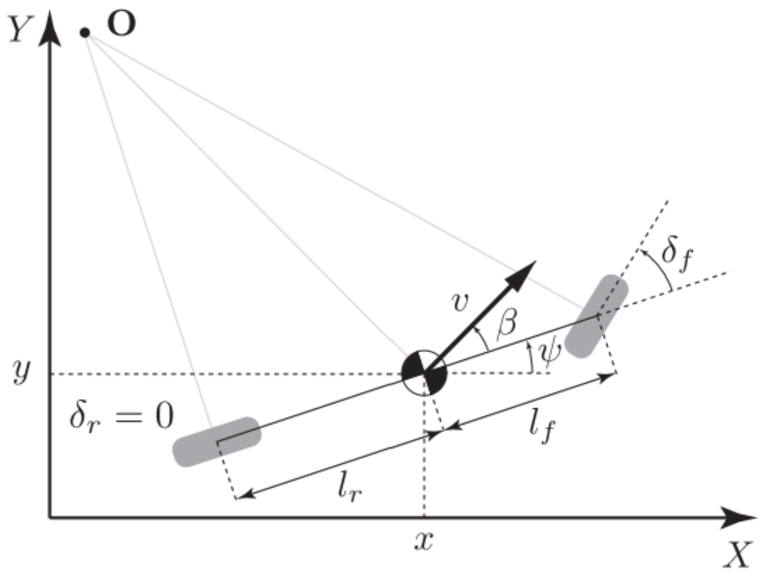


Figure 2.1: Single - track model [7]

A first research was conducted in order to understand which configuration of the bicycle model, between the kinematic and the dynamic one, could behave more efficiently for this application case.

The kinematic bicycle model, which does not take into account the forces that act on the vehicle, is described by the following set of nonlinear continuous time equations [8]:

$$\dot{x} = v\cos(\psi + \beta) \tag{2.1}$$

$$\dot{y} = v\sin(\psi + \beta) \tag{2.2}$$

$$\dot{\psi} = \frac{v}{l_r}\sin(\beta) \tag{2.3}$$

$$\dot{v} = a \tag{2.4}$$

$$\beta = \tan^{-1}\left(\frac{l_r}{l_f + l_r}\tan(\delta_f)\right) \tag{2.5}$$

where x and y represent the coordinates of the center of mass in an inertial frame (X,Y), while the physical lengths of the vehicle are taken into account by l_f and l_r, which represent the distances from the center of mass to the front and rear axles. The angles that show up in the equations from (2.1) to (2.5) are β, ψ, δ_f, which represent respectively the angle between the velocity of the center of mass and the longitudinal axis of the car, the inertial heading and the front steering angle. Finally, a represents the acceleration of the vehicle.

One of the major assumptions of this configuration, as reported in [22], is that the velocity vectors lie exactly in the direction of orientation of the front and rear wheels, which is equivalent to assume that the slip angles at both wheels are zero. For this reason, this simplification is reasonable only for low speed motion of the vehicle (speed less than 5 m/s, which is approximately 18 km/h) or, as proven in [21], with a lateral acceleration a_y limited at values lower than $0.5\mu g$, since, in this situations, the lateral force generated by the tires is small.
At higher vehicle speeds, this assumption can no longer be made: in this case, instead of a kinematic single - track model, a dynamic model for lateral vehicle motion must be developed.

The paradigm adopted for the description of a dynamic bicycle model, characterized by two degrees of freedom y and Ψ, is represented in Figure 2.2. The vehicle lateral position y can be identified along the lateral axis of the car to the center of rotation O, while the heading angle Ψ (in Figure 2.1 ψ was utilized for the heading angle) is measured with respect to the global X axis.

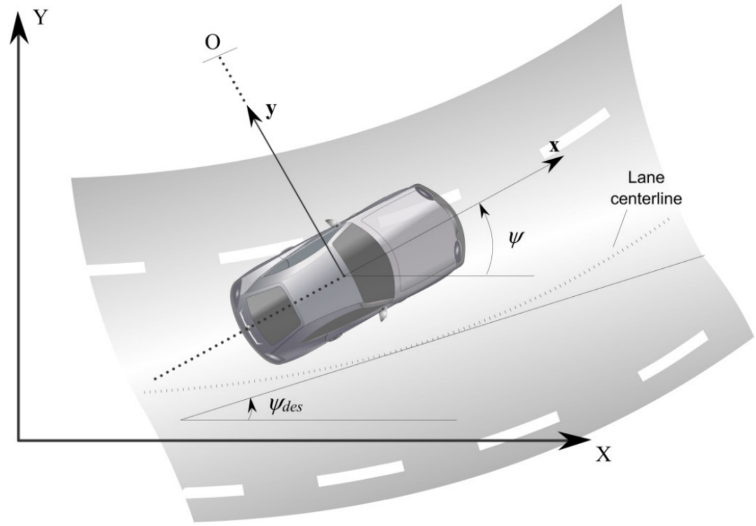


Figure 2.2: Lateral vehicle dynamics [22]

In this case, the differential equations for the dynamic single - track model are the relations from (2.6) to (2.10):

$$\ddot{x} = \dot{\Psi}\dot{y} + a_x \tag{2.6}$$

$$\ddot{y} = -\dot{\Psi}\dot{x} + \frac{2}{m}\left(F_{c,f}\cos\delta_f + F_{c,r}\right) \tag{2.7}$$

$$\ddot{\Psi} = \frac{2}{I_z}\left(l_f F_{c,f} - l_r F_{c,r}\right) \tag{2.8}$$

$$\dot{X} = \dot{x}\cos\Psi - \dot{y}\sin\Psi \tag{2.9}$$

$$\dot{Y} = \dot{x}\sin\Psi + \dot{y}\cos\Psi \tag{2.10}$$

where $\dot{x}$ and $\dot{y}$ represent the longitudinal and lateral speeds, while $\dot{\Psi}$ stands for the yaw rate. The parameters m and I_z denote the vehicle mass and the yaw inertia. The variables $\dot{X}$ and $\dot{Y}$, instead, indicate the vehicle speed with respect to the global axes. Finally, $F_{c,f}$ and $F_{c,r}$ respectively describe the lateral tire forces at the front and rear wheels.

Considering the previous equations of the dynamic model, it would therefore seem advisable to use this configuration, but there are some disadvantages that, instead, orient the choice to the kinematic model. In fact, the dynamic model behaves better at higher speeds, but it is characterized by a very high computational effort.

Moreover, usually this second configuration is utilized with tire models, which have the problem to become singular at low speeds.

For these reasons, the choice initially went towards the kinematic single - track model, since it requires less computational power and, as demonstrated in [7], can be implemented at a wide range of vehicle speeds (better at low speeds), including also zero speed, which is a recurrent situation in use case scenarios like the road intersection analysed in this project.

Anyway, even if the kinematic model can behave in a proper manner in a situation like the one analysed, it was decided not to pursue this idea in the continuation of the project. This decision was taken due to the high presence of parameters strongly dependant on the vehicle type and model considered. Factors like l_f, l_r, m, I_z and many more do not allow to obtain a simulation extendable to as many vehicles as possible, which is one of the major goals of this thesis.

This project aims to realize a scheme capable of representing a situation no matter the actor who plays it, so as to be able to apply it not only to a specified vehicle. For this number of reasons, this project is build following another model, which can be applied to several realities, not being linked to specific factors of the vehicle that faces these situations.

2.2 Motion models

In order to extend the analysis and build an application case independent from the vehicle type, another mathematical model instead of the single - track one has been chosen. This thesis work is then based on the performances and characteristics of the so - called CTRA model (Constant Turn Rate and Acceleration model), which belongs to the group of the physics - based motion models.

2.2.1 Classification

The motion models are mainly divided into three levels, showed in Figure 2.3. As well described in [9], these are characterized by the following aspects:

- **Physics - based motion models** describe the vehicles as dynamic entities dependant on the laws of physics. The prediction of the future movement of the vehicle is conducted using kinematic and dynamic models characterized by some control inputs, car properties and external conditions, which permit to calculate the evolution of the main state variables of the vehicle.

- **Maneuver - based motion models** consider that the future movements of the vehicle strongly depends on the maneuver that the driver plans to carry out. They represent vehicles as independent maneuvering entities, which means that they assume that the actions performed by one entity correspond to a series of known maneuvers independent from the route taken by the other vehicles. In practice, this strong assumption of independence between vehicles fatigue to remain valid, since the analysed subject share the road with other actors and the maneuvers performed by one vehicle necessarily influence the path of the others.

- **Interaction - aware motion models** take into account the several interactions between vehicles. They develop the analysis conducted by the maneuver - based motion models and improve it, since they represent vehicles as entities that interact with each other, and so the movements and the taken path of the analysed subject is influenced by the motion of the other vehicles on the road.

This classification shows that the interaction - aware motion models are the most complete level of description, since they allow a more complete analysis with respect to the ones conducted for the physics - based and maneuver-based motion models. However, also this third classification is characterized by some drawbacks, among

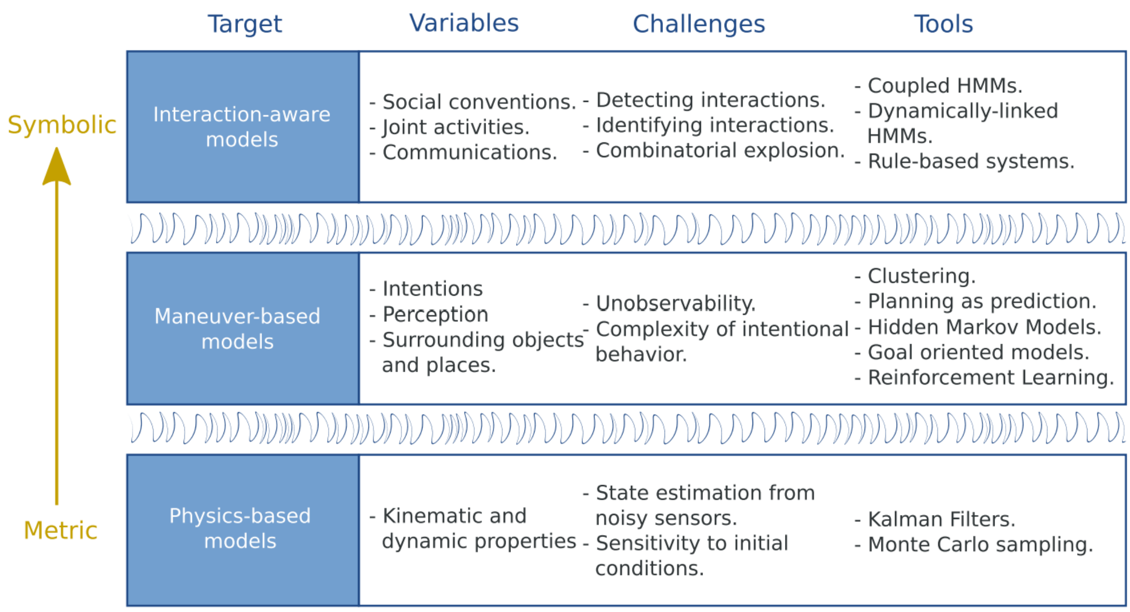


Figure 2.3: Motion modeling overview [9]

which the most relevant is the expensive computation effort. This aspect makes the interaction - aware models not compatible with real - time prediction and risk assessment.

It was then decided to build this thesis project around a physics - based motion model since, as said before, the goal is to work on an application independent not only from the vehicle type, but also from the path performed by the analysed subject, which also excludes the choice of maneuver - based models.

This means that with this work it has been tried to create a model able to adapt to many possible situations, being able to work and simulate a real - time trajectory working only on sensors information and by exploiting the equations of which the chosen model is composed.

2.2.2 Physics - based motion models

The motion models that belong to this group are numerous. A first classification, as proposed in [23], can be made on the basis of the assumptions that characterize each model and on their level of complexity.

The lower level is occupied by the linear motion models, which have the advantage of the linearity of the state transition function, responsible for the evolution of the state variables. However, their drawback is that they consider always a straight motion, and so are not able to take into account motions like rotations.

1) One of the main linear models is the **CV** model, which assume **Constant Velocity** of the vehicle. It is described by the following state space:

$$\vec{x}(t) = (x \;\; v_x \;\; y \;\; v_y)^T \tag{2.11}$$

where x and y represent the longitudinal and latitudinal positions of the vehicle, while v_x and v_y are the speed components on the x and y axes.

The linear state transition function of the CV model is given by:

$$\vec{x}(t+T) = A(t+T)\vec{x}(t) \tag{2.12}$$

where T represents the sample time.

In order to be used with the framework of the trajectory prediction technique analysed in the following chapter, the previous state transition function can be modified in the following way:

$$\vec{x}(t+T) = \begin{pmatrix} x(t) + Tv_x \\ v_x \\ y(t) + Tv_y \\ v_y \end{pmatrix} \tag{2.13}$$

Increasing the complexity of the models, also rotations around the z - axis must be considered. This is the case of the curvilinear models, which can be further classified on the basis of which state variables are assumed to be constant.

2) The simplest model of this level is the so - called **CTRV** model, which assumes **Constant Turn Rate** and **Velocity**. It is described by the following state space variables:

$$\vec{x}(t) = (x \;\; y \;\; \theta \;\; v \;\; w)^T \tag{2.14}$$

where θ is the heading angle and w represents the yaw rate.

The evolution of the state is managed by a non - linear state transition function:

$$\vec{x}(t+T) = \begin{pmatrix} \frac{v}{w}sin(wT+\theta) - \frac{v}{w}sin(\theta) + x(t) \\ -\frac{v}{w}cos(wT+\theta) + \frac{v}{w}sin(\theta) + y(t) \\ wT+\theta \\ v \\ w \end{pmatrix} \tag{2.15}$$

3) By deriving the velocity and considering also the acceleration as a state variable, as reported in [29], the **CTRA** model can be obtained, which assume **Constant Turn Rate** and **Acceleration**. This particular model expands the previous one's state space:

$$\vec{x}(t) = (x \;\; y \;\; \theta \;\; v \;\; a \;\; w)^T \tag{2.16}$$

Now, the state transition function is given by:

$$\vec{x}(t+T) = \begin{pmatrix} x(t+T) \\ y(t+T) \\ \theta(t+T) \\ v(t+T) \\ a \\ w \end{pmatrix} = \vec{x}(t) + \begin{pmatrix} \Delta x(T) \\ \Delta y(T) \\ wT \\ aT \\ 0 \\ 0 \end{pmatrix} \tag{2.17}$$

where:

$$\begin{aligned} \Delta x(T) =& \frac{1}{w^2}[(v(t)w + awT)sin(\theta(t)+wT) + acos(\theta(t)+wT) \\ & -v(t)wsin\theta(t) - acos\theta(t)] \end{aligned} \tag{2.18}$$

and

$$\begin{aligned} \Delta y(T) =& \frac{1}{w^2}[(-v(t)w - awT)cos(\theta(t)+wT) + asin(\theta(t)+wT) \\ & +v(t)wcos\theta(t) - asin\theta(t)] \end{aligned} \tag{2.19}$$

4) Finally, the **CCA** model is presented, which assume **Constant Curvature** and **Acceleration**. Its state space is very similar to the one of the CTRA model:

$$\vec{x}(t) = (x \;\; y \;\; \theta \;\; v \;\; a \;\; c)^T \tag{2.20}$$

with the exception that the yaw rate w is substituted by the curvature $c = R^{-1}$, where R is the radius of the path that the vehicle is travelling.

Since the radius R is equal to:

$$R = \frac{1}{c} = -\frac{v(t)}{w(t)} = const. \tag{2.21}$$

and the speed v is given by:

$$v(t) = v(t_0) - at \tag{2.22}$$

the expression of the yaw rate w in this case becomes:

$$w(t) = (-v(t_0) - at)c \tag{2.23}$$

Knowing that the continuous system that characterizes this model is specified by:

$$\dot{\vec{x}} = \begin{pmatrix} v(t)cos(w(t)t + \theta(t_0)) \\ v(t)sin(w(t)t + \theta(t_0)) \\ w(t)t \\ a \\ 0 \\ 0 \end{pmatrix} \tag{2.24}$$

and inserting the equations (2.22) and (2.23), it becomes:

$$\dot{\vec{x}} = \begin{pmatrix} (v_0 + at)cos((-v_0 - at)ct + \theta_0) \\ (v_0 + at)sin((-v_0 - at)ct + \theta_0) \\ (-v_0 - at)c \\ a \\ 0 \\ 0 \end{pmatrix} \tag{2.25}$$

In order to obtain the discrete state transition function as for the previous models, the continuous expression must be integrated:

$$\vec{x}(t+T) = \int_t^{t+T} \dot{\vec{x}}(t)dt + \vec{x}(t) \tag{2.26}$$

Anyway, carry out this step lead to have a very complex state transition function, which, as will be said in the next section, is one of the reasons why the usage of the CCA model is not recommended.

2.2.3 Choice of the motion model

In order to choose the most suitable model for the use case of this project, the analysis and the considerations formulated in [23] and [24] are considered. These surveys compare all the previous motion models in a trajectory tracking application, conducted both in urban and in a highway scenario.

It is demonstrated that the most complex curvilinear models as CTRV and CTRA perform better than the simple CV linear model in every case, since they produce lower lateral and longitudinal errors with respect to the real trajectory of the vehicle. Moreover, the introduction of the acceleration as state variable in the CTRA model permits to obtain better results than the CTRV model, additionally enhancing the overall tracking result. Especially in situations where the acceleration grows and overcomes the limit of 0.5 m/s^2, the CTRA model performs better than the CTRV, which produces large position errors.
No particular difference can be found in the results proposed by the CTRA and the CCA models. However, due to the very high calculation effort that characterizes the second model, the usage of the CTRA model is recommended.

For this number of reasons, in this thesis project it has been chosen to study the CTRA model, whose implementation in the simulation program is showed and explained in the following sections.

2.3 Preparatory work on data

After finding that the CTRA model is the most suitable for this type of application, the next phase consists in implementing it into the simulation program, which is MATLAB. However, before this step, a preparatory work on the provided data must be done, in order to obtain a set of values concordant both with the state space of the model and the simulation program.

2.3.1 Selection of data concerning the use case

This thesis work is based on the set of data provided by LINKS Foundation, which are the data available from the CAN bus communication system of the analysed vehicle. These values represent and describe the evolution of the main variables during the path followed by the vehicle, which is shown in Figure 2.4:

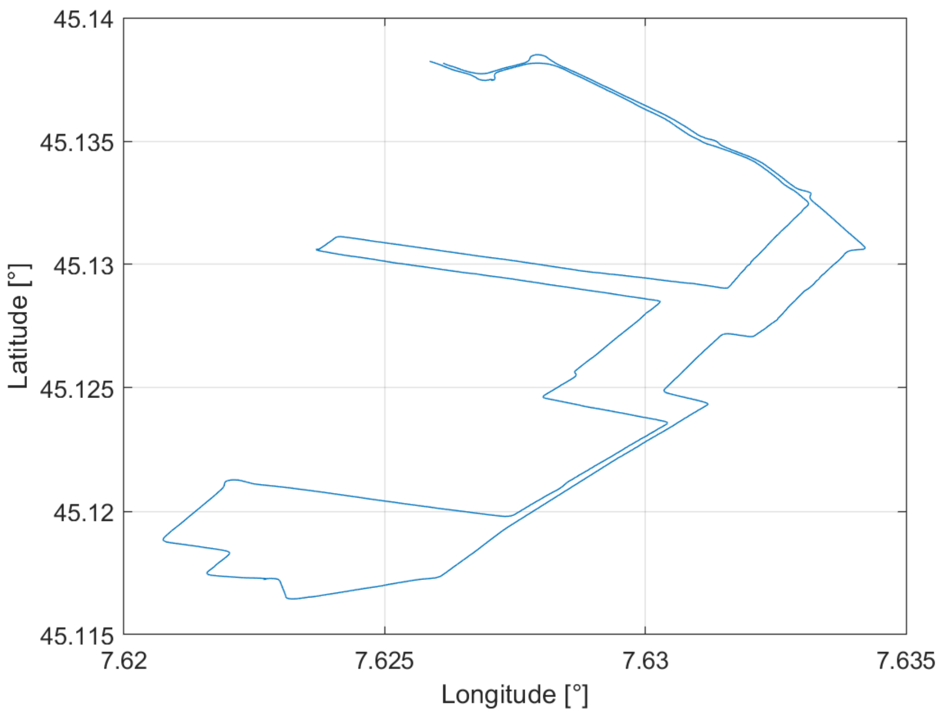


Figure 2.4: Path followed by the vehicle

In fact, from the CAN bus, it is possible to receive information about:

- **Timestamp** $[ms]$: indicates the instant when the values of the observed variables are measured. In this set the interval between one measure and the next one is approximately 0.25 s.

- **Latitude** [deg, °] and **Longitude** [deg, °]: are the geographical coordinates that permit to univocally identify the position of the vehicle on the Earth's surface.
- **Heading** [deg, °]: represents the angle of orientation of the vehicle. It is defined with origin to the true North and the rotation is positive if clockwise.
- **Yaw Rate** [deg/s, °/s]: it is defined as the angular velocity during a rotation, or also as the rate of change of the heading angle.
- **Speed** [km/h]
- **Steering angle** [deg, °]: it is the angle between the front of the vehicle and the steered wheel direction. The rotation to the right is defined positive.

As seen in Chapter 1.3, this project is built around the study of the path taken by the vehicle in a road intersection. For this reason, from the overall data that reproduce the trajectory showed in Figure 2.4, the subset highlighted in Figure 2.5 is selected:

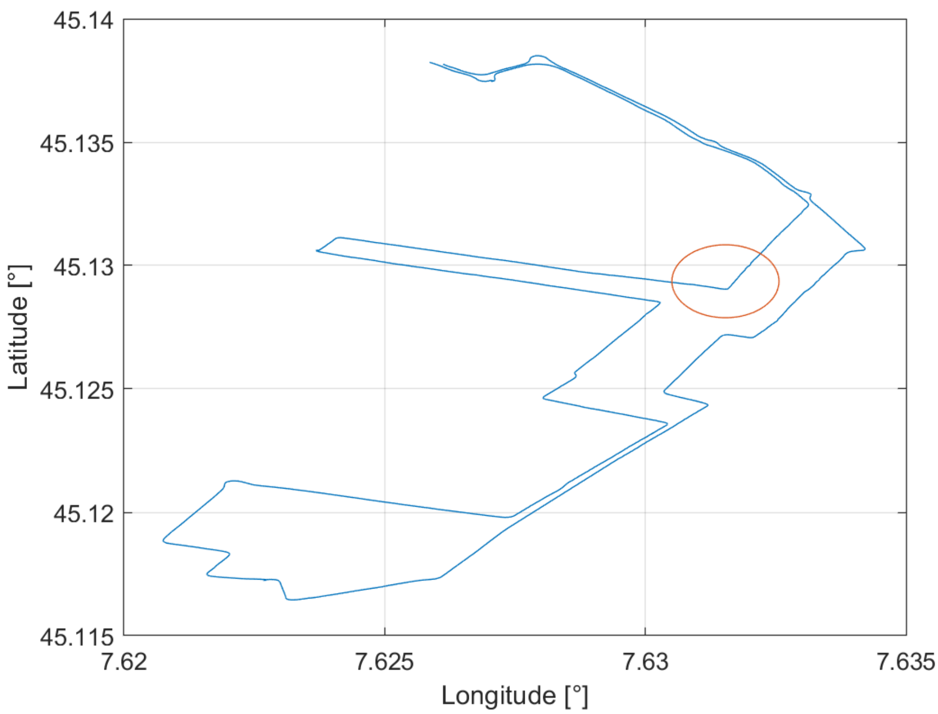


Figure 2.5: Selected path

and the scenario represented in Figures 2.6 and 2.7 is analyzed:

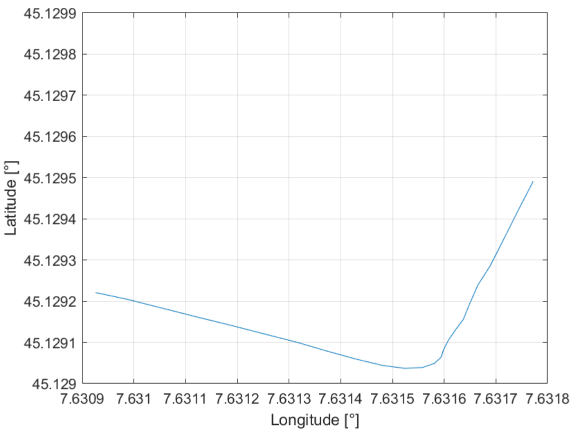


Figure 2.6: Road intersection

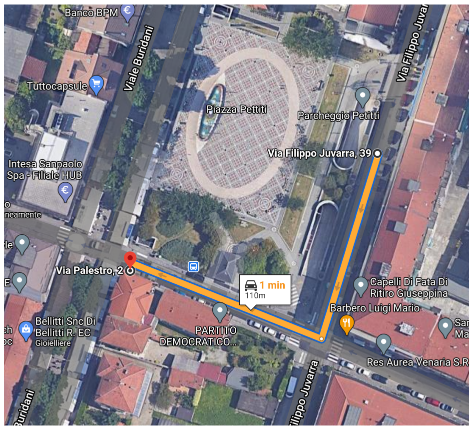


Figure 2.7: Real scenario

where the vehicle runs the road intersection from the right to the left of the figures.

2.3.2 Transition to SI units

As written before, in order to have a set of variables concordant with the chosen equations and with the simulation program, before moving on to the implementation of the model, the first thing to do is to transform the selected data into SI units. From the CAN bus of the analysed vehicle five of the six state space variables of the CTRA model are available, and these are: longitudinal position, latitudinal position, speed, heading and yaw rate.

The first change, however, is made on the timestamp data. The overall selected data are transformed from ms to s, and then an offset equal to the first value of the set is subtracted. In this way, a set of data that exactly starts from 0 s and ends at 22.615 s is obtained.

After that, in order to convert the measurement units of the longitudinal and latitudinal positions from degrees to meters, from Figure 2.6 an origin point with coordinates $[7.6309°, 45.129°]$ is chosen, and for all the available data the distance from this point can be calculated. After that, thanks to the *deg2km* MATLAB function [12], all the computed distances are converted into km, and then in m, obtaining the result showed in Figure 2.8:

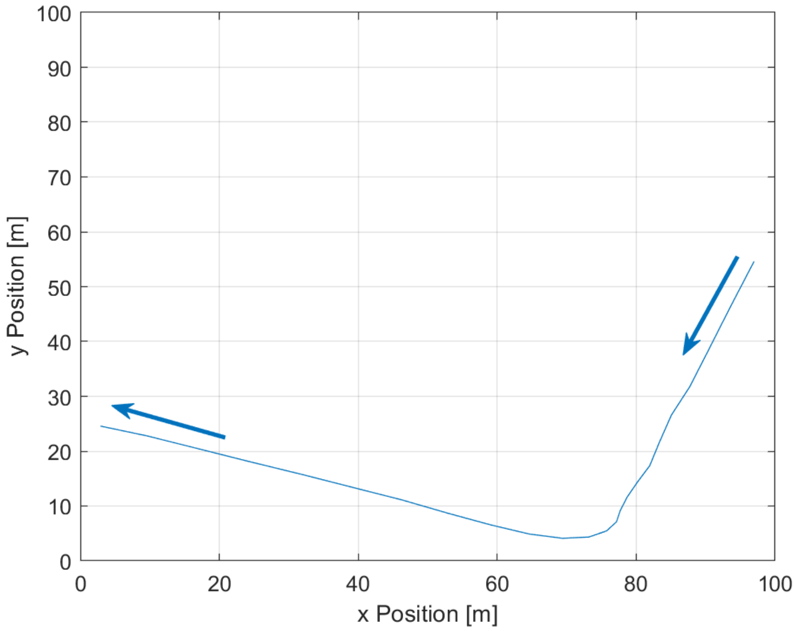


Figure 2.8: Trajectory - Transition to SI

Regarding the speed data of the road intersection, they are already provided into SI units, so they are only transported into m/s, obtaining the set in Figure 2.9:

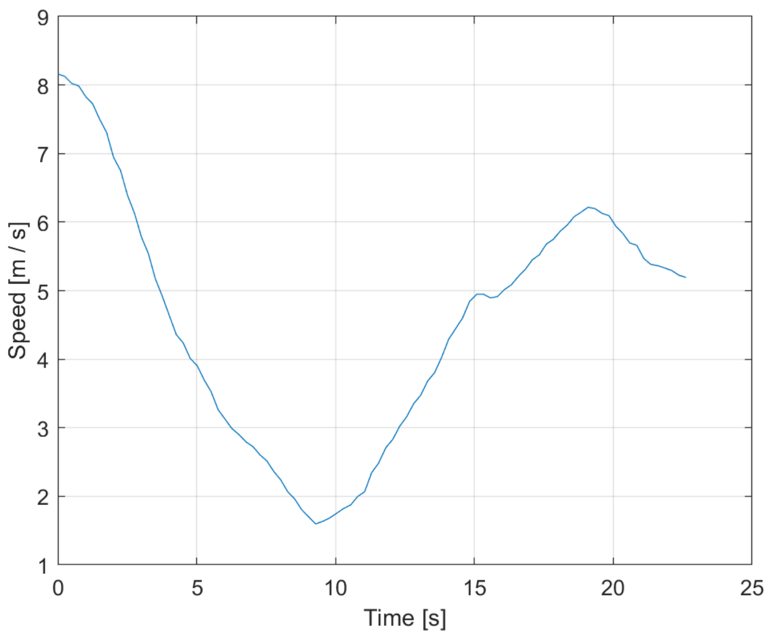


Figure 2.9: Speed data

For the heading data, in order to convert the measurement units from *deg* to *rad*, the *deg2rad* MATLAB function [11] can be used. The result of this operation is showed in Figure 2.11, together with the original data set, inserted in Figure 2.10 to be aware of the path taken by the vehicle:

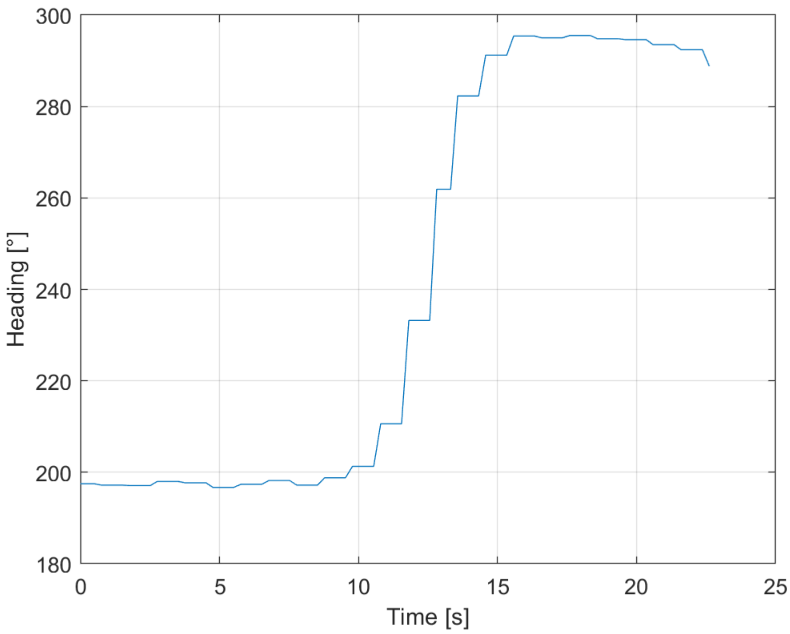


Figure 2.10: Heading data - Original set

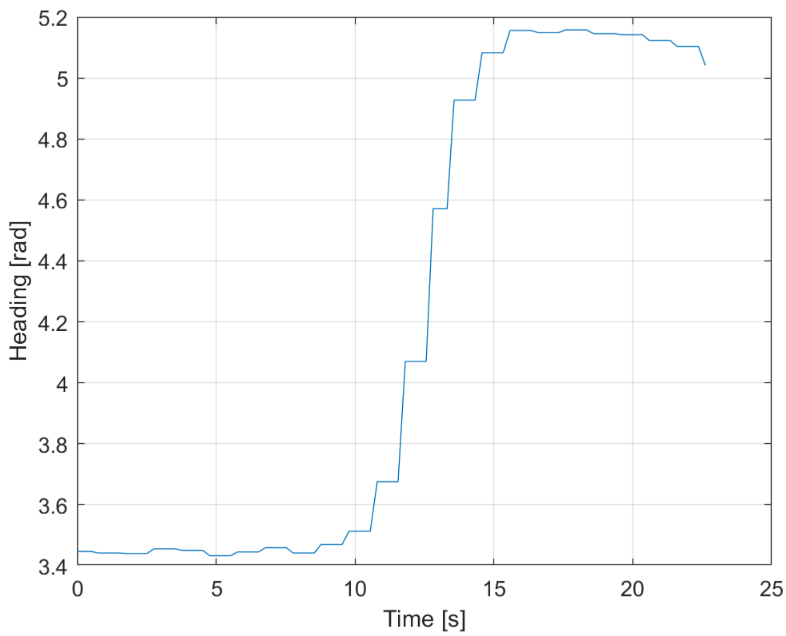


Figure 2.11: Heading data - Transition to SI

The same approach can be followed for the yaw rate data, which must be transformed from deg/s to rad/s. This group of data set shows the behaviour in Figure 2.12 and Figure 2.13:

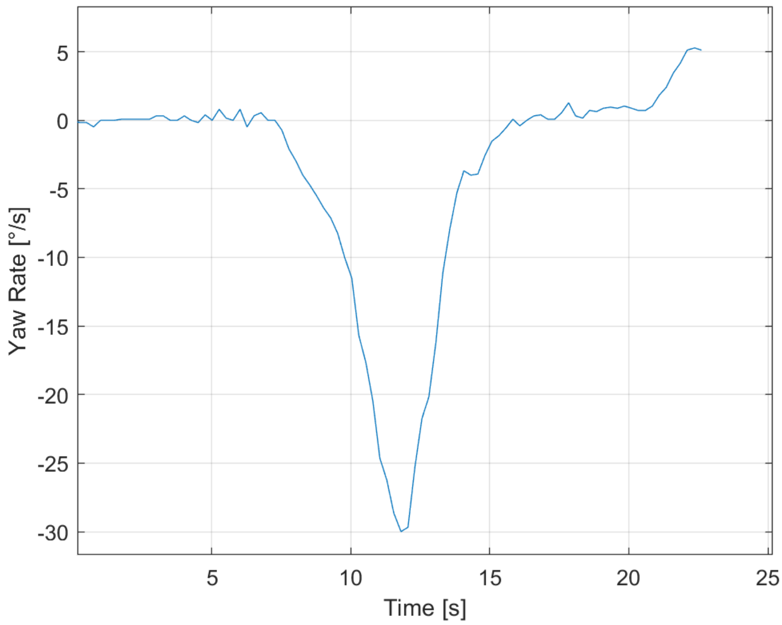


Figure 2.12: Yaw rate data - Original set

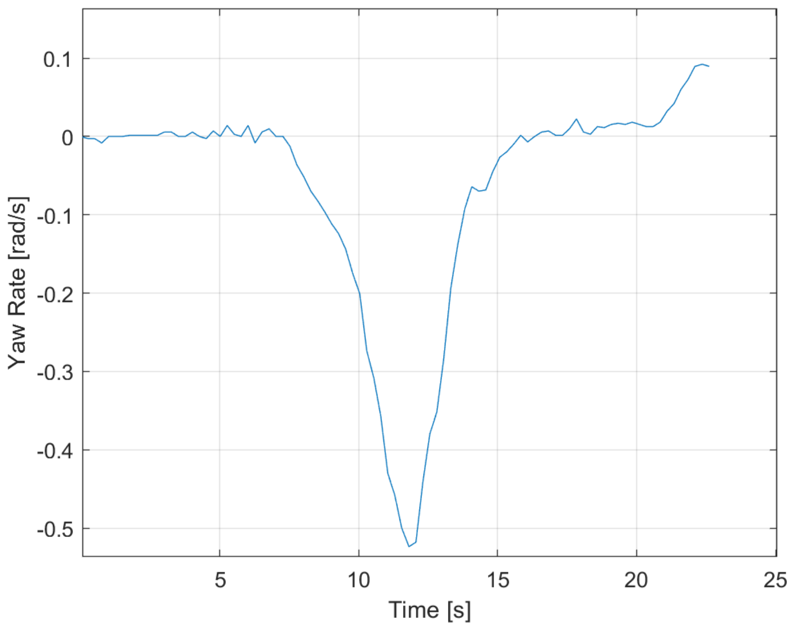


Figure 2.13: Yaw rate data - Transition to SI

2.3.3 Moving average of the acceleration

In the state space of the CTRA model also the acceleration appears, which, however, is not part of the data provided by the CAN bus of the vehicle. For this reason, a data set for this state space variable must be calculated. To do this, two approaches can be followed:

1. **Speed derivative:** the acceleration data set is provided by a cycle that computes, for each iteration, the ratio between the speed variation and the time interval between one measurement and the successive one.

2. **Moving average:** for this approach the *polyfit* MATLAB function is used, which, as explained in the MathWorks documentation [16], permits to find the coefficients of a polynomial of degree N that fits in the best way the provided data. Here, providing the speed and the timestamp data, and building a polynomial of degree $N = 1$, it is possible to find the value of the acceleration imposing it equal to the angular coefficient of the straight line. In order to obtain the trend that better approximate the real acceleration, this approach is followed using a buffer of five and ten values of speed and timestamp data.

The results provided by these two approaches show the behaviour of Figure 2.14:

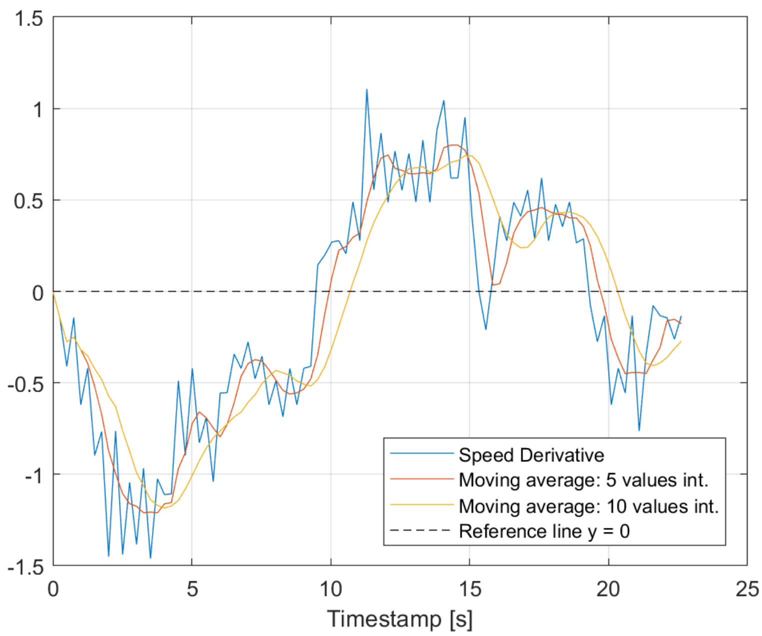
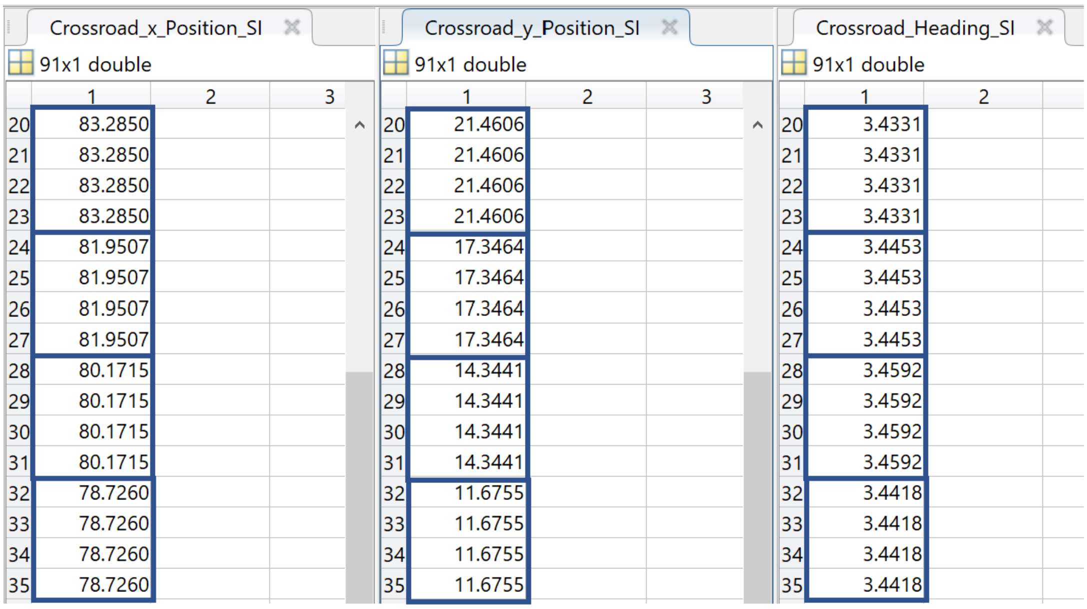


Figure 2.14: Acceleration

Since the behaviour proposed by the speed derivative is characterized by high discontinuities, the moving average method is chosen. Even if this approach shows a sort of delay with respect to the speed derivative method, it presents a smoother behaviour, which is easier to replicate in a real - time scenario. Between the moving average obtained with a buffer of five and ten measurements, the first one is adopted in the used data, since it shows only a short delay and its trend is almost equal to the speed derivative one.

2.3.4 Interpolation of position and heading data

Before moving on to the simulation of the *CTRA* model, another adjustment on the provided data must be done. In fact, during the use of the longitudinal position, latitudinal position and heading data, it was found that these data sets are measured by sensors characterized by a lower sampling frequency with respect to the one of the speed and yaw rate data. To be precise, the highlighted sets are saved with a frequency four times lower than the others, which leads to have blocks of four equal measurements of position and heading, as shown in Figure 2.15, while the values for the speed and yaw rate data change at every measurement.



Crossroad_x_Position_SI — 91x1 double

	1	2	3
20	83.2850		
21	83.2850		
22	83.2850		
23	83.2850		
24	81.9507		
25	81.9507		
26	81.9507		
27	81.9507		
28	80.1715		
29	80.1715		
30	80.1715		
31	80.1715		
32	78.7260		
33	78.7260		
34	78.7260		
35	78.7260		

Crossroad_y_Position_SI — 91x1 double

	1	2	3
20	21.4606		
21	21.4606		
22	21.4606		
23	21.4606		
24	17.3464		
25	17.3464		
26	17.3464		
27	17.3464		
28	14.3441		
29	14.3441		
30	14.3441		
31	14.3441		
32	11.6755		
33	11.6755		
34	11.6755		
35	11.6755		

Crossroad_Heading_SI — 91x1 double

	1	2
20	3.4331	
21	3.4331	
22	3.4331	
23	3.4331	
24	3.4453	
25	3.4453	
26	3.4453	
27	3.4453	
28	3.4592	
29	3.4592	
30	3.4592	
31	3.4592	
32	3.4418	
33	3.4418	
34	3.4418	
35	3.4418	

Figure 2.15: Equal measurements blocks

In order to solve this problem and obtain a group of data with the same sampling frequency, a linear interpolation between the extreme values of each block of the considered sets must be done. To do this, the equation (2.27) is applied to the three sets of data indicated before:

$$y = y_1 + \frac{x - x_1}{x_2 - x_1}(y_2 - y_1) \tag{2.27}$$

where y represents the wanted data (longitudinal position, latitudinal position or heading) and x stands for the utilized timestamp data.

After this procedure, the positions and heading data sets show the behaviours in Figures 2.16 and 2.17:

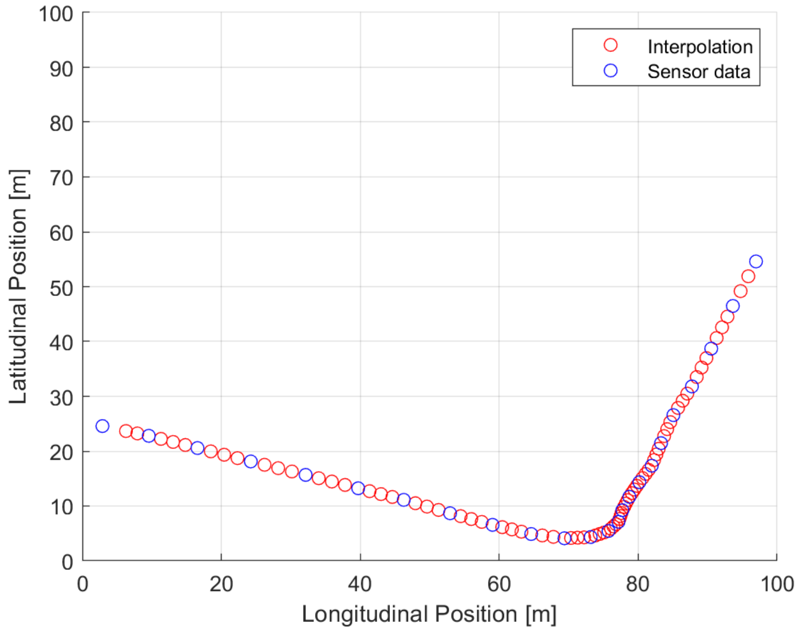


Figure 2.16: Longitudinal and latitudinal positions data interpolation

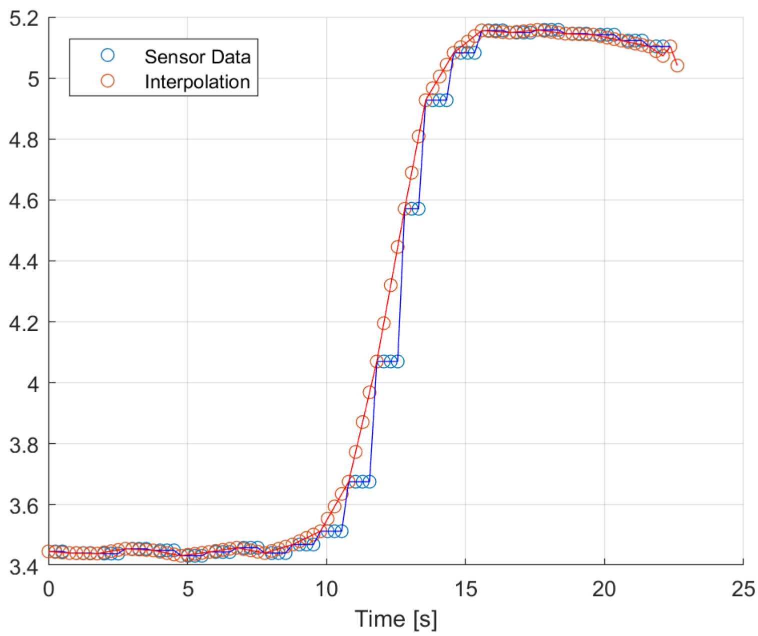


Figure 2.17: Heading data interpolation

2.4 Implementation of the CTRA model

Now that all the data result concordant both with the model and the simulation program, the implementation of the CTRA model can begin.

As explained in Section 2.2.2, the chosen model is characterized by a six - variables state space:

$$\vec{x}(t) = (x \;\; y \;\; v \;\; \theta \;\; w \;\; a)^T \tag{2.28}$$

where the presented state space variables stand for the longitudinal and latitudinal positions, speed, heading angle, yaw rate and acceleration, in this order.

Moreover, the evolution of all the state space variables from the instant k to the instant $k+1$ is guided by some transition equations. Regarding the evolution of longitudinal and latitudinal positions, these have already been explained in equations (2.18) and (2.19), and are here reported for simplicity:

$$\begin{aligned} x_{k+1} = x_k + \Delta x = x_k + \frac{1}{w_k^2}[&(v_k w_k + a_k w_k T) sin(\theta_k + w_k T) + a_k cos(\theta_k + w_k T) \\ &- v_k w_k sin\theta_k - a_k cos\theta_k] \end{aligned} \tag{2.29}$$

$$\begin{aligned} y_{k+1} = y_k + \Delta y = y_k + \frac{1}{w_k^2}[&(-v_k w_k - a_k w_k T) cos(\theta_k + w_k T) + a_k sin(\theta_k + w_k T) \\ &+ \; v_k w_k cos\theta_k - a_k sin\theta_k] \end{aligned} \tag{2.30}$$

The evolution of speed and heading, instead, is dependant on the acceleration and yaw rate behaviour, respectively:

$$v_{k+1} = v_k + a_k T \tag{2.31}$$

$$\theta_{k+1} = \theta_k + w_k T \tag{2.32}$$

The assumption made by the CTRA model regards essentially the yaw rate and the acceleration, which are considered constant. For this reason, their values remain always the same:

$$w_{k+1} = w_k \tag{2.33}$$

$$a_{k+1} = a_k \tag{2.34}$$

Obviously, as it possible to see in Figure 2.13 and Figure 2.14, the behaviour of the acceleration and yaw rate data sets is not constant in the analysed scenario. This means that, even if the choice of the CTRA model is the most recommended, as reported in Section 2.2.3, already at this stage of the work it is possible to understand that the equations (2.33) and (2.34) introduce some assumptions that will lead to have a simulated trajectory not perfectly consistent with the authentic one.

2.4.1 Coherence between yaw rate data and transition equation

As seen in equation (2.32), the evolution of the heading and yaw rate data must be concordant. This means that, since the yaw rate represents the derivative of the heading, an upward increase of the heading values must coincide with a positive development of the yaw rate.

Therefore, a further modification on the data provided by the CAN bus of the vehicle must be executed. In particular, the most effective way to respect the relation (2.32) consists in changing the sign of the overall yaw rate data, in order to obtain a concordant development of the two state space variables, as shown in Figures 2.18 and 2.19:

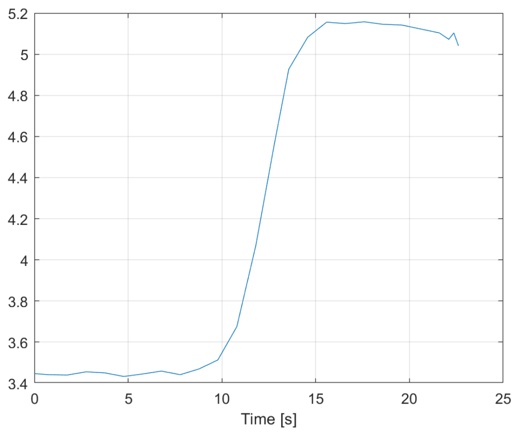


Figure 2.18: Heading behaviour

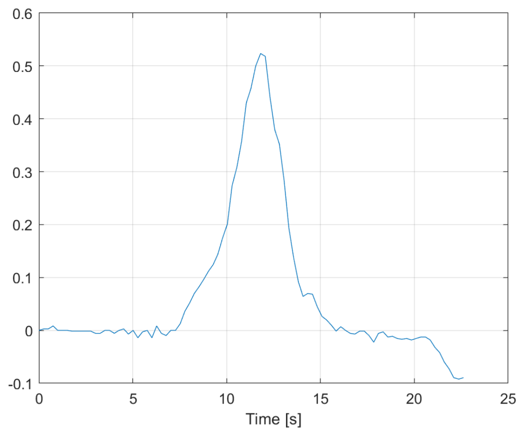


Figure 2.19: Yaw rate behaviour

2.4.2 Introduction of the straight motion equations

The CTRA model belongs to the class of the curvilinear models, which introduce also the rotation around the z - axis in the description of the vehicle motion. The chosen model, therefore, better describe situations where the yaw rate values are different from zero, which means that the vehicle is not following exactly a straight line. This is demonstrated by the fact that, in the computation of the vehicle change in position, as showed in equations (2.29) and (2.30), a factor $\frac{1}{w^2}$ is present. If the yaw rate w is equal to zero, in fact, the equations assume an indeterminate form, and do not permit to obtain the variations in longitudinal and latitudinal positions.

For this reason, it is necessary to complete the CTRA model description of the system by introducing the straight motion equations:

$$x_{k+1} = x_k + v_{x,k}\Delta t + \frac{1}{2}a_{x,k}\Delta t^2 \tag{2.35}$$

$$y_{k+1} = y_k + v_{y,k}\Delta t + \frac{1}{2}a_{y,k}\Delta t^2 \tag{2.36}$$

However, this couple of equations is applied only when the values of the yaw rate are very close to zero. Hence, the equations (2.35) and (2.36) are invoked only when the following condition is verified:

$$-0.025 < w < 0.025$$

which is the optimal condition that permits to neglect the small yaw rate oscillations near the zero, as shown by the rectangular box in Figure 2.20. Below these values, the equations (2.29) and (2.30) of the CTRA model are not anymore capable to provide useful values, since they return the indeterminate form mentioned before.

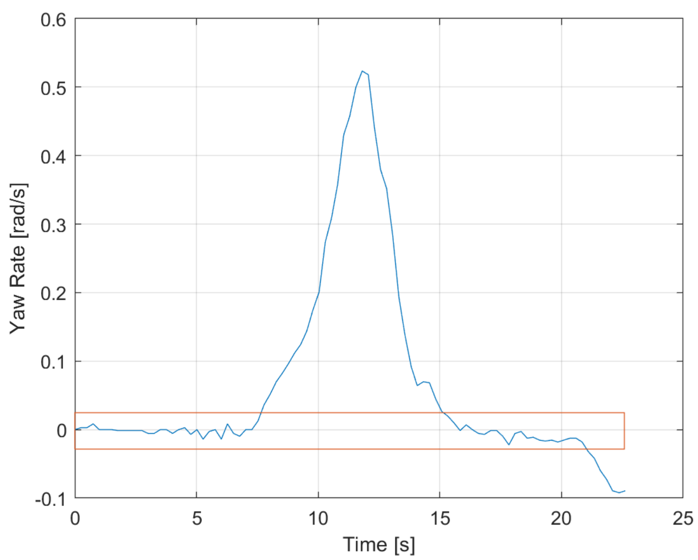


Figure 2.20: Condition on the yaw rate

Moreover, the CAN bus of the vehicle provides the data relating to the resulting speed in the direction of motion, while, for the calculation of the longitudinal and latitudinal positions with the straight motion equations, it is necessary to have the speed and acceleration components projected on the two axes. Consequently, every time the equations (2.35) and (2.36) are invoked, a decomposition of the provided speed data must be actuated.

In order to perform the speed decomposition, it is necessary to focus on how the heading angle is defined, *i.e.* where its origin is located and what values the angle assumes during the data acquisition.
Looking at the situation showed in Figure 2.21, it is clear that, since the origin of the heading angle is located in the vertical axis, the speed decomposition at each iteration of the cycle can be done in the following way:

$$v_{x,k} = v_k sin(\theta_k) \tag{2.37}$$

$$v_{y,k} = v_k cos(\theta_k) \tag{2.38}$$

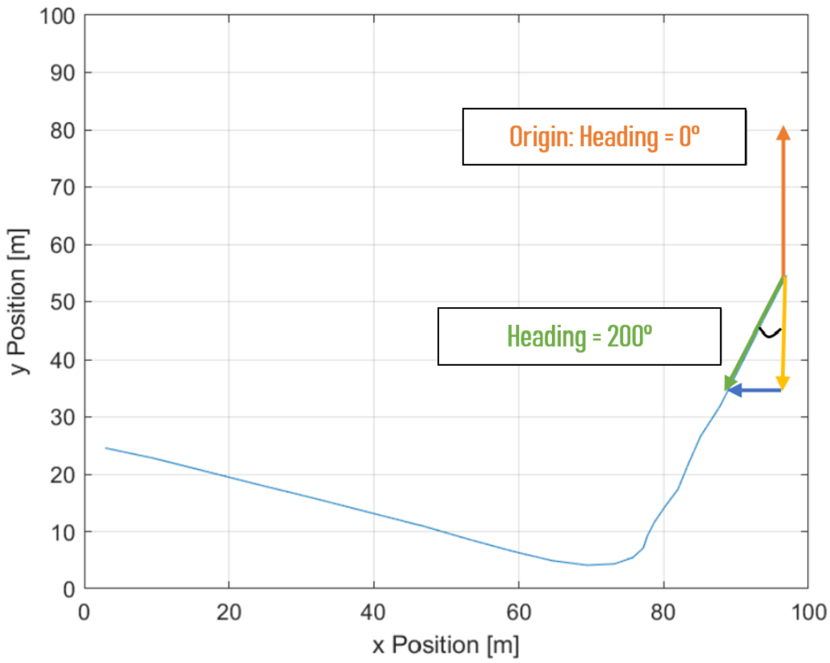


Figure 2.21: Speed decomposition

Consequently, the components of the acceleration that are needed in the equations (2.35) and (2.36) can be obtained by deriving the variations of the speed components, between one measure and the previous one, in the corresponding time interval:

$$a_{x,k} = \frac{v_{x,k+1} - v_{x,k}}{t_{k+1} - t_k} = \frac{\Delta v_x}{\Delta t} \tag{2.39}$$

$$a_{y,k} = \frac{v_{y,k+1} - v_{y,k}}{t_{k+1} - t_k} = \frac{\Delta v_y}{\Delta t} \tag{2.40}$$

2.4.3 Consistency check between model equations and data flow

Before moving on to the simulation of the CTRA model completed with the straight motion equations, a last check on the equations (2.31) and (2.32) is necessary. These relations, which describe the speed and heading evolution, must be compared with the behaviour showed by the CAN bus for the same state space variables, in order to ensure that, when the implemented model is simulated in an open - chain situation, it is able to replicate the same performances.

The scenario showed in Figure 2.22 is obtained by comparing the speed data with the behaviour of equation (2.31), in which the values of the moving average of the acceleration obtained with a buffer of five measurements are supplied at each iteration of the cycle.

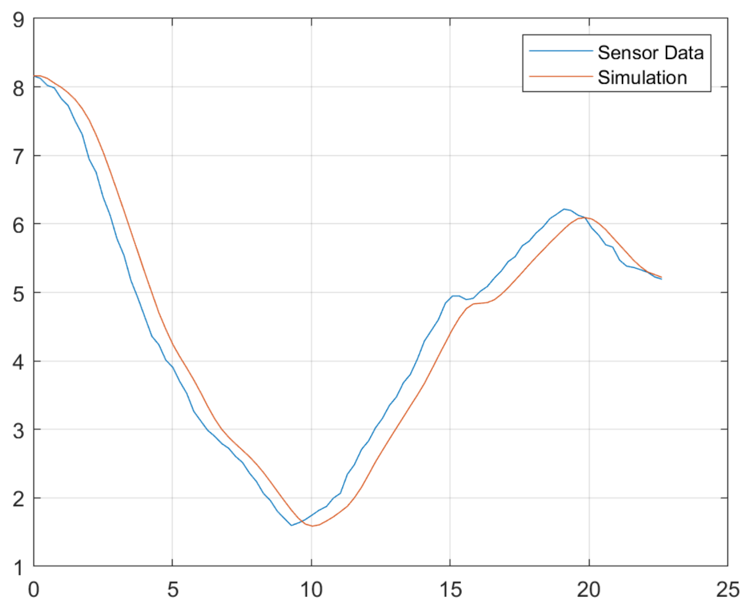


Figure 2.22: Check on speed

The evolution of the speed, calculated with the CTRA model equation, is the same of the one proposed by the real data. The only difference, which has already been taken into account in Section 2.3.3 (Moving average of the acceleration), consists in the usage of the moving average of the acceleration for the computation of the speed variation. In fact, as written before, the proposed approach for the calculation of the acceleration is free from discontinuities, but is characterized by a slight delay with respect to the time derivative approach. This fact explains why, in Figure 2.22, the simulation performance is lightly shifted with respect to the the sensor data.

Regarding the behaviour obtained with the usage of the heading equation (2.32), this is roughly equal to the one described by the sensor data, with a slight difference in the intermediate and final phase. The comparison between the results provided by the equation, in which the real yaw rate data are supplied at each iteration, and the information of the sensor data is indicated in Figure 2.23:

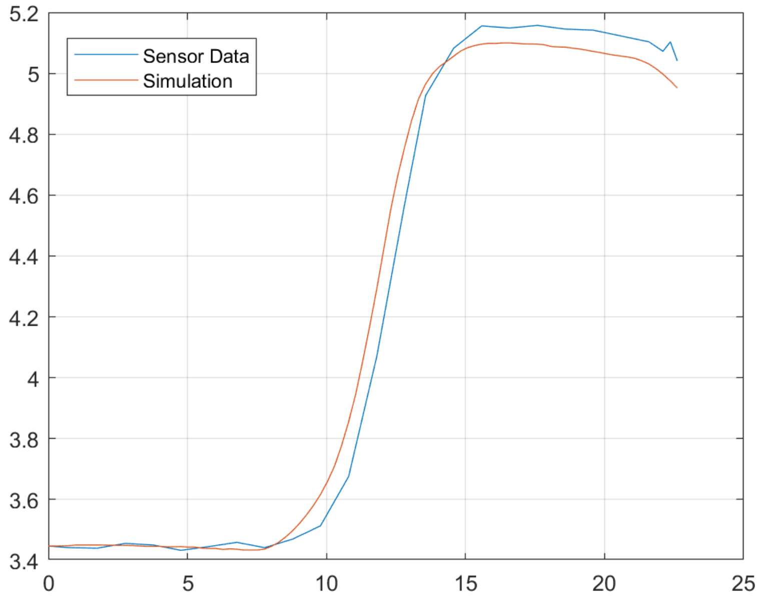


Figure 2.23: Check on heading

It can therefore be concluded that both the checks results passed and the model succeeds in replicating adequately the behaviour of these two state space variables.

However, it is necessary to specify that, in this implementation phase of the model, there are some factors that permit to facilitate the simulation. At this stage of determination of the capabilities of the model, when the overall cycle is simulated and iterated, the values that are used within the equations previously explained are

directly taken from the information coming from the real vehicle. This means that, for each iteration of the cycle, it is allowed to calculate the actual Δt between one measure and the previous one, as well it is possible to make use of the real values for the speed, heading and yaw rate state space variables.
These simplifications, instead, will not be able to be used in the real - time application implemented in the next chapter, where the assumptions made by the CTRA model assume a fundamental role.

2.4.4 Simulations of the total model

After carrying out the preliminary work on the provided data and having implemented the overall model in the simulation program, it is now possible to conduct some simulations concerning the trajectory of the vehicle in the studied case of application, with the aim of understanding if the examined model is able to replicate the same trajectory shape of the one described by the CAN bus data.

Two simulations of the overall model are performed:

1. **Separate simulation of the straight and curve sections:** in this test the total route is divided into three sections. The model has to simulate this three areas starting from the initial position of each one of them.
2. **Simulation of the total path:** in the second test, instead, the model starts the simulation with the initial position provided by the data, but then it has to simulate the overall path followed by the vehicle.

2.5 Results

The results performed by the separate simulation of the straight segments are showed in Figure 2.24:

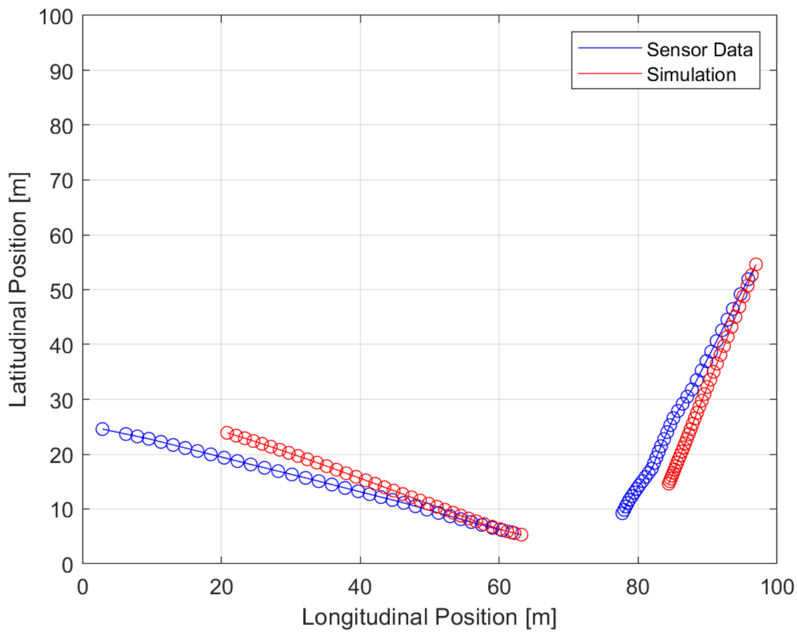


Figure 2.24: Simulation of the straight paths

Therefore, the model is able to replicate the rectilinear shape of these segments, with a small difference in the orientation. This is attributable to the assumptions introduced by the speed decomposition described in Figure 2.21, and by the fact that, in the straight motion equations (2.35) and (2.36), there is no term which takes into account the values of the the yaw rate.

Then, the simulation of the curve is conducted, and the results are highlighted in Figure 2.25:

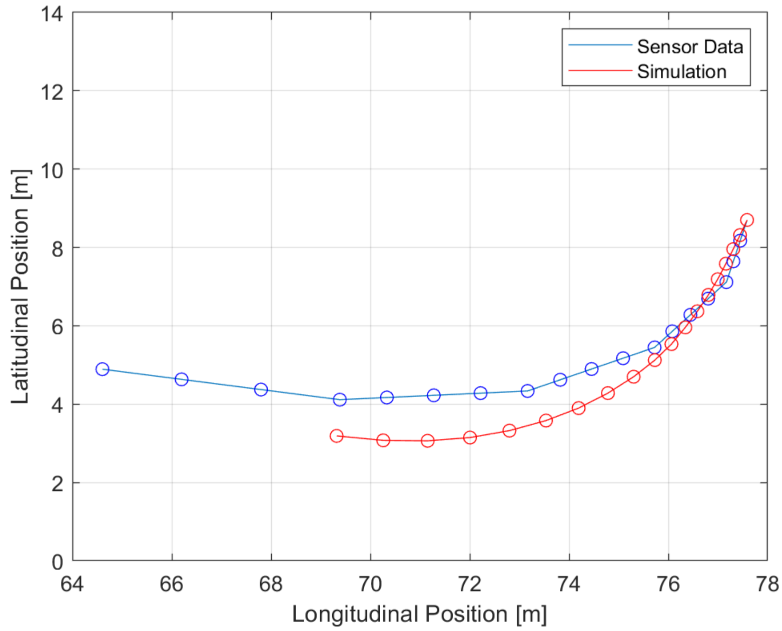


Figure 2.25: Simulation of the curve section

The obtained behaviour shows the capability of the CTRA model to adequately reproduce a curve section, obtaining also a reproduction of the trajectory that is smoother with respect to the one derived by the data. However, already at this stage the assumptions regarding the acceleration and the yaw rate have an effect on the results. In fact, by considering them constant between one iteration of the cycle and the next one, the model obtains a series of simulated curve sections which are slightly shorter than the authentic ones. This will be one of the aspect that the path prediction technique used in the following chapter will seek to improve.

The second test, instead, is performed on the overall trajectory followed by the vehicle. The result, showed in Figure 2.26, brings together all the aspects obtained in the previous tests and demonstrates the ability of the chosen model to follow the rectilinear paths and, when required, change its orientation to perform a curve in the better way.

Obviously, in the simulation of the overall trajectory, all the inaccuracies of the first tests are summed up, obtaining then a resultant path slightly different from the authentic one. However, in this phase, the result is satisfactory, since the aim was to study and implement a model able to perform in an adequate way in most situations, and here, thanks to the condition imposed to the yaw rate, the model shows its ability to correctly switch from the straight motion equations to the CTRA model relations, which permit to have a proper response both in a rectilinear than in a curve situation.

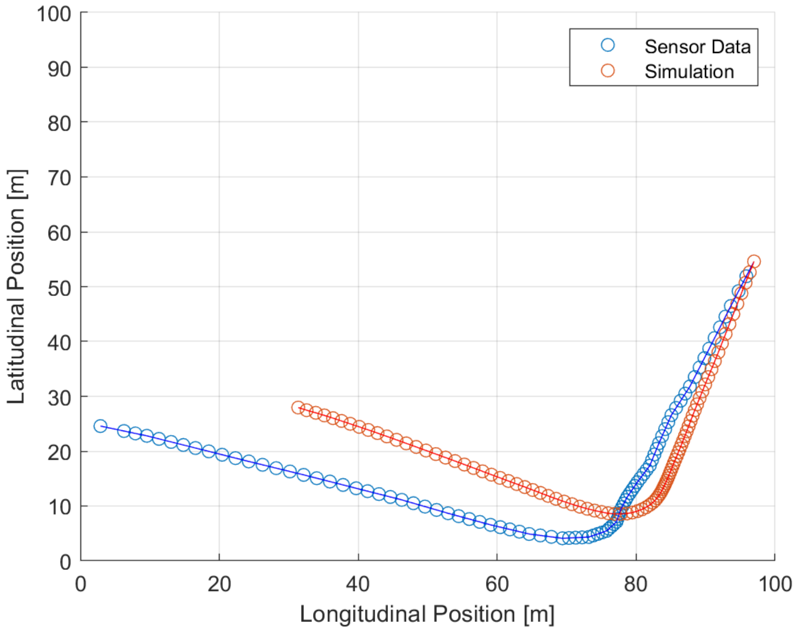


Figure 2.26: Simulation of the overall trajectory

In the next chapter, the study conducted on the overall model results fundamental to calculate the positions and the trajectory followed by the vehicle. The implemented model will be then used in conjunction with a particular path prediction technique, which, between its different tasks, will be responsible for resolving and correcting the position inaccuracies coming from the open chain simulation of the model.

Chapter 3

Vehicle path prediction

The third chapter of this thesis project is developed following a structure similar to that of the previous part. First of all, a state of the art concerning the technique used for path prediction is presented. Here, the well - known Kalman filter algorithm is initially treated, and then its main drawback is presented. After that, the adopted procedure, which is the unscented transform, is introduced and the chosen Unscented Kalman Filter algorithm is described. Thereafter, the phases of implementation of the filter are explained, which include the plant modeling, the sensor modeling, the filter construction and the prediction and correction phases. Finally, the analysed simulations are introduced, followed by the final results obtained with the combination of the CTRA model and the Unscented Kalman Filter.

3.1 Unscented Kalman Filter: State of the Art

After having ensured the suitability of the implemented model, the description of the employed path prediction technique must be presented. For this phase, a particular declination of the KF (Kalman Filter) has been chosen.

3.1.1 Kalman Filter algorithm

In order to achieve path prediction goals, the model described in the previous chapter must be simulated in the state prediction step of the filter. In fact, as is well known, the KF is mainly developed around two steps, which are the state prediction and the state update. The state prediction step is based on the actual state of the dynamic system, characterized by a random Gaussian noise, and the model permits to obtain the predicted state. The state update step, instead, relies on the information present in the noisy measurements and in the predicted state, with the aim of updating the system state.

Summing up, as described in [28] and [19], the ordinary KF algorithm consists of the following elements:

a) State transition equation:

$$x_{k+1} = F_{k+1,k} x_k + v_k \tag{3.1}$$

where $F_{k+1,k}$ is the transition matrix that takes the state x_k from the instant k to $k+1$, while v_k stands for the additive process noise characterized by a Gaussian distribution with zero mean and covariance identified by:

$$E[v_n {v_k}^T] = \begin{cases} Q_k & for \;\; n = k \\ 0 & for \;\; n \neq k \end{cases} \tag{3.2}$$

b) Measurement equation:

$$y_k = H_k x_k + w_k \tag{3.3}$$

where y_k is the available measurement at time instant k, and H_k represents the measurement matrix. The measurement noise w_k, as before, is assumed to be additive and it is characterized by a Gaussian distribution with zero mean and covariance given by:

$$E[w_n {w_k}^T] = \begin{cases} R_k & for\ \ n = k \\ 0 & for\ \ n \neq k \end{cases} \tag{3.4}$$

c) State estimate propagation: firstly, a priori estimate of the state distribution is identified in terms of the previous a posteriori estimate with the equation (3.5):

$$\hat{x}_k^- = F_{k,k-1}\hat{x}_{k-1} \tag{3.5}$$

d) Covariance matrix propagation:

$$P_k{}^- = F_{k,k-1}P_{k-1}{F_{k,k-1}}^T + Q_{k-1} \tag{3.6}$$

where this relation expresses the dependence of the a priori covariance matrix $P_k{}^-$ on the previous a posteriori covariance matrix P_{k-1}.

e) Kalman gain computation:

$$G_k = P_k{}^- {H_k}^T (H_k P_k{}^- {H_k}^T + R_k)^{-1} \tag{3.7}$$

where the Kalman gain is defined with respect to the a priori covariance matrix $P_k{}^-$.

f) State estimate update:

$$\hat{x}_k = \hat{x}_k^- + G_k(y_k - H_k\hat{x}_k^-) \tag{3.8}$$

g) Covariance matrix update:

$$P_k = (I - G_k H_k)P_k{}^- \tag{3.9}$$

3.1.2 Drawback of the Kalman Filter algorithm

The algorithm and the equations presented are involved in the study of linear systems, and this basic situation is represented in Figure 3.1. Assuming that the state transition function is linear, then, after undergoing the linear transformation, the initial Gaussian distribution maintains its properties. Even if it is not shown in Figure 3.1, the same concept is valid for the measurement function.

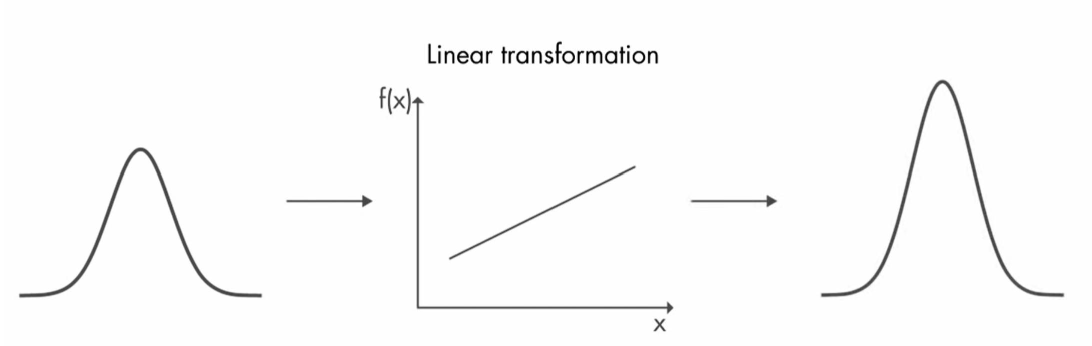


Figure 3.1: Linear transformation [18]

However, if the state transition function is nonlinear, then the resulting state distribution may not be Gaussian, as represented in Figure 3.2. If this happens, the KF algorithm may not converge, as instead happens for a linear case.

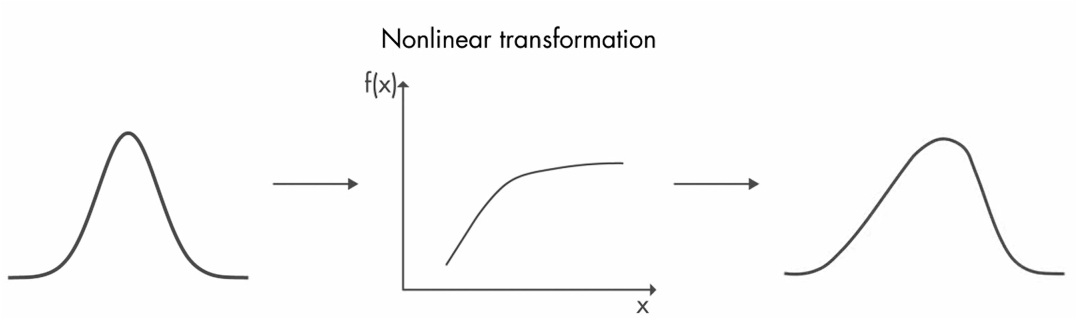


Figure 3.2: Nonlinear transformation [18]

In fact, one of the main drawbacks of the KF algorithm consists in the consideration that both the state prediction equation (3.1) and the measurement equation (3.3) are linear. If this condition is not anymore verified, the analysis conducted by the filter can no longer be considered adequate, as reported in [18].

Considering the state transition function of the CTRA model, and in particular the equations (2.29) and (2.30) for the variations of longitudinal and latitudinal positions, it is clear that this project has to deal with a strongly nonlinear system. When this type of systems is analysed, instead of an ordinary KF, nonlinear filtering techniques such as the EKF (Extended Kalman Filter) and the UKF (Unscented Kalman Filter) must be taken into account. However, in this thesis project, the UKF is used, since, as written in [4] and in [30], its performances are superior over the EKF as regards the filter convergence, position accuracy and velocity accuracy.

3.1.3 Unscented transform

The UKF approaches the problem of propagating Gaussian random variables through nonlinear systems by applying a deterministic sampling procedure, showed in Figure 3.3.
As explained in [5] and [25], the state distribution is again represented by a Gaussian random variable, but now the filter allows to approximate it through the usage of carefully chosen sample points. These points, called sigma points, are selected by the filter such that their mean and covariance is the same as the state distribution, leading to have a set of sigma points symmetrically distributed around the mean.
Each one of these is then propagated through the nonlinear system model, as a result of the passage through the nonlinear state transition function mentioned before. After this operation, the mean and covariance of the nonlinearly transformed points are calculated, and an empirical Gaussian distribution is computed, which is then used to calculate the new state estimate.

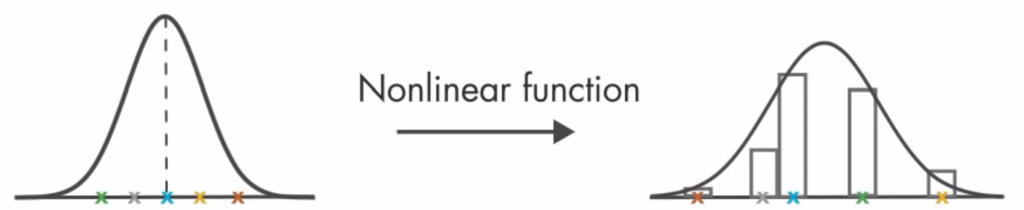


Figure 3.3: UKF procedure [18]

The procedure just described is based on the unscented transform, which is a method that permits to calculate the statistics of a distribution that encounters a nonlinear alteration.

Assume that a random variable x, which is characterized by dimension L, mean $\overline{x}$ and covariance P_x, must be propagated through a nonlinear function $y = f(x)$. In order to obtain the statistics of y, the unscented transform permits to build a matrix X of $2L+1$ sigma vectors X_i:

$$
\begin{aligned}
&X_0 = \overline{x} \\
&X_i = \overline{x} + (\sqrt{(L+\lambda)P_x})_i \qquad && i = 1, ..., L \\
&X_i = \overline{x} - (\sqrt{(L+\lambda)P_x})_{i-L} \qquad && i = L+1, ..., 2L
\end{aligned}
\tag{3.10}
$$

where $\lambda = \alpha^2(L+\kappa) - L$ represents a scaling parameter, with α that helps to determine the spread of the chosen sigma points around the mean of the state distribution (usually $1 \leq \alpha \leq 10^{-4}$), and κ denotes another scaling parameter, which is usually equal to $3 - L$.

The sigma vectors X_i are propagated by means of the nonlinear state transition function:

$$Y_i = f(X_i) \qquad i = 0, ..., 2L \tag{3.11}$$

and the calculation of the mean and covariance of the posterior sigma points is computed:

$$\bar{y} \approx \sum_{i=0}^{2L} W_i^{(m)} Y_i \tag{3.12}$$

$$P_y \approx \sum_{i=0}^{2L} W_i^{(c)} (Y_i - \bar{y})(Y_i - \bar{y})^T \tag{3.13}$$

where the weights W_i are provided by:

$$\begin{aligned} W_0(m) &= \frac{\lambda}{L+\lambda} \\ W_0(c) &= \frac{\lambda}{L+\lambda} + 1 - \alpha^2 + \beta \\ W_i(m) = W_i(c) &= \frac{1}{2(L+\lambda)} \qquad i = 1, ..., 2L \end{aligned} \tag{3.14}$$

with the parameter β that permits to include prior knowledge of the distribution of x.

The overall process performed by the unscented transform and its main passages are shown in Figure 3.4, which sums up all the aspects just described.

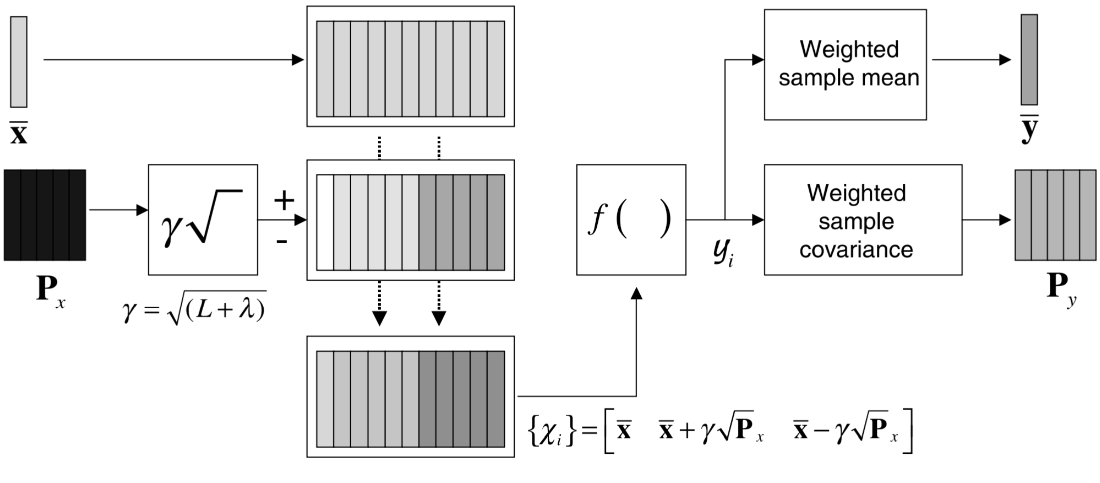


Figure 3.4: Main passages of the unscented transform [5]

3.1.4 UKF algorithm

The UKF represents a direct application of the unscented transform to the recursive algorithm estimation (3.15):

$$\hat{x}_k = (prediction\ of\ x_k) + G_k[y_k - (prediction\ of\ y_k)] \tag{3.15}$$

In this thesis project the case where the characteristic process and the measurement noises are purely additive is analyzed. This special case, but very often met, permits to lower the complexity of the UKF, reducing the number and dimensions of the needed sigma points.

The algorithm proposed by the UKF for this type of problem, as explained in [5] and [14], begins with the initialization of the state estimate and covariance matrix:

$$\hat{x}_0 = E[x_0] \tag{3.16}$$

$$P_0 = E[(x_0 - \hat{x}_0)(x_0 - \hat{x}_0)]^T \tag{3.17}$$

Then, the sigma points are calculated:

$$X_{k-1} = [\hat{x}_{k-1} \quad \hat{x}_{k-1} + \gamma\sqrt{P_{k-1}} \quad \hat{x}_{k-1} - \gamma\sqrt{P_{k-1}}] \tag{3.18}$$

The time update of the sigma points is obtained by passing them to the nonlinear state transition function F:

$$X^*_{k|k-1} = F(X_{k-1}) \tag{3.19}$$

while the time update for the a priori state estimate and the covariance matrix is given by:

$$\hat{x}_k^- = \sum_{i=0}^{2L} W_i^{(m)} X^*_{i,k|k-1} \tag{3.20}$$

$$P_k^- = \sum_{i=0}^{2L} W_i^{(c)} (X^*_{i,k|k-1} - \hat{x}_k^-)(X^*_{i,k|k-1} - \hat{x}_k^-)^T + Q \tag{3.21}$$

where Q represents the process noise covariance.

Now, as done in equation (3.10), it is necessary to increase the number of the sigma points by augmenting the parameter L to $2L$, and calculating again the different weight W_i, as reported in equation (3.14).

After having performed this phase, the new set of sigma points (3.22) permits to obtain the following steps (3.23) and (3.24):

$$X_{k|k-1} = [X^*_{k|k-1} \quad X^*_{0,k|k-1} + \gamma\sqrt{Q} \quad X^*_{0,k|k-1} - \gamma\sqrt{Q}] \tag{3.22}$$

$$Y_{k|k-1} = H(X_{k|k-1}) \tag{3.23}$$

$$\hat{y_k}^- = \sum_{i=0}^{2L} W_i{}^{(m)} Y_{i,k|k-1} \tag{3.24}$$

Finally, the UKF performs the measurement update, described by:

$$P_{\tilde{y}_k\tilde{y}_k} = \sum_{i=0}^{2L} W_i{}^{(c)} (Y_{i,k|k-1} - \hat{y}_k^-)(Y_{i,k|k-1} - \hat{y}_k^-)^T + R \tag{3.25}$$

$$P_{x_k y_k} = \sum_{i=0}^{2L} W_i{}^{(c)} (X_{i,k|k-1} - \hat{x}_k^-)(Y_{i,k|k-1} - \hat{y}_k^-)^T \tag{3.26}$$

where R represents the measurement noise.

Thanks to the results provided by (3.25) and (3.26), the Kalman gain can be computed:

$$G_k = P_{x_k y_k} P^{-1}_{\tilde{y}_k\tilde{y}_k} \tag{3.27}$$

The value of the Kalman gain permits to obtain the final results, which are the update of the state estimate and covariance matrix:

$$\hat{x}_k = \hat{x}_k^- + G_k(y_k - \hat{y_k}^-) \tag{3.28}$$

$$P_k = P_k^- - G_k P_{\tilde{y}_k\tilde{y}_k} G_k^T \tag{3.29}$$

3.2 Implementation of the UKF

Now that the characteristic process of the UKF has been presented, it is necessary to proceed with its implementation within the simulation program. In order to do this, the approach explained in [15] is followed.

3.2.1 Plant modeling

As described in Chapter 2.2.2, the chosen CTRA model state space is based on six variables:

$$x = (x \;\; y \;\; v \;\; \theta \;\; w \;\; a)^T \tag{3.30}$$

which are the longitudinal and latitudinal positions x and y, speed v, heading angle θ, yaw rate w and acceleration a.

First of all, the implementation of the UKF needs the definition of the state transition function, which is the function that describes accurately the transition of the state space variables from the time instant k to $k+1$.
Since in this project the presence of additive process noise is considered, which means that the state and process noise are related linearly, the expression that describes the evolution of the state x is the equation (3.1), here reported for simplicity:

$$x_{k+1} = F(x_k) + v_k \tag{3.31}$$

where F represents the nonlinear state transition function and v is the additive process noise.

In the implementation phase of Chapter 2.4.2, the characteristic equations of the CTRA model are completed with the straight motion relations in order to better represent the trajectory of the vehicle. Now, the state transition function has to consider this modification, and both the groups of equations must be included.

An additional difficulty in the implementation of the filter and in the analysis of a real - time application is constituted by the fact that, during the usage of the state transition function, the filter works only with the variables supplied as inputs of the function. This makes it necessary to consider a series of aspects in the definition of the cycle:

- the time interval between the iterations of the cycle can no longer be calculated every time, and so it can not be considered variable anymore. From now on, it is considered constant and equal to the inverse of the sampling frequency of the messages sent by the CAN bus, which is equal to 4 Hz. This means that the adopted dt is now equivalent to 0.25 s.

- the evolution of the speed and heading variables is managed by the equations introduced in Chapter 2.2.2:

$$v_{k+1} = v_k + a_k dt \tag{3.32}$$

$$\theta_{k+1} = \theta_k + w_k dt \tag{3.33}$$

which permit to utilize only the values introduced in the function at time k.

- during the time in which the implemented model is simulated, the values of the acceleration and yaw rate do not change, due to the assumptions of the CTRA model, which consider them constant:

$$w_{k+1} = w_k \tag{3.34}$$

$$a_{k+1} = a_k \tag{3.35}$$

Considering all the aspects just described, the nonlinear state transition function is implemented as shown in Figure 3.5 and as described below.

At every iteration of the cycle, when the intervention of the UKF is invoked, the state transition function decides which equations to use on the basis of the condition concerning the yaw rate. If, at instant k, the condition $-0.025 < w_k < 0.025$ results verified, the calculation are then managed by the straight motion equations. In this case, the speed decomposition on the two axes is carried out first, as showed by the equations from (3.36) to (3.39):

$$v_{x,k+1} = (v_k + a_k dt) sin(\theta_k + w_k dt) \tag{3.36}$$

$$v_{y,k+1} = (v_k + a_k dt) cos(\theta_k + w_k dt) \tag{3.37}$$

$$v_{x,k} = v_k sin\theta_k \tag{3.38}$$

$$v_{y,k} = v_k cos\theta_k \tag{3.39}$$

Thanks to these values, the acceleration from instant k to $k+1$ can be obtained:

$$a_{x,k} = \frac{v_{x,k+1} - v_{x,k}}{dt} \tag{3.40}$$

$$a_{y,k} = \frac{v_{y,k+1} - v_{y,k}}{dt} \tag{3.41}$$

Now that all the terms that appear in the straight motion equations are available, the calculation concerning the longitudinal and latitudinal positions of the vehicle can be conducted:

$$x_{k+1} = x_k + v_{x,k}dt + \frac{1}{2}a_{x,k}dt^2 \tag{3.42}$$

$$y_{k+1} = y_k + v_{y,k}dt + \frac{1}{2}a_{y,k}dt^2 \tag{3.43}$$

On the contrary, if the initial condition $-0.025 < w_k < 0.025$ is not verified, this means that the vehicle is following a curvilinear path, and the CTRA equations must be taken into account for the determination of the position:

$$\begin{aligned} x_{k+1} = x_k + \Delta x = x_k + \frac{1}{w_k^2}[&(v_k w_k + a_k w_k dt)sin(\theta_k + w_k dt) + a_k cos(\theta_k + w_k dt) \\ &- v_k w_k sin\theta_k - a_k cos\theta_k] \end{aligned} \tag{3.44}$$

$$\begin{aligned} y_{k+1} = y_k + \Delta y = y_k + \frac{1}{w_k^2}[&(-v_k w_k - a_k w_k dt)cos(\theta_k + w_k dt) + a_k sin(\theta_k + w_k dt) \\ &+ v_k w_k cos\theta_k - a_k sin\theta_k] \end{aligned} \tag{3.45}$$

Now that the position of the vehicle for the successive time instant $k+1$ has been derived, the update of the other state space variables must be conducted. As said before, the speed and heading update is computed thanks to the equations (3.32) and (3.33), while, when the model is simulated in an open chain mode, the acceleration and yaw rate are considered constant, as written in equations (3.34) and (3.35).

```
function x = StateTransitionFunction(x)

% x=[Pos_x ; Pos_y ; Speed ; Heading ; Yaw Rate ; Acc]

dt = 0.25; %[s] Sample Time / Prediction time interval

  if x(5) < 0.025 && x(5) > -0.025

         %Speed
         Speed_x_2 = (x(3)+x(6)*dt)*sin(x(4)+x(5)*dt);
         Speed_y_2 = (x(3)+x(6)*dt)*cos(x(4)+x(5)*dt);

         Speed_x_1 = x(3)*sin(x(4));
         Speed_y_1 = x(3)*cos(x(4));

         %dv
         dv_x = Speed_x_2 - Speed_x_1;
         dv_y = Speed_y_2 - Speed_y_1;

         %Acc
         Acc_x = dv_x/dt;
         Acc_y = dv_y/dt;

         %Update positions (Straight motion equations)
         x(1) = x(1) + Speed_x_1*dt+(Acc_x*dt^2)/2;

         x(2) = x(2) + Speed_y_1*dt+(Acc_y*dt^2)/2;

  else
         %Update positions (CTRA model equations)
         x(1) = x(1) + (x(3)*cos(x(4))-(x(3)+x(6)*dt)*cos(x(5)*dt+x(4)))/x(5)
                       +x(6)*(sin(x(5)*dt+x(4))-sin(x(4)))/x(5)^2;

         x(2) = x(2) + ((x(3)+x(6)*dt)*sin(x(5)*dt+x(4))-x(3)*sin(x(4)))/x(5)
                       +x(6)*(cos(x(5)*dt+x(4))-cos(x(4)))/x(5)^2;

  end

%Update Speed
x(3) = x(3) + x(6)*dt;

%Update Heading
x(4) = x(4) + x(5)*dt;

%Update Yaw Rate
x(5) = x(5);

%Update Acceleration
x(6) = x(6);

end
```

Figure 3.5: State transition function

3.2.2 Sensor modeling

The UKF needs also the so - called measurement function, which describes how the state space variables of the model are related to sensors measurement.
Due to the presence of additive measurement noise, the form of the measurement

function is the one proposed in equation (3.46):

$$y_k = H(x_k) + w_k \tag{3.46}$$

where y_k represents the considered measurements, H is the measurement function and w denotes the additive measurement noise.

As explained in Chapter 1.2 during the description of the context and scenario analysed in this thesis project, two different points of view are considered:

1) Vehicle: from the CAN bus it is possible to obtain information about five of the six state space variables. The only values that are not provided by the communication system are those of the acceleration, which instead have been obtained with the moving average method. For these reasons, the measurement function for this first case assumes the form shown in Figure 3.6.

```
function yk = MeasurementFunction(xk)

yk = xk(1:5);

end
```

Figure 3.6: Measurement function - Vehicle

2) Camera: the infrastructure placed at the road intersection is able to observe the movements of all the vehicles, and it is particularly important for the study of the non - connected ones. Since it works by exploiting the captured images, the only measured variables are represented by the longitudinal and latitudinal positions. This means that, in this second case, only the first two state space variables are considered by the measurement function, which has the form shown in Figure 3.7.

```
function yk = MeasurementFunction(xk)

yk = xk(1:2);

end
```

Figure 3.7: Measurement function - Camera

3.2.3 UKF construction

The state transition and measurement functions just described must be provided in the UKF construction phase. These information are then completed with the initial state space of the model, which assumes the initial values of the six state space variables:

$$x_0 = (x_0 \ \ y_0 \ \ v_0 \ \ \theta_0 \ \ w_0 \ \ a_0)^T \tag{3.47}$$

In this construction phase of the filter, also the knowledge of the measurement noise covariance R must be provided. The best results in terms of filter correction, which is based on sensor measurements, and errors between the predicted trajectory and the authentic one, have been found with the value:

$$R = 0.1 \tag{3.48}$$

The same is done for the process noise covariance Q, which is set in order to take into account model inaccuracies and the effect of unknown disturbances on the plant. As before, the best results in terms of errors between the predicted trajectory and the real data have been provided by the diagonal matrix:

$$Q = diag(0.01 \ \ 0.01 \ \ 0.01 \ \ 0.01 \ \ 0.025 \ \ 0.05) \tag{3.49}$$

which is characterized by higher values for the last two variables in order to reflect that the yaw rate and acceleration states are more impacted by modeling errors due to the assumptions of the CTRA model.

3.2.4 Prediction and correction phases

The prediction and correction phases represent the heart of the UKF implementation, and these are defined within a cycle that consider all the available measurements.

During the prediction phase, the UKF performs the passages shown in the equations from (3.18) to (3.21) thanks to the *predict* MATLAB command [17]. In particular, the filter carries on the choice of the sigma points and their propagation through the nonlinear function, which is represented by the state transition function defined in Section 3.2.1. Thanks to these passages, then it can proceed with the calculation of the state prediction and its covariance matrix.

The second intervention of the UKF is characterized by the correction phase. When the *correct* MATLAB command [13] is invoked, the measurement data arriving from the CAN bus of the vehicle or from the images captured by the camera are provided to the filter in order to correct the six state space variables memorized in the UKF state at that moment. Therefore, during this phase, the filter performs the passages shown in the equations from (3.22) to (3.29), which permit to correct the a - priori state estimate and the covariance matrix on the basis of the real observed measurements coming from the sensors and specified by the measurement function.

3.2.5 Simulations performed with the UKF

In order to test the capabilities of the UKF in different situations, several tests are carried out. In particular, the first two tests are performed studying both the vehicle and the camera points of view, in order to understand the differencies between them:

1. The first simulation is conducted by predicting the vehicle trajectory in the following 2.5 s and, after this amount of time, the UKF intervenes to correct the estimation of the state. This means that the model is free to simulate its future state space variables for ten successive iterations of the cycle thanks to the state transition function, and then the real data of the observed variables, highlighted by the measurement function, are considered in order to correct the estimation done in the previous iterations.

2. The structure of the second test is equal to that of the first one, with the only exception that the model predicts the vehicle trajectory for 1.25 s, which represent five successive iterations of the implemented cycle. After this amount of time, the filter intervenes to correct the state.

During the tests 1 and 2 the prediction of all the six state variables, and their successive correction, are compared between the vehicle and the camera cases. These tests, in fact, are really useful in order to study and understand the different performances offered by the UKF correction when the number of the available and measured state space variables changes.

The tests 3 and 4, instead, focus the attention on the implemented model and state transition function, in order to better comprehend what are their advantages and defects and, by consequence, their trajectory prediction capabilities:

3. In the third test the model again predicts its future states in the successive time instants, but this time the correction provided by the UKF is performed at every iteration of the cycle, and so every 0.25 s. This means that now, for every new position, the implemented model has the possibility to predict its future path on the basis of already corrected values.

4. The last simulation extends the analysis previously done to all the available data that describe the overall path shown in Figure 2.4. Here, new sections are illustrated and analysed in order to observe how the UKF behaves in a scenario not anymore limited only to the road intersection studied up to this point.

3.3 Results

For the first and second tests the vehicle and camera cases are conducted in parallel, since these simulations help to better understand and compare all the different state space variables and their behaviour, and so permit to see the various differences between the actions performed by the UKF in the two different scenarios. During these two tests, the implemented model is simulated in an open chain mode for a predefined prediction horizon, indicated with ph, and then the filter performs the correction on the state space of the model.

3.3.1 Tests 1 and 2: vehicle and camera points of view

The first test performed with the presence of the UKF is conducted by simulating the overall model in an open chain mode for a prediction horizon of ten successive iterations of the cycle, which are almost 2.5 s. Therefore, the model performs the prediction of what will be its future trajectory and, after that, the filter intervenes in order to correct the state space parameters on the basis of the measured variables.

The results and the comparison between the trajectory obtained in the vehicle and camera cases are shown in Figure 3.8:

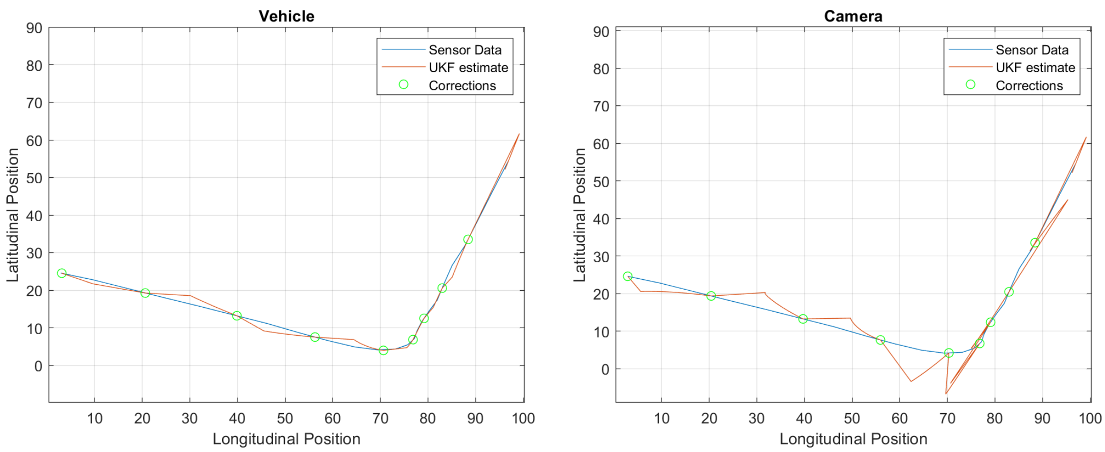


Figure 3.8: Trajectory comparison ($ph = 10$)

while the behaviour of all the six state space variables for this first test is illustrated in the Figures from 3.9 to 3.14, where the original data are represented by a blue line, the predictions performed by the model with an orange one, and the instants when the filter intervenes to correct the state space variables are indicated with green circles.

The Figures 3.9 and 3.10 represent the longitudinal and latitudinal positions calculated by the implemented model, and these are corrected every 2.5 s by the UKF. These plots confirm the difference between the predictions performed by the two different points of view: in fact, it is easy to notice how the behaviour of the camera case further deviates from the original track.

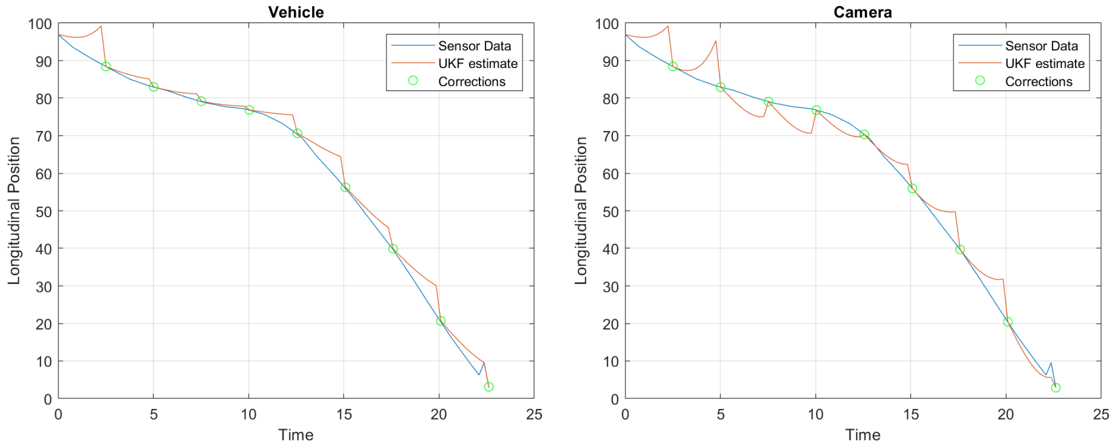


Figure 3.9: Longitudinal position comparison ($ph = 10$)

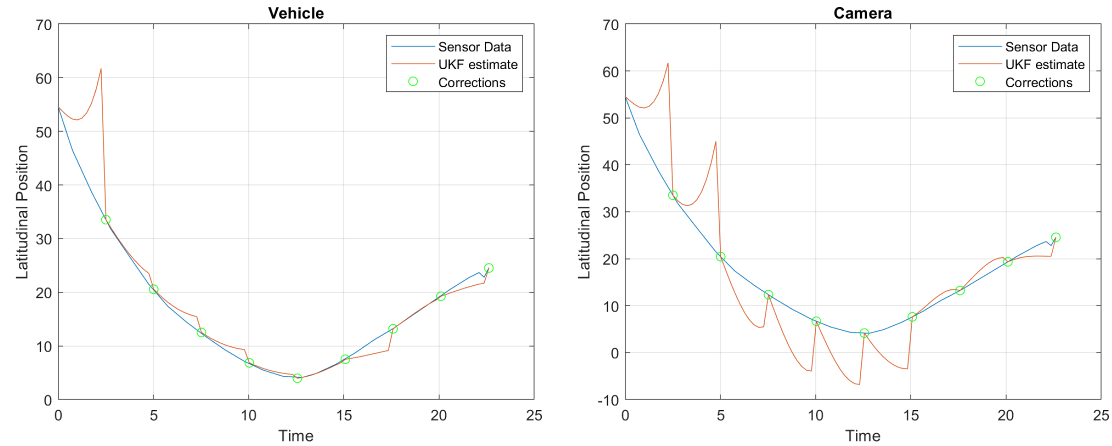


Figure 3.10: Latitudinal position comparison ($ph = 10$)

In Figure 3.11 the the model predictions regarding the speed variable are showed. Here it is possible to see how this state space variable is characterized by a constant variation, due to the CTRA model assumption on the acceleration.

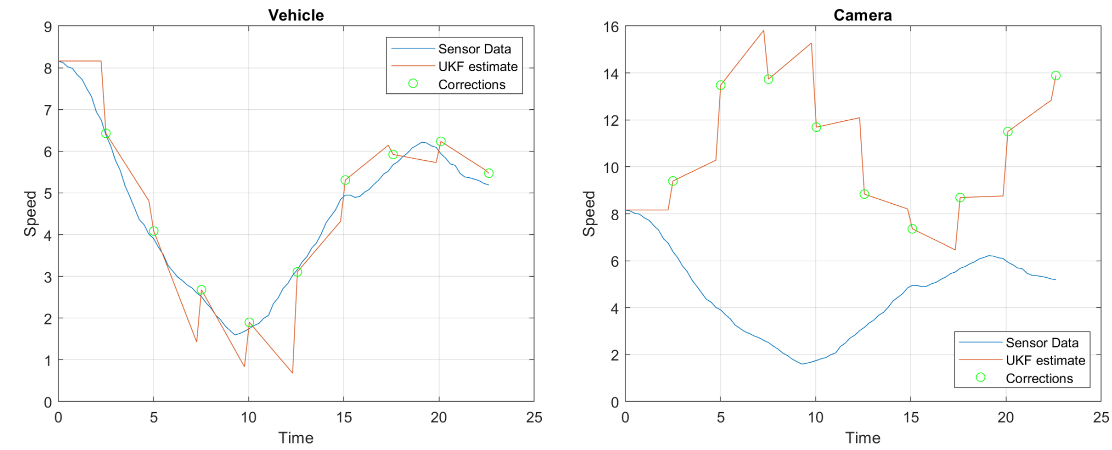


Figure 3.11: Speed comparison ($ph = 10$)

The effects of the aforementioned assumption on the acceleration are showed in Figure 3.12. Here it is possible to see that this state space variable can only assume constant values, which change after every correction performed by the UKF.

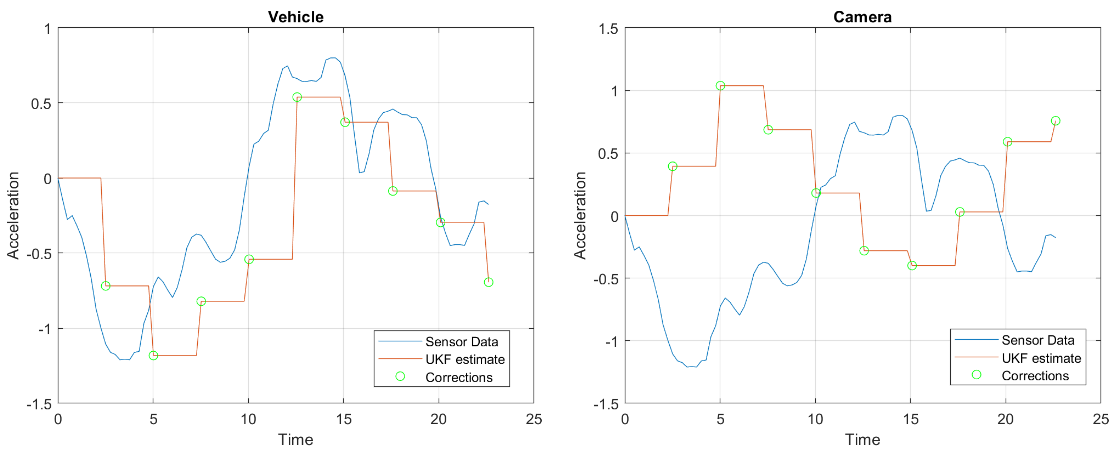


Figure 3.12: Acceleration comparison ($ph = 10$)

The heading behaviour is represented in Figure 3.13 and, as for the speed simulations of Figure 3.11, it is clear how also this variable is affected by a constant variation. This is due to the second assumption of the CTRA model that regards the yaw rate variable, which it is supposed to assume only constant values during the simulation.

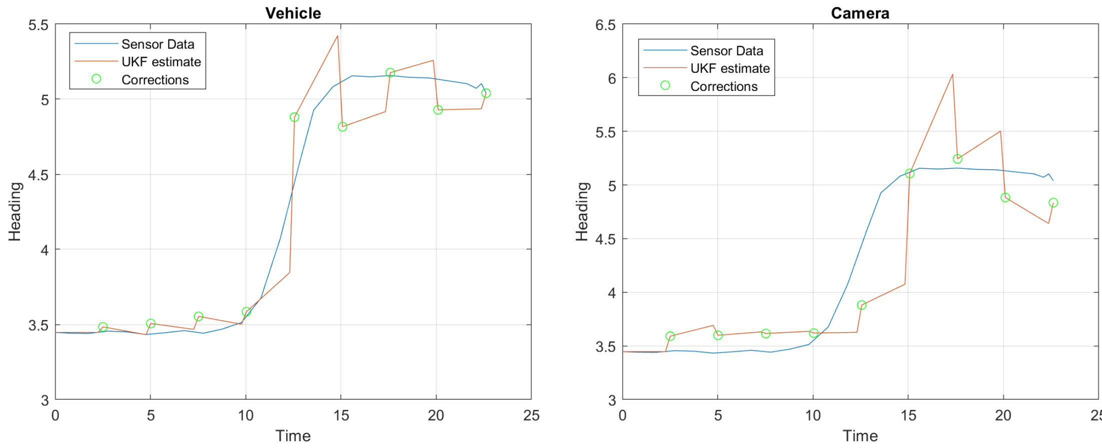


Figure 3.13: Heading comparison ($ph = 10$)

The yaw rate behaviour during the prediction simulations is showed in Figure 3.14. As for the acceleration, due to the assumptions of the CTRA model, this state space variable can assume only constant values during the predictions, which are imposed by the correction done by the filter.

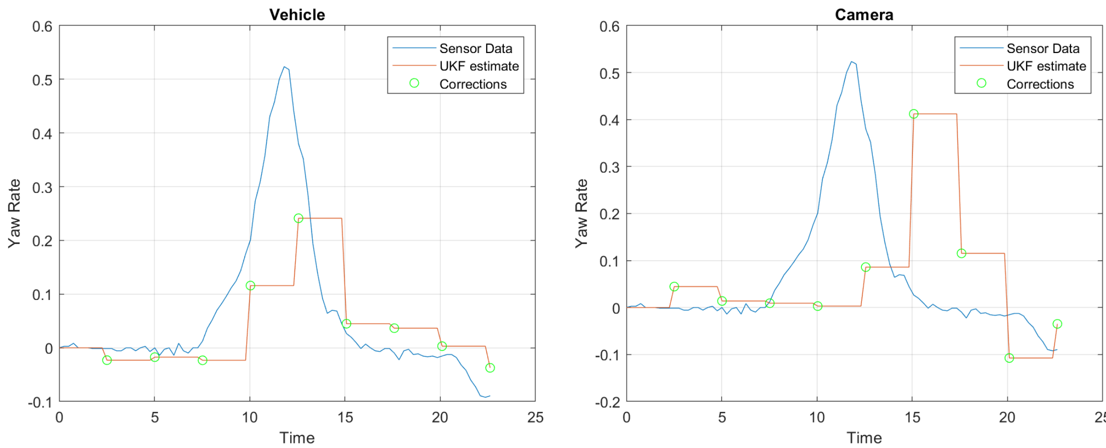


Figure 3.14: Yaw rate comparison ($ph = 10$)

Already from this first test, characterized by a prediction horizon of ten iterations of the cycle, it is possible to see how the performance of the vehicle case appear to be better than the ones of the camera case. In particular, it is clear that the UKF correction, represented by the green points in the figures, results to be more successful in the vehicle case, obtaining a shape of the state space variables closer to the authentic one with respect to the camera case.

The behaviour of the illustrated variables is improved if the prediction horizon is reduced, which means that the UKF correction becomes more frequent. In the second test, in fact, the overall model simulates its future states for five successive iterations of the cycle ($ph = 5$), which correspond to 1.25 s, and then the filter intervenes.
The results of this test and, as for the previous one, the comparison between the state space variables of the two cases are shown in Figures from 3.15 to 3.21:

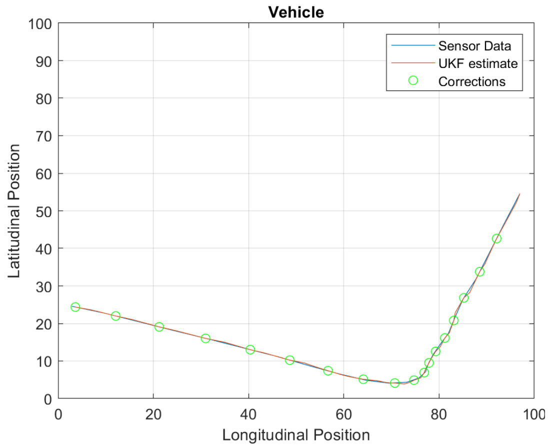


Figure 3.15: Trajectory comparison ($ph = 5$)

The results for the longitudinal and latitudinal positions are showed in Figures 3.16 and 3.17. It is clear how the simulation results for both the vehicle and camera cases result to be improved in this second test, leading to have an overall performance closer to the real positions.

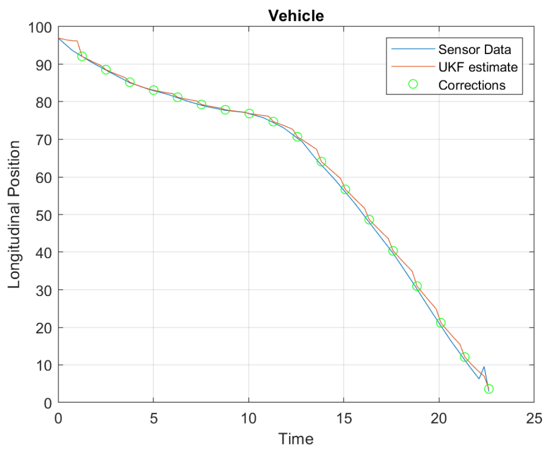


Figure 3.16: Longitudinal position comparison ($ph = 5$)

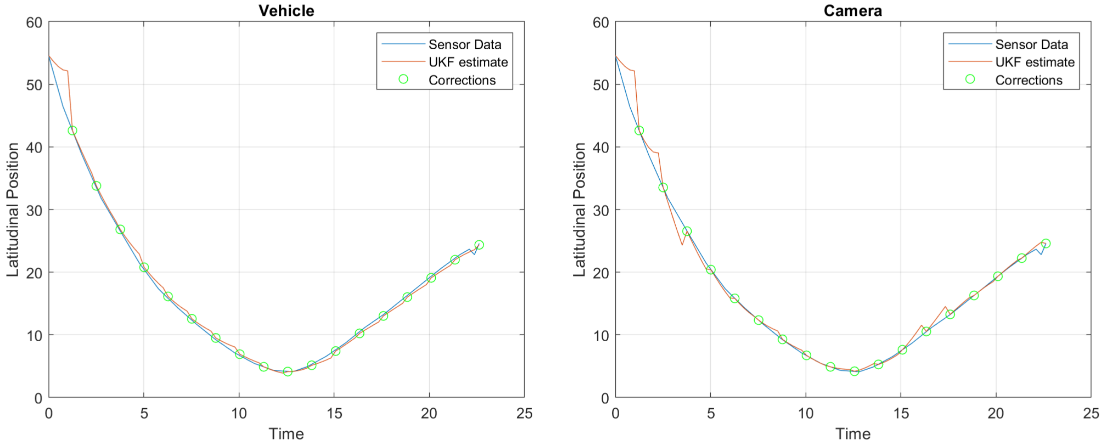


Figure 3.17: Latitudinal position comparison ($ph = 5$)

The speed behaviour for this second test is represented in Figure 3.18. In addition to the previously described aspects regarding the speed behaviour, which explain the constant variation for this state space variable, it is possible to see also how the performance for the camera case reveals to be improved with respect to the first test, even if it still remains lower in relation to the simulations performed studying the vehicle point of view.

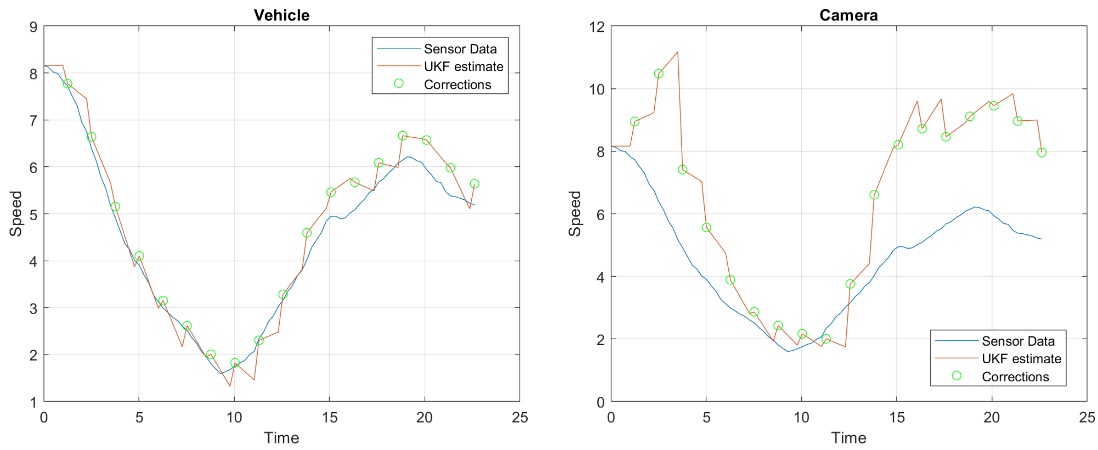


Figure 3.18: Speed comparison ($ph = 5$)

The values assumed by the acceleration are illustrated in Figure 3.19, where results clear how the corrections performed by the UKF in both the tests appear to be improved with respect to the previous test, obtaining a trend for the analysed variable which is closer to the authentic data. However, the performance offered by the vehicle case remains superior.

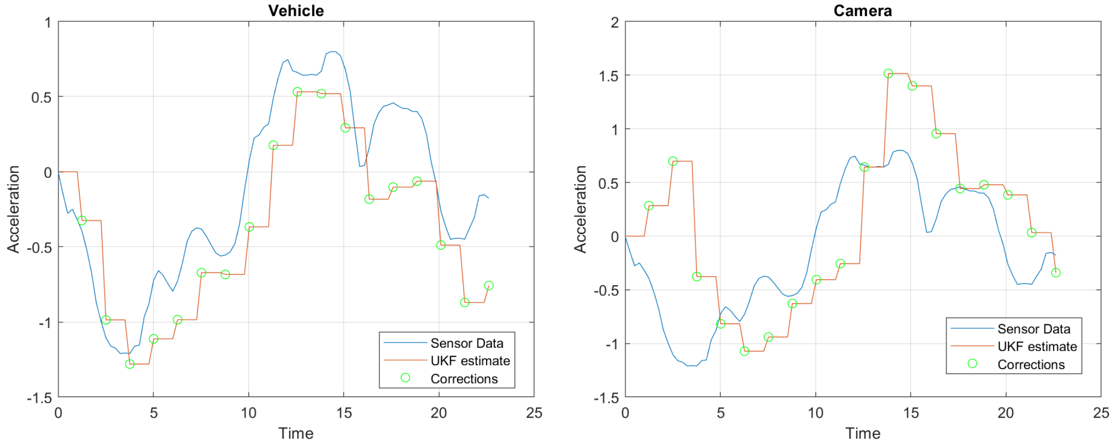


Figure 3.19: Acceleration comparison ($ph = 5$)

Also the behaviour of the heading variable, illustrated in Figure 3.20, is improved in this second test with $ph = 5$.

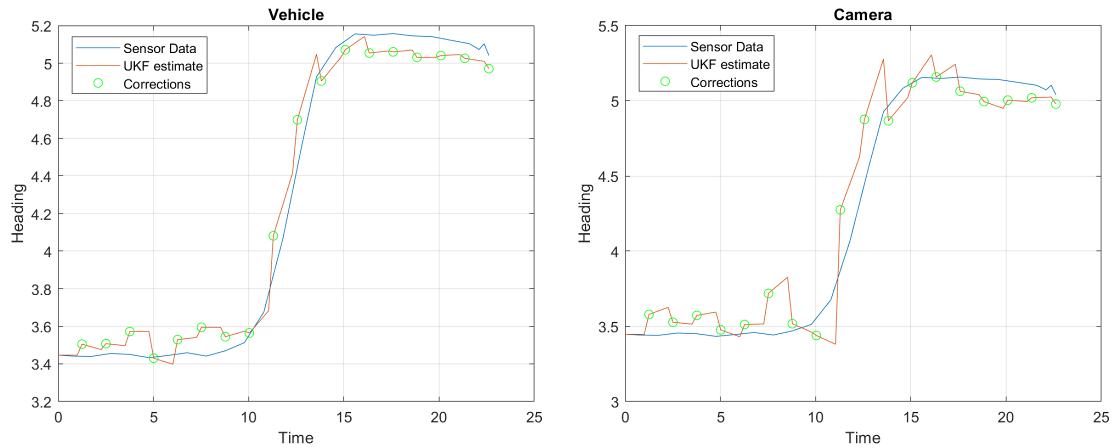


Figure 3.20: Heading comparison ($ph = 5$)

The shape assumed by the heading predictions is strictly related to the one of the yaw rate, showed in Figure 3.21. In this second test, for both the scenarios analysed, the increase of the number of UKF interventions leads to have a shape of the variable which is able to better replicate also the intermediate peak, while this was not possible in the previous test due to the greater range between every UKF correction.

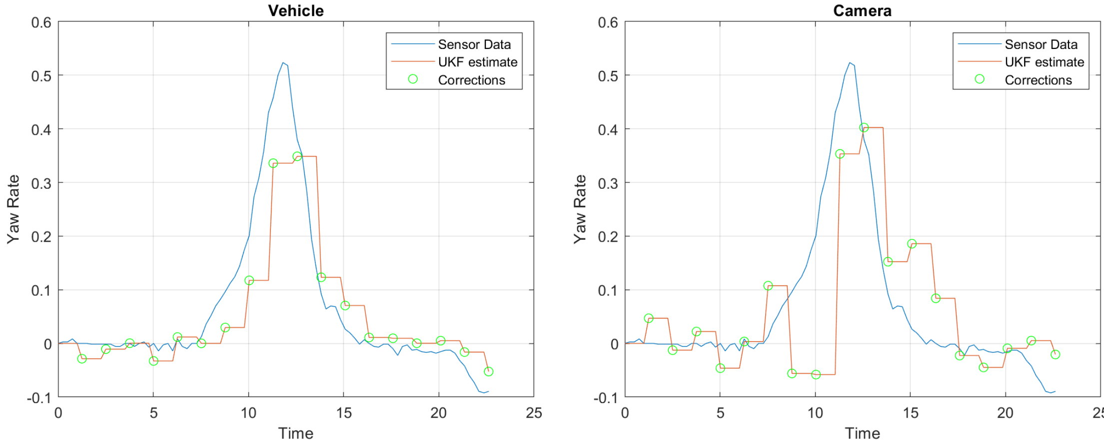


Figure 3.21: Yaw rate comparison ($ph = 5$)

The first two tests just shown result fundamental to understand two main concepts.

The first aspect that appears evident from Figures 3.8 and 3.15 is the superiority of the path prediction done in the vehicle case with respect to the camera one. In fact, both with a prediction horizon of ten and five iterations of the cycle, the trajectory prediction obtained in the vehicle case is closer to the authentic path, and this consideration is strictly related to the behaviour of the six state space variables shown in Figures from 3.9 to 3.14 and from 3.16 to 3.21.
Looking at the comparison between the mentioned state variables of the two cases, it is possible to note how the behaviours obtained in the vehicle case are able to better replicate the shape of the real information transmitted by the CAN bus. Thinking about the difference between the two cases explained in Section 3.2.2, this characteristic is attributable to the different definition of their measurement function. A prediction done from the vehicle point of view can rely on information about five of the six state space variables, and so the UKF can base the procedure explained in Sections 3.1.3 and 3.1.4 on five different variables distributions. The camera, instead, can rely only on the captured images of the vehicles, which provide information about longitudinal and latitudinal positions, and so the UKF, for this case, can work only on two different state distributions.
These aspects help to better understand the entity of the correction done by the UKF. In the vehicle case, in fact, when the UKF intervenes, the correction phase manages to bring back the simulated values close to the ones described by the sensor data, since it receives information about their distribution. In the camera case, instead, the correction is more effective for the first two state space variables, which information are provided by the captured images, while, especially for a large pre-

diction horizon, the filter is not able to appropriately correct the other state space variables, of which it does not receive information.

The second fundamental aspect that is possible to understand from the first two tests is related to the length of the prediction horizon. In the transition between the first and the second test, the overall performance for both vehicle and camera cases is improved. It is easier to note that the path prediction and the behaviour of the state space variables is more accurate if the model simulation is more helped by the UKF correction. This happens if the prediction horizon is reduced: the Figures from 3.15 to 3.21, which illustrate the scenario characterized by $ph = 5$, show that the path prediction done by the overall model results to be nearer to the authentic trajectory if the UKF has the possibility to intervene more frequently, and so to operate more corrections on the state. In this case, the model has the possibility to begin its simulation on the base of already corrected values, and so it succeeds in obtaining a more accurate trajectory.
If the prediction horizon is then increased, in the extra iterations of the cycle the model fatigue to obtain an adequate prediction, and this is attributable to the assumptions that characterize the chosen model. However, this aspect results clearer from the following tests.

3.3.2 Test 3: prediction and correction in parallel

In the third test there are two processes that go forward in parallel: for each iteration of the cycle, the UKF corrects the simulated state and, at the same time, the model tries to predict the future trajectory for the decided prediction horizon. In order to reason about the aspects concerning the assumptions of the CTRA model, the vehicle case response, with a prediction horizon of $ph = 10$, is analysed in Figure 3.22:

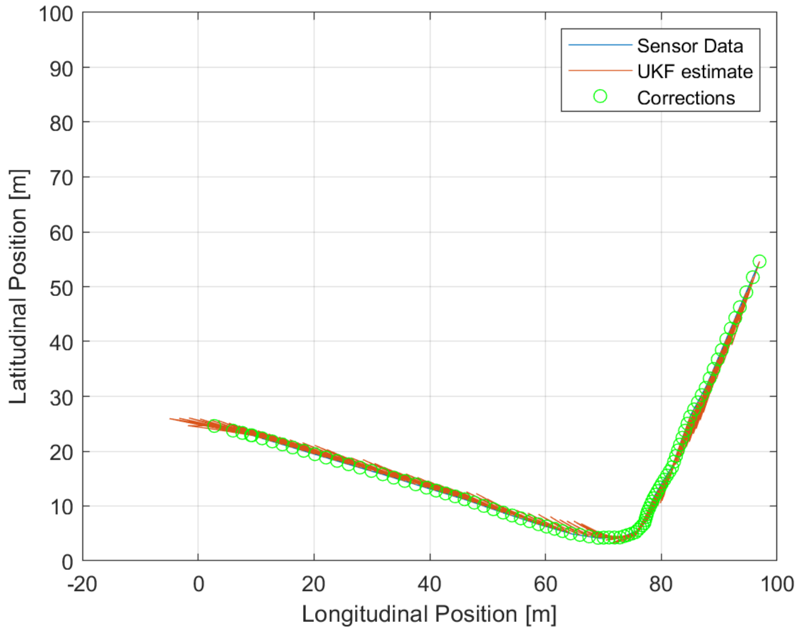


Figure 3.22: Correction and prediction in parallel

and the curve, which is the most critical point, is better visible in Figure 3.23:

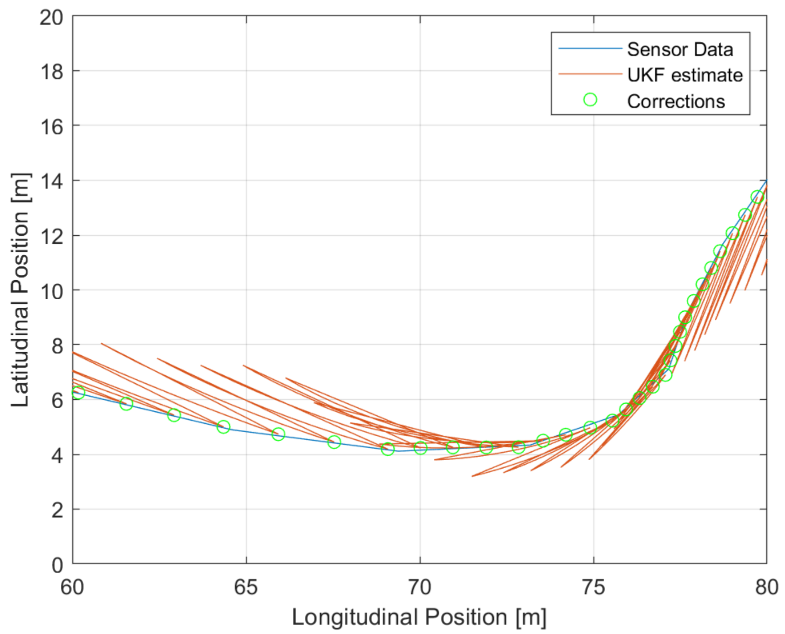


Figure 3.23: Correction and prediction in parallel, curve section

Observing this section, it is possible to understand another interesting aspect of the implemented model. As said several times, the chosen CTRA model is characterized by two assumptions, which are constant yaw rate and acceleration. If these two

variables are constant, it means that the speed and heading state space variables are characterized by a constant variation. This aspect characterizes the simulation done by the model for the overall prediction horizon and, if in the real situation the acceleration and yaw rate are not constant, it causes a prediction result that gets further away from the authentic trajectory.
Looking at Figure 3.23, where a prediction horizon of ten iterations is assumed, it is possible to note that, during the end of the curve, the prediction done by the model in a particular instant does not succeed in reaching the overall ten green points away from the starting one, where the green points represent the instants when the UKF intervenes and another iteration of the cycle begins. This happens because the model considers only the acceleration and yaw rate values that they have at the beginning of the simulation, and then it keep them constant, but in reality they are changing rapidly. Therefore, in situations like a real curve section where acceleration and yaw rate are characterized by a high rate of change, the assumptions of the CTRA model fatigue to remain valid for a large prediction horizon.

The aspects just described are confirmed in Figure 3.24, where, for each path prediction performed in this scenario with $ph = 10$, the mean of the errors between the predicted positions and the real ones is evaluated:

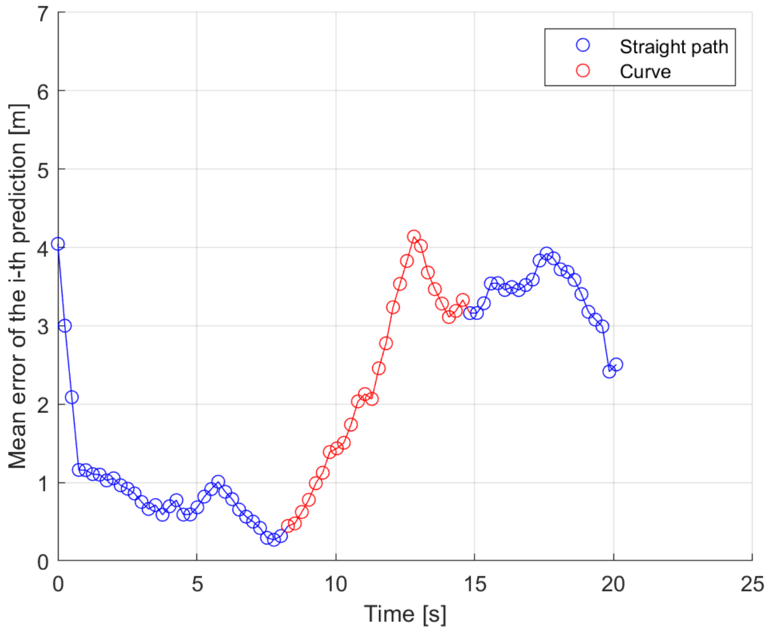


Figure 3.24: Mean errors of the path predictions ($ph = 10$)

Here two of the aspects previously defined are even clearer. The first one concerns the behaviour of the model when the curvilinear path has to be analysed: in fact, as it is possible to see from Figure 3.24, there is a slight increase of the prediction errors during the curve, and this is strictly related to the behaviour of the variables that are considered constant by the implemented model. During the curve we have the strong combined variation of both the acceleration and yaw rate, but, in the course of the trajectory prediction simulations, these are considered constant. For this reason, this third test provides the confirm that the assumptions proposed by the CTRA model are no longer sustainable when large prediction horizons, as $ph = 10$, are considered for the trajectory predictions.

Another confirm of the greater reliability of the model for short prediction horizons can be obtained comparing the scenario in Figure 3.24 with the one in Figure 3.25, where the prediction horizon is halved, and $ph = 5$ is considered.

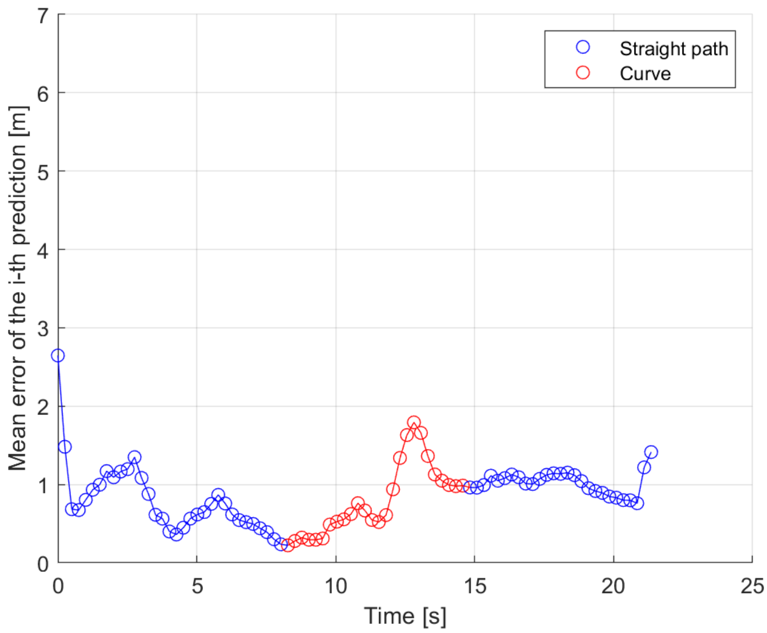


Figure 3.25: Mean errors of the path predictions ($ph = 5$)

Obviously, also in this case the curve sections is the one responsible of the larger errors, but comparing this situation with the previous one, it is clear how the reduction of the prediction horizon, and so the higher number of UKF corrections, slightly improves the path predictions in terms of errors with respect to the previous scenario.

3.3.3 Test 4: extension of the analysis to the overall data

In the fourth and last test, the explained combination of the overall model and the UKF is extended to all the available data provided by LINKS Foundation. In particular, the same vehicle case with $ph = 10$ of the previous test is applied to the particular and interesting sections highlighted in Figure 3.26, which are a long rectilinear path and an area characterized by two successive right and left curves:

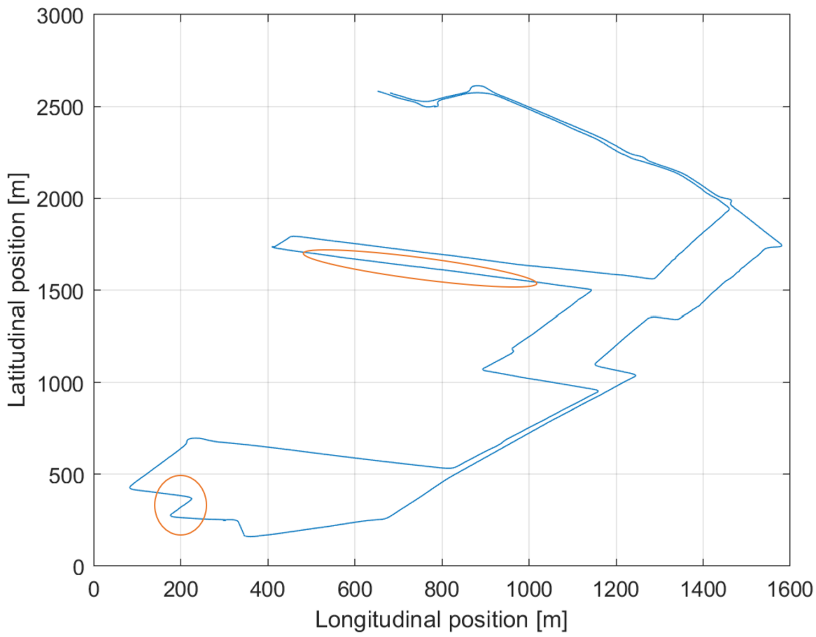


Figure 3.26: Highlighted sections of the overall path

A fraction of the rectilinear path highlighted in Figure 3.26 is shown in Figure 3.27:

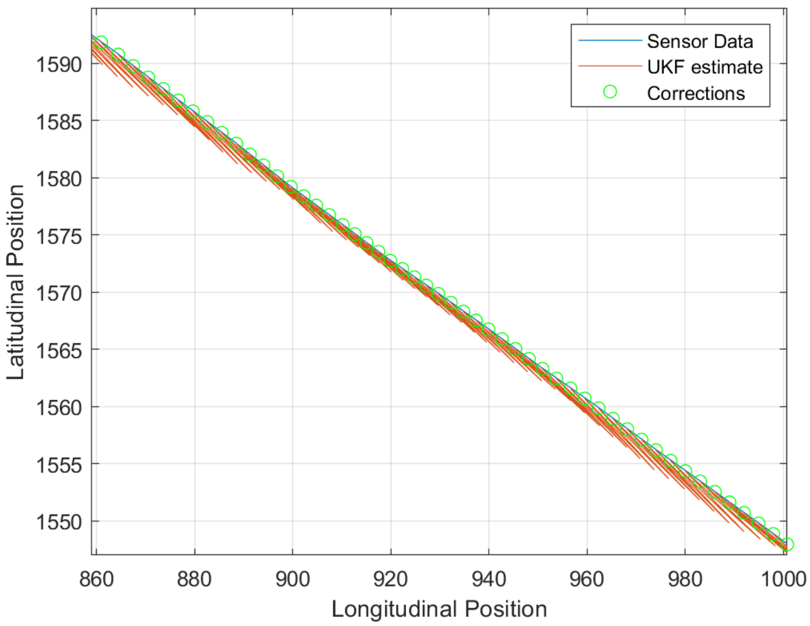


Figure 3.27: Rectilinear curve section

while the results for the double curve section are illustrated in Figure 3.28:

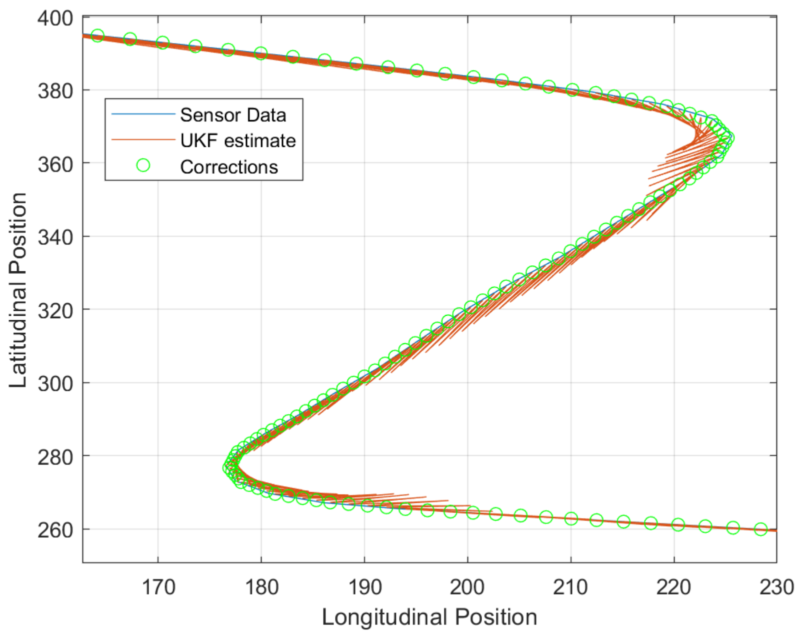


Figure 3.28: Double curve section

From this last test it is possible to confirm how the created model is actually able to describe both a rectilinear and a curve section, properly calculating its future trajectory switching between the straight motion equations and the characteristic relations of the CTRA model, following the condition imposed on the yaw rate.

However, this last attempt also confirms the aspects found in the third test. In fact, here a prediction horizon of ten iterations of the cycle is considered, and this large anticipation has an effect on the accuracy of the prediction due to the assumptions of the CTRA model. Looking for example at the curve sections, as for the previous test, it is clear how the consideration of constant acceleration and yaw rate becomes an obstacle for longer predictions horizons, leading to have a prediction that in the last simulated time instants moves away from the authentic trajectory.

Therefore, summing up the aspects determined in the last two tests, it is possible to understand and conclude that the implemented CTRA model is more suitable for short - term trajectory prediction scenarios, since, especially for situations where the acceleration and the yaw rate are characterized by high rates of change, its characteristic assumptions begin to be a weight for long - term trajectory prediction purposes.

Chapter 4

Conclusions

The fourth and last chapter of this thesis project is composed by an overview of all the results obtained during the development of the model and the analysis of the different tests performed studying its combination with the UKF. All the strengths and weaknesses of the proposed implementation are reported, followed by a final reasoning about possible future works that could improve the results obtained in this thesis work.

4.1 Conclusions and future works

This thesis project is born as a response to the will of studying the cooperation between an autonomous and connected vehicle and an infrastructure, which has the capability to observe the movements of both connected and non - connected vehicles. The information coming from the infrastructure permit to compare the predicted trajectory of the connected vehicle with the movements of the non - connected ones. In this way, if the system detects possible collisions, it is possible to intervene in time to avoid them and continue the path safely.

After a preparatory work on the provided data, which has allowed to obtain a set of variables concordant both with the model and the simulation program, the performances of the CTRA model have been studied in a scenario representing a real road intersection. The tests performed in this section have established that, thanks to the combination of the straight motion equations and the CTRA relations, which is guided by the condition imposed on the yaw rate variable, the implemented model is able to replicate the shape of the trajectory of the examined vehicle both in rectilinear and curve sections.

Then, the implemented model has been combined with the UKF in order to perform several tests regarding the prediction of the vehicle trajectory, and three main conclusions have been drawn.
In the first two tests, where both the vehicle and camera points of view have been analysed, the trajectory prediction performed by the connected vehicle turns out to be more accurate than the camera one, and this is strictly related to the number of different variables that it can monitor during its path. The UKF correction and, consequently, the following prediction phase emerged to be more precise if based on a higher number of state space variables, and so on an higher number of data sets that can be studied in order to perform a better correction of the simulated state.
The second fundamental result regards the prediction horizon, that is the number of cycle iterations or time instants for which the model is simulated in an open chain situation. In both the studied cases, the accuracy of the calculated trajectory prediction results to be directly proportional to the length of the prediction horizon and, in particular, it offers much better results if the UKF has the possibility to intervene and correct the simulated state more frequently. This means that, if the prediction horizon is reduced, the corrections performed by the UKF are closer to each other, and the overall performance of the model benefits of this situation.
The third and last conclusion of this thesis work is stricty related to the assumptions

of the CTRA model. In fact, its main relations consider both the acceleration and the yaw rate as constants, and these assumptions are directly linked to the speed and heading behaviour, which are then characterized by a constant variation. In situations where the acceleration and the yaw rate are characterized by a high rate of change, the assumptions proposed by the CTRA model fatigue to remain valid, especially for long prediction horizons. This last conclusion reveals that the usage of this type of model is more recommended for short - term prediction scenarios.

The obtained results can be a good base for further improvements regarding trajectory prediction scenarios like the one analysed in this work.
In particular, since the performed simulations have shown a decrease in accuracy in the camera case with respect to the vehicle one, a possible future work could treat an improvement of the infrastructure capabilities by introducing sensors able to obtain additional data other than the position ones.
Another possible implementation could be directed towards improving the main drawback of the CTRA model, which regards its assumptions on acceleration and yaw rate. In particular, the implemented model could be enhanced by considering this time variable acceleration and yaw rate, while their derivatives are maintained constant. In this way, an overall model with more degrees of freedom than the one implemented in this project could be obtained.

Bibliography

[1] NHTSA (National Highway Traffic Safety Administration). Automated vehicles for safety, 2018.

[2] Oscar Amador, Ignacio Soto, Manuel Urueña, and Maria Calderon. Got: Decreasing dcc queuing for cam messages. *IEEE Communications Letters*, 24(12):2974–2978, 2020.

[3] Keshav Bimbraw. Autonomous cars: Past, present and future a review of the developments in the last century, the present scenario and the expected future of autonomous vehicle technology. In *2015 12th international conference on informatics in control, automation and robotics (ICINCO)*, volume 1, pages 191–198. IEEE, 2015.

[4] Ahmed Eltrass and Mohammed Khalil. Automotive radar system for multiple-vehicle detection and tracking in urban environments. *IET Intelligent Transport Systems*, 12(8):783–792, 2018.

[5] Simon Haykin. *Kalman filtering and neural networks*, volume 47. John Wiley & Sons, 2004.

[6] K. Jo, J. Kim, D. Kim, C. Jang, and M. Sunwoo. Development of autonomous car - part i: Distributed system architecture and development process. *IEEE Transactions on Industrial Electronics*, 61(12):7131–7140, 2014.

[7] Jason Kong, Mark Pfeiffer, Georg Schildbach, and Francesco Borrelli. Kinematic and dynamic vehicle models for autonomous driving control design. In *2015 IEEE Intelligent Vehicles Symposium (IV)*, pages 1094–1099. IEEE, 2015.

[8] Che Kun Law, Darshit Dalal, and Stephen Shearrow. Robust model predictive control for autonomous vehicles/self driving cars. *arXiv preprint arXiv:1805.08551*, 2018.

[9] Stéphanie Lefèvre, Dizan Vasquez, and Christian Laugier. A survey on motion prediction and risk assessment for intelligent vehicles. *ROBOMECH journal*, 1(1):1–14, 2014.

[10] Nikita Lyamin, Alexey Vinel, Magnus Jonsson, and Boris Bellalta. Cooperative awareness in vanets: On etsi en 302 637-2 performance. *IEEE Transactions on Vehicular Technology*, 67(1):17–28, 2017.

[11] MathWorks. Convert angle from degrees to radians - *deg2rad*, 2020.

[12] MathWorks. Convert spherical distance from degrees to kilometers - *deg2km*, 2020.

[13] MathWorks. Correct state and state estimation error covariance using extended or unscented kalman filter, or particle filter and measurements - *correct*, 2020.

[14] MathWorks. Extended and unscented kalman filter algorithms for online state estimation, 2020.

[15] MathWorks. Nonlinear state estimation using unscented kalman filter and particle filter, 2020.

[16] MathWorks. Polynomial curve fitting - *polyfit*, 2020.

[17] MathWorks. Predict k - step - ahead model output - *predict*, 2020.

[18] MathWorks. Understanding kalman filters, nonlinear state estimators, 2017.

[19] Richard J Meinhold and Nozer D Singpurwalla. Understanding the kalman filter. *The American Statistician*, 37(2):123–127, 1983.

[20] SAE International (Society of Automotive Engineers). Taxonomy and definitions for terms related to driving automation systems for on-road motor vehicles, 2018.

[21] Philip Polack, Florent Altché, Brigitte d'Andréa Novel, and Arnaud de La Fortelle. The kinematic bicycle model: A consistent model for planning feasible trajectories for autonomous vehicles? In *2017 IEEE intelligent vehicles symposium (IV)*, pages 812–818. IEEE, 2017.

[22] Rajesh Rajamani. *Vehicle dynamics and control.* Springer Science & Business Media, 2011.

[23] Robin Schubert, Eric Richter, and Gerd Wanielik. Comparison and evaluation of advanced motion models for vehicle tracking. In *2008 11th international conference on information fusion*, pages 1–6. IEEE, 2008.

[24] Daniel Svensson. Derivation of the discrete-time constant turn rate and acceleration motion model. In *2019 Sensor Data Fusion: Trends, Solutions, Applications (SDF)*, pages 1–5. IEEE, 2019.

[25] Manolis Tsogas, Aris Polychronopoulos, and Angelos Amditis. Unscented kalman filter design for curvilinear motion models suitable for automotive safety applications. In *2005 7th International Conference on Information Fusion*, volume 2, pages 8–pp. IEEE, 2005.

[26] Balázs Várszegi, Dénes Takács, and Gábor Orosz. On the nonlinear dynamics of automated vehicles–a nonholonomic approach. *European Journal of Mechanics-A/Solids*, 74:371–380, 2019.

[27] Massimo Violante. Technologies for autonomous vehicles, lecture notes.

[28] Greg Welch, Gary Bishop, et al. An introduction to the kalman filter. 1995.

[29] Wei Xiao, Lijun Zhang, and Dejian Meng. Vehicle trajectory prediction based on motion model and maneuver model fusion with interactive multiple models. *SAE International Journal of Advances and Current Practices in Mobility*, 2(2020-01-0112):3060–3071, 2020.

[30] Guotao Xie, Hongbo Gao, Lijun Qian, Bin Huang, Keqiang Li, and Jianqiang Wang. Vehicle trajectory prediction by integrating physics-and maneuver-based approaches using interactive multiple models. *IEEE Transactions on Industrial Electronics*, 65(7):5999–6008, 2017.