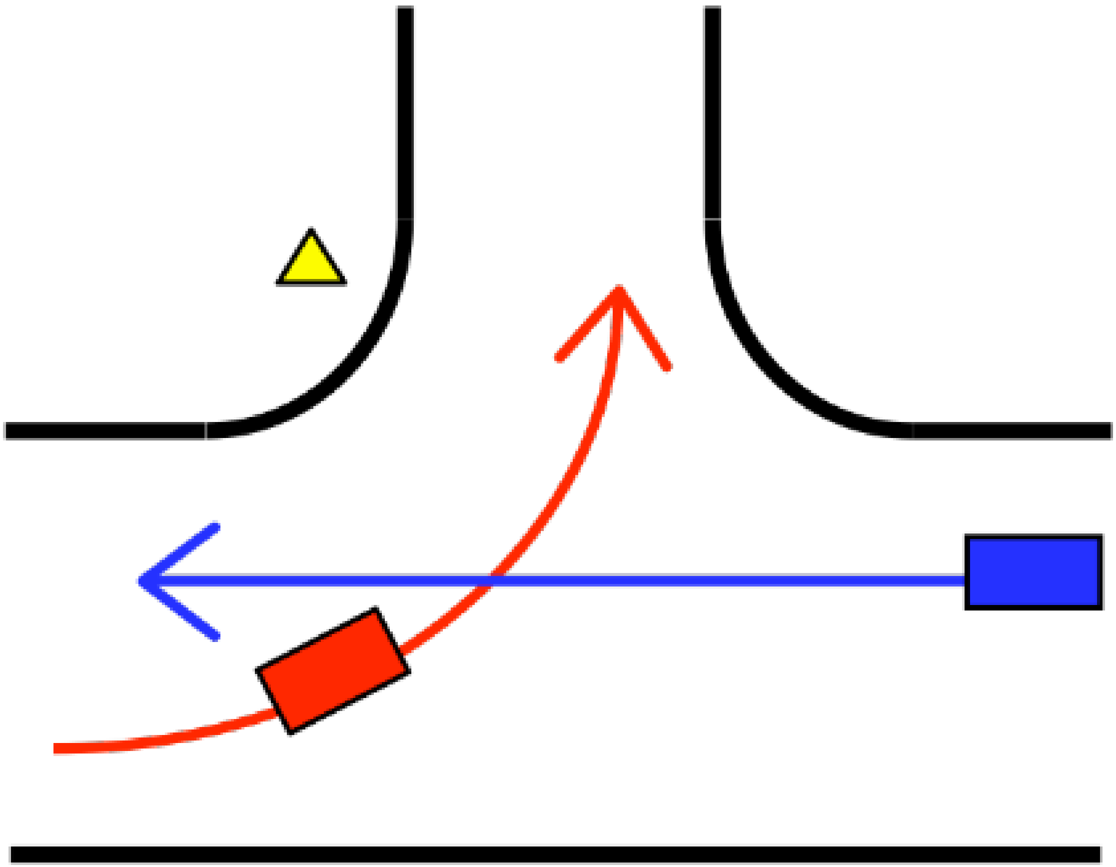

Field assessment of driver decision making at intersections

A real-time wireless application to manipulate encroachment time

Master's thesis in Automotive Engineering

CHRISTIAN-NILS BODA
JUAN CAMILO MUÑOZ CANTILLO

Department of Applied Mechanics
Division of Vehicle Safety
CHALMERS UNIVERSITY OF TECHNOLOGY
Göteborg, Sweden 2013

Master's thesis 2013:15

MASTER'S THESIS IN AUTOMOTIVE ENGINEERING

Field assessment of driver decision making at intersections

A real-time wireless application to manipulate encroachment time

CHRISTIAN-NILS BODA
JUAN CAMILO MUÑOZ CANTILLO

Department of Applied Mechanics
Division of Vehicle Safety
CHALMERS UNIVERSITY OF TECHNOLOGY

Göteborg, Sweden 2013

Field assessment of driver decision making at intersections
A real-time wireless application to manipulate encroachment time
CHRISTIAN-NILS BODA
JUAN CAMILO MUÑOZ CANTILLO

Master's thesis 2013:15
ISSN 1652-8557
Department of Applied Mechanics
Division of Vehicle Safety
Chalmers University of Technology
SE-412 96 Göteborg
Sweden
Telephone: +46 (0)31-772 1000

Chalmers Reproservice
Göteborg, Sweden 2013

Field assessment of driver decision making at intersections
A real-time wireless application to manipulate encroachment time
Master's thesis in Automotive Engineering
CHRISTIAN-NILS BODA
JUAN CAMILO MUÑOZ CANTILLO
Department of Applied Mechanics
Division of Vehicle Safety
Chalmers University of Technology

Abstract

Accidents at intersections are of the most common causes of fatalities on roads. Statistics from the EU and the USA show that fatalities at intersections represent more than 20% of all traffic fatalities. A left turn across the path of a vehicle coming from the opposite direction (LTAP/OD) is one of the riskiest scenarios at intersections. In this scenario, we can refer at the vehicle turning left as the Subject vehicle and to the one going straight as Confederate vehicle. In order to understand driver behavior in this scenario during field trials, the interaction between the two vehicles should be manipulable by the experimenters.
In this study, a system able to manipulate LTAP/OD scenarios by controlling the difference in time-to-intersection between the Subject and the Confederate vehicles has been developed. By means of a human machine interface installed in the Confederate vehicle, an experimenter driver was instructed about the velocity he/she should follow to control the difference in time-to-intersection with respect to the Subject vehicle. This velocity instruction was determined from 1) the position of the vehicles, 2) their kinematics and 3) historical intersection data. The system comprised of 1) a single board computer with one GPS device and one 3G modem for each vehicle, 2) a cloud application, 3) a computational server and 4) a computer to render the human machine interface. The algorithm providing velocity instruction addressed six experimental phases – waiting, be ready, start-up, regulation, stabilization and releasing. This algorithm relied on predictions: profiles extracted from historical intersection data were used to estimate the current time-to-intersection, which was the base for the whole evaluation. The novelty of this algorithm consisted in estimating in real-time the optimal velocity that Confederate vehicle needed to follow to control for a certain difference in time-to-intersection with the Subject vehicle. The velocity estimation was based on wireless communication of GPS position and velocity. The algorithm was validated in the real world by showing that the actual difference in time-to-intersection achieved by following the velocity instruction from the algorithm was very close to the one aimed for. Furthermore, the system used in this study is affordable and accessible to anyone; therefore this system can be easily reproduced and employed to understand driver behavior at intersection as well as for developing active safety system to support other road users at intersections.

Keywords: Time-to-arrival algorithm, Trailing buffer, Post Encroachment Time, Driver behavior, Intersection, Field assessment, Wireless, Cooperative, Application, Human-Machine Interface, Historic profiles, Distance to intersection, Time to intersection

Preface

This thesis work has been carried out as a partial requirement for the Master of Science degree in Automotive Engineering at Chalmers University of Technology in Gothenburg, Sweden. All the stages of the project were performed at SAFER, Vehicle and Traffic Safety Centre at Chalmers University of Technology during January – June 2013. The testing stage of the project was carried out with the help of Volvo Car Corporation.
We would like to acknowledge and thank our supervisor Jonas Bärgman and our examiner Marco Dozza for their constant guidance and support along the development of this project. Thanks also to Andre Fernandez for his support in configuring and putting together the system, and to Julia Werneke and Kip Smith for their valuable comments and suggestions. Our words of gratitude also to the people at Volvo Car Corporation for their help during the testing of the system; Regina and Pelle, and to SAFER for providing a pleasant space to conduct our work.

Christian-Nils would like to acknowledge his fiancée, his parents, his sister and his friends. Thanks to my friends for your support during these years in Sweden and having made this period even more enjoyable than it was possible. Thanks to my parents and my sister for your eternal support and having made this Swedish period of my life realizable. All my love for You for having supported me during these two years and having been patient with me. I dedicate this work to You.

Juan Camilo would also like to acknowledge his wife and parents for their permanent support throughout this process. To my wife: thank you for encouraging me when times were difficult and for celebrating with me when times were better; your unconditional love has made this process even more rewarding. A mis padres: gracias por el gran esfuerzo que realizaron para que yo cumpliera este objetivo y por su permanente apoyo desde la distancia.

Gothenburg, June 2013.

Christian-Nils Boda, Juan Camilo Muñoz

Contents

1 Introduction

1.1 Background

Accidents at intersections are the second most common cause of fatalities on roads. Statistics from the EU and the USA show that fatalities at junctions represent more than 20% of all traffic fatalities. In the European Union, fatalities at intersections have remained relatively constant between 2000 and 2009, where this percentage has only fluctuated between 20% and 22% [1]. These numbers are comparable to the USA, where in 2011, fatalities at intersections represented 21.63% and in 2010, 22.31% [2, 3].

The causes of accidents at junctions are numerous, but research suggest that in Europe around 60% of the incidents are related to an inappropriate timing from the driver (e.g. premature, late, or no action) [1]. Additionally, inappropriate timing could be linked to other causes such as faulty diagnosis, information failure, observation missed, or inadequate plan, among others [1]. In the case of accidents at junctions the problem is usually that the driver who encroaches the right-of-way vehicles path makes the wrong decision. As in several previous studies, this study is focused on the left turn situations, or more specifically, the left-turn-across-path-opposite-direction (LTAP/OD) (Figure 1.1) since this situation is likely to have more encroachments than other situations and since the crashes for this situation represent a big part of the intersection-related crashes [4]. As a result, it turns out to be one of the riskiest situations at intersections [5]. The car which turns is defined as the Subject Vehicle (SV) while the ones which is going straight is considered as the Principal Opponent Vehicle (POV). In this project, the POV is considered as the Confederate vehicle as its driver will be instructed by experimenters in order to manipulate the intersection scenario.

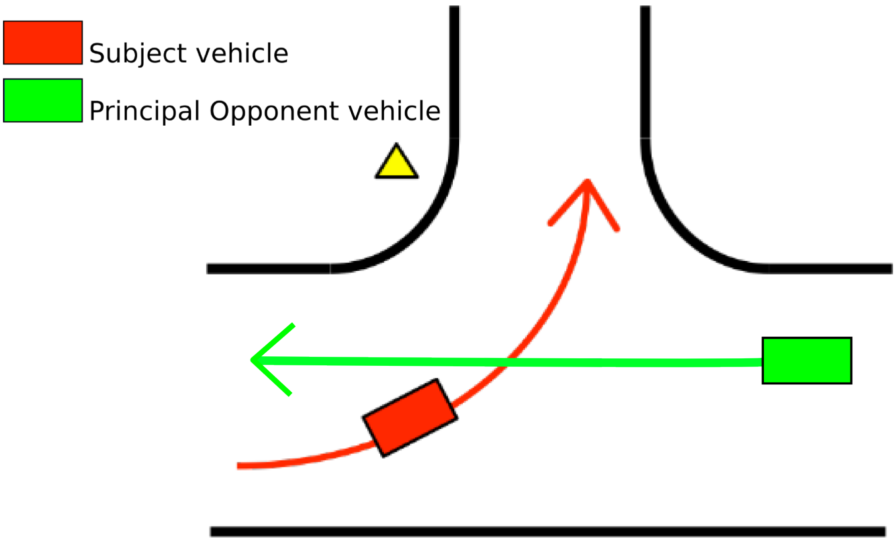

Figure 1.1: *Left turn across path from the opposite direction (LTAP/OD)*

One solution amid others would be to use driver assistance systems in intersection; i.e. intersection decision support (IDS). These systems are aimed at providing the needed assistance to avoid making the wrong Go/No-Go decision [6].

1.1.1 Criteria in intersection decision support systems

The development of these systems is far from being straight forward since in intersection situations (and in most of road safety situations) the driver behavior, as the dynamics, should be taken into account [7, 8]. It is necessary to identify and quantify in what situations drivers would accept the planned intersection supports, i.g. warnings or interventions. To be more specific, some setting parameters have to be defined in order to evaluate the situations. These parameters, also called *safety thresholds*, can differ from one IDS design to another. Commonly, this *safety threshold* corresponds to a gap between vehicles. This gap can be expressed as a distance or as a time. The safety threshold chosen for the IDS design can be the temporal gap between two vehicles on the major-road to determine when it is safe for an opponent car to turn at the intersection (Figure 1.2)

[8–10]. Another safety threshold can be the trailing buffer [11]. This so called trailing buffer corresponds to the time measured from the moment when the SV passes the point of conflict to the moment when the POV (Principal Opponent Vehicle) reaches the same point. Finally, the PET (Post-Encroachment Time) criterion is commonly used to characterize *a posteriori* how risky the encroachment was. This criterion is defined as the time measured from the moment in which the SV leaves the encroachment zone (Figure 1.3 – t1) to the moment in which the POV enters this zone (Figure 1.3 – t2) [12].

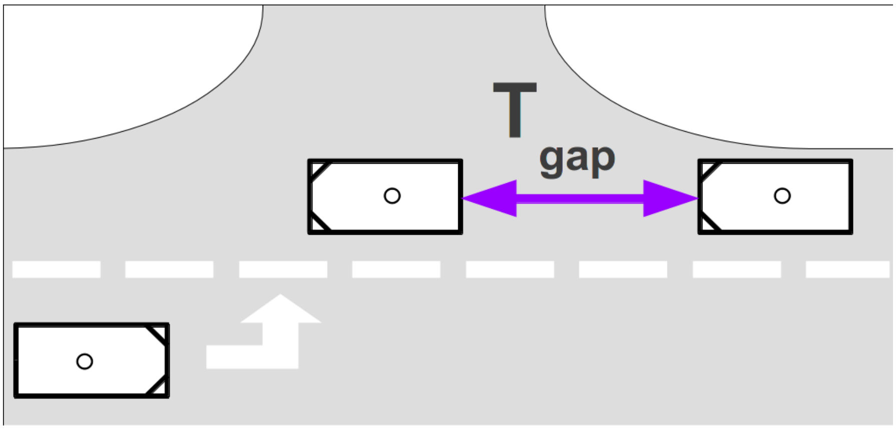

Figure 1.2: *gap definition sketch*

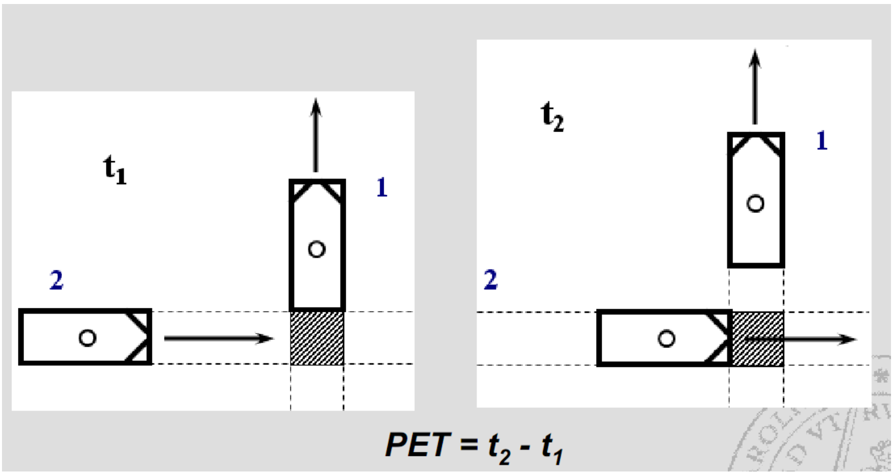

Figure 1.3: *PET definition sketch [13]*

1.1.2 Arrival time evaluation algorithms

The previously defined criteria are all about time closely related to the time to intersection (TTI). Chan defined the TTI as the distance to the intersection divided by the instantaneous velocity. This is a definition which can be more or less true but which cannot be used for this present study because the velocity is not constant during all the travel. That is why another way to determine precisely the time to intersection should be investigated. For example, the bus time-to-arrival algorithms determine the remaining time before arriving to a given bus stop. This time-to-arrival which can correspond to a time to intersection is there not defined linearly. Different studies [14–16] propose ways to get an accurate time-to-arrival. But usually, a schedule table is generated from historic bus journeys and then an algorithm uses it with the current bus position and the current time of the day to define the probable time-to-arrival. This is the base of most of time-to-arrival algorithms for buses.

Even though it exists many time-to-arrival studies for one vehicle (in these cases a bus), no study dealing with time-to-arrival for one vehicle related to another ongoing vehicle has been found.

1.1.3 Current Global Navigation Satellite Systems (GNSS)

Different positioning systems exist today worldwide. One of the most common and powerful of these technologies is the satellite based technology known as Global Navigation Satellite System (GNSS). Within this system some of the solutions currently available are global, meaning they are capable of operating anywhere in the world; while some others are regional, meaning they are only operational in a limited geographical region. For applications with higher precision requirements, these GNSS systems are commonly aided by either Satellite Based Augmentation Systems (SBAS) or by Ground Based Augmentation Systems (GBAS). These augmentation systems help overcome most of the common error sources for GNSS positioning, which include multipath and atmospheric errors, as well as satellite and receiver related biases [17].

Global navigation systems consist of three basic components to enable positioning [17–19]: the network of satellites that orbit the earth, the infrastructure on earth that continuously controls the status and operation of the satellites, and finally the different receivers owned by the users that capture the signals broadcasted by the satellites. There are currently two of these systems in operation: GPS, developed by the Department of Defense of the U.S.; and GLONASS developed by Russia. There are also two other systems under development: GALILEO, developed by the European Union and expected to be in operation by 2014; and COMPASS, developed by China and expected to be operative by 2015 [17].

Satellite based augmentation systems provide corrections for the GNSS measurements by means of additional messages broadcasted by the satellites. These messages contain information regarding the disturbances associated to the typical sources of error in order to correct them. The monitoring and dispatch of these messages are originally performed by several ground stations located at highly accurate surveyed locations. Examples of these systems include the WAAS for North America, EGNOS for Europe, MSAS for Japan and GAGAN for India [17].

Ground based augmentation systems provide improvements for the GNSS measurements by broadcasting corrections focused on a specific area. These corrections can be sent from ground stations via radio, cellular or internet communication. Among the more popular of these augmentation systems are differential GPS (DGPS) and Real time kinematics networks (RTK or RTN). Detailed information on how these systems work can be found in [17].

While single recreational-type GNSS units with no augmentation aids can provide accuracies of between 5 to 20 meters, DGPS systems can improve accuracy to around 0.5 meters with not extremely expensive solutions. On the other hand, RTK enabled units usually are highly expensive but can improve the accuracy of the measurement up to centimeter or even millimeter level for applications that so require it.

GPS technology is becoming more frequently used in automotive applications, not only for navigation support or experiment purposes but also for automotive safety systems [20].

1.1.4 Current vehicle-to-vehicle communication systems

Communication in current vehicles is a field that is rapidly evolving. This includes communication within the vehicle, for instance communication between a user's mobile phone or music device and the entertainment system of the vehicle; but also communication between the vehicle and external actors, for example the road infrastructure (known as vehicle-to-infrastructure communication) or other vehicles (known as vehicle-to-vehicle communication).

The most important vehicle-to-infrastructure and vehicle-to-vehicle communication technologies in use today are cellular, Wi-Fi (802.11), and 5.9 GHz Dedicated Short Range Communication (DSRC) [21].

Cellular communication is based on the existing cellular networks. They thus have the advantage of having a high coverage, only limited by the coverage of the network as such. It also has an implemented and well developed infrastructure, which makes it easier to use. However, it does not provide direct links between the devices, which renders in uncertain latencies [21].

Wi-Fi communication, which is based on the 802.11 protocol, has the possibility of providing high-speed broadband internet access and also has the advantage of providing direct links between the communicating devices, which renders in low latencies. On the other hand, they have the disadvantage of being susceptible to possible interference from other in-band users, and also to obstruction from obstacles on the road or near it. Additionally, the infrastructure needs to be implemented when this type of communication is used (i.e. the transceivers and transponders need to be installed and the network needs to be configured) [21–23].

5.9 GHz DSRC communication is a technology developed exclusively for automotive safety applications. It thus has dedicated channels and spectrum for its use, which renders in benefits regarding possible interference

Table 1.1: Summary of features for the considered communication technologies

Communication technology	Features
Cellular	• High coverage. Only limited by cellular network coverage • Already implemented infrastructure • No direct links between the devices, which renders in uncertain latencies
Wi-Fi	• High-speed broadband internet access • Possible interference from other in-band users • The infrastructure needs to be implemented • Direct link between the devices, which renders in low latencies
5.9 GHz DSRC	• Dedicated channels to automotive safety • Dedicated spectrum • High speed broadband links • The infrastructure needs to be implemented • Direct link between the devices, which renders in low latencies

from other users. It can also provide high-speed broadband links and has direct links between the communicating devices, which renders in low latencies. However, as for the Wi-Fi communication, it has the disadvantage of requiring implementation of the infrastructure when intended to be used [21, 24, 25].

Table 1.1 shows a summary of the features of the existing vehicle-to-vehicle communication. This table is an adaptation from the table elaborated by Gallagher and Akatsuka [21].

1.2 Objectives

The present study aimed to design an application which provides velocity instruction to the confederate driver in order to manipulate the encroachment time in intersections. This time to be controlled depends on how two vehicles interact at an intersection. An algorithm has been developed based on the theory of bus time-to-arrival algorithms. The system has been optimized for the LTAP/OD scenario. The confederate driver knew the details of the study and the intended interaction, while the subject driver was unaware of the interaction. The interaction should be as natural as possible, as if two on-road vehicles were meeting at the intersection as part of everyday driving. The behavior of the subject driver will be recorded for the purposes of the project DCBIN (Driver Comfort Boundaries in Intersection Negotiation), of which the present study is a part of. The developed system is a close to real-time system based on the position and current kinematic of both vehicles which informed the confederate driver when to start driving and what velocity to strive for.

These objectives brought different problematics such as:

"How to manipulate an intersection scenario using a system developed on current and accessible technologies?"

"How to estimate the time-to-intersection of a car at a given moment?"

1.3 Scope

The present study aimed to develop an application that helps to provoke a specific traffic situation for the purpose of studying driving behavior under this scenario. The application was intended only for the LTAP/OD situation, and was not proposed to force safety critical situations upon the drivers, but rather normal driving situations with different timings. Furthermore, the application stopped sending instructions to the confederate driver just before reaching the intersection, so that he / she was free from distractions and could take any actions considered appropriate given the traffic conditions. The study just focused on manipulating difference in time to intersection instead of trailing buffer parameter in order to simplify the system development. Finally, it is worth mentioning that the present study did not perform any analysis of driver behavior whatsoever, as this is a task that is part of the parent DCBIN project.

2 Method

2.1 System definition

2.1.1 Study scenario

The present study scenario is composed of three things: the *place*, the *actors* and the *objectives*. The scenario takes place at intersections, and, more precisely, at intersections where one driver is turning left accross the path of the opponent car. The name of this scenario is called LTAP/OD for “Left Turn Across Path from the Opposite Direction” (Figure 1.1). The actors are the two drivers. The driver who turns left is defined as the Subject and the opponent driver is the Confederate driver. The objectives of this scenario are about to make the two drivers interact at the intersection to analyze the Subject’s behavior, the dynamics of the situation, and more analyses topic left to be developed by experimenters.
In order to get the two drivers at the time defined by the experimenters in the intersection, a system has been developed in this project to instruct the Confederate driver what velocity he/she should follow.

2.1.2 Use case

The use case shown in Figure 2.1 represents the interaction between the different actors of the scenario. The black box corresponds to the overall system and the other shapes are the actors which play a specific role in this system. The two drivers’ cars are symbolized by the car shape. The two computer shapes represent the two devices which intervene in the system to execute a task.
The position of the cars and their velocity are the key parameters of the system. The configured devices embedded in the cars send the current kinematic to the system. This one computes the corresponding optimal speed to be provided to the Confederate driver. As soon as this result is ready to be sent, it is transmitted to the HMI which is installed in the confederate car and displays the instruction to its driver.

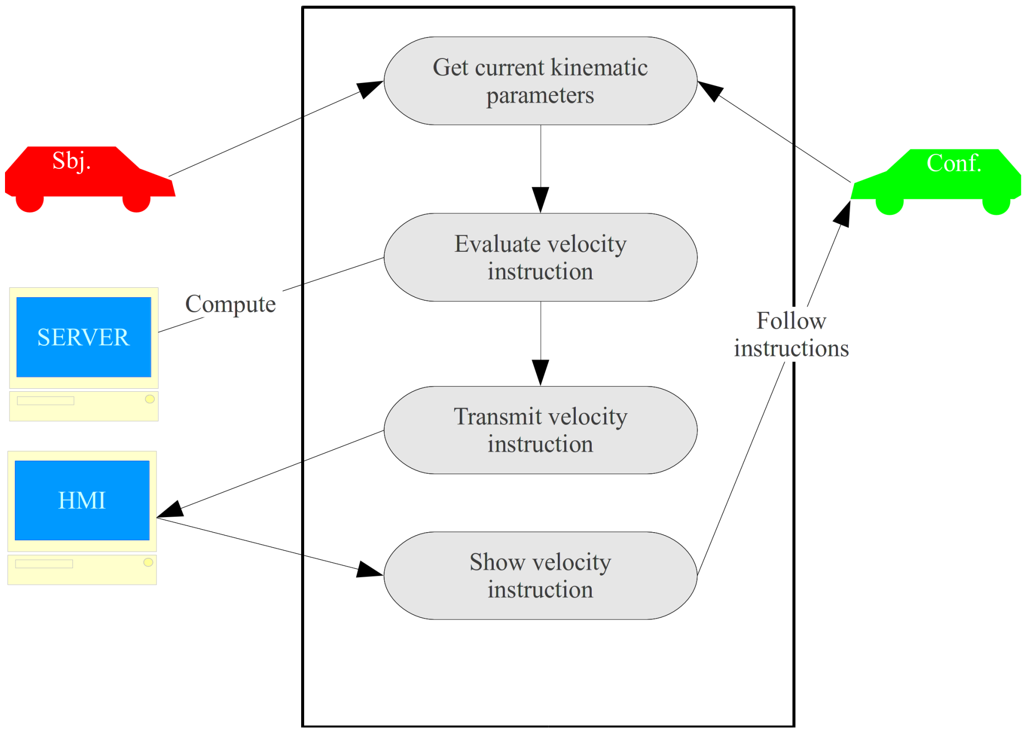

Figure 2.1: *Overall use case*

2.1.3 Requirements specification

The requirements specification table (Table 2.1) has been extracted from the needs of the system described in the previous use case. For example, the devices embedded in the cars should retrieve the car's kinematic parameters (position & velocity), therefore the localization device is restricted to some requirements as accuracy or update rate. Basically, these requirements are used to specify how to design the previously defined system (Section 2.1.2).

2.1.4 System work flow

The flowchart of the overall system in Figure 2.2 shows how the devices are connected all together. The design of the system has been based on this chart. Each singular flowchart represents an application running on a device. The devices of the car are meant to send the kinematic parameters to the infrastructure. The computational server is listening to the same database for new parameters from both cars. As soon as this data is retrieved, the same server application assesses the velocity instruction and sends it to the infrastructure. Since the HMI is listening to the infrastructure, as soon as the instruction is updated, the HMI fetches it and transmits to the confederate driver.

Table 2.1: Requirements specification table

Requirement	Description	Wish/Demand	Priority
	Localization		
Accuracy	The accuracy should be lower than 2.5m CEP (Circular Error Probable)	D	5
Communication interfaces	The device should have at least one communication interface compatible with the main controller device	D	5
Update rate	The device should have an update rate of at least 1 Hz	D	5
Cost	The device shouldn't be too expensive	D	3
Implementation	The device should be easy-to-use in order to be use in the system without difficulties	W	*
	Communication device		
Latency	The latency should be lower than 500ms	D	5
Cost	The device shouldn't be too expensive	D	4
Implementation	The device technology should be easy to implement	W	*
	Main controller device		
Performance	The device should handle several devices (localization & communication)	D	5
Simplicity	The device should be easy to program and configure	D	4
	Communication technology		
System model	The technology should be compatible with the communication devices selected	D	5
Latency	The latency with the communication devices selected should be less than 200ms	D	5
Simplicity	The technology should be easy to implement	W	*
Range	The range should be enough wide to get good connection quality with the communication devices	D	4
	Database platform		
Compatibility	The database should be accessible by all other devices	D	5
Simplicity	The data stored in the database should be easily handled	W	*
	Computational server		
Platform	The OS should run MatLab software in its latest version (2013a)	D	5
Libraries	The needed libraries for the script should be available for the platform (multi-threading, calls over URL, ...)	D	5
Language	Use MatLab language	D	4
	HMI		
Communication	The HMI should have a connection to the database	D	5
Design	The design should be adequate to driver behavior analyses	D	5

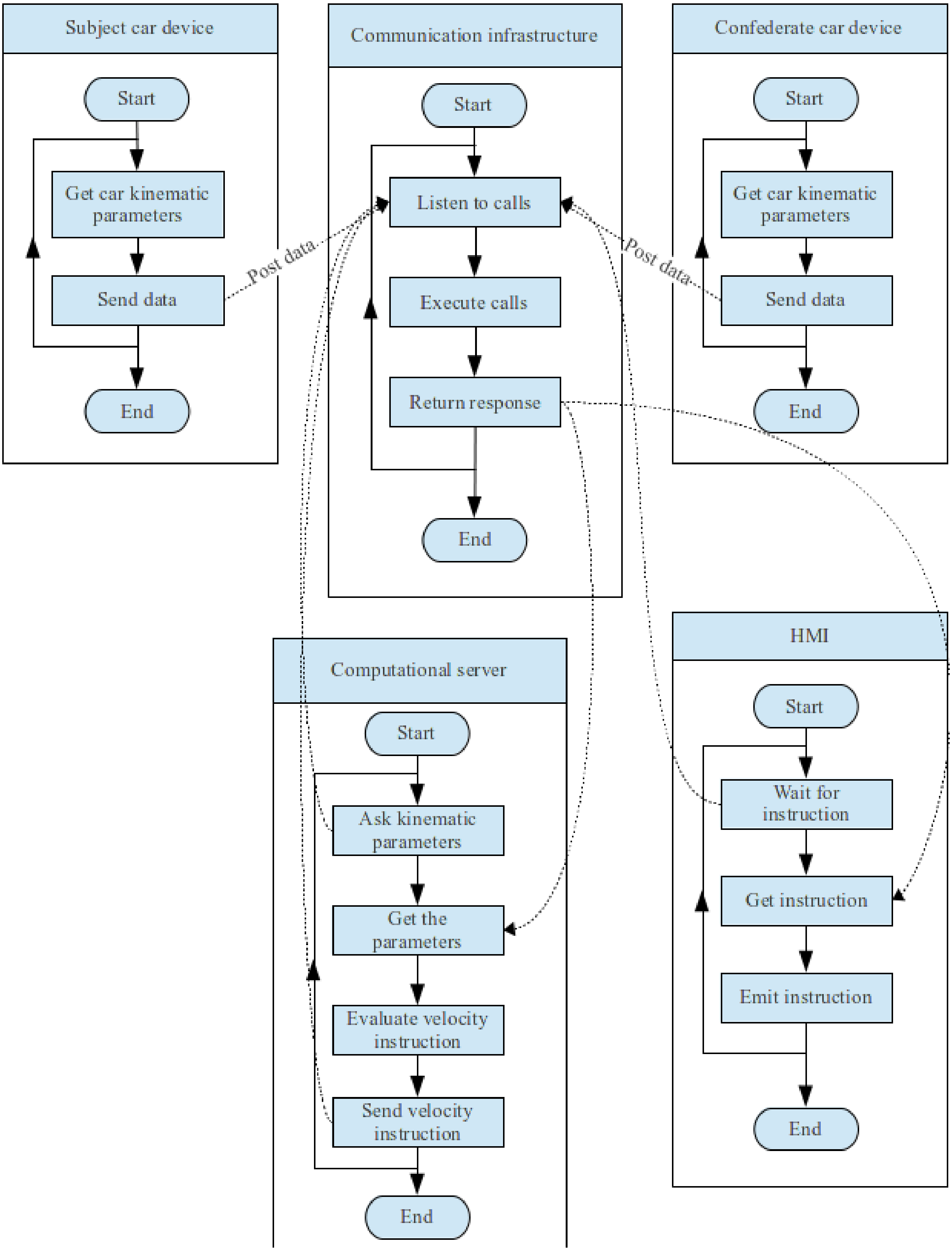

Figure 2.2: *Overall work flow*

2.2 Hardware selection

Considering that the system to be developed is a cooperative system that relies on real-time GNSS data information from two vehicles, it becomes of great importance to use the most accurate GNSS device as possible, as well as the fastest and most effective communication method to transfer the GNSS data; as implied in the Requirements Specification table (Table 2.1). The following sections thus examine the method followed to determine and select the most suitable GNSS and communication devices, considering the requirements, the scope and the resources of the project; see the Requirements Specification table (Table 2.1). A final section will also be dedicated to the selection of the device to be used as the main controller device to manage the operation of both the GNSS and the communication appliances.

2.2.1 GNSS selection

In order to choose an appropriate GNSS device, a market survey of the different existing GPS / DGPS systems was carried out as a first step. This market survey would give a clear and concise overview of the advantages and disadvantages, as well as the costs inherent to each device, thus facilitating the comparison and ultimate selection of the different possibilities of GNSS solutions.
Different GNSS technologies were studied. Among the considered solutions were common GPS devices (i.e. without any sort of augmentation system), GPS devices with ground based augmentation systems (e.g. DGPS and RTK), as well as GPS devices with satellite based augmentation systems (e.g. EGNOS). A comparative table with the main features and approximate prices of the different devices was made in order to ease the assessment process. As seen in the Requirements Specification table (Table 1), the device characteristics of main interest for the project were position accuracy, communication interfaces, update rate, and price; however other characteristics were noted such as velocity and heading accuracy, as well as storage capabilities.
Two specific solutions that were cost effective and readily available, and as such likely solutions for our purpose, were tested for performance. These solutions were: 1) the GPS included in current smartphones and 2) the Phidgets GPS. Phidgets is a commercial brand of appliances for USB sensing and control. Its GPS and control board, named Single Board Computer (SBC), were available due to the fact that they were used in a previous SAFER project. The GPS positioning precision and velocity accuracy of two Phidgets GPS devices as well as three different sorts of smartphones were evaluated. A summary of the tested devices and their maximum update rate is presented in Table 2.2.

Table 2.2: Summary of the GPS devices tested and their update rate

Type of device	Phidgets GPS	Samsung Galaxy GIO smartphone	Samsung Galaxy SIII smartphone	Sony ST21i smartphone
Maximum update rate	10 Hz	1 Hz	1 Hz	1 Hz

The test consisted of two main parts: a position precision test, where the precision of different GPS measurements on a defined location for each device would be evaluated and compared among each other; and a velocity accuracy test, where the GPS track of a defined displacement at a certain speed for each device would be evaluated and compared among each other and to the real path.
In the position precision test, all five devices were placed simultaneously (one next to each other, so that they would be as close together as possible) at a defined location that featured an open sky view and no proximity to buildings or trees (in a radius of at least 100 meters). GPS data was then logged in all five devices during two minutes, at the maximum logging rate for each device, see Table 2.2. This logging procedure was then repeated at four additional points, which were located 15 meters away of the initial logging point, one towards each cardinal point, so that the resulting five locations would resemble a cross pattern when seen from a top view, see Figure 2.3.

In the velocity accuracy tests, the three smartphones were placed on a bicycle that had the Phidgets GPS previously installed on it, making sure to have all four devices as close together as possible. Only four of the five devices were evaluated in this test, due to the fact that the Phidgets GPS devices were mounted on two different bicycles, and as such it was considered irrelevant to perform the test twice. The bicycle was then ridden at a speed (as constant as possible) of 12 km/h, aided by a speedometer installed on the bike, along a 30 m long straight line from south to north (i.e. from point 3 to point 2 in Figure 2.3). GPS data was logged from

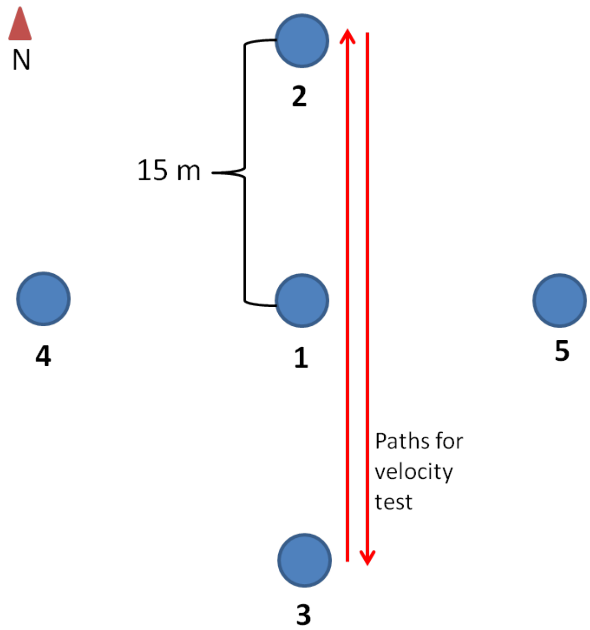

Figure 2.3: *Top view sketch of the GPS test layout*

each of the four devices at the same rate as in the position accuracy test. The procedure was then repeated one time from north to south (i.e. the “return trip” from point 2 to point 3 in the figure), see Figure 2.3. The GPS measurements in these tests are managed in the RT90 coordinate system, which is a system based on metric measures (thus its units are meters) used in Sweden. Being its units in meters makes it convenient and that is why it is used for these results.

2.2.2 Communication method selection

In the same way as for the GPS selection, the selection of the communication method was started by a market survey of the different methods currently used in the automotive industry for vehicle-to-vehicle communication. This market survey was basically carried out with a literature review on the subject (see section 1.1.4), focusing on the main attributes of each technology and how they would influence the behavior of the intended system, based on the statements in the Requirements Specification table (Table 2.1). Among the technologies considered were cellular, Wi-Fi, and 5.9 GHz Dedicated short-range communications (DSRC). Restrictions in range and availability for this project were also included in the evaluation, as well as the infrastructure required.
The online server that will be used as communication platform is a Data Hub Service available from a remote server and developed previously to be used in different SAFER projects. The service consists basically of a database were different type of information can be uploaded, stored and fetched; and a cloud application that has a set of different predefined functions that allow an easy data transfer to and from the server.
In order to assess which of these cloud application functions would be most suited for the intended application in terms of speed and general performance, an evaluation of the three more appropriate upload/download function combinations was performed (see section 2.3). These function combinations were (see Figure 2.4):

- Set and Get functions. The Set function allows uploading name and value of a desired variable to the server. The Get function retrieves the stored value of that variable in the server by specifying the name of the variable and the Client ID of the user that uploaded it.

- Upload and Data functions. The Upload function is used for uploading a set of different GPS data at a given instant, time-stamped by the user. The data that is uploaded for this study includes latitude, longitude, and speed. The Data function downloads the data uploaded by any Client ID with the Upload function, within a specified time span between which data is retrieved for the specific Client ID.

- Setsuperbeacon and Subscribe2 functions. The Setsuperbeacon function also allows uploading a set of different GPS data at a given instant, time-stamped by the user. The data that is uploaded for this study includes latitude, longitude, and speed.The Subscribe2 function works as a “listener” function. By specifying a latitude-longitude pair, a Client ID, and a radius in meters; the function will “listen” (during a limited period of time) and fetch the first set of data uploaded with the Setsuperbeacon function, provided that this data has a latitude-longitude pair that falls within the “listening” radius and that it is posted by the specified Client ID.

Figure 2.4 illustrates the structure of the Cloud Application with the functions evaluated in the analysis.

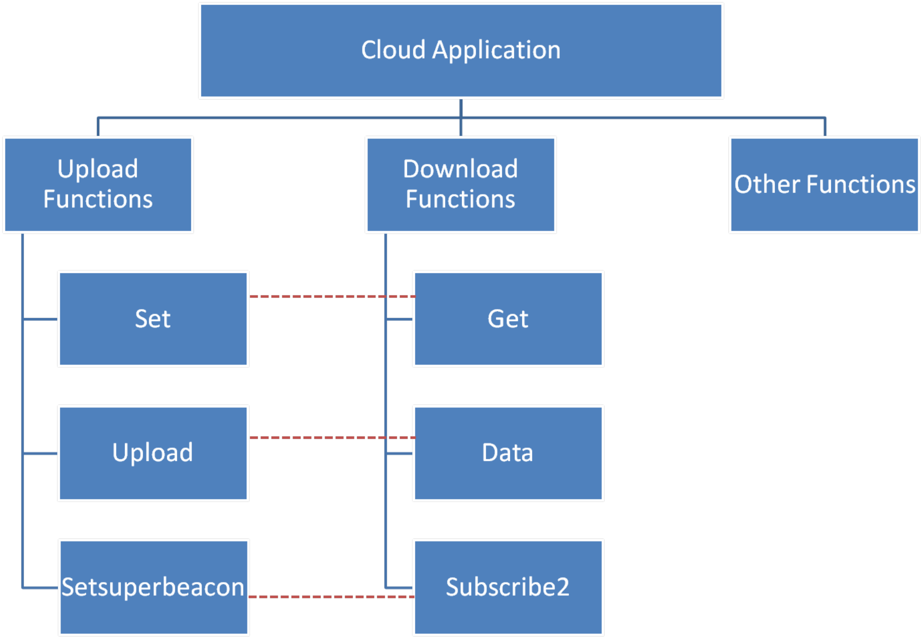

Figure 2.4: *Structure of the Cloud Application including the functions evaluated in the analysis*

2.2.3 Main controller device selection

As a final step in the hardware selection process, an analysis of the possible main controller device to be used as link and manager of the GPS and the communication devices was done. As seen on the Requirements Specification table (Table 2.1), the selection of this appliance was highly dependent on the device and methods chosen for communication and GPS acquisition, since for example some gadgets may be more suitable for one master device than others, or maybe others not even be viable for certain devices. Thus the selection of the master device was left as a final step. The options considered were: Phidgets single board computer (SBC) 3, Raspberry Pi computer, regular laptop computer and smartphone. Two of these devices, namely the Phidgets SBC and the smartphone, were inherently tested in the tests performed for GPS selection. However, in that case the Phidgets SBC used was the earlier version: SBC 2.

2.3 Server timing analysis

Considering that the selected communication method is with an online server via 3G internet connection, it is important to analyze the time delays that this type of communication can create on the performance of the system. Recall at this point that there are communication paths between each device and the server, as well as between the Matlab computer and the server. This analysis is of great importance since this is a cooperative system, which implies constant real-time communication needs among the different devices.
The server timing was evaluated by a set of previously acquired GPS data to the server and then fetching back that data from it, while measuring the time elapsed in these operations. Specifically, three tests were performed for each function combination: one where the data was only uploaded to the server and the time elapsed for this upload measured; another one where the data was only downloaded from the server and the time elapsed for this download measured; and a final one where the data was uploaded and then immediately downloaded, with the total upload plus download time measured. Each of the tests was performed 950 times, which corresponds

to the length of the previously collected GPS data set. The tests were executed with Matlab. That is, the GPS data was sent and retrieved using the respective function combination within the same script. All evaluation was carried out separately with a Wi-Fi internet connection and a 3G internet connection. Figure 2.5 illustrates the evaluation flow.

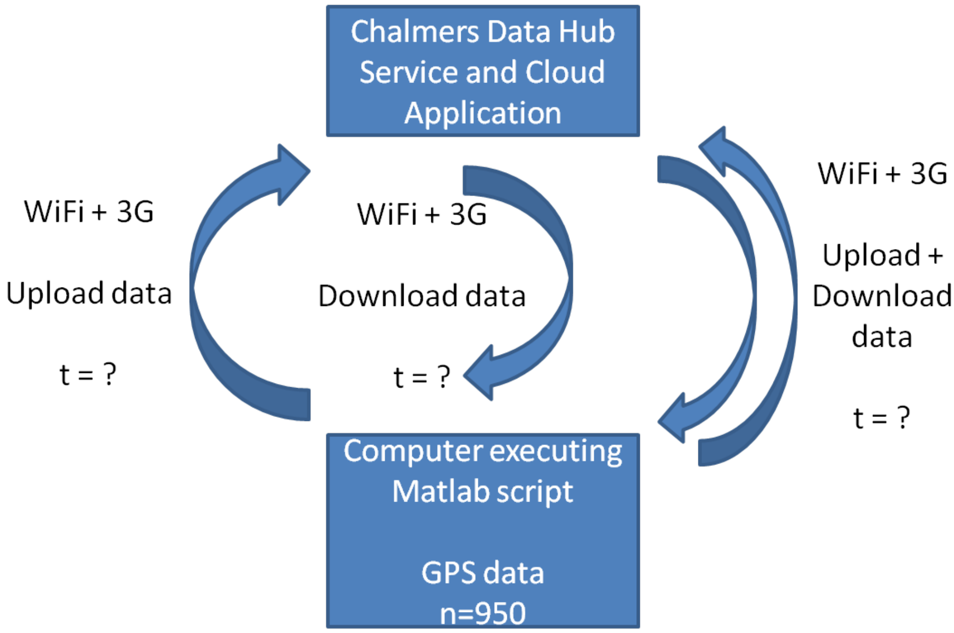

Figure 2.5: *Evaluation flow of the server timing analysis*

The characteristics of the Set function (see description in section 2.2.2) only permit uploading of one single variable with each use of the function; i.e. it is possible to upload, for instance, only GPS latitude. Thus the remainder of the GPS data; e.g. longitude, speed, etc; would need to be uploaded with another call of the function for each variable. As a result, the three previously mentioned tests were performed two times for the set function: one time by only sending one variable and measuring the stated execution times, and another time by sending all the GPS variables with consecutive calls of the Set function and measuring the total execution for transferring all the variables.
Once the Phidgets boards were properly configured for acquiring the GPS data and sending it to the server, server timing analysis tests were also performed in order to assess the possible connection related delays on the real system.

2.4 Historic profiles analysis and selection

Considering the previously described driving situation to be addressed, the foundation of the performance of the system lies in predicting the TTI for each of the vehicles (confederate and subject). However, as seen in Section 1.1.2, the prediction of this TTI becomes difficult to achieve by means of using only mathematics. One way to address this issue is to use historical information from previous passes through the intersection, in a similar way as for the time-to-arrival algorithms for buses (see Section 1.1.2).
In the hypothetic case of a perfect prediction, the system should be able to make the confederate car reach the intersection at *exactly* the desired moment. Hence, the further the prediction is from the timing outcome, the more the arrival time of the confederate vehicle to the intersection gets shifted in time. It is then reasonable to conclude that gathering higher quantities of historical information, i.e. by making many different drivers go through the intersection several times, would make the prediction of time to the intersection likely to be improved. Also, if the quality of each of this historical data is high, meaning that no uncommon driving behavior or situation is present during the collection of this data, then the quality of the prediction should also be enhanced. An example of an undesired situation that would negatively influence the quality of the data for the purpose of this study is when a vehicle is also turning in front of the subject vehicle, thus making the subject vehicle need to brake. In this case, the time to intersection will be shifted and this pass would need to

be discarded.
The next sections will explain how historical data is defined and collected, and how the algorithm created for the time to intersection prediction is devised and how it was developed.

2.4.1 Profile definition

There are different possible ways to characterize the kinematic behavior of a vehicle when approaching an intersection, in order to gather the historical data. One of the most common is to analyze how a variable, usually speed or distance, evolves with respect to another variable, usually time or distance. One intuitive way to visualize this evolution is by comparing these variables against each other. To make this comparison more understandable, it is common to express the time and distance variables as time-to-intersection (TTI) and distance-to-intersection (DTI), respectively. Here, the point considered is the intersection center, which is defined as the point in space where the paths of both vehicles intersect. Since one comparison data set will be produced for each time a vehicle (either the subject or the confederate vehicle) drives from a certain point and through the intersection of interest, then many different sets of data, visualized as plotting curves, can be gathered for a specific car and driving situation. This set of resulting set of data is what will be called historic profiles, and is the data that will be used as the base of the prediction of the time to intersection.
Figure 2.6 and Figure 2.7 show two examples of historic profiles: one where speed is plotted against the distance from the intersection center (i.e. distance to intersection), Figure 2.6; and another one where the distance to intersection is plotted against the time to intersection, Figure 2.7. Note that these two profiles do not have any correlation with each other, and are merely illustrations of how different profile curves look like.

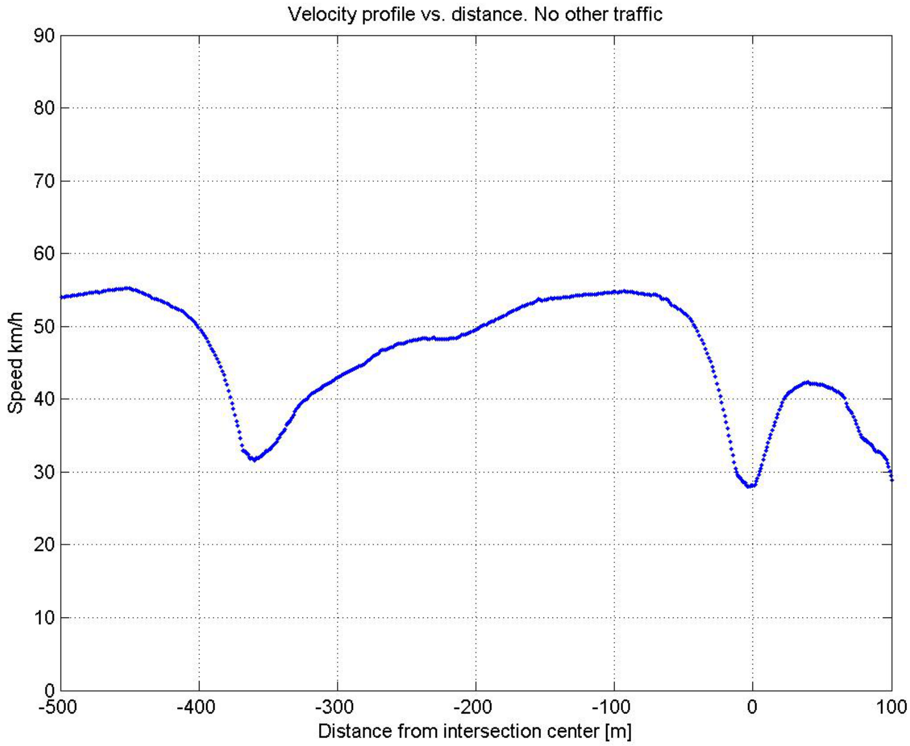

Figure 2.6: *Example of a speed versus distance to intersection profile*

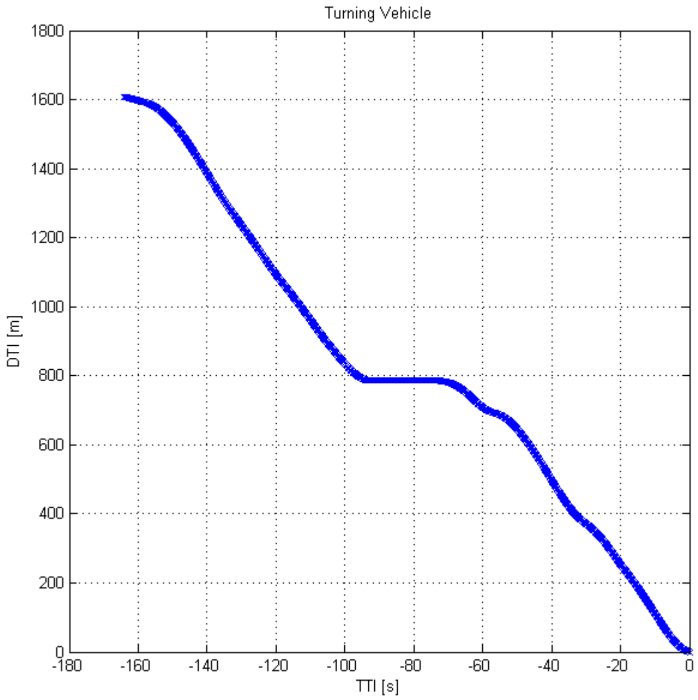

Figure 2.7: *Example of a distance to intersection (DTI) versus time to intersection (TTI) profile*

2.4.2 Extraction and filtering of historic profiles

By the time this project started there was already some data collected from a specific intersection in Gothenburg, for trajectories of both the subject and the confederate vehicle. However, during the course of the project, the intersections of interest were changed for two new intersections also in Gothenburg, and so the historic data had to be collected parallel to the development of the project. Nevertheless, the originally collected data was used for the development of the profile selection algorithm since the theoretical concept would be the same regardless of which intersection is to be studied.

The intersections ultimately decided to be studied in this project are the Skattegårdsvägen – Önneredsvägen intersection and the Grevegårdsvägen – Opalgatan intersection, both in the Västra Frölunda region in Gothenburg, Sweden. Figure 2.8 shows these intersections highlighted in a red circle in a Google Maps image.
For this particular project, the information from historic passes at the intersections of interest comes from data collected by cars from Volvo Cars specially equipped with a Field Operational Test (FOT) data logger. This data includes, among other, information logged from the CAN bus regarding vehicle status: e.g. speed, lateral acceleration, braking information. In addition a GPS is attached for which the data is also collected by the same data acquisition system. The method to extract the historic profiles from the raw FOT data is detailed next. First it was decided that the type of profiles that would be used for the algorithm would be distance to intersection (DTI) vs. time to intersection (TTI). Then the coordinates (latitude and longitude pair) of the intersection center were defined based on the historic GPS data and with the help of Google Maps. Once the intersection center was defined (see Section 2.4.1), a program was created to check when the vehicle had been driven through the intersections. Special attention was given to the path of the vehicle before and after passing through the intersection center in order to determine whether the profile should be classified as for the subject vehicle or for the confederate vehicle, and also to avoid making profiles out of passes through the intersection that do not fit into any of these two categories (i.e. subject or confederate). The DTI and TTI vectors were created from the raw data by changing the time origin to the time when the car is passing the intersection.

As suggested earlier, it could happen due to natural traffic conditions that some of the gathered profiles do not adequately represent the natural approach of the driver to the intersection in free flow (i.e. without other

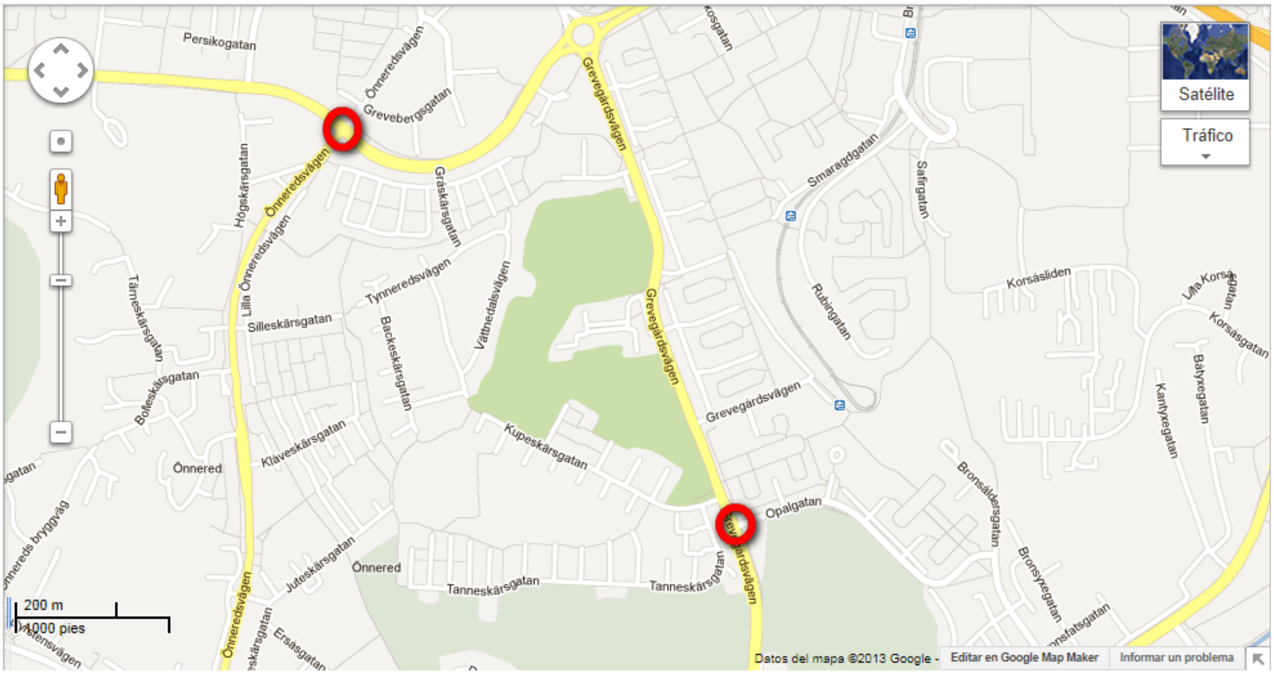

Figure 2.8: *Google Maps view of the intersections to be studied in this project*

traffic). This could happen for instance if the driver was forced to stop and pull over for a given reason, e.g. if traffic in front would slow down to exit the road at a given point before the intersection, or if the subject vehicle was forced to stop before turning due to oncoming traffic, among other possible situations. This type of profiles can have a negative impact on the performance of the profile selection algorithm if they are used for it. It is therefore important to make sure that the set of historic profiles used in the selection algorithm does not include this type of behavior. In this project, this was achieved by performing a visual inspection of the profile curves and either manually removing the undesired profiles or creating simple programs to remove profiles that share a common unwanted characteristic. Finally, an additional program was created in order to check and remove duplicated profiles that were present in the data set.

2.4.3 Selection algorithm

The general concept behind the time to intersection prediction algorithm is as follows. When the main algorithm has fetched the GPS data, namely latitude, longitude and speed, it will use this information to decide and choose the historical profile that best fits to the current vehicle situation, or even create a new virtual profile. This selected or created profile will be used to extract a TTI for each DTI, or in other words, to make the prediction of the time it will take the vehicle to arrive to the intersection.

Mean and median approach

Different profile selection criteria, with increasing complexity, were proposed and implemented for the selection algorithm. The simplest one evaluated consisted of the creation of one unique DTI vs. TTI profile that would be representative of all the profiles. Program scripts were then created to generate the mean and the median profiles, which would be the ones to be tested (separately).

Speed based approach

A more elaborate alternative implemented and evaluated consisted of using the current coordinates of the vehicle to identify the distance to the intersection, using the historic profiles. Then a scan is made through all the historic profiles looking for the DTI point closest to the current DTI of the vehicle, i.e. searching the point in the profile where the “historic car” was at the same location as the current vehicle. The speed that each of the vehicles had in the historic profiles at that point is then compared to the speed of the current vehicle. The profile of the vehicle with the smallest difference in speed compared to the current vehicle speed at that particular point is thus chosen, and so the TTI corresponding to the DTI of the chosen profile is selected as the time it will take for the vehicle to reach the intersection. Figure 2.9 illustrates this procedure. The assumption with this method is that if the current vehicle is going at a given speed at that specific point of the road, and this speed is close or even equal to the speed another vehicle had when driving through that same point on the road, then it is likely that the current driver will keep driving in a similar manner as the “historic driver”; and thus it would take roughly the same time for both drivers to reach the intersection. It is important to note that the selected profile can change in the next evaluation instant. The resulting DTI vs. TTI profile is thus a virtual profile made out of points of the different historic profiles chosen at different instants. This method will be referred as the speed based approach.

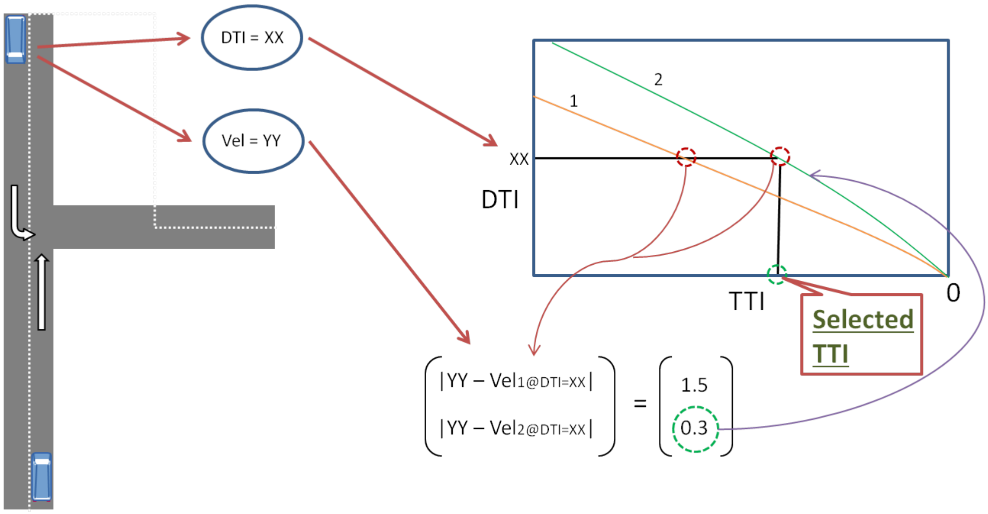

Figure 2.9: *Illustration of the TTI prediction process on the speed based approach*

Moving window approach

Two subsequent improvements to the previous method were sequentially implemented, in attempts to increase the effectiveness of the prediction. In the first one, which will be called the moving window approach, the objective is to take into account not only the current speed, but also the speed the vehicle has had for a certain amount of time before the current point (i.e. the current and certain previous data points). This applies for both the current vehicle information as well as for the information of each of the historic profiles. The difference in velocity is then calculated for each of the corresponding data points and then averaged. The result is then used to decide which of the profiles has had the most similar speed over that period of time, thus choosing it. Figure 2.10 illustrates this procedure. The expected improvement with this approach comes from the fact that it chooses the profile whose speed has been more similar to the current vehicle speed over a period of time, rather than just at the considered instant. The number of previous data points to be handled can be adjusted easily, but in principle was adjusted to 30, which corresponds to three seconds since the data logging rate is 10 Hz for both the GPS (current vehicle) and the FOT data logger (historic data).

Weighting factors approach

In the final improvement, called the weighting factors approach, a weighted selection of the most suitable profiles is used, rather than just choosing only one profile. In this case the number of suitable profiles can also be adjusted easily, but as a first instance was decided to be ten. Thus, the predicted time to intersection is the weighted mean of the ten most suitable profiles. The suitability of the profiles comes, as before, from the calculating the difference in speed. The weighting factors are 10 for the first profile, 9 for the second, and so on until being 1 for the tenth profile. The expected improvement with this approach is to have the resulting prediction be more balanced between the most suitable options.

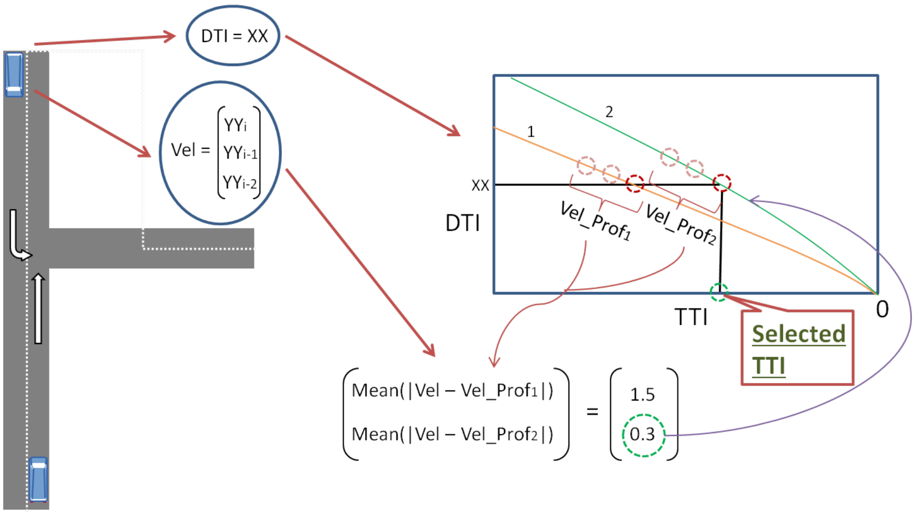

Figure 2.10: *Illustration of the TTI prediction process on the moving window approach*

Assessment

For the implementation and evaluation of the above mentioned selection criteria, simulations were carried out by using the available historical profiles as input source. In these simulations, one of the historical profiles is randomly selected and its relevant information, i.e. latitude, longitude and vehicle speed, is used as simulated information of the "current vehicle". This profile is then removed from the historic profiles set, and the profile selection algorithm is run. To visualize the results, all the historic profiles, as well as the input profile and the predicted profile are plotted in a same graph.
In order to compare and assess the performance of the different selection criteria, a performance indicator for the algorithm was established. This performance indicator was the difference in TTI between the predicted profile and the actual outcome TTI for the simulated "real" profile. This performance indicator becomes more relevant where the vehicle is getting closer to the intersection. It was thus decided to use the TTI difference at a DTI of 100, 75, 50 and 25 meters as the performance indicator. A table with the mean and the standard deviation of the TTI difference for each of these DTI points was thus created to make this comparison and assessment. A total of 15 simulation runs were carried out for this assessment, which corresponds to the total number of historic profiles available for the subject vehicle in one of the intersections. Thus, each profile was chosen one time (i.e. on one simulation) as the input profile.

2.5 Velocity instruction algorithm

The velocity instruction algorithm is the core of the system assessment. It uses the previously defined profile selection to assess the best velocity instruction that the confederate car driver should follow in order to reach the intersection at the wanted time.

2.5.1 Definition of parameters

The algorithm uses the predicted time-to-intersection extracted by the historic profiles selection, the complete work flow is detailed below. Before this definition, one should get an overview of all the variables and parameters, Table 2.3 shows them and their respective definition. There are two different types of data: those which represent the state of the subject (Sbj) car and the confederate (Conf) car, and those which are only script parameters.

Table 2.3: Variables definition

Category	Name	Description	Algorithm parameters	Car state variables
Acceleration	aMax	The maximum acceleration value that the algorithm should not exceed to assess the velocity instruction	X	
Velocity	Speed	Current speed of the corresponding car: $\{SbjCar.Speed, ConfCar.Speed\}$		X
	V_{High}	The upper limit of the velocity instruction, the algorithm cannot give an instruction above it	X	
	V_{Targ}	The target velocity which should be given as the reference velocity while starting-up or stabilizing	X	
	V_{Low}	The lower limit of the velocity instruction, the algorithm cannot give an instruction below it	X	
Position	Latitude, Longitude	The GPS position for each car: $\{\{SbjCar.Latitude, SbjCar.Longitude\}, \{ConfCar.Latitude, ConfCar.Longitude\}\}$		X
Distance	DTI	The distance to the intersection for each vehicle: $\{SbjCar.DTI, ConfCar.DTI\}$		X
	Checkpoints	There are four different checkpoints: the *start-up*, *regulation*, *stabilization* and *releasing* checkpoints respectively symbolized as the time where the areas' color changes in Figure 2.11. When the subject car passes a checkpoint this changes the velocity instruction algorithm phases.	X	
Timing	TTI	The time to intersection for each vehicle: $\{SbjCar.TTI, ConfCar.TTI\}$		X
	$\Delta T_{prediction}$	The prediction timespan, i.e. the base time used to estimate the next step within the algorithm	X	

2.5.2 Definition of phases

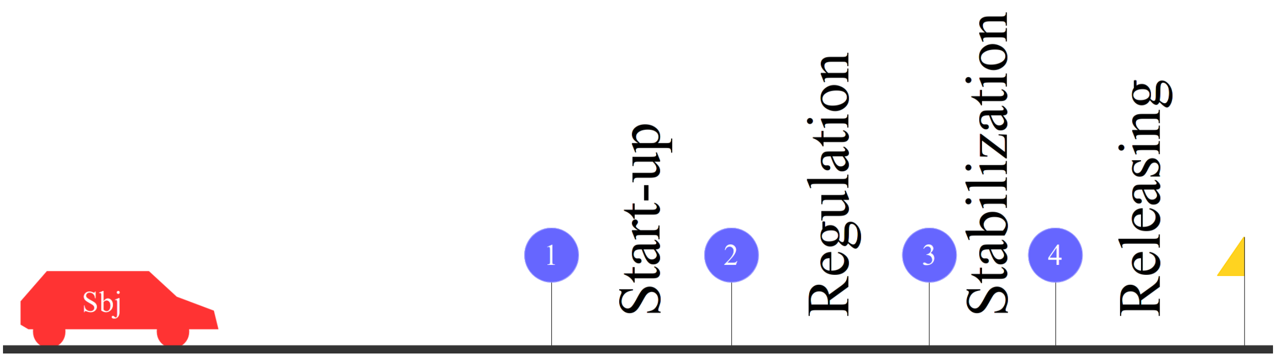

Figure 2.11: *Velocity regulation algorithm phases*

The instruction is divided into different phases in order to take into account the configuration of the scenario. This is done because the Confederate car will be standing still while waiting instruction and that it will have to be freed while entering the intersection. Meaning that the Confederate car should start-up, regulate its speed, stabilize its speed, and then enter the intersection. The checkpoints (Figure 2.11) which demarcate the phases are related to the Subject's DTI. Since the Confederate is standing still at the beginning it is straight forward to put checkpoints related to the Subject than to the Confederate. When the Subject is passing through a checkpoint, the velocity instruction assessment should follow a pre-defined set of rules. The Figure 2.11 shows the phases and their definition is as follow:

- 1^{st} phase: *Start-up* phase, the algorithm instructs the Confederate driver to start-up to the velocity target
- 2^{nd} phase: *Regulation* phase, the algorithm evaluates the best velocity instruction related to the current Subject's DTI and velocity to reach the aimed dTTI
- 3^{rd} phase: *Stabilization* phase, the algorithm instructs the Confederate driver to stabilize his/her speed to the velocity target
- 4^{th} phase: *Release* phase, the algorithm stops providing instruction in order to let the driver to decide how to enter the intersection

Note: The velocity target is a pre-defined value of speed which is chosen according to the velocity limits on the current road. For instance, the roads where the system will be evaluated are limited to 50km/h therefore the velocity target will be tuned to 45km/h in order to have a top margin, in term of velocity, about 5km/h. This is because, in an experiment, one can not ask the Confederate driver to exceed speed limits and thus, to be able to adjust speed upwards, the target speed should be lower than the speed limit.

Another phase, which is not defined here and displayed on the figure, is the "*Be Ready*" phase. The program will warn the Confederate driver 3 seconds before start-up to be ready to start.

The *Regulation* phase is more complex than the others, the following 4 figures illustrate the different steps to determine the optimal Confederate car velocity instruction. This phase uses the reference profiles extracted by the historic profiles selection. In the figures, there are 2 different profiles, one for the subject car and one for the confederate car.

2.5.3 Description of the regulation

This regulation algorithm provides the velocity instruction that the Confederate should follow in order to reach the targeted dTTI. The positions of both cars (Subject and Confederate) are used in order to estimate the current dTTI and to determine the optimal velocity.
The *first step* graph (Figure 2.12) shows the current estimated DTI of both cars. This gives two different TTI. The difference of these values is then the estimated dTTI. In this example the aim of the system is to have both vehicles entering the intersection at the same moment, therefore the dTTI should be equal to zero (dTTI = 0).
The *second step* graph (Figure 2.13) shows how the Subject predicted DTI and the optimal Confederate DTI are found. The first arrow shows how to find the Subject predicted DTI while the following arrow shows how to determine the optimal DTI the Confederate car should be at the next state. As a remark, the optimal DTI for the Confederate is on the same vertical line as the Subject car predicted step because the aimed dTTI is equal to zero.
In the *third step* graph (Figure 2.14), the Confederate optimal DTI is reported to the Confederate predicted state (the green vertical line).
The *fourth step* graph (Figure 2.15) shows the velocity instruction (slope between the current Confederate car DTI to the reported optimal DTI, red single arrow). If the car follows this instruction, the difference in TTI at the first step, should be caught up. The green curve is thus shifted (red curve) by this initial difference in time.

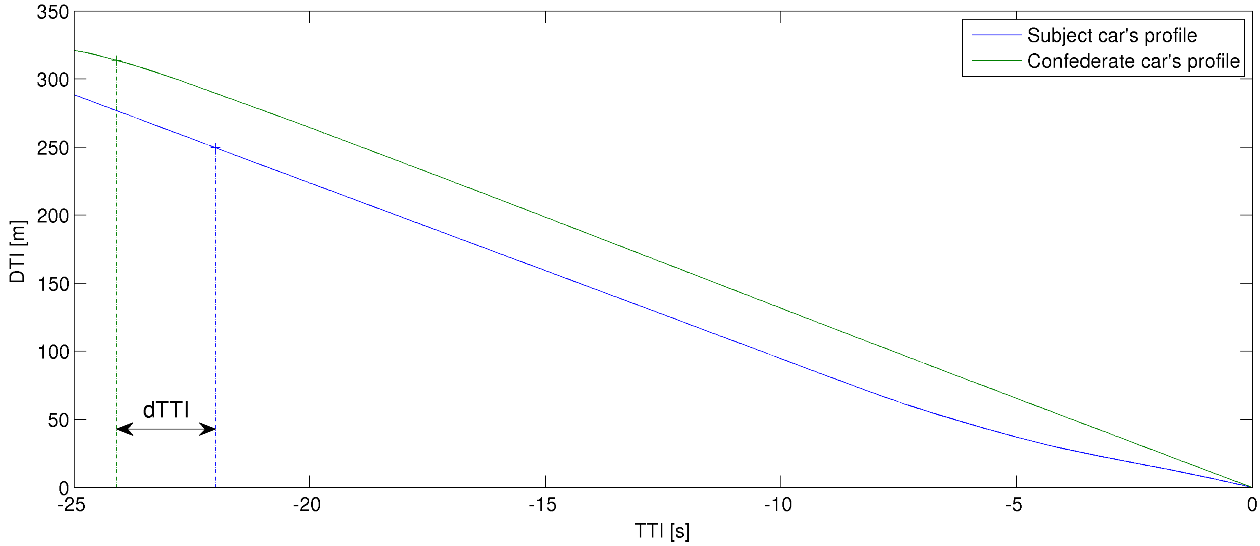

Figure 2.12: *First step: current state*

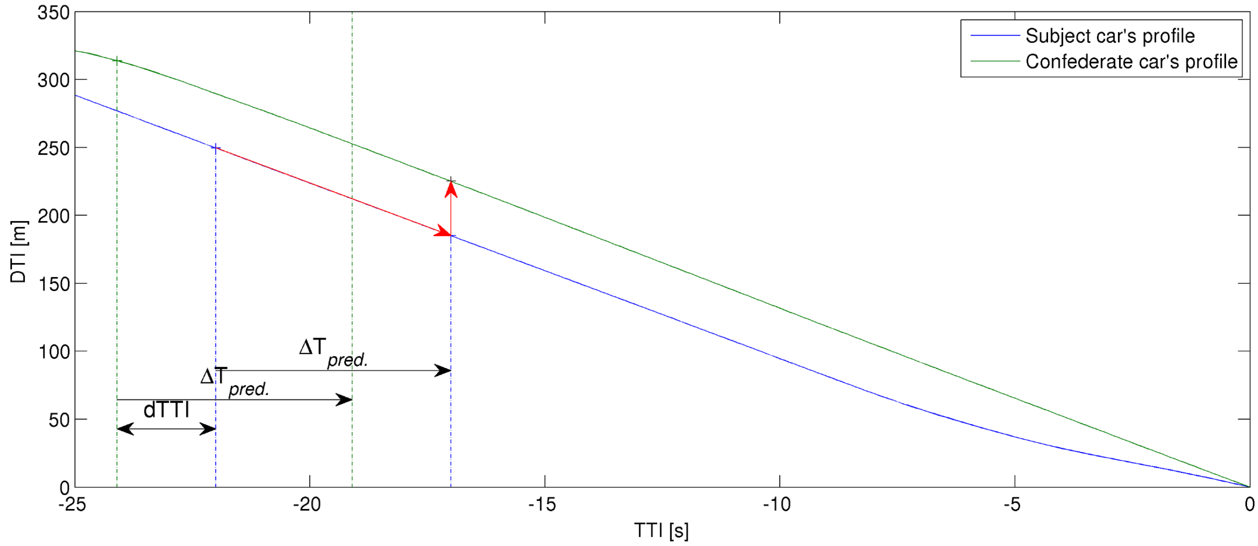

Figure 2.13: *Second step: predicted next step*

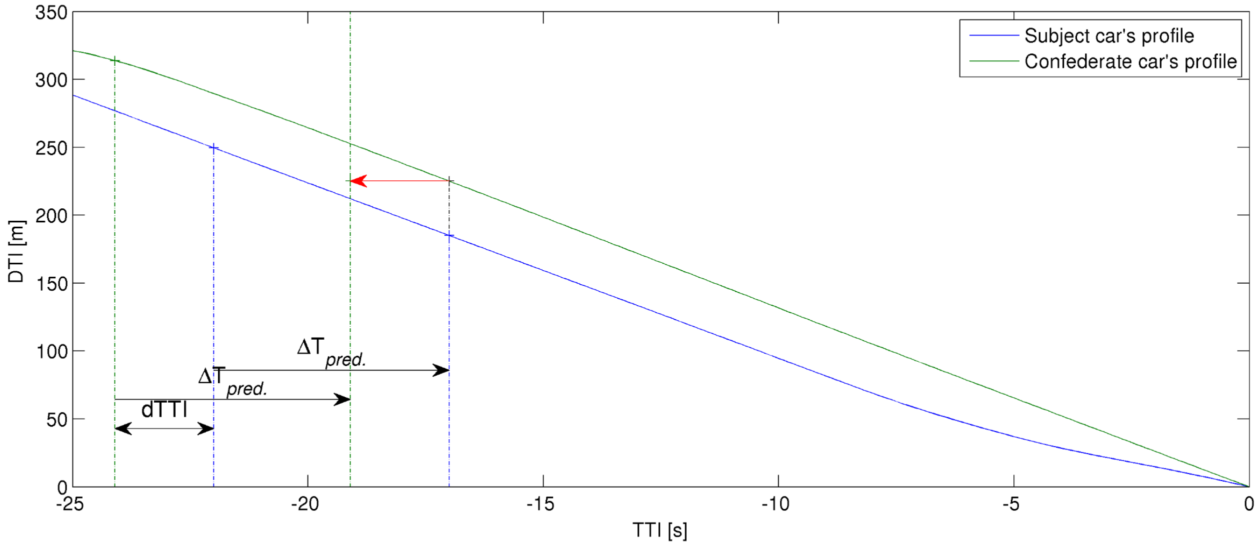

Figure 2.14: *Third step: the ideal next DTI*

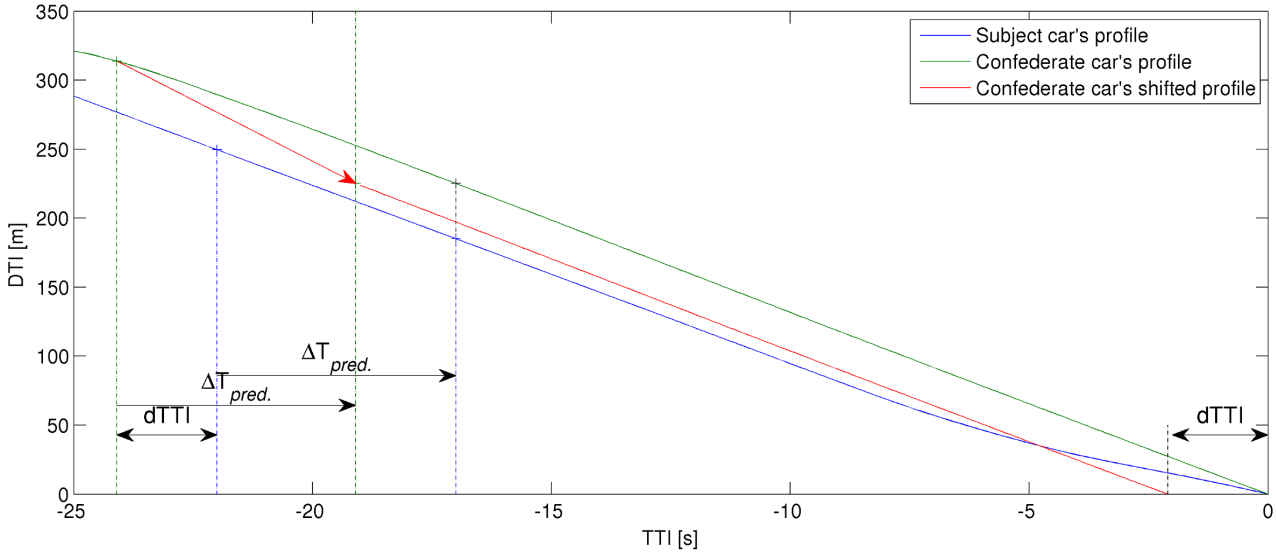

Figure 2.15: *Fourth step: velocity instruction evaluated*

Following what stated in Table 2.3, the velocity instruction is limited by V_{low} and V_{High} and the variation of velocity between the current actual speed and the instruction can not exceed aMax. The last step of the regulation is then to change the velocity instruction accordingly to these requirements.

2.6 Development of algorithms

As it has been shown in the previous sections, the system is divided over 4 different devices. One of them, the communication infrastructure, already existed and was working properly. Software for each of the other devices needed to be developed.

2.6.1 Detailed flowcharts

Based on the general flowchart (2.1.2), the following flowcharts show how the different algorithms should work.

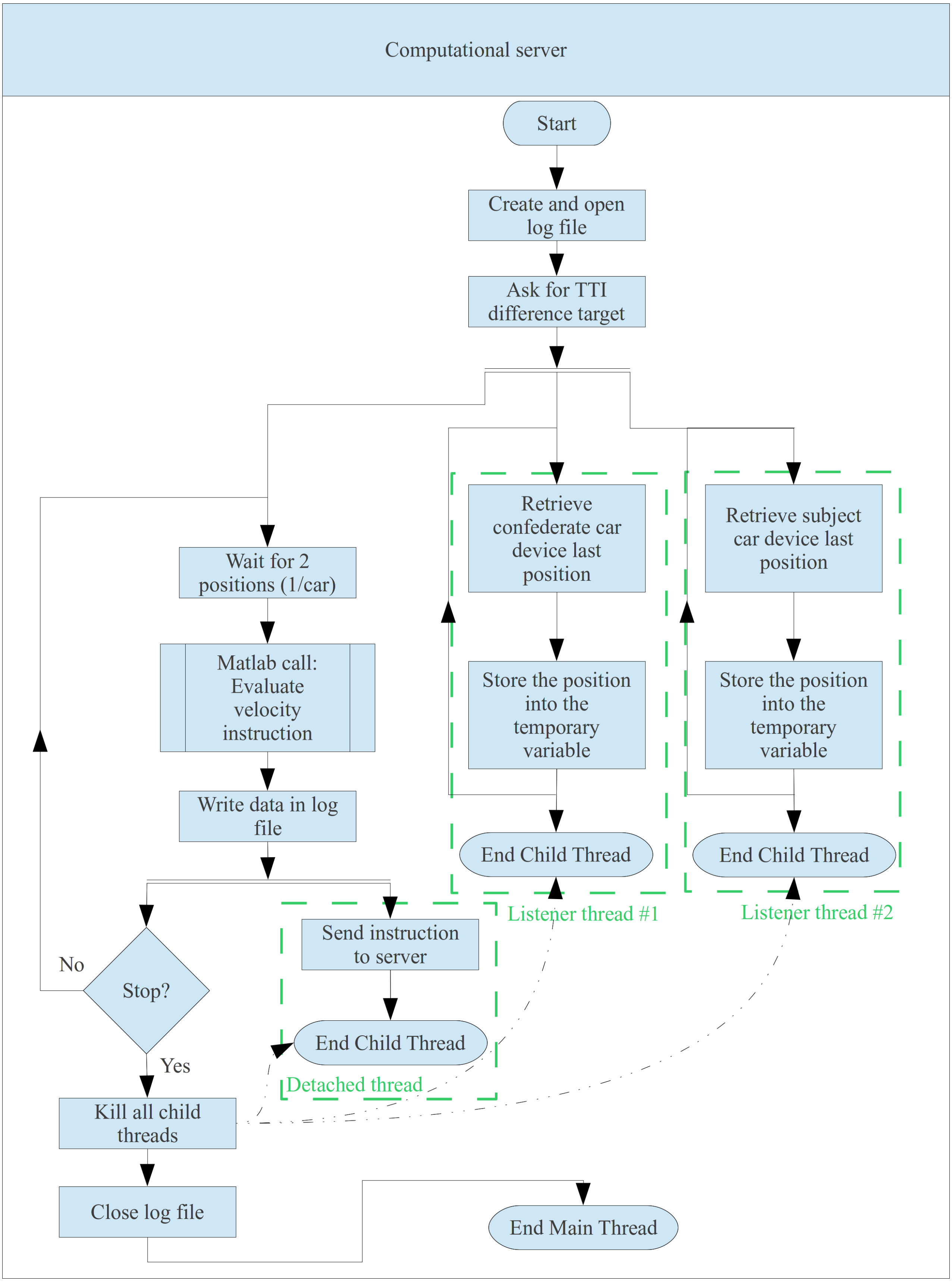

Figure 2.16: *Server-side application flowchart*

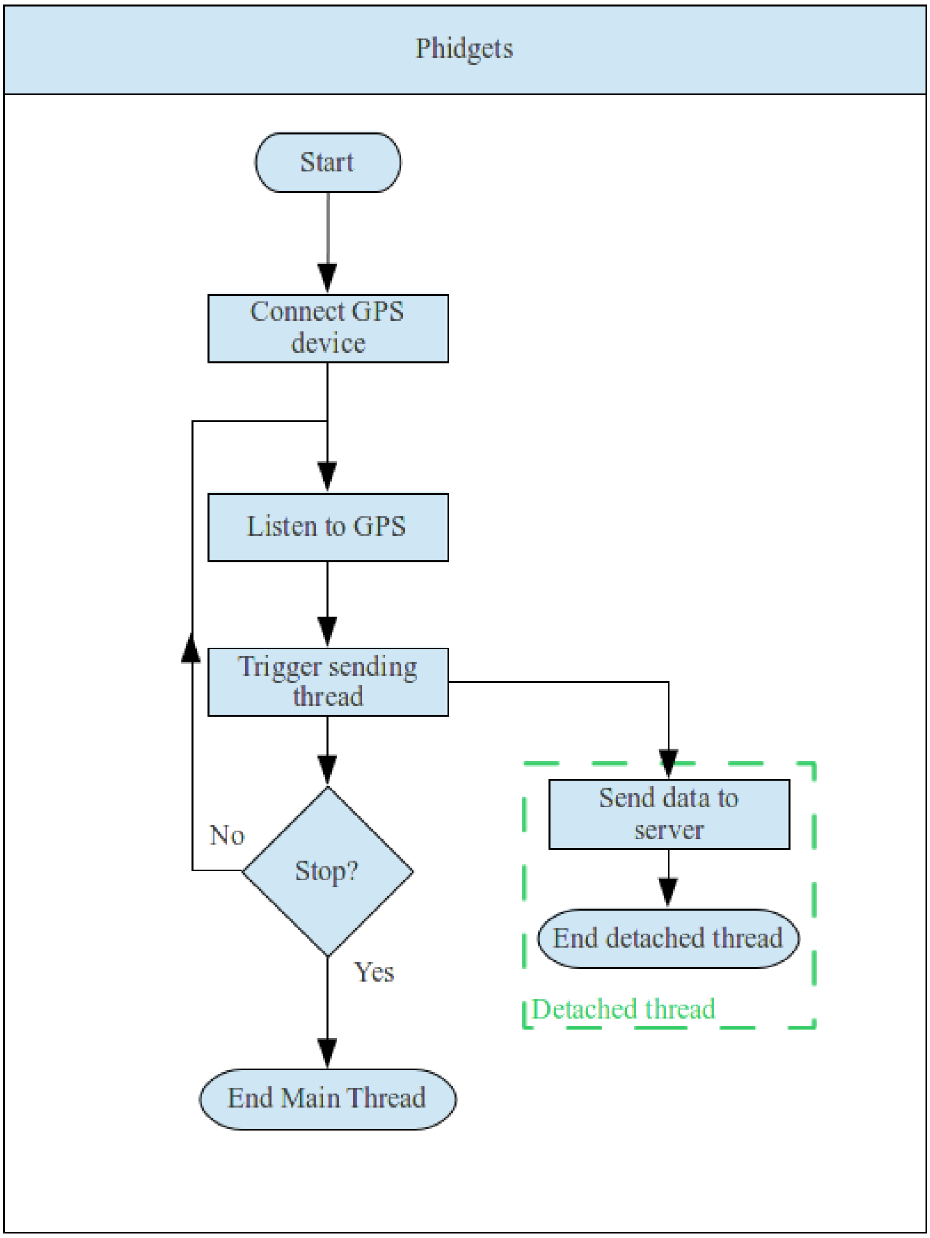

Figure 2.17: *Phidgets algorithm flowchart*

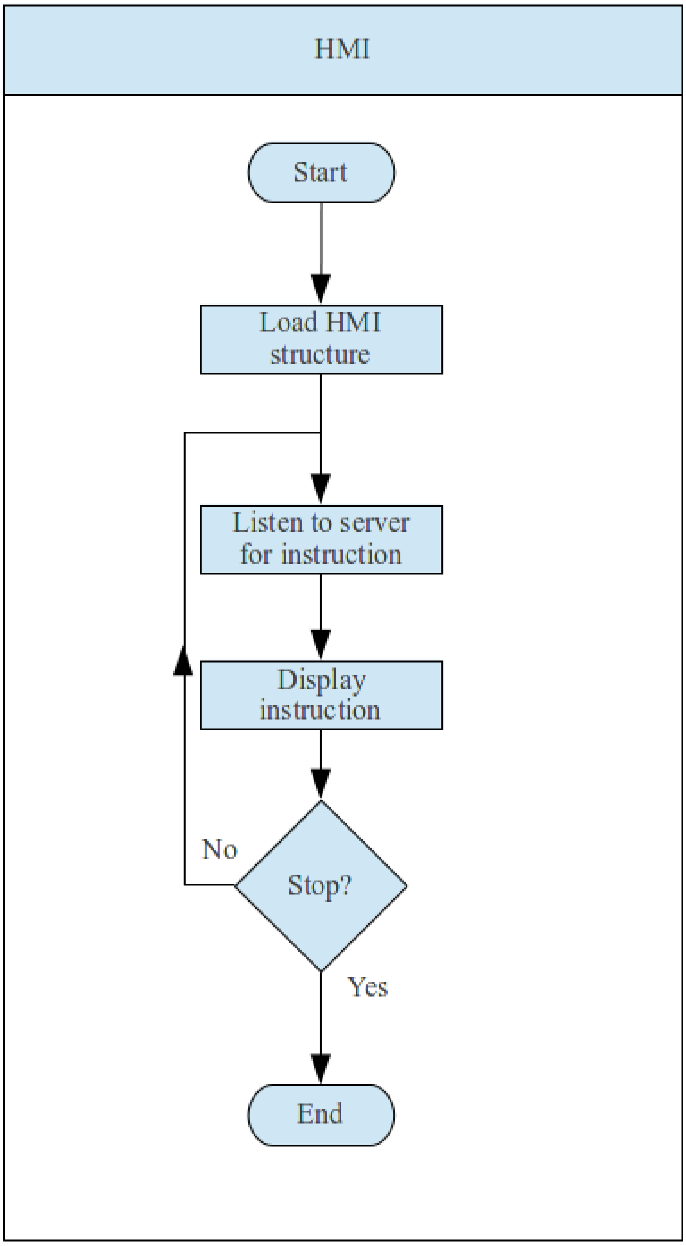

Figure 2.18: *HMI algorithm flowchart*

2.6.2 Algorithms characteristics

Server-side algorithm

MatLab was chosen as the main algorithm platform for several reasons, of which one important is that all FOT data is available in MatLab format. The corresponding flowchart (Figure 2.16) shows different characteristics that were implemented in the final application.

- Multi-threading

  Multiple threads should run at the same time. The parallel computing toolbox (PCT) provided by Mathworks cannot handle as many parallel threads as was needed, while at the same time communicating the required data between them. Therefore, the capability of MatLab to use C functions (MEX-files) was used to implement the needed multi-threading. The MEX C function was compiled and called from MatLab. This function creates the multiple threads needed and each respectively calls the MatLab functions it needs (as created for the project purposes).

- Data logging management

  The database server came with a bug in the function used within the application. Therefore, it was required that the MEX function creates and populates a log file. When the system is shut down this log file is accessible and is used to post-process the collected data.

- TTI difference target submission

  In order to let the user make the proper test on field, it was a requirement that the test instructor should be able to change the difference in TTI at the application start. Thus, a GUI (Figure 2.19) was added to allow the user to set the time difference between when the confederate car arrives in the intersection and the subject does. This is to allow study designs with different timings, e.g. to study Go/No-Go decisions.

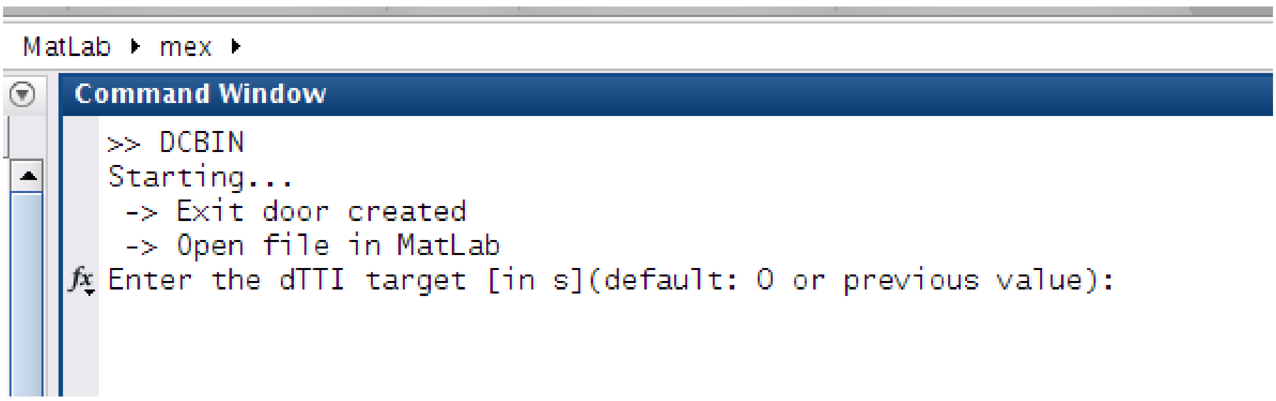

Figure 2.19: *Interaction with the experimenter*

Phidgets algorithm

The ease of coding in C++ on the Phidgets Sbc3 was the reason for using this programming language. Moreover, as the server-side application was also developed in C++, it was more coherent to use the same language. The Phidgets company provides the libraries needed to fetch the GPS data, while the cURL library provides the functions to send data to the server and the pthread library provides the multi threading functionality.

HMI algorithm

The HMI should be useable on different mobile devices with access to internet, in order to easily evaluate and use the system. Its function is to fetch the instruction from the server and provide the confederate driver with the information. Therefore, the HMI is implemented as a web page which can be accessible from anywhere, and on any device. The languages used is HTML 5 and CSS 3 to render the page, Javascript with the jQuery library to make it dynamic, and Php5 to retrieve the velocity instruction from the server.

2.6.3 Simulations

Before any testing, the server-side application has been simulated using a subject vehicle historic profile as simulated current kinematic parameters. For the simulation, a model of the confederate car has been shaped. It follows exactly the velocity instruction in the regulation phase, and for the other phases it goes to the velocity instruction with a constant acceleration (linear velocity model). In the releasing phase, the velocity is equal to the target velocity. Thus, the MatLab application was run as it would run in reality since a JavaScript string generator was created to build the same kind of input that it gets from the cloud application. The algorithm parameters have been set as follow: $\Delta T_{prediction} = 0.1s$, $aMax = 3m/s^2$, $Vtarg = 45km/h$, $checkpoint_{stabilization} = -40m$, $checkpoint_{releasing} = -20m$, $T_{start-up} = 4s$.

First simulation: To evaluate the confederate car model

To evaluate the velocity regulation algorithm behavior, the evolution of the velocity instruction has been plotted aside the confederate car model's velocity. The aim of this simulation was to check if the model was perfectly running before evaluating the algorithm overall performance.

Second simulation: To evaluate the result

In this simulation, the algorithm has been run several times with the previously defined model. The program provided the final difference in TTI between the subject car and the confederate car. Each run time, the initial subject car DTI was modified. The distribution of the resulting difference in TTI can be analyzed.

2.7 HMI development

The HMI development is based on two characteristics: its technology and its behavior.

2.7.1 Technologies

Since the instruction is sent to a server, the technology should support web calls; and since the majority of programming languages have these characteristics, the spectrum of choices was broad. Nevertheless, the basic web development languages: HTML, CSS, JavaScript and PHP have been chosen. This gives a complete compatibility between the different possible supporting devices. At the first step and what is used in the evaluation of the system in this thesis, the PHP server is run on a local machine in the confederate vehicle, the same machine integrates the HMI. The second step would be to put the web page on a remote server and, thus, make it accessible to anyone with a web browser. The HTML renders the HMI while the JavaScript script calls a PHP script which retrieves the new velocity instruction from the communication node (Figure 2.20).

2.7.2 Behavior

Since it was decided that the HMI would be designed for the instructor (the person who will sit beside the confederate car driver), the interface should display more information than just the velocity instruction. Therefore, the following data was decided to be shown:

- The velocity instruction
- The DTI of both vehicles
- The difference of the estimated TTI between both vehicles based on the current historic profiles selection
- The current confederate car velocity as available in the GPS.
- The variation of the speed instruction (either increasing or decreasing)
- The time before the driver should start (3 seconds before until start instruction)

In addition to this list of data, another *block* should display a Google map with the current position of both cars. All of this information is required so that the instructor can evaluate whether the given instruction is relevant or not.

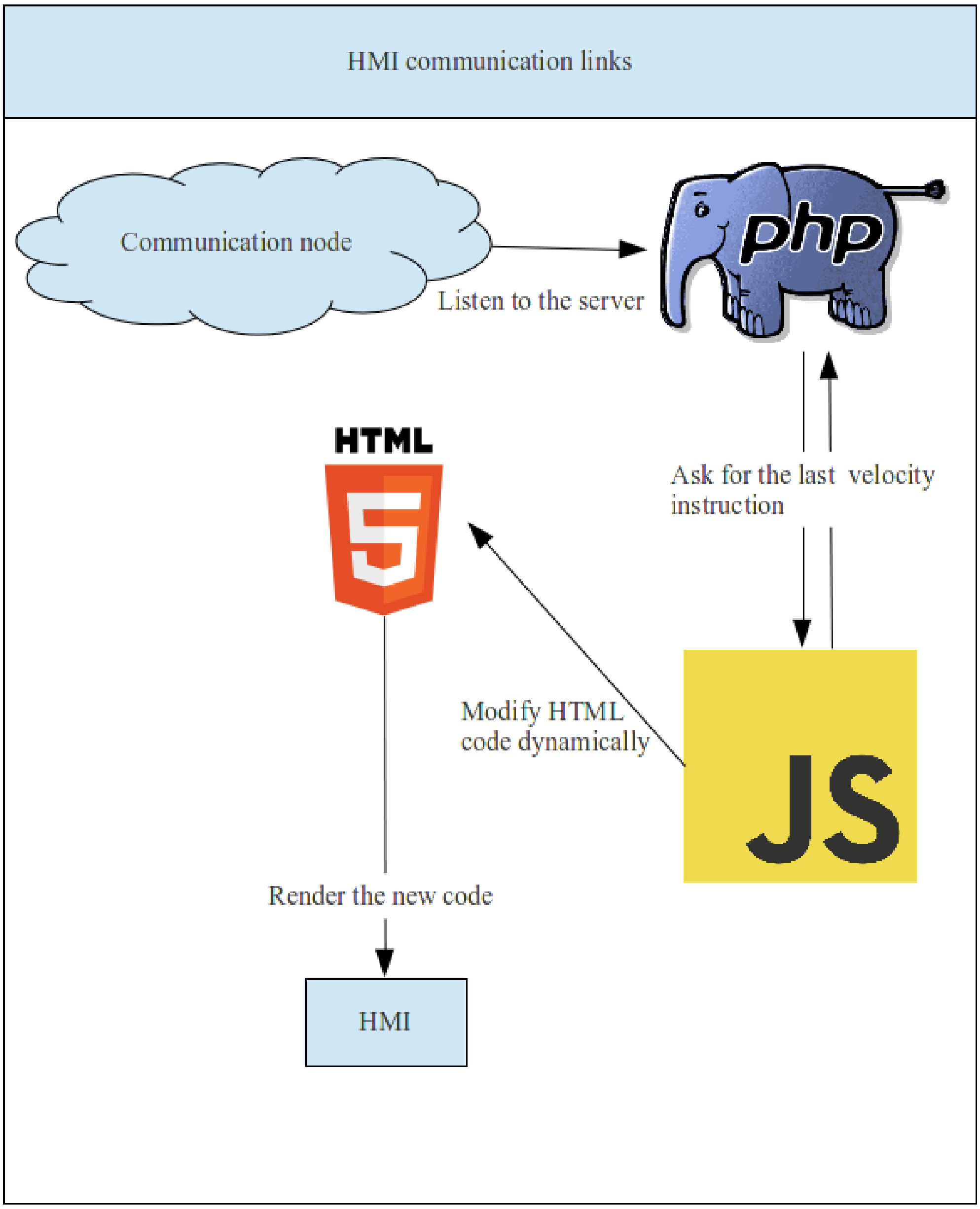

Figure 2.20: *HMI communication paths*

2.8 Testing

One important part of this project is testing the operation of the system. This includes testing of the functioning of the boards and the integration of the GPS and the 3G modem into them, the 3G internet connection on the boards, the GPS data acquisition, the exchange of data with the server, the HMI, the plotting of the GPS traces in Google maps on real time; and in general the performance of the whole system. The following sections detail how this testing was carried out.

2.8.1 Preliminary tests

Along the configuration and setup of the boards and their complementary systems (i.e. GPS and 3G modem), some preliminary tests were carried out to check that the different parts of the system were working as intended. During the first stages of the development the testing was mainly focused on evaluating the 3G internet connection on the boards, the GPS data acquisition and the data upload to the server. This testing was

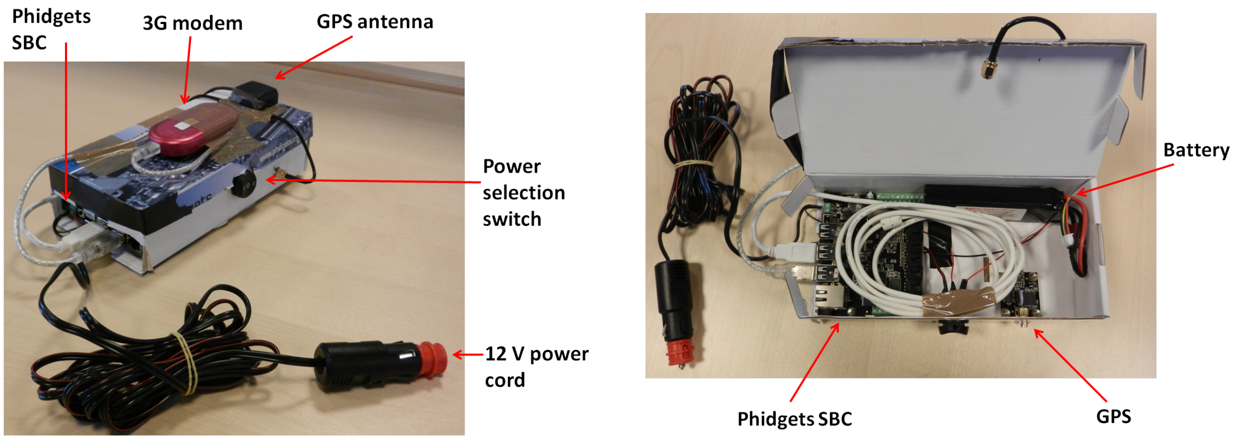

Figure 2.21: *Illustration of the final system with its components and casing*

performed indoors, assuring first the availability of the GPS signal. When it comes to the internet connection, the factors evaluated were the following: if the board was automatically establishing the connection and how fast was it doing it, the speed of the internet connection, and if the connection was maintained during the whole time the board was running. With respect to the GPS data acquisition and the data upload to the server, the aspects evaluated were: if the GPS device was correctly acquiring GPS data and the board sending that data to the server, if the delays were not too long, and finally if the board kept executing this process repeatedly and steadily until instructed otherwise.

Once it was verified that the board was carrying out all the previously mentioned tasks, the aim was focused on testing the behavior of the complete system (i.e. boards, cloud server, Matlab server, HMI and Google Maps traces). Since these tests needed to be performed outdoors, a simple casing for holding the boards and its complementary devices was built. Additionally, some special features were added to the system in order to ease its handling. These features included the possibility to power the board with either: grid power (230V), a battery, or a vehicle's 12 V supply; and a switch to easily change and select among these options. Figure 2.21 shows the final setup of the system. The preliminary outdoor tests were then performed by two persons walking simultaneously on different random paths, each holding one of the boards. The boards were activated and sending its GPS information to the cloud server. A third person was indoors checking the remaining aspects of the system in real time: the performance of the Matlab server, the exchange of data with the cloud server, the plotting of the paths in Google Maps, and the HMI giving a velocity instruction.

2.8.2 Pilot test on the field

Once the complete system was running properly, a pilot test on one of the intersections of interest was carried out. For this test, two vehicles provided by Volvo Cars were used; one was driven as the subject vehicle and the other one would be driven as the confederate vehicle. The Matlab server computer was the same computer used for displaying the HMI and the Google Maps interface, and thus this was a laptop computer that was placed in the confederate vehicle. The test procedure is detailed next.

The subject vehicle started at a distance of approximately 1 km from the intersection, while the confederate vehicle was parked at a distance of approximately 500 meters from the intersection. Once the two boards were running and sending GPS data; and the Matlab server, the HMI and the Google Maps were ready, the subject car was instructed to start driving towards the intersection. A second person inside the subject vehicle was constantly checking the HMI and the Google Maps traces. The main algorithm would thus eventually, based on the subject car's position and speed, send a go instruction to the subject vehicle by starting to send velocity instructions to the HMI. The driver of the confederate vehicle was then asked to maintain as close as possible the speed instructed through the HMI. It was then checked if both vehicles reached the intersection at the same time. This procedure was then repeated three times, for a total of four runs.

2.8.3 Final test on the field

Tuning of the system

In order to have the system perfectly calibrated, the tuning of parameters was required. There were two main parameters which should be tuned. The first one was the duration of the start-up phase. This duration parameter has a great importance since it controls the position of the start-up checkpoint as well as the speed at the beginning of the regulation phase. If this duration is too short, the driver won't have time to reach the target velocity and then the confederate car would be farther from the intersection than it was estimated by the algorithm. To get the best start-up duration, it has been asked to the drivers to accelerate to 45km/h from standing still state while keeping a comfortable level of acceleration. Finally, the start-up time is set to 6s, which leaves enough time to reach 45 km/h in a comfortable manner. The second parameter to tune was the prediction timespan by post-processing the data logged during the pilot test. The result of this process was the time-of-flight of the data from the GPS device to the end of the algorithm evaluation. The Figure 2.22 shows the travel of the data, what time was measured by the system and the total time which corresponds to the prediction timespan. An empiric estimation was used to determined this total time-of-flight from the GPS device to the HMI. The evaluation execution time (on the computational server) was considered really small compared to the total time, thus this duration was neglected. Then, as the travel of the data from the computational server to the HMI is the same as the one from the GPS to the computational server, it was considered that the total time-of-flight was the double of the measurable time. Finally, the prediction timespan was tuned to be equal to this total time.

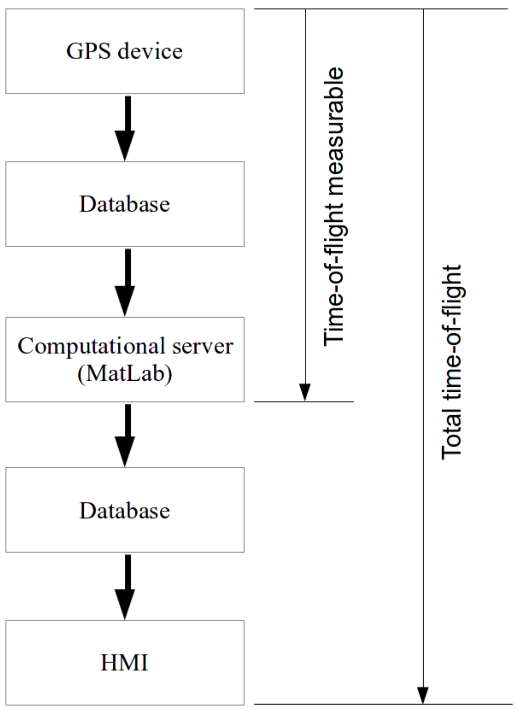

Figure 2.22: *Description of the travel of the data*

Evaluation of the system

The evaluation of the system passed by the evaluation of the difference in TTI (dTTI) at the intersection. To compare all situations, i.e. go and no-go decisions, a criteria had to be stated. This criteria was the current dTTI when the Subject car was passing through the Stabilization checkpoint. In the test, it was 40 meters before the center of the intersection. This choice is relevant because after this checkpoint there is no more velocity regulation and the decision to go or not to go seemed to be after this checkpoint when comparing the dynamic of the vehicle. To evaluate the behavior of the system with different dTTI target, this one has been tuned to +2, 0 and -2 seconds and several runs have been performed. Finally, the criteria was extracted and compared to the target submitted within the MatLab application.

3 Results

3.1 Hardware selection

This section presents the results of the procedures performed to compare and ultimately select the hardware to be used for GPS data acquisition, for the communication between the boards, and for controlling these two devices.

3.1.1 GNSS selection

The results of the market survey of the different GNSS technologies showed that a rather wide range of performances could be achieved. For instance, position accuracies ranging from 0.3 m up to 5 m could be reached with current devices. When it comes to communication interfaces, different possibilities could also be found: memory cards, USB, RS 232, Bluetooth, Wi-Fi, cellular, WLAN. Update rates ranging from 1 Hz up to 20 Hz were available. However, those devices with the highest performance characteristics were obviously linked to higher prices. The price range was between 94 SEK up to 17 000 SEK. A complete summary of the market survey devices can be found in the comparative table in Appendix XX.
Figure 3.1 shows the distribution of the measured GPS coordinates errors around the actual spots coordinate for the different devices. These results are also summarized in Table 3.1. The results show that the device that exhibited the highest position precision is the Samsung Galaxy GIO smartphone, since it showed the lowest standard deviation on the measurements. This means that if repeated measurements were to be carried out on the same spot (i.e. same latitude – longitude pair), the measurements would tend to be the same, or very similar among each other. The same device also exhibited the highest position accuracy, since the mean of its measurements were the ones closest to zero. This means that this is the device that most closely measures the real coordinate values. On the other hand, the device that evidenced the worst performance in both of these aspects was the Phidgets GPS.

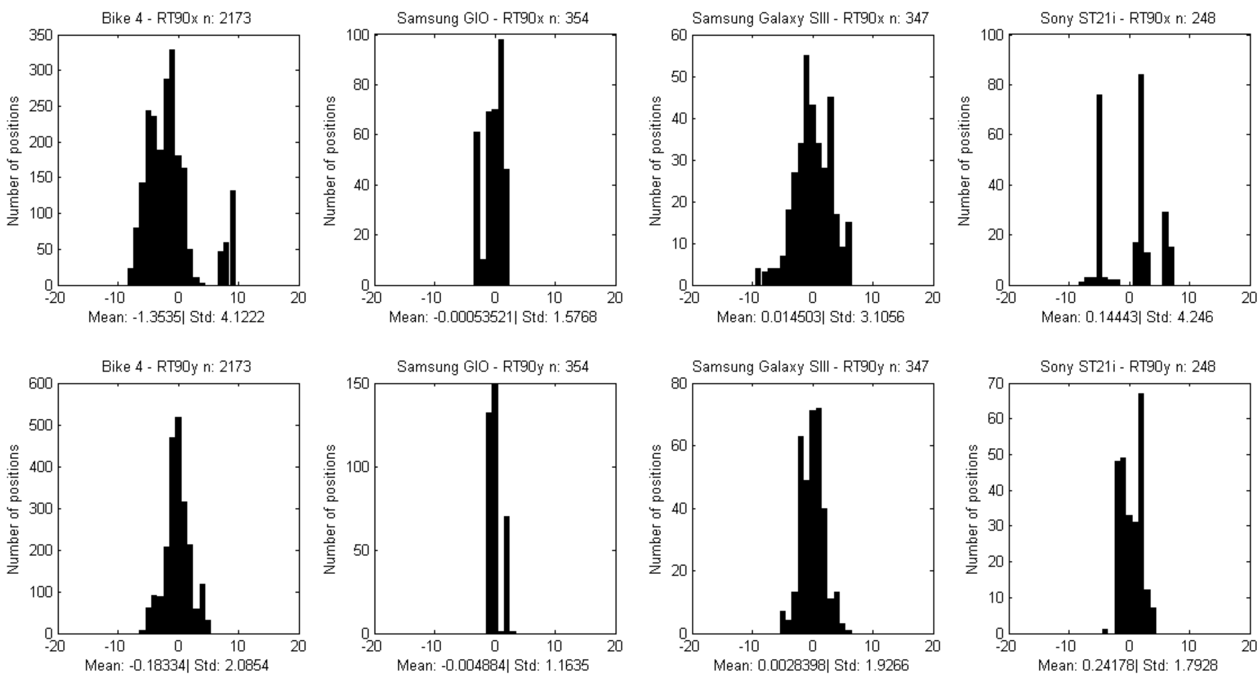

Figure 3.1: *Distribution of GPS coordinates (RT90) measurements for the different devices tested in the position precision test*

Figure 3.2 shows the evolution of heading with time for the different devices during the velocity accuracy test. The results of the velocity accuracy test show that all four devices had roughly a similar performance when it comes to measuring the heading of the vehicle, see Figure 19. In the figure it can be seen that all the

Table 3.1: Position precision test results

Device	RT90x [m]	RT90y [m]
Phidgets	-1.354 ± 4.122	-0.1833 ± 2.0854
Samsung GIO	-0.00005352 ± 1.577	-0.004884 ± 1.1635
Samsung GSIII	0.01450 ± 3.106	0.002839 ± 1.927
Sony ST21i	0.1444 ± 4.246	0.2418 ± 1.793

devices were acceptably measuring correctly the real heading of the bicycle, being it 0° for the first trip, which was intended to be from south to north; and 180° for the return trip, i.e. for north to south.

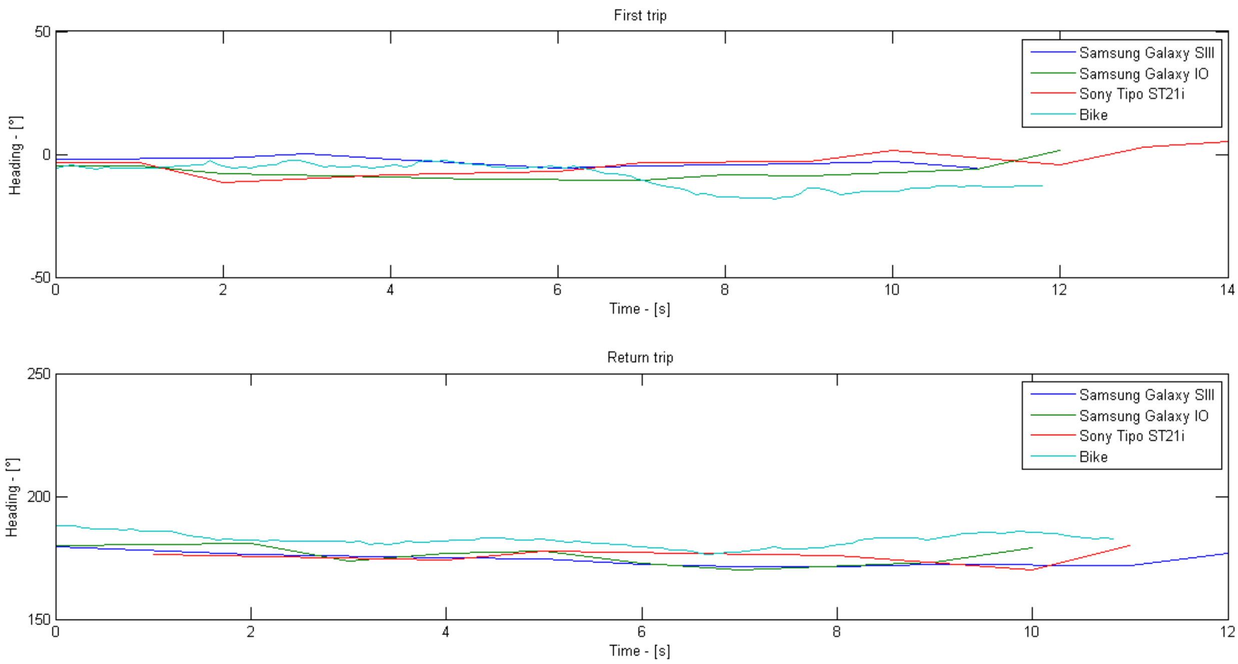

Figure 3.2: *Evolution of heading with time for the different tested devices during the velocity accuracy test*

Figure 3.3 and Figure 3.4 show the GPS path traces for the different devices in the velocity accuracy test, for both the south – north trip and the north – south trip (return trip). The GPS traces of this test show that in general the four devices traced a straight line, as it was intended. Furthermore, this agrees with the results of the evolution of heading, as stated previously. It is however worth noticing that the Phidgets GPS (labeled as bike in the figure) has subsequent measurements really close to one another, whereas all three smartphones have them much more separate. This is a consequence of the higher update rate of the Phidgets GPS (10 Hz) compared to the smartphones (1 Hz).

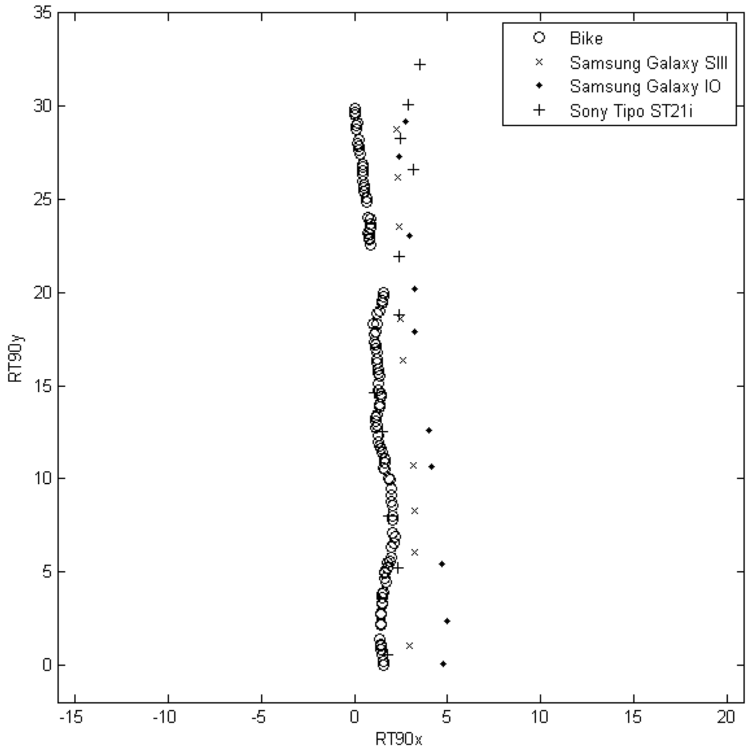

Figure 3.3: *GPS path traces for the different devices during the velocity accuracy test. South - north trip*

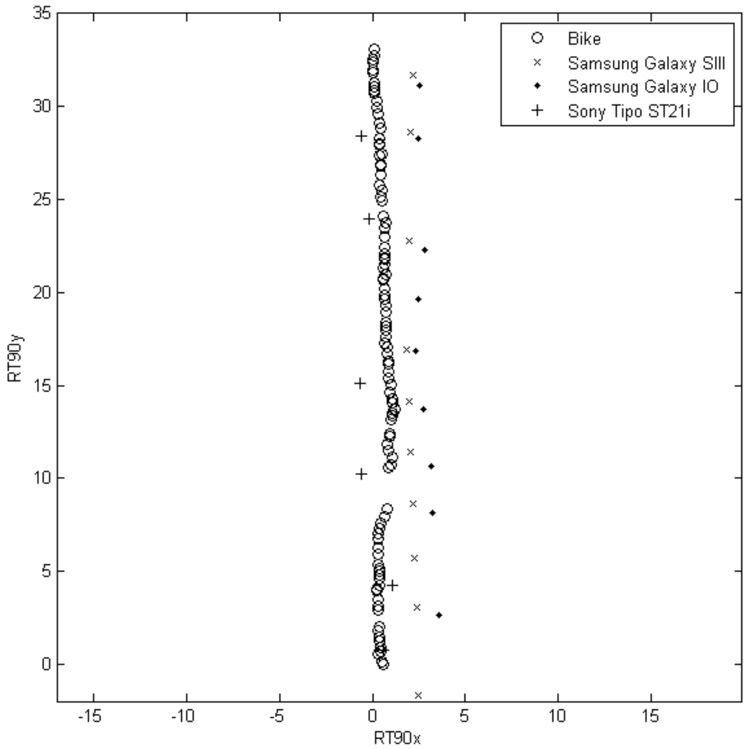

Figure 3.4: *GPS path traces for the different devices during the velocity accuracy test. North - south trip*

3.2 Server timing analysis

This section presents the results of the tests performed for evaluating the timing for data exchange with the cloud server.

Table 3.2 summarizes the results for the tests using a Wi-Fi connection, while Table 3.3 shows the results for the tests using a 3G connection. The results of the tests performed for the server timing analysis show that the Setsuperbeacon and Subscribe2 set of functions were the fastest functions when it comes to both uploading and downloading data to and from the cloud server. On the other hand, the Set and Get functions were the ones exhibiting the slowest performance, especially when all the set of variables were uploaded and downloaded at the same time. Both of these statements apply for the Wi-Fi connections tests as well as for the 3G connection tests. Other relevant things to notice from these tests include the fact that the upload time was in all cases higher than the download time. Also noticing that the time taken in the upload + download test is roughly the same time as the sum of the times of the individual tests of upload and download, which goes hand in hand with what could be intuited. Finally, it is worth noticing that in almost all the tests the time is at least doubled with the 3G connection as compared to the Wi-Fi connection. Results for the download time with the Setsuperbeacon and Subscribe2 functions are not available due to the fact that the subscribe function inherently includes both the upload and the download time (see section II.2.2), and so the download time alone cannot be easily measured.

Table 3.2: Timing results over Wi-Fi connection

Wi-Fi connection	Upload time [ms]	Download time [ms]	Upload & download time [ms]
Set and *Get* functions, one variable at a time N=950	• Mean: 305.1 • Std: 137.8	• Mean: 124.3 • Std: 55.0	• Mean: 420.8 • Std: 76.6
Set and *Get* functions, all set of variables at the same time N=950	• Mean: 1873.2 • Std: 204.4	• Mean: 727.3 • Std: 240.2	• Mean: 2585.8 • Std: 327.7
Upload and *Data* functions N=950	• Mean: 147.7 • Std: 68.8	• Mean: 119.3 • Std: 43.1	• Mean: 266.4 • Std: 69.7
Subscribe2 and *Setsuperbeacon* functions N=950	• Mean: 136.7 • Std: 27.5	• Mean: 125 • Std: 28.2	• Mean: - • Std: -

Table 3.3: Timing results over 3G connection

3G connection	Upload time [ms]	Download time [ms]	Upload & download time [ms]
Set and *Get* functions, one variable at a time N=950	• Mean: 558.6 • Std: 239.1	• Mean: 347.9 • Std: 134.4	• Mean: 887.4 • Std: 292.5
Set and *Get* functions, all set of variables at the same time N=950	• Mean: 3383.4 • Std: 925.9	• Mean: 2100.1 • Std: 645.7	• Mean: 5681.9 • Std: 1422.8
Upload and *Data* functions N=950	• Mean: 428.6 • Std: 186.1	• Mean: 401.8 • Std: 159.7	• Mean: 791.7 • Std: 310.1
Subscribe2 and *Setsuperbeacon* functions N=950	• Mean: 375.5 • Std: 174.9	• Mean: - • Std: -	• Mean: 695.7 • Std: 270.1

3.3 Historic profiles analysis and selection

This section contains the results of the historic profiles and analysis and the selection algorithm.

3.3.1 Extraction and filtering of historic profiles

Figure 3.5 and Figure 3.6 show the historical DTI vs. TTI profiles used in the profile selection algorithm. Figure 3.5 shows the profiles without filtering for the subject vehicle on the Grevegårdsvägen – Opalgatan intersection, whereas Figure 3.6 shows these profiles after the filtering criteria had been applied and the duplicate profiles had been removed. Both figures illustrate the evolution of the time to reach the intersection as the vehicle approaches it (i.e. the progress of distance to intersection), for several passes (each curve represents a pass). A comparison of these figures reveals how the aim of filtering out some of the profiles was to remove those were the vehicles were standing still for some time (curves that remain at a constant DTI over a range of TTI in Figure 3.5), plus those that were clearly out of the common pattern (curves that cover the lower region of the plot in Figure 3.5).

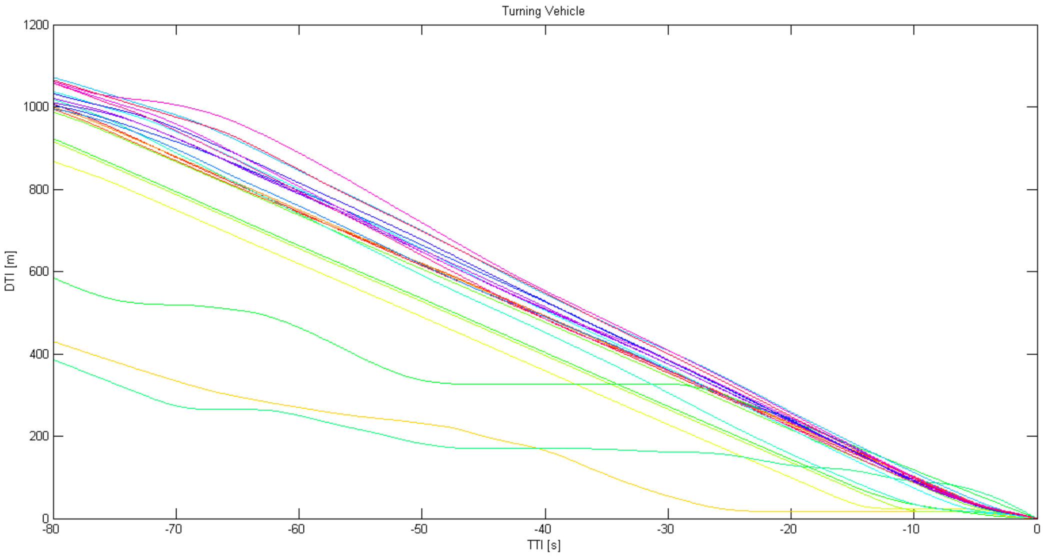

Figure 3.5: *Subject vehicle profiles without filtering*

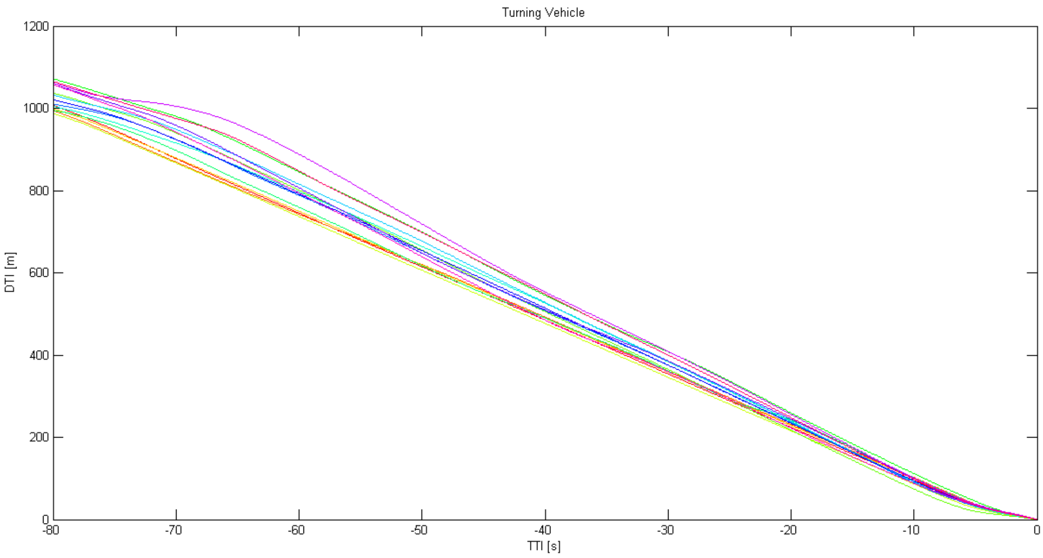

Figure 3.6: *Subject vehicle profiles with filtering*

3.3.2 Selection algorithm

An example of a reference profile created under simulation, using a randomly selected historic profile as virtual input of the "real vehicle", is shown in Figure 3.7 for the case of the subject vehicle. In this case, the reference profile (blue thick curve) is the mean of the historic profiles. The input profile is the red thick curve. The thin curves are the historic profiles used for predicting the reference profile.

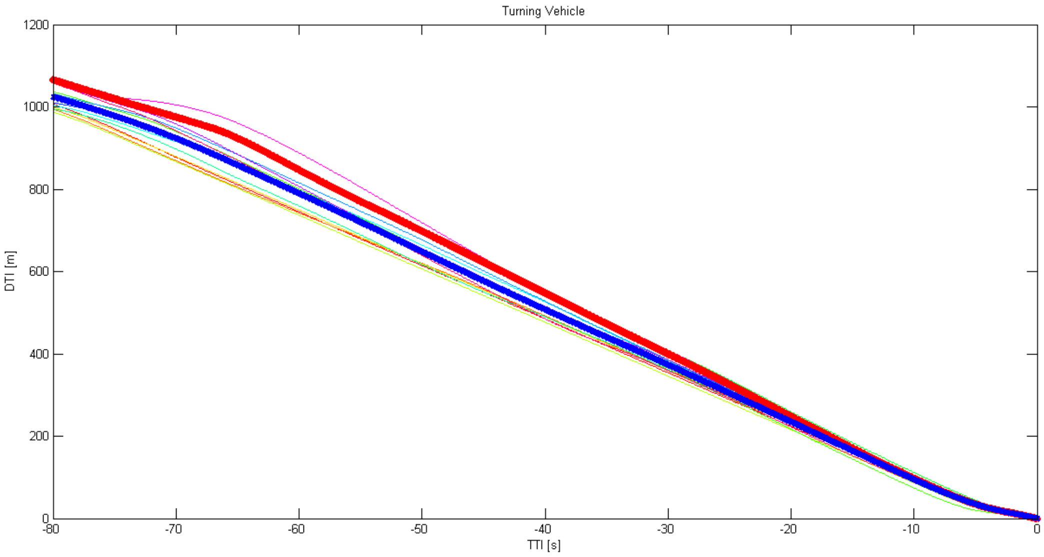

Figure 3.7: *Predicted profile (thick blue) compared to the input profile (thick red). Predicted profile created with the mean profile approach. The thin curves are all the historic profiles used for the prediction*

The same results are shown from Figure 3.8 to Figure 3.11 for the different prediction criteria, using the same randomly selected "real" profile for the purpose of comparison. Figure 3.8 shows the result for the median profile as the reference one; Figure 3.9 is for the speed based approach; Figure 3.10 for the moving window approach, using a window of 3.10 points; and Figure 3.11 for the weighting factors and moving window approaches, using a moving window of 3.10 points as well as weighting over the three most suitable profiles.

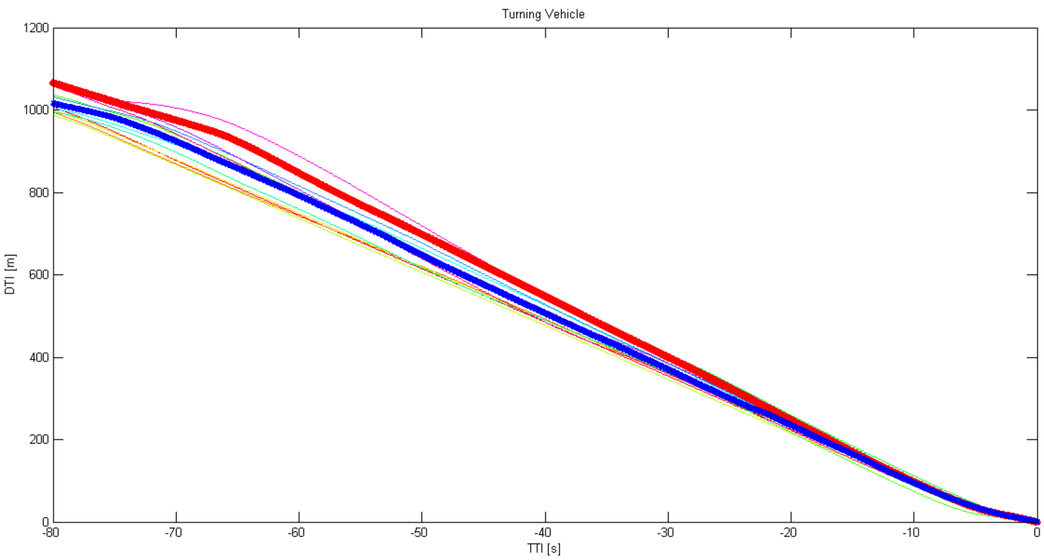

Figure 3.8: *Predicted profile (thick blue) compared to the input profile (thick red). Predicted profile created with the median profile approach. The thin curves are all the historic profiles used for the prediction*

Figure 3.7 illustrates the behavior of the selection algorithm when the mean profile criterion is applied, while Figure 3.8 shows the behavior when the median profile criterion is applied. These two criteria have the advantage of creating a reference profile (blue thick curve) that is smooth.

As seen in the example of Figure 3.9, the speed criterion by itself does not seem to be reliable enough as it does not create a smooth profile curve, but also in some DTI points it does not provide a TTI close to the real one.

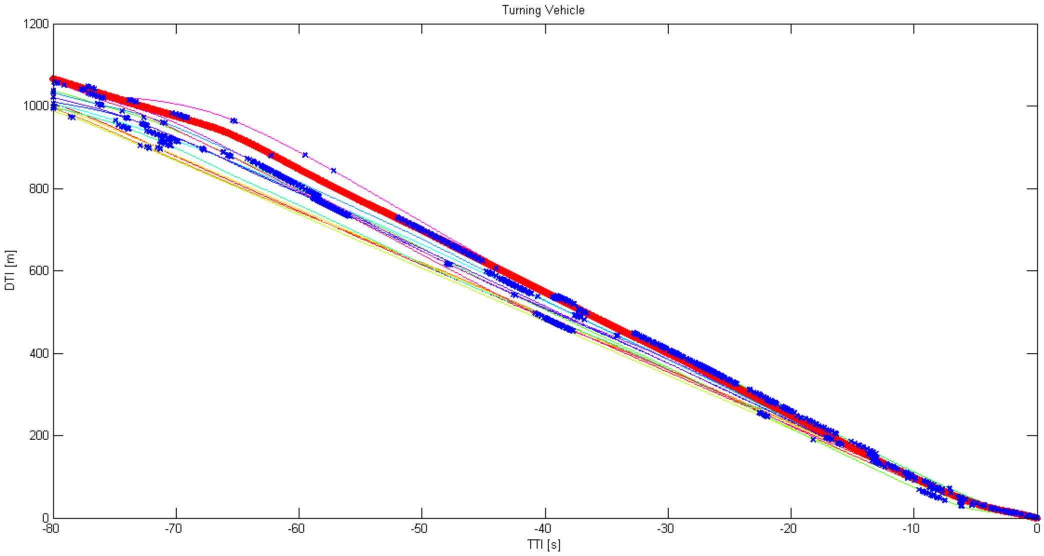

Figure 3.9: *Predicted profile (thick blue) compared to the input profile (thick red). Predicted profile created with the speed based approach. The thin curves are all the historic profiles used for the prediction*

The results of the first improvement criterion (the moving window approach), as illustrated in Figure 3.10, seems to produce small traces of smoother but yet discontinuous reference profiles curves, and also seems to produce predictions a bit closer to the real profile.

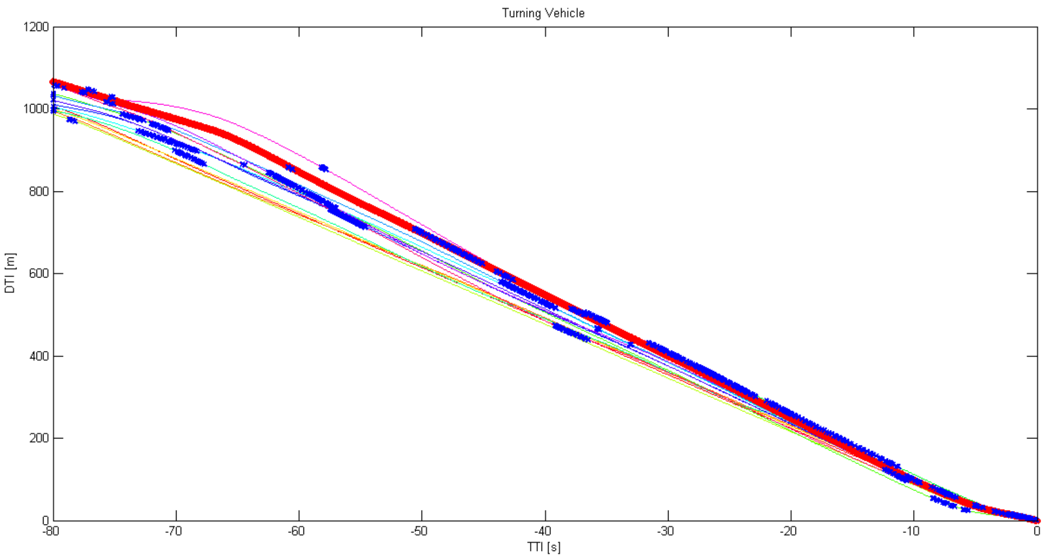

Figure 3.10: *Predicted profile (thick blue) compared to the input profile (thick red). Predicted profile created with the moving window approach. Window size: 30 points. The thin curves are all the historic profiles used for the prediction*

A final criterion was implemented with the aim of further improvement: the weighting factors approach. As seen in Figure 3.11, this method seems to produce a fairly smooth and continuous reference profile curve that at some points comes rather close to the real profile curve (or even over it).

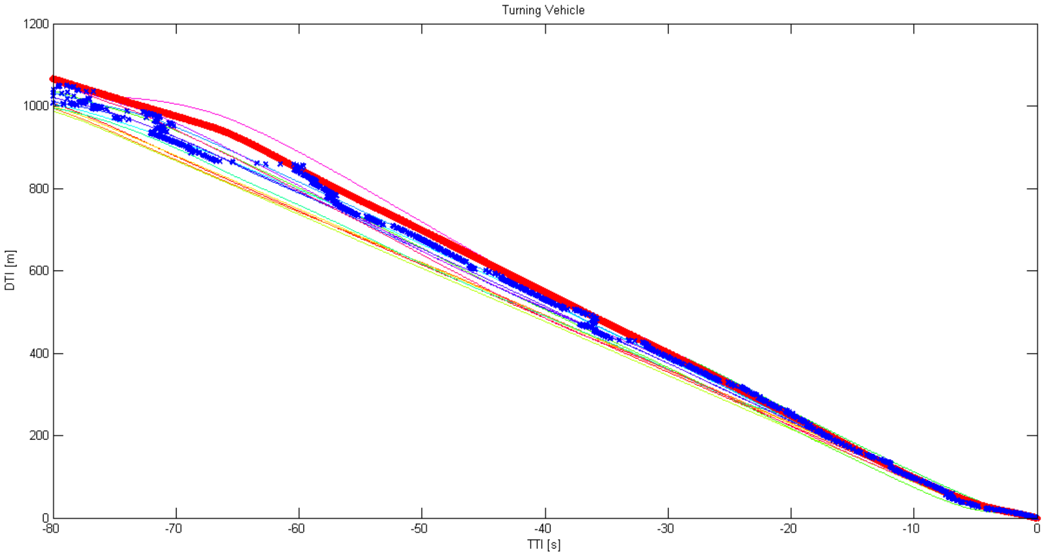

Figure 3.11: *Predicted profile (thick blue) compared to the input profile (thick red). Predicted profile created with the weighting factors and the moving window approaches. Window size: 30 points, number of weighted profiles: 3. The thin curves are all the historic profiles used for the prediction*

Table 3.4 summarizes the results of the evaluation of the different prediction criteria for the 15 simulation runs performed, based on the performance indicators previously defined. As seen in the table, the five different profile selection criteria exhibited similar results when evaluated under the defined performance indicators, with a maximum difference of about 0.3 seconds between the better performed criterion and the worst performed criterion on the different evaluated DTIs. The fact that the weighting factors plus the moving window approach had the best performance when compared to the speed based and the moving window approaches confirms the initial need and ultimately the improvement achieved by implementing these additional criteria. Finally, it can be seen that as would be expected, the mean of the TTI difference between the predicted and the real profile decreases as it is measured closer to the intersection (smaller DTIs).

Table 3.4: Results of the evaluation of the different prediction criteria, based on the defined performance indicators

	DTI = 100		DTI = 75		DTI = 50		DTI = 25	
Approaches	Mean [s]	σ [s]	Mean [s]	σ [s]	Mean [s]	σ [s]	Mean [s]	σ [s]
Mean	0.48	0.43	0.53	0.43	0.43	0.47	0.39	0.44
Median	0.52	0.40	0.52	0.42	0.45	0.48	0.32	0.47
Speed based	0.65	0.87	0.72	0.64	0.67	0.51	0.61	0.78
Moving window	0.66	0.51	0.59	0.85	0.72	0.78	0.46	0.61
Weighting factors + moving window	0.65	0.46	0.55	0.64	0.60	0.65	0.44	0.46

3.4 Algorithm simulations

3.4.1 Confederate car model simulation

Figure 3.12 shows the behavior of the confederate car model. The blue curve representing the model increases towards the velocity target for the first phase ($-305m < DTI < -280m$). Then it follows the velocity instruction provided by the algorithm until $DTI = -70m$. There is the stabilization phase where the curve goes to 45km/h (the target velocity). At 48m before 0, the instruction stops, the red curve disappears while the blue curve keeps going to 45km/h.

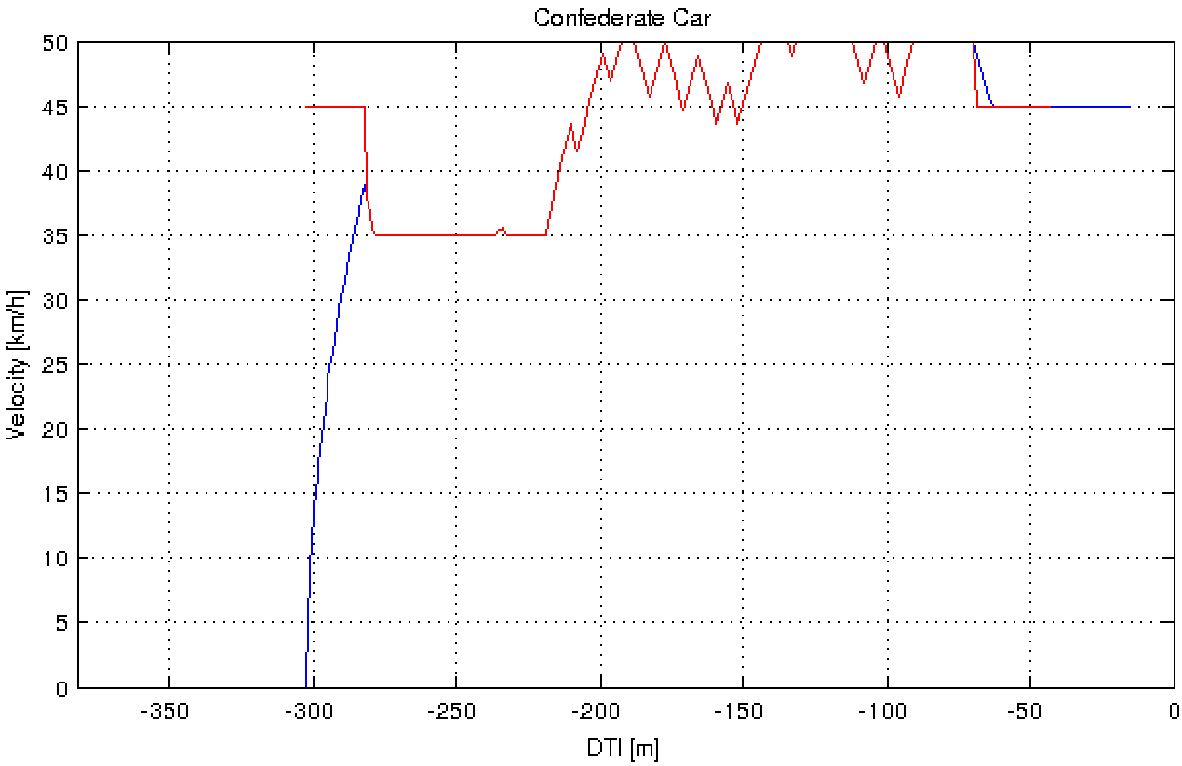

Figure 3.12: *Simulation result: right-top graph shows instruction (red) & confederate car velocity (blue)*

3.4.2 Evaluation of the algorithm overall behavior

The distribution of the initial confederate car's DTI can be seen on Figure 3.13. This distribution of inputs is set within the interval of [-310, -200] m and the subject car profiles have been picked up randomly in the data collected, the simulation ran 50 times. The results of the simulation are shown in Figure 3.14. The resulting difference in TTI is between -700ms and -500ms, where negative means that the confederate car is arriving at the intersection late compared to the subject car. The GPS update rate being 10Hz, the result has a resolution of 100ms.

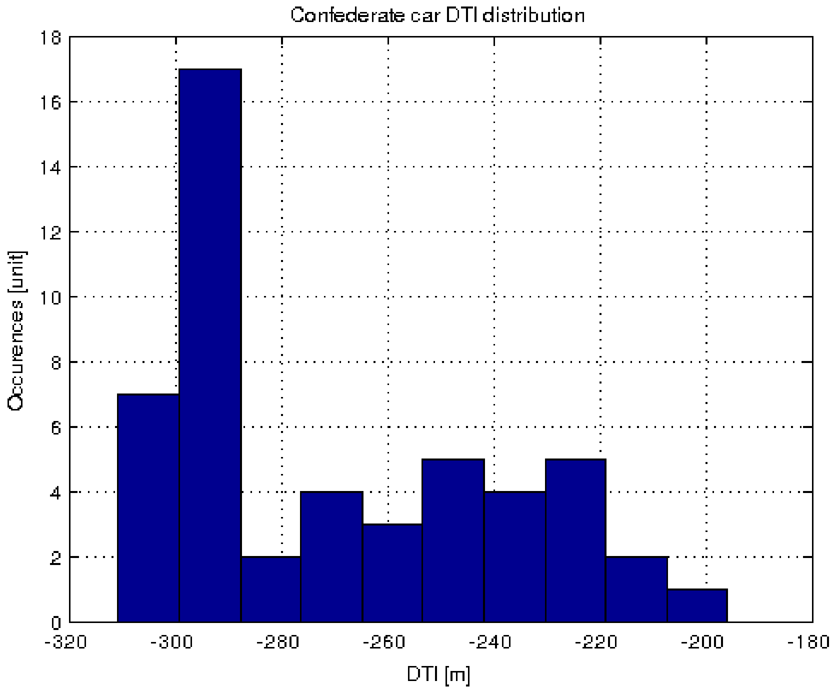

Figure 3.13: *Distribution of the confederate car DTI*

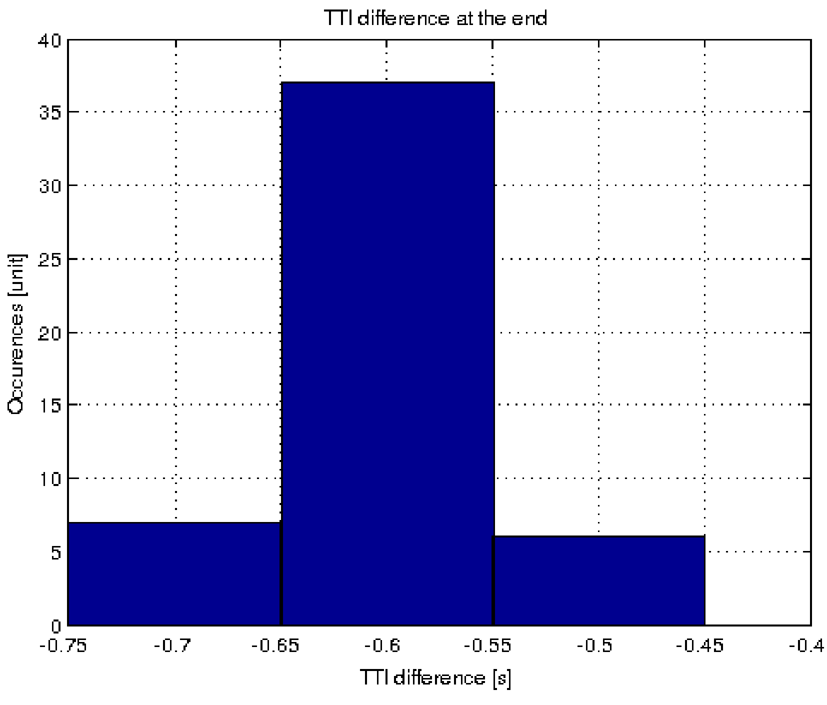

Figure 3.14: *Distribution of the final difference in TTI between both cars*

3.5 HMI development

Since the technology chosen is the set of web languages, a standard web browser can be used for rendering the HMI. The HMI is divided into two columns; the left-column shows the GPS traces in real time while the right-column shows the instructions. Figures 3.15 to 3.20 show the interface for the 6 different steps; *waiting*, *be ready*, *start-up*, *regulation*, *stabilization* and *releasing*. Some comments should be done on these steps. The be ready step shows the progression of the SV on the progression bar, the units are in seconds, this means that at 0 the confederate car should start up. The releasing step does not give any more instruction, the confederate driver is fully the master of the situation.

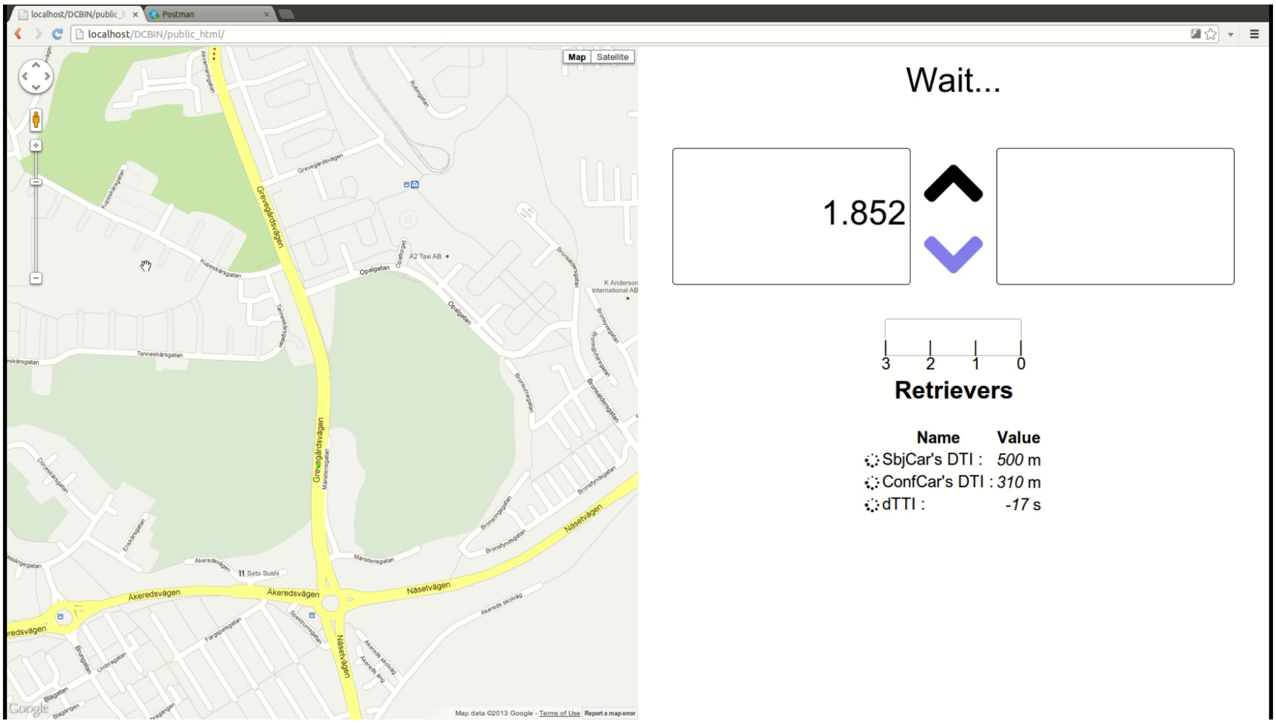

Figure 3.15: *HMI is waiting for instruction*

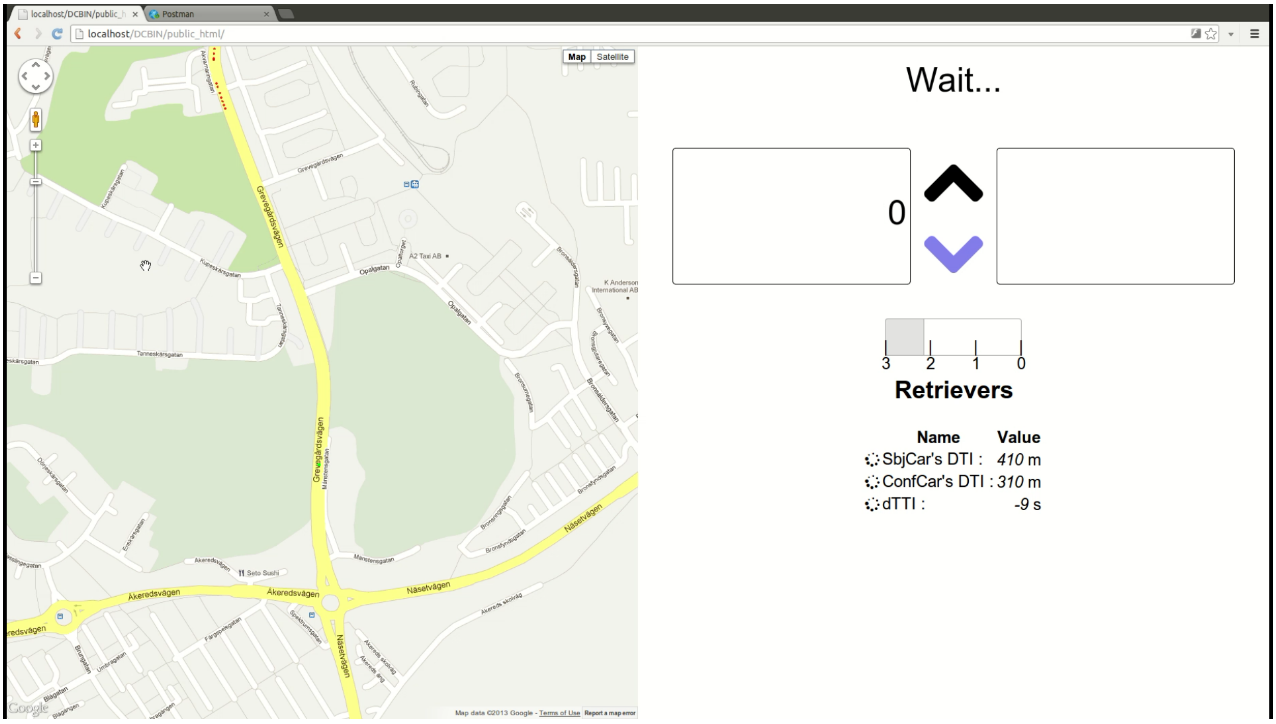

Figure 3.16: *HMI instructs the confederate driver to be ready, here 2 seconds remain before start-up*

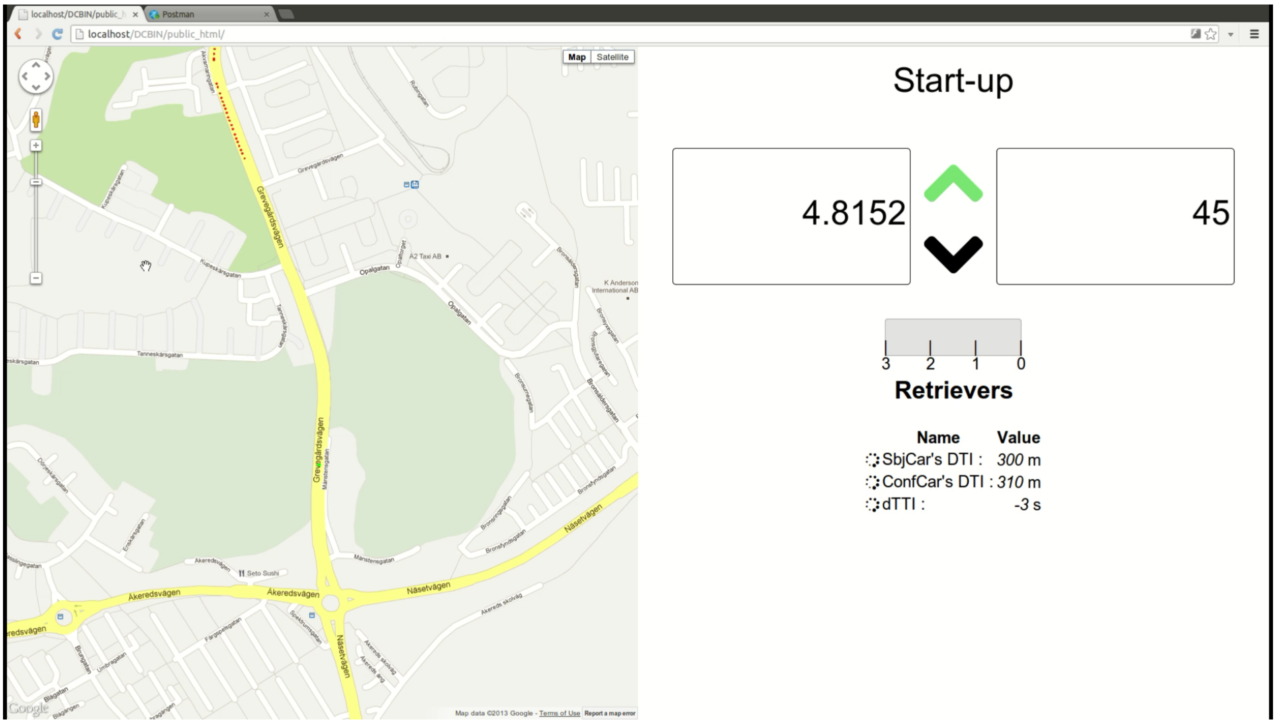

Figure 3.17: *HMI instructs the confederate driver to start and reach the target velocity (45km/h)*

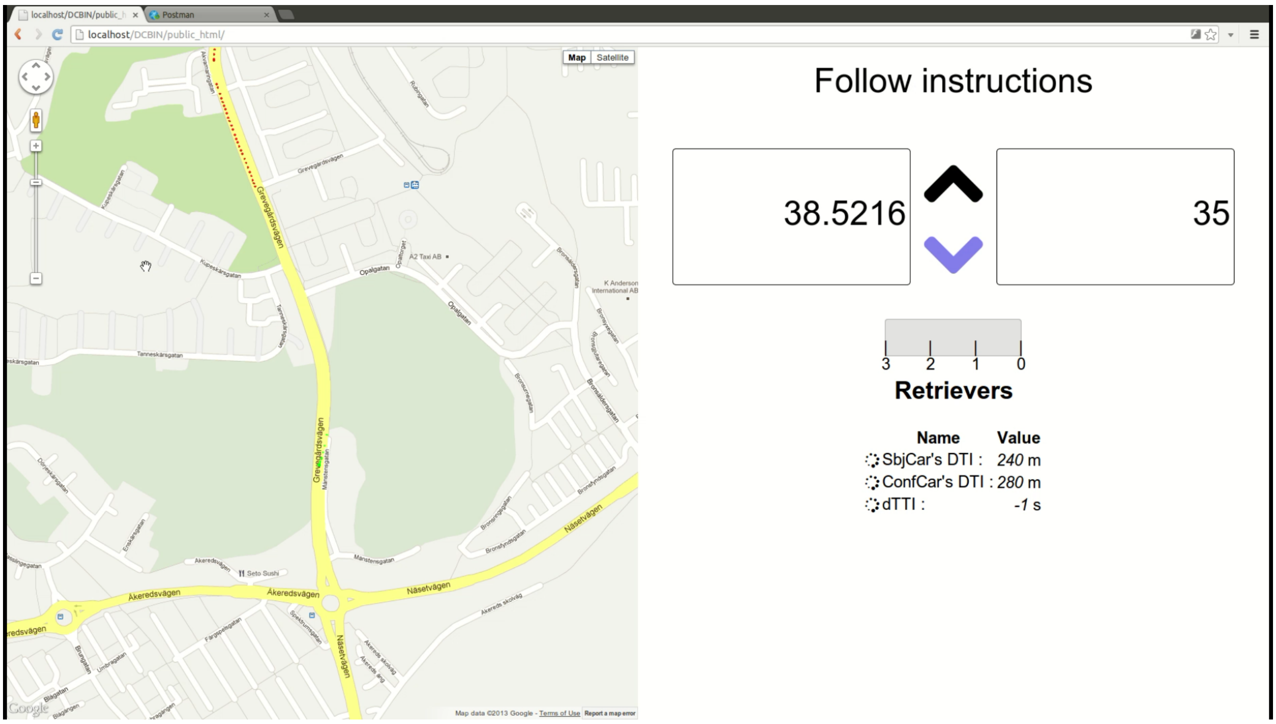

Figure 3.18: *HMI instructs the confederate driver the velocity to follow*

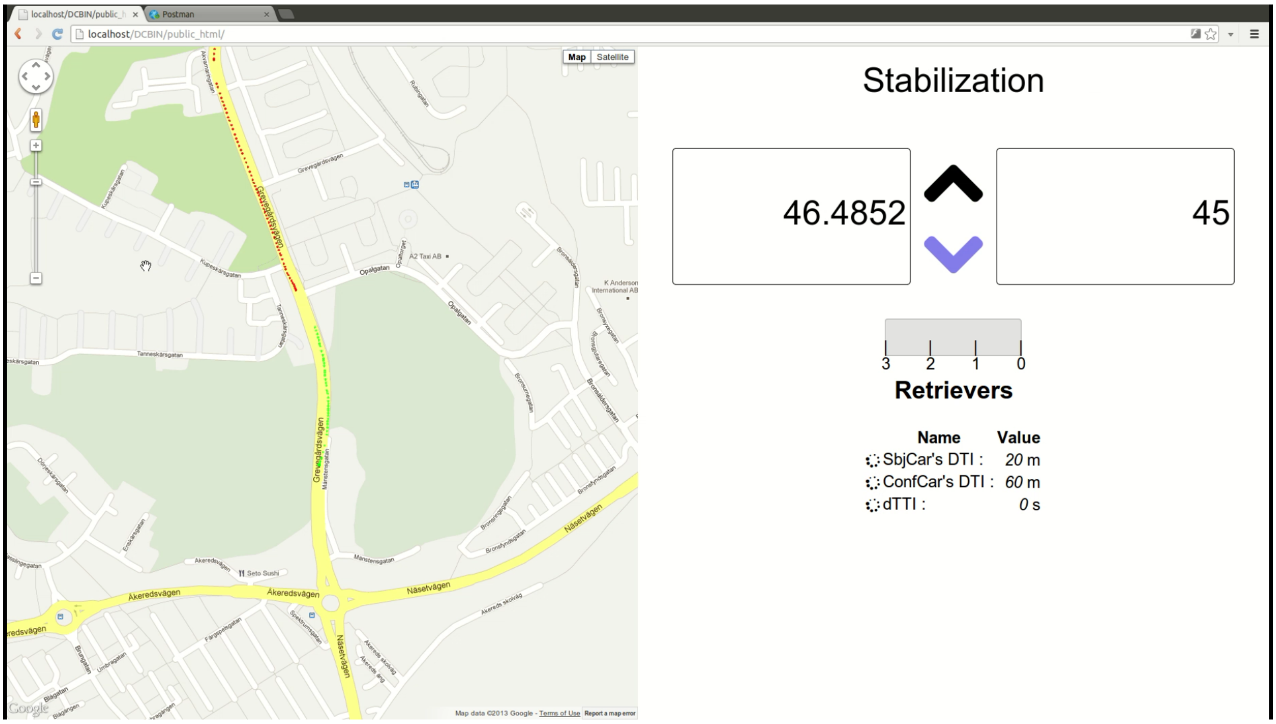

Figure 3.19: *HMI instructs the confederate driver to stabilize at the target velocity (45km/h)*

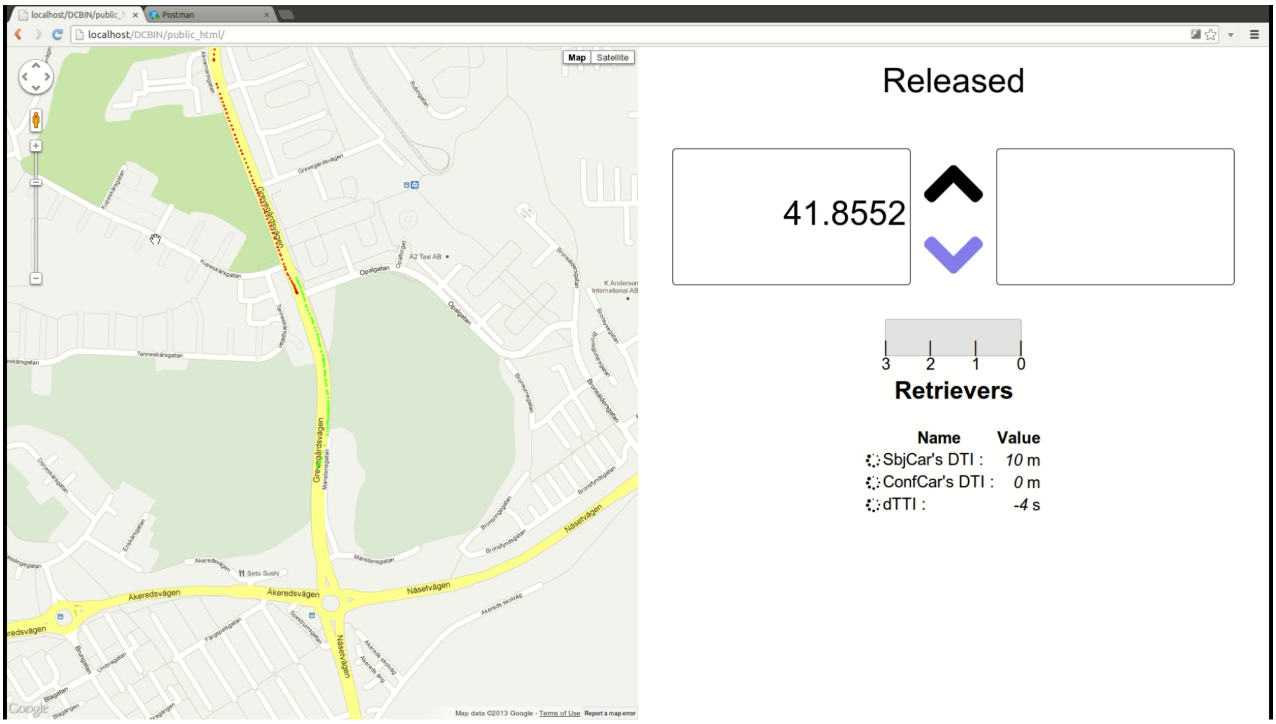

Figure 3.20: *HMI does not provide any more instruction to the confederate driver*

3.6 Testing

This section presents the results of all the tests performed to evaluate the performance of the final system.

3.6.1 Preliminary tests

Figure 3.21 shows the traces of the GPS paths on Google Maps for both of the boards on the walking tests. The green and red curves represent the trace of one board each. Both boards were held by the same person at the same time during the test. The figure confirms what had been observed in the position precision tests (see section 3.1.1) in the sense that the precision of the Phidgets GPS devices is not the best, since the path of both curves is very dissimilar and none of them represent completely the actual path taken by the walker. It can be seen however that the update rate of the GPS device is good enough to produce a continuous trace.

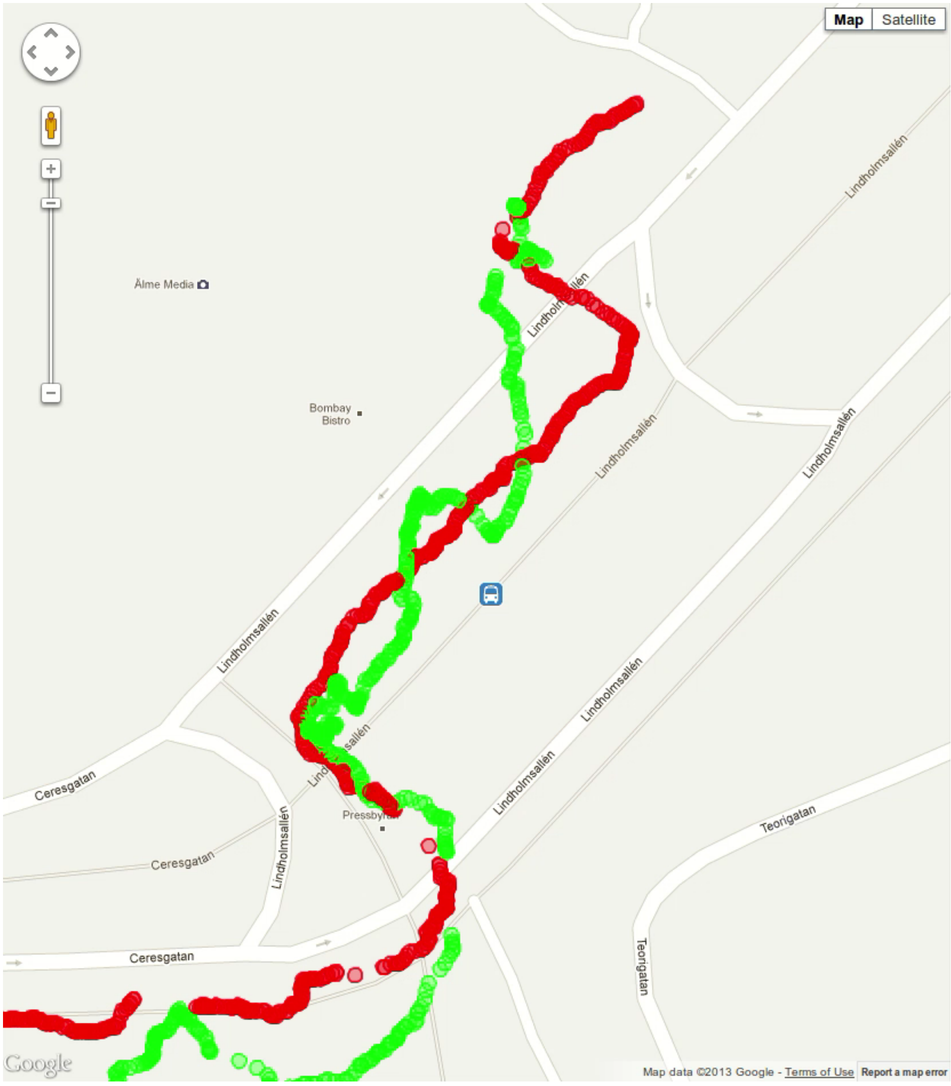

Figure 3.21: *Traces of the GPS paths*

3.6.2 Pilot test on the field

Figure 3.22 shows traces of the GPS paths on Google Maps along with the preliminary version of the HMI. On the Google Map traces, the green trace represents the confederate vehicle and the red trace represents the subject vehicle. The number on the HMI shows the velocity that the driver of the confederate vehicle should

follow.
In the four passes through the intersection carried out in this test it was seen that the system performed as expected: GPS information from both vehicles was fetched, sent to the server and retrieved by the Matlab computer; the latter performed the calculations and sent the appropriate instructions to the HMI; and the HMI displayed this instructions in all of the phases: waiting for instructions in the start-up phase, show target speed during the first acceleration, show the required velocity during the regulation, and provide no further instructions after the release phase. It was additionally observed that in the four passes both vehicles reached the intersection at roughly the same time, i.e. at a time where the driver of the subject vehicle was forced to make a go/no-go decision due to the presence of the confederate vehicle.

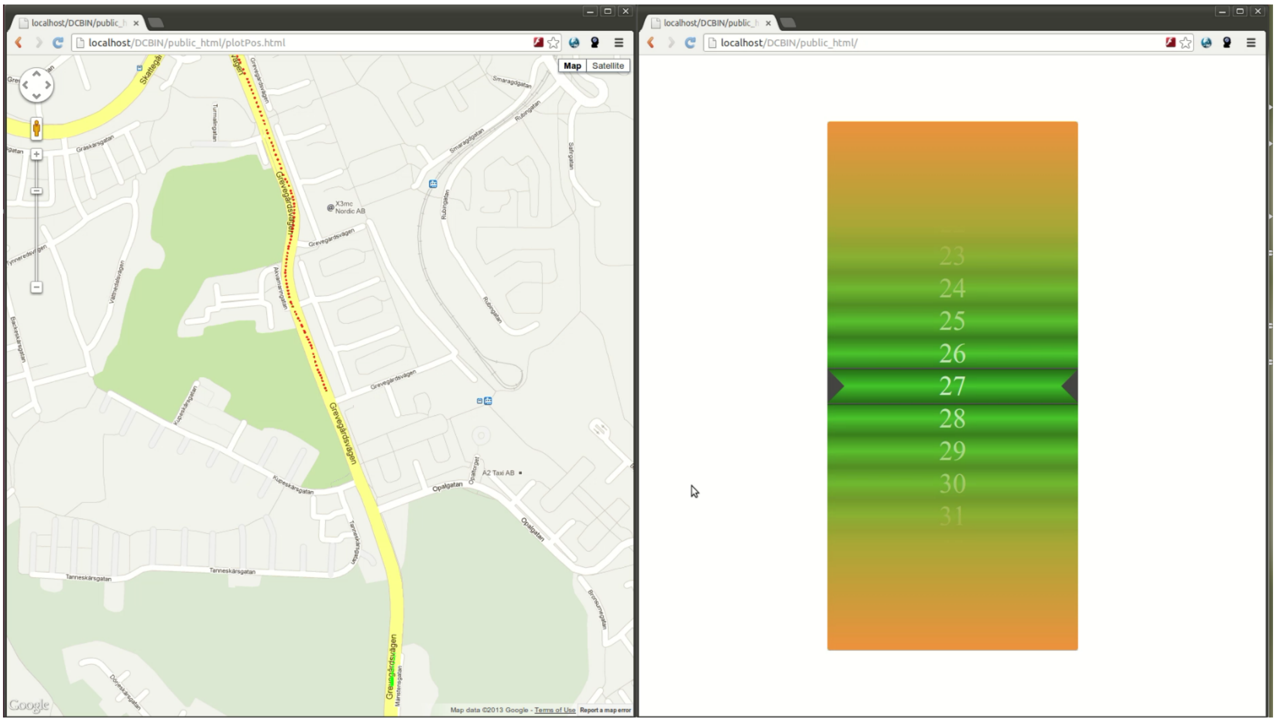

Figure 3.22: *On the left: screen shot of the Google Maps traces of the subject vehicle (red trace) and the confederate vehicle (green trace). Highlighted with a red circle is the intersection for intended interaction. On the right: preliminary version of the HMI.*

3.6.3 Final test on the field

Figure 3.23 shows the distribution of the time-of-flight from the GPS to the instruction evaluation, the distribution could be considered as exponential, the mean value being 542ms with a 95% confidence interval of [496, 595] ms. Therefore and following what stated in 2.8.3, the whole time-of-flight for the information from the GPS to the HMI is estimated to be 1.2s to keep a small safety margin.
Because of a dense traffic, only few runs were relevant and exploitable. The Table 3.5 shows the dTTI criteria for each of them. The closer to zero the criteria is, the better is the performance of the system.

Table 3.5: Results of the system evaluation

dTTI target [s]	Performance indicator [s]
+2	+0.33
0	+0.85
-2	-0.45
-2	-0.15
-2	-0.50

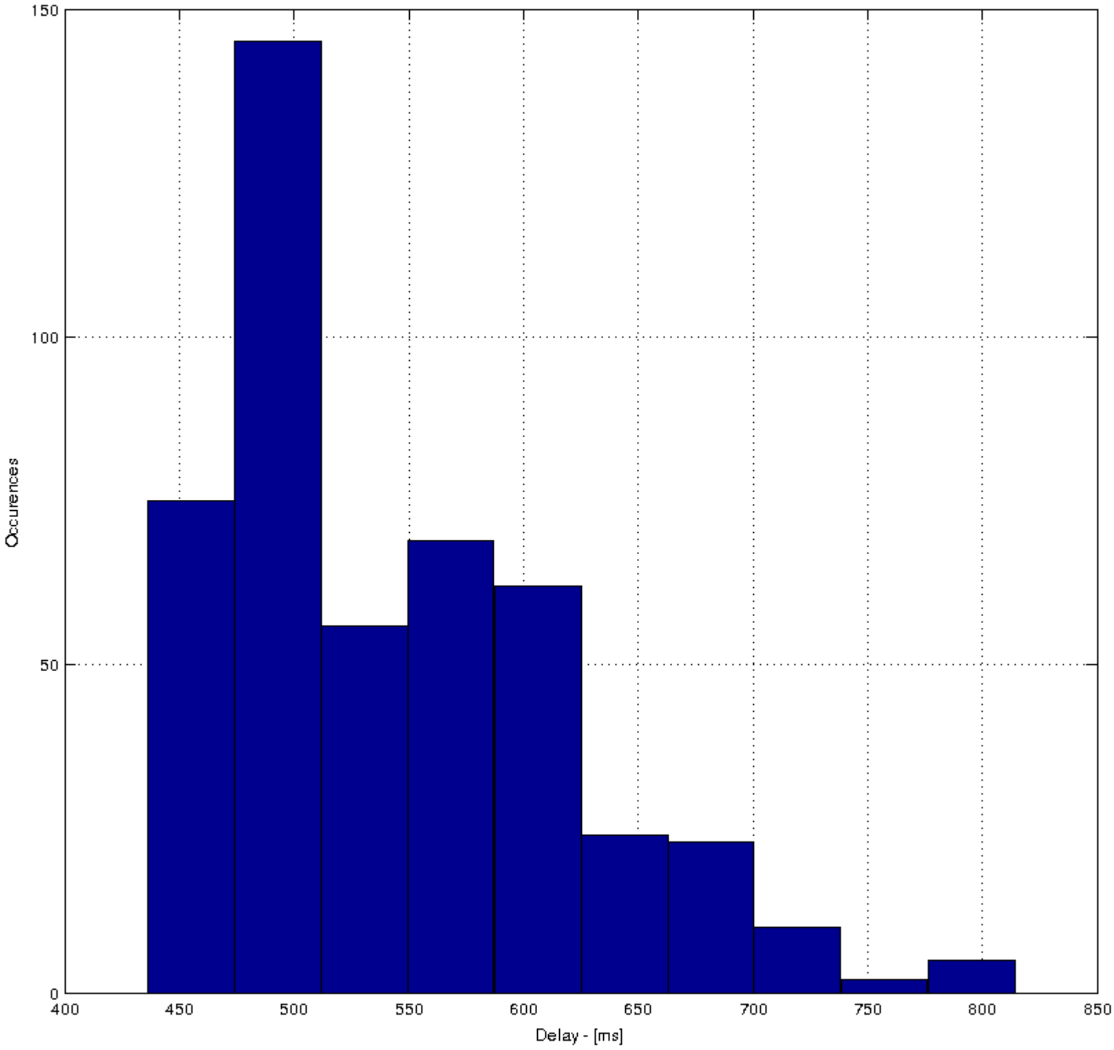

Figure 3.23: *Distribution of the delays from GPS update to the instruction evaluation*

4 Discussion

4.1 Hardware selection

4.1.1 GNSS selection

The market survey of the different GNSS technologies showed that, in general, all the requirements specified for this device in the Requirements Specification table (Table 2.1) are reachable with current technology. Many of those devices even had features that would be of great benefit for the purpose of this project and would simplify its execution, while at the same time improving its chances of achieving more reliable results. The possibility of achieving sub-meter position accuracies, having high velocity accuracies, or the capability of managing update rates of more than 10 Hz are examples of these features. However, as stated in the results section, the devices that exhibited the highest performance parameters were also the ones that had the highest prices. Unfortunately, the budget limitations on the project ruled out most of these high performance equipment and left fewer devices as available options. There were left, still, some interesting choices, as will be seen in the next sections.

Based on the results of these two tests, the decision was made for selecting and using for the project the Phidgets GPS over any of the smartphones. Despite of the Phidgets GPS being the device that exhibited the worst static position precision and accuracy, the fact that it is capable of measuring more frequently as a consequence of its update rate was considered as a decisive factor. This is because of the fact that the real tests on the project would be performed at higher speeds than the ones reached during the tests (car vs. bicycle). Hence, at higher speeds, having a higher update rate plays a very important role since there could be a big distance between two consecutive measurements if lower update rates are used, and this would have a great impact on the target performance of the system. Additionally, the performance of the Phidgets GPS on the velocity accuracy test was seen to be as good as for any of the smartphones.

4.1.2 Communication method selection

The comparison of the different communication technologies shows that cellular communication offers the best advantages in terms of the wishes and demands of the project: it has possibly the biggest coverage, and it has an already implemented infrastructure. The latter feature comes to be of great importance due to two important implications: no money investment and no time are needed for implementing the infrastructure, whereas for the Wi-Fi and the 5.9 GHz DSRC technologies this implementation would need to be carried out. Another additional advantage of using cellular internet communication is that it allows using the Chalmers Data Hub Service introduced in section II.2.2, which has the extra feature of storing the data for future reference. These were thus the major decision criteria for choosing cellular communication over the other technologies.

4.1.3 Main controller device selection

As stated in the Method section, the selection of the master device was highly dependent on the devices selected for GPS acquisition and for communication. The third version of the Phidgets single board computer (SBC 3) was selected as the master device. The main reasons for this selection were: same brand as the GPS device (Phidgets), which thus implies simplicity; compact size; capable of handling 3G modems for the communication among the different boards; and relatively easy to program and configure for controlling these two devices. The third version (SBC 3) was chosen over the second version (SBC 2), which was the one used in the GPS tests, due to the fact that its specifications rate it as more efficient than its predecessor.

4.2 Server timing analysis

The results of the server timing analysis show that the most suited combination of functions to be used for the project are the Setsuperbeacon and Subscribe2 functions. Their operation implies a faster approach since the Subscribe2 function stays listening to possible uploaded data once activated, and retrieves immediately and automatically the data once it is uploaded to the server with the Setsuperbeacon function. On the other hand, the other functions require an instruction for uploading, and once the data is uploaded a new instruction for downloading, which makes the process slower. One implication of the operation of the Setsuperbeacon

and Subscribe2 functions is that it requires the use of multi-threading in the coding. This is due to the fact that three different users are sending information to the server at the same time (the two vehicles and the Matlab computer), and so three listening (Subscribe2) instructions need to be working simultaneously to fetch all the data. The use of any of the other two set of functions, namely the Upload and Data or the Set and Get functions would probably avoid this issue, but would be much slower in the performance (even at an unacceptable level), as seen in the timing analysis.
As stated in the results section, the fact that the 3G timing was always slower than the Wi-Fi timing leaves room for improvement in terms of increased performance of the system when faster cellular internet connections become readily available (e.g. 4G connection). Another possible improvement for the timing performance of the system would be to avoid one of the communication paths, specifically the path between the server and the Matlab computer. This could be achieved by integrating these two functions into the server, i.e. making the calculations on Matlab on the same server computer where the data is stored.

4.3 Selection algorithm

Considering that the mean and median profiles are created based on central tendency measures, they probably represent the best of all the set of historic profiles. Nevertheless, they have two main disadvantages that are actually linked with each other. First, they do not take into account the current kinematics of the "real" vehicle when going through the road; and second, they are the result of a calculation based on the historic profiles only, which means that the predicted reference profile will always be the same, irrespective of the actual profile the "real" vehicle is developing when going through the road. This brings the following complication: the more the behavior of the "real" vehicle differs from that of the predicted profile, the less correct the TTI prediction would become. And this is something that could occur in the real tests, so it needs to be addressed in some way. As stated in the method section, this was intended to be addressed by creating a reference profile with a criterion that takes into account the kinematics of the vehicle, namely the one referenced here as the speed based approach. However, as seen in the results, this criterion by itself does not seem to be reliable as it does not create a smooth profile curve, but also in many DTI points it does not provide a TTI close to the real one. The moving window and weighting factors approaches together showed to increase considerably the performance of the prediction.
The size of the moving window in the moving window approach simulations was chosen to be of 30 points as this number seemed to demonstrate the best compromise between effectiveness and time taken for the simulation, after some trial and error runs. As for the weighting factors approach, after some trial and error runs, the number of profiles to be weighted that seemed to have the most adequate performance was three, and thus this was the selected number for these simulations.
It is worth noting how the performance of the central tendency based approaches (i.e. mean and median profiles) is remarkably good, especially when the vehicle is getting closer to the intersection. This would raise the question of whether it is worth implementing the more complex speed based approach and its improvements, over the simpler central tendency approaches. However, the speed based approach and its improvements have the advantage of being versatile, i.e. it is theoretically capable of adapting the prediction based on the kinematics of the vehicle. This would play an important role in the case where there are several and highly varied historic profiles available, and the input profile would be that of an "extreme" profile type, e.g. an aggressive, fast driving driver or a slow hesitant one. Other alternative approaches can be considered for the profile selection and the prediction of the TTI algorithm. For instance, an approach that takes into consideration further kinematic parameters of the vehicle such as acceleration. This might increase the complexity of the algorithm, but may allow making a better assessment of the way vehicles were historically driven through the intersection and a better comparison with the kinematics of the currently driven vehicle. Another alternative includes making mathematical models of the historical profiles and extracting parameters that characterize them in terms of the kinematics of the vehicle and the geometry of the intersection. This approach would be probably easier to implement if the historic profiles have all an acceptably similar behavior, but could become more complex if the profiles turn out to be diverse, which has been observed to be the case before filtering. An example of this approach can be seen in the work carried out by Nobukawa [26].
It is left as a future work the development and experimentation of these and other possible alternative approaches, as well as the creation of appropriate performance indicators for each of them. Moreover, further tests on the road, with the availability of a large enough set of historic profiles, are suggested to evaluate and assess the performance of these criteria and decide which would demonstrate to be the most suitable.

4.4 System global simulation

The simulation of the confederate car dynamic model shows that it was following exactly the requirements which were to accelerate constantly to reach the target velocity and to trace exactly the regulation phase instruction. This model is reasonable for a basic model but it should be improved in order to be closer to real human behavior. One way of investigation would be to implement a reaction time between the instruction and the current confederate car velocity. In addition of this feature, a program should implement a random delay between the time when the instruction is sent and the time when the confederate car received it. An idea of delays distribution can be the one post-processed during the testing (Figure 3.23).
The simulation of the overall algorithm behavior shows that the distribution of the results was around 600ms while aiming for a difference in TTI of 0s. This shows that the algorithm is not perfect since it would have been expected to be closer to zero.
The actual testing shows that the performances of the system were relatively good since the performance indicators are close to zero which means that the confederate car was delayed comparing to the subject car as it was aimed. Regarding the regulation part, it has been noticed that the instructions were too jumpy and it may be interesting to add some filtering in the algorithm to avoid that.

4.5 HMI development

The HMI fulfilled the requirements since it followed the recommendations provided by experts. In term of functionality, this system based on an internet connection may bring random delays which would be incompatible with the predefined prediction time. This means that the instructions are shown on the HMI at an incorrect timing and thus this results in false instructions. I could be possible to analyze the current delays within the HMI code to implement a kind of buffer. The prediction time would be higher than the time from the GPS position update to the instruction, then the HMI buffer would store the instruction and as soon as the right time-stamp is reached, the instruction is displayed.
The field assessment showed that the developed HMI was able to properly provide the desired information. However, it showed to be excessively distractive for a potential use by the driver of the vehicle. Hence, further development could be done on the HMI to optimize for this purpose. Possible improvements may include the design of a compact device that can be strategically positioned for convenience of the driver, as well as an auditive instruction for the velocity the driver should follow. Additionally, a filtering of the velocity instruction data to be displayed to the driver should be implemented in order to tune the frequency of the instruction in accordance to the capacity humans have for reacting and implementing such instructions.

4.6 Testing

4.6.1 Preliminary tests

It has been shown on the GPS traces that the GPS update rate was good enough to get small gaps between positions. This is very important for the purposes of this project, i.e. when higher velocities are reached with vehicles, the gaps should be the smallest.
These preliminary tests were useful and important because they showed that the system was actually working as intended: both devices sending GPS information in real time to the server and a third appliance retrieving and plotting this data, also in real time. Hence, the system was ready to be used in the road pilot tests on the field with the vehicles.

4.6.2 Pilot test on field

The first tests on field showed that the system was working as intended although some tunings still to be done. Moreover, the set of historic profiles was not broad enough to have a proper determination of the subject TTI. This set of data should be extended in the future.

4.6.3 Final test on field

The tunings of the algorithm, e.g prediction timespan and start-up timespan, was needed to get better results.

As soon as this was done, the indicator of performance showed that the actual difference in TTI was close to the dTTI defined by the instructor. These results are good but some things still to be improved in order to get a better estimation of the system performance. The reaction time of the drivers should be taken into account in order to get the instruction at the right time and, thus, let the driver go to the instruction velocity at the expected moment. In addition to that, the velocity instruction should be filtered in order to get less *jumpy* instructions and more natural. Finally, the amount of runs should be increased to get better statistics on the performance estimation, in these results, only 5 proper runs could be exploitable because of the traffic.

5 Conclusion

Accident statistics show that intersections are a major cause of road fatalities because they require an accurate and timing decision making from the driver, therefore this project is very relevant in its effort to develop an innovative wireless cooperative system for controlling the previously defined LTAP/OD situations in the field to analyze driver behavior.

Different algorithms were developed: one for the devices embedded in the vehicle, one for the computational server and one for the human machine interface. All individual algorithms performed correctly under simulation, which proves that the theory is appropriate. The pilot test proved that the devices running their respective algorithms were working all together as a unit. This would not have been possible without a cloud application such as the one used in this project: it is the central node of the communication, and is essential for a correct system behavior. In the evaluation side, the velocity assessment algorithm required an accurate profile selection algorithm. The selection of a reference profile and the consequent prediction of the TTI is a key factor in the performance of the overall system. Even though the central tendency approaches for this algorithm performed better when evaluated under the defined performance indicator, the selection approach chosen (weighting factors) would be more efficient for larger set of historic profiles and would have the ability to adapt the prediction to different driving styles. The human machine interface developed using web languages can be rendered on any devices which are capable of web-browsing, which brought flexibility to the system.

All devices and technologies used are affordable and accessible by anyone and therefore the system can be easily reproduced. Despite the low accuracy of the GPS, and the high latencies induced by the communication infrastructure, the performance of the system was able to fulfill the requirement regarding the encroachment time parameter, chosen to be dTTI in this study.

The final testing shows that the theory developed in this study is useable to control properly a confederate vehicle in a LTAP/OD situation. The system can therefore create the desired scenario in order to be able to analyze driver behavior in this specific intersection situation. By its cooperative and predictive features, the system developed in this project could even be used as a base for an active safety system such as an intersection decision support system. Furthermore, the system could also be used as a tool to test or validate such systems.

References

[1] ERSO, *Traffic safety basic facts 2008 - junctions*, English. [Online]. Available: `http://erso.swov.nl/safetynet/fixed/WP1/2008/BFS2008_SN-NTUA-1-3-Junctions.pdf` (visited on 04/23/2013).

[2] NHTSA, “Fatality analysis reporting system (FARS)”, NHTSA, USA, Tech. Rep., 2010.

[3] ——, “Fatality analysis reporting system (FARS)”, NHTSA, USA, Tech. Rep., 2011.

[4] C.-Y. Chan, California intersection decision support: a systems approach to achieve nationally interoperable solutions 2005, 2005. [Online]. Available: `http://www.escholarship.org/uc/item/9sr3x54h`.

[5] D. F. Preusser, A. F. Williams, S. A. Ferguson, R. G. Ulmer, and H. B. Weinstein, Fatal crash risk for older drivers at intersections, *Accident Analysis & Prevention* vol. **30**, no. 2 Mar. 1998, 151–159, Mar. 1998, ISSN: 0001-4575. DOI: `10.1016/S0001-4575(97)00090-0`. [Online]. Available: `http://www.sciencedirect.com/science/article/pii/S0001457597000900` (visited on 04/24/2013).

[6] M. Donath, C. Shankwitz, N. Ward, and J. I. Creaser, “Intersection decision support: an overview”, English, Department of transportation, Minnesota, Tech. Rep. Mn/DOT 2007-33, Sep. 2007. [Online]. Available: `http://www.path.berkeley.edu/PATH/Research/Featured/032703/IDSWebFullReport.pdf` (visited on 05/29/2013).

[7] P. J. Cooper and Y. Zheng, Turning gap acceptance decision-making: the impact of driver distraction, *Journal of Safety Research* vol. **33**, no. 3 Oct. 2002, 321–335, Oct. 2002, ISSN: 0022-4375. DOI: `10.1016/S0022-4375(02)00029-4`. [Online]. Available: `http://www.sciencedirect.com/science/article/pii/S0022437502000294`.

[8] J. C. Laberge, J. I. Creaser, M. E. Rakauskas, and N. J. Ward, Design of an intersection decision support (IDS) interface to reduce crashes at rural stop-controlled intersections, *Transportation Research Part C: Emerging Technologies* vol. **14**, no. 1 Feb. 2006, 39–56, Feb. 2006, ISSN: 0968-090X. DOI: `10.1016/j.trc.2006.03.001`. [Online]. Available: `http://www.sciencedirect.com/science/article/pii/S0968090X06000192`.

[9] A. G. Hawkes, Gap-acceptance in road traffic, *Journal of Applied Probability* vol. **5**, no. 1 Apr. 1968, 84–92, Apr. 1968, ArticleType: research-article / Full publication date: Apr., 1968 / Copyright © 1968 Applied Probability Trust, ISSN: 0021-9002. DOI: 10.2307/3212079. [Online]. Available: http://www.jstor.org/stable/3212079 (visited on 02/13/2013).

[10] D. R. Ragland, S. Arroyo, S. E. Shladover, J. A. Misener, and C.-Y. Chan, Gap acceptance for vehicles turning left across on-coming traffic: implications for intersection decision support design Apr. 2006, Apr. 2006. [Online]. Available: http://escholarship.org/uc/item/8455h5gq# (visited on 04/24/2013).

[11] C.-Y. Chan, Characterization of driving behaviors based on field observation of intersection left-turn across-path scenarios, *IEEE Transactions on Intelligent Transportation Systems* vol. **7**, no. 3 2006, 322–331, 2006, ISSN: 1524-9050. DOI: 10.1109/TITS.2006.880638.

[12] B. L. Allen, B. T. Shin, and P. J. Cooper, ANALYSIS OF TRAFFIC CONFLICTS AND COLLISIONS, *Transportation Research Record* no. 667 1978, 1978, ISSN: 0361-1981. [Online]. Available: http://trid.trb.org/view.aspx?id=85806 (visited on 06/06/2013).

[13] A. Laureshyn, *Assessment of traffic safety and efficiency with the help of automated video analysis*, English, Minsk, Belarus, Oct. 2006. [Online]. Available: www.ictct.org/dlObject.php?document_nr=439&/05_Laureshyn.pdf (visited on 06/06/2013).

[14] D. Sun, H. Luo, L. Fu, W. Liu, X. Liao, and M. Zhao, Predicting bus arrival time on the basis of global positioning system data, *Transportation Research Record: Journal of the Transportation Research Board* vol. **2034**, no. -1 Dec. 2007, 62–72, Dec. 2007. DOI: 10.3141/2034-08. [Online]. Available: http://dx.doi.org/10.3141/2034-08 (visited on 02/13/2013).

[15] W.-H. Lin and J. Zeng, Experimental study of real-time bus arrival time prediction with GPS data, *Transportation Research Record: Journal of the Transportation Research Board* vol. **1666**, no. -1 Jan. 1999, 101–109, Jan. 1999. DOI: 10.3141/1666-12. [Online]. Available: http://dx.doi.org/10.3141/1666-12 (visited on 04/24/2013).

[16] T. Zhu, J. Dong, J. Huang, S. Pang, and B. Du, "The bus arrival time service based on dynamic traffic information", *2012 6th International Conference on Application of Information and Communication Technologies (AICT)*, 2012, pp. 1–6. DOI: 10.1109/ICAICT.2012.6398482.

[17] C. A. Ogaja, *Applied GPS for Engineers and Project Managers*. American Society of Civil Engineers (ASCE), 2011.

[18] B. Hofmann-Wellenhof, H. Lichtenegger, and J. Collins, *Global Positioning System: Theory and Practice*, 5th. Springer, 2001.

[19] C. Rizos, *Trends in gps technology & applications*, 2003.

[20] T. Williams, P. Alves, G. Lachapelle, and C. Basnayake, Evaluation of gps-based methods of relative positioning for automotive safety applications, *Transportation Research Part C* vol. **23** 2012, 11, 2012.

[21] B. Gallagher, H. Akatsuka, and H. Suzuki, Wireless communications for vehicle safety: radio link performance & wireless connectivity methods, *SAE Technical Paper Series* no. 2006-21-0030 2006, 15, 2006.

[22] M. Jerbi, P. Marlier, and S. Senouci, "Experimental assessment of V2V and I2V communications", *IEEE Internatonal Conference on Mobile Adhoc and Sensor Systems, 2007. MASS 2007*, 2007, pp. 1–6. DOI: 10.1109/MOBHOC.2007.4428739.

[23] C.-M. Chou, C.-Y. Li, W.-M. Chien, and K.-c. Lan, "A feasibility study on vehicle-to-infrastructure communication: WiFi vs. WiMAX", *Tenth International Conference on Mobile Data Management: Systems, Services and Middleware, 2009. MDM '09*, 2009, pp. 397–398. DOI: 10.1109/MDM.2009.127.

[24] W. Wiesbeck and L. Reichardt, "C2X communications overview", *2010 URSI International Symposium on Electromagnetic Theory (EMTS)*, 2010, pp. 868–871. DOI: 10.1109/URSI-EMTS.2010.5637329.

[25] Y. J. Li, "An overview of the DSRC/WAVE technology", *Quality, Reliability, Security and Robustness in Heterogeneous Networks*, ser. Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering 74, X. Zhang and D. Qiao, Eds., Springer Berlin Heidelberg, Jan. 2012, pp. 544–558, ISBN: 978-3-642-29221-7, 978-3-642-29222-4. [Online]. Available: http://link.springer.com/chapter/10.1007/978-3-642-29222-4_38 (visited on 06/13/2013).

[26] K. Nobukawa, "A model based approach to the analysis of intersection conflicts and collision avoidance systems", PhD thesis, 2011.

6 Appendices

6.1 Division of work

Christian-Nils:

- Literature review
- Conception of the use cases and algorithm flowcharts including system work flow diagram
- Velocity regulation algorithm design & conception
- Configuration of the devices:
  - Configuration of the boards
  - Design and manufacturing of the Phidgets boards' packaging
  - Configuration of the laptop used as HMI
- Coding:
  - Phidgets C++ programming
  - Server-side MatLab and C++ (MEX) function programming
  - HMI development (php, html, javascript)...
- Hardware selection and design of evaluation tools
- Test and simulation:
  - Simulation program coding to analyze the velocity regulation algorithm
  - Web interface to see the current positions of the Phidgets on a Google map for the preliminary test
  - Participation of the pilot test as instructor in the confederate car...
- Report writing

Juan Camilo:

- Literature review
- Hardware selection:
  - Tests
  - Benchmarking of technologies...
- Server timing analysis:
  - Tests
  - Analysis...
- Historic profiles analysis and selection:
  - Extraction of profiles
  - Filtering of profiles
  - Development of the selection algorithm...
- Testing:
  - Walking tests
  - Preliminary tests on the field...
- Guide for the drivers during the acquisition of the profiles
- Report writing