

University of Trento
University of Brescia
University of Bergamo
University of Padova
University of Trieste
University of Udine
University IUAV of Venezia

Eng. Marco Galvani (Ph.D. Candidate)

OPTIMAL-CONTROL-BASED ADAS FOR
DRIVER WARNING AND AUTONOMOUS INTERVENTION
USING MANOEUVRE JERKS FOR RISK ASSESSMENT

Prof. Eng. Francesco Biral (Tutor)

2013

UNIVERSITY OF TRENTO

Optimal-Control-Based ADAS for Driver Warning and Autonomous Intervention
using Manoeuvre Jerks for Risk Assessment

Ph. D. Head's Prof. Davide Bigoni

Final Examination 09 / 12 / 2013

Board of Examiners

Prof. Sergio Pellegrino (California Institute of Technology)

Prof. Sergio Savaresi (Politecnico di Milano)

Prof. Pasquale Russo Spena (Libera Università di Bolzano)

SUMMARY - SOMMARIO

In this research work, two ADAS have been proposed, both based on optimal control and manoeuvre jerks as parameters for threat assessment.

The first is named “Codriver”, and is a system for driver warning. The second is a sort of completion of the first, since it is designed for autonomous vehicle intervention if the driver does not react to the warnings. The Codriver has been developed by the Mechatronics Group of the University of Trento, which the author is part of, in the framework of the European Project “interactIVe”, to warn the driver for all-around threats safety. It has been then implemented on a real vehicle of Centro Ricerche Fiat, which has been widely tested at the end of the project. On the other hand, for the second system only the main components have been developed by the author during a research period at the University of Tokyo, Japan, and its application is restricted to autonomous obstacle avoidance. In particular, a motion planning algorithm has been used together with a control loop designed to execute the planned trajectories.

Both systems exploit Optimal Control (OC) for motion planning: the Codriver uses OC to plan real-time manoeuvres with humanlike criteria, so that they can be compared to what the driver is doing in order to infer his/her intentions, and warn him if these are not safe; the second system uses OC instead to plan emergency manoeuvres, i.e. neglecting driver actuation limitations and pushing the vehicle towards its physical limits.

The initial longitudinal and lateral jerks of the planned manoeuvres are used by both the systems as parameters for risk assessment. Manoeuvre jerks are proportional to pedal and steering wheel velocities, and their initial values thus describe the entity of the correction needed by the driver to achieve a given goal. Since human drivers plan and act with minimum jerk criteria, and are jerk-limited, more and more severe manoeuvres at a given point are not reachable anymore by a human driver, since they require too high initial jerks: initial jerks can be thus considered proportional to the risk level of current situation. For this reason, when the manoeuvres to handle current scenario require jerks beyond a given threshold, the Codriver outputs a warning. This threshold must be lower than driver limits, so that he/she will be able to react to the warning and still have the chance to perform a safe manoeuvre. When the required jerks exceed drivers’ actuation limits, the risk level raises to an upper step, where driver warning would be not effective and autonomous vehicle intervention should be enabled.

In obstacle avoidance scenarios, it was demonstrated during driving simulator tests that manoeuvre jerks are more robust parameters for risk assessment than for example time headways, since they are less affected by driver’s age and gender.

In questo lavoro di ricerca vengono proposti due sistemi di assistenza al guidatore (ADAS), entrambi basati su controllo ottimo, che utilizzano i jerk di manovra iniziali come parametri per la valutazione dello stato di rischio. Il primo sistema, denominato Codriver, è un sistema di avviso al guidatore, mentre il secondo, che potrebbe completarlo, ha invece lo scopo di intraprendere manovre autonome di evasione ostacolo qualora il guidatore non reagisse agli avvisi. Il Codriver è stato sviluppato dal Gruppo di Meccatronica dell'Università di Trento, di cui l'autore fa parte, per avvisare il guidatore riguardo tutti i possibili stradali rischi in cui potrebbe incorrere. È stato quindi implementato su un veicolo dimostrativo di CRF ed è stato infine testato all'interno del progetto. Il secondo ADAS è stato invece sviluppato dall'autore solo nelle sue componenti principali, durante un periodo di ricerca presso l'Università di Tokyo, in Giappone, e tratta solamente l'evasione di ostacoli mediante manovre di sterzata. In particolare, è stato utilizzato un algoritmo di pianificazione di traiettoria, ed è stato sviluppato un sistema di controllo automatico per ottenere la manovra desiderata.

Entrambi i sistemi utilizzano il Controllo Ottimo per la pianificazione di manovre: il Codriver per pianificare manovre con gli stessi criteri di un guidatore, in modo da poterle poi confrontare con quelle che il guidatore sta effettivamente utilizzando per stimarne le intenzioni; il secondo sistema utilizza invece il controllo ottimo per pianificare manovre di emergenza, trascurando i limiti di attuazione del guidatore e spingendo il veicolo al limite delle sue prestazioni.

I jerk iniziali (longitudinale e laterale) delle manovre pianificate vengono poi utilizzati da entrambi i sistemi come parametri per la valutazione dello stato di rischio. I jerk di manovra sono infatti proporzionali alle velocità di attuazione dei pedali e del volante, e descrivono quindi l'entità della correzione che il guidatore deve effettuare per raggiungere il suo obiettivo. Poiché l'uomo agisce con criteri di jerk minimo, ed è limitato nella velocità di azione, manovre sempre più severe oltre una certa soglia non sono più attuabili da un guidatore, poiché richiederebbero jerk troppo elevati: il jerk può quindi essere considerato proporzionale allo stato di rischio.

Per questo motivo, quando le manovre sicure per un certo scenario iniziano a richiedere jerk oltre una certa soglia, il Codriver avvisa il guidatore del pericolo. La soglia deve ovviamente essere al di sotto dei limiti del guidatore, affinché questo abbia il tempo di reagire e poter effettuare comunque una manovra sicura. Se però i jerk superano quelli limite, il livello di rischio cresce ancora, ed il sistema di intervento autonomo deve prendere il controllo del veicolo.

Per scenari di evasione ostacolo è stato dimostrato con prove al simulatore di guida che i jerk sono parametri più robusti ad esempio dei time headway come parametri per la valutazione dello stato di rischio, essendo meno influenzati dall'età e dal sesso del guidatore.

This work is dedicated to my beloved country
and to its citizens, who have funded these years of research.

*Education is an admirable thing.
But it is well to remember from time to time that
nothing that is worth knowing can be taught.*

O. Wilde

CONTENTS

| | | |
|---------|---|----|
| 1 | Introduction | 1 |
| 1.1 | Background | 1 |
| 1.2 | Research description | 2 |
| 2 | ADAS for collision avoidance | 7 |
| 2.1 | Introduction | 7 |
| 2.2 | ADAS overview | 8 |
| 2.2.1 | ADAS Classifications | 8 |
| 2.2.2 | ADAS for active safety | 9 |
| 2.3 | State of the art of collision avoidance systems | 16 |
| 2.3.1 | Architecture | 16 |
| 2.3.2 | Details on main aspects of the Decision module | 19 |
| 2.3.2.1 | Warning or Intervening Systems | 19 |
| 2.3.2.2 | Threat assessment and driver modelling | 20 |
| 2.4 | Contribution of this work | 25 |
| 3 | An ADAS for driver warning: the Codriver in the interactIVe project | 27 |
| 3.1 | Introduction | 28 |
| 3.1.1 | The interactIVe project | 28 |
| 3.1.2 | The continuous support function in CRF demonstrator | 31 |
| 3.2 | The Codriver | 36 |
| 3.2.1 | Concept overview | 36 |
| 3.2.2 | Theoretical cognitive basis for the architecture of the Codriver | 37 |

| | | |
|---------|---|----|
| 3.2.2.1 | Sense-think-act versus hierarchical perception-action | 37 |
| 3.2.2.2 | The empathic link and joint action | 39 |
| 3.2.3 | Architecture of the Codriver | 42 |
| 3.2.3.1 | Overall Architecture | 42 |
| 3.2.3.2 | Forward emulators | 44 |
| 3.2.3.3 | Inverse models | 46 |
| 3.2.3.4 | First ECOM layer: Motor Primitives | 46 |
| 3.2.3.5 | Second ECOM layer: Simple trajectories dealing with obstacles or lanes | 51 |
| 3.2.3.6 | Third Layer: Navigation | 57 |
| 3.2.3.7 | Fourth Layer: Strategy | 60 |
| 3.2.4 | Implementation of the Codriver for continuous support | 62 |
| 3.3 | Results and other features emerged from field operational tests | 64 |
| 3.3.1 | Test description | 64 |
| 3.3.2 | Observed behaviour of the Codriver | 67 |
| 3.3.2.1 | Inference of intentions | 67 |
| 3.3.2.2 | Behaviours from basic principles | 69 |
| 3.3.2.3 | Limitations | 69 |
| 3.3.3 | User response on the Continuous Support function | 70 |
| 4 | From driver warning towards autonomous intervention: a theoretical and experimental analysis of collision avoidance | 73 |
| 4.1 | Introduction | 74 |
| 4.2 | Driver limits: analysis of driver behaviour from driving simulator data | 75 |
| 4.2.1 | The driving simulator | 75 |
| 4.2.2 | Test description | 77 |
| 4.2.2.1 | Braking scenario | 77 |
| 4.2.2.2 | Steering scenario | 79 |
| 4.2.2.3 | Test procedure | 79 |
| 4.2.3 | Results and discussion | 80 |
| 4.2.3.1 | Braking tests | 80 |
| 4.2.3.2 | Steering tests | 85 |

| | | |
|---------|---|-----|
| 4.2.3.3 | Further discussion | 86 |
| 4.3 | Manoeuvres towards vehicle limits: more accurate OCPs and suitable models | 89 |
| 4.3.1 | Optimal control formulation for obstacle avoidance | 89 |
| 4.3.2 | OCP formulation for brake-to-avoid manoeuvres | 89 |
| 4.3.2.1 | Vehicle model | 89 |
| 4.3.2.2 | OCP formulation | 91 |
| 4.3.3 | OCP formulation for steer-to-avoid manoeuvres | 92 |
| 4.3.3.1 | Vehicle model | 92 |
| 4.3.3.2 | OCP formulation | 94 |
| 4.4 | Analysis of OCP manoeuvres with different models and formulations | 97 |
| 4.4.1 | Codriver and more complex OCP manoeuvres for warning | 97 |
| 4.4.2 | Strategies for obstacle avoidance | 101 |
| 4.4.2.1 | Steering versus braking | 101 |
| 4.4.2.2 | Two-wheel steering versus four-wheel steering | 104 |
| 5 | An ADAS for autonomous collision avoidance: tests with FPEV2-Kanon | 111 |
| 5.1 | Introduction | 112 |
| 5.2 | Overview | 113 |
| 5.2.1 | Objectives | 113 |
| 5.2.2 | The FPEV2-Kanon vehicle | 115 |
| 5.2.3 | System Architecture | 118 |
| 5.3 | Manoeuvre generation with OCP | 120 |
| 5.3.1 | General strategies for manoeuvre planning | 120 |
| 5.3.2 | Vehicle Modelling and Identification | 122 |
| 5.3.2.1 | Vehicle Model | 122 |
| 5.3.2.2 | Model Identification | 125 |
| 5.3.3 | Example optimal manoeuvres for Kanon: double lane changes | 129 |
| 5.4 | Threat assessment | 134 |
| 5.5 | Control Algorithms | 135 |
| 5.5.1 | General strategies for trajectory tracking | 135 |
| 5.5.2 | Early algorithms | 136 |
| 5.5.3 | Final algorithm | 140 |

| | | |
|-------|---|-----|
| 5.6 | Tests and results | 144 |
| 5.6.1 | Experiment details | 144 |
| 5.6.2 | Results | 146 |
| 6 | Conclusions | 149 |
| 6.1 | General considerations | 149 |
| 6.2 | A Codriver for driver warning | 150 |
| 6.3 | From driver warning to autonomous intervention | 152 |
| 6.4 | Autonomous collision avoidance | 153 |
| 6.5 | Future developments | 154 |
| A | Appendix - List of Symbols | 155 |
| B | Appendix - Optimal Control | 159 |
| B.1 | Overview | 159 |
| B.2 | Optimal control formulation and solution | 160 |
| B.3 | Tools for OCP developed by the University of Trento | 162 |
| C | Acknowledgements | 165 |

INTRODUCTION

| | | |
|-----|----------------------|---|
| 1.1 | Background | 1 |
| 1.2 | Research description | 2 |

1.1 BACKGROUND

According to the European Community database on Accidents on Roads in Europe (CARE) [17], more than 54000 fatalities occurred for car accidents in 2001 in Europe. It is estimated that for each death there are 4 permanent disabling injuries, 8 serious injuries and 50 minor injuries, with significant human and monetary costs for the European society. The number of fatalities was cut to 30000 in 2011, showing a decreasing trend reported in Figure 1.1.

Research studies such as the 2000 General Estimates System [110] or the 100 car naturalistic study [114] show that 80% of the accidents is caused by driver faults. The percentage raises to 93% of the total considering only rear-end collisions, which constitute the 28% of the total accidents in the US (Figure 1.2) and 32% in Japan [66], while accidents at crossing paths constitute another 25% of the total. It is reasonable to infer that the diffusion of ADAS with better capabilities than those of human drivers, e.g. enhanced sensing, situation analysis and risk assessment capabilities, superior reaction time, precision and absence of drowsiness, results in road safety enhancements and in accident reduction. This is also confirmed by insurance companies statistics [136] and specific field operational tests such as those conducted during the European Project euroFOT [19].

For this reason, the European Union is funding research projects and promoting collaboration among car manufacturers and research institutes, in order to develop a new generation of ADAS for road safety enhancement, addressed to integration of different functions, anticipation of the warnings and increase of intervention capabilities. European projects such as PReVENT [10], HAVE-it [15], DIPLECS [3] or SARTRE [20] are milestones on the way towards the objective reduction to 15000 of the number of fatalities in Europe by 2020.

In particular, the development of ADAS to avoid collisions with other vehicles (rear-end collisions) or pedestrians would tackle more than 50% of annual car accidents. This makes these systems very interesting and effective research topics worldwide. For this purpose it is possible to develop ADAS for driver warning and for active vehicle intervention, which can in its turn exploit different principles such as autonomous avoidance by braking [41] or steering [42].

In particular, autonomous obstacle avoidance by steering is a topic of great interest in research, since no systems are available on market for this purpose, while it has been estimated that in 40% of the occurring rear-end collisions a steering manoeuvre would be possible (Gidas database [5]). In many of those cases, steering or combined braking and steering would be even safer (Adams, 1994 [29]), if not the only possible manoeuvre for collision avoidance, in certain conditions of high relative velocity when braking would not be effective [42].

1.2 RESEARCH DESCRIPTION

The main activities of the research described in this report have been driven by the considerations above. The objective has been the development of safety driver assistance systems for driver warning and autonomous vehicle intervention, mainly addressing collision avoidance, but also other tasks. Namely, two different ADAS have been developed within two different frameworks, but both based on optimal control and jerk-based triggering of system actions:

- The first is a system, named Codriver, for continuous driver support with only warning purposes, which keeps the driver in the control loop but understands his intentions, and provides him with risk information through visual, acoustic and haptic devices. This system was developed within the European Integrating Project “interactIVe” [26], heir of the PReVENT

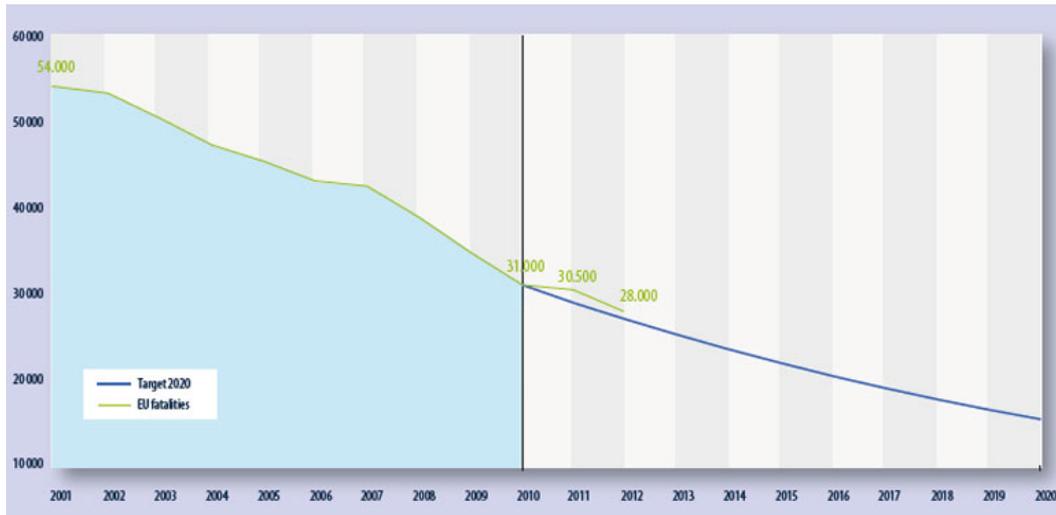


Figure 1.1: Statistics on fatalities for car accidents, from CARE website [17]

project, and implemented on a Lancia Delta of Centro Ricerche Fiat. The system provides real-time holistic support for different driving tasks, such as obstacle handling, lane keeping, curve approaching and landmark compliance. The system proved to be effective during several tests conducted in private proving grounds as well as in public roads, and its warnings showed wide acceptance among the test drivers. In this report, the description will pay particular attention to its usage for collision avoidance warnings.

- The second system is an ADAS for autonomous obstacle avoidance, developed at the Hori-Fujimoto Laboratory of the University of Tokyo. The system exploits the capabilities of the experimental electric vehicle FPEV2-Kanon, equipped with 4 electric in-wheel motors and front and rear steering systems, to autonomously execute manoeuvres for obstacle avoidance such as lane changes. In this case, the system takes over authority from the driver and autonomously tracks a proper reference avoidance manoeuvre thanks to a suitable control loop. Even if the system has not been fully engineered into a unique application, a threat assessment method has been proposed, and the process of manoeuvre planning and autonomous execution has been designed and validated in road tests.

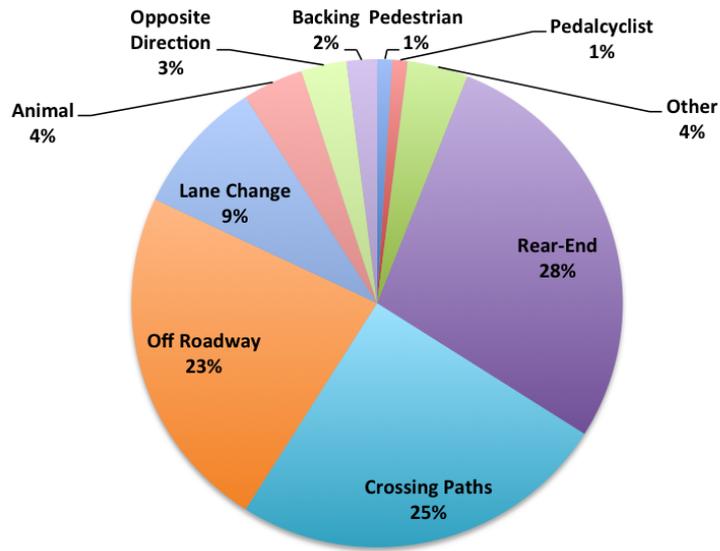


Figure 1.2: Statistics on accident type distribution, from Najm et al., 2003 [110]

Both the systems are based on optimal control, used for understanding driver intentions in the first case, and for generating reference manoeuvres to be executed in the second case. In addition, in both cases the thresholds to release the warnings or to enable the autonomous manoeuvring are based on the initial jerks of the reference manoeuvres. Initial longitudinal and lateral jerks are proportional respectively to pedal rates and to steering wheel velocities, so that they describe the correction required to achieve a certain goal given current situation. In other words, they describe the severity of the desired manoeuvre, and they can be used as parameters for threat assessment.

The two systems, even if developed separately and with different purpose, share similar principles for core functionalities and could be integrated into a unique ADAS, which would take care of all the phases to avoid the occurrence of an accident 1.3. As a matter of fact, the Codriver has been developed for driver information and warning tasks, while the second application could be enabled if the driver does not react, so that if the risk level increases the system can take control of the vehicle and execute autonomous manoeuvres.

This report provides at first a description of state of the art ADAS in Chapter 2, to clarify the contribution of this work to worldwide research.

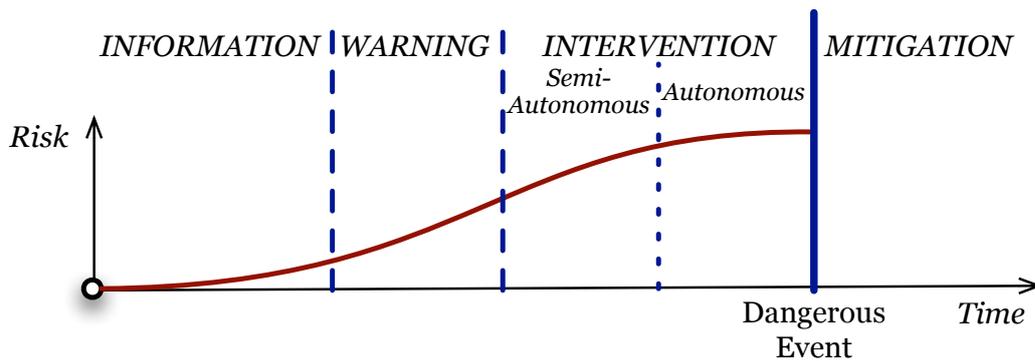


Figure 1.3: Accident handling: Information-Warning-Intervention-Mitigation

In Chapter 3, the architecture of the Codriver is deeply described, depicting also the cognitive framework according to which it was designed. Some details of the technical implementation onboard will be also given, together with a summary of the results obtained on the field operational tests.

The limits of the Continuous Support function in interactIVe raise in Chapter 4, where the possibility of extending the Codriver for handling autonomous obstacle avoidance is considered. A theoretical and experimental analysis of collision avoidance will show what are the possible strategies and what is required for this purpose.

Finally, in Chapter 5 a new application, or an extension of the Codriver, is proposed to meet those requirements, and achieve the autonomous obstacle avoidance when the situation can no longer be handled by the driver. A threat assessment method is proposed, together with a framework for planning optimal avoidance manoeuvres. Finally, a control algorithm to execute them is described and validated, featuring the tracking of yaw rate and sideslip angles and the usage of disturbance observers for the feedback loop.

At the end, Appendix B contains an introduction to optimal control and its implementation, being the basis the developed systems are built on.

ADAS FOR COLLISION AVOIDANCE

| | | |
|-------|---|----|
| 2.1 | Introduction | 7 |
| 2.2 | ADAS overview | 8 |
| 2.2.1 | ADAS Classifications | 8 |
| 2.2.2 | ADAS for active safety | 9 |
| 2.3 | State of the art of collision avoidance systems | 16 |
| 2.3.1 | Architecture | 16 |
| 2.3.2 | Details on main aspects of the Decision module | 19 |
| 2.4 | Contribution of this work | 25 |

2.1 INTRODUCTION

In this chapter, a classification and a description of different ADAS is first reported, following the approach of the European eSafety programme website [13] and the work of Lindgren et al, 2006 [100].

Then, a deeper analysis on ADAS for collision avoidance follows, since this is the focus of this research and the framework where the systems developed in this work are inserted. The summary will involve the systems already available on the market as well as the state of the art of the research.

In the last section, the contribution of this work to the state of the art of the ADAS will be reported.

2.2 ADAS OVERVIEW

2.2.1 *ADAS Classifications*

Nowadays road vehicles are equipped with more and more mechatronic systems to enhance their performances. These devices carry out a large variety of tasks, from improving engine efficiency to properly actuating the brakes or informing the driver on current vehicle state.

Among them, a particular category is that of the ADAS. The purpose of these systems is to help the driver in his duties, which can range from reducing fuel consumption to maintaining the vehicle into safe conditions, from keeping a certain speed with no effort to reaching his destination efficiently, from informing him about road conditions to enhancing his communication needing.

A subcategory among them is that of the safety ADAS, which are in charge of helping the driver in maintaining the vehicle into conditions which must be safe both for him and for the other road users. These systems can be differentiated in many ways, while the most relevant are:

- Active vs. passive safety systems.

Active safety systems are designed to prevent unsafe situations, such as collisions with obstacles, adherence losses, unintended lane departures, etc. On the other hand, passive safety systems aim at mitigating the consequences of such occurrences, when it has not been possible to avoid them. ABS and ESC are typical examples of active safety systems, while seatbelts and airbags are different kinds of passive safety systems.

- Informative/warning vs. intervening systems.

The first category is constituted by systems which keep the driver in the control loop of the vehicle, but inform and warn him about possible dangers and maybe suggest him proper corrections to avoid them. On the contrary, intervening systems take over authority from the driver for the control of the vehicle (Figure 1.3): when the system estimates that the driver will not be able to make a proper correction anymore, they output proper control inputs which override those of the driver. Possible examples of informative/warning systems are those which monitor driver's drowsiness

or vehicle blind spot, while ABS and ESC are once again typical examples of intervening systems.

- Single function vs. holistic systems

Traditional ADAS usually have a single objective, they monitor only a particular aspect of vehicle behaviour, e.g. tyre slip or lane compliance, and they only act to prevent dangers coming from those sources. It also happens that systems equipping the same car are completely separated one from another, possibly manufactured by different suppliers, with their own hardware devices. On the other hand, research is pushing on integration of different systems into a unique one, where different functions share sensors, computation resources and information, or where there is only one function, able to consider all-around threats and to take different actions depending on the cause (i.e. the objective of the Continuous Support function of the interactIVe project [26]).

- Autonomous vs. cooperative systems.

Autonomous systems rely only on onboard sensors to reconstruct present situation, and have no outputs to the surrounding environment. These systems can be further differentiated depending on what they sense: some only use information about ego vehicle, while others need also information about the environment, e.g. road conditions, presence of obstacles, etc. On the other hand, cooperative systems can communicate with other vehicles and/or with the infrastructure, to receive information and enrich their knowledge, or to send information which can be useful for the driver or for the other road users.

2.2.2 *ADAS for active safety*

ADAS cover a large variety of applications and scenarios, and few possible classifications were reported in the pages above. The analysis hereon is restricted to only active safety ADAS: here follows a list of the most common, both for warning and for intervention, i.e the categories where the two systems developed in this work lie, with a short explanation of their operating principles.

- Warning Systems

- Driver drowsiness monitoring

Such systems monitor driver drowsiness analysing his/her actions on the steering wheel and the pedals, or the motion of the vehicle with respect to the lane boundaries (Figure 2.1). Others can also check the movements of his/her head and eye gaze by means of an onboard camera. Usually, an acoustic warning is then issued if they detect that the driver is about to fall asleep.

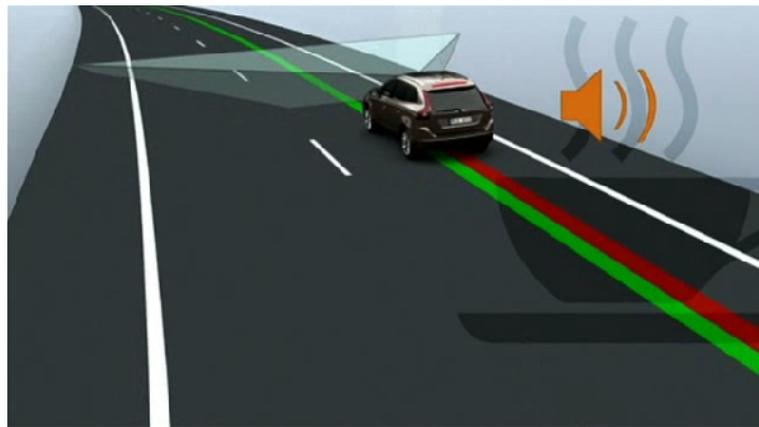


Figure 2.1: Example of driver drowsiness monitoring systems: Volvo's Driver Alert (Volvo Cars ADAS, [23])

- Blind spot information systems (BLIS)

These systems use cameras and/or radars to detect whether vehicles in adjacent lanes are approaching in the mirror blind spot. Usually a warning is issued in a visual device near the rear-view mirrors, so that when the driver looks at them before manoeuvring he can notice the danger (Figure 2.2)

- Curve Warning

Curve management systems are described below, among the intervening systems.

- Lane Departure Warning (LDW)

Lane keeping systems are described below, among the intervening systems.

Blis Blind Spot Information System

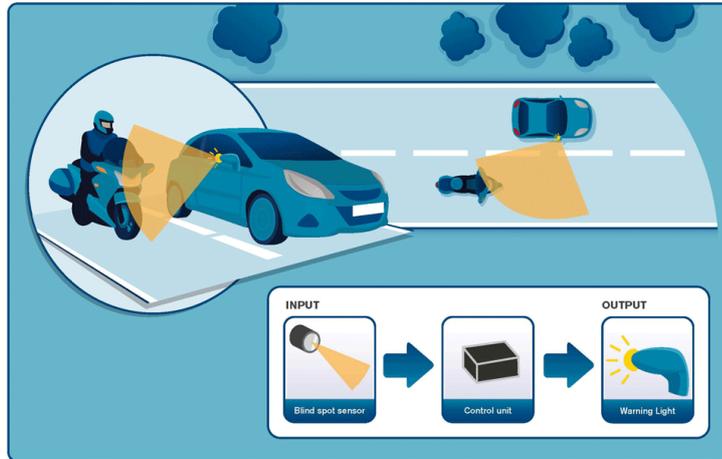


Figure 2.2: Overview on BLIS systems (euroFOT, [19])

- Collision Warning (CW)
Collision avoidance systems are described below, among the intervening systems.

- Intervening Systems

- Anti-lock Braking System (ABS)
ABS (Figure 2.3) can be considered the first active safety intervening ADAS made available on market, and is now compulsory on European cars. It is an intervening system for braking manoeuvres which prevents wheel locking, and consequently reduces stopping distance and avoids the loss of vehicle control. When the driver is deeply pushing the braking pedal, if the wheels are almost locking the system ignores his input and iteratively releases and increases the pressure in the hydraulic braking circuit. In this way, the tyre grip is maintained around its peak value until the braking demand stops, thus maximising the braking performance.
- Electronic Brake assist System (EBS)
EBS is usually integrated with ABS to strengthen the braking action. When the driver deeply pushes the braking pedal, the system interprets

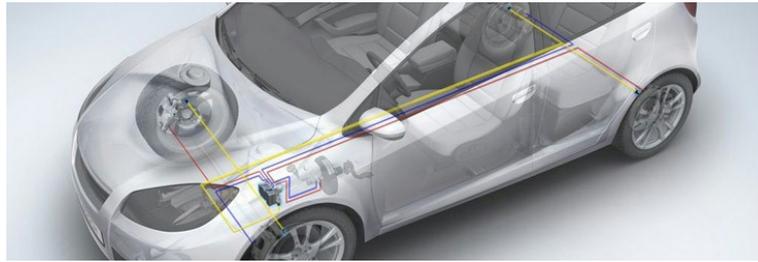


Figure 2.3: Vehicle configuration for ABS (Bosch Automotive Technologies, [2])

it as an emergency manoeuvre and helps him in obtaining the most powerful braking possible, adding braking power beyond the threshold at which the ABS is switched on.

- Electronic Brake force Distribution (EBD)

Vehicle wheels are subject to different loads, which influence their peak braking force and thus the overall vehicle braking performance. EBD systems distribute the braking pressure to each wheel proportionally to its vertical load (Figure 2.4), braking more on wheels with higher vertical load, thus optimising the braking force and preventing excessive tyre slips.

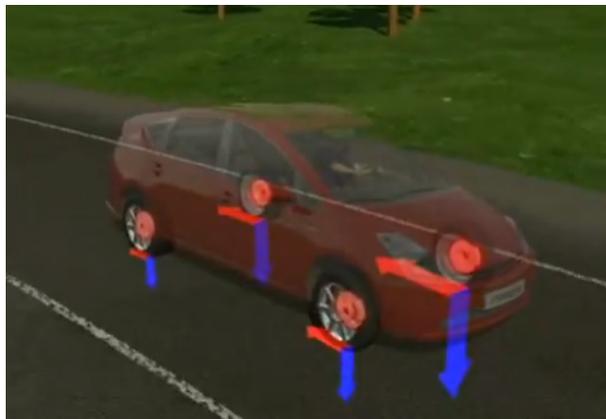


Figure 2.4: Repartition of braking force according to tyre vertical load (Toyota Safety Technology, [7])

- Traction Control Systems (TCS)

Traction control systems are the equivalent of EBD during the accel-

eration phase. When a vehicle wheel starts slipping because of low friction or excessive driving torque, its rotational velocity is individually reduced acting on engine output and eventually also on braking.

- Electronic Stability Control (ESC)

Such systems are used to keep the vehicle in stable conditions, i.e. to avoid excessive tyre slips and losses of adherence. They work combining the action of different elementary ADAS such as ABS, TCS, and EBD. This system regulates the torque output by the engine, and usually controls the braking force of each wheel independently, to generate proper corrective yaw moments (Figure 2.5) in order to reduce vehicle sideslip and excessive yaw. Simpler system only evenly brake to reduce velocity, while more refined systems such as BMW's DSC [1] also use active steering to reduce the sideslip.



Figure 2.5: Yaw moment generated by individual wheel braking to avoid vehicle skidding in Toyota Vehicle Stability Control - VSC (Toyota Safety Technology, [7])

- Cruise Control (CC) and Adaptive Cruise Control (ACC)

The Cruise Control takes over the control of longitudinal dynamics of the vehicle and keeps it at the constant speed set by the driver, when he/she enables it. ACC combines this function, active when the way ahead is clear, with that of keeping a safe distance from an obstacle ahead when present, as shown in Figure 2.6. Also in this case the distance can be manually tuned by the driver.

ACC Adaptive Cruise Control

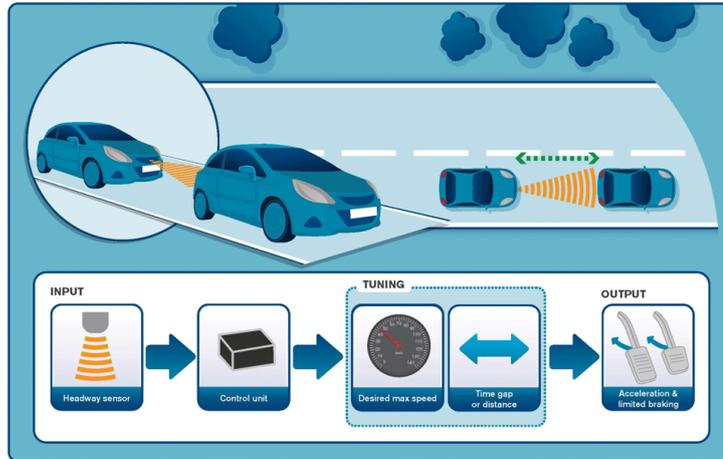


Figure 2.6: Overview on ACC systems (euroFOT, [19])

– Lane keeping systems

Differently from ACC, these systems monitor the vehicle lateral dynamics, and detect if the driver is unintentionally departing from current lane. Then, they can either warn him about the risk, i.e. constituting Lane Departure Warning (LDW, Figure 2.7) systems, or autonomously brake (such as Active Lane Keeping Assist by Mercedes [6]) or steer (as for Lane Keeping Assist by Toyota [7]) to keep the vehicle within lane boundaries. However, in such latter systems the amount of torque applied to the steering wheel is limited, to avoid the driver only rely on the system.

– Curve Management

Curve Management systems prevent the vehicle approaching a dangerous curve at excessive speed. Such systems exploit road maps databases and GPS for vehicle positioning on them, know or evaluate in real-time optimal velocities for the curves ahead, and compare them with current vehicle state to assess the risk level. If the difference is beyond a certain threshold, they can either warn the driver (Curve Speed Warning-CSW systems, Figure 2.8) or autonomously reduce vehicle speed.

LDW Lane Departure Warning / **LA** Lane Assist / **IW** Impairment Warning

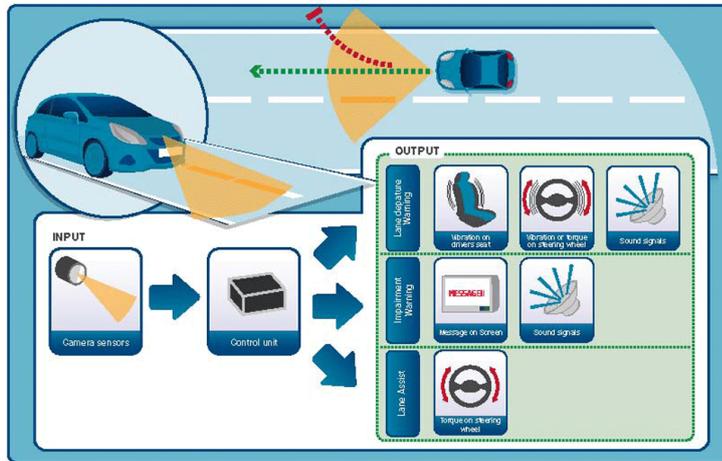


Figure 2.7: Overview on LDW systems (euroFOT, [19])

CSW Curve Speed Warning

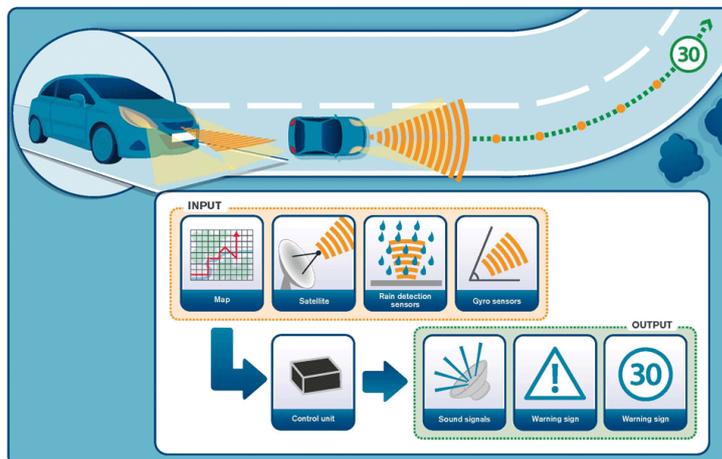


Figure 2.8: Overview on CSW systems (euroFOT, [19])

– Collision avoidance systems

These systems aim at preventing a collision with front and rear obstacles. They monitor the area behind and in front of the vehicle, detect obstacles which could collide with it, and assess the risk level, so that they can then react accordingly. Their reaction can range from simply warning the driver, e.g. Frontal Collision Warning systems (FCW, Figure 2.9) to autonomously taking control of the vehicle and performing an evasive manoeuvre, either by braking and/or steering. These systems will be described in detail in the next section, being the main focus of this work.

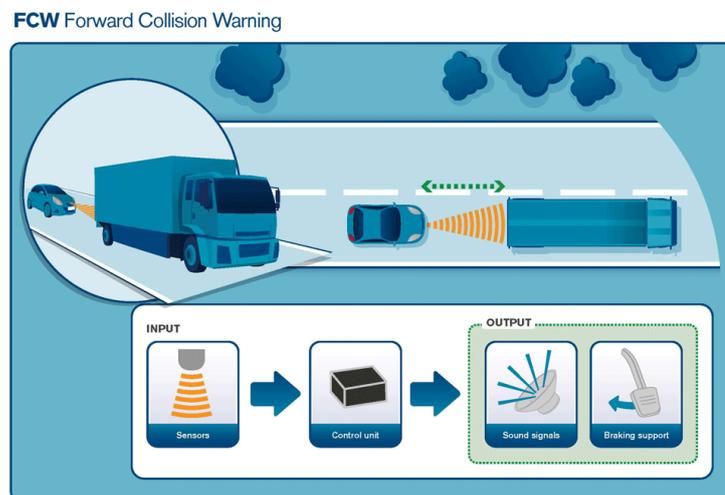


Figure 2.9: Overview on FCW systems (euroFOT, [19])

2.3 STATE OF THE ART OF COLLISION AVOIDANCE SYSTEMS

2.3.1 Architecture

Collision Avoidance systems cover a wide range of safety applications which aim at avoiding collisions with other objects, which can be other vehicles, pedestrians or other obstacles, e.g. animals, rocks or other rubbish on the road ahead. These objects can be all around the ego vehicle, and when the systems assesses that the risk level of a collision is beyond a certain threshold it can both warn the driver

or take control of the vehicle. This control can use only longitudinal dynamics, i.e. braking or in some cases accelerating, only lateral dynamics, i.e. steering, or it can combine the two. The strategy should also take into account traffic conditions, and choose the proper manoeuvre accordingly, e.g. opting for braking if there is an oncoming vehicle in opposite direction, as portrayed in Figure 2.10. To achieve this objective, collision avoidance systems (and ADAS in general) are



Figure 2.10: Braking and Steering strategies for collision avoidance depending on current scenario (images from video logging of interactIVe project [26])

usually built using the well-known architecture based on the three layers of Perception, Decision and Action, as shown in Figure 2.11, even if Perception-Action shortcuts can be used, too (Windridge, 2008 [138]), as it will be explained in the next chapter, while describing the design of the Codriver.

In a general ADAS for collision avoidance, different sensors are in charge of

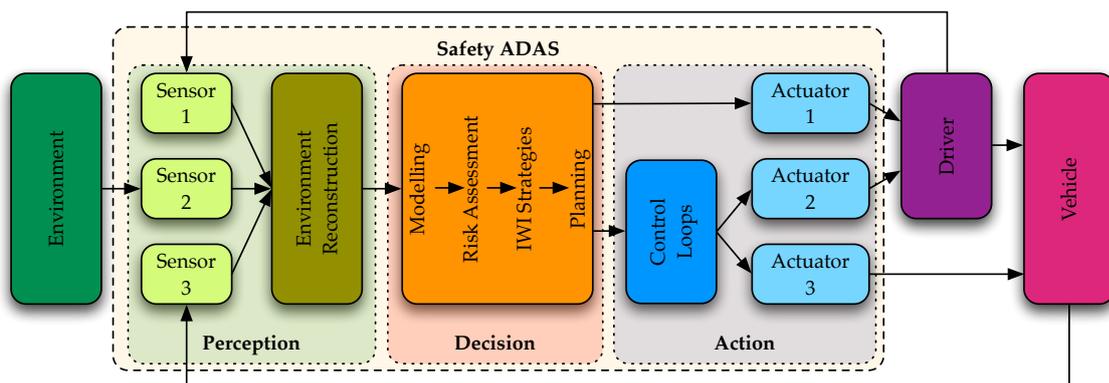


Figure 2.11: Usual three-layered architecture for ADAS

perceiving the environment and the ego vehicle state, e.g. GPS, accelerometers and gyros for ego vehicle, radars, lidars and cameras to detect surrounding obstacles and road local characteristics, and possibly maps for long-horizon road description. Driver monitoring through sensors on pedals and steering wheel, as well as with cameras to detect his eyes gaze, is also important to infer his intentions and tailor system actions. This information is then fused into a unique environment reconstruction, upon which decisions can be made by the downstream module. Since the entire system relies on this data, the development of efficient sensing and estimation technologies is a basic requirement for future ADAS. The perception layer must be as fast and accurate possible, and is needed both for vehicle state estimation and environment detection (Scaramuzza et al. 2009 [123]). This pushes the research in the direction of both better hardware sensors and enhanced software algorithms for state estimation and environment representation (Schmuedderich et al. 2010 [124], Windridge et al. 2008 [138]). It is common in ADAS for safety that the decision module is in charge of modelling the situation and assessing its risk level, usually in a procedural way (i.e. successive situation modelling, risk assessment, choice of an IWI strategy and planning of it), and decide whether it is the case to Inform/Warn the driver or Intervene to directly control the vehicle (i.e. IWI strategy selection), depending on ADAS purpose. On this side, research is pushing the software algorithms to enhance all the described steps, towards cars which are more and more intelligent, able to understand driver intentions and predict his behaviour, also learning from collected data (Wen 2011 [137]) or predict the future evolution of current scenario (Ferguson, Urmson et al. 2008 [68]), for enhanced decision making. Finally, in the ADAS there is an action level, composed of actuators which execute the plan output by the decision layer, if necessary helped by proper control loops. Actuators can be visual, acoustic and/or haptic devices for information and warning, or mechanical, electric and/or pneumatic systems to take the control of the vehicle. For warning strategies, research is conducted to provide more intuitive and efficient feedbacks (e.g. Flemisch et al. 2008 [71], Abbink et al. 2008 [27]), which can quickly gain driver's attention and also suggest him/her proper corrections. On the other hand, for autonomous intervention systems most effort is concentrated on control techniques (Nagai 2007 [109], Hrovat 2012 [87]).

In such a system there are key aspects which differentiate one application from another.

- Purpose: warning or autonomous intervention
- Method for risk assessment: deterministic or probabilistic
- Covered scenarios: urban or extra-urban, fixed or moving obstacles, high or low vehicle speed, etc.

Examples of state of the art systems will be reported and analysed in the following sections, focusing the purpose and the method of the decision module, which is the component of the collision avoidance system which is mainly addressed in this research work.

2.3.2 *Details on main aspects of the Decision module*

2.3.2.1 *Warning or Intervening Systems*

Collision avoidance systems are a large set of safety ADAS, and many of them are already available on market. Some are used only for warning, as for instance those developed by Nissan [22] or Ford [18]. Others combine warning and autonomous braking, such as Volvo Collision Warning with Autobrake (Coelingh et al. 2007 [50]), and City Safety (Distner et al. 2009 [57]), operating between 0 and 30 km/h and able to avoid collisions when relative velocity is below 15 km/h, or Mercedes-Benz's Pre-safe Brake [12], which is instead active at velocities higher than 30 km/h. Similar systems have been developed within the VolksWagen Group (e.g VolksWagen's Front Assist and City Emergency Brake [8]). Toyota's (Pre-crash Safety System [7]) and Honda's (Collision Mitigation Brake System [14]) systems feature autonomous braking as well, but with more mitigation purposes. No systems available on market execute autonomous avoidance manoeuvres by steering.

Warning systems use sensors as radars, lidars and cameras to detect obstacle positions. If they detect an imminent collision, a head-up display usually shows a warning message, supported by an acoustic feedback. Generally, the brakes are also pre-charged in order to obtain a prompter response if the driver presses the pedal. In some cases, the brakes are activated when the driver quickly releases

the accelerator, before he actually reaches the brake pedal, supposing that he is about to do it. In such systems, if the driver does not intervene, the vehicle does not take any initiative, and the collision is not avoided.

On the other hand, it is possible to avoid collision regardless of driver actions with more sophisticated systems, which integrate also autonomous intervention if he/she driver does not correct the manoeuvre. Namely, the system warns the driver when the risk level is beyond a certain threshold, and if the driver does not correct the manoeuvre after a while, they autonomously intervene. Autonomous avoidance can basically include braking strategies, i.e. front braking, rear braking or 4-wheel braking, or steering strategies, i.e. front wheel or 4-wheel steering, (Alleyne 1997 [30]) or a combination of the two. For those systems on market, intervention is limited to autonomous braking, but new systems under development are considering also obstacle avoidance by steering (Brannstrom, 2010 [42], Lidberg et al., 2013 [98]), i.e. the objective of the second ADAS developed in this work.

2.3.2.2 *Threat assessment and driver modelling*

All the three levels of the architecture are of key importance for the overall performance of the ADAS. In the perception layer, accurate and fast sensors are required for environment reconstruction, while from the action point of view, proper IWI strategies are needed for the effectiveness of the system, to give clear warnings when needed, and avoid annoying the driver with too much information or taking over authority when it is not required.

However, in this section the attention is focused on the decision module, and especially on the threat assessment method, i.e. the analysis of current situation to understand whether a collision is imminent or not. Another task of the decision module could be the evaluation of better manoeuvres to be suggested to the driver, which can be either enhancements of current manoeuvre or alternative ones, with different purposes, e.g. steering instead of braking. The threat assessment strategy influences the timing of the warning or intervention of the system, and its reliability. The objective is both avoiding false negatives, i.e. missing the detection of dangerous situations, and false positives, i.e. assessing a safe situation as dangerous, and thus giving useless warnings. Threat assessment

is usually based on the estimation of future evolution of current situation, to which a risk level is then assigned. This requires to infer the evolution both of ego vehicle state, where the identification of driver intentions and its modelling have key importance, and of other vehicles behaviour (these considerations will be repeated in next chapter, while describing the principles of the Codriver). There are two possible approaches for forecasting the evolution of the scenario:

- probabilistic approach
- deterministic approach

In the first case, the evolution of current situation is estimated calculating the probability of a collision, based on the statistical evolution of ego vehicle and obstacle states taking into account different scenarios. On the other hand, the deterministic approach assumes a certain future behaviour for the scenario actors, and calculates its future evolution based on this stated models, without taking into account other possibilities.

Probabilistic approach

The probabilistic approach forecasts future evolutions of the scenario as a probability distribution. Starting from current situation, possible evolutions are generated, both for ego vehicle and for the obstacles, assuming likely behaviour patterns and taking into account feasibility limits. Different manoeuvres are generated for each vehicle involved, and a probability is assigned to each of them. In this way it is possible to evaluate also the probability of ego vehicle - obstacle manoeuvre combinations which lead to collisions, and the system can be tuned to warn and intervene when the probability of collision is beyond given thresholds.

The key features of this approach are the criterium for the generation of possible evolutions, and the assignment of a probability to them and to the complete scenario (these considerations will be further analysed in Chapter 5, while describing possible path planning algorithms). Broadhurst et al. [43] proposed in 2005 a warning system which takes into account two different goals for each obstacle, i.e. straight line following and road following. These determine different control histories which result in different vehicle behaviours, obtained integrating simple equations of motion. For ego vehicle, different goals are investigated, i.e. stop, turn and stop, change lane, turn, overtake, random trajectory. It is important

that the manoeuvres achieving these goals try to mimic human behaviour [62], taking into account in this case the distance to an intended path, deviations from desired velocity, longitudinal accelerations and steering angles. The probability of a collision is obtained using the widespread method of Monte Carlo sampling (also in Althoff et al. 2012 [32] and Thrun 2011 [130]), and warnings can be released accordingly. However, this approach can be very demanding in terms of computational load, due to the several possibilities the system has to take into account.

For this reason, Eidehall and Petersson [59] proposed an enhancement of this system, increasing its calculation efficiency. The authors basically take into account only the safe manoeuvres, neglecting at an early stage the colliding manoeuvres, and they assume that the obstacles will try to avoid the ego vehicle with higher probability than that of ignoring its presence. Furthermore, obstacles in the frontal field of view are weighted more, to resemble that driver's attention will be focused more in that area, and curvilinear coordinates are used to ease the calculations (as in the Codriver equations of motion (3.1)). In this way, higher evaluation velocities are obtained, but still there are no examples of real-time application on a real vehicle.

A probabilistic method for risk assessment has been instead really tested by Sandblom and Brännström [121], but considering only braking manoeuvres for ego vehicle. An interesting feature is the inclusion of a driver model in the algorithm, as it was done in both the applications developed in this work. Instead of estimating his intent, the model directly understands if the driver will consider an intervention as motivated, to avoid false positives in the warnings. The authors introduce the concept of driver safety margins and acceptability of system interventions, and use a decision-making process based on probabilistic driver acceptance and need of intervention. Warnings are then triggered based on required longitudinal accelerations for braking manoeuvres, taking also into account the availability of evasive steering manoeuvres, as expressed by Brännström et al. 2011 [41].

Deterministic approach

The second way for threat assessment is the deterministic approach, which has been also followed for the development of the applications described in this work,

thanks to its higher real-time capabilities. In this case, specific assumptions are made on the future evolution of the scenario, and warning/intervention strategies are decided on this basis. In other words, while the probabilistic way takes into account different possible simple evolutions assigning them a probability value, the deterministic way tries to directly predict the real evolution, possibly based on more comprehensive models. The advantage of this kind of systems is the rapidity of the calculations necessary for the threat assessment: there are not anymore different possible evolutions taken into account, and the future is forecast using specific expressions. As it happens for the probabilistic approach, it is possible to even simplify the problem in atomic subproblems, such as splitting longitudinal and lateral dynamics. On the other hand, it is necessary to simplify the current situation and make assumptions on future evolutions of the scenario, which may result in inaccuracies such as false positives, false negatives and wrong timing for system warning or intervention. In fact, even only considering the prediction of ego-vehicle state, inaccuracies on driver behaviour can regard from high level objectives such as maintaining current route or changing road, to lower level such as overtaking or queueing, as it will be described for the Codriver architecture. To avoid this inconvenience and better predict future evolution, including driver models has a key importance in deterministic threat assessment, and enhances the performance of ADAS anticipating and smoothing their intervention, with higher effectiveness and user acceptance (Sjöberg et al. 2010 [127]). In general, several studies have been indeed conducted for modelling driver behaviour and obtain control laws which resemble human driving, based on collected experimental data. For instance, Shino et al. ([126], [125]) focused for instance on reaction times, time headway policies and time to intersection policies, while Malta et al. [102] analysed driver actions on braking pedal during hazardous situations, using the data from CIAIR Driving Corpus. This field of research has historically interested not only the engineering area, but also psychology and cognitive sciences (further insight is given in section 3.2.2). The general idea is a key concept for the development of new ADAS: using Sjöberg's [127] words, *systems which intervene when it is physically impossible to avoid a collision are less effective than others, which anticipate the intervention to when it is judged impossible for the driver to avoid a collision*. In this family, there are systems which only monitor driver attention to decide if he can correct the current manoeuvre or not. In more

sophisticated cases, the usual controls applied by the drivers are parametrized in deterministic laws, such as steering with constant rate and then constant angle, or braking with constant jerk and then constant acceleration (Brännström et al. 2011 [42]). Even more accurate models take into account also the dynamics of the closed loop system composed of driver and vehicle. Falcone et al. [65] developed an algorithm for safe lateral dynamics, including a driver model assuming his steering behaviour as a weighted sum of two contributions: a pursuing part such as that previously described, and a closed loop corrective part. Their weights could be in general time variant, and they could be also identified online.

Another interesting approach for driver modelling is the usage of minimum jerk theory, i.e. the attitude of humans of acting and planning motion with minimum jerk. An application of this principle for driver modelling, even if used only on simulator and not on a real vehicle, has been developed by Hiraoka et al. in 2005 [81]. This has also found real-time applications in the University of Trento team involved in the development of the Codriver, i.e. by Da Lio, Biral and Bertolazzi in the SASPENCE and INSAFES subprojects within the PReVENT European project ([33],[36]), where minimum jerk was used as the objective of an optimal control problem, thus obtaining smooth and humanlike manoeuvres which could be used as references for warning triggering. This basis has been also used to develop the Codriver, as described in next chapter.

Once the system composed by driver and vehicle has been modelled, it can be used for forecasting system evolution and assessing the risk level. In the collision avoidance system described by Coelingh et al. 2007 [50], warning timing is based on relative velocity and driver reaction times, while autonomous intervention is activated when the system estimates that the accelerations necessary to avoid a collision will be too high to be obtained by a driver. In the work of Falcone et al [65], threat assessment is considered as a constraint satisfaction problem, i.e. assuming that the vehicle will remain within lane boundaries, with small slip angles errors in orientation. Threat assessment is generated via a reachability analysis of safe state sets starting from current situation, and the system intervenes when it estimates that the constraints will be exceeded.

In order to anticipate warnings even more, research was conducted to understand driver intentions, so that the support can start before threats can be foreseen by usual ADAS. The basic approaches are the usage of sensors on pedal and steering

wheel to analyse current driver behaviour and a camera to detect his eye-gaze (resumed by Doshi and Trivedi, 2011 [58]), and the usage of optimal control (Saroldi, Da Lio et al. 2012 [122]), which will be deeply explained in the next chapters.

2.4 CONTRIBUTION OF THIS WORK

In this work, two different ADAS have been developed, one for driver warning and one for vehicle autonomous intervention. Both the systems exploit optimal control, which allows not only to foresee driver actions as advanced deterministic systems, but also to ground them to some goals as some mentioned stochastic threat assessment systems. On the other hand, the execution is very fast as in deterministic systems, as it will be explained in next chapters.

Both systems include into the OCP a driver model to resemble his/her path planning criteria and his/her actuation limits. In this way, it is possible to obtain a non-invasive system, with reduced false positives and increased user acceptance, as suggested in the mentioned literature sources.

Furthermore, manoeuvre jerks have been used as unique parameters for risk assessment, both for longitudinal and lateral dynamics, since they describe the corrections required by the driver to reach his goal, starting from current situation. This has a solid theoretical basis coming from the naturalistic driving studies presented above, and allows reducing significantly the parameters to be tuned to obtain proper timing for system actions.

Finally, the warning system covers a wide set of scenarios and operating ranges, both in urban and extra-urban environments, and merges several safety functions providing holistic support.

The system for autonomous intervention, in his turn, exploits robust planning and novel fast control techniques to obtain the vehicle follow the desired evasive path, as it will be specifically described in next chapters.

AN ADAS FOR DRIVER WARNING: THE CODRIVER IN THE INTERACTIVE PROJECT

| | | |
|-------|---|----|
| 3.1 | Introduction | 28 |
| 3.1.1 | The interactive project | 28 |
| 3.1.2 | The continuous support function in CRF demonstrator | 31 |
| 3.2 | The Codriver | 36 |
| 3.2.1 | Concept overview | 36 |
| 3.2.2 | Theoretical cognitive basis for the architecture of the Codriver | 37 |
| 3.2.3 | Architecture of the Codriver | 42 |
| 3.2.4 | Implementation of the Codriver for continuous support | 62 |
| 3.3 | Results and other features emerged from field operational tests | 64 |
| 3.3.1 | Test description | 64 |
| 3.3.2 | Observed behaviour of the Codriver | 67 |
| 3.3.3 | User response on the Continuous Support function | 70 |

3.1 INTRODUCTION

3.1.1 *The interactIVe project*

This chapter describes the activities carried on during the project interactIVe, describing in detail the architecture of the Continuous Support application, and of the Codriver developed for it by the University of Trento. Finally, qualitative results of the Continuous Support function during the project will be shown. On the other hand, a special focus on the usage of the Codriver for collision avoidance applications will be given in Chapter 4.

However, here follows first a short overview of the overall project and its purposes. “interactIVe” is currently the flagship European project in the area of Intelligent Vehicles [26], and it inherits the results of the PREVENT project [10]. The consortium (Figure 3.1) includes almost all the major european car manufacturers (OEMs), some of their suppliers and several research institutes and universities. The objective of the project is to provide holistic support to the driver according



Figure 3.1: The consortium of the interactIVe project, including several OEMs, suppliers and research institutes [26]

to the needing of the scenario, as explained in Figure 3.2. Namely, different

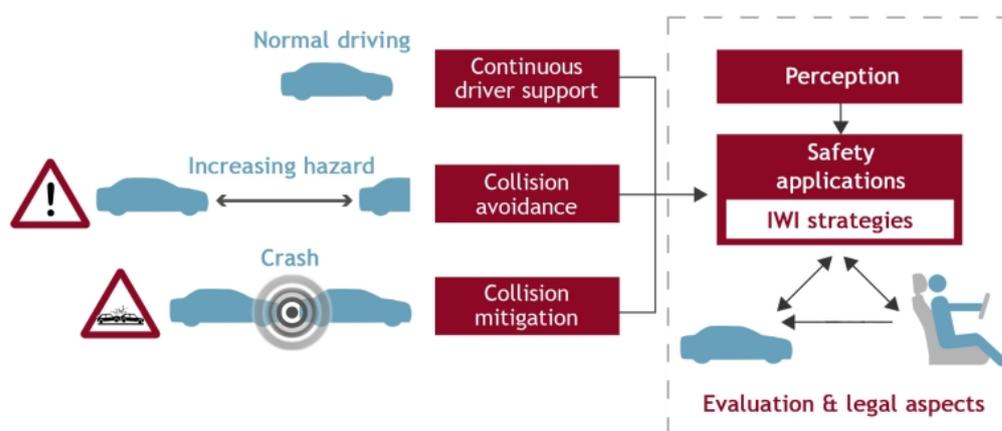


Figure 3.2: Overview of the interactiVe project applications [26]

applications have been built for the different phases of danger occurrence, i.e. for normal driving conditions, to prevent dangerous situations through continuous driver support, for increasing hazard situations, with the usage of autonomous collision avoidance strategies, and for unavoidable collision scenarios, with collision mitigation strategies.

The project was organised in different sub-projects (SP), which are reported in Figure 3.3 together with their leader institutions. SP1 was only intended for top level management. SP2 was in charge of providing a Perception Platform common to each application, while SP3 had to support the development of suitable HMI for each application. SP4, SP5 and SP6 developed instead the complete applications described above: in SECONDS, the continuous support functions have been developed, while INCA dealt with autonomous collision avoidance and EMIC with collision mitigation. A final SP was then in charge of evaluating the performances and the safety impact of each application.

Different applications were developed within each SP and implemented on different demonstrator vehicles, according to the OEMs needs (Figure 3.4).

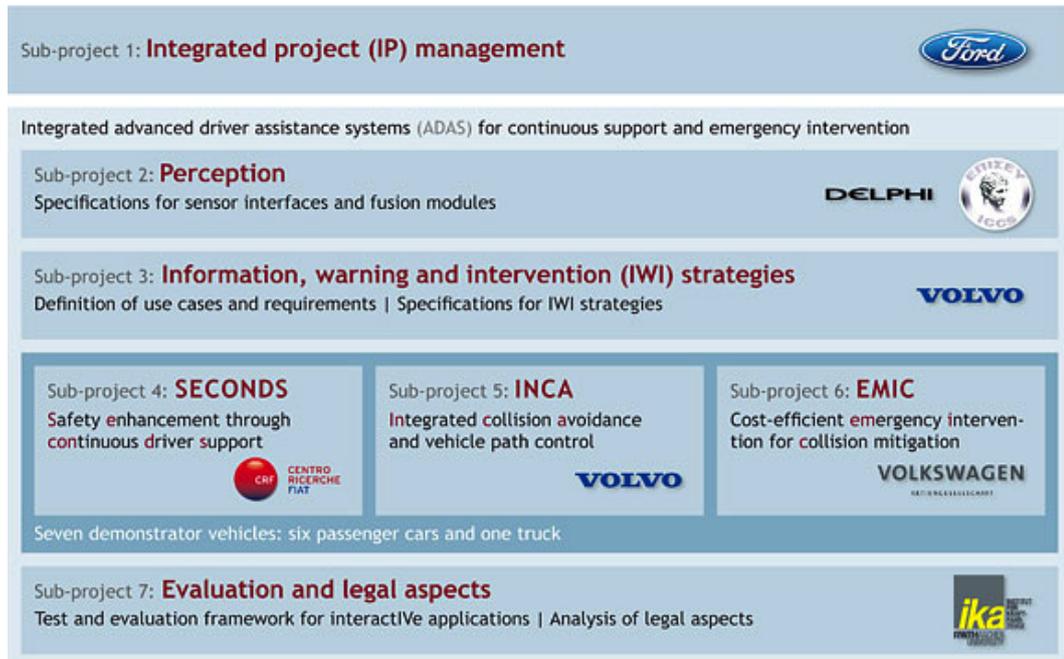


Figure 3.3: The subprojects interactIVe is divided into [26]

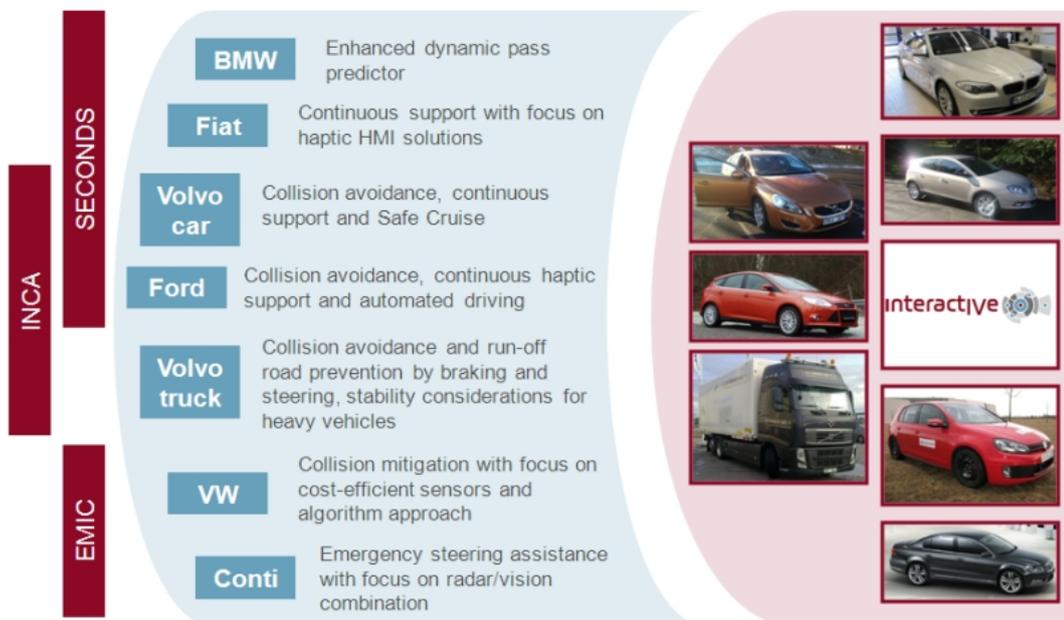


Figure 3.4: The demonstrator vehicles developed in interactIVe [26]

3.1.2 The continuous support function in CRF demonstrator

The activity of the University of Trento was inserted in the SECONDS subproject, to develop an application, named Codriver, to be integrated in the Continuous Support function (Da Lio et al., 2013 [55]) with warning capabilities (Figure 3.5) of Centro Ricerche Fiat (CRF).

The novel feature of the application is that it considers safety in a holistic way,

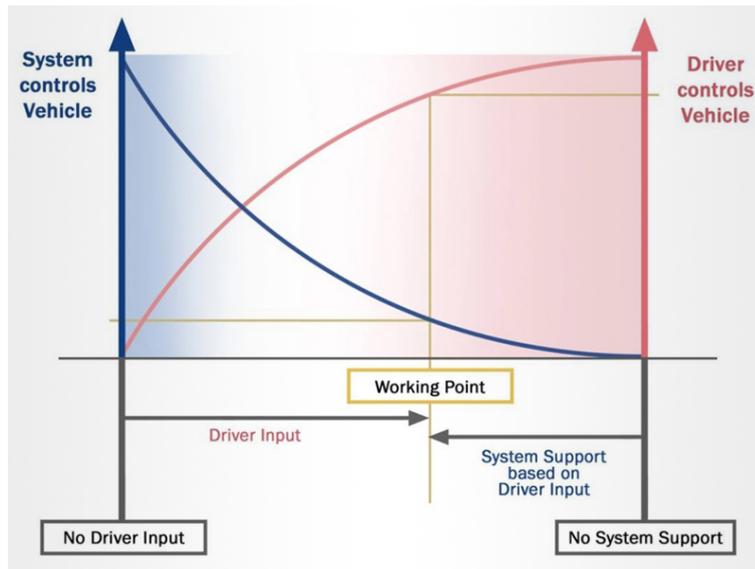


Figure 3.5: Continuous Support application working point, oriented to driver warning with limited system intervention, i.e. haptic feedback [26]

including into an unique application the prevention of all-around threats, such as collisions with obstacles, lane departures, adherence losses, bad speed limit and landmark compliance etc. In other words, the application comprehends in itself several functions which are usually found in different devices, with advantages both in system effectiveness (Kusano and Gabler, 2011, [93]) and in the usage of sensors and computational units, which can be shared.

The architecture of the continuous support function is based on Perception-Action behaviours, organised in hierarchies, as it will be explained in the next section. In this section, the focus will be on the applications which surround the hierarchical Codriver, i.e. the Perception Platform and the HMI for driver warning. The perception level has been developed within SP2, and exploits front and rear radars

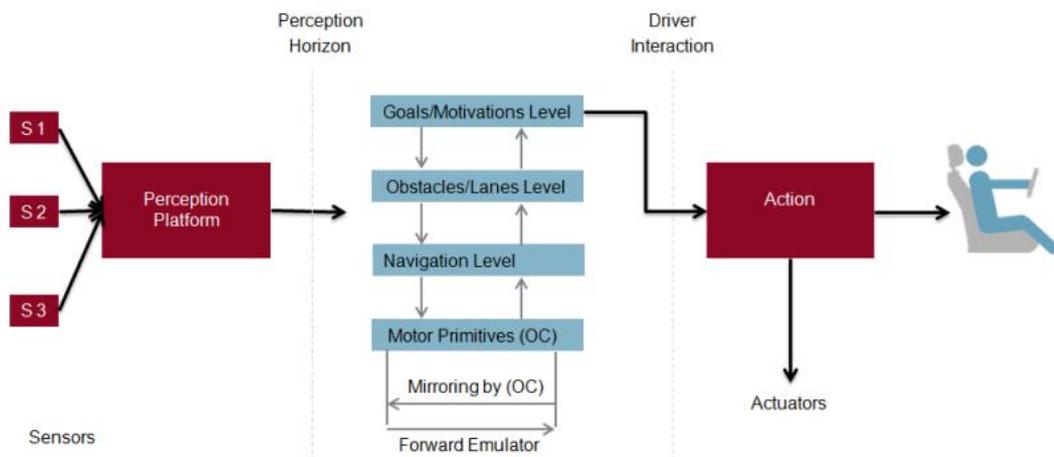


Figure 3.6: Architecture of the continuous support function

and a front lidar for obstacle detection, a camera for lane recognition, vehicle positioning with the respect to the road and obstacle classification, and a GPS combined with a map database for vehicle absolute positioning (Figure 3.7). This information is fused to provide a complete scenario reconstruction, and is output to the decision layer together with information from onboard sensors concerning vehicle state, e.g. vehicle velocity and acceleration, and driver behaviour, e.g. steering wheel position and velocity and pedal positions.

The Codriver is then in charge for the threat assessment and the warning management, and is described in detail in section 3.2. A key feature of the system is that it bases its warning strategies on the estimation of driver intentions, which is a wide topic of research worldwide, with applications which take advantage of driver modelling (MacAdam, 2003 [101], Liebner et al. 2012 [99] and monitoring (McCall and Trivedi, 2006 [105]). A review on this systems can be found in Doshi and Trivedi, 2011 [58]. However, the approach in this work has been to ground his actions to an objective by means of optimal control-based motion planning, as it will be described in detail.

Another basic aspect is the strategy to identify driver intended manoeuvre. Each

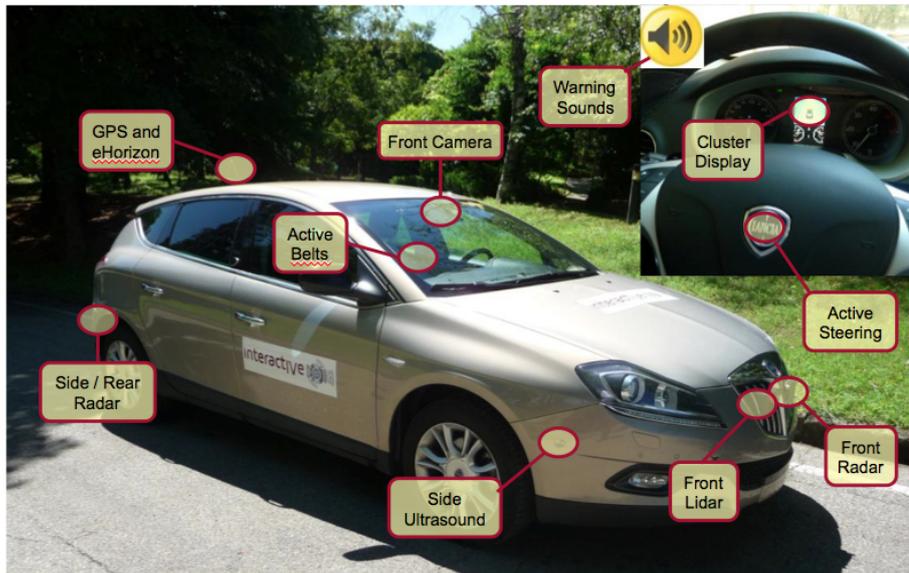


Figure 3.7: Sensors and HMI actuators in the CRF demonstrator [26]

manoeuvre is described and identified by only two parameters, i.e. their initial longitudinal and lateral jerks, which describe the entity of the correction required to obtain a given objective. Comparing driver's jerks to the initial jerks required by several manoeuvres with different purposes, evaluated in real time, it is possible to find the best-matching pair among them, which is identified as its intended manoeuvre.

Downstream to the Codriver, a suitable HMI for driver warning has been developed within SP3 in collaboration with CRF, comprehensive of acoustic warnings, visual descriptions and haptic feedback (vibrating seatbelt in case of collision risk, steering wheel feedback in case of unintended lane departures). In particular, a visual display shows an icon of the ego-vehicle surrounded by an all-around "Safety Shield", divided into 3 sectors, i.e. front, rear left and rear right threats. Depending on the direction the hazard is coming, one or more of them can be painted in yellow or red, depending on risk level (yellow alarm, red alarm). Other details are added to better describe the threat, e.g. a front vehicle appears if the threat is a possible rear-end collision, and adjacent lanes are turned red if unintended departures are preview.

The entire application has been integrated on the CRF demonstrator vehicle, a

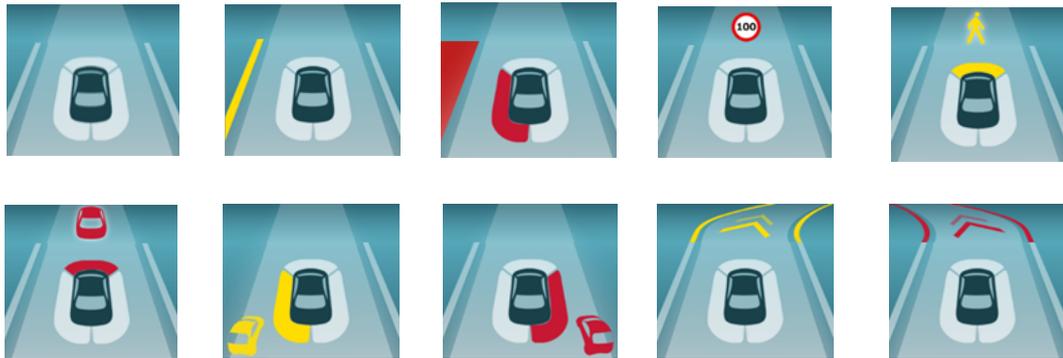


Figure 3.8: The visual display of the continuous support function, with the “Safety Shield” metaphor. Examples of: normal driving, yellow alarm for unintended lane departure, red alarm for drift to side barrier, exceeded speed limit, yellow alarm for vulnerable road users presence, red alarm for rear-end collision, yellow and read alarms for vehicles in the blind spot, and yellow and red alarms for excessive speed in a curve [26]

Lancia Delta with robotic gearbox (Figure 3.8). A separate PC is dedicated to the Perception Platform, managed by Elektrobit ADTF [21] modules, which send environment reconstruction via UDP protocol to the Codriver. This is installed on a Ubuntu Linux PC (Intel Atom N270 with 1.6 GHz, 2 GB DDR2 RAM), embedded in a dSpace MicroAutoboxII [24]. The dSpace then outputs its responses on a CAN bus, connected to the device which manages the HMI. The continuous support function works at 10 Hz. While the codriver is working, it is possible to check its outputs on a wide display onboard, thanks to a Qt application I developed (Figure 3.9). This is installed on the embedded PC, and is also useful for real-time perceived scenario visualisation. In this chapter, the general architecture of the system and its principles will be described in section 3.2, while results from field operational tests will be described in section 3.3, especially focusing obstacle avoidance.

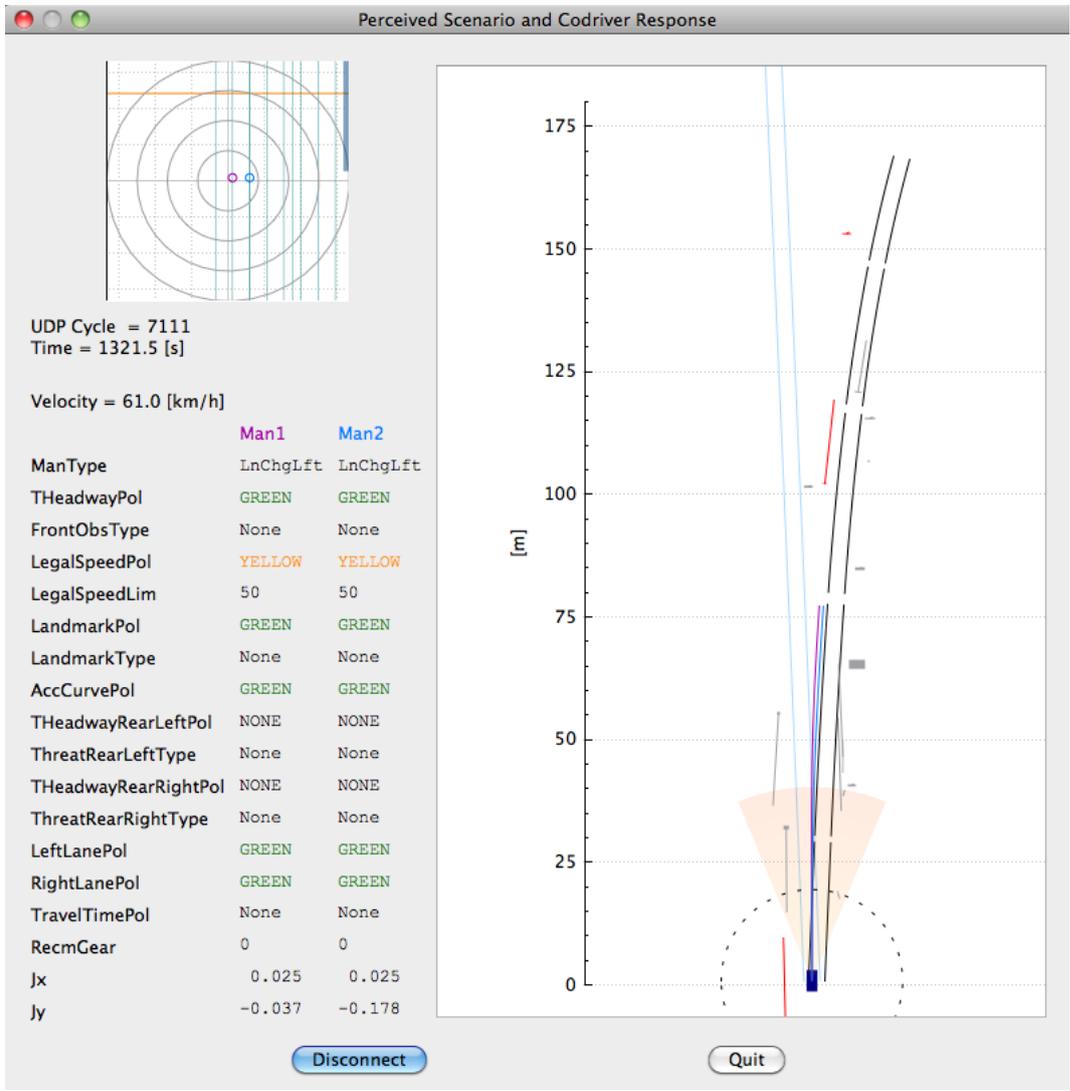


Figure 3.9: The Qt-based tool for real-time scenario visualisation together with the responses of the Codriver

3.2 THE CODRIVER

3.2.1 *Concept overview*

The objective of the project was to build a system which continuously interacts with the driver and supports him in his decisions, to prevent dangerous situations. Our answer to this, i.e. the Codriver, is thus inserted in the framework of smart vehicle-human interaction (Flemisch et al. 2003 [70], Inagaki, 2008 [90], Wen, Li et al. 2011 [137] and 2012 [97], Da Lio et al., 2012 [53]), to build systems which share different levels of authority with the driver in an integrated and continuous way (as in the European Project HAVEit [15] or in Biral et al., 2010 [38]).

The Codriver can be defined as an artificial agent, which is able both to drive like a human, and to infer human intentions interacting accordingly, including the correction of actions he/she has executed by mistake (Da Lio et al. 2014 [54], from which the following sections take wide inspiration).

The guideline for its development has been to imitate “natural” co-drivers: for instance, a driving license tutor has knowledge of human motion patterns and is able to infer the trainee’s intentions, thus acting on the vehicle controls in accordance with the trainee’s needs, given the current scenario.

However, this relationship is not limited to humans: the rider-horse metaphor (or H-metaphor) [70] describes a symbiotic system, in which an animal can *read* human intentions, and, reciprocally, the rider can *read* those of the animal, with the horse trying to maximally reduce the human’s riding burdens consistent with their intentions. This latter metaphor stresses the symbiotic nature of the collaboration between the two, and in particular constitutes an original solution for the sharing and dynamic reallocation of authority.

Having thus outlined an appropriate control metaphor for the co-driver ideal, the problem is now its technological implementation. To implement the reciprocal understanding, the *like-me* theory (Meltzoff, 2007 [107]) has been followed, so that an agent with a planning architecture as similar as possible to that of human drivers was built, to better reproduce and understand the driver’s states of mind, as it will be explained in the following pages.

3.2.2 Theoretical cognitive basis for the architecture of the Codriver

3.2.2.1 Sense-think-act versus hierarchical perception-action

The usual architecture for ADAS is the so called *sense-think-act* procedure (case (a) of Figure 3.10). This architecture corresponds to the traditional view of psychology and artificial intelligence (Pylyshyn, 1984 [119]) which divides the agent function in sequential steps. The objective of perception is in this case to make an internal model of the world, which is thereafter symbolically manipulated and then output.

A contrasting view is the *behavioural* architecture introduced by Brooks, 1986

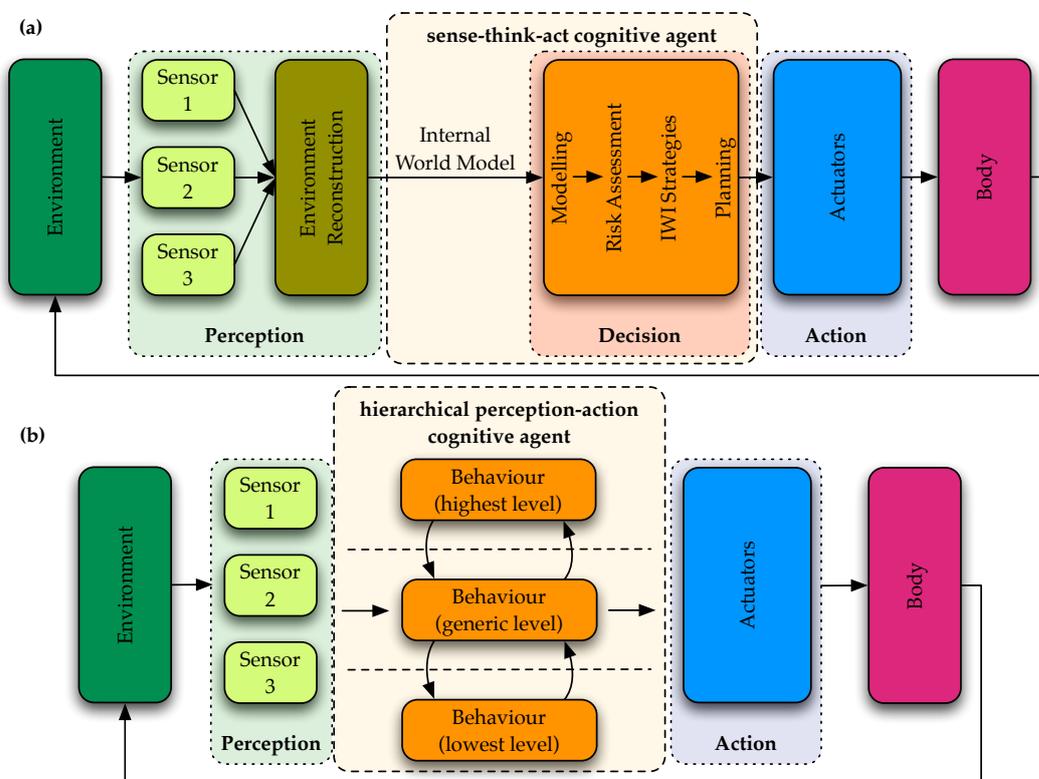


Figure 3.10: Comparison between the traditional sense-think-act architecture (a) and the proposed perception-action behaviours organised in hierarchies (b)

[44]. This architecture starts with simple sensory-motor loops, representing elementary behaviours (e.g. steering, accelerating, etc.) and grows by adding behavioural layers that *subsume* (i.e. set the goals for) the already implemented

lower ones, to achieve newer more complex functions. In contrast to the sense-think-act architecture, the agent is now decomposed by horizontal levels, enabled via control loops without explicit internal models of the world.

The two views have been extensively debated, not only in terms of questioning the idea of cognition as symbol manipulation, but also in terms of the related concept of *internal (symbolic) representation*. Although some researchers supported the idea of *intelligence without any representation* at all (e.g., Brooks, 1991 [45]), other researchers have developed novel concepts of representation, among which *emulators* (Grush, 2004 [77]), which can be used for anticipation of sensory input, predictive and optimal control, deliberation (i.e. going beyond reactive behaviour), imitation, learning, co-operation, etc. A review of these can be found in Hesslow, 2012 [80].

The sense-think-act architecture has hitherto been the natural choice in the past for systems that were fully engineered by design, as is the case of most driver assistance systems (e.g. in the PREVeNT project [36], [33]). However, as the complexity of the driver support function grows, the shortcomings of this type of architecture become evident, e.g. the rigidity and non-scalability of a priori symbol systems, and the lack of flexibility inherent in pre-programmed approaches. Indeed, predefined symbol systems inherent in the computer metaphor impose a rigidity that inhibits evolution and adaptation of the agent. However, even if it was possible to a priori enrich the internal model representation with all the relevant information, the result would be a copy of the external world, with its complexities and without any indication of which is the important information and how it should be used. In this way, the task of extracting the relevant information is only postponed to the symbol manipulation stage. Thus, in case (a) of Figure 3.10, the perception layer has the purpose of producing a symbolic copy of the world. On the other hand, in case (b) perception is intrinsically linked to action, emphasising the notion that cognitive representation is meaningful only as far as it relates behaviourally to the potential for action.

This latter behavioural architecture has also been used in intelligent vehicles, in particular within the European Project DIPLECS [3], which adopted the architecture of case (b) of Figure 3.10, with layers inspired by the Extended Control Model (ECOM), described by Hollnagel, 1999 and 2002 [84], [83]. ECOM is a psychological driver model, which describes the driving as a hierarchy of con-

current subsuming control loops, occurring at different time rates. DIPLECS implemented the three bottom-most ECOM layers: tracking (minute chassis control and disturbance rejection), regulating (producing space-time trajectories) and monitoring (keeping track of the progress towards destination and setting related short term goals, e.g., overtaking). Each layer was built from learned perception-action cycles [67], used both to model human intentions and to carry out these intentions.

In the perception-action (PA) framework, the embodied cognitive agent's environment is represented only in the domain it is able to handle, thus eliminating much of the representational redundancy in standard approaches. In this way, environment representation and action planning are no longer separated.

In hierarchical PA systems a task-subsuming hierarchy exists, consisting of progressively higher-level (more abstract) perceptions and actions loops. The idea is that higher levels of the hierarchy switch sub-tasks on and off in the levels below, as required by the situation, with each subtask having an associated perceptual goal set from the level above, in a slightly different view from Brooks [44]). In driving terms, a hierarchical PA representation of a human driver's intentions considers only those entities relevant to a particular task level, e.g. roads at the very high level of the PA hierarchy, and kerb at the lower level (thus the high level goal "navigate junction" also includes the lower-level sub goal "avoid hitting the kerb").

Following the example of DIPLECS, hierarchical ECOM layers have been chosen as the architecture for the Codriver instead of the traditional sense-think-act procedure, since it granted flexibility and scalability to adapt to complex scenarios and progressively add advanced functionalities. The main reason, however, lays in the need to interact with the driver in an effective way, which can be achieved with a system resembling the driver's states of mind, as it will be explained in next section.

3.2.2.2 *The empathic link and joint action*

In the H-metaphor, rider and horse can "read" each other [70]. However, the question of how to form such empathic links between two agents is a general research topic in human-robot interaction, and an active field of studies in neuro-

sciences, psychology, social sciences.

For the implementation of the Codriver, inspiration has been taken from the Simulation Theory of Cognition. This is a conceptual framework [80] which essentially states that *thinking is simulation of interactions*, carried out as covert motor-sensory activity, e.g. Hesslow, 2002 [79]. Understanding of others' intentions is thus also a simulation process, this time carried out via the *mirroring* of observed motor activities of the others, e.g. the *like-me* framework by Meltzoff, 2007 [107], who summarises the concept as: *others who act like me have internal states like me*, which implies similarity of the two agents sensory-motor strategies. With this respect, also the discovery of the mirror neurone system has to be mentioned [48].

Inference of intentions has also been further studied by Wolpert et al., e.g. [140] and Demiris et al. e.g. [56]: their approach is to generate agent behaviours under a number of alternative hypotheses, which are then tested by comparison with observed behaviours. This means that multiple simulations are run in parallel, and the most salient one(s) are selected. This method of intention prediction is named the *generative approach*, and it has been used for the development of the Codriver, testing different possible driver purposes and matching the actions they require with the actual driver's behaviour.

An alternative could have been the *descriptive approach*, which does not rely on the *like-me* modelling strategy. It rather predicts the probability of the next move based on the classification of motor stereotypes, which are learnt for instance by means of a variety of machine learning algorithms. These approaches thus extrapolate the recent motor pattern of e.g. drivers, often with no link to meaningful targets and states on the road. A recent review is given by Doshi and Trivedi [58]. Hurley, in 2008, combined many ideas into a *Shared Circuit Model* [89], showed in Figure 3.11, which is useful for introducing the various contributions to the development of the Codriver. Overall, Figure 3.11 shows an agent, which interacts with the environment (the feedback dashed loop) in agreement with the embodied cognition approach. The first element of its internal structure is the internal loop that goes backwards from action to perception. i.e. a *forward emulator*, which predicts the effect of agent motor activity. The presence of this element in the Codriver is due to Grush [77], who proposed the idea that human brain, while interacting with the body and the environment, constructs neural

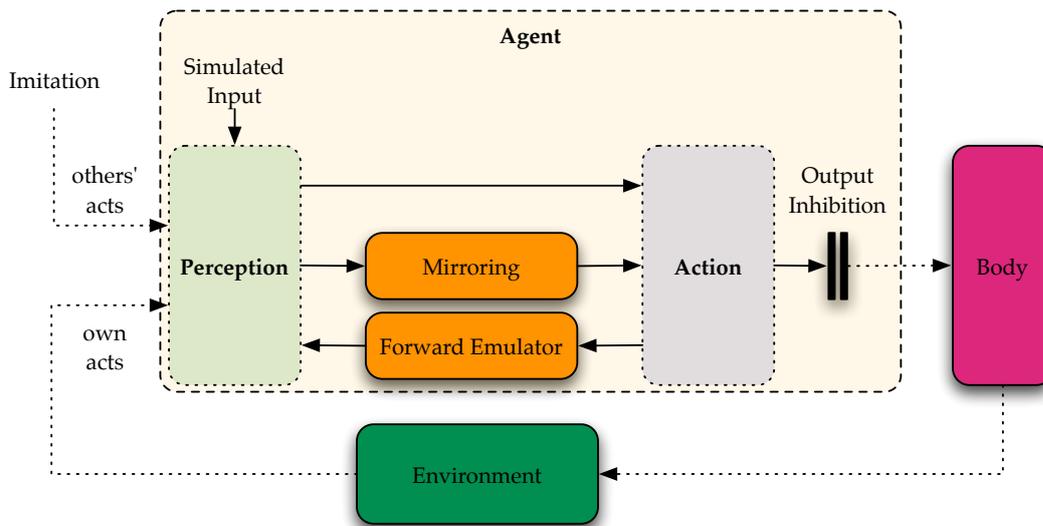


Figure 3.11: Architecture of a cognitive agent featuring emulation and mirroring, capable of imitation, mind reading and deliberation, i.e. Hurley's Shared Circuit Model [89]

circuits that model the body and environment forward dynamics. This function is analogue to control theory, where predictive models can be used to stabilise sparse noisy sensory input (in analogy for instance to Kalman Filters), to enhance the processing of sensorial information, as trackers (retaining awareness of things coming into and out of the senses), and to implement efficient (model) predictive control. The usefulness of forward emulators extends to covert motor activity, where they may be used offline to produce motor imagery, to estimate the effect of hypothetical actions, to develop and evaluate plans, such as model-based planning in control theory (e.g., model predictive control and Optimal Control), and to simulate the observed actions of other people.

On the other hand, the forward loop, i.e. mirroring, should reproduce human action planning. Many researchers have focused on human movement, and how it accords to basic motion rules that have an origin in the central nervous system. In this respect, work by Todorov [131], Viviani [135], and many others (going back to the 1970s) show evidence that human beings move according to optimal criteria, such as minimum jerk and others. Viviani says that human movement responds to minimum jerk, Todorov suggests the existence of a trade-off between

elementary cost terms.

Optimal Control (OC) thus reproduces simple sensory-motor primitives. OC modelling of human motor strategies is thus general: where alternative models of the neural control of movement require a detailed description of how the desired goal should be accomplished, with Optimal Control it emerges naturally from basic principles. Application of OC with a Receding Horizon scheme explains another observed fact, known as the *minimum intervention principle* [132], which states that task-irrelevant deviations are left uncorrected. Viviani [135] also showed that lateral acceleration in movements approaches a limit that depends upon speed and curvature (the *two-thirds power law*, and the related *isochrony* principles). Moving to driving tasks, it has been determined that similar principles apply to path planning: in particular minimum jerk (sometimes as a trade-off with minimum time) can explain how human drivers generate trajectories (Cossalter et al. 1999 [51], Da Lio et al. [52], Bertolazzi et al, 2009 [36]). There also exists an analogy of the two-thirds power law by which human drivers use only a limited amount of acceleration, which is called the *willingness envelope* [40].

3.2.3 *Architecture of the Codriver*

3.2.3.1 *Overall Architecture*

The Codriver used on the CRF demonstrator is a cognitive agent designed on the principles described above. Figure 3.12 shows its architecture: the agent's body is the vehicle, its environment is the road and the other users, and the moro input is the longitudinal and lateral control. With respect to Figure 3.11, the perception-action link is expanded into a subsuming hierarchy of different PA loops, able to control the switch between covert and over actions. In this way the Codriver is able to analyse many hypothetical actions, i.e. goals, matching them with observed driver behaviour, and thus selecting which action the driver is sending to execution, and, what is most, knowing why. A key feature of the application is included in the FIE box, which contains *forward and inverse emulators*. Forward emulators model the dynamics of the body and the environment: they receive efferent copies of motor commands and ultimately anticipates sensory input. On the other hand, the inverse dynamics, i.e. the problem of how to achieve desired perceptual goals, is obtained, in our implementation, by optimal control, which

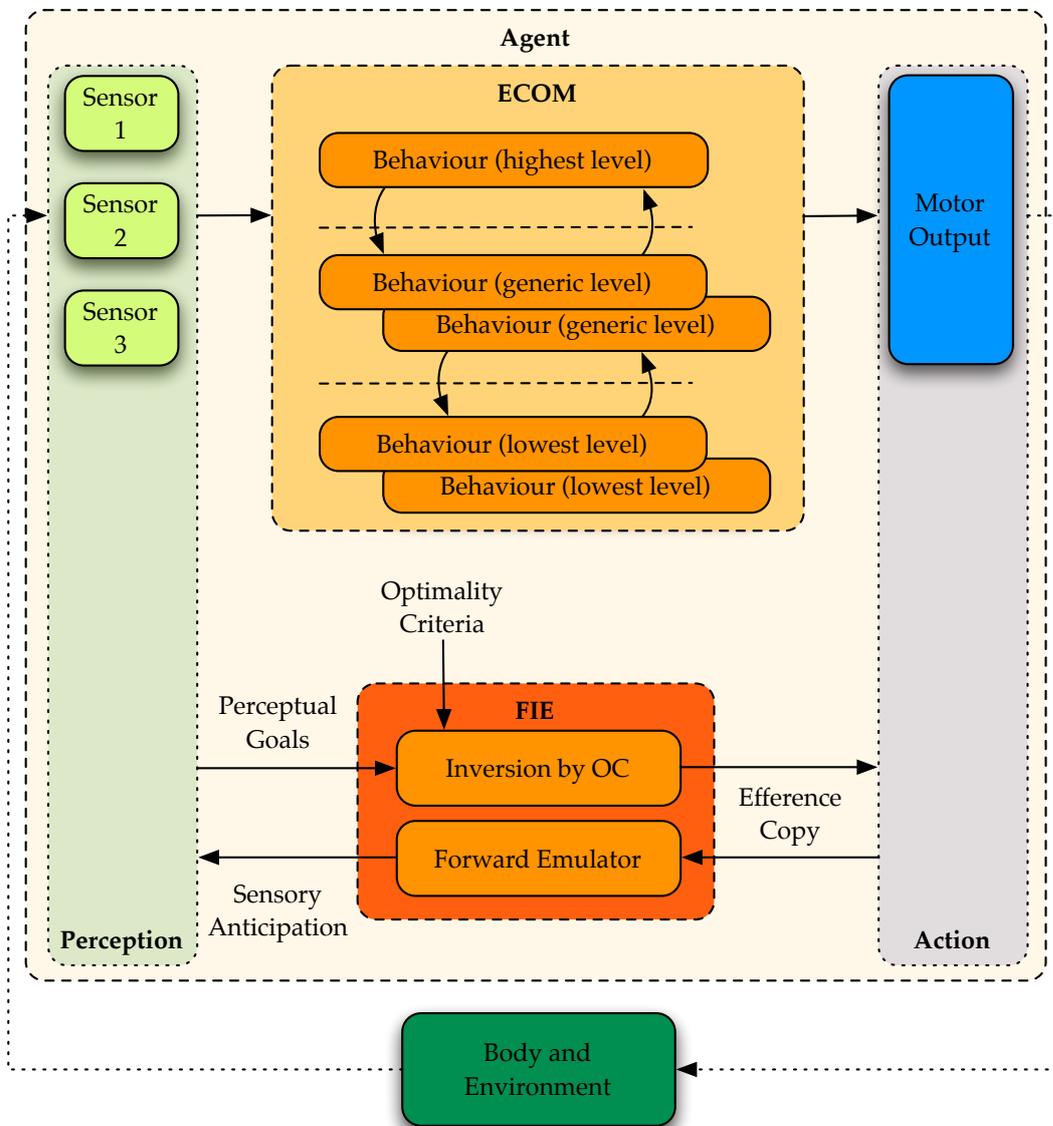


Figure 3.12: Architecture of the Codriver for Continuous Support on CRF demonstrator

is justified by the considerations given at the end of previous section, where the advantages of modelling driver planning by means optimal control were pointed out. The optimality criteria used for the optimal control problem selects the inverse model from among the potentially infinite number of ways of inverting the dynamics.

The other key feature is the ECOM box, containing the implementation of the Extended Control Model. Four layers have been used, ranging from *motor primitives* (the lowest regulating level) up to the topmost strategy layer, which manages the inference of intentions and interactions.

Here follows a description of all the blocks composing the Codriver.

3.2.3.2 *Forward emulators*

In this implementation, emulators are focused on the prediction of the lateral and longitudinal ego vehicle dynamics. In addition, very simple predictive models for the dynamics of the other objects are used.

There are many vehicle dynamics models in the literature that might be adopted as forward emulators, remembering that modelling simplifications and assumptions affect the accuracy of the results, so that the choice of the model depends on the form of analysis that has to be carried out. In this context, models are used to test the viability of different hypotheses of human driving intentions, which is accomplished by comparing observed to predicted behaviour. Thus, the aim is to develop human-like coupled inverse/forward models, which should resemble more the idea the driver has of vehicle dynamics rather than the real vehicle dynamics themselves.

Thus, quasi-static models have been used, which ignore transient phenomena that are unlikely to be conducted by humans because of the limited frequency response of humans in either sensing or actuation (humans are actually able to handle vehicle dynamics until a certain frequency, but to keep the problem simple and obtain closed-form solution this trade-off has been adopted). On the other hand, these models capture phenomena like under-steering, which, if otherwise ignored, would lead to mismatch between human and the Codriver. Lateral and longitudinal dynamics are considered in a separate way.

For lateral dynamics, classical steady-state cornering is used, as in Chapter 3.3 of

Abe, 2008 [28]. The equations are then written in curvilinear coordinates as in Cossalter et al., 1999 [51] (Figure 3.13):

$$\begin{aligned} \dot{s}_n(t) &= u_0 \sin(\alpha(t)) \\ \dot{\alpha}(t) &= -\frac{\kappa(s_s(t))}{s_n(t)\kappa(s_s(t))}u_0 \cos(\alpha(t)) + u_0\Delta(t) \\ \dot{\Delta}(t) &= \frac{1}{k_1 + k_2u_0^2}j_\Delta(t) \end{aligned} \quad (3.1)$$

Three state variables are considered: the curvilinear ordinate s_n , the angle

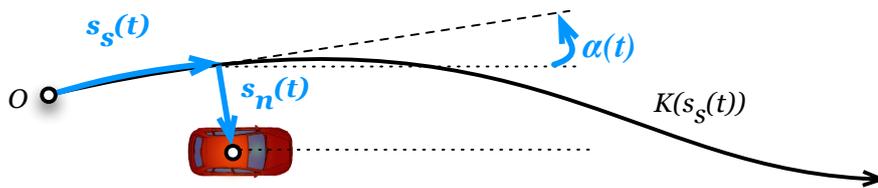


Figure 3.13: Curvilinear coordinates $s_s(t)$, $s_n(t)$, and $\alpha(t)$ for a given road geometry (i.e. curvature profile $\kappa(s_s(t))$)

between vehicle velocity and lane direction α , and the curvature of current trajectory Δ . κ is lane curvature, which is a function of the curvilinear abscissa, s_s . Longitudinal velocity is considered constant at the value of u_0 , so that:

$$s_s(t) = u_0 t \quad (3.2)$$

The control input is j_Δ , the steering rate applied by the driver, which is also proportional to lateral jerk. These equations aim at describing how the driver intends the lateral control: the relation in the third equation means that he perceives the curvature as proportional to steering angle, with a gain which depends on speed. The constants k_1 and k_2 describe the vehicle understeer behaviour, and take into account also the steering wheel ratio and vehicle wheelbase. These constants have been designed based on vehicle parameters and validated using real measurements on vehicle yaw rate and steering angle.

The model for longitudinal dynamics is simpler:

$$\begin{aligned} \dot{s}_s(t) &= u(t) \\ \dot{u}(t) &= a(t) \\ \dot{a}(t) &= j_p(t) \end{aligned} \quad (3.3)$$

In this case, the variables are the curvilinear abscissa s_s , forward velocity u and acceleration a . The control input is longitudinal jerk j_p , which is a non-linear function of the accelerating and braking pedal rates.

3.2.3.3 *Inverse models*

Inverse models are produced by means of optimal control (OC). In this way, motor primitives are obtained, which require only the forward model already described in equations 3.1 and 3.3 and an optimality criterion.

Inverse models link perceptual goals to the actions needed to achieve those goals (Figure 3.12). For the dynamic system described in the previous section 3.1, 3.3, perceptual goals are the desired states to be achieved at some time horizon T . Thus, inverse models answer to the question of how to reach a desired state at some future time T in an optimised way (which, formulated in this way, is indeed an optimal control problem). Since there may be several types of final states and optimisation criteria, the inversion problem produces one solution each, which constitute different motor primitives. Thus, in the proposed architecture in Figure 3.12, the block labeled “inversion by OC” contains the algorithm for solving the optimal control problem, while the lowest level behaviours of the ECOM architectures are the motor primitives, which may be conceptually thought of as instantiations of optimal control problems.

3.2.3.4 *First ECOM layer: Motor Primitives*

A motor primitive is a class of inverse model produced by certain types of final conditions and optimisation criteria. A few motor primitives have been determined to be relevant for the Codriver application, which will be described below, divided into longitudinal and lateral.

Longitudinal motor primitives

Longitudinal motor primitives deal with the adaptation of speed. They are generated by the following Optimal Control problem:

$$\begin{aligned} \text{Minimise : } & J_s = \int_0^T w_T + j_p^2(t) dt & (3.4) \\ \text{Subject to} & \\ \text{Vehicle dynamics : } & \text{Equations (3.3)} \\ \text{Initial conditions : } & s_s(0) = s_{s,0} \\ & u(0) = u_0 \\ & a(0) = a_0 \\ \text{Final conditions : } & \text{Manoeuvre goal} \\ \text{where : } & T = \text{FREE} \end{aligned}$$

The cost functional J_s is a tradeoff between minimum jerk j_p and minimum time T , since minimising the integral of a constant w_T from 0 to free T means in the end to minimise T . This allows modelling a variety of longitudinal behaviours, from minimum effort (setting $w_T = 0$) to different amounts of driver hurry, with ideally minimum time manoeuvring for $w_T \rightarrow \infty$. Thus, w_T plays the role of a *motivation* parameter, modelling how fast the intention can be executed, and in this sense it will be set by higher ECOM layers.

At the bottommost level, instead, only the different basic driver goals are set, using different final conditions (and initial conditions are set according to actual state):

Speed Matching (SM)

The first motor primitive achieves a desired uniform speed u_T at a given location x_T . It is thus produced by the final conditions:

$$\begin{aligned} s_s(T) &= s_{s,T} \\ u(T) &= u_T \\ a(T) &= 0 \end{aligned} \quad (3.5)$$

The solution of this simple OCP can be defined as:

$$\begin{aligned}
s_s(t) &= s_{s,SM}(t, s_{s,T}, u_T, w_T) \\
u(t) &= u_{SM}(t, s_{s,T}, u_T, w_T) \\
a(t) &= a_{SM}(t, s_{s,T}, u_T, w_T) \\
j_p(t) &= j_{p,SM}(t, s_{s,T}, u_T, w_T)
\end{aligned} \tag{3.6}$$

Namely, the SM motor primitives are functions of time which depend also on parameters such as target final states $s_{s,T}$ and u_T and the time pressure w_T . In Cognitive System terminology, $s_{s,T}$ and u_T would be defined as the *perceptual goals* which are set at the immediately higher layer, while w_T would stand for an inner motivation state, and is thus set at even higher level. The functions 3.6 are not executed in reality, so that they represent a *covert* motor activity of the Codriver, i.e. *motor imagery*.

The output of this lowest level of the ECOM architecture is $j_{p,SM}$, i.e. the action required to achieve the perceptual goal.

The time horizon T_{SM} which optimises 3.4 is also a function of the perceptual goals and of the internal state:

$$T = T_{SM}(x_T, u_T, w_T) \tag{3.7}$$

The computation of the solution (3.6) and (3.7) could be carried out in different ways: online Perception-Action learning was used in DIPLECS (Felsberg et al., 2010 [67] and Windridge et al., 2012 [139]), but the OCP could be also solved in real-time, as it was done in [35], [33], [36]. Training of perception-action learning could be also done synthetically with offline computed solutions of the OCP. However, with these simple models, the solution of the OCP can be found in a closed form (which cannot be published here), as a function of its parameters, and this way has been used within the project. This allowed fast computation of several solutions, updating all the parameters in a discrete range and thus investigating a large variety of driver objectives and internal states.

Speed Adaptation (SA)

A second type of longitudinal motor primitive is the adaptation of speed. It differs

from SM in that the location at which speed u_T has to be reached is unconstrained. Thus, final conditions are:

$$\begin{aligned} s_s(T) &= FREE & (3.8) \\ u(T) &= u_T \\ a(T) &= 0 \end{aligned}$$

In a similar fashion to SM, SA motor primitives can be defined as parametric functions of target speed u_T and driver motivation w_T , obtaining equations analogous to 3.6 and 3.7:

$$\begin{aligned} s_s(t) &= s_{s,SA}(t, u_T, w_T) & (3.9) \\ u(t) &= u_{SA}(t, u_T, w_T) \\ a(t) &= a_{SA}(t, u_T, w_T) \\ j_p(t) &= j_{p,SA}(t, u_T, w_T) \end{aligned}$$

$$T = T_{SA}(u_T, w_T) \quad (3.10)$$

Lateral motor primitives

Lateral motor primitives deal instead with the adaptation of travel direction and lateral position. They are generated by the following Optimal Control problem:

$$\text{Minimise : } J_n = \int_0^T j_{\Delta}^2(t) dt \quad (3.11)$$

Subject to

Vehicle dynamics : Equations (3.1)

Initial conditions : $s_n(0) = s_{n0}$

$\alpha(0) = \alpha_0$

$\Delta(0) = \Delta_0$

Final conditions : Manoeuvre goal

where : $T = \text{SET}$

In this case, there is not the equivalent of w_T of equations 3.11. For this reason the problem is formulated in an alternative way, and final time is set at T . Final

time T at which the motor primitive should be completed has thus the same role as w_T in the longitudinal case.

The different driver goals for longitudinal dynamics are resumed below.

Lateral Displacement (LD)

This motor primitive involves adjusting the lateral position in the lane. This is described by the following final conditions:

$$\begin{aligned} s_n(T) &= s_{n,T} \\ \alpha(T) &= 0 \\ \Delta(T) &= \kappa_T \end{aligned} \tag{3.12}$$

This means that, at the end of a manoeuvre of given duration T , the lateral position must be $s_{n,T}$, the final travel direction must be parallel to the lane, and the trajectory curvature must match the lane curvature κ_T .

The motor primitive is thus made of functions parametric in $s_{n,T}$ and T (one motor goal and one motivation parameter):

$$\begin{aligned} s_n(t) &= s_{n,LD}(t, s_{n,T}, T) \\ \alpha(t) &= \alpha_{LD}(t, s_{n,T}, T) \\ \Delta(t) &= \Delta_{LD}(t, s_{n,T}, T) \\ j_{\Delta}(t) &= j_{\Delta,LD}(t, s_{n,T}, T) \end{aligned} \tag{3.13}$$

Lane Alignment (LA)

This primitive assumes that, sooner or later, the travel direction will be realigned with the lane and the curvature will match the lane curvature κ_T . This process is supposed to occur in a time T and describes the lane- following task in so far as the driver is not concerned about lateral position. The goals for this motor primitive are:

$$\begin{aligned} s_n(T) &= FREE \\ \alpha(T) &= 0 \\ \Delta(T) &= \kappa_T \end{aligned} \tag{3.14}$$

The corresponding primitive may be concisely summarized as a function of one parameter only, i.e. the time T necessary for the alignment.

$$\begin{aligned}
 s_n(t) &= s_{n,LA}(t, T) \\
 \alpha(t) &= \alpha_{LA}(t, T) \\
 \Delta(t) &= \Delta_{LA}(t, T) \\
 j_\Delta(t) &= j_{\Delta,LA}(t, T)
 \end{aligned}
 \tag{3.15}$$

Summing up, 4 motor primitives have been defined: one that achieves a specified speed at a specified location, another that adapts speed, a third that reaches a target lateral position and a final one which realigns the travel direction to the lane. These yield both the space/time trajectory and the control necessary to obtain them, thereby forming the bottom layer of the ECOM architecture of Figure 3.12. Other primitives are of course possible, but these were sufficient for implementing the remaining part of the ECOM architecture.

Both lateral and longitudinal primitives include a parameter that represents a desired “quickness”, modelling the fundamental fact that human sensory-motor activities are always a tradeoff between movement accuracy and time. For the longitudinal control, this parameter was introduced as a minimum-time term in a free time OC problem; for the lateral case an alternative formulation was given, with a fixed final time, in which the final time is thus used to model temporal requirements.

3.2.3.5 *Second ECOM layer: Simple trajectories dealing with obstacles or lanes*

The second behavioural layer of Figure 3.12 starting from the bottom combines the above motor primitives in order to achieve simple manoeuvres that individually deal with one obstacle, or one lane or road feature (curve or landmark) at a time. In the hierarchical architecture, upper layers set the goals for lower layers, which practically means that this layer sets the perceptual goals of the motor primitives, i.e. their final states $s_{s,T}$, u_T , $s_{n,T}$, κ_T . As in the case of motor primitives, the trajectories can be divided into different categories, depending on

the objectives they set.

Obstacles

To deal with obstacles, a predictive model of obstacle motion is first needed. One approach would be to carry out inference of the observed vehicle intentions the same inference of host vehicle intentions is carried out, namely: re-using the framework here developed. In other words, in this case the Codriver might also “stand in the shoes” of the other vehicle users as well, which actually is what people do when driving.

However, although fascinating, this approach has not been used for a number practical reasons: the speed and direction of travel of other vehicles is known with less accuracy than that of ego-vehicle, acceleration measurement tends not to be reliable, other driver controls are not directly observable and a view of the road network from the host vehicle perspective is not available (e.g., at intersecting roads, where the direction observed vehicles might be travelling in is not sufficiently known). Thus, obstacle forward models had to be simplified.

The different strategies to deal with obstacles are listed below.

Follow Object (FO)

The purpose of this manoeuvre is to approach a preceding vehicle, as in Figure 3.14 using manoeuvre a, producing a desired time headway gap t_h .

The obstacle longitudinal motion model assumes that longitudinal velocity v_o (i.e., the obstacle velocity projected into lane direction) is fairly constant. To deal with accelerating obstacles it is necessary to rely on the continuous updating of motor plans of the receding horizon iterations. Such updating of motor plans to improve robustness of human sensory-motor strategies are described, for example, by Todorov [132], and they are also implicit in artificial systems in schemes like Receding Horizon Optimal Control and Model Predictive Control, which is essentially what is proposed here.

The Follow Object manoeuvre is thus a perception-action map, which takes as input the object, the desired time headway, and the time pressure parameter w_T and returns a Speed Matching (SM) primitive such as 3.6:

$$\text{FollowObject} : (\text{object}, t_h, w_T) \rightarrow SM(s_{s,T}, u_T, w_T) \quad (3.16)$$

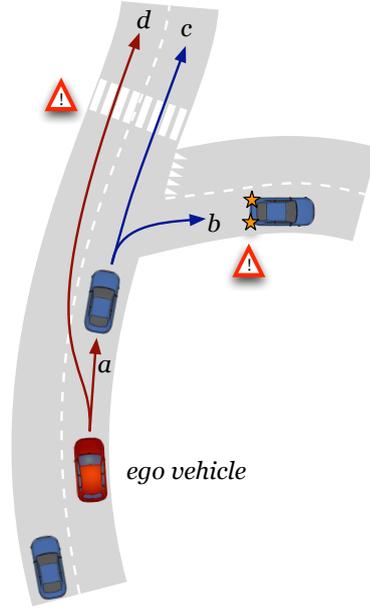


Figure 3.14: A possible scenario, where ego vehicle travels in traffic, with road intersections, landmarks and multiple possible goals for all road users

This means computing the target distance $s_{s,T}$ and velocity u_T that correspond to following the object as required, namely:

$$\begin{aligned} u_T &= v_o & (3.17) \\ x_T &= s_o + v_o T - v_o t_h - l_o \end{aligned}$$

where l_o is the longitudinal distance that accounts for the lengths of host vehicle and obstacle plus any extra desired clearance, $v_o t_h$ is the time headway gap, s_o is the initial distance of the object and T is the manoeuvre duration, which is obtained by solving 3.17 together with 3.7.

The Follow Object function thus instantiates an SM primitive after computing its input parameters. In Figure 3.12 arrows descending from a level to the one below indicate this form of input relationship. The details of the control movement (if needed) are given by the primitive 3.6. The current value of the longitudinal control:

$$j_{p,0} = j_{p,SM}(0, s_{s,0}, u_0, w_T) \quad (3.18)$$

is here of particular significance, because it indicates how the Codriver should drive in order to follow the object, which can be directly compared with the longitudinal control that the human driver is currently employing.

As in the case of the motor primitives, these second-layer perception-action cycles may be either overt or covert, representing, in the latter case, manoeuvres under evaluation.

Finally, the followed object does not need to be in the host vehicle's lane for this function to apply. If it is travelling in a parallel lane, including the case where it is behind the host car, then this function may be used to compute manoeuvres that, for instance, open a gap before a lane change may be executed.

Clear Object (CO)

The purpose of this manoeuvre is to clear a frontal object on either side of the host vehicle, as showed in Figure 3.15.

As indicated, understanding the directional intentions of the object vehicle would

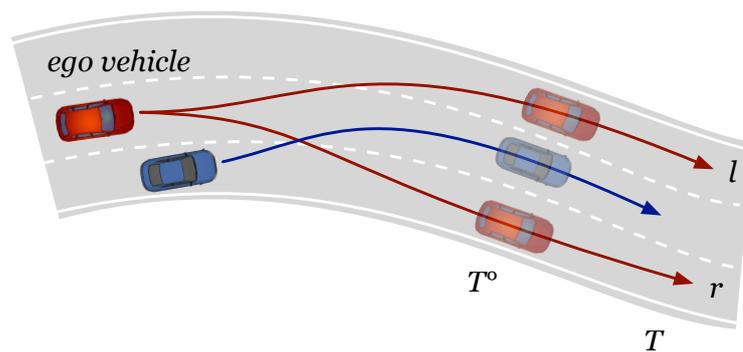


Figure 3.15: Possible evasive manoeuvres for obstacle handling

ideally require knowing the road network in order to find which lane the obstacle might be following. However, in the present version of the system only the geometry of ego vehicle road/lane is known, so that it is only possible to assess whether the object is moving in ego vehicle road, or whether it is moving across the road, in which case the understanding of its intentions will be correspondingly degraded.

If the object was moving in ego vehicle road, its lateral movement would follow

a model similar to (3.15), i.e., the object would sooner or later realign with the lane. Since it is not possible to measure the curvature of the object trajectory directly, the problem is simplified by setting $\Delta_0 = 0$ in boundary conditions 3.11, and $\kappa(\cdot) = 0$ in 3.1. With these simplifications, the OCP given by (3.11), (3.1), and (3.14) can be solved analytically, yielding the approximate predictive model for object lateral motion employed here:

$$s_n(t) = s_{n,0} + v_n t \left[1 - \left(\frac{t}{T^*} \right)^2 + \frac{1}{2} \left(\frac{t}{T^*} \right)^3 \right] \quad (3.19)$$

The model contains the parameter T^* , which stands for how long the object manoeuvre will last: in essence, a kind of intentional assessment. At this point T^* is not estimated, but the heuristically-derived value of $T^* \simeq 2.5$ s is used. Thus, the maximum lateral displacement of the object will be achieved at $t = T^*$:

$$s_{n,max} = s_{n,0} + v_n \frac{T^*}{2} \quad (3.20)$$

If this position falls within one lane of the current object lane then model (3.19) is confirmed (i.e., the object is assumed to follow our road, possibly changing one lane only). If not, the object is considered to be crossing ego vehicle road. In this case its transverse motion is taken to be uniform:

$$s_n(t) = s_{n,0} + v_n t \quad (3.21)$$

With an object predictive model (in this case the simple equations (3.17)-first, (3.19), (3.21)) it is now possible to compute evasive manoeuvres, as Figure 3.15 shows. The blue vehicle is the obstacle and the blue trajectory is its predicted motion. Even if a more sophisticated obstacle intention prediction was available, the process hereafter would be the same: first compute the encounter time T^o by combining the longitudinal motion models; then produce an LD primitive (i.e., parameters $s_{n,T}$ and T) so that a specified clearance c_o , is obtained at T^o .

The ClearObject function returns two LD primitives, one for clearing on the left (l), another for clearing on the right (r):

$$\text{ClearObject} : (\text{object}, c_o) \rightarrow LD(s_{n,T}, T) \quad (3.22)$$

The format above implies that the second layer function ‘‘ClearObject’’ produces the parameters $s_{n,T}$ and T of the first layer ‘‘Lateral Displacement’’ motor primitive

(thus subsuming it) that clears the specified object at the specified distance c_o , which necessarily accounts for the width of the two vehicles plus any desired clearance. The sign of c_o may be conveniently used to specify whether the clearing occurs to the right or to the left.

The ClearObject behaviour above is simplified: firstly there is only one hypothesis for T ; secondly there is also one hypothesis only for the longitudinal control used during the evasive manoeuvre, which is FF (below) with a plausible value for w_T .

Lanes

Free Flow (FF)

This manoeuvre produces a SA primitive by guessing a target speed u_T . This is achieved by assuming a plausible value for T and solving (3.10) for u_T .

$$\text{FreeFlow} : (w_T, T) \rightarrow SM(u_T, w_T) \quad (3.23)$$

In theory, FreeFlow should be a function of two parameters, with T dictating how long the acceleration lasts. However, since T weakly influences the first part of the manoeuvre, the Codriver always generates only one hypothesis for T (which is $T = 5$ s) and relies on receding horizon updates to refine the estimate of the latter part of the manoeuvre.

Lane Follow (LF)

This is a wrapper for the LA/LD motor primitives. It takes as input a desired lateral position in a specified lane (in lane units) and the manoeuvre time T and returns a LD primitive. The final lateral position may alternatively be “free”, in which case the LA primitive is before converted into a LD one.

$$\text{LaneFollow} : (\text{lane}, \text{position}, T) \rightarrow LD(s_n, T) \quad (3.24)$$

Landmark/SpeedLimit (LM)

Landmarks are used to represent speed limits at specified locations. These are set in salient locations, e.g. a pedestrian crossing, and at the beginning of a road section with a posted speed limit. The Landmark function takes the speed limit

and position and returns a SM primitive (making only one hypothesis for w_T). This is a manoeuvre that aims at matching the speed limit, and which thus serves to assess in advance whether the driver is going to observe that limit.

$$\text{LandMark} : (\text{speedLimit}, \text{position}) \rightarrow SM(s_{s,T}, u_T, w_T) \quad (3.25)$$

Curve (CU)

This returns a SM primitive that approaches a curve with the correct speed. The existence of a curvature-acceleration-speed relationship has already been mentioned [135]. Curvature data from ADAS digital maps is used to compute the appropriate location and speed [40]. Two hypotheses are made representing two different percentiles of driver lateral acceleration. Only one hypothesis is made for w_T .

$$\text{Curve} : (\text{curve}, \text{driverPercentile}) \rightarrow SM(s_{s,T}, u_T, w_T) \quad (3.26)$$

3.2.3.6 *Third Layer: Navigation*

So far, the functions of the second layer may be regarded as operators that translate simple goals, into motor primitives. Except for Clear Object, they return either a longitudinal or a lateral primitive. The third layer is thus responsible of require executable navigation plans, putting together these potential motor tasks. This means that combined longitudinal and lateral control, i.e., couplings of longitudinal-lateral primitives, are produced, such that each couple accounts for all the obstacles simultaneously.

In other words, the third layer produces manoeuvres that represent higher-level goals/intentions, such as a, b, c, d of Figure 3.14.

It is worth noting that this layer still produces multiple manoeuvres. For an autonomous system, they represent covert motor alternatives, such as in 3.14, from among which to choose. For the Codriver, they constitute hypotheses to be tested against the observed behaviour of the driver.

The Codriver thus need to produce a number of hypothetical motor activities spanning the space of possible intentions. To generate a complete set of hypotheses, the Codriver starts guessing what the lateral intentions of the driver might

be. For a single road, with possibly multiple lanes, it thus generates an array of Lane Follow (LF) motor tasks as shown in Figure 3.16. Three of these, labeled

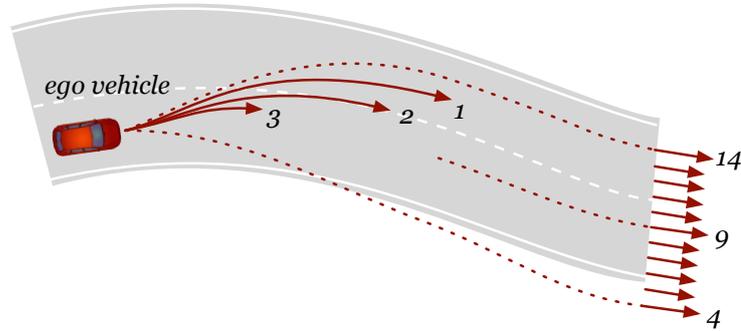


Figure 3.16: Possible lateral intentions of the driver

1-3, are of type LA, with three hypotheses for the alignment time $T \in \{T_1, T_2, T_3\}$. These will be used to test whether the driver is going to simply realign with the lane without any particular care for the exact lateral position in it. In this implementation implementation the times are set at:

$$\begin{aligned} T_1 &= 3 \text{ s} \\ T_2 &= 1.5 \text{ s} \\ T_3 &= 1 \text{ s} \end{aligned} \tag{3.27}$$

with the first two turning out to be the most frequent matches. In addition the co-driver makes 11 hypotheses of type LD, labeled from 4 to 14, in which the final lateral position $s_{n,T}$ spans three lanes, from the adjacent right lane to the left lane in steps of 0.25 lane widths. The hypothesis labeled 9, corresponds to $s_{n,T} = 0$, which represents the intention to return to the center of the lane. Hypotheses 8 and 10 are respectively 0.25 lanes off-center, which is approximates to the intention to move near one edge or the other of the lane. Hypotheses 7 and 11 represent the intention to go over the lane divider (possibly as the beginning of a lane change) and the remaining hypotheses are complete lane changes. Even if in the situation of Figure 3.16 there is no right lane at all, hypotheses 4-7 are nonetheless generated, which represent “running out” of the lane. The reason for this excess hypothesis generation is also to test for possibly incorrect driver

actions. For these LD primitives, only one hypothesis is made for the final time T in initial conditions of (3.11).

In the case of road forks, such as in Figure 3.14, another array of hypotheses like the above would be needed to accommodate the bifurcation, but this has not been implemented in the system.

The following step is the association of each lateral motor plan with one or more longitudinal plans. Obstacles are first considered: for each, the Follow Object and Clear Object manoeuvres are computed. For consistency with the matching lateral manoeuvre, the final time T used in (3.13) has been also used in (3.22).

At this stage, two cases can happen:

1. if the i -th lateral manoeuvre of Figure 3.16 clears all the obstacles, this is associated to a FreeFlow longitudinal primitive representing the intention of travelling along a non-colliding path. For simplicity, only one hypothesis is made for the FF parameter, w_T , i.e. a value for low hurry driving, since it is not as important to understand how fast the driver wishes to drive in this case.
2. if the i -th lateral manoeuvre Figure 3.16 falls within the left-right boundaries of one or more objects, then the FollowObject manoeuvre for the most demanding object is selected. In this case, two hypotheses for the time headway t_h are made:

$$\begin{aligned}t_{h,1} &= 1.1 \text{ s} \\t_{h,2} &= 0.6 \text{ s}\end{aligned}\tag{3.28}$$

Then, curves are also considered: for every curve the CU primitive is computed and the most demanding one retained. This is associated with all the trajectories sharing the same road path, e.g. in Figure 3.14 a, c, d share the most demanding curve primitive of the main road path, and b of the bifurcation path.

Lastly, landmarks and speed limits are considered the same way as for curves.

After the above process, a list of manoeuvres is obtained, with both lateral and longitudinal control of two types: the first are trajectories that do not collide with obstacles which are associated to FreeFlow longitudinal control; the second are trajectories that might collide with an object which are associated with two

hypotheses for longitudinal control, corresponding to tight and very tight time headway. In addition, every manoeuvre is also associated to an alternative curve and a landmark longitudinal primitive. The reason for keeping the longitudinal primitives separate, instead of taking the most critical one, is the same as for the out-of-lane hypotheses of Figure 3.16. Namely, the purpose is also to test for mistakes, such as a driver approaching the front vehicle correctly but not a curve within the same road.

3.2.3.7 Fourth Layer: Strategy

The topmost layer is where the highest-level functions are produced. For instance an autonomous agent would here deliberate upon which manoeuvre of the third layer to execute. For the Codriver, this level is where inference of intentions is completed and interactions are born.

In order to correctly infer driver intentions, the hypotheses generated at layer 3 must thus be tested. A saliency-based approach is used, which considers both the difference between the hypothesis and actual driver behaviour and also the plausibility of the hypothesis itself. For the i -th candidate manoeuvre, a penalty is therefore computed as follows:

$$J_i = w_\Delta \|j_{\Delta,i} - j_\Delta^*\|^2 + w_p \|j_{p,i} - j_p^*\|^2 + w_n J_n \quad (3.29)$$

The first term is the square of a proper difference function between actual Codriver steering control j_Δ and driver's control j_Δ^* with weight w_Δ . The second term is the distance between the longitudinal controls, j_p versus j_p^* with weight w_p . These two together measure a kind of distance between the driver and the i -th Codriver hypothesis. The third term is the steering cost of the manoeuvre, i.e. the objective function of (3.11). Adding this term means that manoeuvres with higher steering costs are considered less plausible. Thus, for instance, a manoeuvre that requires less steering activity such as going straighter would be preferred to one that steers more, for the same distance to the driver behaviour. A similar term for the longitudinal control has not been included, though.

The computation of the distance between driver and Codriver manoeuvres implies a time window for the comparison, which determines a tradeoff between accuracy and delay of inferred intentions (longer observations may be more accurate but cause more delay). This implementation aims at discriminating quickly among

the tactical manoeuvres of layer 3, and thus uses nearly instantaneous comparison windows: for every hypothesis the current values of the predicted longitudinal and lateral controls, $j_{\Delta,i}(0)$ and $j_{p,i}(0)$, are respectively compared with a 200 ms first-order filtered value of the steering wheel rate \bar{j}_{Δ}^* and with a Kalman estimation of the longitudinal jerk \bar{j}_p^* . Equation (3.29) then becomes:

$$J_i = w_{\Delta} f^{-} (j_{\Delta,i}(0) - \bar{j}_{\Delta}^*)^2 + w_p (j_{p,i}(0) - \bar{j}_p^*)^2 + \frac{1}{T} J_n \quad (3.30)$$

The function $f^{-}(\cdot)$ is the negative part, returning zero if the codriver is faster than the human driver.

For better comprehension of how the upper layer infers driver's intentions, is useful to introduce the chart in Figure 3.17, where the control output space is represented: the steering wheel rate j_{Δ} is reported in abscissa and the longitudinal jerk j_p in ordinate. A manoeuvre is represented here by a curve parameterised by time, as it happens in the last relations of Equations (3.6), (3.9), (3.13) and (3.15). Conceptually, this chart may be regarded as a projection of the internal

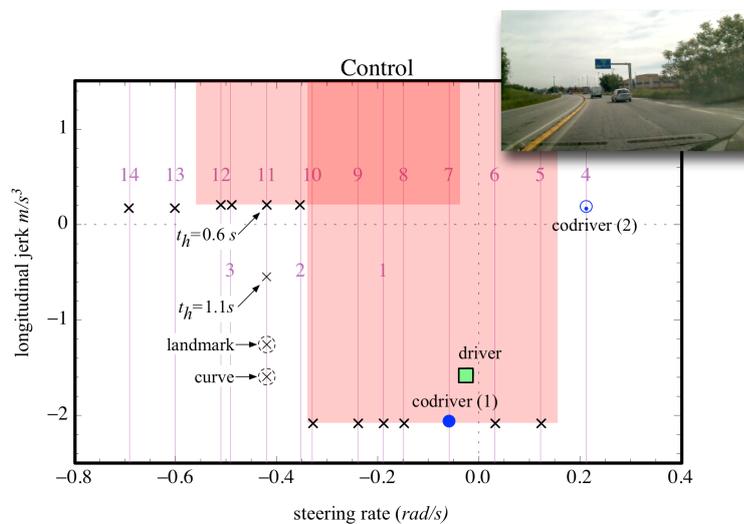


Figure 3.17: Chart representing the projection of system states into the motor output space

state of the cognitive systems onto the motor output space. This is inspired by embodied cognition, in which perception-action links imply that internal perceptual states of the system may be represented in terms of their motor output or a

common code (Hommel et al., 2001 [85]).

In the chart, the vertical lines labeled 1-14 represent the lateral control $j_{\Delta}(0)$ of each lateral motor primitive generated at layer 3, i.e. those of Figure 3.16. The two red rectangles stand for two vehicles, which can be seen also in the corresponding image at top right, acquired during the real tests. The shapes are limited to rectangles due to the simplifications made in the clear object section. The tallest rectangle is the nearest car; the other is the far van. Rectangles are bounded by the left and right ClearObject manoeuvres, and by the FollowObject manoeuvre at the bottom side. The rectangles look darker where they overlap. The cross markers (x) are the manoeuvres produced at layer 3. Manoeuvres 4, 13 and 14 clear both objects and are associated with FF longitudinal primitives, whereas the others are associated with the FO primitives of the critical object. Since there are two hypotheses for time headway ($t_h = 1.1 s$ and $t_h = 0.6 s$) there are indeed two FO primitives for each trajectory. For clarity, only that with $t_h = 0.6 s$ is showed, except for trajectory 11, in which the second one is shown in light gray $t_h = 1.1 s$.

In the case given here, there is no critical curve and no landmark in front of the vehicle. If there were, additional manoeuvres would be generated for every trajectory according to the limiting jerk of curves and landmarks, such as those shown in dotted circles, for example trajectory 11.

The filled and empty circles, respectively trajectory 7 with time headway $t_h = 0.6 s$, and trajectory 4 with free flow, are the matching Codriver manoeuvres. The filled circle “co-driver (1)” corresponds to criterion (3.30) with $w_p = 0$. By weighting only the lateral control, this manoeuvre represents the intentional trajectory of the driver. Conversely, manoeuvre “co-driver (2)” uses a large weight w_p and typically preserves the longitudinal speed, producing an evasive manoeuvre.

3.2.4 *Implementation of the Codriver for continuous support*

In the CRF application, the Codriver is used as a peer, exploiting the two alternative manoeuvres above. It was already pointed out that the Codriver makes hypotheses, not just for correct behaviours, but also for incorrect behaviours. For example, 3.17 shows four longitudinal control hypotheses for trajectory 11; and

Figure 3.16 shows hypothetical trajectories that fall outside of the road area. For manoeuvre “co-driver (1)”, the Codriver seeks to determine what the intentional trajectory is, and, within this trajectory, what the intentional longitudinal policy is. Thus it seeks to determine whether the longitudinal control corresponds to a correct manoeuvre or, if not, which longitudinal control should be used to implement the correct manoeuvre. For example, if the “co-driver (1)” manoeuvre was a trajectory with $t_h = 0.6 s$, the correct manoeuvre should be $t_h = 1.1 s$ (not considering landmarks or curves that might be present), and the Codriver would also then implicitly know how to correct the driver’s incorrect behaviour. This information is thus used by the HMI of the Continuous Support function, which is inactive when the driver’s manoeuvring is correct, but produces a feedback with the required correction (and the cause of activation) if greater than a set threshold.

The Codriver also establishes whether the lateral control is correct, and this information is also used by the HMI for lateral feedback. For instance, trajectories 11-14 of Figure 3.17 correspond to a non-existent lane, and would produce HMI interactions if they were chosen by the driver (the same happens if the driver selects a lane which is occupied by a lateral/rear vehicle).

On the other hand, the evasive manoeuvre “co-driver (2)” is an alternative option for rectifying driver mistakes, which acts on the lateral control instead. By way of example, in Figure 3.17 the Codriver estimates that it is possible to change lane to the right to preserve FF longitudinal control. Within the CRF implementation, evasive manoeuvres are used to a limited extent in two cases:

1. if the evasive manoeuvre is within the same lane of manoeuvre (1) (which happens for example if manoeuvre (1) is affected by an obstacle that occupies only part of the lane, such as a vehicle on a nearby lane but very close or slightly inside our lane), in which case this manoeuvre is selected for generating feedback in place of the original manoeuvre. Thus, if there is an in-lane evasive manoeuvre, the driver will not receive longitudinal feedback but rather a lateral feedback, e.g., because he/she should open a greater clearance with respect to a vehicle very close or slightly inside his/her lane. It is worth noting how this feature, which is an adaptive lane keeping, is thus produced from more basic principles that reproduce human driving.

2. if manoeuvre “co-driver (1)” is a dangerous lane change and “co-driver (2)” is in the current lane; i.e., the driver is changing lane but there is a problem in longitudinal control. In this case “co-driver (2)” is used for HMI feedback, indicating to the driver he/she should better remain in the current lane.

3.3 RESULTS AND OTHER FEATURES EMERGED FROM FIELD OPERATIONAL TESTS

3.3.1 Test description

The Continuous Support application, i.e. the described Codriver combined with its Perception Platform upstream and the HMI downstream, has been tested extensively in two different sessions. One was required for a technical assessment, while the other regarded user-related issues [61].

The technical tests took place at the “Centro Sicurezza” private proving ground at CRF headquarters in Orbassano (TO), Italy (Figure 3.18). The vehicle was

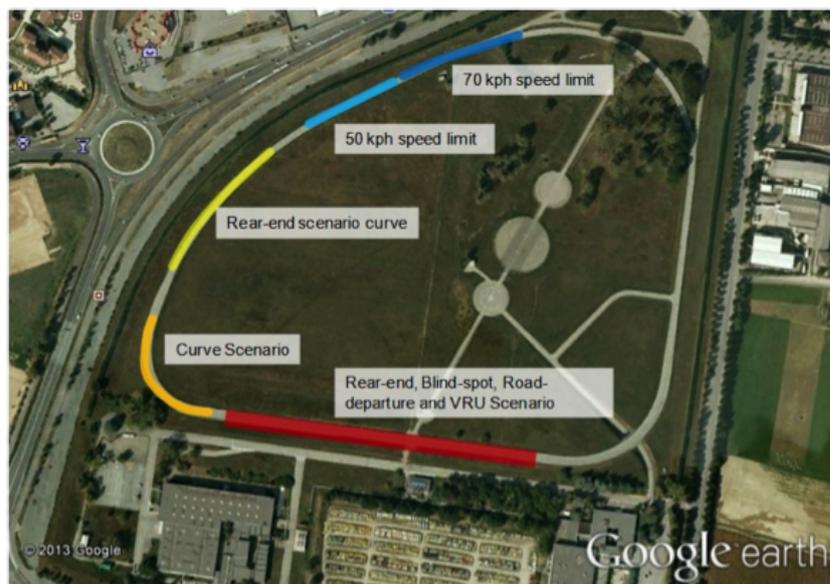


Figure 3.18: Centro Sicurezza in Orbassano (TO), Italy, the proving ground for the technical assessment tests [61]

conducted by professional drivers in predefined use cases:

- Rear-end collisions
- Blind-spot collisions
- Collisions with vulnerable road users
- Unintended lane/road departure accidents
- Traffic-rule violations

and objective performance was evaluated, e.g. number of false negatives and false positives, timing of the warnings, repeatability, etc. However, the results of these tests are subject to non-disclosure agreements and can not be detailed here. On the other hand, user-related tests were conducted on real roads, on a test track (Figure 3.19) including urban, extra-urban and motorway scenarios, 53 km long, which took between 40 and 45 minutes to be completed. 24 drivers (employees

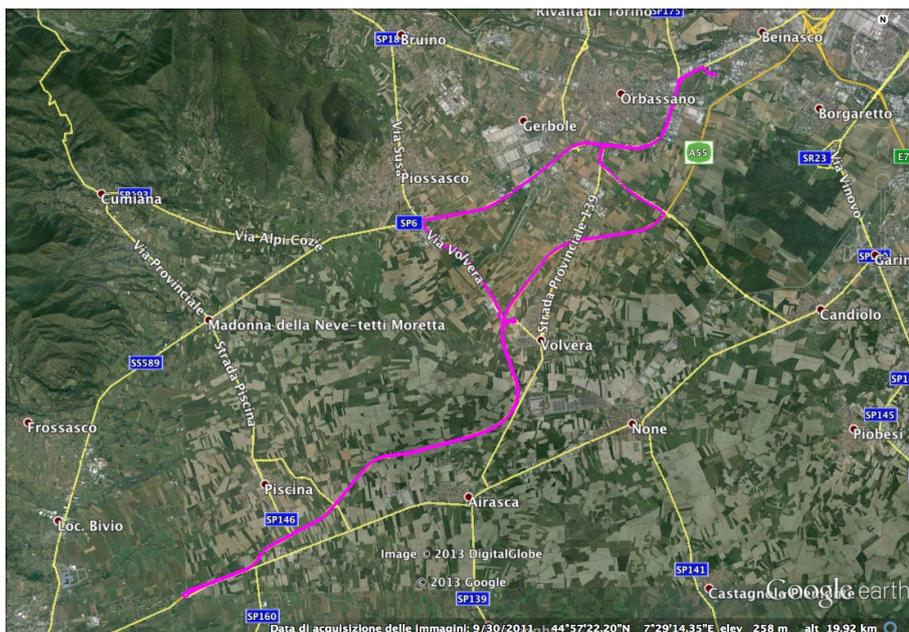


Figure 3.19: Test track around Orbassano (TO), Italy for the user-related assessment tests, comprehensive of urban, extra-urban and motorway environments

at CRF not involved in the interactive project) of different age and gender (Table 3.1) have driven on the test track twice, with the system always on, but the HMI

switched once off and once on (in a random order).

Before the tests, all the drivers were informed that the objective was to evaluate the performance of the system and not theirs, and that all data collected would be anonymous. The drivers were also instructed to drive as normal as possible and ask for whatever doubts or questions they might have during the test. Before the usage of the system, a brief explanation was given them, too.

During the tests, the drivers were observed by means of an in-car observation method (Wiener Fahrprobe method [49]) by two observers, one (the *coding observer*) studying standardised variables such as speed behaviour, yielding behaviour, lane changes and interaction with other road users, and the other carrying out other observations such as conflicts, communication and other special events. The main objectives of the observation were to determine whether the driver's behaviour changes when the system is switched on and off, his/her workload is relieved by the application or increased, and he/she trusts and accepts the intervention of the application.

At the end of the test, questionnaires were issued to the drivers to assess their comprehension of and experiences with the system, their subjective workload, the usefulness and satisfaction of the system, perceived benefits of the system, as well as willingness to have and pay for the system.

In total, the system has been tested for more than 35 hours and 2500 km. The

| | 45-64 | 25-44 | Total |
|--------|-------|-------|-------|
| Male | 6 | 7 | 13 |
| Female | 2 | 9 | 11 |
| Total | 8 | 16 | 24 |

Table 3.1: Distribution of age and genders among test drivers for the interactIVe project[61]

results of the tests are summarised in next paragraph.

3.3.2 *Observed behaviour of the Codriver*

3.3.2.1 *Inference of intentions*

A relevant aspect of the Codriver regards the inference of driver's intentions, which has been analysed on logged data from the tests. The inference is achieved by matching the observed behaviour of the driver with sensory-motor hypotheses of another agent. In other words, the Codriver observes driver behaviour and interprets it in terms of its own internal states that would produce it. This is the reason why, for the effectiveness of the Codriver, it is necessary to design it with a sensory-motor architecture as human-like as possible, so that its outputs will resemble human behaviour and motivations.

A perfect matching between the two is obviously not feasible: the hypotheses of the Codriver are simplified and in a limited number, which may result in missing or approximating the real case. For instance, in Figure 3.17 the observed driver lateral control is between trajectory 7 and 6. However the former is a borderline in-lane manoeuvre, while the latter definitely constitutes a lane change: the two options are qualitatively different, and the Codriver switches the driver control onto trajectory 7 because it is closer, but also because it is more intrinsically plausible. In fact, it is useful to recall that equation (3.29) describes how each manoeuvre is assigned a penalty expression, whose last term is its likelihood. At this point, it could both happen that the driver's intended trajectory is really somewhere between the two, so that the switching produces an approximation, but also that measurement noise affects the accuracy of Codriver hypotheses. In the showed case, analysing the following frames it was clear that the intended trajectory laid between 7 and 8 in reality: the plausibility term combined with the discrete set of hypotheses worked properly, and even helped to improve the signal-to-noise ratio thanks to behavioural discretisation. However, this could also cause inaccuracy of the inference in other cases, so that denser discretisation could be required, but this should be also accompanied by an improvement in the accuracy of the perception platform, to avoid the occurrence of very accurate hypotheses made on inaccurate premises.

Differently from final lateral displacement, there are other states for which only one hypothesis is made, e.g. the duration T of FreeFlow manoeuvres, the time T^* in equation (3.19) for ClearObject manoeuvres, time pressure w_T for FollowObject

(3.16) manoeuvres and T and the longitudinal motor primitive SA used in the ClearObject function (3.22) etc. Apart from simplifying the problem, this is done also because the dimensionality of the cognitive system states is much greater than the motor output space. For instance, considering the LD primitive (3.13), there are different combinations of $s_{n,T}$ and T which produce the same $j_{\Delta}(0)$. These represent osculating trajectories that cannot be discriminated in the short-term observations possible for real-time applications. Thus, only one plausible value for T is used in the application, while this method could be improved e.g. using multiple time-scales for fast and slow varying states. Generally, short-time observations of the motor output (i.e. driver jerks) discriminate easily between manifolds in the state spaces (i.e. the resulting manoeuvre), but the entire manifold of possible manoeuvres projects onto very similar motor output, and would require much longer observations in order to distinguish states.

Finally, the sensory-motor system is not complete, and it lacks of perception-action loops at each layer of the ECOM architecture. For example, there is not a built-in behaviour for navigating around an object. This functionality may be obtained with a sequence of three lower-level behaviours such as, i.e. change lane (LD primitive), free flow until the object is passed (FF), and lane change back (LD). However, this sequence is not implemented at the higher level, so that the Codriver is not able to conceive overtaking manoeuvres, even though it can understand the single phases of the manoeuvre. Moreover, in the case that the third phase starts before having completely passed the obstacle, the Codriver can only interpret the situation in terms of the motor plans it is aware of, i.e. follow object (FO) and clear object (CO). The former involves returning to the lane behind the vehicle, which would be evaluated as dangerous, while the latter would mean maintaining lateral safety clearance, probably resulting in a lane change delay. In any case, the Codriver never forms the behavioural hypothesis “returning to the lane in front of the obstacle”, because no such behaviour is represented in its sensory-motor system. The described situation has been observed several times, but interpreting the situation in terms of either clearing or following the obstacle anyway produces the warning to “keep clear”. In this way, the system is somehow robust to the lack of complex motor plans, which can be decomposed into simpler units of which the system is aware. However, it has been already emphasised that the behavioural architecture would allow to easily extend and scale the system,

adding possible complex manoeuvres such as overtaking without changing the design and the interfaces of the application.

3.3.2.2 *Behaviours from basic principles*

It was described how adaptive lane keeping manoeuvres can be generated from basic principles, but this is not the only such case. In fact, a system that works by producing human-like behaviours is very likely to give birth to complex behaviours that have not been programmed explicitly. Some of these cases have been observed during the tests, while the most relevant case is probably that depicted in Figure 3.17. The car remains in the right lane at the very beginning of a change of lane. Nonetheless, the Codriver uses its motor imagery to predict its own and other road users behaviour, and concludes that the car, invading ego vehicle lane, will block manoeuvres from 5 to 10, which will require braking, with the only available alternative being changing to the right lane. In subjective human evaluation, the warning produced by the system was indeed appreciated, being both appropriate and anticipated, giving the impression that the system could understand the road situation and correctly anticipate its outcomes.

3.3.2.3 *Limitations*

During the tests, some limitations of the system have also raised, the most relevant being noise and inaccuracy of the perception platform upstream the Codriver. For example, it has been observed that errors in the measurements of the lateral speed of other vehicles can cause false warnings, inducing the Codriver think they will invade ego vehicle lane.

Another limitation comes from incompleteness of the sensorial data. For example, knowing only the geometry of the host vehicle lane, and not that of the intersecting lanes, caused some false alarms in which vehicles moving to other lanes could not be predicted accurately. A typical example is the entrance into a roundabout, where the vehicle exiting in the opposite lane may seem again to invade the ego vehicle lane. One of these false warnings occurred during the user tests.

Other limitations are instead due to the Codriver architecture, i.e. to its incompleteness and to the simplifications of the sensory-motor system. For example, the FollowObject manoeuvre assumes that the obstacle is moving with only a

little longitudinal acceleration, while no information about obstacle acceleration is provided. If this is true, then the small errors due to the model assumptions in equation (3.17) are corrected by the receding horizon scheme. However, in the technical tests it happened that the obstacle hard braked, so that the Codriver was continuously underestimating the deceleration needed by the FO manoeuvre, resulting in a late warning. A better perception and tracking of the objects combined with improved motor primitives could of course fix this limitation.

3.3.3 *User response on the Continuous Support function*

Analysing then the results of the entire Continuous Support application, a summary of the results is here reported. For further details, the reader can refer to Fahrenkrog et al., 2013 [61], i.e. the final report of the subproject 7 of interactIVe, in charge for test and evaluation of the developed applications.

Looking first at drivers' behaviour with respect to speed limits, it can be said that they did not change driving style with and without the system HMI, and there were no significant changes in the number and in the length of the warnings (it is useful to remember that the system was active in both cases, so that the warnings were always issued by the Codriver, by only during one test these activated also the HMI).

Sensible changes happened instead in the velocities used for curves, showing the effectiveness of the system: roundabouts were travelled at significantly lower velocity (p-value $p \leq 0.05$) when the warnings were on. Furthermore, driving too fast according to the situation was observed significantly less often while the warning system was active.

The same can be said also for lateral dynamics, where driving too close to lane/road boundaries and dangerous lane changes were observed statistically significantly less often when the system warned the drivers about the risk. Finally, the test persons chose a wrong lane when driving through an intersection or roundabout less frequently when driving with the system active.

On the other hand, however, slightly later speed adaptations were observed before intersections and obstacles with the active system. In addition, statistically significantly more errors regarding dangerous distance to the side were observed

with the system active, and only during driving with the system active it was observed that the test persons turned with too high speed. This possibly shows the tendency of the drivers to excessively relax when the system is on, relying on it and thus paying lower attention to what is happening.

Another issue concerns the tuning of the Codriver for collision warning, since the profiles of velocities and accelerations show that in some cases the warnings were late, so that the drivers had already started to brake when the warnings were issued. This can be avoided raising the jerk thresholds the warnings are released at, being careful not to increase also the number of false positives in that way.

In all the other aspects, no significant differences were detected in the behaviour with and without the HMI.

Finally, analysing the questionnaires it was clear that the system was perceived as useful by the drivers. They felt that the system would enhance safety, especially while overtaking in motorways thanks to the blind-spot monitoring, and help also in maintaining the speed below the limits, and thus also in avoiding fines. The users also appreciated that the system was not too invasive, with a reasonable number of warnings. In other words, the number of false positives was very low. Looking at the weaknesses of the system, they also required the elimination of those few false alarms which still occurred, and they noticed the delay at which some frontal collision warnings were issued.

However, the drivers mainly also asked for some improvements in the HMI. First, longer permanence of the visual explanation of the warnings was wished, since in some cases the driver could not realise why a warning was issued, and they also criticised the position of the cluster display, partially covered by the steering wheel. Furthermore, they suggested to adapt the strength of the seatbelt vibration proportionally to the collision risk level, and to eliminate this feedback for other situations such as excessive speed. On the other hand, they wished an additional haptic feedback for blind-spot warnings.

In the end, when the drivers were asked if they would pay to have this system on their cars, all of them answered affirmatively. This seems to confirm the overall impression of having obtained good results with a system people would considerably pay for, even if on the market it would be probably necessary to double those amounts to make it profitable.

FROM DRIVER WARNING TOWARDS AUTONOMOUS INTERVENTION: A THEORETICAL AND EXPERIMENTAL ANALYSIS OF COLLISION AVOIDANCE

| | | |
|-------|---|-----|
| 4.1 | Introduction | 74 |
| 4.2 | Driver limits: analysis of driver behaviour from driving simulator data | 75 |
| 4.2.1 | The driving simulator | 75 |
| 4.2.2 | Test description | 77 |
| 4.2.3 | Results and discussion | 80 |
| 4.3 | Manoeuvres towards vehicle limits: more accurate OCPs and suitable models | 89 |
| 4.3.1 | Optimal control formulation for obstacle avoidance | 89 |
| 4.3.2 | OCP formulation for brake-to-avoid manoeuvres | 89 |
| 4.3.3 | OCP formulation for steer-to-avoid manoeuvres | 92 |
| 4.4 | Analysis of OCP manoeuvres with different models and formulations | 97 |
| 4.4.1 | Codriver and more complex OCP manoeuvres for warning | 97 |
| 4.4.2 | Strategies for obstacle avoidance | 101 |

4.1 INTRODUCTION

In the previous chapter, the Codriver was described together with its architecture. It has been widely explained how it was designed resembling human states of mind and human motion planning criteria, which are necessary to identify driver intentions. Some limits of the Codriver have already been emphasised when discussing the results of the interactIVe project, but here some additional considerations will be drawn from a different point of view, i.e. possible other applications the Codriver can be used for.

The Codriver has been conceived for driver warning, and has proved to be a suitable way to achieve this objective and increase road safety, but it could also happen that the driver does not react to the warnings, and an autonomous intervention of the system would be required. The objective is now to understand if the Codriver can be extended to plan this intervention. Applying the concept to the interactIVe project, the idea is to see if the Codriver could be used for a unique application both for the SECONDS and INCA subprojects, to handle both driver intention inference and eventually the planning of evasive manoeuvres to be executed when the risk level raises.

This discussion focuses only obstacle avoidance, neglecting all the other threats which the Codriver addresses. For this purpose, in this chapter it will emerge that the manoeuvres planned by the Codriver, without any modification, can not be directly used for this purpose, for two main reasons:

- The plans of the Codriver are the result of extremely simplified OCPs modelling only kinematic relations, which was necessary to obtain closed-form solutions for fast evaluation, while for autonomous intervention the vehicle dynamics and their physical limits must be taken into account
- The plans of the Codriver are humanlike, with limited actuation rates, while autonomous intervention could exploit faster and more accurate actuation

On the other hand, they can provide good guess functions to feed more complex OCPs and accelerate their convergence.

In this chapter, an analysis of humanlike driving is carried on in section 4.2, based on real data collected during tests on a Driving simulator for steering and braking manoeuvres, which confirmed some results of traditional research on

driver behaviour (Summala, 1981 [129], McGehee et al., 1999 [106]). These data can be used to set the thresholds for risk assessment and driver warning, but they give no idea of the margins left for autonomous intervention. To do this, more accurate models must be developed, to be included in more complex OCPs which push the vehicle to its limits (Section 4.3). Finally, the plans of the Codriver are compared with the limit manoeuvres coming from these accurate OCPs, to derive indications on which simplifications can be done and which detail is necessary for autonomous intervention systems (Section 4.4). Furthermore, the detailed OCPs are there used to compare emergency brake-to-avoid and steer-to-avoid manoeuvres and derive information on which of the two is more suitable depending on the scenario. Proving that steering manoeuvres are more effective at high relative velocity, these are pushed to the limits investigating the improvements achievable by means of four-wheel-steering.

4.2 DRIVER LIMITS: ANALYSIS OF DRIVER BEHAVIOUR FROM DRIVING SIMULATOR DATA

4.2.1 *The driving simulator*

In previous section it was anticipated that a driving simulator was used to collect data on drivers' behaviour during braking and steering manoeuvres. In particular, the driving simulator (Zendri, 2012 [144]) used has been that developed within the Mechatronics Group [16] at the Department of Industrial Engineering of the University of Trento (Figure 4.1).

The simulator is composed of hardware components which resemble a real car cabin, and a PC which manages all the components. The cabin is reproduced using a real car seat, a gearbox stick featuring both manual and sequential behaviour, a pedal box and a dashboard, all provided by Dynamotion [4]. In the dashboard, a steering system is hosted, i.e. a TRW Active Steering Wheel System 2. Three 50" plasma monitors surround the driver and resemble what he/she could see from the windscreen, the lateral windows and the rear mirrors. An audio system completes the simulator, reproducing engine and environment sounds.

The management of the simulator is handled by a PC equipped with a dual core processor Intel Core i7-2600 CPU @ 3.40GHz, running a 32 bit Windows 7 OS which exploits 3 of 8 available GB RAM. Moreover, a CAN board (Adlink



Figure 4.1: The driving simulator of the University of Trento

7841) has been installed to communicate with the simulator steering wheel. Here the OKTAL SCANeR software is installed, which receives driver inputs from the steering wheel, the pedals and the gearbox, and accordingly commands the graphics and the sound thanks to interval vehicle and road models, and also the traffic.

The steering wheel position is directly sent to the software via CAN-bus, measured by a built-in encoder. On the other hand, the pedal positions are sensed by Linear Variable Displacement Transducers (LVDT) and the requested gear is measured by Hall Effect sensors, and their values are acquired using a cheap micro-controller, an Arduino Mega 2560, which then outputs them to the SCANeR software.

The software also allows wide customisation, letting the user develop his own scenarios, and developing his own modules for logging, etc. Thus, different modules have been developed to represent sensors equipping real vehicles, to acquire vehicle, obstacle and road states and use them both for logging and for real-time exploitation (i.e. sending them to another custom module representing an ADAS). In this way, custom scenarios have been developed, and simulation data have been logged for offline analysis.

4.2.2 Test description

4.2.2.1 Braking scenario

In order to derive indications on how drivers behave in emergency manoeuvres, two different tests have been set up on the driving simulator: one for obstacle avoidance by steering, and one for braking.

The braking scenario takes place in a straight road. The driver is asked to follow a vehicle ahead at a distance as close as possible, but which he/she still perceives as safe in the case the obstacle reduces its speed. The driver is not allowed to overtake it or anyway use the steering wheel to avoid it. The vehicle ahead has variable speed, performing 5 braking manoeuvres with different constant decelerations and different final velocities. The braking manoeuvres are activated when the driver reaches a given time-headway, i.e. the time required to reach obstacle current position at present constant speed, in order to be sure that the ego vehicle is close enough and a braking will be required. However, the given time headway could be too short for the driver, which may prefer remaining at a safer combination of relative speed and distance, thus never activating the braking manoeuvre in the obstacle. To avoid this inconvenience, a more complex procedure has been adopted to activate the braking: when ego-vehicle reaches a high time headway $t_{h,a} = 2.5 \text{ s}$, a very safe condition which is always exceeded by the drivers, the algorithm is ready to make the obstacle brake when a shorter limit time headway $t_{h,b0} < t_{h,a}$ is reached. However, if this value is not reached within some seconds, this value is iteratively raised to values $t_{h,b}$ so that $t_{h,b0} < t_{h,b} < t_{h,a}$, until $t_{h,b}$ meets the actual time headway of the vehicle. When this happens, the obstacle starts braking and the the ego vehicle is forced to brake as well. The velocity profile of the obstacle is showed in Figure 4.2. Namely, the obstacle reaches a constant speed $u_{i,1} = 50 \text{ km/h}$ and performs a first braking with deceleration $a_1 = -0.5g$ until it stops at $u_{f,1} = 0 \text{ km/h}$, then it remains a while at this final velocity and accelerates again to $u_{i,2}$. Four other braking manoeuvres are performed with the same procedure, i.e. from $u_{i,k}$ a generic deceleration a_k is applied until a generic final velocity $u_{f,k}$ is reached, which is kept for a while, and then left to reach again $u_{i,k+1}$. The values of the decelerations and the final velocities are reported in Table 4.1.

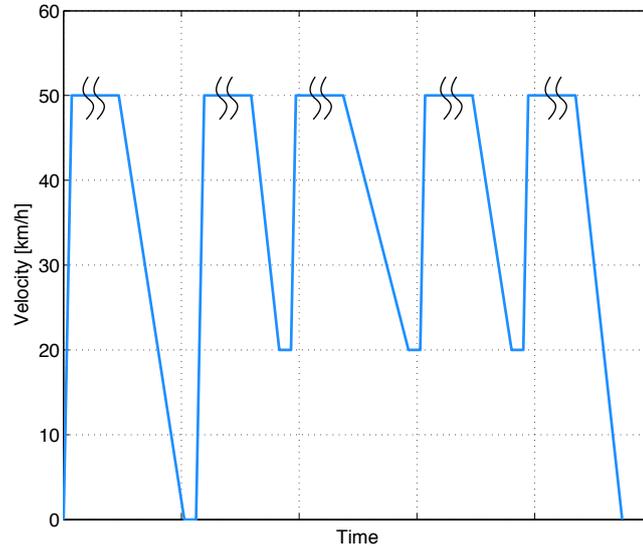


Figure 4.2: Velocity profile of the obstacle in the brake-to-avoid scenario

| Braking manoeuvre k | Initial velocity $u_{i,k}$ [km/h] | Deceleration a_k [g] | Final velocity $u_{f,k}$ [km/h] |
|------------------------------|--|-------------------------------|--|
| 1 | 50 | 0.5 | 0 |
| 2 | 50 | 0.7 | 20 |
| 3 | 50 | 0.3 | 20 |
| 4 | 50 | 0.5 | 20 |
| 5 | 50 | 0.7 | 0 |

Table 4.1: Characteristics of braking manoeuvres on the driving simulator

4.2.2.2 Steering scenario

The steering manoeuvre takes place again in a straight two-lane road, with a fixed obstacle $w_o = 2.5 \text{ m}$ wide which occupies the lane of the ego vehicle, at a distance $L_o = 500 \text{ m}$ ahead (Figure 4.3). The driver is asked now to ap-

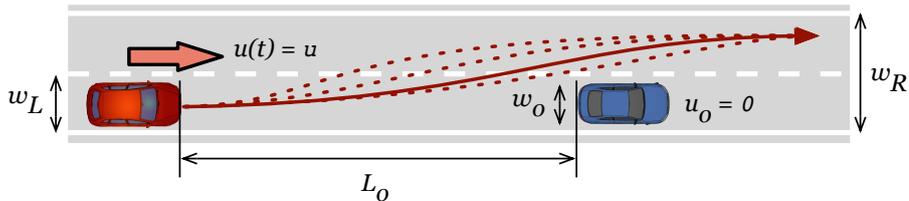


Figure 4.3: The steer-to-avoid scenario on the driving simulator

proach the obstacle at a constant speed, between 60 and 70 km/h , which can be easily achieved within the L_o distance, and then to avoid it at the instant he judges as the last possible, by means of only steering, without braking. However, the steering manoeuvre is constrained to remain within road boundaries: the driver is asked to change lane and remain in it, until the simulation is stopped.

4.2.2.3 Test procedure

Given the scenarios described in previous paragraphs, 20 drivers (students and employees of the University of Trento) have been chosen according to the same criteria of different age and gender used in the interactIVe project (Table 3.1). In this case the repartition between the categories has been more rigorous, featuring 10 male and 10 female, 5 younger than 40 and 5 older for each gender, as reported in Table 4.2. Each driver was required to perform this sequence of tests:

1. 5 steering scenario tests
2. 1 braking scenario test, with $t_{h,b0} = 1.0 \text{ s}$
3. 5 steering scenario tests

| | 40-60 | 19-39 | Total |
|---------------|-------|-------|-------|
| Male | 5 | 5 | 10 |
| Female | 5 | 5 | 10 |
| Total | 10 | 10 | 20 |

Table 4.2: Distribution of age and genders among test drivers for the driving simulator

4. 1 braking scenario test, with $t_{h,b0} = 1.6 s$

In this way, each driver performed in the end 10 braking manoeuvres and 10 steering manoeuvres. The sequence of tests lasted around 25 min for each driver, during which several variables were logged by suitably developed modules in the driving simulator. These values have been stored in anonymous form, where each driver was assigned an ID according to Table 4.3. The results are summarised in

| | 40-60 | 19-39 |
|---------------|---------|---------|
| Male | 1 → 5 | 6 → 10 |
| Female | 11 → 15 | 16 → 20 |

Table 4.3: Driver anonymous IDs assignation, based on age and gender

next section.

4.2.3 *Results and discussion*

4.2.3.1 *Braking tests*

In this paragraph, some key variables of the braking tests will be showed and discussed. First, it is useful to look at the time headways at which the drivers started braking, distinguishing between the scenario where the iterative braking procedure started from $t_{h,b0} = 1.0 s$ (Figure 4.4) and that where it was $t_{h,b0} = 1.6 s$ (Figure 4.5). In both scenarios the behavioural difference between men and women is evident: men tend to follow other vehicles much closer than women regardless of the age, at time headways $1 \leq t_{h,M} \leq 1.4 s$ while women are positioned at $1 \leq t_{h,M} \leq 2.6 s$. This allows to derive a first important conclusion:

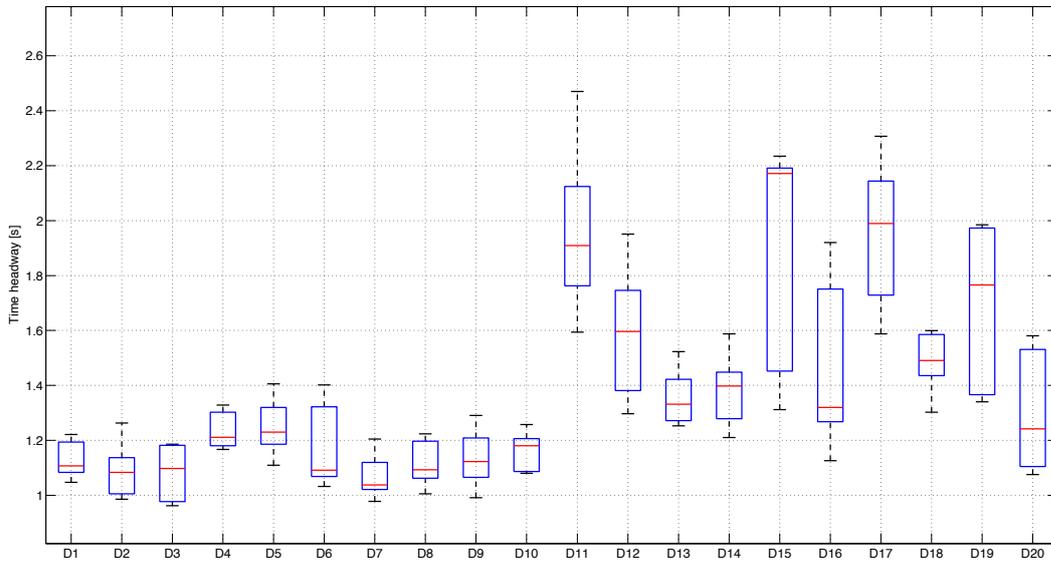


Figure 4.4: Distribution of time headways at which each driver started the braking manoeuvre, when $t_{h,b0} = 1.0$ s

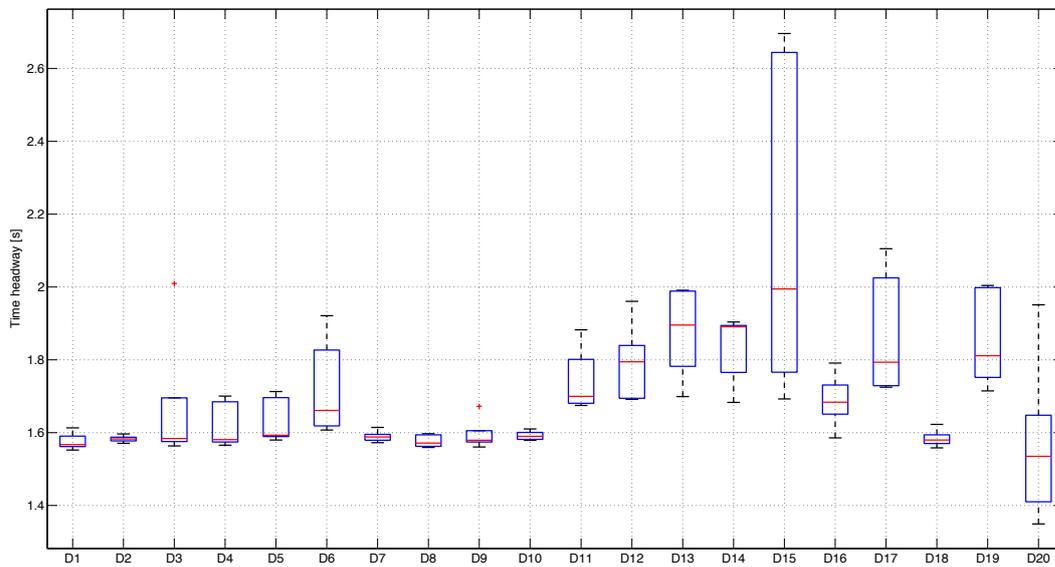


Figure 4.5: Distribution of time headways at which each driver started the braking manoeuvre, when $t_{h,b0} = 1.6$ s

time headway is not a good parameter to trigger ADAS warnings and interventions for obstacles, since situations around $t_{h,M} = 1.6$ s would be perceived as dangerous by women, while man would be annoyed by warnings in those situations.

Analysing other characteristics of driver behaviour, it is interesting to show the distribution of driver maximum applied pedal positions (Figure 4.6) and velocities the pedals were pressed with (Figure 4.7), which are proportional respectively to vehicle longitudinal accelerations and jerks. Under the hypothesis that

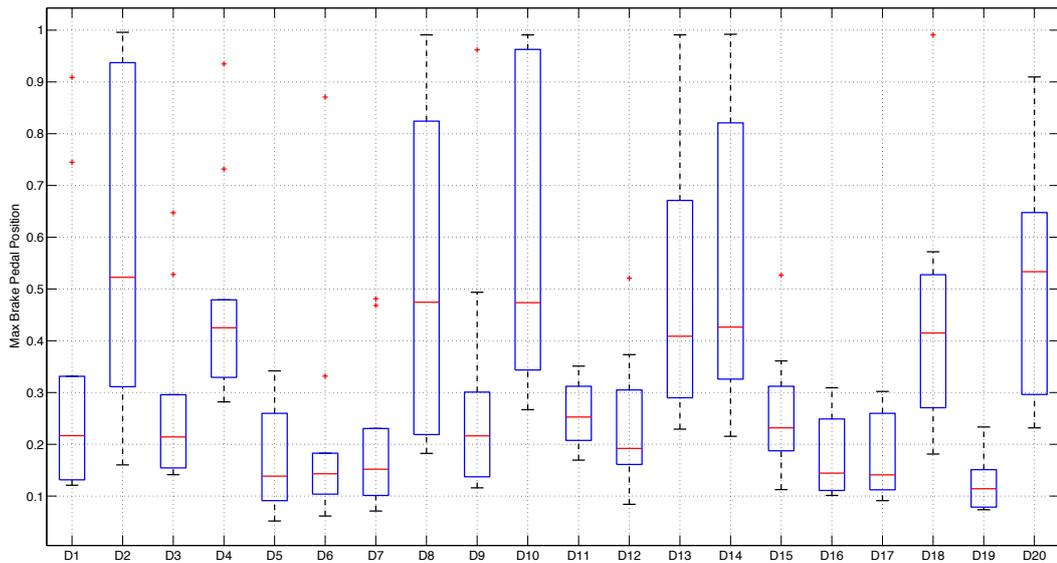


Figure 4.6: Distribution of maximum values for braking pedal positions

approximately $0.8 - 1$ g decelerations can be obtained with the braking pedal completely pressed (i.e. $p_b = 1$), these values of braking decelerations and jerks are consistent with others found in literature, e.g. in Brännström et al., 2012 [42].

Looking at the distributions, both for the accelerations and jerks there are not significant differences between genders and ages, apart from a tendency in younger women to use low jerks. However, it is quite clear that almost all the drivers use pedal velocities $j_p \leq 2 - 2.5$ s^{-1} , apart from the exceptions of drivers 2, 10 and 18 which use a wider range of velocities. These cases, however, can be also seen as exceptional emergency manoeuvres or not reasonable actuations, probably due to the characteristics of the test, conducted on a simulator instead of

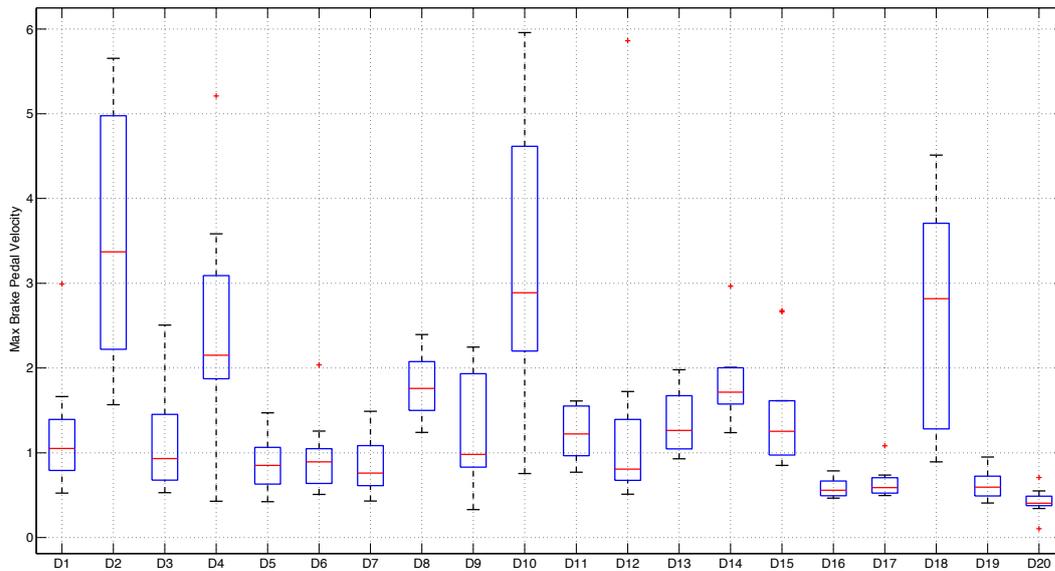


Figure 4.7: Distribution of maximum values for braking pedal velocities

the a real vehicle. Anyway, this allows to state that *longitudinal jerks can be used as more robust parameters for risk assessment* (e.g. in accordance to Viviani, 1995, [135]) since they are limited within well specified boundaries, and avoidance manoeuvres requiring higher values can be classified as not humanlike.

The distribution of all the longitudinal jerk values is showed in Figure 4.8. Looking at the whole dataset it is even more clear that normal drivers barely exceed the value of $j_p = 2.5 \text{ s}^{-1}$ even for emergency manoeuvres such as those of the simulated scenario, meaning that drivers may not be prompt or even capable of acting with higher rates. This could be used to assess that *manoeuvres requiring higher jerks may be not feasible by a driver*. However, this does not mean that harder braking manoeuvres would not be feasible: if a faster actuation was available, there may be still the margin for an autonomous intervention system to perform them before reaching the physical limits of the vehicle.

This idea can be used to build a risk assessment method: given an ADAS which continuously plans manoeuvres to handle current scenario in a safe way, risk levels could be assigned basing on the jerks required by these manoeuvres, e.g. as described in section 5.4, where this principle is proposed for the autonomous obstacle avoidance ADAS. For warning purposes, assuming that drivers are not

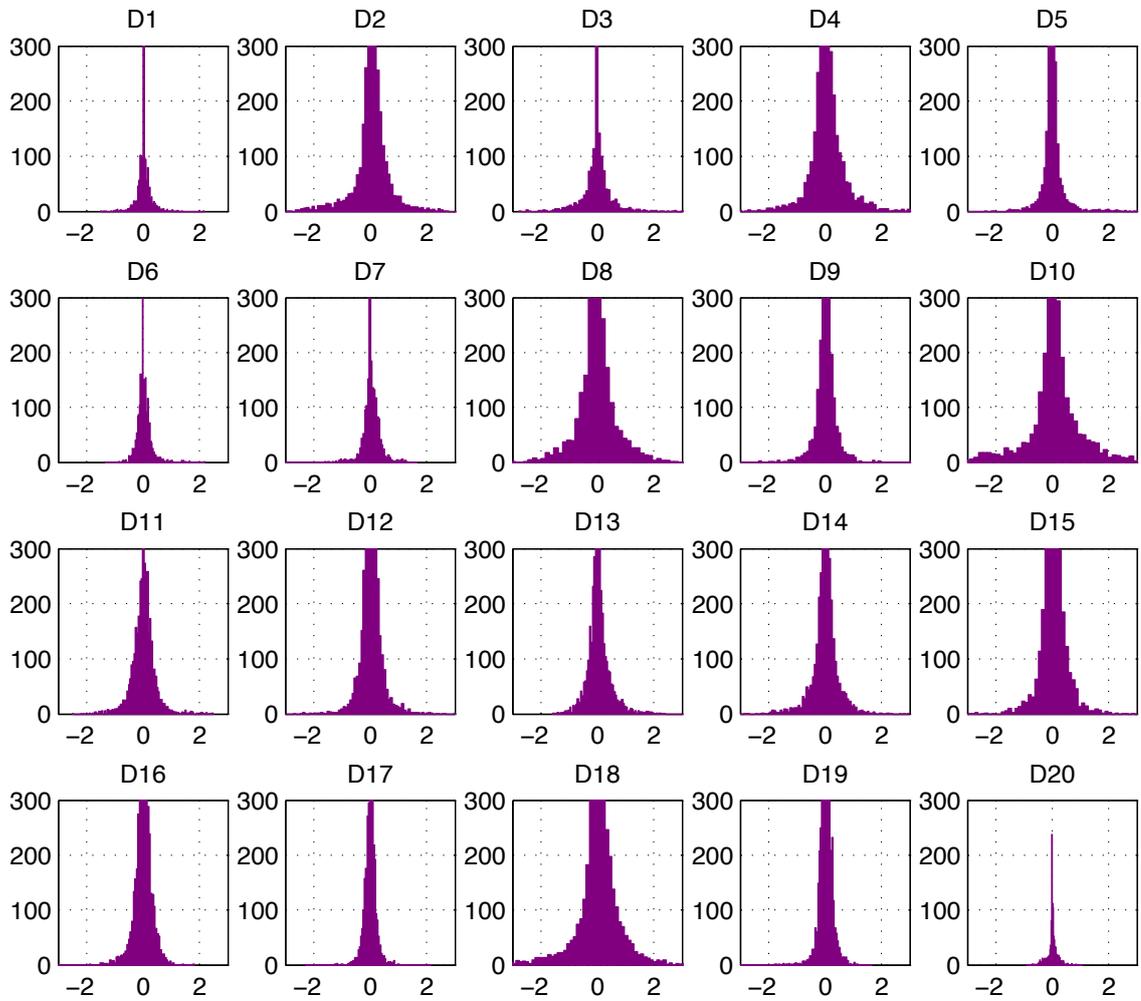


Figure 4.8: Distribution of all the braking pedal velocities for each driver

able to apply longitudinal jerks higher than $j_{p,Max} = 2.5 \text{ s}^{-1}$, warnings could be issued if safe manoeuvres require jerks slightly slower, e.g. $j_{p,Warn} = 2 \text{ s}^{-1}$, to give them time to react and still be able of avoid a collision, which is actually the value used by the Codriver. If the situation worsens to the point that jerks higher than $j_{p,Max}$ are required, autonomous intervention could be triggered, actuating harder manoeuvres which a driver would not be able to perform, but still in the feasibility range for faster actuators, and still within the physical limits of the vehicle.

4.2.3.2 Steering tests

Analysing then the steering manoeuvres, similar conclusions can be derived. Even if for the steering the time headways showed lower influence of the gender of the driver (Figure 4.9). Even in this case, however, the steering wheel angular

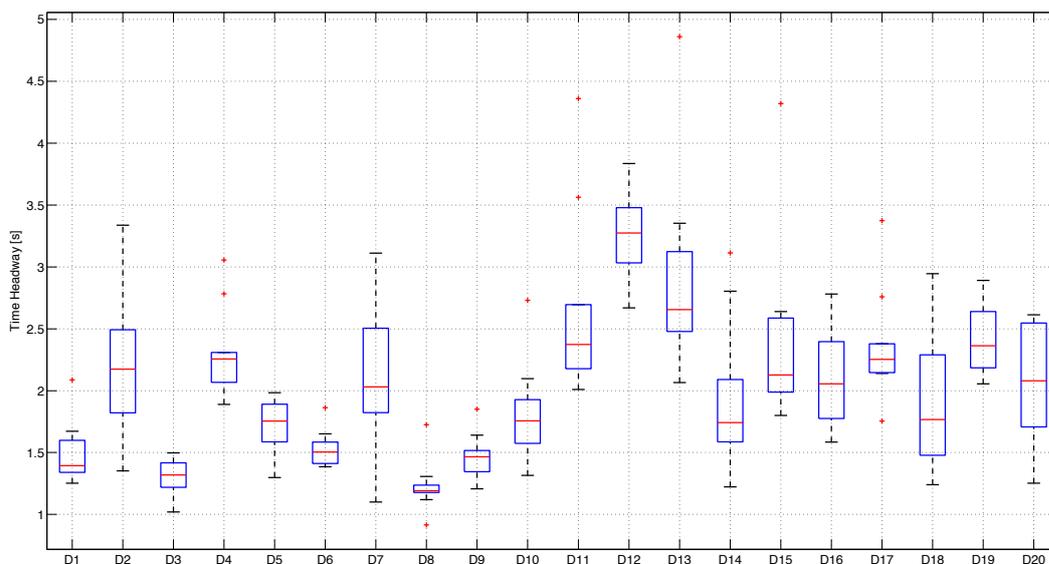


Figure 4.9: Distribution of time headways at which each driver started the steering manoeuvre

velocity (Figure 4.11), which is proportional to lateral jerk and is thus analogue to pedal velocity, can be used to set proper thresholds for ADAS triggering. In this case, it is reasonable to assume values of $j_{\delta} > 500 - 700 \text{ deg/s}$ as not feasible by human drivers, so that an ADAS should issue the warnings at a lower threshold,

when safe manoeuvres require velocities higher than $j_{\delta,Warn} > 400$ deg/s. These maximum values are actually a bit higher than the thresholds used in the already cited work of Brännström [42], which uses a value of $j_{\delta,Max} > 400$ deg/s. On the other hand the maximum value of the steering wheel angle is lower than 180 deg, while Brännström [42] considers the maximum available angle, i.e. 720 deg. The

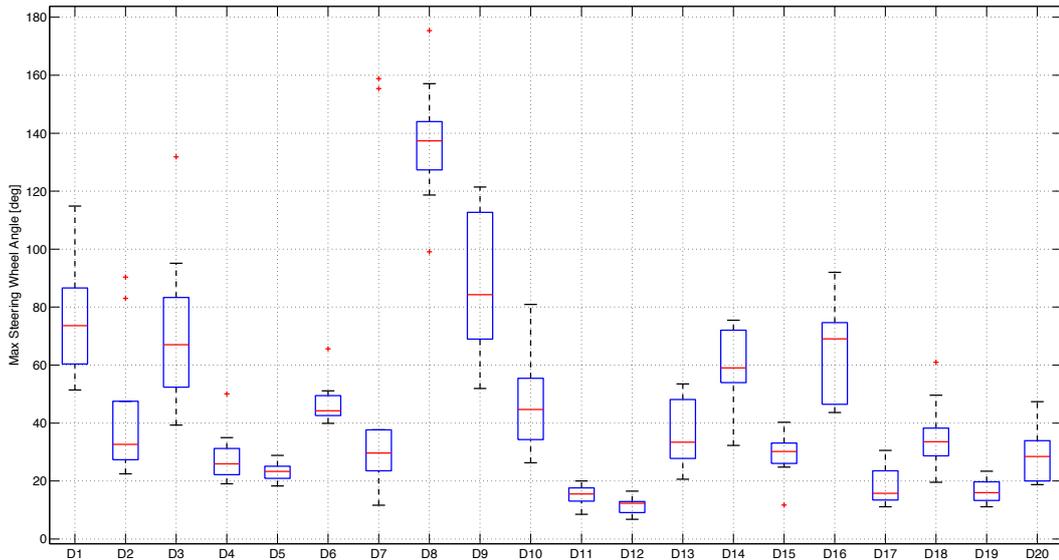


Figure 4.10: Distribution of maximum values for steering wheel angles

distributions of the entire set of steering wheel velocities used by the drivers is finally showed in Figure 4.12, where it is clear that for the great majority of the time, steering velocities higher than $j_{\delta} = 500$ deg/s are not required.

4.2.3.3 Further discussion

The proposed thresholds for warnings and interventions consider manoeuvres requiring only braking or only steering. However, when a braking manoeuvre for obstacle avoidance requires too high longitudinal jerks, it is not correct to directly issue a warning, since steering manoeuvres may be still perfectly feasible by the driver. In other situations the contrary may happen as well, i.e. steering manoeuvres not feasible while there is still time to start a braking manoeuvre. Furthermore, there is a third category of manoeuvres which combines both steering and braking which should be taken into account, even if in this case

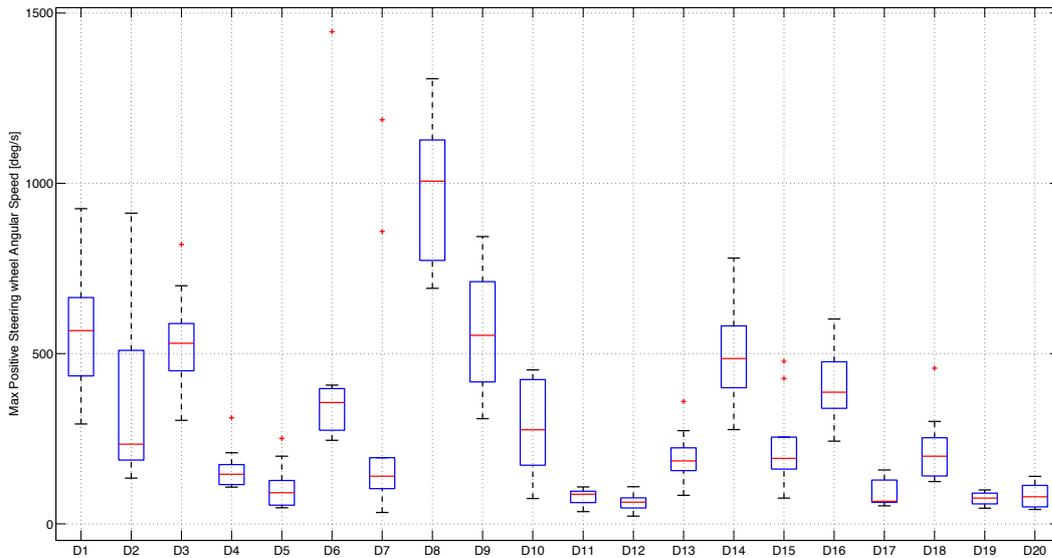


Figure 4.11: Distribution of maximum values for steering wheel angular velocities

longitudinal and lateral dynamics cannot be decoupled, and the complexity of the models increases.

In general, in a given situation with an obstacle in front of ego vehicle, to avoid a collision the driver can choose to brake, steer, or combine the two. One or more of these options may be precluded by the characteristics of the scenario, depending on relative velocity, relative position, road geometry, traffic conditions etc.

A detailed analysis of this situation will be given in section 4.4, but it will be necessary first to illustrate the models needed for calculating limit manoeuvres, i.e. leaving the humanlike simplified manoeuvres of the Codriver for more complex considerations, which must take into account the vehicle dynamics. These manoeuvres may be not reachable by human drivers, both for their limits in the actuation and for their lack of knowledge of the complex dynamics of the vehicle, but may be still planned by an optimal control problem and actuated by superior automatic equipment.

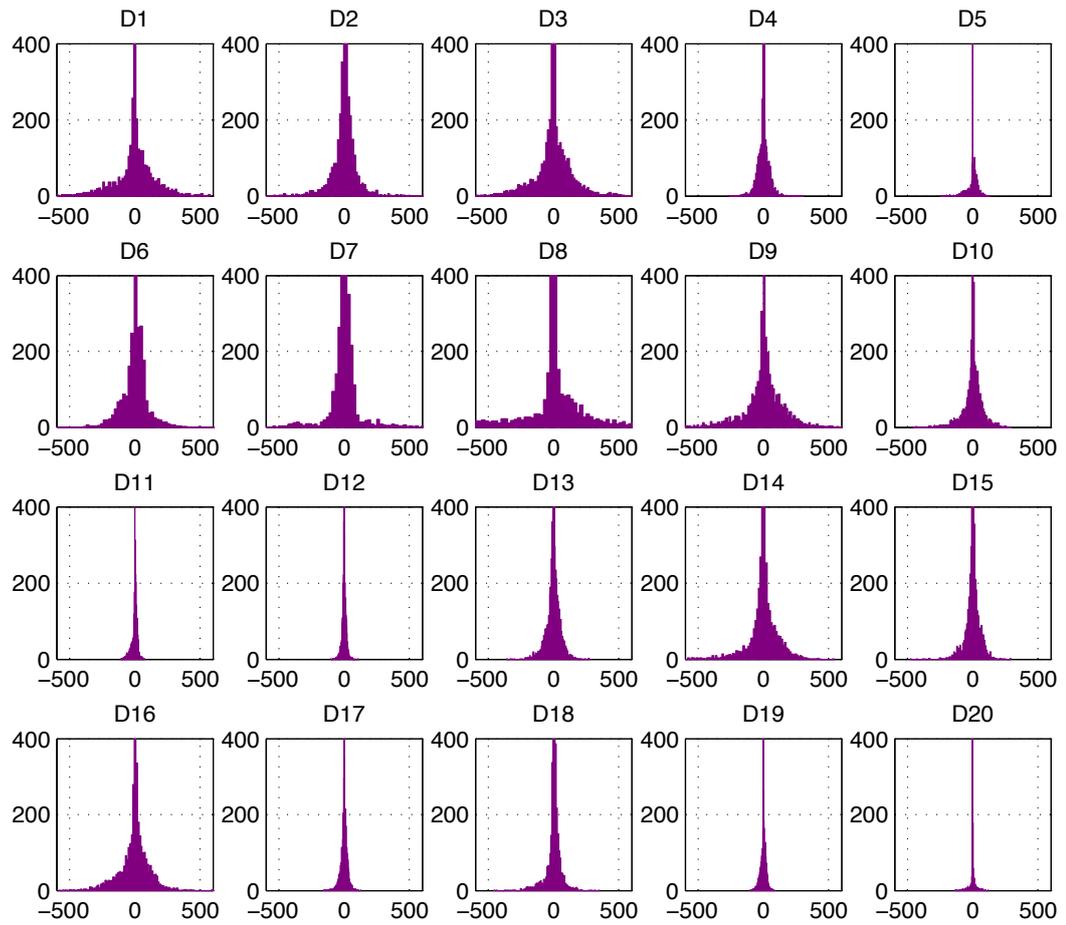


Figure 4.12: Distribution of all the steering wheel angular velocities for each driver

4.3 MANOEUVRES TOWARDS VEHICLE LIMITS: MORE ACCURATE OCPS AND SUITABLE MODELS

4.3.1 *Optimal control formulation for obstacle avoidance*

It has been explained that human drivers use optimised controls while driving, and in particular they try to minimise the longitudinal and lateral jerks, along with other criteria depending on road, scenario and personal goals [52].

It has been also demonstrated how this planning strategy can be reproduced using optimal control, as it has been done developing the Codriver. However, optimal control can be used for more complex problems, taking into account constraints for variables and controls. In this way, it will be sure that the resulting manoeuvres, even if very severe, are physically feasible. In Appendix B, the entire formulation and the solving details are reported, while here the specific features of the formulation of the optimal control problem for avoidance manoeuvres will be presented.

In particular, braking manoeuvres and steering manoeuvres will be presented separately, together with the decoupled vehicle models of longitudinal and lateral dynamics they require.

4.3.2 *OCP formulation for brake-to-avoid manoeuvres*

4.3.2.1 *Vehicle model*

A first set of OCP problems is that for avoidance manoeuvres by braking. In this case, a simple bicycle model can be used, but which considers also the traction and braking forces in this case, so that the equations are a bit more complex than (3.3).

$$\begin{aligned}\frac{dx(t)}{dt} &= u(t) \\ m \frac{du(t)}{dt} &= F_{xf,T}(p(t)) + F_{xr,T}(p(t)) + F_{xf,B}(p(t)) + F_{xr,B}(p(t)) - k_v u^2(t) \\ \frac{dp(t)}{dt} &= j_p(t)\end{aligned}\tag{4.1}$$

Without considering longitudinal tyre sideslips, the expression for the traction and braking forces can be directly assumed as proportional to traction and braking torques according to wheel radius:

$$\begin{aligned}
F_{xf,T}(p(t)) &= \frac{T_{xf,T}(p(t))}{r_f} \\
F_{xr,T}(p(t)) &= \frac{T_{xr,T}(p(t))}{r_r} \\
F_{xf,B}(p(t)) &= \frac{T_{xf,B}(p(t))}{r_f} \\
F_{xr,B}(p(t)) &= \frac{T_{xr,B}(p(t))}{r_r}
\end{aligned} \tag{4.2}$$

The torques applied to the wheels are a function of pedal position $p(t) \in [-1, 1]$. Engine and brake maps can be used for this purpose, or simpler linear models such as the following example:

$$\begin{aligned}
T_{xf,T}(p(t)) &= k_{Tf} T_{T,Max} p(t) \\
T_{xr,T}(p(t)) &= k_{Tr} T_{T,Max} p(t) \\
T_{xf,B}(p(t)) &= k_{Bf} T_{B,Max} p(t) \\
T_{xr,B}(p(t)) &= k_{Br} T_{B,Max} p(t)
\end{aligned} \tag{4.3}$$

While the torque repartition coefficients for a common front traction vehicle can be:

$$\begin{aligned}
k_{Tf} &= 1 \text{ if } p(t) > 0; 0 \text{ otherwise} \\
k_{Tr} &= 0 \\
k_{Bf} &= 0.6 \text{ if } p(t) < 0; 0 \text{ otherwise} \\
k_{Br} &= 0.4 \text{ if } p(t) < 0; 0 \text{ otherwise}
\end{aligned} \tag{4.4}$$

To help the convergence of the system, these piecewise values should be regularised. In this way, this simple model can be used to describe the longitudinal dynamics of the vehicle, and can constitute the differential constraints of equation (B.1). In this case, the system is expressed in cartesian coordinates instead of the curvilinear coordinates of the Codriver, and pedal velocity $j_p(t)$ is once again the control input.

4.3.2.2 OCP formulation

For braking manoeuvres, the objective of the optimisation of (B.1) can be a unique weighted Mayer term which requires minimum time T :

$$J = w_T T(\zeta_T) \quad (4.5)$$

Manoeuvre time T is not a-priori known, since it depends on the controls which will be applied. To leave it free, an additional differential equation is written:

$$\dot{T} = 0 \quad (4.6)$$

simply meaning that T is a constant, and an auxiliary variable $\zeta(t) \in [0, 1]$ is introduced instead of t . ζ is a fraction of final time T , so that:

$$\frac{d\zeta(t)}{dt} = \frac{1}{T(\zeta)} \quad (4.7)$$

Differently from the Codriver formulation, in this case, constraints are set on state variables or other expressions. In particular, it can be required that tyre longitudinal forces do not exceed adherence limits, taking into account the friction coefficients between the tyre and the terrain μ_x , i.e.:

$$\begin{aligned} F_{xf,T}(t) + F_{xf,B}(t) &\leq \mu_{xf} N_f(t) \\ F_{xr,T}(t) + F_{xr,B}(t) &\leq \mu_{xr} N_r(t) \end{aligned} \quad (4.8)$$

where the vertical loads $N(t)$ can be defined to take into account also longitudinal load transfers.

If humanlike manoeuvres need to be simulated, constraints should be set also on the control $j_p(t)$, which should be bounded according to the thresholds e.g. found in Section 4.2.

Finally, to require braking manoeuvres it is necessary to set boundary conditions. Initial conditions must be fixed, since they describe current situation, which could be:

$$\begin{aligned} x(\zeta = 0) &= 0 \\ u(\zeta = 0) &= u_0 \\ p(\zeta = 0) &= \text{free} \\ T(\zeta = 0) &= \text{free} \end{aligned} \quad (4.9)$$

while final conditions describe the desired final state.

$$\begin{aligned}
 x(\zeta = 1) &= x_o - r_V - r_O & (4.10) \\
 u(\zeta = 1) &= 0 \\
 p(\zeta = 1) &= 0 \\
 T(\zeta = 1) &= \text{free}
 \end{aligned}$$

so that the vehicle will stop before the obstacle position x_o taking into account also the dimensions of the two agents.

Examples of manoeuvres resulting from this kind of OCP will be described in Section 4.4.

4.3.3 OCP formulation for steer-to-avoid manoeuvres

4.3.3.1 Vehicle model

For steering manoeuvres, the OCP formulation is slightly different, since other aspects rise in this case. In this case, the manoeuvre is similar to a single lane change (Figure 4.13). To handle this scenario, a vehicle model with constant

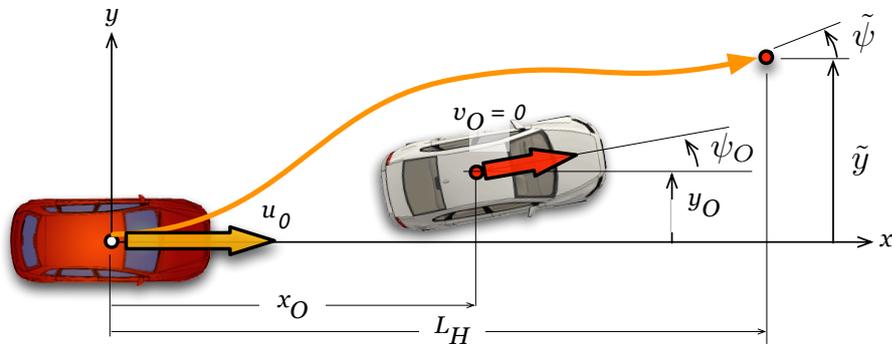


Figure 4.13: A general steer-to-avoid manoeuvre

forward velocity u_0 has been derived to be used in the optimal control formulation. The model (4.11) is once again a single track model, which this time includes tyre lateral force saturation, tyre linear dynamics (4.11c),(4.11d) and also first order approximation of the actuator (4.11e),(4.11f) dynamics (Figure 4.14).

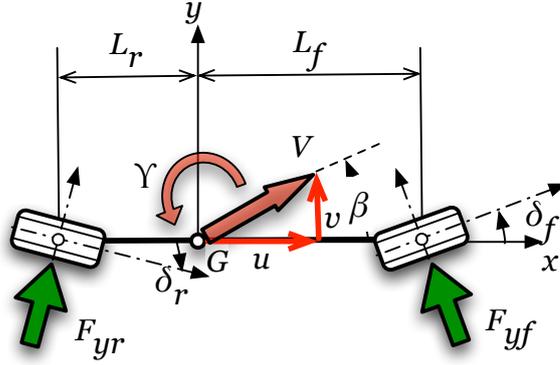


Figure 4.14: Bicycle model adopted in the Optimal Control formulation. Main geometrical parameters described.

$$m u_0 \left(\frac{d\beta(t)}{dt} + \gamma(t) \right) = F_{y_f}(t) + F_{y_r}(t) \quad (4.11a)$$

$$I_z \frac{d\gamma(t)}{dt} = L_f F_{y_f}(t) - L_r F_{y_r}(t) \quad (4.11b)$$

$$\frac{l_y}{u_0} \frac{dF_{y_f}(t)}{dt} = F_{y_f0}(t) - F_{y_f}(t) \quad (4.11c)$$

$$\frac{l_y}{u_0} \frac{dF_{y_r}(t)}{dt} = F_{y_r0}(t) - F_{y_r}(t) \quad (4.11d)$$

$$p_r \frac{d\delta_r(t)}{dt} = k_r \delta_r(t) - \delta_{r_c}(t) \quad (4.11e)$$

$$p_f \frac{d\delta_f(t)}{dt} = k_f \delta_f(t) - \delta_{f_c}(t) \quad (4.11f)$$

Absolute vehicle position and orientation are then described as function of vehicle velocities projected in vehicle frame:

$$\begin{aligned} \frac{dx(t)}{dt} &= u_0 \cos(\psi(t)) - v(t) \sin(\psi(t)) \\ \frac{dy(t)}{dt} &= u_0 \sin(\psi(t)) + v(t) \cos(\psi(t)) \\ \frac{d\psi(t)}{dt} &= \gamma(t) \end{aligned} \quad (4.12)$$

The lateral tyre forces as modelled with the Pacejka magic formula (Pacejka, 2002 [117]), thus already incorporating the saturation of the force, so that a constraint such as equation (4.8) will not be required:

$$\begin{aligned} F_{y_f 0} &= D_f \sin(C_f \operatorname{atan}(B_f \alpha_f - E_f(B_f \alpha_f \operatorname{atan}(B_f \alpha_f)))) \\ F_{y_r 0} &= D_r \sin(C_r \operatorname{atan}(B_r \alpha_r - E_r(B_r \alpha_r \operatorname{atan}(B_r \alpha_r)))) \end{aligned} \quad (4.13)$$

The lateral tyre slip angles are used in their linear form since it is assumed that the optimal control will not generate manoeuvres over the force peak, being it a non-optimal condition:

$$\begin{aligned} \alpha_f(t) &= \beta(t) + \frac{L_f \gamma(t)}{u_0} - \delta_f(t) \\ \alpha_r(t) &= \beta(t) - \frac{L_r \gamma(t)}{u_0} - \delta_r(t) \end{aligned} \quad (4.14)$$

The steering angle δ_r is zero for a 2-wheel-steering (2WS) vehicle model. Since the model features constant forward velocity and has only one track, load transfers are not considered in this case, so that vertical loads correspond to the steady static conditions:

$$\begin{aligned} N_f &= m g \frac{L_r}{L_r + L_f} \\ N_r &= m g \frac{L_f}{L_r + L_f} \end{aligned} \quad (4.15)$$

4.3.3.2 OCP formulation

The general target function in this case can be described as:

$$J = w_T T(\zeta_f) \quad (4.16a)$$

$$+ w_\psi \left(\frac{\psi(\zeta_f) - \tilde{\psi}}{\Delta_\psi} \right)^2 + w_y \left(\frac{y(\zeta_f) - \tilde{y}}{\Delta_y} \right)^2 \quad (4.16b)$$

$$+ \int_0^1 w_\beta \beta^2(\zeta) d\zeta \quad (4.16c)$$

The Mayer terms 4.16a and 4.16b are respectively used to require the minimisation of manoeuvre time $T(\zeta_f)$, free as in previous case, using the same auxiliary variable and of the final conditions on vehicle heading ψ and lateral position y (as

shown in Figure 4.13). \tilde{y} and $\tilde{\psi}$ are the desired final state values to be reached as close as possible according to the values of Δ_ψ and Δ_y , which are the tolerances (in least square sense) for each final conditions. This is to take into account that final conditions may not be satisfied exactly. However, it is possible to deactivate such conditions by setting to zero their weights, w_ψ and w_y and force the exact final conditions. Boundary conditions can be set as:

$$\begin{aligned}
x(\zeta = 0) &= 0 & (4.17) \\
y(\zeta = 0) &= 0 \\
\psi(\zeta = 0) &= 0 \\
\beta(\zeta = 0) &= 0 \\
\gamma(\zeta = 0) &= 0 \\
\delta_f(\zeta = 0) &= 0 \\
\delta_r(\zeta = 0) &= 0 \\
T(\zeta = 0) &= \text{free}
\end{aligned}$$

$$\begin{aligned}
x(\zeta = 1) &= x_o & (4.18) \\
y(\zeta = 1) &= \tilde{y} \\
\psi(\zeta = 1) &= \tilde{\psi} \\
\beta(\zeta = 1) &= 0 \\
\gamma(\zeta = 1) &= 0 \\
\delta_f(\zeta = 1) &= 0 \\
\delta_r(\zeta = 1) &= 0 \\
T(\zeta = 0) &= \text{free}
\end{aligned}$$

The objective function considers this time also a Lagrange term 4.16c minimises the side slip angle β which is used for improving the comfort of the manoeuvre. The objective can be once again subjected to inequality constraints. In particular, the novel aspect is equation (4.19), which requires that the distance $d(t)$ between the centre of the vehicle and the obstacle is greater than the sum of their main dimensions, as showed in Figure 4.15.

$$d(t) \geq (r_V + r_O) \quad (4.19)$$

where:

$$d(t) = \sqrt{(x(t) - x_o(t))^2 + (y(t) - y_o(t))^2} \quad (4.20)$$

In other words, it is assumed that both vehicle and obstacle are inscribed in circles

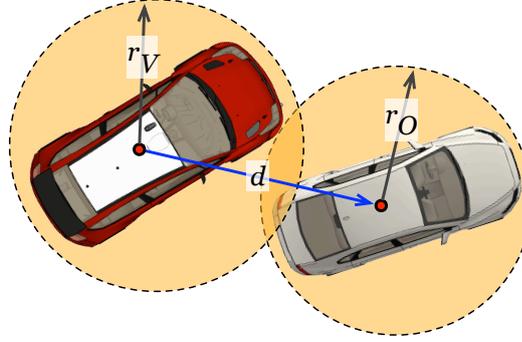


Figure 4.15: Meaning of distance $d(t)$ between vehicle and obstacle modelled as circles.

respectively of radius r_V and r_O (see Figure 4.15). The assumption is in favour of safety and easier to implement. An alternative approach which considers the rectangular shape of the vehicles was used for instance in [33]. If obstacle position $(x_o(t), y_o(t))$ is not constant, i.e. if the obstacle is moving, a motion description must be defined, which could be derived from assumptions and estimations on its future evolution, for instance as it was done in the interactIVe project for the ClearObject manoeuvres (equation (3.19)).

The inequality constraints:

$$|\delta_{fc}(t)| \leq \delta_{fc_{max}}, |\delta_{rc}(t)| \leq \delta_{rc_{max}} \quad (4.21)$$

limit the maximum front and rear steering angles. Otherwise, for the 2WS vehicle with a driver in the loop, they are constraints on state variables and the control is the derivative of the steering angle $v\delta_{fc}(t)$, described in an additional differential equation:

$$\dot{\delta}_{fc}(t) = v\delta_{fc}(t) \quad (4.22)$$

Finally, the inequality constraint (4.23):

$$|v\delta_{fc}(t)| \leq v\delta_{fc_{max}} \quad (4.23)$$

limits the maximum front steering rate. It is used only to simulate humanlike manoeuvres and it is implemented with a special penalty function which is quadratic

in the centre of admissible controls and it is constrained at the borders. This special penalty simulates the minimisation of steering rates when the scenarios do not require aggressive manoeuvres but it allows the maximum steering rate for extreme manoeuvres.

4.4 ANALYSIS OF OCP MANOEUVRES WITH DIFFERENT MODELS AND FORMULATIONS

4.4.1 *Codriver and more complex OCP manoeuvres for warning*

Once the criteria to tune the humanlike manoeuvres of the Codriver have been stated, and the models and OCPs increase the representativeness of the manoeuvres have been presented, it is possible to compare the two approaches.

In particular, the objective is to determine if the simplifications of the Codriver formulation affect the quality of the results, and, in this case, how they limit its applications. The parameter for the performance analysis is the initial jerk they require, being the parameter used for the risk assessment and the release of the warnings by the Codriver.

This difference between the two approaches has been tested by simulation, generating manoeuvres to handle a particular obstacle avoidance scenario, i.e. ego vehicle approaching a fixed obstacle ahead (Figure 4.16), using optimal brake-to-avoid and steer-to-avoid manoeuvres, generated with the two approaches. For the braking scenario, the solved OCP are the FollowObject manoeuvres (equation (3.16)) for the Codriver, and the more detailed OCP described in section 4.3.2. For the steering scenario, the Codriver manoeuvres have been instead of type ClearObject (equation (3.22)), while the more detailed OCP used the formulation of section 4.3.3.

Several scenarios have been tested, in particular all the combinations of relative velocity (i.e. ego vehicle velocity, since the obstacle is fixed) and relative position (i.e. obstacle position x_o , since initial vehicle position is always $x_i = 0$) within a certain range.

In particular, for each velocity the optimal manoeuvres are calculated starting from a high distance obstacle, and iteratively reducing obstacle distance, thus increasing scenario severity. What happens is that at a given point the situation will be so severe to require jerks beyond the usual ranges found during the tests

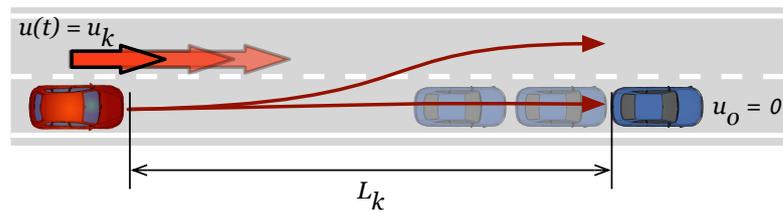


Figure 4.16: Brake-to-avoid and steer-to-avoid manoeuvres at different combinations of relative speed and distance

at the driving simulator (i.e. figures 4.8 and 4.12), so that a warning would be issued. Limiting the longitudinal and lateral jerks respectively to the values of 2 m/s^3 and 500 rad/s , the results are showed in figures 4.17 and 4.18. In

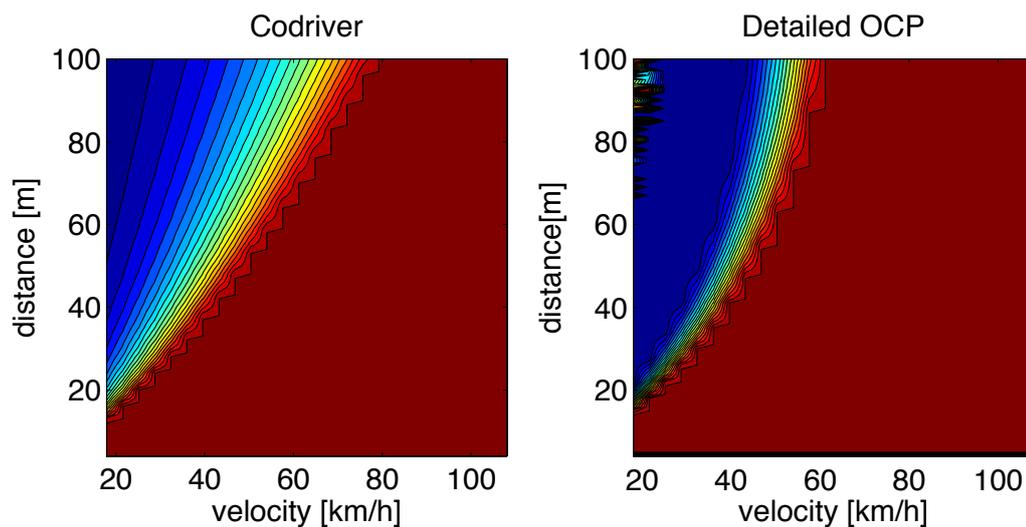


Figure 4.17: Initial jerk levels for brake-to-avoid manoeuvres at different relative velocity and distance, calculated by the Codriver and by more detailed OCPs

both figures, the dark red areas are those where jerks beyond the cited limits are required, i.e. the emergency manoeuvres. The difference is noticeable both for braking and steering manoeuvres, showing that the detailed OCPs require higher jerks than the Codriver. This is because the former formulation takes into

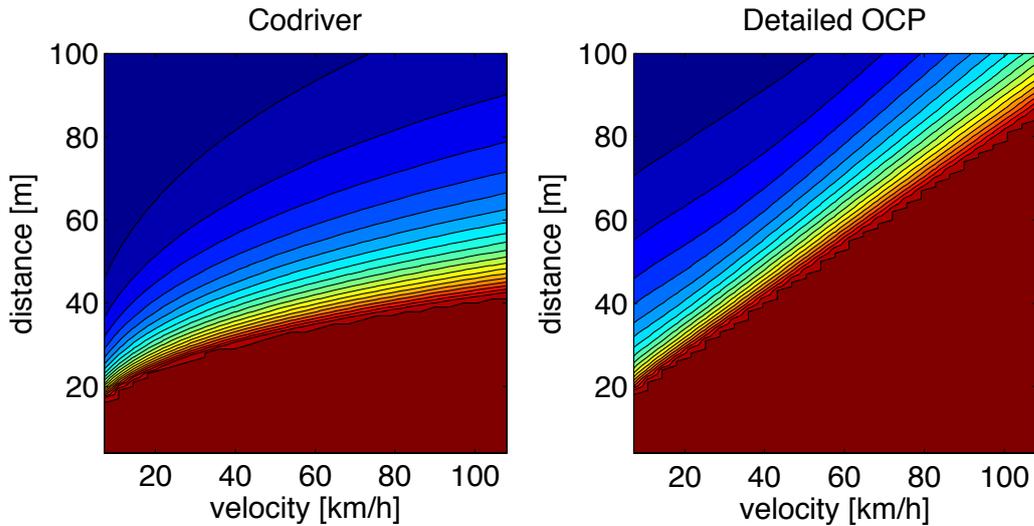


Figure 4.18: Initial jerk levels for steer-to-avoid manoeuvres at different relative velocity and distance, calculated by the Codriver and by more detailed OCPs

account the physical limits of the vehicle, i.e. delays in the actuation (e.g. the first order delays considered in the generation of lateral forces and steering angles), adherence limits etc., while the Codriver simplifies the situation, assuming that the required dynamics are instantaneous and without saturations. In this way, the complex OCP require faster driver actions to compensate the delays and the saturations of the physical limits, thus requiring higher jerks.

However, while for the braking manoeuvres the behaviour is almost similar, big differences can be noticed for lateral dynamics, where the jerks required by the Codriver are much lower, especially at high speed. To understand what happens, the transfer functions between steering angle and vehicle trajectory curvature of the two vehicle models have been compared at different longitudinal velocities. Namely, the Codriver equations for the curvature (equations (3.1)) were compared to the more detailed model (4.11), where however linear tyre model with no saturation was considered. These latter equations have been also first

put into curvilinear coordinates to obtain the curvature equation and make the comparison possible:

$$\begin{aligned}
 \frac{ds_s(t)}{dt} &= u_0 & (4.24) \\
 \frac{ds_n(t)}{dt} &= v(t) + \alpha(t)u_0 \\
 \frac{d\alpha(t)}{dt} &= u_0\Delta(t) \\
 \frac{dv(t)}{dt} &= -u_0^2\Delta(t) - \frac{(C_{yf}L_f - C_{yr}L_r)\Delta(t)}{m} - \frac{(C_{yf} + C_{yr})v(t)}{m} + \frac{C_f\delta(t)}{m} \\
 \frac{d\Delta(t)}{dt} &= \frac{-C_{yf}L_f^2 - C_{yr}L_r^2}{u_0I_z}\Delta(t) + \frac{C_{yf}L_f\delta(t)}{u_0I_z} + \frac{(-C_{yf}L_f + C_{yr}L_r)v(t)}{u_0I_z}
 \end{aligned}$$

The result is showed in Figure 4.19. Looking at their shape the reason of the

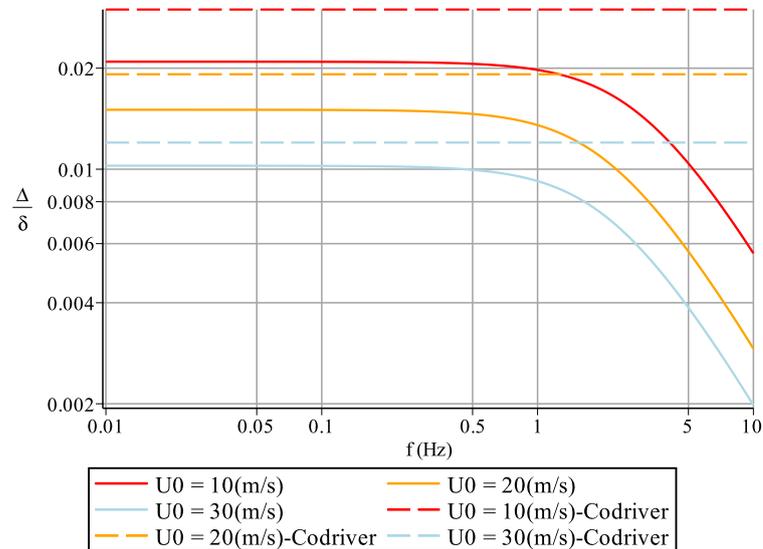


Figure 4.19: Transfer functions between trajectory curvature and steering wheel angle for the Codriver model (dashed) and the one considering also lateral velocity (solid)

different behaviour is clear: the Codriver considers higher gains between the output curvature and the input steering wheel angle even if the model parameters have been the same. This means that the model of the Codriver requires lower steering angles to achieve a given curvature, which then results in lower jerks.

Furthermore, when the steering angle frequency increases, the curvature of the more complex model low-pass filters the value of the steering wheel. On the other hand, integrating the third element of equations (3.1), the transfer function of the Codriver is a constant:

$$\frac{\Delta(s)}{\delta(s)} = \frac{1}{k_1 + k_2 u_0} \quad (4.25)$$

This difference allows to derive a first conclusion: *the dynamics equations of the Codriver neglect important dynamics*. Its assumptions proved to be suitable to generate warnings, but now it is clear that they cannot be used to control the vehicle. As a matter of fact, *more than representing vehicle behaviour, the manoeuvres of the Codriver are designed to represent the internal model of vehicle dynamics the driver has in mind*, which is not particularly accurate for emergency manoeuvres, and especially concerning the lateral dynamics.

These plans can be thus used for warning purposes, but not to directly control the vehicle in emergency manoeuvres. These will be investigated in next section, using only the more refined OCPs.

4.4.2 *Strategies for obstacle avoidance*

4.4.2.1 *Steering versus braking*

The differences between the simplified manoeuvres generated by the Codriver and the more complete OCPs have been showed in previous section. In particular, it was demonstrated that while for warning purposes the Codriver accomplishes its tasks, its results are not completely reliable if one wants to use them for vehicle control.

This can be achieved instead using the more complete models, which can be used to evaluate plans for emergency manoeuvres for obstacle avoidance. These have been planned for the same scenarios of the previous section, i.e. different combinations of relative velocity and position. However, instead of stopping increasing the severity of the scenario when the jerks reached a given thresholds (i.e. when the manoeuvres were not feasible anymore by a human driver), this was done only when the manoeuvres started violating the constraints (i.e. when the manoeuvres were not physically feasible anymore). It is important to note that *the difference between the two represents the margin for autonomous intervention*

when the driver can not intervene successfully anymore (see also Figure 4.23). The duration of the last feasible manoeuvre is thus reported as a limit TTC together with the corresponding velocity.

The performance analysis of the braking and steering manoeuvres is conducted comparing their Time To Collision (TTC), as described also in Brännström et al., 2010,[42]. TTC is here defined as the duration of the limit manoeuvre. A limit manoeuvre is that which handles the limit scenario, i.e. the already mentioned combination of relative velocity and position which allows the last feasible manoeuvre to avoid the collision without violating the constraints. The results are showed in Figure 4.20 for the braking manoeuvres and in Figure 4.21 for the steering manoeuvres (using a front steering vehicle). In the upper charts,

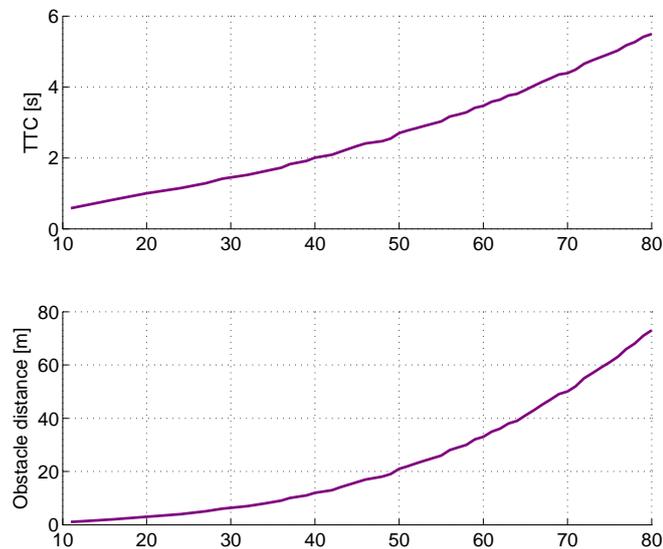


Figure 4.20: Times to collision and obstacle limit position as functions of relative speed for braking manoeuvres

their TTCs are showed as functions of relative velocity (i.e. the duration of the manoeuvre at that velocity which could avoid the nearest obstacle possible). The corresponding distances are reported in the lower chart. The lines describe the limit scenarios: those over the lines can be safely handled, while below the lines the scenario is too severe and obstacle avoidance is not possible anymore. In

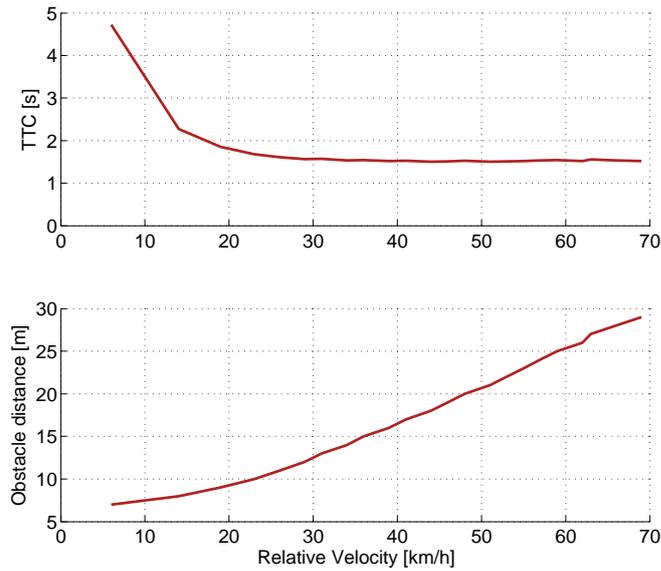


Figure 4.21: Times to collision and obstacle limit position as functions of relative speed for steering manoeuvres

braking manoeuvres, the TTCs increase almost linearly with vehicle velocities, meaning that higher and higher distances from the obstacle are needed for start braking and successfully avoid it. On the contrary, steering manoeuvre require almost constant TTCs after some high values at low velocities. This means that at high speed, the time required to avoid an obstacle by steering is almost the same. Merging them into the interesting Figure 4.22, important conclusions can be derived on the two avoidance approaches. The meaning of the figure, which well matches the results of Brännström et al. [42] (the same numerical values cannot be obtained, since they depend on vehicle performances, obstacle side and position, etc), is in fact simple. At low velocities, approximately below 30 km/h, the braking curve is below the steering curve, while at higher velocities the contrary happens. This means that *braking manoeuvres at low velocities allow the avoidance of closer and more dangerous obstacles* or, in other words, that *at low velocity obstacles at a given distance can be avoided by braking at higher (but still low) relative velocity than by steering*. On the contrary, given a high relative velocity, *steering manoeuvres allow avoiding closer obstacles*, or, again, *at high*

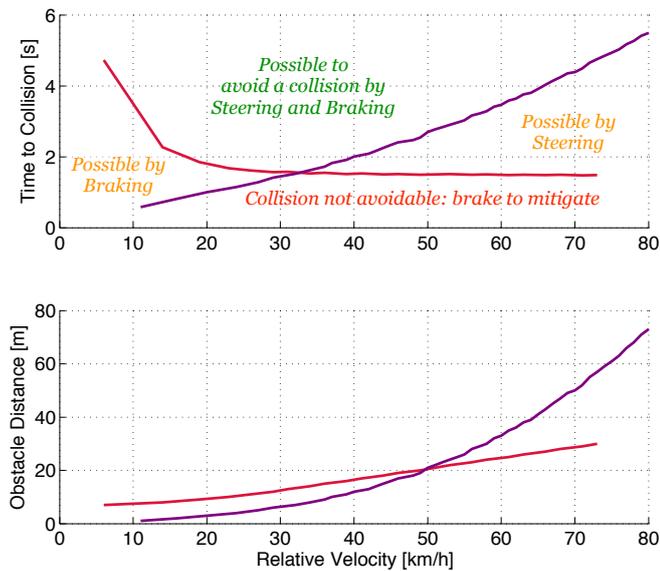


Figure 4.22: Brake TTCs and limit obstacle distances (purple) compared with those of the Steering manoeuvres (red)

velocity obstacles at a given distance can be avoided by steering at an even higher relative velocity than by braking.

These conclusions are very important for an ADAS for obstacle avoidance, which, depending on the scenario, will a priori know if it is the case to enable a braking or a steering manoeuvre, remembering that the threshold of 30 km/h is only valid for the sample vehicle and obstacle modelled here. However, while autonomous braking manoeuvres have been already widely analysed worldwide, and ADAS with these capabilities are already available on market, the analysis now will focus autonomous emergency steering manoeuvre.

4.4.2.2 *Two-wheel steering versus four-wheel steering*

Focusing now the analysis on autonomous steering manoeuvres, the objective is how to design vehicles with more and more effective emergency steering characteristics.

In this paragraph it will be showed that an *optimally* driven four-wheel-steering (4WS) vehicle achieves superior performance with respect to a 2WS vehicle in

emergency obstacle avoidance manoeuvres. This result is in contrast with the achievement of Lee, 1995 [95], where it is shown that there is not a sensible improvement in lane-change manoeuvres with 4WS compared to those obtained by proficient drivers with 2WS vehicles. The point is that here the focus is on

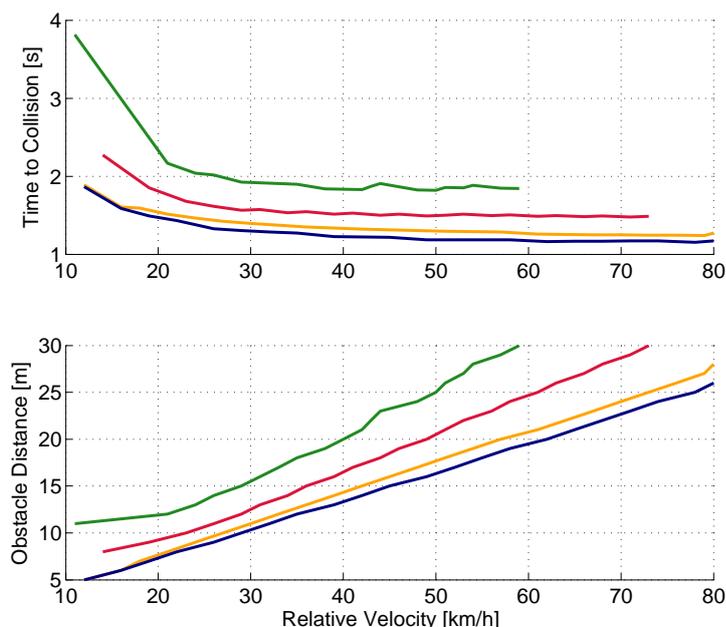


Figure 4.23: Times to collision and obstacle limit position as functions of relative speed, with humanlike 2WS (green), autonomous 2WS (red), comfortable 4WS (orange) and minimum time 4WS (blue)

autonomous avoid by steer manoeuvres instead of evaluating the two steering systems (i.e. 2WS versus 4WS) using a human-like driver model. In fact, what Lee does is to compare humanlike manoeuvres, i.e. some kind of Codriver manoeuvres even if more refined, on 2WS and 4WS vehicles, while here the detailed OCP will exclude humanlike planning criteria, thus exploiting vehicle performances to its limits thanks to automatic actuation, instead of limiting them considering driver limits.

The analysis regards four OCP problems, which compare front steering humanlike manoeuvres (but calculated with the detailed OCP and not that of the Codriver), front steering autonomous manoeuvres (those already showed in Figure 4.21),

and front and rear steering autonomous manoeuvres (comfortable or requiring maximum performance). Humanlike manoeuvres model a human-driven vehicle, considering his/her limits in the steering wheel velocity and minimising lateral jerk. On the other hand, the autonomous manoeuvres push the vehicle to its limits in a minimum time manoeuvre ($w_T = 1$ in equation (4.16)). The resulting manoeuvre with a 4WS vehicle could be uncomfortable due to the high lateral velocity and almost zero yaw rate. Therefore, comfortable autonomous manoeuvres that minimise side slip angles have been evaluated as well ($w_\beta = 1$).

The difference between their performances is clear in Fig. 4.23. In an analogue way to Figure 4.22: combined front and rear steering manoeuvres at a given speed allow the avoidance of closer and more dangerous obstacles or, in other words, obstacles at a given distance can be avoided with 4WS even if relative velocity is higher. Moreover, autonomous 2WS manoeuvres allow handling more severe scenarios than when a driver, with slower actuation capabilities, is controlling the vehicle and that minimum sideslip 4WS manoeuvres are less effective than pure minimum time manoeuvres.

The reason is clearer in Fig. 4.24 which shows similar trajectories at the highest velocity at which collisions can be avoided in all cases: when minimum sideslip is required, the trajectory and the yaw angle are very smooth, while with only minimum time the manoeuvre exploits rear steering to obtain almost pure translation, with yaw close to zero. This allows the saturation of lateral forces earlier and for longer period which means avoiding the at higher velocities (Figures 4.27 and 4.28). This behaviour is possible by means of steering front and rear wheels in phase, i.e. with the same sign, while to obtain minimum lateral velocity the wheels are initially steered with opposite sign. This detailed analysis allows also to notice the much slower and smoother steering angle profiles obtainable by human drivers, which result in lower performances (Figure 4.25).

The difference between the 2WS humanlike manoeuvre and the most effective 4WS manoeuvre is the margin for an ADAS for autonomous avoidance to intervene, exploiting also rear wheel steering. Quantitatively, it is shown that *at 60 km/h there are still 10 m to intervene for the ADAS when the driver cannot act effectively anymore, which result in approximately more than 0.6 s more to handle a dangerous situation, values that are the double of those achievable by autonomous*

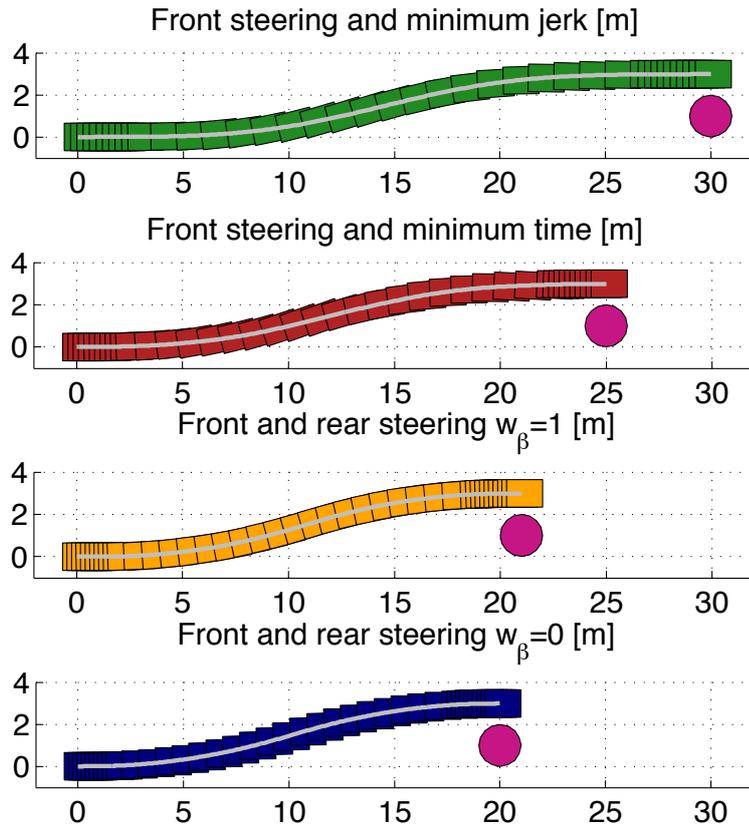


Figure 4.24: Limit trajectories for forward velocity $u_0 = 60 \text{ km/h}$, with only front steering and minimum jerk, only front steering and combined front and rear steering

2WS. For this reason, in next chapter an ADAS for autonomous obstacle avoidance by means of 4WS will be proposed.

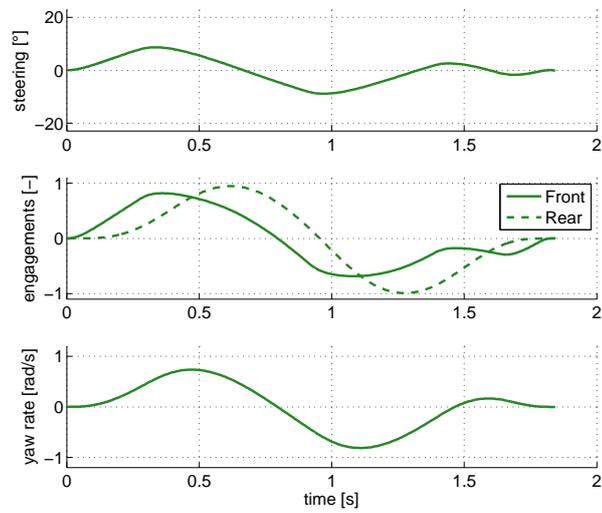


Figure 4.25: Humanlike manoeuvre characteristics at $u_0 = 60 \text{ km/h}$ with only front steering, solid for front axle and dashed for rear axle

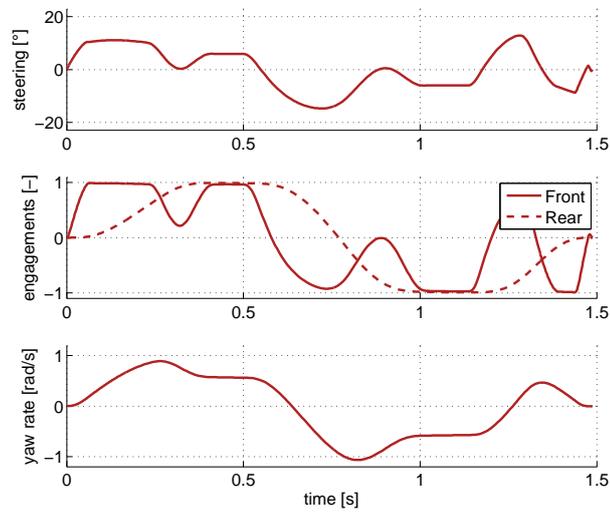


Figure 4.26: Limit manoeuvre characteristics at $u_0 = 60 \text{ km/h}$ with only front steering, solid for front axle and dashed for rear axle

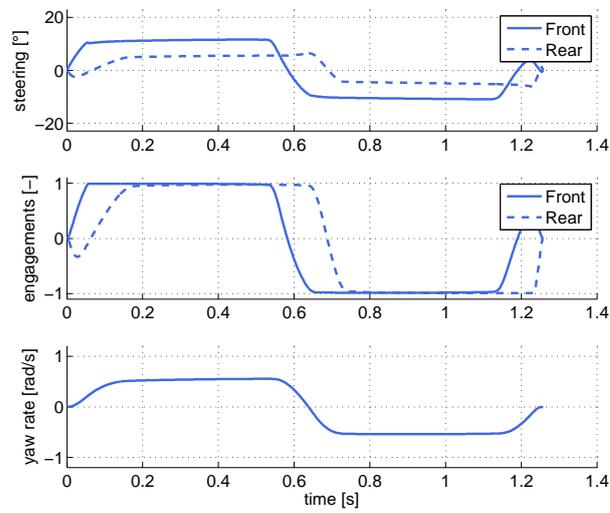


Figure 4.27: Comfortable manoeuvre characteristics at $u_0 = 60 \text{ km/h}$ with front and rear steering, solid for front axle and dashed for rear axle

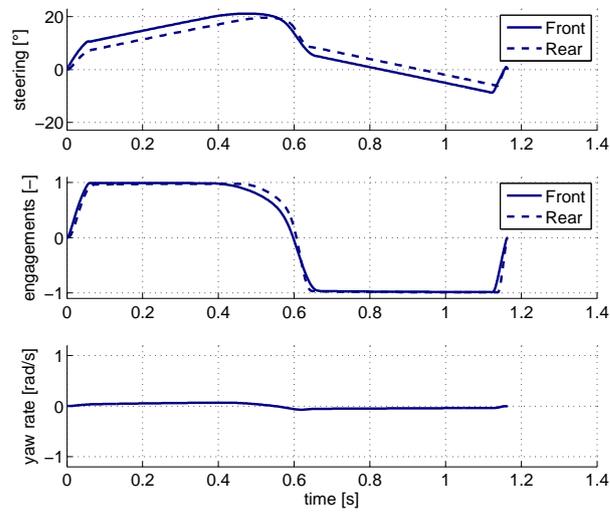


Figure 4.28: Limit manoeuvre characteristics at $u_0 = 60 \text{ km/h}$ with front and rear steering, solid for front axle and dashed for rear axle

AN ADAS FOR AUTONOMOUS COLLISION AVOIDANCE: TESTS WITH FPEV2-KANON

| | | |
|-------|---|-----|
| 5.1 | Introduction | 112 |
| 5.2 | Overview | 113 |
| 5.2.1 | Objectives | 113 |
| 5.2.2 | The FPEV2-Kanon vehicle | 115 |
| 5.2.3 | System Architecture | 118 |
| 5.3 | Manoeuvre generation with OCP | 120 |
| 5.3.1 | General strategies for manoeuvre planning | 120 |
| 5.3.2 | Vehicle Modelling and Identification | 122 |
| 5.3.3 | Example optimal manoeuvres for Kanon: double lane changes | 129 |
| 5.4 | Threat assessment | 134 |
| 5.5 | Control Algorithms | 135 |
| 5.5.1 | General strategies for trajectory tracking | 135 |
| 5.5.2 | Early algorithms | 136 |
| 5.5.3 | Final algorithm | 140 |
| 5.6 | Tests and results | 144 |
| 5.6.1 | Experiment details | 144 |

5.1 INTRODUCTION

This chapter regards the second part of this research work, namely the design of an ADAS for autonomous obstacle avoidance in emergency manoeuvres. As it was explained in previous section, the focus of the ADAS is on steer-to-avoid manoeuvres, exploiting an autonomous front and rear steering system to get the best performance possible out of the vehicle, as described in the previous Chapter.

Neglecting the perception layer, this kind of ADAS mainly requires a threat assessment method, and the ability of planning and executing safe manoeuvres when the risk level is beyond a certain threshold. The research activities have regarded both these aspects.

The risk assessment method is based on manoeuvre initial lateral jerks: if a safe manoeuvre requires an initial jerk beyond a certain threshold, it means that a high correction is required and the safety function must be activated. A second stage of the activity regarded the calculation of optimal manoeuvres for collision avoidance, and the development of control algorithms to make the vehicle track them, if enabled by the threat assessment stage. This part has been successfully implemented and tested during a research period at the Hori-Fujimoto Laboratory [25] of the University of Tokyo, Japan. In particular, the electric experimental vehicle FPEV2-Kanon (Kanon hereon) has been used, developed within the same laboratory (Fujimoto et al. [73],[76] and Figure 5.1), which successfully tracked single-lane change manoeuvres which allowed to avoid a dummy obstacle.

In this activity, the two components of threat assessment and motion planning/execution have not been merged into an unique application, due to hardware and time limitations during the activity at the Hori-Fujimoto Lab. However, the possible objectives and architecture of such a complete ADAS will be described in section 5.2, followed by some specific characteristics of optimal control problem formulation to generate avoidance manoeuvres (section 5.3.3), which is actually similar to that described in Section 4.3.3. The risk assessment strategy is then described in Section 5.4 (and based on the considerations made in 4.2.3.2),



Figure 5.1: The experimental vehicle FPEV2-Kanon

while the control algorithms design and implementation are reported in section 5.5). Finally, results will be presented and discussed in section 5.6.

5.2 OVERVIEW

5.2.1 *Objectives*

The objective of this activity is the development of an ADAS for autonomous obstacle avoidance, i.e. an ADAS for active safety, which autonomously takes over the authority from the driver when it judges it is not possible anymore for him to make a correction. In particular, the system focuses obstacle avoidance by autonomous 4WS steering at constant speed, as the result of tracking an optimal manoeuvre.

As a first activity, it has been investigated when the intervention of such system should be triggered, implementing a threat assessment algorithm. This work proposes a method based on manoeuvre jerks: if a corrective manoeuvre requires an initial jerk beyond a certain threshold, i.e. if the correction required is too high, the system assumes that the driver will not be able to do it anymore, so that the situation is not safe and the safety function must be activated. It has been already

demonstrated in section 4.2 that initial manoeuvre jerks can be used as efficient parameters to assess the risk level and enable the autonomous intervention with proper timing.

The second topic described here is the planning and control of the avoidance manoeuvre. The implemented manoeuvre is an optimal single lane-change to avoid obstacles in front of the ego vehicle, obtained solving an optimal control problem. The resulting manoeuvre is then used as a reference for a proper control algorithm, which has been developed to make the vehicle track the desired trajectory. In particular, it will be showed that tracking the yaw rate and the sideslip profiles output by optimal control is sufficient to obtain the vehicle follow the desired path, without using GPS and position control. Furthermore, it will be showed that good tracking of the desired manoeuvre can be obtained using quite cheap and standard sensors, substituting expensive sensors with good state observers as in the case of vehicle lateral velocity.

The activity is limited to the separate development of these two components, which have not been merged into an unique ADAS. For the completion of it, also a perception platform for obstacle detection would have been necessary.

As it was anticipated, this intervention ADAS could be also merged to the warning functions of the interactIVe project described in Chapter 3, to build a complete ADAS for active safety which takes care of all the phases before a possible accident (Figure 1.3), even though this integration has not been even tried during this research.

As a final remark, even though the objective of this activity is the autonomous execution of steer-to-avoid manoeuvres because they allow the avoidance of closer obstacles than brake-to-avoid when speed is high, during the activity only low-speed steer-to-avoid manoeuvres have been autonomously executed. This is because of the internal rules of the campus where the test field lays, which did not allow the Kanon to travel faster than 30 km/h. The objective is thus restricted to the validation of the planning-execution method. Different velocities have been tested up to 20 km/h without requiring tuning modifications, so that here it is assumed that nothing would change beyond 30 km/h.

5.2.2 The FPEV2-Kanon vehicle

For the threat assessment algorithm, the data coming from the driving simulator test campaign can be used to set the thresholds. The process for data logging and the used tools have already been described in Section 4.2. On the other hand, for the testing of manoeuvre tracking, the electric vehicle FPEV2-Kanon has been used. The vehicle has been designed and developed by the Fujimoto Lab of the University of Yokohama and then further improved at the Hori-Fujimoto Laboratory of the University of Tokyo. The vehicle has one seat to host one driver. Its main characteristics are reported in Table 5.1, while others, identified during the development of the application, are reported in next sections. The vehicle is

| Parameter | Value |
|---|-------|
| Mass [kg] | 870 |
| Yaw Inertia [kgm^2] | 631 |
| Wheelbase [m] | 1.7 |
| Front axle distance from CoG [m] | 0.999 |
| Rear axle distance from CoG [m] | 0.701 |
| Track (front and rear) [m] | 1.3 |
| Wheel radius (front and rear) [m] | 0.302 |
| Front wheel rolling inertia [kgm^2] | 1.26 |
| Rear wheel rolling inertia [kgm^2] | 1.24 |

Table 5.1: Vehicle parameters

electric, equipped with 4 in-wheel motors (IWM) produced by Toyo-Denki Seizo K.K. Ltd., connected to the tyres by direct drive, which allow independent control of the torque of each wheel. Their characteristics are resumed in Table 5.2 ([76]). Furthermore, another key feature of the vehicle is the capability of both front and rear steering. Front steering can be actuated both by the driver using the steering wheel, and by electric actuators controlled by an Electronic Power Steering (EPS) device. Rear steering can be only controlled by its dedicated EPS. Their main features are reported in Table 5.3. Finally, the vehicle is equipped with several sensors to reconstruct its state, whose positions are reported in Figure 5.2. First of all, a GPS and an IMU are positioned on the centre of gravity of the vehicle, to

| Characteristic | Front | Rear |
|-------------------------|---------------------|-------------|
| Manufacturer | Toyo Denki | |
| Type | Direct Drive System | |
| Rated torque [Nm] | 110 | 137 |
| Maximum torque [Nm] | 500 | 340 |
| Rated power [kW] | 6.0 | 4.3 |
| Maximum power [kW] | 20.0 | 10.7 |
| Maximum speed [rpm] | 1113 | 1500 |
| Weight [kg] | 32 | 26 |
| Cooling system | Air cooling | |

Table 5.2: Electric motor characteristics

| Parameter | Value |
|--------------------------------------|--------------|
| Front steering angle range [rad] | ± 0.4 |
| Rear steering angle range [rad] | ± 0.08 |

Table 5.3: Steering parameters

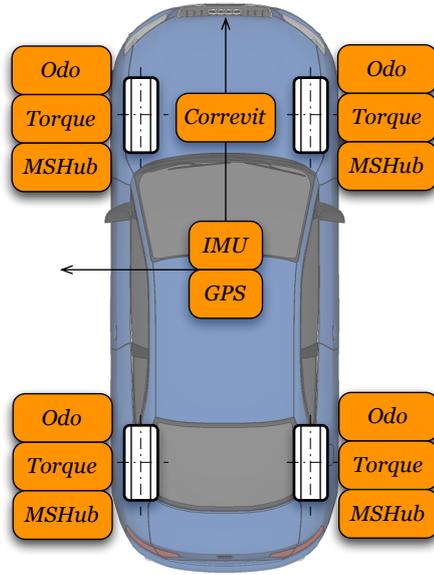


Figure 5.2: Sensors equipping Kanon vehicle for state reconstruction

measure its position, accelerations and rotational velocities. To reconstruct also vehicle velocity both in longitudinal and lateral directions, a non-contact optical sensor is used, i.e. “Correvit” manufactured by Corrsys-Datron. This sensor is positioned at the centre of front axle and not in the CoG, so that measured lateral velocity and sideslip angle must be corrected of the term due to vehicle yawrate:

$$\begin{aligned}
 v(t) &= v_M(t) - L_f \gamma(t) \\
 \beta(t) &= \beta_M(t) - L_f \frac{\gamma(t)}{u(t)}
 \end{aligned}
 \tag{5.1}$$

Finally, on the wheels common odometers are installed to measure their rotational velocity, and it is also possible to easily determine the torque applied by the motors measuring their current. In addition, novel lateral force sensors are installed in each wheel, and allow the direct estimation of tyre forces without estimating tyre slips (Nam et al. 2011 and 2012 [111],[112],[113]). Such sensors, named Multi-Sensing Hubs (MSHub) have been developed by NSK Ltd [116].

The vehicle is then equipped with Lithium batteries as power storage devices, which supply all the electronic equipment through suitable inverters. Finally, the

complete vehicle is controlled by a dSpace Autobox DS1103, with a power PC 750GX controller board running at 933 MHz, with a 16-channel analog-to-digital converter and an 8-channel digital-to-analog converter, which is also used for data logging.

5.2.3 System Architecture

A complete ADAS for autonomous obstacle avoidance could be designed with the architecture reported in Figure 5.3. As a first attempt, it could be built with the

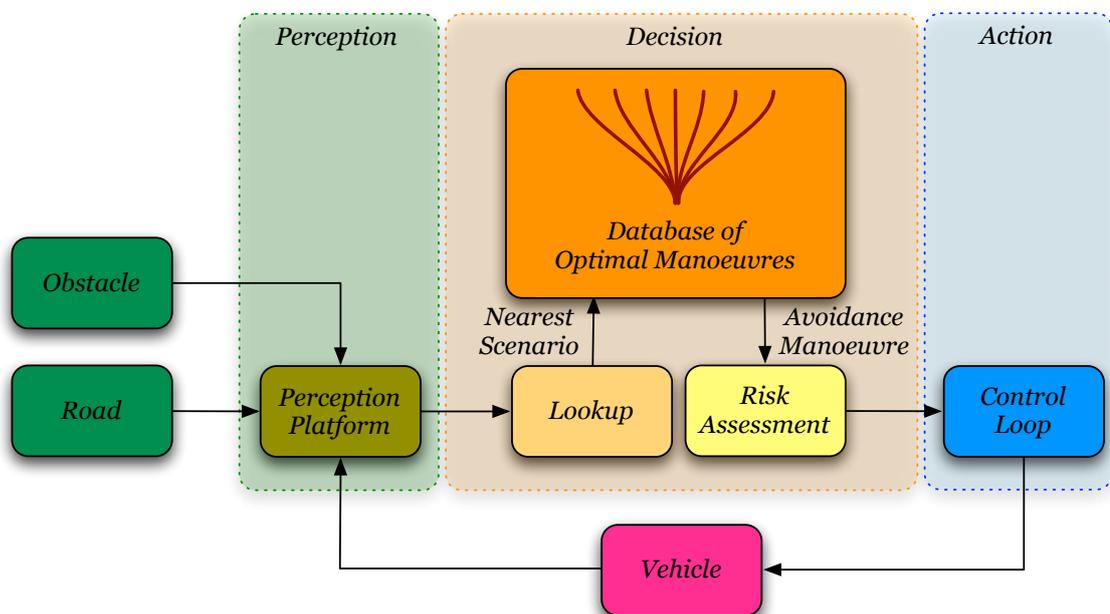


Figure 5.3: Possible architecture for an ADAS for autonomous obstacle avoidance

usual three-layered architecture, with Perception, Decision and Action modules. A perception platform, not developed during this activity, should be in charge of reconstructing current scenario, i.e. sensing vehicle state, road characteristics and obstacle states.

The decision module should then assess the risk level, for which it is proposed to use initial jerks as manoeuvre parameters, and steer-to-avoid manoeuvre planning, proposed here by means of optimal control. Since it was demonstrated that for the

planning of autonomous avoidance manoeuvres detailed models and OCP should be used, this task can be very demanding in terms of computational power, so that the system could be provided with a database of pre-evaluated manoeuvres, so that it is only necessary to lookup among them instead of calculating them in real-time, in a manner inspired by Urmson et al., 2007 [134]. The manoeuvres should cover all the possible scenarios addressed by the system, and could be used both for threat assessment and as references for motion control, as it will be explained later. This means that the manoeuvres should take into account different initial states of the vehicle, i.e. velocities and yaw rates, and different positions of the obstacles, as showed in Figure 5.4. Evasive manoeuvres would be the result of optimal control problems, and could consider avoidance both on the left and the right side. The system should lookup in the database, and choose

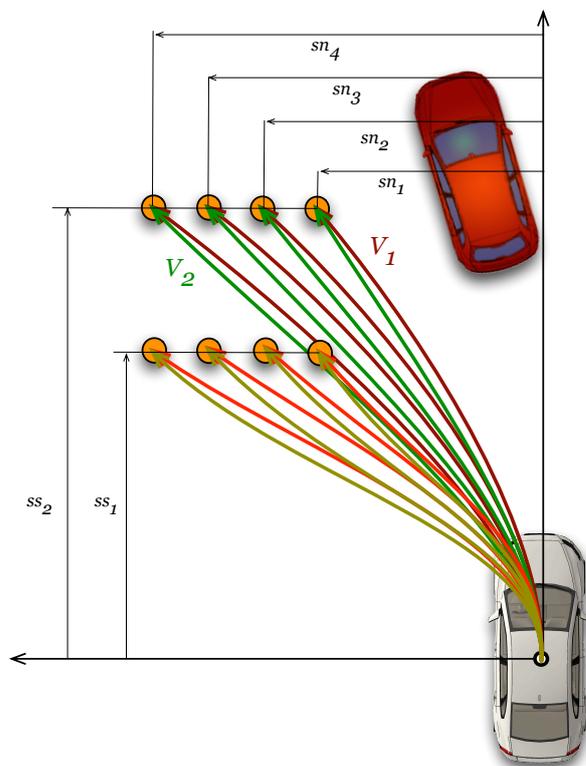


Figure 5.4: Example of possible scenarios covered by the manoeuvre database: different ego vehicle initial velocities, different obstacle positions with respect to ego vehicle

the manoeuvre for collision avoidance which handles the scenario which best matches current scenario: current vehicle state, obstacle position and possibly motion, availability of free space on the left and on the right, etc. In other words, the system should look for the manoeuvre which allows safe collision avoidance in current scenario, as if it was calculated in real-time for the same situation. The database scenarios would be probably discrete, so that it could happen that the chosen manoeuvre is designed for a slightly different one, chosen in favour of safety. Another alternative would be to try interpolating between the discrete values have been evaluated, thus constituting a multi-dimensional plane where also the manoeuvres in between match the current scenario perfectly.

Once the closest scenario is chosen, one manoeuvre will be related to it: this could be used for threat assessment, since it represents the optimal way for handling current scenario. The manoeuvre is known, and thus so are its initial jerks. If they are too high, beyond given thresholds and thus requiring too sudden corrections, this means that the driver may not be able to perform the manoeuvres, so that autonomous intervention can be enabled. It was already showed in paragraph 4.4.2.2 that beyond the limits of human actuation there is still a considerable margin for autonomous intervention.

If this is the case, the Action module is enabled, and it is in charge to generate proper control inputs which override those of the driver to achieve the given goal. In this work, this was successfully achieved designing a control algorithm tracking yaw rate and lateral velocity profiles, using front and rear steering as inputs to the vehicle and keeping constant velocity.

5.3 MANOEUVRE GENERATION WITH OCP

5.3.1 *General strategies for manoeuvre planning*

Vehicle trajectory planning and execution is a very important research topic worldwide, which deals with increasing automation of transport systems and autonomous driving. A survey on motion planning algorithms is reported in this section, followed by the approach chosen for this activity, while state-of-the art trajectory execution and the algorithm developed in this work will be reported in section 5.5.

Real-time trajectory planning is a relatively recent research topic, due to the

difficulties in environment perception and to the demanding calculation resources needed for this purpose (Stentz and Hebert, 1995 [128], [91]), and only in the last years it has been possible to achieve real-time motion planning in complex scenarios. The accomplishment of this task has been heavily pushed by the 2007 DARPA Urban Challenge [11], where a large variety of algorithms has been developed for autonomous navigation in a nearly urban environment. The algorithm used in the winning vehicle Boss, described in Ferguson et al., 2009 [69], used a database of pre-evaluated parametric manoeuvres as guess solutions for an optimisation problem, which adapted them to desired goals and contingent constraints such as final desired relative position and orientation. The algorithm distinguishes between planning manoeuvres which follow road geometry, and others in a not organised environment, such as intersections, parking lots, etc., which instead are built using the lattice idea, again in [69].

The same feature was also used in Junior, the vehicle which arrived second in the race (Montemerlo et al., 2009 [108]). The high level plans were made using dynamic programming (Howard, 1960[86]), while low level trajectory was selected among a set including smoothed road centreline and similar trajectories with different discrete final lateral displacements. However, in non organised environments such as intersections, a version of A* algorithm (Hart et al. 1960 [78]) was used to generate an overall path, which was later smoothed using Conjugate Gradient. A* was also used in the third vehicle, Odin [120], which also has in common with the others the separation of trajectory planning and velocity planning. A different approach is that used in Talos vehicle [96], which uses Rapidly exploring Random Tree (La Valle and Kuffner, 2001 [94]), i.e. a fast generation of random points followed by forward integration to check the feasibility of the path.

All these algorithms feature the separation of longitudinal and lateral dynamics, which is a widely used approach also outside the environment of the DARPA challenges. Althoff et al. [31] propose the generation of several polynomials both for trajectory description and vehicle velocity. Also in this case, a set of manoeuvres is first generated, and their feasibility is assessed only afterwards, checking if they cause collisions etc. (Frazzoli et al. 2002 [72])

In other words, to the author's knowledge it seems that there is not an approach available to handle the entire problem into an unique formulation, which takes

into account combined lateral and longitudinal dynamics, initial and final desired state, constraints such as obstacles and road boundaries, and only outputs one result manoeuvre as the unique optimal solution to handle the situation according to given goals. In this way, the approach of manoeuvre generation using OCP as described below appears to be an innovation in common practice for vehicle motion planning, with the advantage of obtaining a globally optimal manoeuvre as a result, and not the sub-optimal best manoeuvre in a discrete set. The obtained manoeuvres can be also humanlike, and they are the natural output of the algorithm if the problem is properly set, instead of being the result of closed loop empirical rules fitting driver behaviour (e.g. the Rule-based trajectory planner used by Broggi et al. [141] in the DARPA Urban Challenge).

This holistic OCP approach, in his turn, has also some drawbacks. As an example applied to the framework where this research is collocated, some issues raise when more obstacles are present in the scenario: in this case, multiple solutions may be available, and very robust solvers should be used to avoid finding local optima. For the sake of simplicity, however, in this work only one obstacle is considered.

5.3.2 *Vehicle Modelling and Identification*

5.3.2.1 *Vehicle Model*

Vehicle modelling is the first step to generate the database of optimal avoidance manoeuvres. As it is described in Appendix B, a model of vehicle dynamics is required for optimal control formulation, to be included as an ODE constraint. This allows the resulting manoeuvre to be physically feasible. The vehicle model used for the autonomous motion planning is very similar to that for lateral dynamics developed in section 4.3.3.1. The parameters used for the model are those of the Kanon vehicle of the Hori-Fujimoto Laboratory of the University of Tokyo. While some of them were already known (i.e. tables 5.1, 5.2 and 5.3) others had to be identified, as it will be described in next paragraph.

The model is a classic bicycle model, as represented in Figure 4.14, where the meaning of the symbols is reported in Appendix C. In this model, constant forward

vehicle velocity is considered, considering manoeuvres for avoidance by means of only steering, without braking, so that:

$$u(t) = u_0 \quad (5.2)$$

The equations for lateral dynamics and yaw rate are described according to Abe [28], and are those already mentioned in paragraph 4.3.3.1, i.e. (4.11a) and (4.11b) For convenience they are reported here again:

$$\begin{aligned} m u_0 \left(\frac{d\beta(t)}{dt} + \gamma(t) \right) &= F_{y_f}(t) + F_{y_r}(t) \\ I_z \frac{d\gamma(t)}{dt} &= L_f F_{y_f}(t) - L_r F_{y_r}(t) \end{aligned} \quad (5.3)$$

while position and orientation are again described by projection equations 4.12:

$$\begin{aligned} \frac{dx(t)}{dt} &= u_0 \cos(\psi(t)) - v(t) \sin(\psi(t)) \\ \frac{dy(t)}{dt} &= u_0 \sin(\psi(t)) + v(t) \cos(\psi(t)) \\ \frac{d\psi(t)}{dt} &= \gamma(t) \end{aligned} \quad (5.4)$$

Dynamics for tyre force generation are again included as first order lags (see equations (4.11d) and (4.11c)), to consider the time necessary for the generation of reference tyre forces F_{y_f0} and F_{y_r0} and their transmission to the chassis:

$$\begin{aligned} \frac{l_y}{u_0} \frac{dF_{y_f}(t)}{dt} &= F_{y_f0}(t) - F_{y_f}(t) \\ \frac{l_y}{u_0} \frac{dF_{y_r}(t)}{dt} &= F_{y_r0}(t) - F_{y_r}(t) \end{aligned} \quad (5.5)$$

This time, however, the Pacejka Model (4.13) was not used: since the model should have the correct parameters of the Kanon, it would be necessary to identify 4 parameters for each wheel on a complex model, which was not possible in the limited amount of time of the research. Thus, a simpler linear model has been using, assuming that the saturation of the forces would never been reached (it is useful to remember that the vehicle could not exceed 30 km/h):

$$\begin{aligned} F_{y_{0f}}(t) &= -C_{y_f} \alpha_f(t) \\ F_{y_{0r}}(t) &= -C_{y_r} \alpha_r(t) \end{aligned} \quad (5.6)$$

where lateral tyre slips are defined with constant forward velocity, and consider also non-zero rear steering angle:

$$\begin{aligned}\alpha_f(t) &= \beta(t) + \frac{L_f \gamma(t)}{u_0} - \delta_f(t) \\ \alpha_r(t) &= \beta(t) - \frac{L_r \gamma(t)}{u_0} - \delta_r(t)\end{aligned}\quad (5.7)$$

In an analogue way to equations (4.11e) and (4.11f), also the dynamics for the actuation of front and rear steering angles have been considered. In fact, once the reference steering angle is set to the EPS, the combined dynamics of inner control loop and actuation are not negligible. However, this time the steer dynamics have been identified and modelled for the Kanon as of second order:

$$\begin{aligned}M_f \frac{d^2 \delta_f(t)}{dt^2} + C_f \frac{d \delta_f(t)}{dt} + K_f \delta_f(t) &= \delta_{0f}(t) \\ M_r \frac{d^2 \delta_r(t)}{dt^2} + C_r \frac{d \delta_r(t)}{dt} + K_r \delta_r(t) &= \delta_{0r}(t)\end{aligned}\quad (5.8)$$

The identification process which lead to this conclusion will be better described in section 5.3.2.2.

Since the model features constant forward velocity and has only one track, load transfers have not been considered, so that vertical load repartition is only based on the distance of vehicle CoG from rear and front axle. Cornering stiffnesses are thus described as constant parameters:

$$\begin{aligned}C_{yf} &= K_{yf} F_{zf} = K_{yf} m g \frac{L_r}{L_r + L_f} \\ C_{yr} &= K_{yr} F_{zr} = K_{yr} m g \frac{L_f}{L_r + L_f}\end{aligned}\quad (5.9)$$

In the end, the model has 9 states, which can be extended to 11 when the two second order equations 5.8 are split into four first order equations:

$$X(t) = \left\{ \beta(t), \gamma(t), x(t), y(t), \psi(t), F_{yf}(t), F_{yr}(t), \frac{d \delta_f(t)}{dt}, \frac{d \delta_r(t)}{dt}, \delta_f(t), \delta_r(t) \right\}^T \quad (5.10)$$

and 2 inputs, which will be autonomously controlled by the system:

$$U(t) = \{ \delta_{0f}(t), \delta_{0r}(t) \}^T \quad (5.11)$$

Longitudinal dynamics are not considered in optimal motion planning, and there will be a separate control loop in charge of keeping constant longitudinal velocity acting on the driving torques of the 4 wheels, which was already available on Kanon and which will not be described here.

5.3.2.2 Model Identification

Given the model of the section above, it has been necessary to identify some key parameters, in order to generate manoeuvres which resemble as much as possible real vehicle dynamics. In this way, properly tracked references in vehicle sideslip and yaw rate, generated by means of optimal control, result in the desired trajectories without using position control, i.e. tracking of longitude and latitude profiles.

Some model parameters, such as vehicle mass and yaw inertia, CoG position, time lag for lateral force generation and others have not been identified during this activity, relying on previous research results, e.g. Zendri, 2010 [143], and data already available within the Hori-Fujimoto Laboratory. Special attention has been put on vehicle cornering stiffnesses and steering dynamics instead.

Cornering stiffnesses

To identify cornering stiffnesses C_{yf} and C_{yr} , offline data from a curve manoeuvre at constant speed have been used, filtered at 5 Hz (forward and backwards, to avoid filtering lags). The general idea is to find estimated cornering stiffnesses solving the tyre model equations 4.13 for them:

$$\begin{aligned} C_{yfE} &= -\frac{F_{yfE}(t)}{\alpha_{fE}(t)} \\ C_{yrE} &= -\frac{F_{yrE}(t)}{\alpha_{rE}(t)} \end{aligned} \quad (5.12)$$

so that it is necessary to estimate lateral forces and tyre sideslips. Lateral forces are estimated using lateral dynamics equations 5.3:

$$\begin{aligned} F_{yfE}(t) &= L_r m \left(a_{yM}(t) + I_z \frac{d\gamma_M(t)}{dt} \right) \\ F_{yrE}(t) &= L_f m \left(a_{yM}(t) - I_z \frac{d\gamma_M(t)}{dt} \right) \end{aligned} \quad (5.13)$$

where the suffix E means estimated variables, while M means those directly measured. In this case, the measurement of $\gamma_M(t)$ comes from the gyros, and

is then filtered and numerically derived to obtain yaw acceleration $\frac{d\gamma_M(t)}{dt}$, while lateral acceleration a_{yM} is measured by accelerometers. Another way for obtaining it, which has been investigated as well, is to use the expression of equation 5.3:

$$a_{yM}(t) = u_0 \left(\frac{d\beta_C(t)}{dt} + \gamma_M(t) \right) = \frac{dv_C(t)}{dt} + u_0\gamma_M(t) \quad (5.14)$$

approximating:

$$\beta(t) = \arctan \left(\frac{v(t)}{u_0} \right) \simeq \frac{v(t)}{u_0} \quad (5.15)$$

The additional information on corrected lateral velocity is obtained by Correvit measurements $v_M(t)$, corrected of the yawrate term due to the longitudinal displacement of the sensor at L_f distance from vehicle CoG:

$$v_C(t) = v_M(t) - L_f\gamma_M(t) \quad (5.16)$$

In order to complete the estimation of cornering stiffnesses, it is necessary to determine tyre sideslips. These are obtained using definition 5.7:

$$\begin{aligned} \alpha_{fE}(t) &= \frac{v_C(t) + L_f\gamma_M(t)}{u_0} - \delta_{fM}(t) \\ \alpha_{rE}(t) &= \frac{v_C(t) - L_r\gamma_M(t)}{u_0} - \delta_{rM}(t) \end{aligned} \quad (5.17)$$

where the measurements of real steering angles δ_{fM} and δ_{rM} are directly output by EPS. Estimated quantities are represented in Figure 5.5 For lateral forces estimation, the direct measurement of $a_{yM}(t)$ has been used instead of equation 5.14, since it is less noisy. In the lower part of Figure 5.5, the identified values are represented: cornering stiffnesses C_{yf} and C_{yr} on the left, friction coefficient derivatives on the right K_{yf} and K_{yr} , considering no load transfer. Only the central values of the dataset are used, which correspond to the steering manoeuvre, and RLS algorithm (Recursive Least Squares) has been used to find those unique values which best fit relation 5.12, finding these values:

$$\begin{aligned} C_{yfE} &= 15835 \text{ [N/rad]} \\ C_{yrE} &= 36380 \text{ [N/rad]} \\ K_{yfE} &= 4.60 \text{ [-]} \\ K_{yrE} &= 7.42 \text{ [-]} \end{aligned} \quad (5.18)$$

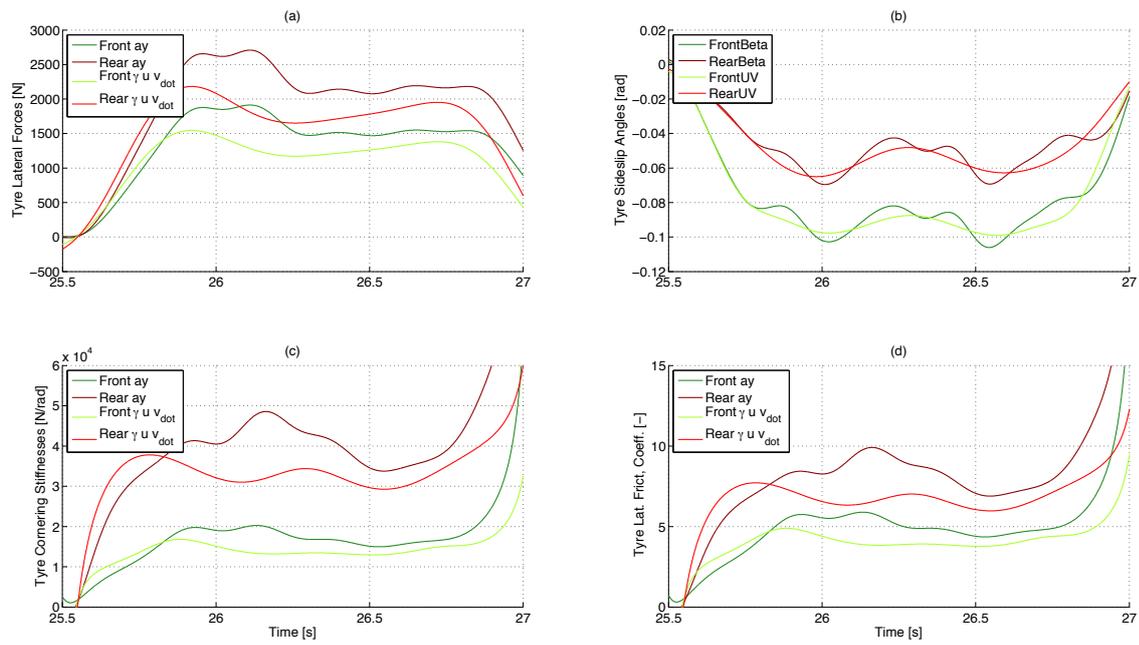


Figure 5.5: Identification of cornering stiffnesses: estimated lateral forces, estimated sideslips, final cornering stiffnesses and lateral friction slopes $K_{y,f,r}$ in the hypothesis of no load transfer

The identification process is quite rough, but it will be demonstrated that it has been sufficient to find results accurate enough to plan manoeuvres reproducible by the real vehicle.

Steering dynamics

Steering dynamics have been modelled as of second order, according to equation 5.8. However, this solution has not been determined a priori, but it is the result of an identification process, exploiting again the RLS algorithm. Helped by the System Identification Toolbox of Matlab, first order and second order models have been tested, looking for that which best resembled the lag in steering angle generation which was evident in logged data, e.g. the lane change showed in Figure 5.6. The RLS algorithm has been applied only to the first steering portion of the manoeuvre, between 3 and 5 seconds approximately. Identification results

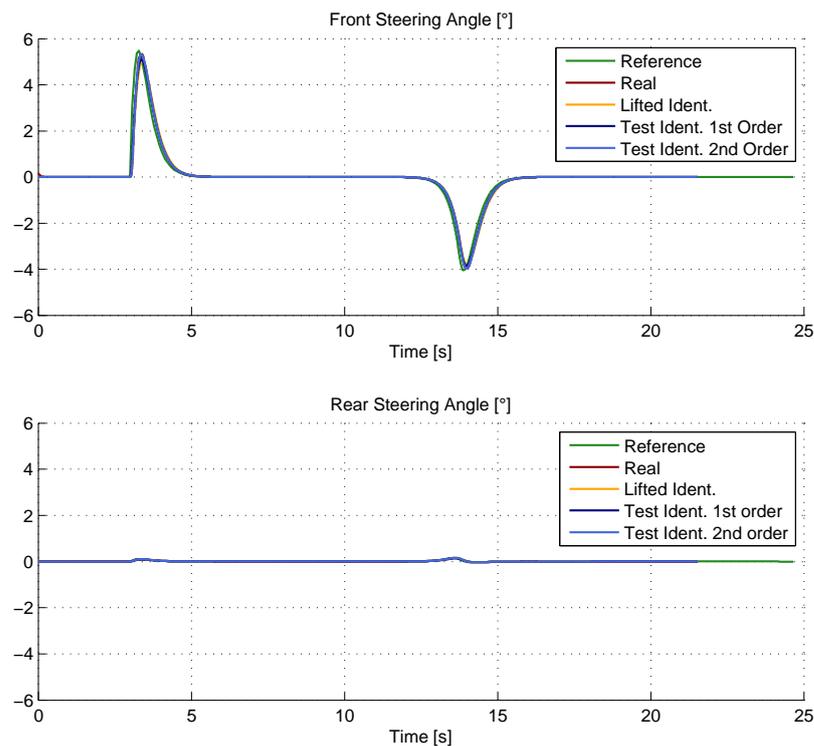


Figure 5.6: Lane change manoeuvre with rear steering

are portrayed in detail in Figure 5.7. The reference value is plotted in green, while real steering angles measured by EPS are showed in red, so that it is evident that the control and actuation chain causes a lag in steering tracking. A first order model previously developed within the laboratory is showed in yellow, but it was identified when the vehicle was lifted on blocks, with its tyres not attached to the ground. A new first order model has been identified using least squares, which showed better behaviour both for front and rear steer, but which was not satisfactory still. For this reason, a new second order model has been then used and identified, which showed much better behaviour, so that it has been used in the model. The final transfer function has been:

$$\begin{aligned}\frac{\delta_f(s)}{\delta_{0f}(s)} &= \frac{k_f}{s^2 + a_f s + b_f} \\ \frac{\delta_r(s)}{\delta_{0r}(s)} &= \frac{k_r}{s^2 + a_r s + b_r}\end{aligned}\tag{5.19}$$

where the identified parameters have been for front and rear axles:

$$\begin{aligned}k_f &= 563.4 \\ a_f &= 46.0 \\ b_f &= 563.8 \\ k_r &= 556.6 \\ a_r &= 34.4 \\ b_r &= 527.6\end{aligned}\tag{5.20}$$

5.3.3 Example optimal manoeuvres for Kanon: double lane changes

This section focuses the practical calculation of some obstacle avoidance manoeuvres on Kanon model to avoid the obstacle and return at the same lateral displacement, i.e. double lane changes as that portrayed in Figure 5.8. In particular, it is explained which settings have been used to obtain them referring to the OCP formulation of section 4.3.3. These manoeuvres are not those which have been then really executed by the vehicle, but they allow deriving some considerations in a more evident way.

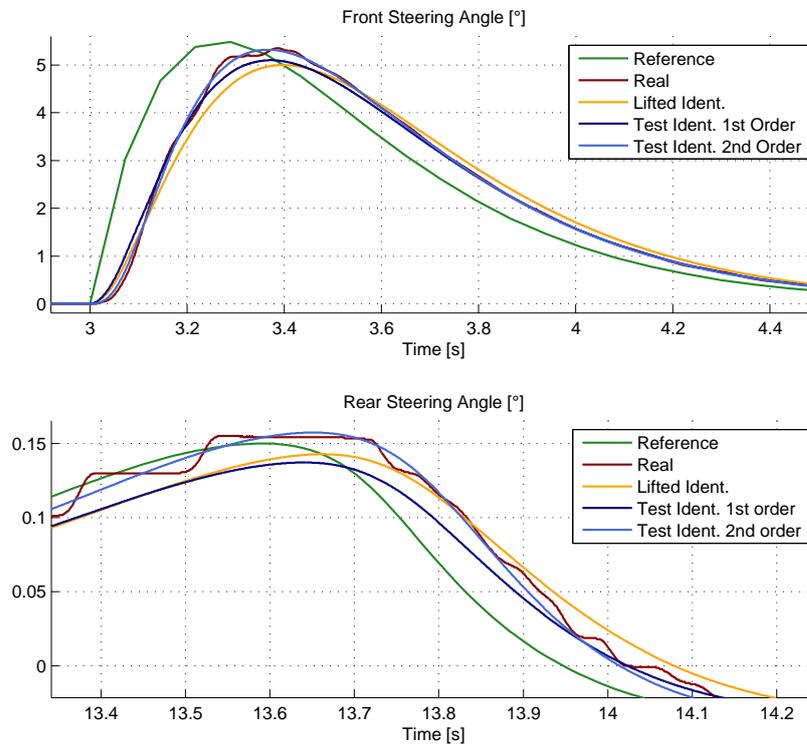


Figure 5.7: Steering dynamics identification: comparison between reference value, real value, and identified models of the first order (with vehicle lifted on blocks or attached to ground) and second order

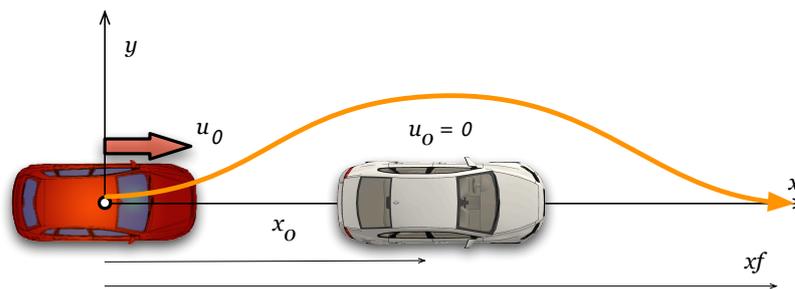


Figure 5.8: Example of obstacle avoidance manoeuvre with null final lateral position

Figure 5.8 represents a scenario where ego vehicle is travelling straight, and it approaches a stationary obstacle. If an ADAS such as that of section 5.2.3 is active on ego vehicle, and the driver is taking no action to avoid the collision, the system could judge that an intervention is needed when the vehicle is travelling at u_0 longitudinal speed, and the obstacle is x_o metres ahead (as described in section 5.4). In that instant, autonomous intervention is enabled and if traffic and road conditions are favourable, the manoeuvre of Figure 5.8, stored in the database, is executed.

The first step to obtain that desired manoeuvre is to set the described model equations (from 5.3 to 5.8) as ODE constraints in ζ domain, adding also that for the final time T , as described in equation 4.6. For this change of variables, in the dynamics equations it is only necessary to substitute t with ζ since they are proportional, and to multiply the contribution of equation 4.7 to each derivative, reminding that, given a general variable y and a function of its $f(y)$, when a new variable $z(y)$ is used the relation is:

$$\frac{df(y)}{dy} = \frac{df(z(y))}{dz} \frac{dz}{dy} \quad (5.21)$$

After this step, these simple boundary conditions were set:

$$b(x(a), p(a)) = \left\{ \begin{array}{l} \beta(0) = 0, \gamma(0) = 0, x(0) = 0, y(0) = 0, \psi(0) = 0, \\ F_{yf}(0) = 0, F_{yr}(0) = 0, \\ \left. \begin{array}{l} \frac{d\delta_f(\zeta)}{d\zeta} \Big|_{\zeta=0} = 0, \frac{d\delta_r(\zeta)}{d\zeta} \Big|_{\zeta=0} = 0, \delta_f(0) = 0, \delta_r(0) = 0, \\ T(0) = \text{free} \end{array} \right\}^T \quad (5.22)$$

$$e(x(b), p(b)) = \left\{ \begin{array}{l} \beta(1) = 0, \gamma(1) = 0, x(1) = x_f, y(1) = 0, \psi(1) = 0, \\ F_{yf}(1) = 0, F_{yr}(1) = 0, \\ \left. \begin{array}{l} \frac{d\delta_f(\zeta)}{d\zeta} \Big|_{\zeta=1} = 0, \frac{d\delta_r(\zeta)}{d\zeta} \Big|_{\zeta=1} = 0, \delta_f(1) = 0, \delta_r(1) = 0, \\ T(1) = \text{free} \end{array} \right\}^T \quad (5.23)$$

Each scenario can be described at each time step in a fixed reference frame, whose origin is placed at ego vehicle initial CoG, so that initial position and orientations are always null. If the vehicle is travelling straight, also all the other initial values are null as well. In an analogue way, if the goal is to finish the manoeuvre with the same lateral displacement and oriented in the same way, final conditions must be set null as well, with the exception of final longitudinal displacement, which can be set at any $x_f > x_o$, according to the user's desire.

The constraints used are similar to those of the formulation in section 4.3.2, here reported for completeness. One is the limit on minimum distance between vehicle and obstacle (i.e. equation (4.19), Figure 4.15)

$$d(t) \geq (r_V + r_O) \quad (5.24)$$

Another one is analogous to equation 4.21 to limit the steering angles, but here the limit is applied to the real ones instead of to the references, since the second order dynamics could cause some overshoots with respect to the references:

$$\begin{aligned} \delta_{fMin} &\leq \delta_f(t) \leq \delta_{fMax} \\ \delta_{rMin} &\leq \delta_r(t) \leq \delta_{rMax} \end{aligned} \quad (5.25)$$

In addition, the tyre model 4.13 used here is linear, and the saturation of the forces is not taken into account, as it happens in more complex models such as Pacejka's used in section 4.3.3. This means that the tyres would provide unlimited force, i.e. all that necessary to perform even the hardest manoeuvres. To avoid this unrealistic behaviour, one last constraint has been put on tyre frictions $\mu_{f,r}$, in an analogue way to the brake OCP case of section 4.3.2 (equation (4.8)), where no longitudinal tyre model was considered:

$$\begin{aligned} -1 &\leq \mu_f(t) \leq 1 \\ -1 &\leq \mu_r(t) \leq 1 \end{aligned} \quad (5.26)$$

where

$$\begin{aligned} \mu_f(t) &= F_{yf}/N_f \\ \mu_r(t) &= F_{yr}/N_r \end{aligned} \quad (5.27)$$

Those described above are the constraints included in the OCP solved to generate some test manoeuvres. It is interesting to note that there is no equivalent of

equation (4.23) to limit steering wheel velocity: here it is assumed that the manoeuvre will be executed by the automatic control, which is thought with much higher actuation capabilities.

Finally, since here the aim is to move from the theoretical analysis of previous chapter towards a real implementation, it would possible to add more constraints to really handle the scenario, e.g. keep vehicle's trajectory into road limits, etc., but this is not comprehended in the objective of the function.

All the settings above are already sufficient to obtain a manoeuvre which starts at $t = 0$ from current vehicle state and finishes at $t = T$ with desired final state, avoiding the obstacle in the middle and remaining within the physical limits of the vehicle. However, there would be at the moment several ways to accomplish this task: the desired one should be described setting the objective function, the only function among equations B.1 to be still defined. The target for the Bolza problem can be defined as:

$$w_\beta \left(\int_0^1 \beta^2(\zeta) T(\zeta) d\zeta \right) + w_T (T(1)) \quad (5.28)$$

i.e. the same objective of equation 4.16 after setting $w_\psi = 0$ and $w_y = 0$, thus requiring strict observance of final conditions set in equation 5.22. Namely, the objective is the weighted sum of a Lagrange term which tries to minimise sideslip β along the whole manoeuvre, and a Mayer term which requires minimum final time T . w_β and w_T are the weights of the two contributions and in general they must be carefully tuned to obtain the desired behaviour. In the Lagrange term, the square of lateral speed is necessary to require minimum velocities both towards the left (positive) and the right (negative). The multiplication of the integral by $T(\zeta)$ is a necessary scaling factor when changing the differentials from dt to $d\zeta$. This objective is set to obtain low lateral velocities, so that the manoeuvre is easier to be tracked, but it is also required to perform it in a short time, to resemble that it is still an emergency manoeuvre. The difference between the two contributions is clear when one of the two weights is alternatively set to 0. The results of these OCP, which consider $u_0 = 20 \text{ km/h}$, $x_o = 20 \text{ m}$, $x_f = 40 \text{ m}$ and $r_O = r_V = 1.5 \text{ m}$ are reported in Figure 5.9, showing the optimal trajectories and the steering angle time histories necessary to obtain them. It is interesting to note that in the first case, with only minimum time requirements, the trajectory is shorter: since the manoeuvre is performed at constant speed, the only way to

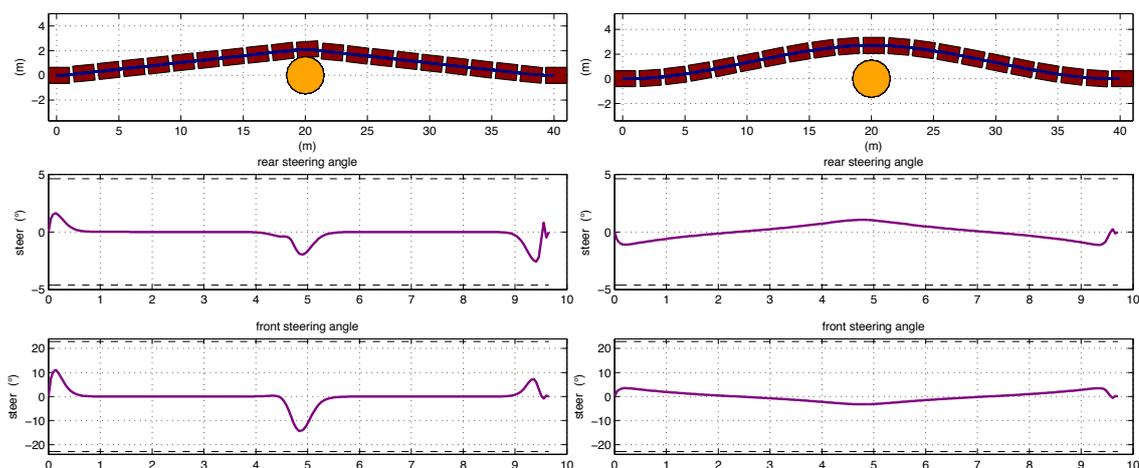


Figure 5.9: Avoidance manoeuvres for minimum time (left) and minimum lateral velocity (right)

reduce manoeuvring time is to travel the least distance possible. This is obtained concentrating the whole steering in a short time, at the beginning, around the obstacle and at the end, and thus travelling straight for the rest of the manoeuvre, as it is evident by the history of the steering angles and the shape of the trajectory respectively. On the other hand, when minimum lateral velocity is required, the trajectory is much smoother, and so the steering angles are. Another interesting aspect is the sign of the steering angles: in the first case they are in phase, to generate high lateral translations and shorten the path, but causing high sideslips, while in the second case they have opposite sign, in order to yaw as much as possible to lower vehicle sideslip.

5.4 THREAT ASSESSMENT

It was already described that a system like this basically needs a threat assessment module to enable the autonomous execution of emergency manoeuvres, and a control loop to make the vehicle physically realise the desired manoeuvre. Concerning the threat assessment, the strategy imagined looking at the driving simulator tests on naturalistic driving (section 4.2.3) can be used. Namely, since drivers showed to be jerk limited, initial manoeuvre jerks could be used as the

parameters for risk assessment: if database manoeuvres require jerks beyond a given threshold, the drivers would not be able to perform them, and the vehicle could take over the control of the vehicle.

The mechanism of real-time threat assessment was not implemented on the Kanon, so that a tuning the thresholds could not be done. However, some first attempt numerical values will be assigned here, just to give a hint on how it would work. For the case of steering manoeuvres, it is reasonable to assume that values of $j_\delta > 500 - 700$ deg/s as not feasible by human drivers. Thus, an ADAS should issue the warnings at a lower threshold, when safe manoeuvres require velocities higher than $j_{\delta,Warn} > 400$ deg/s, and then enable the autonomous intervention when these will require values of $j_\delta > 700$ deg/s.

Even if it not the purpose of this system, an analogue approach could be used for risk assessment of longitudinal avoidance manoeuvres. For instance, assuming that drivers are not able to apply longitudinal jerks higher than $\dot{j}_{p,Max} = 2.5$ s⁻¹, warnings could be issued at $\dot{j}_{p,Warn} = 2$ s⁻¹, and autonomous intervention at $\dot{j}_{p,Max}$.

5.5 CONTROL ALGORITHMS

5.5.1 *General strategies for trajectory tracking*

Another important feature of the proposed approach is the control algorithm to obtain the desired planned manoeuvre. Looking at the participants of DARPA Urban Challenge, the Talos vehicle [96] used for instance a simple PI controller for velocity tracking, and a modified version of the pure pursuit control law (Kelly and Stentz, 1997 [92] and Park et al., 2007 [118]) for the steering angle. The control law was defined as:

$$\delta_f = -\tan^{-1} \left(\frac{L \sin \eta}{\frac{L_1}{2} + l_a \cos \eta} \right) \quad (5.29)$$

where l_a is the distance between the pure pursuit anchor point and the rear axle, η is the angle between vehicle heading and the reference path direction, and L_1 is the look-ahead distance. A similar approach is used in Hoffmann et al., 2007 [82], even if the control law for the steering is slightly different, with a term designed to have null regime errors and a correction proportional to yaw rate

error.

Other approaches use Model Predictive Control (Mayne et al, 2000 [104], Falcone et al. 2007 [63], Falcone et al. 2009 [64]) as a Receding Horizon Control (Boyd et al. 2011 [103]), applying only the first control value for each plan, and then make a new plan at the following step for the new situation and actuate it only for the first value, in a cyclic way.

The novel approach hereby proposed proves that desired trajectory is also obtainable by properly tracking yaw rate and sideslip profiles (Ellefsen, 2012 [60]), in a robust way with respect to model uncertainties.

5.5.2 *Early algorithms*

Since the solution of optimal control includes also the control inputs to obtain the desired behaviour according to the model, the first trial has been to simply use the optimal profiles for $\delta_{0f}(t)$ and $\delta_{0r}(t)$ as inputs to the vehicle, in open loop. The inner loop of the EPS was then in charge of making the steering angles track the references. However, despite the identification process to make the model as close as possible to real behaviour, the final trajectory did not resemble the desired one of Figure 5.16, and after passing the obstacle the vehicle remained with a heading $\psi > 0$, so that final lateral position was $y_f > 3$.

The next step has been to try adding a closed loop contribution to the reference angles, using a PID controller on the error on the yaw rate $\gamma(t)$, as shown in Figure 5.10. Since there was only one feedback contribution for two inputs, this has been added to both, with a scaling factor k for the rear to minimise sideslip angle, as described in Abe, 2009 [28], pp. 222. The control loop was first simulated using Matlab, after exporting there the model developed in Maple. The simulated yaw rate showed almost perfect tracking of the reference (Figure 5.11). Once this control loop has been implemented on the vehicle, the tracking of yaw rate has showed better behaviour than the simple open loop, but this did not improve the tracking of final trajectory, which in the end was very similar to that of the open loop configuration. Figure 5.11 shows this situation. Yaw rate tracking is portrayed on the left part, where reference and measured yaw rates show a good matching, while the trajectory logged with a GPS on the desired path (in green) and the real path (yellow), besides the sensor errors, have a completely different

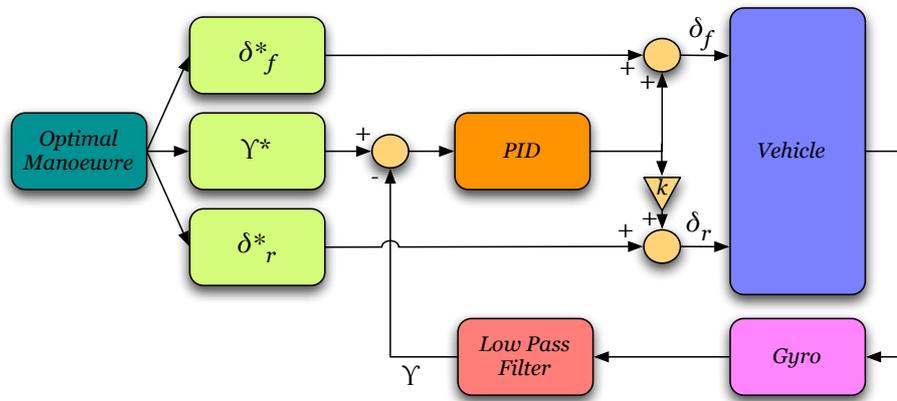


Figure 5.10: Simple control algorithm with feedback on only yaw rate

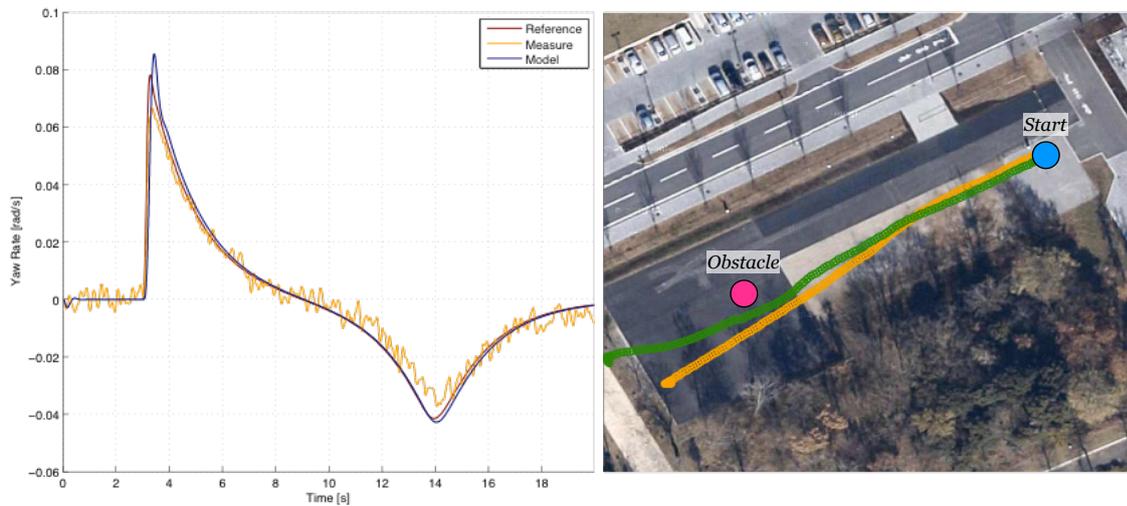


Figure 5.11: Sufficient tracking of yaw rate on the left, which does not result in a sufficient matching of desired trajectory (first lane change of the green path) and real trajectory (yellow) on the right.

shape, and the drift towards the left is evident. This is due to the fact that, on the other hand, vehicle sideslip, without any feedback, was not tracked at all. This is visible in Figure 5.12.

In order to obtain better results, a new control algorithm has been tried, adding

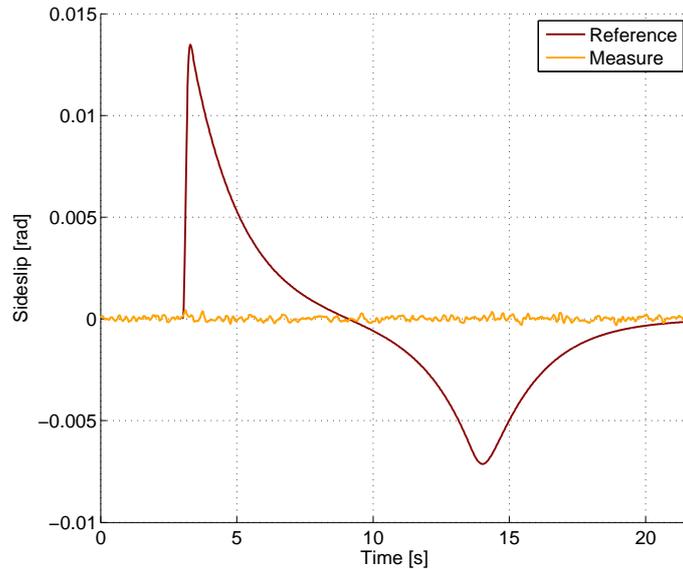


Figure 5.12: Bad tracking of sideslip angle β without feedback

feedback also on vehicle sideslip. In this way, two feedbacks are needed, i.e. yaw rate γ and sideslip angle β , and two vehicle inputs were available, i.e. δ_{0f} and δ_{0r} . A possible technique to handle this situation was the traditional decoupling control. The general concept of this technique is to use one input to control one variable, and the other input to control the other variable.

In particular, it has been chosen to use front steering to control yaw rate γ , and rear steering to control vehicle sideslip β , and the control loop shown in Figure 5.13. The choice has not been casual: looking at transfer functions between the two inputs and the two outputs in a linearised model (in an analogue way to the Relative Gain Array technique [39]), it was clear that while the sensibility of yaw rate to both the inputs is almost the same, sideslip angle is much more sensible to rear steering.

The key aspect of this approach is the definition of the decoupling transfer

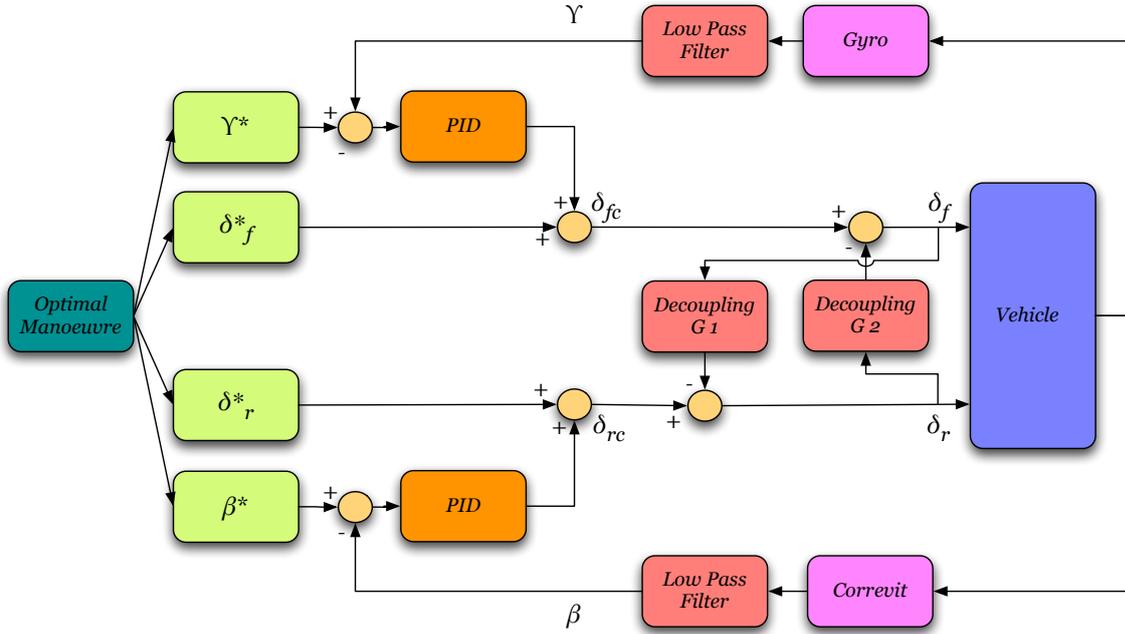


Figure 5.13: Control algorithm with decoupling control

functions G_1 and G_2 . The idea is that front steering is the sum of the optimal one and the correction due to the feedback on yaw rate, but it has also an effect on sideslip, and the same interference happens for rear steering, which has an effect on yaw rate. This superimposition of effects could lead to instability, so that it is necessary to neutralise the effect of each input on the variable it is not designed to control. Defining the problem in mathematical terms, the input-output relations are:

$$\begin{aligned}\Gamma(s) &= F_{\delta_f \rightarrow \gamma}(s)\Delta_f(s) + F_{\delta_r \rightarrow \gamma}(s)\Delta_r(s) \\ B(s) &= F_{\delta_f \rightarrow \beta}(s)\Delta_f(s) + F_{\delta_r \rightarrow \beta}(s)\Delta_r(s)\end{aligned}\quad (5.30)$$

and the objective would be to obtain a system so that:

$$\begin{aligned}\Gamma(s) &= F_{\delta_f \rightarrow \gamma}(s)\Delta_f(s) \\ B(s) &= F_{\delta_r \rightarrow \beta}(s)\Delta_r(s)\end{aligned}\quad (5.31)$$

The control scheme in Figure 5.13 can be a solution to obtain this desired behaviour. In fact, in that way the inputs to the model are:

$$\begin{aligned}\Delta_f(s) &= \Delta_{fc}(s) - G_2(s)\Delta_r(s) \\ \Delta_r(s) &= \Delta_{rc}(s) - G_1(s)\Delta_f(s)\end{aligned}\tag{5.32}$$

so that Equations 5.30 become:

$$\begin{aligned}\Gamma(s) &= F_{\delta_f \rightarrow \gamma}(s)\Delta_{fc}(s) + (F_{\delta_r \rightarrow \gamma}(s) - F_{\delta_f \rightarrow \gamma}(s)G_2(s))\Delta_r(s) \\ B(s) &= F_{\delta_r \rightarrow \beta}(s)\Delta_{rc}(s) + (F_{\delta_f \rightarrow \beta}(s) - F_{\delta_r \rightarrow \beta}(s)G_1(s))\Delta_f(s)\end{aligned}$$

In this way, defining the decoupling transfer functions $G_1(s)$ and $G_2(s)$ so that:

$$\begin{aligned}G_1(s) &= F_{\delta_r \rightarrow \beta}^{-1}(s)F_{\delta_f \rightarrow \beta}(s) \\ G_2(s) &= F_{\delta_f \rightarrow \gamma}^{-1}(s)F_{\delta_r \rightarrow \gamma}(s)\end{aligned}\tag{5.33}$$

the contribution of the undesired inputs are compensated, and the desired behaviour of Equation 5.31 is obtained.

Using this algorithm, a good overall behaviour has been obtained, with final trajectory resembling the reference. However, the control loop generated oscillating steering angles, as it is shown in Figure 5.14 for front steering, which is not acceptable. Its behaviour is due to errors in Correvit measurements, and also to uncertainties in the models, which make the compensations of equations 5.1 not accurate. For this reason, decoupling control has not been chosen as the final algorithm, and the Correvit sensor has been replaced by a proper state observer, which has given the double advantage of stabilising the estimation of β input to the controller, and replacing the usage of an expensive sensor. The final algorithm is described in next section.

5.5.3 *Final algorithm*

In order to fix the sensibility of the decoupling control to model errors, a new approach has been chosen, based on the inversion of vehicle model dynamics and

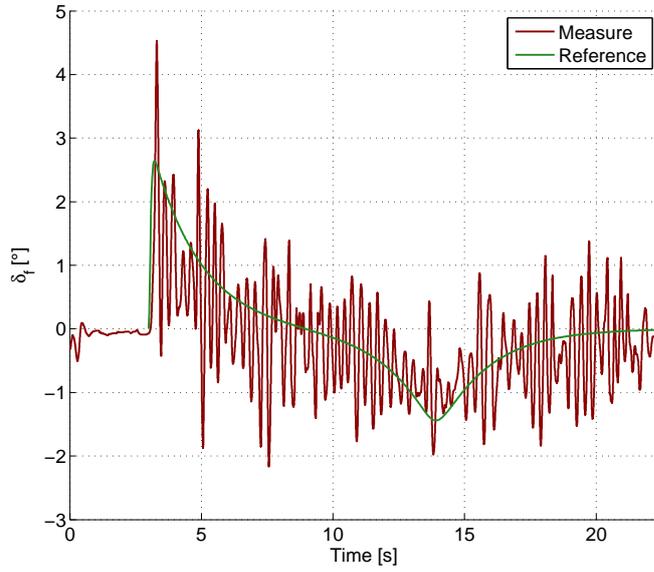


Figure 5.14: Oscillating front steering angle generated by decoupling control

on disturbance observers. The method has been already experimented within the Hori-Fujimoto Laboratory, even if applied to different models (Fujimoto, 2011 [73], Fujimoto, 2010 [75], or Yamauchi and Fujimoto, 2008 [142]). Their inputs were front steering δ_f and yaw moment N_z , given by the driving and braking torques applied to the wheels, instead of front and rear steering angles. In the cited papers, this method proved to provide better disturbance rejection than decoupling control, and has thus be adapted to the new model, to control rear steering instead of yaw moment generated by the torques. A sideslip observer has been also introduced to replace Correvit measurements, and reference steering angles coming from optimal control are not taken into account anymore. The final control scheme is reported in Figure 5.15. The first step of the algorithm is to use the optimal references in yaw rate γ^* and sideslip β^* to generate correspondent references on total yaw moment M_z^{**} and total lateral forces F_y^{**} . This is done

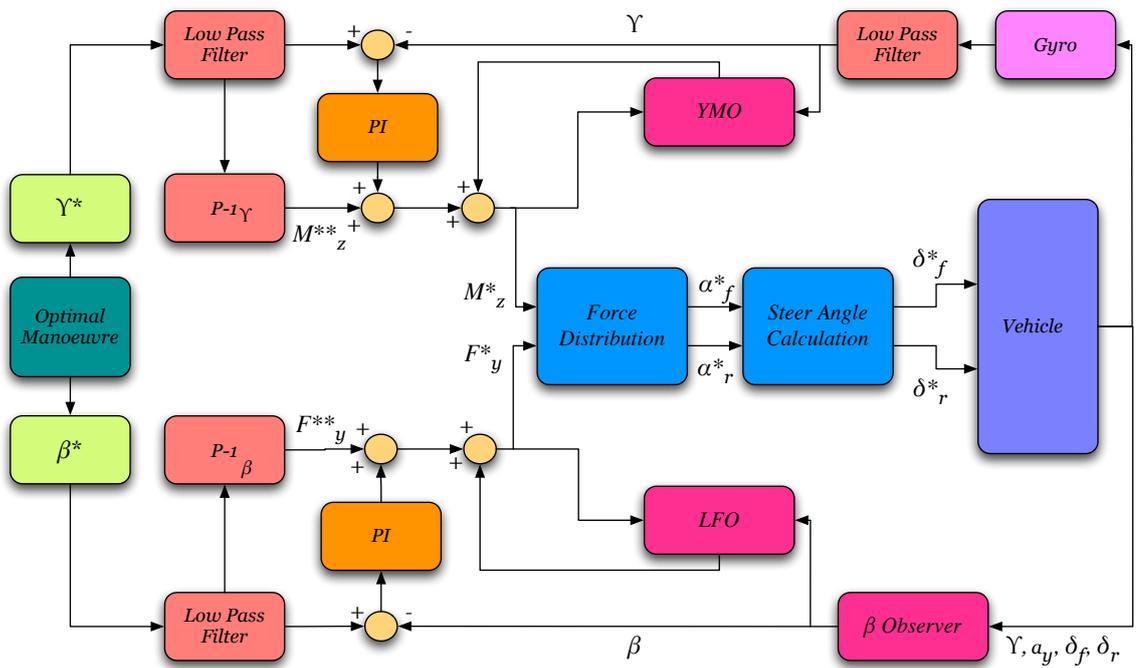


Figure 5.15: Final control algorithm, based on optimal manoeuvre references corrected by PI feedbacks and disturbance observers

by inverting the transfer functions $P_\gamma(s)$ and $P_\beta(s)$ easily derivable from vehicle model, where:

$$\begin{aligned}\Gamma(s) &= P_\gamma(s)M_z(s) \\ B(s) &= P_\beta(s)F_y(s)\end{aligned}\tag{5.34}$$

The references are then filtered, so that multiplying them by the improper fractions $P_\gamma^{-1}(s)$ and $P_\beta^{-1}(s)$, proper transfer functions are obtained again. These references are then corrected with two contributions: one from a PI controller on errors with real variables (i.e. measured yaw rate and observed sideslip), and another one generated by the disturbance observers developed in the already cited works ([73],[75],[142]), one for yaw moment (YMO) and one for lateral force (LFO). The references M_z^* and F_y^* obtained in this way are then used to generate references on sideslip angles α_f^* and α_r^* by substituting tyre model 5.6 into the right hand side of vehicle dynamics equations 5.3, and setting them equal to F_y and M_z respectively. The “Force Distribution” block solves these equations for α_f and α_r and gives the references on tyre sideslips, in an analogue way to the process described by Fujimoto, 2011 [74], but neglecting yaw moment N_z . The “Steer Angle Calculation” block then substitutes the sideslips into their definitions 5.7, inverts the relations and solves them for steering angles. These are the final values for the controls δ_f and δ_r , which are then input to the vehicle.

The feedback loops use filtered measures from the gyroscope for yaw rate γ , and an observed value for sideslip angle β instead of the direct measurement from the Correvit sensor. As a matter of fact, a linear observer has been designed for β basing on that developed by Aoki et al., 2006 [34], adapting again the model to the different vehicle inputs, where δ_r has replaced N_z . In particular, the observer is described by the same equations:

$$\begin{aligned}\dot{\hat{x}}(t) &= A\hat{x}(t) + Bu(t) - K(\hat{y}(t) - y(t)) \\ \hat{y}(t) &= C\hat{x}(t) + Du(t)\end{aligned}\tag{5.35}$$

where:

$$x = \begin{Bmatrix} \beta \\ \gamma \end{Bmatrix}, y = \begin{Bmatrix} \gamma \\ a_y \end{Bmatrix}, u = \begin{Bmatrix} \delta_f \\ \delta_r \end{Bmatrix}\tag{5.36}$$

and the hat sign stands for estimated values. In other words, measurements on yaw rate γ and lateral acceleration a_y are used to estimate vehicle sideslip, on the basis of the developed model, condensed in the matrices A , B , C and D . Gain matrix K has been chosen to maximise robustness, with a procedure analogue to that described in [34]. The resulting expressions are quite long, and it is useless to report them here. On the other hand, tuned values of:

$$\begin{aligned}\lambda_1 &= -35 \\ \lambda_2 &= -30\end{aligned}\tag{5.37}$$

have been set for the observer poles, which must be faster than the outer feedback dynamics. This algorithm has led to good results in trajectory tracking, which will be reported in section 5.6.

5.6 TESTS AND RESULTS

5.6.1 *Experiment details*

Real tests have been conducted on some sample manoeuvres, to prove the feasibility and the advantages of the proposed control method, based on OCP-generated references in yaw rate and sideslip, which can be properly tracked by a control loop based on disturbance observers and force distribution and result in the desired trajectory.

The real tests on the vehicle did not regard the complete ADAS, but only the control algorithm: the database of all the manoeuvres for all the scenarios has not been built, and neither the lookup algorithm for the choice of the closest to current situation. The threat assessment has not been integrated in the system, too.

The tests have been performed on simpler manoeuvres than those described

in previous section 5.3.3, which more resemble the single lane-change idea, as shown in Figure 5.16. To obtain them, test parameters have been set to:

$$\begin{aligned}
 x_f &= 40 \text{ m} \\
 y_f &= 3 \text{ m} \\
 x_o &= 30 \text{ m} \\
 r_1 &= 1.5 \text{ m} \\
 r_2 &= 1.5 \text{ m} \\
 u_0 &= 10 - 15 - 20 - 25 \text{ km/h}
 \end{aligned}
 \tag{5.38}$$

Namely, different manoeuvres have been calculated with the same obstacle

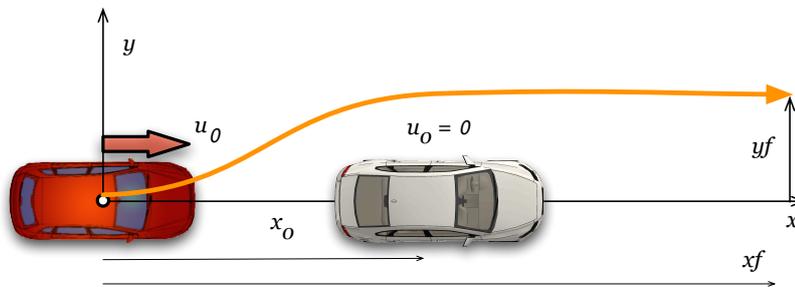


Figure 5.16: Manoeuvres used for real tests, with $x_o = 30 \text{ m}$, $x_f = 40 \text{ m}$ and $y_f = 3 \text{ m}$

position and characteristics, with the same final position for ego vehicle, but with increasingly demanding manoeuvre speed, from 10 km/h to 25 km/h . The objective function required minimum lateral velocity, with a very light contribution of minimum time, namely:

$$\begin{aligned}
 w_V &= 1 \\
 w_T &= 0.001
 \end{aligned}
 \tag{5.39}$$

One may object that these are not properly emergency manoeuvres, since the vehicle speed is not so high. This is true, but on the other hand the tests were only intended to define a method and prove its feasibility more than really building

the manoeuvre database for the real application. It is also true that when velocity increase, the method could be not valid anymore due to higher nonlinearities when the manoeuvre approaches vehicle limits. However, it was not possible to implement more demanding manoeuvres because of speed limits imposed to the vehicle by the authorities of Kashiwa-no-ha Campus, where the test track lays. Also the possibility of implementing double lane changes would have caused some complications, since the test track is hardly sufficiently long, so that there was not sufficient safety margin. Each manoeuvre in fact required also approximately 20 m before the beginning of the steering, in order to let the vehicle reach the desired constant speed in a stable way.

The test sequence was very simple: after compensating all the offsets in the sensors, the driver switched on an autonomous speed controller developed by the Hori-Fujimoto Laboratory, to make the vehicle reach and keep the desired speed. At a fixed spot 30 m before the imagined obstacle position, the driver simulated the output of a threat assessment algorithm which judged autonomous intervention necessary, turning on the reference manoeuvre availability and the EPS control to follow it. Forward velocity is kept constant in the meanwhile by the speed catch up algorithm. In the meanwhile, sensors are logged by a proper Controldesk sheet. At the end of the manoeuvre, the drivers turns off the EPS, finalises the logging and goes back to starting position, ready to repeat the procedure at increasing speed.

5.6.2 *Results*

The results have showed good tracking of the desired manoeuvre at velocities up to 20 km/h. At 25 km/h an initial instability probably due to a sensor offset happened, and for safety reasons it has not been tried to repeat the test, even after the error had been corrected. Steering angles and consequent variable tracking are showed in figures 5.17 and 5.18. Final good trajectory tracking is reported in Figure 5.19. Looking at steering angles, the improvement with respect to decoupling control is evident, with the rejection of the oscillation which was excessive instead in Figure 5.14. It is useful to remind that according to the control scheme of Figure 5.15, the optimal steering profiles are not used as references. However, in the front case the profiles are very similar, while in the

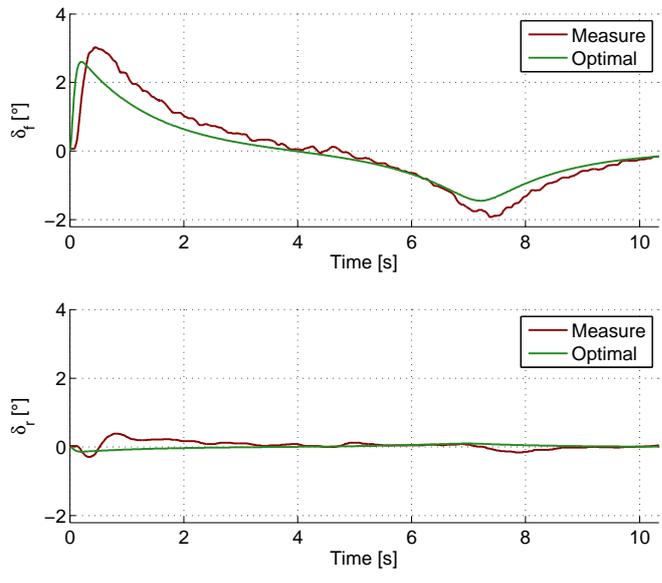


Figure 5.17: Steering angles on test for at 15 *kph*

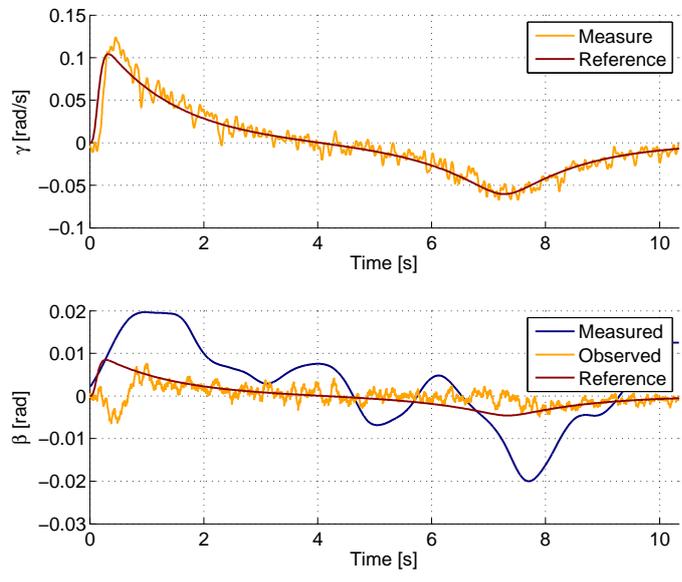


Figure 5.18: Variable tracking on test for at 15 *kph*

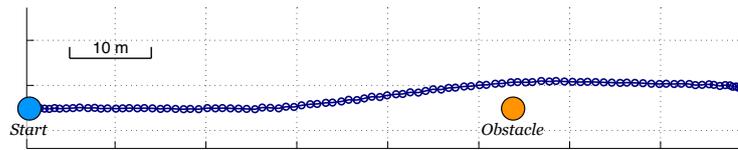


Figure 5.19: Trajectories

second case the control algorithm generates a different pattern.

The tracking of the variables is very good for γ , and satisfactory for β , even if, in this case, during the initial transitory the values do not match at all, and some further tuning would be required. The figure allows also to appreciate the advantage of using the β observer instead of the Correvit sensor, whose measurements oscillate excessively and overestimate the real value.

These results prove the feasibility of the approach of generating references with optimal control, which result in the desired real optimal manoeuvres when tracked by a proper control algorithms. In particular, it has been also proved that to obtain desired trajectory it is possible to only track yaw rate and sideslip profiles.

CONCLUSIONS

| | | |
|-----|--|-----|
| 6.1 | General considerations | 149 |
| 6.2 | A Codriver for driver warning | 150 |
| 6.3 | From driver warning to autonomous intervention | 152 |
| 6.4 | Autonomous collision avoidance | 153 |
| 6.5 | Future developments | 154 |

6.1 GENERAL CONSIDERATIONS

In this research work, two ADAS have been proposed, both based on optimal control and manoeuvre jerks as parameters for threat assessment.

The first is named “Codriver”, and is a system for driver warning. The second is a sort of completion of the first, since it is designed for autonomous vehicle intervention if the driver does not react to the warnings. The Codriver has been developed by the Mechatronics Group of the University of Trento, which the author is part of, in the framework of the European Project “interactIVe”, to warn the driver for all-around threats safety. It has been then implemented on a real vehicle of Centro Ricerche Fiat, which has been widely tested at the end of the project. On the other hand, for the second system only the main components have been developed by the author during a research period at the University of Tokyo, Japan, and its application is restricted to autonomous obstacle avoidance. In particular, a motion planning algorithm has been used together with a control loop designed to execute the planned trajectories.

Both systems exploit Optimal Control (OC) for motion planning: the Codriver uses OC to plan real-time manoeuvres with humanlike criteria, so that they can be compared to what the driver is doing in order to infer his/her intentions, and warn him if these are not safe; the second system uses OC instead to plan emergency manoeuvres, i.e. neglecting driver actuation limitations and pushing the vehicle towards its physical limits.

The initial longitudinal and lateral jerks of the planned manoeuvres are used by both the systems as parameters for risk assessment. A first premise to this is that manoeuvre jerks are proportional to the velocities of driver inputs, i.e. pedal velocity and steering wheel velocity, and their initial values describe the entity of the correction needed by the driver to achieve a given goal. The second premise is that it is known in literature, and it was confirmed by experimental tests conducted on the driving simulator of the University of Trento, that human drivers plan and act with minimum jerk criteria, and are jerk-limited. Thus, more and more severe manoeuvres would require higher and higher initial jerks, which at a given point could not be reachable anymore by a human driver. In this way, initial jerks can be considered proportional to the risk level of current situation. For this reason, when the manoeuvres to handle current scenario require jerks beyond a given threshold, the Codriver outputs a warning. This threshold must be lower than driver limits, so that he/she will be able to react to the warning and still have the chance to perform a safe manoeuvre. When the required jerks exceed drivers' actuation limits, the risk level raises to an upper step, where driver warning would be not effective and autonomous vehicle intervention should be enabled.

In obstacle avoidance scenarios, it was demonstrated during driving simulator tests that manoeuvre jerks are more robust parameters for risk assessment than for example time headways, since they are less affected by driver's age and gender.

6.2 A CODRIVER FOR DRIVER WARNING

The objective of the Codriver is to infer driver intentions, and warn him if these may lead to dangerous situations. The idea to achieve this purpose is to design it as an artificial driver, similar as much as possible to the real driver in his/her

decision processes for motion planning: if they have similar architectures, their states of mind will be similar, allowing better reciprocal understanding.

Thus, its architecture is based on perception-action hierarchical layers inspired by cognitive science theories, the higher settings goals for the lower, as it happens for a driver: from higher states of mind (e.g. being in a hurry), navigation plans are made (e.g. overtaking instead of following an object), which then set the objectives for decomposed simpler problems (i.e. steering to reach a desired lateral position and accelerating to reach a desired final velocity), for which the lowest layer finally finds the control laws necessary to obtain them (i.e. the pedal velocity and steering velocity profiles or, in other words, longitudinal and lateral jerks) by means of optimal control. Actually, several OCPs are evaluated by the Codriver in real-time, in order to cover a large variety of possible driver goals, each requiring certain initial jerks. Matching these jerks (i.e. Codriver inputs) with those the real driver is actuating, the Codriver is able then to understand which manoeuvre the driver intends to perform, and all the way up to the intentions which animate him/her.

The Codriver considers all around threats in its plans, addressing collisions, unintended lane departures, excessive curve velocities, bad compliances of speed limits and landmarks, i.e. it plans optimal manoeuvres which allow handling all these aspects, and assigns them a risk level: if the driver's identified goal requires too high jerks, a warning is output. Warnings can be yellow or red, depending on the risk level, i.e. on the jerk the manoeuvres require.

The Codriver is part of a Continuous Support function implemented on a Lancia Delta: it receives data from a perception platform in charge of environment reconstruction, it assesses the risk level of current situation and eventually the threat source as it was described, and then sends the warnings to an HMI management system which activates the warning interface, which exploits visual, acoustic and haptic feedback. The warning HMI has two levels of invasiveness, depending on the warning level (yellow or red).

The Continuous Support application has been widely tested within the interactIVe project, both from the technical (objective performance, i.e. warning timing etc.) and user-related point of view (i.e. warning acceptance, interface issues, etc.). The tests took place respectively in the "Centro Sicurezza" private proving ground of Centro Ricerche Fiat on predefined use cases, and on a public 53 *km* test track

featuring urban, extra-urban and motorway portions. The results have been satisfactory in both cases, proving the goodness of the approach, i.e. inference of driver intention exploiting a layered subsuming architecture, humanlike motion planning by means of optimal control, and threat assessment based on manoeuvre jerks.

6.3 FROM DRIVER WARNING TO AUTONOMOUS INTERVENTION

A general driver warning ADAS should warn the driver when the severity of safe manoeuvres for a given scenario reaches a given threshold, at which an action of the driver would still be possible and safe. However, if the driver does not react in a given time, he will be not able anymore to avoid the dangerous situation. However, it has been showed that at that point there would be still a margin for properly controlled actuators, with higher capabilities (i.e. bandwidth and accuracy), to perform safe manoeuvres within vehicle physical limits.

An unique ADAS could handle both the warning and the intervention phase, but it is not the case of the Codriver, which could not be used as it is for this purpose. It was demonstrated that the simplifications it makes (motivated by the need of solving multiple OCPs in real-time) lead to planned manoeuvres which do not resemble those obtainable in reality, especially for lateral dynamics. Neglecting nonlinearities, saturations, delays etc., the Codriver manoeuvres are in fact designed to resemble the model of vehicle dynamics the driver has in mind more than the real dynamics model. While this has proved to be a good approach for warning purposes, the manoeuvres it generates can not be used to directly control the vehicle. At most, they could be used as good guess functions to solve more complex OCPs, which should take into account also physical and other constraints, and which should be based on more representative vehicle dynamics models, in order to push the manoeuvres towards vehicle limits.

These more detailed OCPs have been set up focusing obstacle avoidance scenarios, by steering and braking, obtaining emergency manoeuvres which could be used as motion plans for autonomous intervention. In particular, it was proved that, at high relative velocities, steering emergency manoeuvres are more effective than braking manoeuvres to avoid an obstacle (the contrary happens at low velocity).

Being autonomous steering more interesting by the research point of view than autonomous braking (which is already available on the market), further analysis was conducted to see how to even improve its performance. In particular, it was showed that high avoidance performance can be obtained by autonomous steering manoeuvres if also rear steering wheel was available. In this case, lateral displacement can be obtained by almost pure translational manoeuvres (i.e. null yaw rate), with high lateral velocities.

6.4 AUTONOMOUS COLLISION AVOIDANCE

After theoretically demonstrating that superior steering performance obtainable by means of four-wheel steering (4WS), the activity at the University of Tokyo has regarded the development of an ADAS for 4WS autonomous obstacle avoidance at constant speed.

Even though the complete ADAS has not been entirely built, the feasibility of an approach based on motion planning by means of optimal control and manoeuvre execution by proper control loops has been proved. In particular, it has been demonstrated that the manoeuvres output by OCPs are representative of the real ones when the model is sufficiently accurate and its parameters are identified correctly.

From the control point of view, it has been proved that it is possible to obtain the desired trajectory by tracking reference profiles (solutions of the OCP) of yaw rate and sideslip angle, using front and rear steering angles as control inputs. In particular, this result was achieved using disturbance observers for required lateral force and yaw moment to compensate a feedback PID control on yaw rate and sideslip angle. In addition, a linear observer has been developed for sideslip estimation, necessary for the feedback control, which allowed to replace the expensive Correvit sensor and to obtain better results. This approach generated a simple and low-demanding control algorithm, which proved to be more stable and robust than traditional decoupling control.

The described control algorithm has been implemented on the experimental electric vehicle “FPEV2-Kanon” of the University of Tokyo, and successfully tested in lane-change-like minimum-sideslip obstacle avoidance manoeuvres at different velocities up to 20 km/h, obtaining in each case the desired trajectory.

6.5 FUTURE DEVELOPMENTS

In a wide research area as that covered by this research, there are several possible improvements and the aspects which would require further analysis.

Concerning the Codriver, it would be interesting to enrich the set of the possible manoeuvres it conceives, e.g. overtaking, which could be obtained concatenating more of its motor primitives. The boundary conditions the different portions meet at should be in their turn optimised solving another OCP as well, but extending its possible behaviour range should enhance its inference of driver intentions.

The Codriver could be also designed to enrich the interaction with the driver towards a peer-to-peer relationship, in a sort of horse-rider metaphor. Instead of only warning, it would be probably quite easy to implement a more advanced haptic HMI on the steering wheel and on the pedals, where the Codriver could also physically suggest to the driver safer manoeuvres etc.

Considering the collision avoidance analysis, it would be certainly interesting to see the performance of emergency manoeuvres which combine steer and brake. Thinking this issue in terms of time to collision (TTC), it would probably happen that this kind of manoeuvres lowers the limit curves, thus reducing the range of scenarios where collisions can not be avoided.

Many issues have remained opened also concerning the autonomous obstacle avoidance. First of all, it was showed that the tracking of the sideslip angle is not perfect still, and the cause should be investigated and removed.

After fixing this inaccuracy, the robustness of the control algorithm should be then tested on more scenarios, starting from minimum time manoeuvres, i.e. those which require almost translational behaviour and the highest lateral velocities.

Finally, the proposed threat assessment method should be tuned in its thresholds, and then the entire application should be built, enriching the database with all the possible scenarios, implementing a lookup algorithm and integrating the perception platform with additional sensors (e.g. radars, lidars and/or cameras) to perceive the obstacle in real time and send this information to the lookup module.

APPENDIX - LIST OF SYMBOLS

Table A.1: List of OCP Parameters

| Parameter | Unit | Meaning |
|------------------|-------------|--|
| x_o | [m] | Obstacle x position in cartesian ref. frame |
| y_o | [m] | Obstacle y position in cartesian ref. frame |
| $s_{s,o}$ | [m] | Obstacle long. position in curvilinear coordinates |
| $s_{n,o}$ | [m] | Obstacle lat. position in curvilinear coordinates |
| r_V | [m] | Ego vehicle bounding circle radius |
| r_O | [m] | Obstacle bounding circle radius |
| w_T | [–] | Weight for minimum time term in OCP objective |
| w_β | [–] | Weight for minimum sideslip term in OCP objective |
| w_ψ | [–] | Weight to relax final conditions on heading |
| w_y | [–] | Weight to relax final conditions on lateral pos. |

Table A.2: List of Vehicle Parameters

| Parameter | Unit | Meaning |
|----------------------|------------------|--|
| t | [s] | Time |
| g | [m/s^2] | Gravity |
| m | [kg] | Vehicle mass |
| I_z | [$kg\ m^2$] | Vehicle yaw moment of inertia |
| L_f | [m] | Distance from vehicle CoG to front axle |
| L_r | [m] | Distance from vehicle CoG to rear axle |
| L | [m] | Vehicle wheelbase |
| r_f | [m] | Vehicle front tyre radius |
| r_r | [m] | Vehicle rear tyre radius |
| kv | [$N\ s^2/m^2$] | Vehicle aerodynamic coefficient |
| n | [–] | Vehicle steering ratio between tyre and steering wheel |
| C_{yf} | [N/rad] | Vehicle front tyre cornering stiffness |
| C_{yr} | [N/rad] | Vehicle rear tyre cornering stiffness |
| K_{yf} | [rad_{-1}] | Vehicle front lateral friction slope |
| K_{yr} | [rad_{-1}] | Vehicle rear lateral friction slope |
| μ_{xf} | [–] | Vehicle front tyre longitudinal friction coeff. |
| μ_{xr} | [–] | Vehicle rear tyre longitudinal friction coeff. |
| μ_{yf} | [–] | Vehicle front tyre lateral friction coeff. |
| μ_{yr} | [–] | Vehicle rear tyre lateral friction coeff. |
| B_f, C_f, D_f, E_f | [–] | Front tyre Pacejka coefficients |
| B_r, C_r, D_r, E_r | [–] | Rear tyre Pacejka coefficients |
| k_{Tf} | [–] | Traction repartition on front axle coeff. |
| k_{Tr} | [–] | Traction repartition on rear axle coeff. |
| k_{Bf} | [–] | Braking repartition on front axle coeff. |
| k_{Br} | [–] | Braking repartition on rear axle coeff. |
| l_y | [s] | Time constant for first order lateral force dynamics |
| k_f | [–] | Gain for TF of front steering first order lag |
| k_r | [–] | Gain for TF of rear steering first order lag |
| p_f | [s] | Time constant for first order front steering dynamics |
| p_r | [s] | Time constant for first order rear steering dynamics |
| u_o | [m/s] | Ego vehicle constant velocity |

Table A.3: List of Vehicle Variables

| Parameter | Unit | Meaning |
|----------------|--------------------|--|
| $x(t)$ | [m] | Vehicle x position in cartesian coordinates |
| $y(t)$ | [m] | Vehicle y position in cartesian coordinates |
| $\psi(t)$ | [rad] | Vehicle yaw angle in cartesian coordinates |
| $s_s(t)$ | [m] | Vehicle longitudinal pos. in curvilinear coordinates |
| $s_n(t)$ | [m] | Vehicle lateral pos. in curvilinear coordinates |
| $\alpha(t)$ | [rad] | Vehicle yaw angle in curvilinear coordinates |
| $\beta(t)$ | [rad] | Vehicle sideslip angle |
| $u(t)$ | [m/s] | Vehicle longitudinal velocity |
| $v(t)$ | [m/s] | Vehicle lateral velocity |
| $\gamma(t)$ | [rad/s] | Vehicle yaw rate |
| $\Delta(t)$ | [m ⁻¹] | Vehicle trajectory curvature in curvilinear coord. |
| $\alpha_f(t)$ | [–] | Vehicle front tyre sideslip angle |
| $\alpha_r(t)$ | [–] | Vehicle rear tyre sideslip angle |
| $p(t)$ | [–] | Vehicle traction/braking pedal position |
| $\delta_f(t)$ | [rad] | Vehicle front steering angle |
| $\delta_r(t)$ | [rad] | Vehicle rear steering angle |
| $j_p(t)$ | [s ⁻¹] | Vehicle pedal velocity |
| $j_{delta}(t)$ | [rad/s] | Vehicle steering wheel velocity |
| $F_{xf,T}(t)$ | [N] | Vehicle front traction force |
| $F_{xr,T}(t)$ | [N] | Vehicle rear traction force |
| $F_{xf,B}(t)$ | [N] | Vehicle front braking force |
| $F_{xr,B}(t)$ | [N] | Vehicle rear braking force |
| $F_{yf}(t)$ | [N] | Vehicle front lateral force |
| $F_{yr}(t)$ | [N] | Vehicle rear lateral force |
| $F_{yf0}(t)$ | [N] | Vehicle front nominal lateral force |
| $F_{yr0}(t)$ | [N] | Vehicle rear nominal lateral force |
| $N_f(t)$ | [N] | Vehicle front vertical force |
| $N_r(t)$ | [N] | Vehicle rear vertical force |
| $T_{xf,T}(t)$ | [Nm] | Vehicle front traction torque |
| $T_{xr,T}(t)$ | [Nm] | Vehicle rear traction torque |
| $T_{xf,B}(t)$ | [Nm] | Vehicle front braking torque |
| $T_{xr,B}(t)$ | [Nm] | Vehicle rear braking torque |

APPENDIX - OPTIMAL CONTROL

| | | |
|-----|---|-----|
| B.1 | Overview | 159 |
| B.2 | Optimal control formulation and solution | 160 |
| B.3 | Tools for OCP developed by the University of Trento | 162 |

B.1 OVERVIEW

In the automotive field, optimal control has been used for a long time in many different applications, from powertrain to active suspensions or steering systems management, exploiting different techniques such as LQR, LQG, MPC and many others (Bryson, 2002 [46]). Optimal control can be used in all the components of an ADAS. In the Perception layer, optimal control could help estimating non-observable states, e.g. obstacle future evolution: given position and velocities of a vehicle ahead, minimum jerk motion can be hypothesised for it and future positions can be the solutions of the optimisation problem. On the Decision side, optimal control can find an interesting usage in warning systems, as described in Chapter 3. However, it is in the Action layer that Optimal Control usually finds its widest application. In fact, it is possible to use optimal manoeuvres as references for conventional control algorithms, but also to set up an optimal tracking of them, as it happens in MPC. This process can be also repeated at every time step, using only the first value of the output controls and then re-evaluating the optimal tracking, thus implementing a Receding Horizon Control. LQR and LQG are other optimal control techniques used for motion planning and control.

This appendix briefly describes Optimal Control principles for the generation of optimal manoeuvres which achieve a given goal, and, in particular, the approach used within the Mechatronics Group of the University of Trento to generate and solve the problem in an efficient and robust manner. However, further details on optimal control formulation and solving can be found in Bryson and Ho, 1975 [47], Troutman, 1996 [133] or Betts, 2009 [37], while for numerical solving techniques a recommended source is Nocedal and Wright, 2006 [115]. The approach used in this work for OCP solving, though summarised hereon, is explained in Bertolazzi et al., 2007 [35].

B.2 OPTIMAL CONTROL FORMULATION AND SOLUTION

A constrained optimisation problem can be defined in general as:

$$\begin{aligned}
 \text{find } u \text{ to minimise:} \quad & M(x(b), u(b), p(b)) + \int_a^b J(x(s), u(s), p(s)) \, ds \\
 \text{under the following constraints:} \quad & \\
 \text{ODE constraints:} \quad & A(x(s), p(s)) \dot{x} = f(x(s), u(s), p(s)), \\
 \text{Initial BC:} \quad & b(x(a), p(a)) = 0, \\
 \text{Final BC:} \quad & e(x(b), p(b)) = 0, \\
 \text{One side constraints:} \quad & g_k(x(s), u(s), p(s)) \leq 0, k \in I_1 \\
 \text{Two side constraints:} \quad & -1 \leq g_k(x(s), u(s), p(s)) \leq 1, k \in I_2
 \end{aligned} \tag{B.1}$$

where I_1 and I_2 are two index sets such that $I_1 \cap I_2 = \emptyset$.

Namely, the objective is to find proper state x and input u histories which minimise an objective function, subject to given constraints. When applied to motion planning for vehicles, the solution of the optimal control problem is a manoeuvre which reaches a desired final state, in a way which minimises the objective function. This definition of manoeuvre, as the solution of an OCP, includes both the evolution of vehicle states, e.g. its position, velocity, etc., and of the controls necessary to obtain them, e.g. steering angle, pedal positions, etc. This evolution can be intended in time domain or in space domain, depending on how the

problem is defined.

Looking in detail into equation B.1, the objective, called Bolza target, is given by the sum of two contributions: a Lagrange term, which is the integrand J along the whole planning horizon, and a Mayer term M , which is instead only a function of the final state. In other words, the objective can be to minimise either a quantity all over the manoeuvre, e.g. a manoeuvre which minimises jerk, or a quantity at the end, e.g. final lateral displacement, or a weighted sum of the two.

It is then possible to include different constraints. ODE constraints are used to include vehicle dynamics in the optimisation, to grant that the manoeuvre will be physically meaningful. Boundary conditions can be set as well for the states of the ODE system. In particular, for real-time motion planning the initial conditions are set equal to the current vehicle state, so that the manoeuvre will start from current situation. On the other hand, final conditions describe a final desired state at the end of the manoeuvre, e.g. a suitable final lateral displacement for lane-change manoeuvres, null final velocity for braking manoeuvres, etc. It is not necessary to set all the boundary conditions: sometimes it is difficult to specify requirements on final state, so that those constraints can be substituted by respective transversal conditions (details can be found in Troutman, [133], pp. 156). Finally, more constraints can be set in form of inequalities, both on vehicle states and on the controls, according to simulation purposes. For instance, it is possible to include the adherence limits of the tyre requiring lateral forces within a given range, the mechanical limits of the steering system setting steering angle lower than a certain threshold, or humanlike criteria on motion planning, setting requirements on combined vehicle accelerations [52] .

Once the problem is set, it is necessary to solve it. There are several ways to solve optimal control problems, which can be found in already mentioned sources ([133], [115]). A fast and robust indirect solution method has been developed at the University of Trento in the last years, which has proved to be effective also for real-time application, such as those of the PREVeNT ([36],[33]) and Michelangelo ([9]) projects.

The first step is to eliminate the constraints to transform the problem into uncon-

strained optimisation. For the inequalities, different penalty functions p_k are used, e.g. quadratic, logarithmic, etc., so that their contribution J_p :

$$J_p(x(s), u(s), p(s)) = \sum_k p_k(g_k(x(s), u(s), p(s))) \quad (\text{B.2})$$

can be added to the Lagrange term of the Bolza function:

$$\int_a^b (J(x(s), u(s), p(s)) + J_p(x(s), u(s), p(s))) ds \quad (\text{B.3})$$

For the equalities, i.e. ODE and BC constraints, Lagrange multipliers are used (again in Troutman, [133], pp. 160). Neglecting the transcription of the independent variable s , the final objective function F can be defined as:

$$\begin{aligned} F(x, u, v, \lambda, \mu, \nu) &= \int_a^b (J(x(s), u(s), p(s)) + J_p(x(s), u(s), p(s))) ds \\ &+ \int_a^b \lambda (A(x, p)\dot{x} - f(x, u, p)) ds \\ &+ \mu (b(x(a), p(a))) \\ &+ \nu (e(x(b), p(b))) \end{aligned} \quad (\text{B.4})$$

The problem can be now solved as an unconstrained optimisation problem, finding the stationary points of F . To do this, the first variation of F is calculated (e.g. in Hull, 2003 [88]), obtaining a boundary value problem (BVP), which is then discretised using finite differences (FD), e.g. in Nocedal, 2009 [115]. In this way, a nonlinear system is obtained, which can then be solved using optimised code which implements Newton-Broyden, Hyness and Conjugate Gradient algorithms.

B.3 TOOLS FOR OCP DEVELOPED BY THE UNIVERSITY OF TRENTO

The described steps can be automatically done using the tools developed at the University of Trento. In particular, a vehicle model can be developed with Maple, a software for symbolic math, which can be then used also for the generation of the OCP. In fact, the Mechatronics group has developed a Maple library named “XOptima”, which provides suitable commands to generate the complete problem. In this way it is possible to define the objective, and set the boundary conditions

and the constraints, including the ODE model previously developed, according to the user's need. It is also possible to preset problem parameters, e.g. model data, objective weights, constraint thresholds, penalty weights, boundary condition values etc, which can be also modified later, during the solution of the OCP. In fact, this library finally outputs C++ code which describes the problem, and CMakeLists instructions to automatically build projects in different environments, such as Mac OSX, Linux and Windows. These projects include also the "Mechatronics Framework", again developed at the University of Trento, which contains indispensable tools such as the interfaces with the described problem, the non-linear solvers, etc. Once the project is built, it is possible to run it and obtain the solution of the OCP. Different settings can be tried to achieve better solution without need of compiling the project again, thanks to suitable configuration files written in Ruby or Lua languages.

The results are then output in a plain text format, so that it is easy to import and plot them with several softwares the user may prefer, such as Matlab, Excel or Datagraph. Diagnostics information are also available among the outputs, to check the convergence steps of the algorithm if a solution cannot be obtained at the first attempt.

ACKNOWLEDGEMENTS

A first, deep thank goes to my tutor, Francesco Biral, who has supervised me from the bachelor degree through the master degree to this final achievement. To him goes my gratitude for the patience and the competence he has taught me with a large part of my professional knowledge, and, not last, for the friendship he has honoured me with in these last years. The remainder of my knowledge is owed to the other professors who have accompanied me during my PhD studies, especially Mauro Da Lio and Enrico Bertolazzi, and for the final part also Hiroshi Fujimoto, who has welcomed me at his lab and at his home as I was part of his family.

Other colleagues have however helped me with this work, contributing to its successful completion, among which I have to cite Fabrizio, Maximiliano, Antonio, Marco and Michele, for these years of reciprocal enrichment we spent together; Matteo for the help with the simulator and Andrea for reviewing my thesis scripts, as well as Andrea and Fabio at CRF for sharing their experience when working together in such a joyful manner; Binh Minh, for the precious advice during my Japanese experience and Yone for driving the Kanon with my applications loaded at his own risk. These thanks are also enlarged to all the other members of the Mechatronics Lab at the University of Trento and of the Hori-Fujimoto Lab at the University of Tokyo, of course.

A special thank goes now to my family: my parents, Laura and Guido, my sister's family, Marina, Franco and Giulia (and the future heir who is about to come), my girlfriend Carlotta and my aunt Adelina with my cousin Mariangela, and Carlotta's family, who always supported me with love and constant encouragement, without which these results would have never been reached.

Last, but not least, special thanks go also to all my friends, the guys from Montorio,

my former university mates, my flat mates in these last years, and the guys at the Kashiwa Lodge, with whom I have spent great and unforgettable hours which made my studies easier and more relaxed.

Thank you all, see you at my next step.

BIBLIOGRAPHY

- [1] Overview on bmw's dsc, 2013 . URL http://www.bmw.com/com/en/insights/technology/technology_guide/index_category.html?code=4&view=Safety.
- [2] Bosch automotive technology website, . URL http://www.bosch-automotivetechology.com/en/com/home/homepage_com.html.
- [3] European project diplects website, . URL <http://www.diplects.eu>.
- [4] Dynamotion website, . URL <http://www.dynamotion.it/eng/index.htm>.
- [5] Gidas database, . URL http://www.bast.de/nn_82220/EN/e-Aufgaben/e-abteilung-f/e-referat-f2/e-gidas/e-gidas.html.
- [6] Daimler active safety systems, . URL <http://www.daimler.com/technology-and-innovation/safety-technologies/prevention>.
- [7] Toyota safety technology, . URL http://www.toyota-global.com/innovation/safety_technology/safety_technology/technology_file/.
- [8] Volkswagen front assist system, . URL http://en.volkswagen.com/en/innovation-and-technology/technical-glossary/umfeldbeobachtungssystem_front_assist.html.
- [9] Michelangelo website, . URL http://www.questio.it/index.php?option=com_qu3&task=cnt.she&id=408&lang=it.
- [10] Prevent european project website, . URL <http://www.ertico.com/prevent>.

- [11] The darpa urban challenge, 2007. URL <http://archive.darpa.mil/grandchallenge/index.asp>.
- [12] Mercedes pre-safe brake, 2008. URL http://www.emercedesbenz.com/Nov08/12_001507_Mercedes_Benz_TecDay_Special_Feature_PRE_SAFE_And_PRE_SAFE_Brake.html.
- [13] esafety adas classification, 2009. URL http://ec.europa.eu/transport/road_safety/specialist/knowledge/pdf/esafety.pdf.
- [14] Honda cmbs system described on euro ncap website, 2010. URL http://www.euroncap.com/rewards/honda_collision_mitigation_brake_system.aspx.
- [15] Have-it european project website, 2011. URL <http://www.haveit-eu.org/displayITM1.asp?ITMID=6&LANG=EN>.
- [16] Meccablog - the mechatronics group at the university of trento, 2011. URL <http://meccablog.ing.unitn.it/>.
- [17] Care database website, October 2012. URL http://ec.europa.eu/transport/road_safety/specialist/statistics/.
- [18] Ford collision warning with brake support, 2012. URL http://corporate.ford.com/doc/Adaptive_Cruise.pdf.
- [19] eurofot european project website, 2012. URL bib:site_eurofot.
- [20] Sartre european project website, 2012. URL <http://www.sartre-project.eu/en/Sidor/default.aspx>.
- [21] Adtf website, 2013. URL <http://automotive.elektrobit.com/?/home/driver-assistance-software/eb-assist-adtf.html>.
- [22] Nissan predictive forward collision warning, 2013. URL <http://www.nissan-global.com/EN/TECHNOLOGY/OVERVIEW/predictive.html>.
- [23] Volvo cars website for adas equipment, 2013. URL http://www.volvocars.com/uk/top/my_volvo/guides/pages/default.aspx.

- [24] dspace micro autobox and embedded pc, 2013. URL http://www.dspace.com/en/inc/home/products/hw/micautob/microautobox_embedded_pc.cfm.
- [25] The hori-fujimoto laboratory website, 2013. URL <http://hflab.k.u-tokyo.ac.jp/>.
- [26] interactive european project website, 2013. URL <http://www.interactive-ip.eu/>.
- [27] D.A. Abbink, E.R. Boer, and M. Mulder. Motivation for continuous haptic gas pedal feedback to support car following. In *Proceedings of the 2008 IEEE Intelligent Vehicles Symposium, Eindhoven (The Netherlands)*, 2008.
- [28] M. Abe. *Vehicle Handling Dynamics - Theory and Application*. Butterworth-Heinemann, 2009.
- [29] L. A. Adams. Review of the literature on obstacle avoidance manoeuvres: Braking versus steering. Technical report, The University of Michigan, 1994.
- [30] A. Alleyne. A comparison of alternative obstacle avoidance strategies for vehicle control. *Vehicle System Dynamics: International Journal of Vehicle Mechanics and Mobility*, 1997.
- [31] D. Althoff, M. Buss, A. Lawitzky, M. Werling, and D. Wollherr. On-line trajectory generation for safe and optimal vehicle motion planning. *Autonomous Mobile Systems*, pages 99–107, 2012.
- [32] D. Althoff, M. Werling, N. Kaempchen, D. Wollherr, and M. Buss. Lane-based safety assessment of road scenes using inevitable collision states. In *2012 Intelligent Vehicles Symposium Alcalá de Henares, Spain, June 3-7, 2012*, pages 31–36. IEEE, 2012.
- [33] A. Amditis, E. Bertolazzi, M. Bimpas, F. Biral, P. Bosetti, M. Da Lio, L. Danielsson, A. Gallione, H. Lind, A. Saroldi, and A. Sjögren. A holistic approach to the integration of safety applications: the insafes subproject within the european framework programme 6 integrating project prevent. *IEEE Transactions on Intelligent Transportation Systems*, 2009.

- [34] Y. Aoki, T. Uchida, and Y. Hori. Experimental demonstration of body slip angle control based on novel linear observer for electric vehicle. In *Proceedings of the 32nd Annual Conference of the IEEE Industrial Electronics Society, Paris, France, 2006*.
- [35] E. Bertolazzi, F. Biral, and M. Da Lio. Real-time motion planning for multibody systems - real life application examples. *Multibody Systems Dynamics*, 17(2-3):119–139, April 2007.
- [36] E. Bertolazzi, F. Biral, M. Da Lio, A. Saroldi, and F. Tango. Supporting drivers in keeping safe speed and safe distance: the saspenca subproject within the european framework programme 6 integrating project prevent. *IEEE Transactions on Intelligent Transportation Systems*, 2009.
- [37] J. T. Betts. *Practical Methods for Optimal Control and Estimation Using Nonlinear Programming*. Society of Industrial and Applied Mathematics, 2009.
- [38] F. Biral, R. Antonello, E. Bertolazzi, and F. Zendri. Integration of optimal maneuver prediction in active safety control systems: considerations on driving safety improvements. In *Proceedings of the 2010 IEEE Conference on Control Application, Yokohama, Japan, 2010*.
- [39] P. Bolzern, R. Scattolini, and N. Schiavoni. *Fondamenti di controlli automatici*. McGraw-Hill, 2008.
- [40] P. Bosetti, M. Da Lio, and A. Saroldi. On the human control of vehicles: an experimental study of acceleration. *European Transport Research Review*, 2013.
- [41] M. Brännström, J. Sjöberg, L. Hengelsson, and M. Christiansson. A real-time implementation of an intersection collision avoidance system. In International Federation of Automatic Control, editor, *Proceedings of the 18th IFAC World Congress Milano (Italy)*, 2011.
- [42] M. Brännström, E. Coelingh, and J. Sjöberg. Model-based threat assessment for avoiding arbitrary vehicle collisions. *IEEE Transactions on Intelligent Transportation Systems*, 11(3):658–669, September 2012.

- [43] A. Broadhurst, S. Baker, and T. Kanade. Monte carlo road safety reasoning. In IEEE, editor, *Proceedings of the 2005 IEEE Intelligent Vehicle Symposium, Las Vegas NV (USA)*, pages 319–324, 2005.
- [44] R. Brooks. A robust layered control system for a mobile robot. *IEEE Journal of Robotics and Automation*, 2(1):14–23, 1986.
- [45] R. Brooks. Intelligence without representation. *Artificial Intelligence*, 47 (1811):139–159, 1991.
- [46] A. E. Bryson. *Applied Linear Optimal Control - Examples and Algorithms*. Cambridge University Press, 2002.
- [47] A. E. Bryson and Y. Ho. *Applied Optimal Control - Optimization, Estimation and Control*. Taylor and Francis, 1975.
- [48] L. Cattaneo and G. Rizzolatti. The mirror neuron system. *Archives of Neurology*, 66(5):557–560, 2009.
- [49] C. Chaloupka and R. Risser. Don't wait for accidents - possibilities to assess risk in traffic by applying the wiener fahrprobe. *Safety Science*, 19, 1995.
- [50] E. Coelingh, L. Jakobsson, H. Lind, and M. Lindman. Collision warning with auto brake - a real-life safety perspective. Technical report, Volvo Car Corporation, 2007.
- [51] V. Cossalter, M. Da Lio, R. Lot, and L. Fabbri. A general method for the evaluation of vehicle manoeuvrability with special emphasis on motorcycles. *Vehicle System Dynamics: International Journal of Vehicle Mechanics and Mobility*, 31(2):113–135, 1999.
- [52] M. Da Lio, F. Biral, and E. Bertolazzi. Combining safety margins and user preferences into a driving criterion for optimal control-based computation of reference maneuvers for an adas of the next generation. In *Proceedings of the 2005 IEEE Intelligent Vehicle Symposium, Las Vegas NV (USA)*, 2005.
- [53] M. Da Lio, F. Biral, M. Galvani, and A. Saroldi. Will intelligent vehicles evolve into human-peer robots? In *Proceedings of the 2012 IEEE Intelligent Vehicle Symposium, Alcalà de Henares (Spain)*, 2012.

- [54] M. Da Lio, F. Biral, E. Bertolazzi, M. Galvani, P. Bosetti, A. Saroldi, and F. Tango. The driver continuous support function in the fp7 “interactive” project: an implementation based on the “co-driver” metaphor. In *Proceedings of the Transport Research Arena 2014, Paris, France*, 2014.
- [55] M. Da Lio, F. Biral, E. Bertolazzi, M. Galvani, P. Bosetti, D. Windridge, A. Saroldi, and F. Tango. Artificial co-drivers as a universal enabling technology for future intelligent vehicles and transportation systems. *IEEE Transactions on Intelligent Transportation Systems*, 2014.
- [56] Y. Demiris and B. Khadhour. Hierarchical attentive multiple models for the execution and recognition of actions. *Robotics and Autonomous Systems*, 54(5):361–369, 2006.
- [57] M. Distner, M. Bengtsson, T. Broberg, and L. Jakobsson. City safety - a system addressing rear-end collisions at low speeds. Technical Report 09-0371, Volvo Cars Company, 2009.
- [58] A. Doshi and M. Trivedi. Tactical driver prediction and intent reference: A review. In *Proceedings of the 2011 IEEE Intelligent Transportation Systems Conference, Washington DC (USA)*, 2011.
- [59] A. Eidehall and L. Petersson. Statistical threat assessment for general road scenes using monte carlo sampling. *IEEE Transactions on Intelligent Transportation Systems*, 9(1):137–147, 2008.
- [60] S. Ellefsen. *Analysis of Drifting for a Remotely Controlled Car*. PhD thesis, Norwegian University of Science and Technology, Trondheim, 2012.
- [61] F. Fahrenkrog et al. Impact assessment of developed applications – overall interactive assessment. Technical report, European Commission FP7, 2013.
- [62] B. Fajen and W. Warren. Behavioral dynamics of steering, obstacle avoidance and route selection. *Journal of Experimental Psychology*, 29(2):343–362, 2003.
- [63] P. Falcone, F. Borrelli, J. Asgari, H.E. Tseng, and D. Hrovat. Predictive active steering control for autonomous vehicle systems. *Control Systems*

- Technology, IEEE Transactions on*, 15(3):566–580, 2007. ISSN 1063-6536. doi: 10.1109/TCST.2007.894653.
- [64] P. Falcone, M. Tufo, F. Borrelli, J. Asgari, and H.E. Tsengz. A linear time varying model predictive control approach to the integrated vehicle dynamics control problem in autonomous systems. In *Decision and Control, 2007 46th IEEE Conference on*, pages 2980–2985, 2007. doi: 10.1109/CDC.2007.4434137.
- [65] P. Falcone, M. Ali, and J. Sjöberg. Predictive threat assessment via reachability analysis and set invariance theory. *IEEE Transactions on Intelligent Transportation Systems*, 12(4):1352–1361, 2011.
- [66] M. Fausten. Accident avoidance by evasive manoeuvres, 2010.
- [67] M. Felsberg, A. Shaukat, and D. Windridge. Online learning in perception-action systems. In *Proceedings of the 11th European Conference on Computer Vision*, 2010.
- [68] D. Ferguson, M. Darms, C. Urmson, and S. Kolski. Detection, prediction, and avoidance of dynamic obstacles in urban environments. In *Proceedings of the 2008 IEEE Intelligent Vehicles Symposium, Eindhoven (The Netherlands)*, 2008.
- [69] D. Ferguson, T. M. Howard, and M. Likhacev. Motion planning in urban environments. In M. Buehler, K. Iagnemma, and S. Singh, editors, *The DARPA Urban Challenge*, volume 56 of *Springer Tracts in Advanced Robotics*, pages 61–89. Springer, 2009.
- [70] F. Flemisch, C. A. Adams, S. R. Conway, K. H. Goodrich, M. T. Palmer, and P. C. Schutte. The h-metaphore as a guideline for vehicle automation and interaction. *Control*, pages 1–30, December 2003.
- [71] F. Flemisch, J. Kelsch, C. Löper, A. Schieben, J. Schindler, and M. Heesen. Cooperative control and active interfaces for vehicle assistance and automation. In *Proceedings of the 2008 FISITA World Automotive Congress, Munich (Germany)*, 2008.

- [72] E. Frazzoli, M. A. Dahleh, and E. Feron. Real-time motion planning for agile autonomous vehicles. *Journal of Guidance, Control and Dynamics*, 25(1), January-February 2002.
- [73] H. Fujimoto. Regenerative brake and slip angle control of electric vehicle with in-wheel motor and active front steering. In *Proceedings of the 2011 JSAE Annual Congress*, 2011.
- [74] H. Fujimoto and H Sumiya. Range extension control system of electric vehicle based on optimal torque distribution and cornering resistance minimization. In *Proceedings of the 37th Annual Conference of the IEEE Industrial Electronics Society, Melbourne, Australia*, pages 3727–3722, 2011.
- [75] H. Fujimoto and Y. Yamauchi. Advanced motion control of electric vehicle based on lateral force observer with active steering. In *Proceedings of the 2010 IEEE International Symposium on Industrial Electronics (ISIE), Bari, Italy*, pages 3627–3632, 2010.
- [76] H. Fujimoto, J. Amada, and K. Maeda. Review of traction and braking control for electric vehicle. In *Proceedings of the 2012 IEEE Vehicle Power and Propulsion Conference, Seoul (South Korea)*, 2012.
- [77] R. Grush. The emulation theory of representation: motor control, imagery and perception. *The behavioral and brain sciences*, 27(3):377–396, 2004.
- [78] P. E. Hart, N. J. Nilsson, and B. Raphael. A formal basis for the heuristic determination of minimum cost paths. *IEEE Transactions on Systems Science and Cybernetics*, pages 100–107, 1968.
- [79] G. Hesslow. Conscious thought as a simulation of behaviour and perception. *Trends in Cognitive Sciences*, 6(6):242–247, June 2002.
- [80] G. Hesslow. The current status of the simulation theory of cognition. *Brain Research*, 1428:71–79, 2012.
- [81] T. Hiraoka, T. Kunimatsu, O. Nishihara, and H. Kumamoto. Modeling of driver following behavior based on minimum-jerk theory. In *Proceedings of the 12th ITS World Congress, San Francisco CA (USA)*, 2005.

- [82] G. M. Hoffmann, C. J. Tomlin, M. Montemerlo, and S. Thrun. Autonomous automobile trajectory tracking for off-road driving: Controller design, experimental validation and racing. In *Proceedings of the 2007 American Control Conference, New York City, USA, 2007*.
- [83] E. Hollnagel. Cognition as control: a pragmatic approach to the modelling of joint cognitive systems. *Control*, 9:1–23, 2002.
- [84] E. Hollnagel and D. D. Woods. Cognitive systems engineering: new wine in new bottles. *International Journal of human and computer studies*, 51(2):339–356, 1999.
- [85] B. Hommel, J. Müsseler, G. Aschersleben, and W. Prinz. The theory of event coding (tec): a framework for perception and action planning. *Behavioral and Brain Sciences*, 24(5):849–878, 2001.
- [86] R. A. Howard. Dynamics programming and markov processes. *MIT Press and Wiley*, 1960.
- [87] D. Hrovat, S. Di Cairano, H.E. Tseng, and I.V. Kolmanovsky. The development of model predictive control in automotive industry: A survey. In *Proceedings of the 2012 IEEE International Conference on Control Applications, Dubrovnik (Croatia), 2012*.
- [88] D. G. Hull. *Optimal Control Theory for Applications*. Mechanical Engineering Series. Springer, 2003.
- [89] S. Hurley. The shared circuits model (scm): how control, mirroring and simulation can enable imitation, deliberation and mindreading. *The behavioral and brain sciences*, 31(1):1–22, 2008.
- [90] T. Inagaki. Smart collaboration between humans and machines based on mutual understanding. *Annual Reviews in Control*, 32(2):253–261, 2008.
- [91] A. Kelly. *An intelligent Predictive Control Approach to the High Speed Cross Country Autonomous Navigation Problem*. PhD thesis, Carnegie Mellon University, 1995.

- [92] A. Kelly and A. Stentz. An approach to rough terrain autonomous mobility. In *Proceedings of the 1997 International Conference on Mobile Planetary Robots*, 1997.
- [93] K. D. Kusano and H. C. Gabler. Potential effectiveness of integrated forward collision warning, pre-collision brake assist, and automated pre-collision braking systems in real-world rear-end collisions. Technical report.
- [94] S. M. La Valle and J. J. Kuffner. Randomized kinodynamic planning. *International Journal of Robotics Research*, pages 378–400, 2001.
- [95] A. Y. Lee. Performance of four-wheel-steering vehicles in lane change maneuvers. Technical report, Jet Propulsion Laboratory, 1995.
- [96] J. Leonard et al. A perception-driven autonomous urban vehicle. In M. Buehler, K. Iagnemma, and S. Singh, editors, *The DARPA Urban Challenge*, volume 56 of *Springer Tracts in Advanced Robotics*, pages 162–230. Springer, 2009.
- [97] L. Li, D. Wen, N.N. Zheng, and L.C. Shen. Cognitive cars: a new frontier for adas research. *IEEE Transactions on Intelligent Transportation Systems*, 13(1):395–407, March 2012.
- [98] M. Lidberg, A. Nozad, M. Hassanzadeh, and J. Shah. Vehicle dynamics model and path stability control algorithms. Technical report, European Commission FP7, 2013.
- [99] M. Liebner, M. Baumann, F. Klanner, and C. Stiller. Driver intent inference at urban intersections using the intelligent driver model. In *Proceedings of the 2012 IEEE Intelligent Vehicle Symposium, Alcalà de Henares (Spain)*, 2012.
- [100] A. Lindgren and F. Chen. State of the art analysis: An overview of advanced driver assistance systems (adas) and possible human factors issues. In *Proceedings of the Swedish Human Factors Network (HFN) Conference Linköping (Sweden)*, 2006.

- [101] C. C. MacAdam. Understanding and modeling the human driver. *Vehicle System Dynamics: International Journal of Vehicle Mechanics and Mobility*, 40(1-3):101–134, January 2003.
- [102] L. Malta, C. Miyajima, and K. Takeda. A study of driver behaviour under potential threats in vehicle traffic. *IEEE Transactions on Intelligent Transportation Systems*, 10(2):201–210, 2009.
- [103] J. Mattingley, Y. Wang, and S. Boyd. Receding horizon control - automatic generation of high-speed solvers, June 2011.
- [104] D.Q. Mayne, J.B. Rawlings, C.V. Rao, and P.O.M. Scokaert. Constrained model predictive control: Stability and optimality. *Automatica*, 36(6):789 – 814, 2000. ISSN 0005-1098. doi: [http://dx.doi.org/10.1016/S0005-1098\(99\)00214-9](http://dx.doi.org/10.1016/S0005-1098(99)00214-9). URL <http://www.sciencedirect.com/science/article/pii/S0005109899002149>.
- [105] Joel McCall and Mohan M. Trivedi. Driver monitoring for a human-centered driver assistance system. In *Proceedings of the 1st ACM International Workshop on Human-centered Multimedia*, HCM '06, pages 115–122, New York, NY, USA, 2006. ACM. ISBN 1-59593-500-2. doi: 10.1145/1178745.1178764. URL <http://doi.acm.org/10.1145/1178745.1178764>.
- [106] D. V. McGehee, N. E. Mazzae, and G. H. S. Baldwin. Examination of drivers' collision avoidance behavior using conventional and antilock brake systems on the iowa driving simulator. Technical report, NHTSA, 1999.
- [107] A. N. Meltzoff. The 'like me' framework for recognizing and becoming an intentional agent. *Acta Psychologica*, 124(1):26–43, 2007.
- [108] M. Montemerlo et al. Junior: the stanford entry in the urban challenge. In M. Buehler, K. Iagnemma, and S. Singh, editors, *The DARPA Urban Challenge*, volume 56 of *Springer Tracts in Advanced Robotics*, pages 90–123. Springer, 2009.

- [109] M. Nagai. The perspectives of research for enhancing active safety based on advanced control technology. *Vehicle System Dynamics: International Journal of Vehicle Mechanics and Mobility*, 45(5):413–431, May 2007.
- [110] W.G. Najm et al. Analysis of light vehicle crashes and pre-crash scenarios based on the 2000 general estimates system. Technical report, National Highway Traffic Safety Administration, 2003.
- [111] K. Nam, S. Oh, H. Fujimoto, and Y. Hori. Vehicle state estimation for advanced vehicle motion control using novel lateral force sensors. In *Proceedings of the 2011 American Control Conference*, pages 4853–4858, 2011.
- [112] K. Nam, H. Fujimoto, and Y. Hori. Lateral stability control of in-wheel-motor-driven electric vehicles based on sideslip angle estimation using lateral tire force sensors. *IEEE Transactions on Vehicular Technology*, 61(5):1972–1985, June 2012.
- [113] K. Nam, S. Oh, H. Fujimoto, and Y. Hori. Estimation of sideslip and roll angles of electric vehicles using lateral tire force sensors through rls and kalman filter approaches. *IEEE Transactions on Industrial Electronics*, 60(3):988–1000, 2012.
- [114] V. L. Neale et al. An overview of the 100-car naturalistic study and findings. *Traffic Safety*, 2005.
- [115] J. Nocedal and S. J. Wright. *Numerical Optimization*. Springer Series in Operation Research and Financial Engineering. Springer, second edition, 2009.
- [116] K. Ono, T. Takizawa, and M. Aoki. *Preload measuring device for double row rolling bearing unit*. U.S. Patent 20090052825A1, Febraury 2009.
- [117] H. B. Pacejka. *Tyre and Vehicle Dynamics*. Butterworth-Heinemann, 2002.
- [118] S. Park, J. Deyst, and J. P. How. Performance and lyapunov stability of a non-linear path following guidance method. *Journal of Guidance, Control and Dynamics*, 6:1718–1728, 2007.

- [119] Z. M. Pylyshyn. Computation and cognition: Towards a foundation for cognitive sciences. *MIT Press*, 88(1), 1984.
- [120] C. Reinholtz et al. Odin: Team victortango's entry in the darpa urban challenge. In M. Buehler, K. Iagnemma, and S. Singh, editors, *The DARPA Urban Challenge*, volume 56 of *Springer Tracts in Advanced Robotics*, pages 122–162. Springer, 2009.
- [121] F. Sandblom and M. Brännström. Probabilistic threat assessment and driver modeling in collision avoidance systems. In *Proceedings of the 2011 IEEE Intelligent Vehicles Symposium (IV), Baden Baden (Germany)*, June 5-9, 2011.
- [122] A. Saroldi, F. Tango, M. Da Lio, F. Biral, and M. Galvani. Implementation of a co-driver for continuous support. In *Proceedings of the 19th ITS World Congress, Vienna (Austria)*, 2012.
- [123] D. Scaramuzza, L. Spinello, R. Triebel, and R. Siegwart. Key technologies for intelligent and safer cars - from motion estimation to predictive collision avoidance. In *Proceedings of the 2009 IEEE International Conference on Intelligent Robots and Systems*, pages 2803–2808, 2009.
- [124] J. Schmuedderich et al. System approach for multi-purpose representations of traffic scene elements. In *Proceedings of the 2010 13th IEEE Intelligent Transportation Systems Conference, Madeira Island (Portugal)*, 2010.
- [125] M. Shino, T. Abiru, M. Kamata, M. Nagai, Y. Michitsuji, and K. Moro. Research on incident analysis on driving behaviour based in near-miss incident data while crossing non-signalized intersection. In *Proceedings of FISTA 2006*, 2006.
- [126] M. Shino, M. Kamata, M. Nagai, Y. Michitsuji, and K. Moro. Research on incident analysis using drive recorder part 3: Analysis on relationship driving behaviour and traffic circumstance based on forward collision near-miss incident data in car following situation. In *Proceedings of FISTA 2006*, 2006.

- [127] J. Sjöberg, E. Coelingh, M. Ali, M. Brännström, and P. Falcone. Driver models to increase the potential of autonomous safety functions. In *Proceedings of the 18th European Signal Processing Conference, Aalborg (Denmark)*, 2010.
- [128] A. Stentz and M. Hebert. A complete navigation system for goal acquisition in unknown environments. *Autonomous Robots*, pages 127–145, 1995.
- [129] H. Summala. Driver/vehicle steering response latencies. *Human Factors*, 23(6):683–692, 1981.
- [130] S. Thrun, D. Fox, W. Burgard, and F. Dellaert. Robust monte-carlo localisation for mobile robots. *Artificial Intelligence*, 2001.
- [131] E. Todorov. Optimality principles in sensorimotor control. *Nature neuroscience*, 7(9):907–915, September 2004.
- [132] E. Todorov and M. I. Jordan. A minimal intervention principle for coordinated movement. *Advances in Neural Information Processing Systems*, 15: 27–34, 2003.
- [133] J.L. Troutman. *Variational Calculus and Optimal Control - Optimization with Elementary Convexity*. Undergraduate texts in mathematics. Springer, second edition, 1996.
- [134] C. Urmson et al. Autonomous driving in urban environments: Boss and the urban challenge. In M. Buehler, K. Iagnemma, and S. Singh, editors, *The DARPA Urban Challenge*, volume 56 of *Springer Tracts in Advanced Robotics*, pages 1–60. Springer, 2009.
- [135] P. Viviani and T. Flash. Minimum-jerk, two-thirds power law, and isochrony: converging approaches to movement planning. *Journal of Experimental Psychology*, 21(1):32–53, March 1995.
- [136] AA. VV. Mercedes-benz collision avoidance features: initial results. Technical report, Highway Loss Data Institute, 2012.
- [137] D. Wen, G. Yan, N.N. Zheng, L.C. Shen, and L. Li. Towards cognitive vehicles, 2011.

- [138] D. Windridge and J. Kittler. Perception-action learning as an epistemologically-consistent model for self-updating cognitive representation. *Brain Inspired Cognitive Systems*, 2008.
- [139] D. Windridge, M. Felsberg, and A. Shaukat. A framework for hierarchical perception-action learning utilizing fuzzy reasoning. *IEEE Transactions on systems, man and cybernetics*, 43(1):155–169, 2012.
- [140] D. M. Wolpert and M. Kawato. Multiple paired forward and inverse models for motor control. *Neural Networks*, 11(7-8):1317–1329, 1998.
- [141] Chen Y., Sundareswaran V., C. Anderson, A. Broggi, P. Grisleri, P. Porta, P Zani, and J. Beck. Terramax: Team oshkosh urban robot. In M. Buehler, K. Iagnemma, and S. Singh, editors, *The DARPA Urban Challenge*, volume 56 of *Springer Tracts in Advanced Robotics*, pages 594–622. Springer, 2009.
- [142] Y. Yamauchi and H. Fujimoto. Proposal of lateral force observer with active steering for electric vehicle. In *Proceedings of the 2008 SICE Annual Conference, Japan*, 2008.
- [143] F. Zendri. *Development of a Research Vehicle able to Perform Autonomous Manoeuvres*. PhD thesis, University of Trento - Graduate School in Structural Engineering - Modeling, preservation and Controls of Materials and Structures, 2010.
- [144] F. Zendri. Driving simulator. Technical report, University of Trento, Dept. of Mechanical and Structural Engineering, May 2012.