2016

Control Systems Design for Automatic Drive of a Passenger Car in Critical Scenarios

Jerome Blanc
University of Windsor

Control Systems Design for Automatic Drive of a Passenger Car in Critical Scenarios

by

Jerome Blanc

A Thesis
Submitted to the Faculty of Graduate Studies
through the Department of Mechanical, Automotive, and Materials Engineering
in Partial Fulfillment of the Requirements for
the Degree of Master of Applied Science
at the University of Windsor

Windsor, Ontario, Canada
2016

Control Systems Design for Automatic Drive of a Passenger Car in Critical Scenarios

by

Jerome Blanc

APPROVED BY:

Dr. N. Kar
Department of Electrical and Computer Engineering

Dr. B. Minaker
Department of Mechanical, Automotive, and Materials Engineering

Dr. J. Johrendt, Advisor
Department of Mechanical, Automotive, and Materials Engineering

29 August 2016

Declaration of Originality

I hereby certify that I am the sole author of this thesis and that no part of this thesis has been published or submitted for publication.

I certify that, to the best of my knowledge, my thesis does not infringe upon anyones copyright nor violate any proprietary rights and that any ideas, techniques, quotations, or any other material from the work of other people included in my thesis, published or otherwise, are fully acknowledged in accordance with the standard referencing practices. Furthermore, to the extent that I have included copyrighted material that surpasses the bounds of fair dealing within the meaning of the Canada Copyright Act, I certify that I have obtained a written permission from the copyright owner(s) to include such material(s) in my thesis and have included copies of such copyright clearances to my appendix.

I declare that this is a true copy of my thesis, including any final revisions, as approved by my thesis committee and the Graduate Studies office, and that this thesis has not been submitted for a higher degree to any other University or Institution.

Abstract

Advanced Driver Assistance Systems (ADAS) aim at supporting the driver's task in order to improve vehicular safety. One of the most promising and most studied technologies in this direction is Autonomous Driving (AD).

While control systems for AD based on lane markings have been proposed in the literature, few have addressed the problem of coping with the absence of lane references on the ground. Moreover, many of these solutions resort to complex software and/or hardware.

In this project a relatively straightforward way of restoring suitable knowledge of the position of the vehicle when the output of the Lane Recognition Camera (LRC) is not available or degraded is presented. This is done exploiting a relatively new approach to variable recovering which results in a so-called "virtual sensor".

In order to show the potential of this solution, then, a control system based on a LRC is first developed in the Simulink® environment. Subsequently, the virtual sensor for precise vehicle position reconstruction is implemented and evaluated against the first solution. Simulations considering realistic driving conditions showed comparable levels of performance between the two systems, demonstrating the effectiveness of this new approach.

To my family, for their relentless support.

Acknowledgements

This thesis is the final outcome of a two year Double Degree Master program made possible only thanks to the collaboration and organizational efforts of two universities, University of Windsor and Politecnico di Torino, and a prestigious industrial partner such as Fiat Chrysler Automobiles.

Therefore I would like to express my sincere appreciation to the persons who represent the aforementioned institutions and, in particular, to Mohammed Malik from FCA Canada.

I would like to extend my gratitude to my academic advisor at the University of Windsor, Dr. Jennifer Johrendt, who has supported me throughout the project and to my industrial advisor from Centro Ricerche Fiat, Dr. Pandeli Borodani, for his help, encouragement and suggestions.

That said, my deepest recognition goes to my parents, for the education they gave me and for their continuous moral and economic support throughout my studies. A huge "thank you" goes to my brother, for being my biggest fan, just as much as I am his.

Last but not least, special recognition goes to my colleagues, house-mates and friends, Marco "Div" Di Vittorio, Davide "Bore" Borello, Mirko "una brava persona" Pesce, Marco "the tall one" Gerini e Davide "duro ma onesto" Pezzetti, for making this experience unique and unforgettable.

Table of Contents

List of Tables

List of Figures

List of Abbreviations

ABS	Anti-lock Braking System
ACC	Adaptive Cruise Control
AD	Autonomous Driving
ADAS	Advanced Driver Assistance Systems
ALKA	Active Lane Keeping Assist
ANN	Artificial Neural Network
ARX	AutoRegressive model with eXogenous inputs
AV	Autonomous Vehicle
AVCS	Advanced Vehicle Control Systems
BASt	German Federal Highway Research Institute
CAN	Controller Area Network
C(o)G	Center (of) Gravity
CICC	Cooperative Intelligent Cruise Control
CNN	Convolutional Neural Networks
CPU	Central Processing Unit
CRF	Centro Ricerche Fiat S.C.p.A.
DAC	Downhill Assist Control System
DGPS	Differential Global Positioning System
DOF	Degree Of Freedom
DRIVE	Dedicated Road Infrastructure for Vehicle Safety in Europe
DSRC	Dedicated Short-Range Communications
EBA	Electronic Brake Assist

EBD	Electronic Brake Force Distribution
ECU	Electronic Control Unit
$EP(A)S$	Electric Power (Assisted) Steering
ESP/C	Electronic Stability Program/Control
FCA	Fiat Chrysler Automobiles
FCM	Forward Crash Mitigation
FCW	Forward Collision Warning
FEA	Finite Element Analysis
GPS	Global Positioning System
HAC	Hill-Start Assist Control)
$HDOP$	Horizontal Dilution Of Precision
HIL	Hardware-In-the-Loop
HT	Hough Transform
ICC	Intelligent Cruise Control
IMU	Inertial Measurement Unit
IPM	Inverse Perspective Mapping
JDL	Joint Directors of Laboratories
KF	Kalman Filter
LCA	Lane Change Assistance
LD	Lane Detection
LDW	Lane Departure Warning
$LKAS$	Lane Keeping Assist System
$LIDAR$	LIght Detection and Ranging
$LOIS$	Likelihood Of Image Shape
LQG	Linear Quadratic Gaussian control
LQI	Linear Quadratic Integral controller
LQR	Linear Quadratic Regulator
LRC	Lane Recognition Camera
LTI	Linear Time Invariant

LTR	Loop Transfer Recovery
MEX	MATLAB EXecutable
$MISO$	Multiple Inputs Single Output
MSE	Mean Square Error
MSW	Mean Square Weight
NaN	Not a Number
$NARX(nlARX)$	Non-linear ARX
$NHTSA$	National Highway Traffic Safety Administration
NN	Neural Network
$ODE4$	Fourth-order Ordinary Differential Equations
OEM	Original Equipment Manufacturer
PCS	PreCrash System
$PI(D)$	Proportional-Integral-(Derivative) controller
PWM	Pulse Width Modulation
$RADAR$	RAdio Detection And Ranging
$RANSAC$	RANdom SAmple Consensus
ROI	Region Of Interest
$RTK-DGPS$	Real-Time Kinematic Differential GPS
SAE	Society of Automotive Engineers
$SISO$	Single Input Single Output
SMC	Sliding Mode Control
SVM	Support Vector Machine
TCS	Electronic Traction Control System
$V2I$	Vehicle-to-Infrastructure
$V2V$	Vehicle-to-Vehicle
$V2X$	Vehicle-to-X
VDA	German Association of the Automotive Industry
VDO	Vereinigte DEUTA-OTA - Continental AG
VS	Virtual Sensor

$VSAT$	Very Small Aperture Terminal
VSC	Vehicle Stability Control

List of Symbols

α	Tire Sideslip Angle
β	Vehicle Sideslip Angle
γ	Regularization Parameter
Γ_w	Wheel Radius
δ_{steer}	Steering Angle
$\dot{\theta}$	Pitch Velocity
Θ	Road Slope Angle
λ	Wheel Slip
μ	Adhesion Coefficient
$\dot{\phi}$	Roll Velocity
$\dot{\psi}$	Yaw Rate
ω_w	Wheel Angular Velocity
$a = f(wp + b)$	Node Output
$\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$, $\mathbf{D}$	LTI System Matrices
a_n, b_n	ARX Coefficients
a_x	Vehicle Longitudinal Acceleration
b	Node Bias
C_α (C_f, C_r)	Tire Cornering Stiffness (front, rear)
d_t	Disturbance
E_{M_sum}	External “Force” Acting on Road Edges
$E_L(s) + E_R(s)$	Left and Right External “Forces” Acting on Road Edges
f	Node Transfer Function

F_0, F_A	Virtual Sensor Functionals
$F_a\ (D_a)$	Aerodynamic Drag Force
F_f	Friction Force
$F_t, F_x\ (F_{xf}, F_{xr})$	Tractive Force (front, rear)
$F_y\ (F_{yf}, F_{yr})$	Tire Lateral Force (front, rear)
F_z	Vertical Load
g	Gravitational Acceleration
$H = K_{obs}$	Optimal Kalman Gain Matrix
H_∞	H-infinity Mathematical Optimization
$J(u, x)$	LQI Cost Functional
J_w	Wheel Moment of Inertia
J_z	Vehicle Yaw Moment of Inertia
k	Wheel/Steering Angle Ratio
$K\ [deg/g]$	Understeer Gradient
$K\ [m/deg]$	deg to m Conversion Factor
$\mathbf{K}$	State Feedback Gain Matrix
K_d	Coefficient Differential Path PID
K_i	Coefficient Integral Path PID
K_l	Road Curvature
K_p	Coefficient Proportional Path PID
K_u, T_u	Ziegler-Nichols' Tuning Parameters
L	Look-Ahead Distance
$\mathbf{L}$	Luenberger's Observer Matrix
$L = b + c$	Vehicle Wheelbase (sum of axle-CG distances)
l_f, b	Distance CG-Front Axle
l_r, c	Distance CG-Rear Axle
lat	Latitude
$long$	Longitude
m, M	Vehicle Mass

M_r	Equivalent Mass of Rotating Components
$N(t)$	Numerical Data
n_a, n_b, n_k	Orders of ARX Model
N_h	Number of Nodes in the ANN
N_i	Number of Input Neurons
N_o	Number of Output Neurons
N_s	Number of Samples in Training Data
p	Node Input
$\mathbf{Q}, \mathbf{R}, \mathbf{N}$	LQI Tuning Matrices
$R\ [m]$	Radius of Curvature
$R\ [\%]$	Regression Coefficient
R_{hx}	Towing Force
R_x	Rolling Resistance
t	Time
T	Training Target
$T(t)$	Target Data
T_b	Braking Torque
T_{dem}	Torque Applied to Steering Column
T_e	Driving Torque
T_{dr}	Input Torque to EPS Actuator
$W\ (W_f, W_r)$	Vehicle Weight (front, rear)
$x, x_t \in X \subset \Re^n$	Dynamic System State
$X - Y$	Actual Vehicle Trajectory
$X_{des} - Y_{des}$	Desired Vehicle Trajectory
$u_t \in U \subset \Re^{mu}$	Dynamic System Input
V	Validation Metric
$v_x, V, Vel, \dot{x}$	Vehicle Longitudinal Velocity
$v_y, \dot{y}$	Vehicle Lateral Velocity
w	Node Input Weight

Y	Training Output
$y_{(fb)} = q + mL$	Linear Approximation of Lane Centerline
$y_t \in Y \subset \Re^{my}$	Dynamic System Output
$\dot{z}$	Heave Velocity
$Z = N + P$	Nyquist Criterion Parameters
$z_t \in Z \subset \Re^{mz}$	Subset of the Dynamic System State

Chapter 1

Introduction

Vehicle safety is one of the most important factors which is influencing modern automotive development. Over the past decades, passive safety improvements [1], as well as advancements in road construction, government legislation and education, have contributed to decrease the occurrence and severity of vehicular accidents. The pursuit of the ultimate goal of " zero automotive fatalities " [2] fosters the development of *active* safety systems, with the task of assisting drivers in avoiding accidents before they even occur. *EU Commission* reports [3] how the trend of the reduction of deaths linked to vehicular accidents is slowing down in recent years (figure 1.1). In order to satisfy the target of halving the number of road deaths by 2020 a major breakthrough is then to be sought, and the automation of the dynamic driving task by means of suitable safety systems is the path selected by various companies in the automotive sector. These systems form the cluster of the so-called Advanced Driver Assistance Systems (ADAS). SAE Standard J3016 groups the impact of ADAS on the driving task in six levels of driving automation, from *no automation* to *full automation* [4].

Current ADAS applications available on the market do not yet entail high or full automation of steering and acceleration/deceleration actions, but are rather designed to support the driver with his/her driving task. Different authors have classified current and near-future

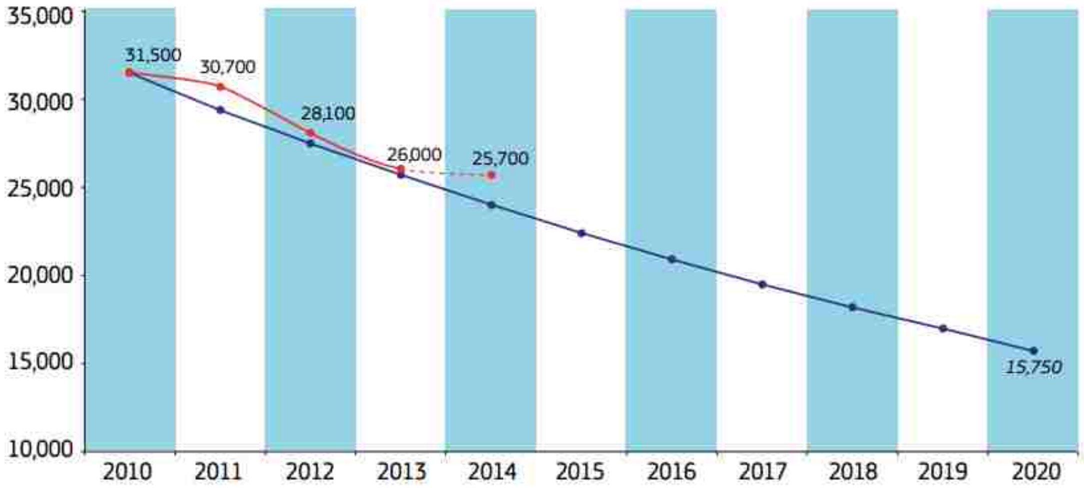


Figure 1.1: *Road deaths in the European Union [3]. Solid blue: proposed objective; solid red: actual data; dashed red: projected estimate*

driver assistance systems ([5], [6], [7] and references therein). Despite the fact that there are no fully automated production vehicles available nowadays, it is clear that the revolution of AD - autonomous driving - is on the brink of flooding automotive mass markets. This is not a revolution that will happen overnight, but rather it will come through a sequence of small steps. Several companies already strive in the direction of AD, to name a few: Google [8], Mercedes-Benz [9] [10], Delphi Automotive [11], Nissan [12], Audi [13] and Bosch [14]. The first automaker that will be able to offer an automated driving system within its production range will gain a crucial advantage in terms of appearance, reputation, technical know-how and economic profit.

The objective of this work is to insert itself into the realm of the current efforts toward AD by proposing an approach to deal with complex driving scenarios.

1.1 Problem statement

Control systems for automated driving based on the recognition of the lane from road markings have been proposed in the literature (see e.g. [15, 63, 64, 65, 66, 67, 68, 69, 70] for references on lane recognition and related applications). Current commercial assistance systems [16] [17] offer motorway aids that can help the driver maintain a predetermined speed and distance from the vehicle ahead (ACC, Adaptive Cruise Control) and even apply a torque on the steering wheel or on the brakes in order to keep the center of the lane (LKAS, Lane Keeping Assist System; ALKA, Active Lane Keeping Assist).

A particularly complex problem arises when such lane references are incomplete or totally missing. Until now, these kinds of situations have been addressed with highly complex experimental hardware and software (see e.g. [18]), often relying on sophisticated mathematical approaches for the detection of the physical road boundaries.

On real roads, references could be missing partially or entirely. They could be hidden by obstacles (like other vehicles) in the line of sight of the camera or their detection could be made difficult by harsh weather or illumination conditions. Some examples of challenging scenarios comprise:

- Toll gates, motorway connections and similar scenarios characterized by lines of vehicles;
- Country roads without road markings;
- Deflected or narrowing lanes due to maintenance work on the highway;
- Intersections governed by roundabouts which require specific rules of insertion and exit of the vehicles.

These scenarios constitute problems encountered globally and represent a step up in complexity compared to a pure highway driving context. A second cluster could also be defined considering the situation in which a vehicle is entering a tunnel or an underpass and the GPS signal is temporarily lost.

All these kinds of circumstances have in common the lack of specific references. This poses the demand for simple and economical control systems, yet powerful enough to deal with the absence of lanes and road markings. The ambition of this work is then to find a relatively straightforward way of restoring suitable knowledge of the position of the vehicle when the output of the Lane Recognition Camera (LRC) is not available or degraded.

1.2 Objectives

In this section a more thorough description of the proposed objective of this project is provided. As previously mentioned, the end goal is the design of a control system able to perform the so-called *precise position reconstruction* for the considered vehicle. Exploited information will include the following:

- Vehicle data from already installed sensors;
- Traditional GPS;
- Lane Detection Camera;
- Maps.

It is then faced the so-called *data-fusion* problem in which the objective is to exploit complimentary and redundant information coming from the sensors in order to enhance precision and robustness in the understanding of the driving situation. This is a widely employed technique that has been proved effective [19] and suitable for system performance enhancement at different levels [20]. Data fusion techniques can be successfully employed in multi-sensor environments in order to obtain lower error probability and higher reliability using data from multiple sources.

The system to be designed will have to be modular in its structure in such a way as to allow the possible addition - at a later time - of other ADAS modules or other, more sophisticated sensors and actuators contributing to the system performance.

Two main clusters of possible applications for the system to be developed have been identified. Firstly, as already mentioned, are those situations in which road signals are missing, and hence the camera is unable to suitably define the boundaries of the driving corridor: vicinity of roundabouts, country roads lacking suitable road signage, vehicle/queue management when approaching toll gates or motorway connections, changing lanes due to works on the highway, *etc.* Secondly, we have those scenarios in which a temporary absence of the GPS may be present. This condition is more complex and, as a consequence, in this work we will concentrate our attention mainly on the first group. Additionally, in order to guarantee a proper operation of the camera (as it will be discussed later), only speeds equal or greater than 60 km/h will be considered here.

When not arranged as the only self-driving feature implemented on the vehicle, the proposed system could be considered to realize a so-called *limp-home* strategy for an already existing, more sophisticated autonomous driving system. This means that a more "traditional" system, heavily relying on the use of a LRC and/or a RADAR/LIDAR apparatus (see, for instance, [19]) could be employed as the standard module for the self-driving functions of the vehicle. In addition to this arrangement, a system similar to what will be presented here could be added on-board the vehicle in such a way as to act as a sort of safety net, capable of maintaining active - although degraded - self-driving capabilities of the vehicle in the event that the road sensing apparatus of the main module is compromised. Alternatively, the system could be employed together with an already existing self-driving functionality in order to improve the reliability and performance of the overall setup.

Regardless of the methodology and design used, attention will be given to the *performances* of the system, in terms of precision and repeatability of the selected algorithm. In particular the system capability of avoiding departure of the vehicle from the road lane will be considered of the utmost importance. Among the most relevant objectives we need to include is the *robustness* of the estimation algorithm, i.e. its ability to tolerate perturbations that might affect functionality, and the *complexity* of the network used to achieve an acceptable performance. It will then be important to realize a system capable of satisfying performance requirements, without resorting - as much as possible - to expensive and overly complex solutions. A more quantitative description of the targets mentioned here will be provided when discussing the results of the validation and simulation of the system.

1.3 Methodology

Let us now anticipate briefly the methodology that will be followed in the development of this Thesis. The vehicle considered in this research is the "crossover" produced by Fiat, the 2015 *500X* (figure 1.2). The data that will be used are those now becoming commonly available on medium-range production vehicles, in accordance with the previously

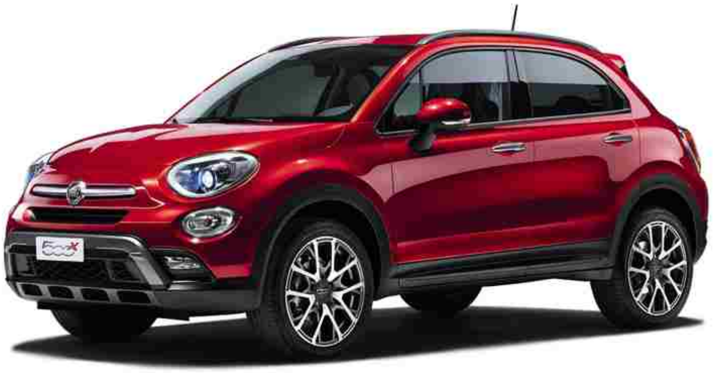


Figure 1.2: *Considered vehicle: 2015 Fiat 500x*

mentioned constraint of developing a system that doesn't require the inclusion of additional complex sensors or processing software. In particular, some of the most important sensors for this project - with their respectively measured signals - are reported in the following list [21]:

1. Vehicle sensors:
    (a) Vehicle speed;
    (b) 4 wheels speeds (or pulse count);
    (c) Yaw-rate;
    (d) Longitudinal acceleration;
    (e) Lateral acceleration;
    (f) Steering wheel angle;
2. GPS:
    (a) Latitude;
    (b) Longitude;
    (c) Elevation;
    (d) Speed;

   (e) Course angle (w.r.t. North);
   (f) HDOP (position accuracy);

3. Lane Recognition Camera (LRC):

   (a) Distance from left line;
   (b) Distance from right line;
   (c) Heading angle (w.r.t. lane);
   (d) Lane curvature;
   (e) Derivative of lane curvature over distance.

These signals are available in real time in the CAN network of the vehicle. It is important to note that the employed GPS is that which is normally installed on a production vehicle. The demo vehicle - which will be used only for the estimator algorithms design - will be also equipped with a differential GPS, which will provide the same outputs as the normal GPS but with much greater accuracy.

When lane markings are missing and hence the LRC functionality is impaired, the real time estimator algorithms will be used to provide the variables necessary for ADAS control functions (latitude, longitude, course angle and relevant state variables employed both in longitudinal and lateral control). During the design stage, the best achievable accuracy will be defined, as well as the confidence interval of the estimate in various clusters of scenarios.

For the sake of an easier comprehension of the next chapters, a simplified representation of the control system that will be studied and developed in this Thesis (figure 1.3) is now introduced. A more detailed description of each block's details and design procedure will be provided at a later time. What follows is a simple introduction for each component.

The block diagram represented in the figure resembles the actual system as it will be developed in the graphical programming environment offered by MathWorks Simulink. As a consequence, having this scheme in mind will help the comprehension of the role and hierarchy of each component that will be added to the overall system.

In this scheme the *Vehicle Dynamics* block constitutes the mathematical description of the *plant*, i.e. the system whose dynamics we want to control. As is suggested by its name,

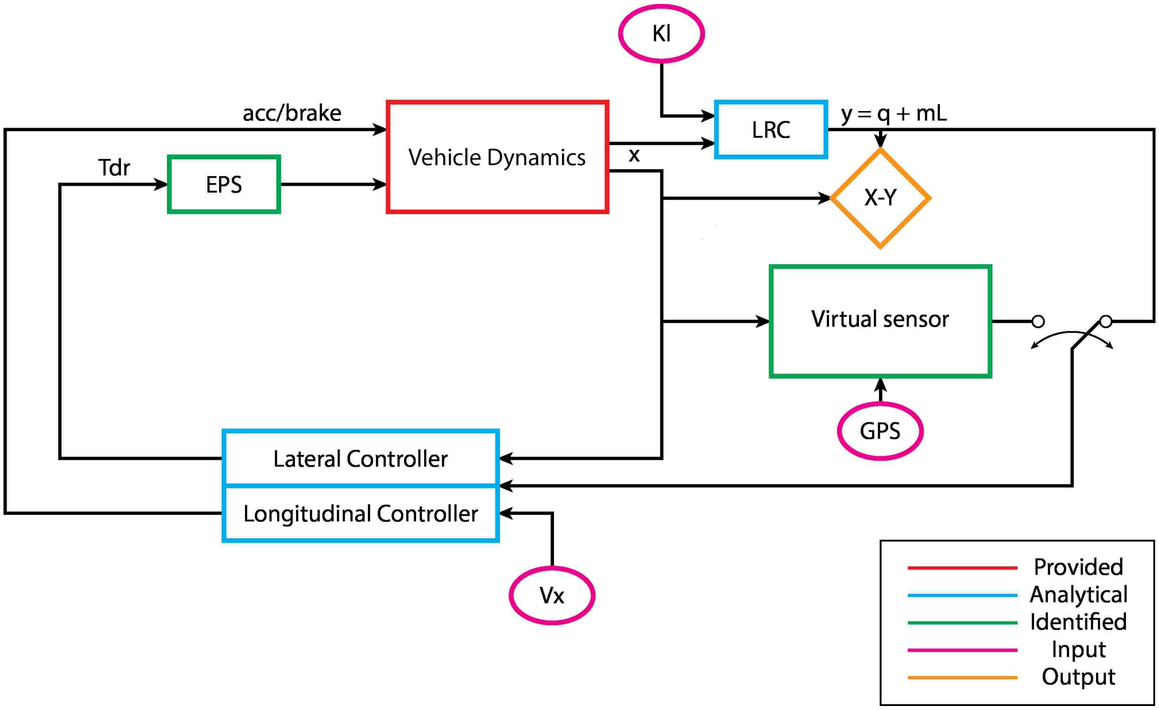


Figure 1.3: *The proposed Control Scheme*

this block contains all the necessary information to appropriately simulate the dynamics of the vehicle of interest (the 2015 Fiat *500X*). This block contains two MATLAB® S-Functions compiled and made available by CRF (Centro Ricerche Fiat) S.C.p.A.: the first one simulates longitudinal dynamics and the second one the lateral dynamics of the vehicle. The two are combined in order to realize a plant whose inputs and outputs allow one to design the control system. The symbol "x" refers to the state variables of the vehicle dynamics block: vehicle speed, wheels speeds, yaw rate, longitudinal and lateral acceleration, *et cetera.*

Nonetheless a discrepancy between the inputs of the vehicle dynamics block and the control algorithm is present. The lateral dynamics function, in fact, takes the angle of the steering wheel as the command input, whereas the steering action is typically performed by the control system, in terms of a torque to be applied to the steering column. This mismatch requires the addition of the *EPS* block, which simulates the action of the Electric Power Steering actuator. As will be discussed in detail in the appropriate section of this Thesis, the EPS device will be modeled as a black-box.

The *LRC* and *GPS* blocks simulate the information provided by the Lane Recognition Camera and the Global Positioning System, respectively. The former provides a linear approximation of the centerline of the lane in which the vehicle is traveling. The latter provides information about the position of the standardly equipped vehicle's on-board navigation system.

Considering suitable inputs coming from the LRC, the Vehicle Dynamics and - when the output of the camera is unavailable - the Position Observer, the *Control System* block drives the inputs of the plant in such a way as to obtain the desired behaviour of the vehicle. In particular, it drives the accelerator and brake pedals in order to match the prescribed longitudinal speed in all driving conditions. It then demands a torque from the EPS actuator which allows the vehicle to precisely track the previously mentioned approximation of the centerline of the lane.

The last block that needs to be introduced is the *Position Observer*, i.e. the "Virtual Sensor" (VS) which is intended to recover the precise position of the vehicle from GPS data. This is the most important component of the developed system and it represents the main unique contribution of this work. Although a precise physical and mathematical description of this block will be provided in the next chapters, it is convenient to introduce the idea that resides behind this crucial unit. Traditionally, when a non-measurable variable of a dynamic system needs to be reconstructed a two-step approach is followed: first, a mathematical model of the real system is derived and then an *observer* is designed in order to obtain an estimate of the missing variables. This approach, widely employed in control theory, allows for good performance with traditional estimator algorithms (Luenberger, Kalman) when the system that is considered is linear. In the presence of highly non-linear systems - as is the case here for the overall plant, sensors and actuators included - modified solutions must be employed (e.g. extended Kalman filter) which often cannot guarantee sufficient performance and robustness. In order to solve this problem a new, one-step approach has been developed by researchers from Politecnico di Torino and Berkeley University [22]. The result of this approach is a *virtual sensor* directly estimated from the non-linear plant (with any preferred identification technique), which can be proven theoretically to provide quasi-optimal solutions even in worst-case estimation scenarios and boundedness properties.

In the Control Scheme introduced in figure 1.3 a virtual observer is employed to precisely reconstruct the position of the vehicle in the lane when the output from the camera is missing or of poor quality.

Finally, figure 1.3 underlines which are the main inputs and outputs of the system. When the controller is driven by the LRC, the main inputs are the desired longitudinal speed of the vehicle v_x and the instantaneous value of road curvature K_l. Conversely, when the controller is fed by the virtual sensor the main inputs are v_x and the GPS signal coming from the on-board navigation system. The main output of the system is the linear approximation $y = q + mL$ of the lane centerline, and, in particular, the instantaneous distance q between the vehicle's center of gravity and the centerline of the lane. This quantity, as it will be shown, can be easily converted into the trajectory followed by the vehicle in the X-Y plane.

1.4 Thesis Organization

This thesis is organized as follows:

In chapter 2, a brief review of some theoretical concepts useful for the following chapters is offered. Furthermore, the mathematical justification of the operation principles of a virtual sensor is discussed.

In chapter 3, an extended review of the literature concerning the general concepts of automated driving, the system components and previous works conducted in the field will be carried out.

In chapter 4, a detailed description of the design of the model will be provided. A section is dedicated to each of the major components of the model.

In chapter 5, a general descriptions of the simulation and validation procedures is offered. Moreover, the mathematical reasoning to obtain the trajectory followed by the vehicle is presented.

In chapter 6, the results pertaining to the three versions of the model (uncontrolled, controlled via LRC and controlled via virtual sensor) are shown and commented.

Finally, in chapter 7, a summarizing conclusion is drawn and some recommendations and suggestions for future work are offered.

Chapter 2

Theory

In this chapter some theoretical concepts are briefly exposed for easier comprehension of the following parts of this thesis.

First, the analytical models of vehicle longitudinal and lateral dynamics are discussed so as to provide a reference for the study of uncontrolled vehicle dynamic behaviour. In particular, the famous “bicycle model” for vehicle lateral dynamics is presented. This model will be widely employed in the definition of the lateral controller.

In the second part of this chapter the mathematical reasonings behind the concept of the virtual sensor are illustrated schematically.

Finally, in the last part of the chapter two important techniques for model identification - Neural Networks and Non-Linear ARX - are briefly introduced.

2.1 Vehicle Dynamics

In the control system for automated driving implemented in this work the block representing vehicle dynamics is provided and considered as a black-box model. This means that the block is treated only in terms of inputs and outputs, without any knowledge of its internal workings. Nonetheless, a brief introduction to a linearized version of the longitudinal and lateral dynamics in a vehicle is sketched here in order to define the analytical relation between the variables involved. Moreover, the development of the control algorithms will consider a linearized rendering of the vehicle dynamics block.

2.1.1 Tire Contact Modeling

Except for aerodynamic forces, all external actions influencing vehicle dynamics are exerted at the tire-road contact [55]. The understanding of the physics behind this phenomenon is essential for the development of meaningful equations of motion. Figure 2.1 illustrates the main variables involved in longitudinal tire dynamics: wheel angular velocity (ω_w), vehicle speed (V), vertical load (F_z), radius of the wheel (Γ_w), driving torque (T_e), braking torque (T_b), traction and "friction"[1] forces (F_t and F_f, respectively).

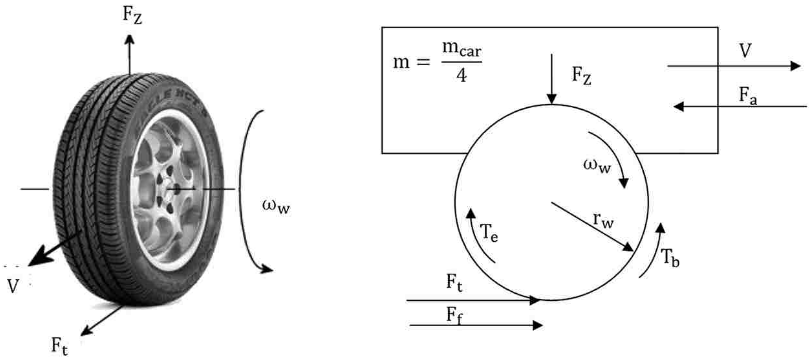


Figure 2.1: *Longitudinal tire dynamics*

The wheel angular dynamic equation is:

$$\dot{\omega}_w = [T_e - T_b - \Gamma_w F_t - \Gamma_w F_f]/J_w \tag{2.1.1}$$

where J_w is the moment of inertia of the wheel.

The tractive force developed at the contact patch is dependent on the (longitudinal) *wheel slip*, i.e. the difference between the theoretical velocity of the wheel[2] and the actual velocity of the wheel, normalized by the maximum of the two (theoretical for braking, actual for acceleration) [56]. The adhesion coefficient $\mu(\lambda)$ is a function of the wheel slip λ (figure 2.2).

[1] Here by "friction force" is intended the force resulting from the rolling resistance of the tire.

[2] The theoretical angular velocity of the wheel is the speed of the vehicle divided by the wheel radius $\omega_v = V/\Gamma_w$, i.e. the velocity of the wheel in pure rolling condition.

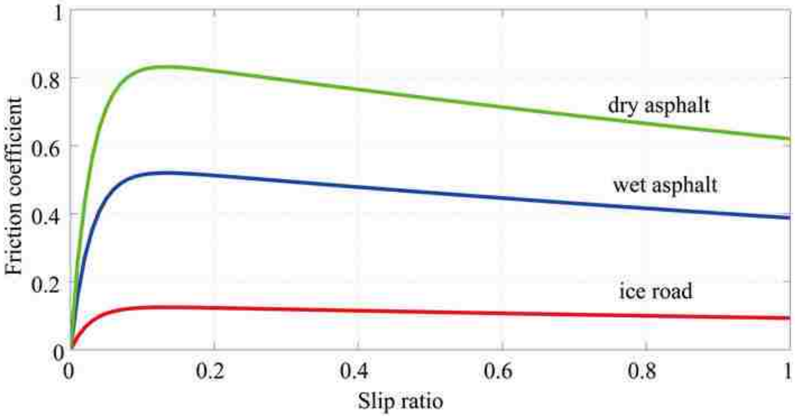


Figure 2.2: *Typical $\mu(\lambda)$ curves for different road conditions. If scaled by the appropriate vertical load F_z the ordinate axis represents the tractive force F_x*

The tire tractive force is given by:

$$F_t = \mu(\lambda) F_z \tag{2.1.2}$$

where the vertical load F_z depends on the mass of the vehicle and the steering and suspension dynamics.

With the tire lateral dynamics, a somewhat similar phenomenon occurs: when the trajectory of the wheel is bended, a deformation in the tire contact patch is introduced (figure 2.3) that determines the production of a force [57]. The *sideslip angle* α is due to

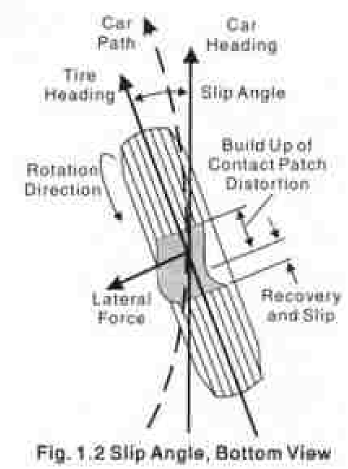


Figure 2.3: *Tire contact patch deformation in a bend. Notice the (Side)slip angle α [58]*

the elastic nature of the tire structure, which allows the creation of an angle between the direction the tire is pointing and its actual direction of travel (when a tread element enters the contact patch, the friction between the road and the tire causes the tread element to remain stationary, yet the tire continues to move laterally). Considering wheel speed components v_x and v_y, α we can be defined as:

$$\alpha = -arctan\left(\frac{v_y}{|v_x|}\right) \tag{2.1.3}$$

The cornering force F_y developed by the tire is dependent on the sideslip angle α according to a function qualitatively very similar to the one shown in figure 2.2. For small angles this relation is approximately linear and it is possible to define the *cornering stiffness* parameter of the tire as:

$$C_\alpha = \frac{F_y}{\alpha} \tag{2.1.4}$$

2.1.2 Longitudinal Dynamics

In figure 2.4 the most relevant forces acting on a vehicle traveling on a straight road are shown [58]. Considering these forces we can write:

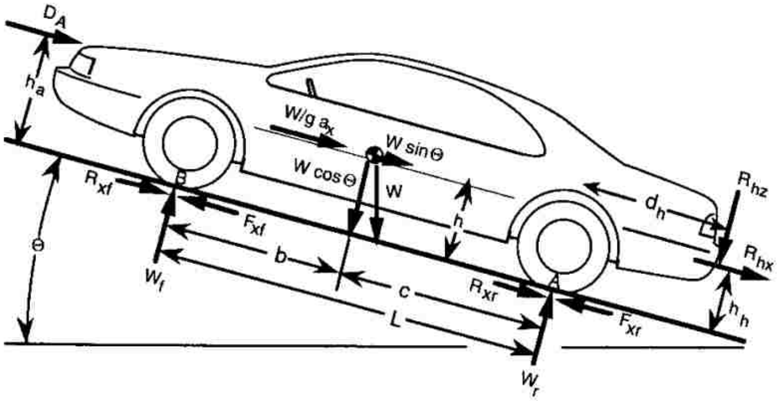


Figure 2.4: *Forces acting on the vehicle in longitudinal motion [58]*

$$Ma_x = F_x - R_x - D_A - R_{hx} - W\sin\Theta \tag{2.1.5}$$

where R_x is the rolling resistance force, D_A the aerodynamic drag force, R_{hx} the towing force and F_x is the tractive force. The longitudinal acceleration a_x, presuming there is adequate power from the engine, is limited by the coefficient of friction at the tire contact patch.

2.1.3 Lateral Dynamics

Assuming there is no interaction between longitudinal and lateral vehicle dynamics, the latter can be described making reference to a basic *bicycle model*, i.e. a simplified model where the two sides of the car are “fused” together (figure 2.5). In this figure, R is the turning radius, δ the steering angle, and α_f and α_r the front and rear slip angles, respectively. These angles allow the tires to develop cornering forces such that the “centrifugal” force acting on the vehicle's center of gravity is compensated:

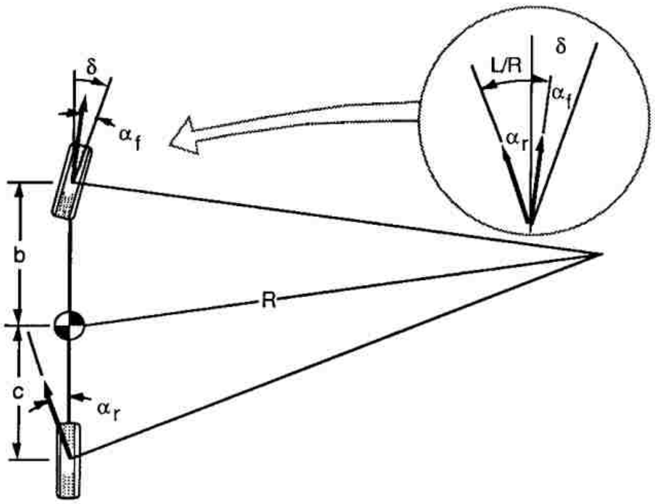


Figure 2.5: *Cornering motion studied with the bicycle model [58]*

$$\sum F_y = F_{yf} + F_{yr} = M\frac{V^2}{R} \tag{2.1.6}$$

Now for the vehicle to be in equilibrium the sum of the moments acting around the center of gravity must be null, hence:

$$F_{yf} = F_{yr}\frac{c}{b} \tag{2.1.7}$$

Substituting this equation in the previous, defining the weight on each axle, W_f and W_r, and solving for the slip angles we find

$$\alpha_f = W_f V^2/(C_{\alpha f} gR) \qquad \alpha_r = W_r V^2/(C_{\alpha r} gR) \tag{2.1.8}$$

which inserted in the geometrical relation expressed by figure 2.5 (i.e. $\delta = 57.3L/R + \alpha_f - \alpha_r$ with L the wheelbase)[3] gives:

$$\delta = 57.3\,\frac{L}{R} + \left(\frac{W_f}{C_{\alpha f}} - \frac{W_r}{C_{\alpha r}}\right)\frac{V^2}{gR} = 57.3\,\frac{L}{R} + K a_y \tag{2.1.9}$$

where K is the *Understeer Gradient* $[deg/g]$.

Equation 2.1.9 is a fundamental equation concerning the turning response of a vehicle: it describes how the steering angle δ varies with the radius of the turn R and the lateral acceleration a_y. The parameter K can take three possible values:

- $K = 0$

  When such a vehicle is negotiating a constant-radius turn, no change in steering angle is required when the speed is varied. This is the so-called *Neutral Steer* condition.

- $K > 0$

  On a constant-radius turn, the steer angle must be increased with increasing speed by K times the lateral acceleration in g's. This condition is termed *Understeer*[4].

- $K < 0$

  On a constant-radius turn, the steer angle must be decreased with the speed. This third possible condition is termed *Oversteer*.

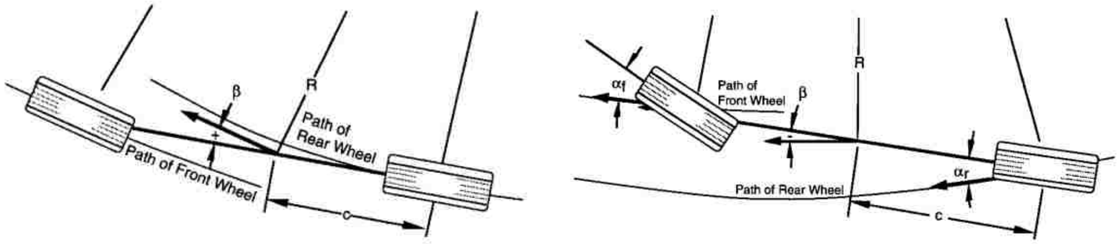
When discussing of lateral dynamics, one last fundamental quantity has to be introduced. A "sideslip angle" was proposed when discussing the tire deformation so as to quantify the mismatch between the direction the tire is pointing and the direction the center of the wheel is following. A similar concept can be applied to the whole vehicle: a sideslip angle may be defined at any point on the vehicle as the angle between its longitudinal axis and the actual direction of travel at that point [58]. When taking the center of gravity, the *Vehicle Sideslip Angle* β is defined as shown by figure 2.6.

For any speed, the vehicle sideslip angle β at the center of gravity is:

$$\beta = 57.3\,\frac{c}{R} - \alpha_r = 57.3\,\frac{c}{R} - \frac{W_r V^2}{C_{\alpha r} g R}. \tag{2.1.10}$$

[3] $1\,rad = 57.2958\,deg$

[4] The vehicle here considered, the 2015 Fiat *500X*, has a positive understeer gradient.



(a) Low-speed turn

(b) High-speed turn

Figure 2.6: *Vehicle Sideslip Angle β (positive in clockwise direction) [58]*

2.2 Virtual Sensors

In this second part of chapter 2, the concept and mathematical foundation of Virtual Sensors (VS) are presented. As already underlined, the application of a virtual sensor for precise vehicle position reconstruction when the camera output is degraded will represent the main original conclusion of this work.

The innovative approach of nonlinear variable identification through virtual sensors was developed by Professors M. Milanese and C. Novara from Politecnico di Torino, Italy and by Professors K. Hsu and K. Poolla from University of California at Berkeley, USA. A detailed mathematical treatment of the problem and an application to the Lorenz attractor nonlinear system are offered in [22].

Mathematical Justification

In control theory, a (state) *observer* is defined as follows [81].

Definition. *State Observer* A state observer is a system that provides an estimate of the internal state of a given real system, from measurements of the input and output of the real system.

How well internal states of a system can be inferred from the knowledge of its external outputs depends on the *observability* of the system. Now, let us consider a *non*-linear discrete time system in state space form with input u and outputs $[y, z]$:

$$x_{t+1} = F_x(x_t, u_t)$$
$$y_t = H_y(x_t, u_t)$$

where $x_t \in X \subset \Re^n,\ u_t \in U \subset \Re^{mu},\ y_t \in Y \subset \Re^{my}$ and F_x, H_y are *unknown* continuous and differentiable functions.

Let z_t be any variable of interest defined in $\Re^{mz}$ ($z_t \in Z \subset \Re^{mz}$, a subspace included in $\Re^n$); also noise corrupted measurements of $\tilde{u}_t, \tilde{y}_t$ are available for all times t. It is of interest to know z_t for $\forall t$ starting from a certain time instant t_m.

The standard approach for observer design considers a two-step procedure:

1. Identification of a process model in terms of state space equations (i.e. of matrices A, B, C and possibly L):

$$\begin{gathered}\dot{x}(t) = Ax(t) + Bu(t) + L\xi(t), \quad E\{\xi(t)\} = 0, \quad \forall t \\ x(t) \in \Re^n, \quad u(t) \in \Re^m, \quad \xi(t) \in \Re^p \\ y(t) = Cx(t) + \theta(t), \quad y(t) \in \Re^p, \quad \theta(t) \in \Re^p \ ;\end{gathered}$$

2. Design of an observer based on the identified model:

$$\begin{aligned}\dot{\hat{x}}(t) &= A\hat{x}(t) + Bu(t) + H\left[y(t) - C\hat{x}(t)\right] \\ &= \left[A - HC\right]\hat{x}(t) + Bu(t) + Hy(t) \\ \hat{y}(t) &= C\hat{x}(t)\ .\end{aligned}$$

The system is to be preferably determined from "inside", i.e. based upon physical laws describing the process. The observer is often a Kalman filter (KF): $H = K_{OBS}$ is the optimal Kalman gain matrix, as in figure 2.7.

The main problem with this procedure is that problems in automotive dynamics are usually strongly non-linear. As a result, the performance of the KF, which is designed to be a minimum variance filter, deteriorate due to modeling errors (unavoidable in such complex

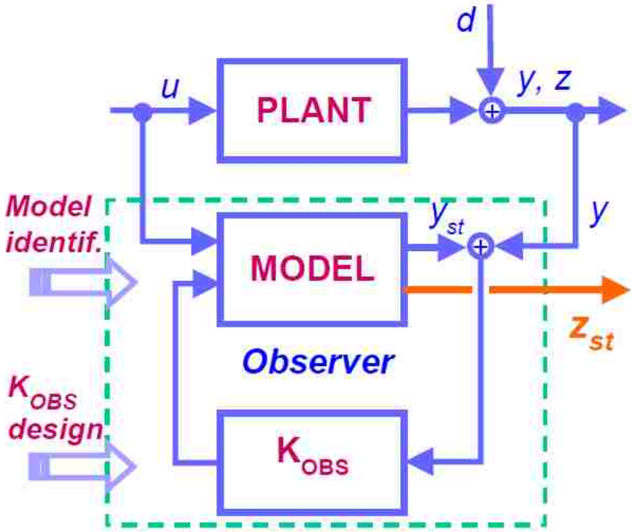


Figure 2.7: *Standard observer design approach (from [82])*

and nonlinear dynamics). Furthermore, observers are typically implemented as Extended KF, i.e. approximations of the KF where the gain matrix H is variable and evaluated on-line at every sampling time. The discrete and recursive algorithm employed for this computation provides poor performance and it is *not* even able to guarantee bounded estimation errors.

The practical realization of this kind of filter is then complex and stable estimation performance is not guaranteed. In more recent times, an increasing interest is developing for new approaches based on the *direct identification* of the estimator model starting from a set of noisy data.

This is the so called one-step approach: it implies the identification of an overall model including by itself the dynamic behaviour described by the previously mentioned equations, as in figure 2.8. Let us consider again the non-linear discrete time system in state space form with input u and outputs $[y, z]$:

$$x_{t+1} = F_x(x_t, u_t)$$

$$y_t = H_y(x_t, u_t)$$

$$z_t = H_z(x_t, u_t)$$

where H_z is a continuous and differentiable function as well. Also, noise corrupted measurements of $\tilde{u}_t, \tilde{y}_t$ are available for all times t whereas z_t is measured only for $t < T_m$. It is

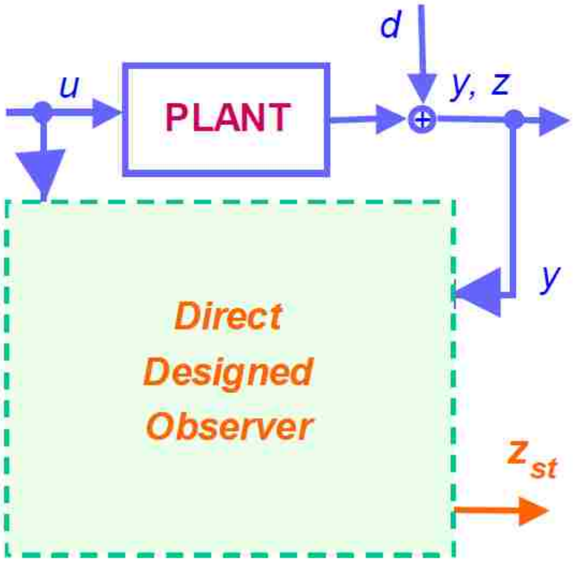


Figure 2.8: *One step approach for virtual sensor design (from [82])*

of interest to determine z_t for $t > T_m$.

Milanese *et al.* demonstrated in [83] that, for (F_x, H_y) observable, $\exists F_0$ and integers n_u, n_y such that the variable of interest z_t may be calculated as:

$$\tilde{z}_t = F_0(\tilde{Y}_t, \tilde{U}_t) + d_t, \quad t = 0, 1, 2, ..., T_m$$

$$\tilde{Y}_t = [\tilde{y}_t, \tilde{y}_{t-1}, ..., \tilde{y}_{t-ny+1}]$$

$$\tilde{U}_t = [\tilde{u}_t, \tilde{u}_{t-1}, ..., \tilde{u}_{t-nu+1}]$$

where F_0 is a parametric functional and d_t is due to the uncertainty of z_t, y_t and u_t and where the sequence of d_t is supposed to be bounded.

As a consequence, the problem of identifying the observer representing the "virtual" sensor is transformed in the *estimation of a functional* $F_A(Y, U)$ as a parametric approximation of F_0, computed using any desired nonlinear estimation method within a bounded subset of the regressor space z_t and its appropriate orders $[n_u, n_y]$. In this one step approach the observer model is built up from "outside", by means of an identification process. Virtual sensors derived in this fashion can be theoretically proven to provide *quasi-optimal* solutions even with worst-case estimations and to guarantee boundedness properties, realizing anyway "minimum" variance estimators (a similar result is not assured by KF techniques).

They appear to provide good results even for complex nonlinear models and work with reduced sets of measured data.

In chapter 3 some applications related to virtual sensors will be discussed. In chapter 4 the implementation in this project of the virtual sensor for the problem of precise vehicle position reconstruction is presented in detail. Finally, chapter 6 describes the performance of the system in which the controller is using the information coming from the virtual sensor.

2.3 Neural Networks and System Identification through ARX

An important role in this project is occupied by the necessity of identifying models from input-output data pairs. As will be discussed in more detail in the following chapters, black-box modeling is necessary mainly for two reasons:

- To identify models from experimental data, when the inner workings of the real system are unknown or too complex to be fully reproduced here without making the system overly intricate in those areas that are not the main focus of the project;

- To linearize complex non-linear dynamics for the purpose of focusing attention on the bulk of the input-output relation, letting out all non-linearities and subtleties not strictly necessary.

The first case - which (more strictly speaking) pertains to the category of black-box models - makes reference to the model of the Electric Power Steering actuator whose internal, highly non-linear characteristics are supposed unknown.

The second point is mainly related to the necessity of developing simple yet high-performing control for the longitudinal and the lateral dynamics of the vehicle. Despite their simplicity, in fact, PID and LQR controllers can achieve good performance if their tuning is done accurately and robustly. Now, because those control techniques are linear in nature, they require first and foremost a linearization of the plant they aim to control.

In both cases model identification techniques can be successfully used to complete the task. Let us introduce the two techniques that were used in this project: Artificial Neural Networks and ARX models.

2.3.1 Artificial Neural Networks

Artificial Neural Networks (ANNs) are models based on the biological structure of the brain. They are constituted by interconnected processing units, the *neurons*. The strength of each connection can be altered by changing the value of a numeric weight. This adaptivity makes ANNs capable of learning relations between data (see [105] for details about training algorithms and procedures). A neural network is then suitable for defining black-box models when only input-output pairs are known. Niu *et al.* in [106] demonstrate how feedforward neural networks can be used to perform model fitting over complex data with performance comparable with traditional regression analysis approaches.

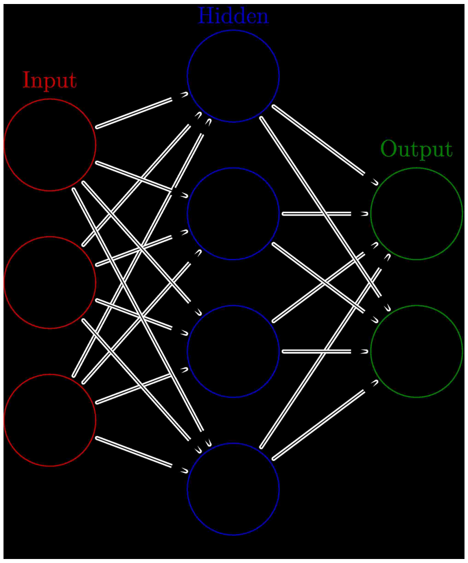


Figure 2.9: *ANN with 3 inputs, 4 hidden nodes, 2 outputs*

As figure 2.9 shows, the term "network" in Artificial Neural Network refers to the interconnection of several neurons: *input* neurons which send data via synapses to the second layer of (*hidden*) neurons, and then via more synapses to the third layer of *output* neurons. More complex networks will have two or more layers of hidden neurons. In all cases, the synapses store parameters (called "weights") that manipulate the data in the calculations.

Let us consider a simple (hidden) neuron: figure 2.10. The processing of the scalar input p is as follows:

- The input p is multiplied by the (scalar) weight w to form the product wp;

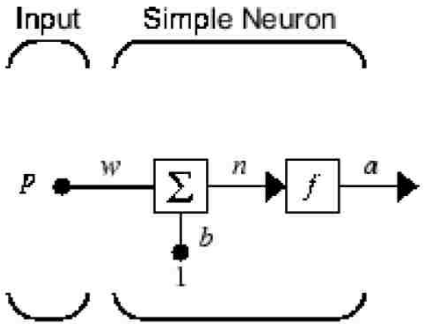


Figure 2.10: *A simple neuron [103]*

- The weighted input wp is added to the scalar "bias" b to form the net input n;

- Finally, the net input is passed through the *transfer function* f, which produces the scalar output a.

The fundamental aspect of this process is that parameters w and b are both adjustable scalar parameters of the neuron: the central idea of neural networks is that such parameters can be adjusted so that the network will predict a known output within a desired level of accuracy.

The output of the neuron $a = f(wp + b)$ depends also on the transfer functions f which maps the net input to the output. Different transfer functions can be used, two of the most common ones are the linear transfer function and the sigmoid transfer function.

The workflow for the neural network design process consists in seven primary steps:

1. Collect data;

2. Create the network;

3. Configure the network;

4. Initialize the weights and biases;

5. Train the network;

6. Validate the network;

7. Use the network.

Configuring the network means making it compatible with the problem at hand, i.e. with the collected sample data. The parameters of the network (weights and biases) must then be tuned in order to optimize the performance of the network. This tuning process is referred to as *training* and requires the network to be provided with example data.

The training of ANN is the fundamental phase in which an *optimal* solution to the problem is sought, where "optimal" means that no other solution has a "cost" lower than that of the optimal solution. If examples of input-output data pairs are given to the network and the objective is to closely match the examples, the learning is referred to as *supervised*. In this case, a commonly used cost function is the MSE (Mean Square Error), which is used to minimize the average squared error between the network's outputs and the target values.

A simple algorithm to minimize the MSE cost function is the *gradient descent*: a local minimum of the function is searched by taking steps proportional to the negative of the gradient of the function at the current point. When one tries to apply gradient descent to feed-forward artificial networks of non-linear neurons (called "multilayer perceptrons"), one obtains the common and well-known *backpropagation algorithm* for training neural networks. MATLAB's Neural Network Toolbox can implement this and more sophisticated training algorithms.

Some of the most important tasks ANNs are used for include:

- Function approximation (regression analysis): system modeling, *etc.*;
- Classification: pattern and sequence recognition, *etc.*;
- Data processing: filtering, clustering, *etc.*
- Robotics;
- Control.

2.3.2 Non-linear ARX

A non-linear ARX (*nlarx* or *NARX*) is a nonlinear autoregressive model which has exogenous inputs [107]. This means that a model of this kind links the current value of a time series to both a) the past values of the same series (*autoregressive*) and b) the current and past values of an externally determined (*exogenous*) series that acts on the series of interest (figure 2.11).

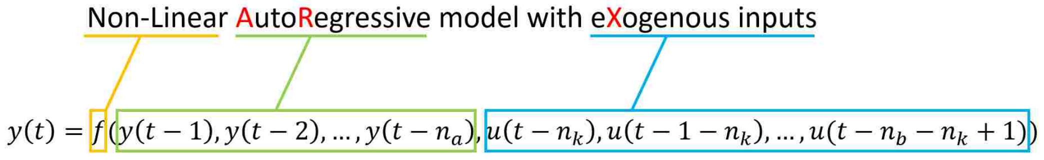

Non-Linear AutoRegressive model with eXogenous inputs
$y(t) = f(y(t-1), y(t-2), \ldots, y(t-n_a), u(t-n_k), u(t-1-n_k), \ldots, u(t-n_b-n_k+1))$

Figure 2.11: `nlarx` *structure*

The basic ARX model from which the *nlarx* is derived is a *linear* SISO (single-input single-output) ARX model, which has the following structure:

$$y(t) + a_1 y(t-1) + a_2 y(t-2) + \cdots + a_{na} y(t-n_a) = b_1 u(t-n_k) + b_2 u(t-1-n_k) + \\ + \cdots + b_{nb} u(t-n_b-n_k+1) + e(t) \tag{2.3.1}$$

where n_a is the number of past output values, n_b the number of (current and) past input values, n_k the input delay and $e(t)$ the error source. Here $y(t)$ is the variable of interest and $u(t)$ is the externally determined variable.

This structure implies that the present value of the output $(y(t))$ can be evaluated as a weighted sum (linear combination) of past output values and current and past input values. The delayed input and output variables $(y(t-1), y(t-2), \ldots, y(t-n_a), u(t-n_k), u(t-1-n_k), \ldots, u(t-n_b-n_k+1))$ are called *regressors*.

The structure of equation 2.3.1 can be extended to create a non-linear form replacing the weighted sum with a non-linear mapping function, f:

$$y(t) = f(y(t-1), y(t-2), \ldots, y(t-n_a), u(t-n_k), u(t-1-n_k), \ldots, u(t-n_b-n_k+1)) \tag{2.3.2}$$

where the regressors can be delayed input-output variables and/or more complex non-linear expressions (e.g. products, powers, *etc.* of delayed input and output variables). Figure 2.12 shows how a non-linear ARX block operates (see [103]). The “nonlinearity estimator block” links the regressors to the output of the system using both linear and non-linear functions.

The numbers n_a, n_b and n_k are the *orders* of the model and, concerning the values they should assume, nothing but initial guidance exists in the literature. Data to be used

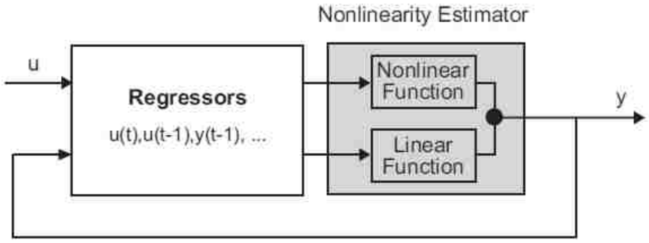


Figure 2.12: *`nlarx` model in a simulation scenario*

in the system identification are first divided into estimation and validation sets (e.g. by splitting the experimental outcomes in two data sets using random indexes). Subsequently, in MATLAB, data are prepared for nonlinear identification by creating `iddata` objects. It is important to split available experimental data into estimation and validation sets in order to properly assess the capability of the model to work well not only on data which were used for the determination of its own coefficients (estimation data), but also on new data not employed to determine the final form of the model (validation data).

Chapter 3

Background and Literature review

In this chapter an in-depth look at the material available in the literature which proved to be useful for the progress and the completion of this work is provided. In order to rationally expose the content of the chapter, the latter is subdivided in three parts.

The first part exposes the general context in which this project is inserted: the importance of vehicle safety, the classification of ADAS systems, the various levels of automation in which they can be arranged and the concept of data fusion.

The second part provides a detailed description and relevant examples for the components constituting the model developed in this Thesis. The practical implementation of virtual sensors is discussed. In this way a backbone for the discussion of the model presented in chapter 4 is offered.

Finally, in the last part, a review of the work done in the field by automakers, startups and research institutes is provided. Additionally, an example of an Advanced Driver Assistance System designed to cope with the absence of ground lanes in urban environments is also examined.

3.1 General Concepts

3.1.1 Vehicle Safety

The problem of automobile safety is a complex one. A thorough understanding of this matter requires one to consider not only the vehicle itself, but also the construction, the equipment and the set of regulations aimed at reducing the number and severity of road accidents. The World Health Organization reports [23] that there were 1.25 million road traffic deaths globally in 2013, i.e. a traffic-related fatality occurs every 25 seconds. Half of these fatalities involve "vulnerable road users" (motorcyclists, pedestrians and cyclist) [24] and only 28 countries (7% of world's population) have suitable legislation concerning traffic accidents [25]. In low-income countries one-percent of the globally registered vehicles causes 16% of the world's road traffic deaths [25]. These numbers underline how important the creation of modern transport infrastructures and the diffusion of suitable legal countermeasures are in the pursuit of a reduction of traffic-related deaths.

Passive Safety

Considering the vehicle itself, it becomes mandatory to design the car body is such a way as to protect occupants during a crash. This is the task of the so-called *passive safety* features. Occupant protection is achieved by mitigating crash forces and limiting the contact of delicate body parts against the structure of the vehicle. For this purpose, passengers are then kept in a "life space" throughout the crash, and this life space is kept as safe as possible. Crumple zones are designed to absorb and distribute crash forces before they can reach passengers. At the same time, seats, airbags and seat-belts help retain the occupants inside the life space. Engineering societies (e.g. SAE [30]), governmental agencies (e.g. NHTSA [31]) and automakers dedicate ample efforts at studying and advertising passive safety systems. These safety features are extremely important when it comes to decreasing the severity of crash-related injuries, and as such they have undergone significant development in recent years. Modern vehicles, in fact, are often equipped with pretensioning systems for a quicker action of seat-belts, advanced headrests for the reduction of the risk of whiplash, numerous airbags to prevent the contact of several structural components with the passengers and sophisticated, FEA-designed, energy-dissipating chassis components.

Active Safety

That said, several studies [26, 27, 28] shows the extensive impact of *human error* on motor vehicle crashes. In particular, the most thorough analysis of crash causation, the *Tri-Level Study of the Causes of Traffic Accidents* published in 1979 [29], found that "human errors and deficiencies" were a definite or probable cause in 90-93% of the incidents examined. In [35] the principal causes of fatal accidents are identified (in order of importance) as the following:

- Loss of alertness;
- Alcohol;
- Fatigue.

A substantial possibility for the reduction of the accidents linked to vehicular accidents appears then to avoid an accident *before* it even occurs. This idea of supporting the driver in their task has fostered the development of *active safety* systems. These safety features are active prior to an accident, in contrast with their passive counterpart that are "active" during an accident. Examples of very successful active safety features comprise the ABS (Anti-lock Braking System), whose objective is to prevent wheel locking and to maintain vehicle drivability even in extreme braking maneuvers, and the ESC/P (Electronic Stability Control/Program), which — trough a combination of actions on the engine power output and the individual brakes — helps in keeping the vehicle on the intended path even in the presence of severe yaw angles (see figure 3.1 [32]).

In recent years, active safety is becoming more and more synonymous with systems able to grasp an understanding of the state of the vehicle with the task of both avoiding and minimizing the effect of a crash. These sensor-based, forward looking systems fall in the already mentioned category of *ADAS* systems. Features like brake assist systems and adaptive cruise controls can be considered active safety systems.

The main reason why these safety-targeted driver assistance systems are becoming increasingly diffused in modern vehicles production is that, although the improvement of

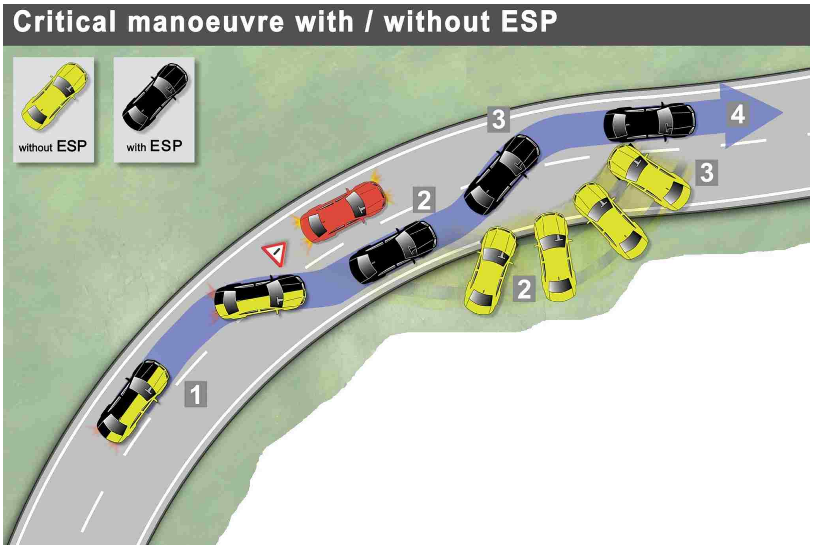


Figure 3.1: *ESP intervention in an extreme steering maneuver*

passive and early active safety systems has allowed a significant reduction in the number of road fatalities (see the already cited European Commission report, [3]), the trend is now slowing down, showing the need for a leap forward in the treatment of road safety. This will be most likely achieved by initially supporting and ultimately replacing the human driver, arguably the greatest threat to car safety.

3.1.2 ADAS

A unique definition of the expression ADAS -Advanced Driving Assistance Systems - appears to be lacking in the literature. Nonetheless, a tentative explanation (see also [33]) of the term can be given as follows:

Definition. *ADAS* Advanced Driver Assistance Systems are systems designed to automate/adapt/enhance vehicle systems for safety and better driving. These technologies allows the prevention of incidents by warning the driver of potential dangers, by implementing preventative actions or by taking control of the vehicle.

These systems could help the driver to manage dangerous situations created by external factors (e.g. a forward collision warning system alerting the driver of a sudden braking of the vehicle ahead) or they could compensate for shortcomings of the driver them-self, in terms of their capabilities or awareness. Even if this last point could pose some problems to salesmen (as jokingly underlined by [6]), these systems are indeed moving into the marketplace.

ADAS Benefits, History and Problems

There are a number of reasons [34] why, in recent years, electronic driving aids are becoming more and more the focus of large investments from automakers and suppliers. As mentioned in the previous section, this clearly relates first and foremost to the unacceptable number of accidents, i.e. to safety. ADAS systems can, in fact, be used to support the driving task, alert the driver of potentially hazardous situations and, if needed, even take over control. Other important arguments are of economic nature: saving time is a way of saving money. Many more vehicles could be accommodated on a highway under some form of automatic control and inter-vehicle communications. Of course, comfort of the driver is another crucial selling point. Drivers with reduced or impaired capabilities — e.g. elderly, inexperience drivers — could benefit of the backing of ADAS systems. Finally, environmental concerns (reduced fuel consumption and hence pollutant emissions) also play a relevant role in the diffusion of electronic driving aids.

Even though, for many drivers, these systems have only recently started to become an integral part of their every-day routes, driver assistance systems have a considerable history of development. One of the very first initiatives concerning these kinds of systems is the "Prometheus" project, started by several European car manufacturers and research institutes in 1986. The European Union launched the "DRIVE" (Dedicated Road Infrastructure for Vehicle Safety in Europe) program shortly after this. Under these projects, considerable efforts were put in solving practical problems and in defining requirements and design standards for intelligent driver support systems [36].

Now, these systems are designed not only for automobile drivers but also for professionals. They promise, in fact, an always active copilot to reduce the stress and improve vigilance on long journeys. Nonetheless, several studies — e.g. Van Ouwerkerk *et al.* (1986), Janssen, Wierda and Van der Horst (1992) — show how carefully any impact on driver alertness and vigilance must be assessed before a new system is released on the market. In particular, behavioral changes could occur in the driver, with a potential threat for safety. Poorly designed systems inside vehicles, in general, can have negative effects on the human driver. Acceptance of this kind of system by the end-user is also a critical aspect: ADAS devices limited to provide information are most likely to be well received, whereas more intrusive devices could be perceived skeptically.

Speaking of behavioral changes, Brookhuis *et al.* [34] point out that automation could be linked to an increase of the reaction time of the driver. There are also studies which show how a continuous monitoring of the systems looking for malfunctions could actually lead to high workload, even if the processing requirements for these tasks are low in itself. Another source of stress for the driver could be the following: if the normal operation of the vehicle is carried out by the automatic control systems, the driver will need to face only the abnormal driving conditions (those resulting from some problem). But, as a consequence of the automation itself, the experience of the driver will be limited, while abnormal conditions may require complex and unusual actions. This scenario gets even worse considering that the response of the driver could be sup-optimal because of the sudden pressure induced by an emergency, see Bainbridge (1983). All of this is linked to so-called “complacency” — Wiener and Curry (1980) — which is an attitude of over-dependence from automated systems. The other side of the coin is, as mentioned, the acceptance of ADAS: the resistance to giving up part of the control of the vehicle. Bekiaris *et al.* showed that the driver population could be reluctant to abandon vehicle control, except for emergency situations.

In conclusion, while ADAS systems hold great potential, their acceptance will be heavily dependent on how good their performance will prove to be, on how clear their benefit to the end-user will be, and on how legislation will distribute responsibility and liability.

ADAS Classification

Currently available Advanced Driver Assistance Systems are mainly designed to support the task of the driver, and not to replace it. These systems, inserted in the domain of active safety, aim at avoiding accidents before they occur. In the following an overview of the most important ADAS devices is presented. Both currently available solutions and near-future prospects are briefly introduced.

The main *commercially available systems* are divided into the following categories [5]:

- **Lateral Control**

  Lateral control ADAS systems are mainly subdivided into the following three subcategories.

  - *Lane Departure Warning* (LDW) systems, whose task is to warn the driver of an unintentional — i.e. not preceded by blinker activation — change of lane (figure 3.2 [37]).

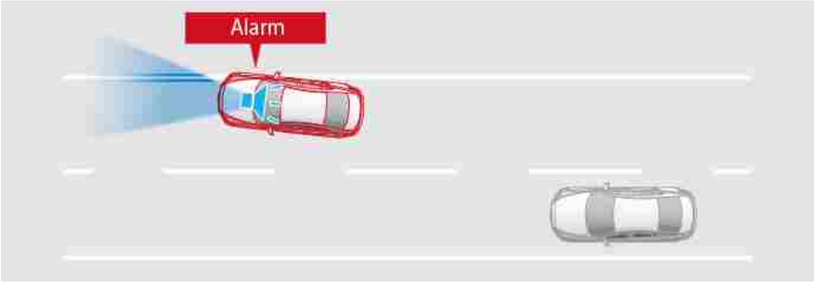


Figure 3.2: Lane Departure Warning system

The relevance of this kind of system is underlined by Headley (2005), who reports how 55 % of fatal crashes are caused by unintended lane departures. These systems relies on the use of video cameras able to recognize lane markings on the road. Variations of these devices can combine steering wheel vibrations with actual corrections of the steering wheel angle via the application of a suitable torque (Siemens VDO, 2005). The action of the system can be overridden if needed.

  - *Blind Spot Detection* systems aim at enhancing driver awareness of vehicle's surroundings by providing information whether vehicles, cyclists or pedestrians are present in the area not directly or indirectly visible to them. The driver is usually informed by a lamp on the side mirror or the A-pillar triggered by a difference in speed between driver's vehicle and incoming traffic. Detections of the objects in the blind spot area is usually carried out through infrared sensors. These *passive* sensors are less expensive and sometimes more reliable (Wang *et al.* (2005)) than active ones and their principle of operations relies on the temperature readings of the blind-spot area: if vehicles or pedestrians are present a difference in temperature with respect to a reference part of the road will be measured.

– *Lane Change Assistance* (LCA) systems integrate the two previously discussed devices in order to warn the driver if they are returning to the original lane too soon after an overtake.

- **Longitudinal Control and Avoidance Systems**

  Some of the most relevant longitudinal control and avoidance systems can be grouped in the following categories.

  – *Adaptive Cruise Control* (ACC) expands the functionality of a traditional cruise control systems by ensuring that an appropriate distance from the vehicle ahead is kept even if the latter is decelerating (figure 3.3 [38]).

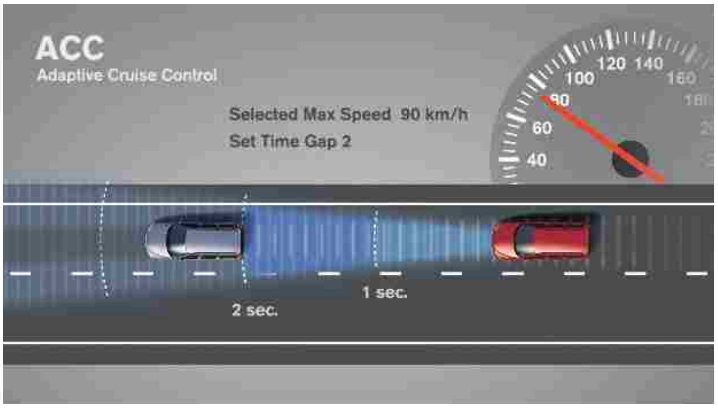


Figure 3.3: *Adaptive Cruise Control system*

When the vehicle in front accelerates again, the car equipped with ACC will follow it, keeping the optimal distance — e.g. Ford (2005). If the deceleration required is more than want the system can provide, warnings will be issued to prompt the driver to act on the brakes. (Future) ACC system performance is enhanced by incorporating information on road curvature (in such a way as to more accurately determine which vehicle is occupying the same lane as the ego-vehicle[1]) and digital maps that can be used to estimate if vehicle speed is too high to negotiate the curve ahead. These so-called *Curve Management* systems

[1] The term “ego-vehicle” indicates the car, truck or motorbike on which the ADAS system is installed and/or whose longitudinal and lateral behaviour we want to control.

require the use of GPS antennas and maps specifically developed for ADAS.

- *Traffic Jam Assist/Stop&Go* systems are a variant of ACC designed to conduct the vehicle in congested traffic conditions by following a reference car ahead.
- *Forward Collision Warning* (FCW) systems are employed to minimize the risk of collision. A set of sensors (laser or microwave radars) are used to compute the relative position and speed with respect to an obstacle. If a risk of crashing is recognized, the driver is alerted.
- *Electronic Brake Assist* (EBA) devices are commonly found on ABS-equipped vehicles. These systems recognize an emergency braking action and, if the driver has not applied a sufficient force on the pedal, they can support driver action in order to reach maximum braking power. *Forward Crash Mitigation* (FCM) systems expand the functionalities of FCW and EBA by, in case of high crash probability, warning the driver and, if the latter does not respond, applying the maximum brake force and simultaneously pretensioning the seat-belts.
- *Traffic Sign Recognition* supports the driver with information about the current speed limit. System performance is enhanced by integrating data from the GPS to avoid un-realistic limits that could result from faded or partially-covered signals.

- **Reversing/Parking Aids**

  These devices perform obstacle detection at low speeds and notify the driver when the vehicle is becoming too close to an obstacle. A signal of increasing frequency and/or volume is employed to warn the driver of the distance reduction from an obstacle. Some commercial variants of the system can provide visual indications to the driver on how to maneuver the vehicle and can also detect the presence of curbs or pedestrians. Some manufacturers have developed systems that can detect a suitable potential parking space and automatically perform the necessary steering actions to complete a parallel parking while the driver controls accelerator and brake (figure 3.4 [39]).

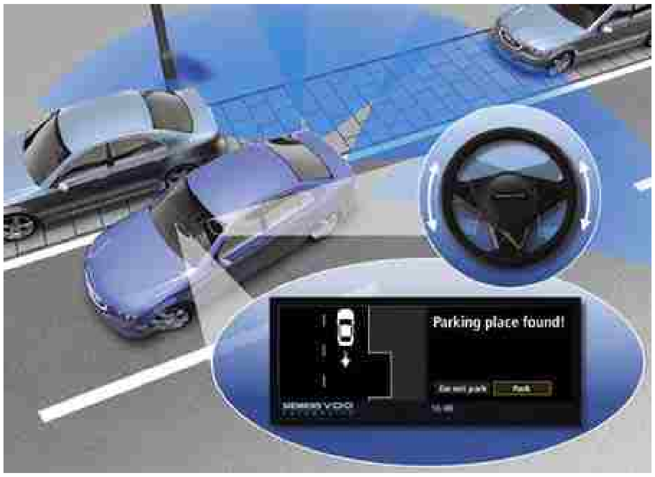


Figure 3.4: *Siemens VDO's Park Mate*

- **Vision Enhancement Systems**

  In order to support the driver task in driving environments with reduced visibility, different technological aids have been developed:

  - *Night Vision* systems employ cameras and near-infrared lights to illuminate the area in front of the vehicle. Another possible approach is to use infrared sensors which create a thermal map of the vehicle surroundings that allows the driver to "see" the external world, even in case of extreme darkness. These devices allow a considerable improvement of safety at night and in bad-weather conditions as well as a significant reduction of stress.
  - *Smart Headlamps* grant useful functionalities to increase driving safety of the car on which they are installed, as well as of oncoming traffic: headlamps that can follow the profile of a curve, automatically dim when there is the risk of dazzling other drivers, or adjust their height as a function of the load and speed of the vehicle.

- **Driver Monitoring Systems**

  These devices are employed to study the driver's physiological condition to discover potential abnormal statuses. Driver impairment could be the result of fatigue, alcohol or drug abuse, distraction or sudden illness. In order to detect potential problems

Driver Vigilance Monitoring systems examine driving patterns (e.g. erratic lateral position changes) or eyelids movements.

- **Pre-Crash Systems**

  These systems are intended to detect *unavoidable* accidents and take suitable countermeasures. Some of the most relevant applications are the following:

  - *Smart Restraint Systems* determine imminent crashes, warn the occupants and activate the airbags best positioned to achieve maximum protection. Critical conditions are determined on the basis of different sensors and systems like those pertaining to ABS and ESC. System operation is made less invasive by making interventions reversible if the crash does not occur.
  - *Rear End Collision Avoidance* and *Rear-Collision Warning* systems are intended to sense the possibility of an accident with a vehicle on the same driving lane and to warn the driver if the time to collision is below a pre-programmed limit. If the driver takes no action then the system can activate the brakes or trigger the pretensioner of the seat-belts. Certain systems — e.g. Ford Advanced Telematics (2005) — can alert incoming traffic with the rear-mounted lights.

- **Road Surface/Low Friction Warning Systems**

  Systems triggered by road surface conditions fall into this category. The information about road friction is gathered by vehicle (and, in the future, road-mounted) sensors and used by the relevant active safety systems. Studies are conducted to assess the possibility to transmit this information to other vehicles (V2V — *Vehicle to Vehicle*) or to the infrastructure administrators (V2I — *Vehicle to Infrastructure*).

  Active safety features that makes use of road information are ABS (Anti-Lock Braking System), EBD (Electronic Brake Force Distribution), TCS (Traction Control System), VSC (Vehicle Stability Control), DAC (Downhill Assist Control System), HAC (Hill-Start Assist Control) and others.

What was briefly summarized here are the currently available ADAS systems. In the *near-future* it is to be expected a strong development in the field. Let us then review some of the solutions that will reach the market in the next few years.

The ACC and LDW functions discussed previously will most likely be more and more combined into a unique "autonomous" driving capability in the future. Several companies, e.g. Honda [16], Mercedes [9], Google [8], Delphi [11], *etc.*, are working in this direction and a glimpse of the future can be seen in the highway autonomous drive function currently offered by Tesla Motors on its production range, nicknamed "Autopilot". This feature, cited as one of the ten 2016 Breakthrough Technologies by the MIT Technology Review [40], "allows Model S [and Model X] to steer within a lane, change lanes with the simple tap of a turn signal, and manage speed by using active, traffic-aware cruise control. Digital control of motors, brakes, and steering helps avoid collisions from the front and sides, as well as preventing the car from wandering off the road. Model S can also scan for a parking space, alert you when one is available, and parallel park on command" [41].

Of course a requisite for the implementation of more and more effective and complete driving automatic control is the substitution of the traditional, mechanical links of steering and pedals, with *by-wire* controls. This has already been done extensively in the past for the accelerator pedal (i.e. Electronic Throttle Control, a system widely diffused since the early 2000s [42]), the diffusion of by-wire brake and steering control has been slower. In particular, besides a few exceptions (the first production vehicle to implement this was the Infiniti Q50), by-wire steering is still a system yet to be fully released to the market. The combination of LDW, ACC and by-wire controls is a key element of future highway systems that strive for safe and efficient travel.

Other important developments are the integration of vehicle speed control with traffic control systems (a topic that will be discussed better in the section dedicated to V2I). This synthesis between vehicle data and traffic data (read from traffic signs and lights and/or received from regional traffic centers) could help drivers through adverse traffic and weather conditions and facilitate navigation (e.g. reduce congestion and limit speeding).

Further development of ADAS expected in the near future comprises *Intersection Collision Avoidance*, a system able to warn the driver of a possible risk for collision at an intersection, and *Pedestrian or Obstacle Detection*, a system aimed at detecting and avoiding possible incidents with the most vulnerable road users: cyclists and pedestrians. The detection of these obstacles is made more difficult by their reduced dimension, which calls for very robust detection systems.

The implementation of this and other sensing technologies is aimed at achieving a *Perception of Vehicle Surroundings* that allows the ego-vehicle to create an instantaneous map of the surrounding area — Headley (2005). This strong understanding of the driving situation is the key to enhance the vehicle's ability to perform intelligent and effective decisions to provide safe and efficient driving. The "situation awereness" thus obtained from the on-board sensors can be further enhanced by integrating information from Digital Maps and Satellite Positioning. Maps purposefully developed for ADAS could empower active safety systems by helping reduce false positives in the computation of road curvature, by orienting smart headlamps before the vehicle has even entered a curve, by enhancing radar vehicle detection that could mistakenly interpret the disappearance of the vehicle in front behind the crest of a hill as the sign that the road is clear and that it is possible to accelerate, *etc.*

Finally, in the near future the increase of the CPU's computational power, of sensor precision and of the amount of gathered data is expected to significantly enhance the current implementations of ADAS systems, by making them not only able to perform more and more functions, but also by blending them more and more un-noticeably in the habits of millions of drivers worldwide.

3.1.3 Levels of Automations

It was mentioned in the introduction to this Thesis that the revolution of Autonomous Drive (AD) will not happen overnight. Many steps will be involved, and the journey has just started. It is going to be a "revolution through evolution" as some journalists have termed it. It appears, then, important to introduce a distinction between the different terms that have been introduced to describe the world of automated vehicles.

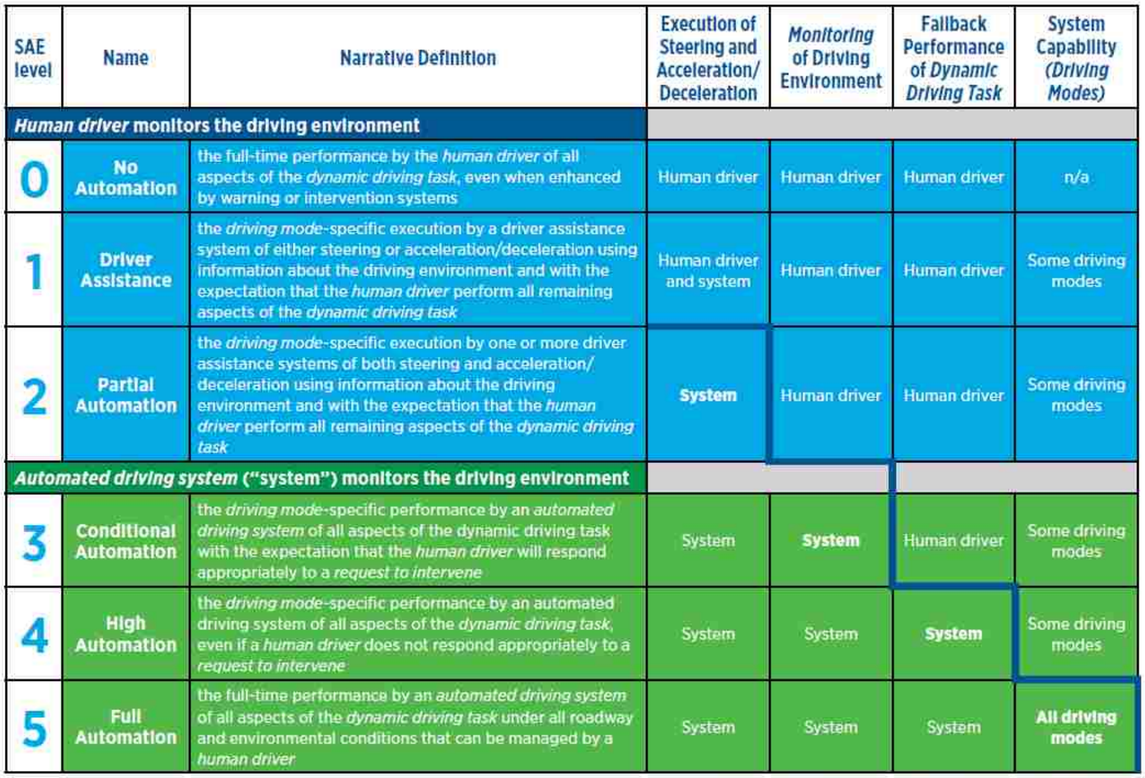
This need has been understood by the key institutions of the field and several classifications have been proposed. In particular, in January 2014, the SAE - Society of Automotive Engineers - has introduced a system based on *six degrees* of automated driving to try to shed light on the confusion that was previously reigning. This standard, which has now become one of the most widely used classification systems, identify the share of responsibility in performing the various aspect of the so-called "dynamic driving task"[2] that must be fulfilled by the driver and the system itself. The use of this classification allows the different actors involved in making autonomous driving a reality - from the R&D Engineer to the Legislator and the public - to be on the same page. SAE, it must be mentioned, is not the only institution who has devoted effort in classifying the various faces of AD: the US National Highway Traffic Safety Administration (NHTSA), the German Association of the Automotive Industry (VDA) and the German Federal Highway Research Institute (BASt) have all generated taxonomies. Nonetheless, since these classification systems typically correspond to a high degree, only the SAE one will be here reported and used as a reference in the following chapters of this Thesis.

SAE Standard J3016 [4] reports six levels of driving automation, ranging from *no automation* to *full automation*. The six levels are described in figure 3.5 and elements refer to the minimum system capabilities for each level. The term "system" is here employed to define the driver assistance system, combination of driver assistance systems or *automated driving systems*. Warning and momentary intervention systems, as SAE specifies, are excluded because they do *not* automate any part of the driving dynamic task on a sustained basis and therefore do not change the human driver's role in performing the dynamic driving task.

Making reference to the summarizing table in figure 3.5, it can be seen that the distinction between a level and the next one is offered by the four elements reported on the top right, namely:

- Who (among the driver and the system) is in charge of controlling lateral (steering) and longitudinal (acceleration/deceleration) vehicle dynamics;
- Who (among the driver and the system) is in charge of monitoring the driving environment;
- Who (among the driver and the system) is in charge of the fallback performance of

[2] SAE International's standard J3016 defines the *dynamic driving task* as including "the operational (steering, braking, accelerating, monitoring the vehicle and roadway) and tactical (responding to events, determining when to change lanes, turn, use signals, etc.) aspects of the driving task, but not the strategic (determining destinations and waypoints) aspect of the driving task."



SAE level	Name	Narrative Definition	Execution of Steering and Acceleration/ Deceleration	*Monitoring* of Driving Environment	Fallback Performance of *Dynamic Driving Task*	System Capability *(Driving Modes)*
***Human driver* monitors the driving environment**						
0	No Automation	the full-time performance by the *human driver* of all aspects of the *dynamic driving task*, even when enhanced by warning or intervention systems	Human driver	Human driver	Human driver	n/a
1	Driver Assistance	the *driving mode*-specific execution by a driver assistance system of either steering or acceleration/deceleration using information about the driving environment and with the expectation that the *human driver* perform all remaining aspects of the *dynamic driving task*	Human driver and system	Human driver	Human driver	Some driving modes
2	Partial Automation	the *driving mode*-specific execution by one or more driver assistance systems of both steering and acceleration/ deceleration using information about the driving environment and with the expectation that the *human driver* perform all remaining aspects of the *dynamic driving task*	**System**	Human driver	Human driver	Some driving modes
***Automated driving system* ("system") monitors the driving environment**						
3	Conditional Automation	the *driving mode*-specific performance by an *automated driving system* of all aspects of the dynamic driving task with the expectation that the *human driver* will respond appropriately to a *request to intervene*	System	**System**	Human driver	Some driving modes
4	High Automation	the *driving mode*-specific performance by an automated driving system of all aspects of the *dynamic driving task*, even if a *human driver* does not respond appropriately to a *request to intervene*	System	System	**System**	Some driving modes
5	Full Automation	the full-time performance by an *automated driving system* of all aspects of the *dynamic driving task* under all roadway and environmental conditions that can be managed by a *human driver*	System	System	System	**All driving modes**

Copyright © 2014 SAE International. The summary table may be freely copied and distributed provided SAE International and J3016 are acknowledged as the source and must be reproduced AS-IS.

Figure 3.5: *SAE International's standard J3016*

the dynamic driving task;

- What is the system capability in terms of driving modes, i.e. of different driving scenarios with characteristic dynamic driving task requirements (e.g. high-speed cruising *vs.* low-speed traffic jam).

At *level zero*, all aspects of the dynamic driving task are managed by the driver. Warning or intervention systems could help the driver in specific situations, but - as mentioned - they do not automate the driving task.

At *level one*[3], part of the driving task (either steering or acceleration/deceleration) is automated by a driver assistance system, while all the remaining aspects are carried out by the driver.

[3] The five levels of driving automation are sometimes referred to as: "hands on", "hands off", "eyes off", "mind off" and "driverless".

At *level two*, both steering and acceleration/deceleration are performed by the system. “Partial Automation” is then achieved, since the driver is still expected to monitor the driving environment.

At *level three*, a key distinction occur: if up to level two the human driver is still involved in part of the dynamic driving task, at level three the automated driving system performs the *entire* dynamic driving task. We still refer to level three as “Conditional Automation”, though, because the driver is still expected to intervene when notified by the system.

At *level four*, this aspect of human intervention as a fallback is dismissed and a completely autonomous system, as far as some driving scenarios are considered, is achieved.

Finally, at *level five*, “Full Automation” is achieved by an automated driving system that can carry out all the aspects of the dynamic driving task under all possible environmental conditions that can be managed by a human driver. Systems of this kind are sometimes referred to as “self-driving”, “autonomous” or “driverless”. If level four is a somewhat short-term goal (experts believe that by year 2025 there will be fully automated systems able to carry out all aspects of the driving task *in certain scenarios*), level five is probably a more distant future prospect. Vehicles with this level of automation, in fact, do not require a driver *at all*.

3.1.4 Look-down and Look-ahead Approaches

In order to implement driver assistance systems able to help the driver in specific tasks or in automating part or all of the dynamic driving task it is necessary to equip the ego-vehicle of sensing technology able to detect suitable references on the roadway.

While a more precise description of the sensors specifically used in this project will be offered in the following sections, it is worth mentioning here how these technologies can be divided on the basis of their working principle and operation.

In particular, two (not mutually exclusive) approaches can be followed in developing a (partially) automated driving system. References can, in fact, be detected *in front* of the vehicle (so-called “look-ahead” sensing) or *beneath* the vehicle (so-called “look-down” sensing) [43]. Clearly, each sensing scheme has its advantages and disadvantages.

Look-ahead sensing can be obtained using vision sensors. As will be discussed in the following, vision systems can be either realized with monocular cameras or stereo-cameras. Other kinds of look-ahead sensors include, but are not limited to, RADAR (RAdio Detection And Ranging) and LIDAR (LIght Detection And Ranging) which can be used to detect physical objects in front of the vehicle. The electromagnetic wave generated by the device is reflected from obstacles in its path and it is received and processed by an unit that allows to determine properties of the objects. The characteristics of this sensors — the fact that they "look" *ahead* of the vehicle — provides systems which make use of them with the advantage of a more stable close-loop control [43]. The literature offers several examples of AD applications that rely on the use of these kinds of sensors. Figure 3.6 shows an application of a camera sensor described in [19]. The output of the front-facing camera is suitably pre-processed and analyzed in such a way as to make possible the extraction of features from the image that can then be modeled as lane markings and used for LDW and lateral control. The drawback of this approach is that image processing is expensive, both in terms of equipment needed and in terms of computational power demanded to the analyzing softwares. Moreover, difficult weather and/or road conditions can make the processing of optical images complex and poor-performing.

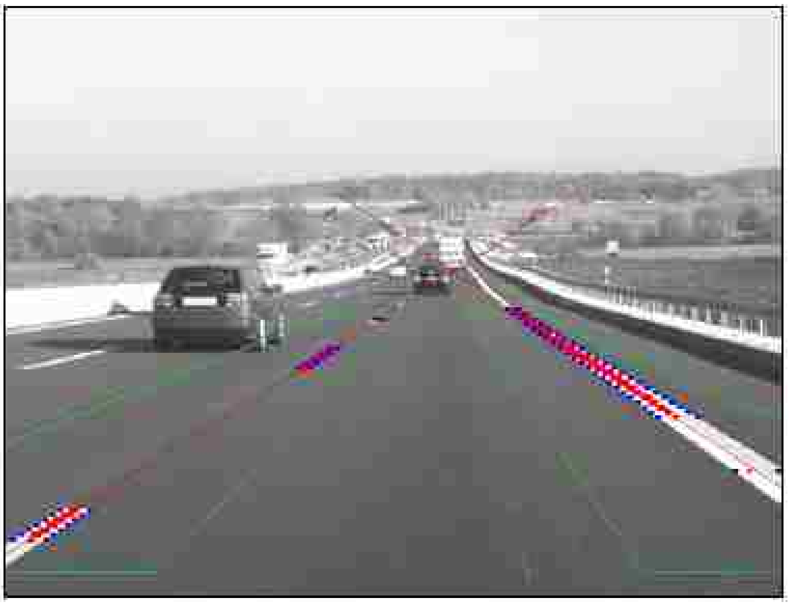


Figure 3.6: *Lane Recognition through a camera sensor*

Look-down sensors, on the other hand, are realized in such a way as to interact with the roadway beneath the vehicle. These devices can be classified as continuous or discontinuous and are usually based on electrified wires or buried permanent magnets. Malan *et al.* [45] show an example of look-down control in which the vehicle lateral displacement is obtained through an antenna (mounted on the front part of the vehicle) able to interact with transponders "drowned" into the road (see image 3.7). The use of of either electrified wires or permanent magnets can be carried out with relatively inexpensive sensors and minimum data processing. Moreover, this kind of technology is independent from weather and road conditions. Nonetheless, these schemes give no indications about the upstream road conditions (this problem was non-existent in [45] because the route to be followed was known *a priori*) causing the control system to be less stable and performance to be affected, and they can require an expensive infrastructure to be built on the road.

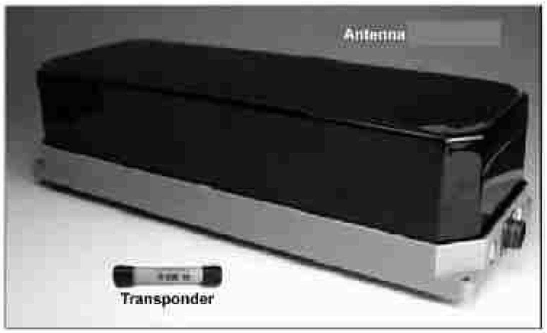


Figure 3.7: *Look-down sensing apparatus composed of an antenna and transponders*

An intelligent combination of the two approaches can be exploited to take advantage of the best characteristics of each sensing technology ([43]). For what concern this project a look-ahead structure based on a camera sensor is the basic layout onto which the concept of a virtual sensor is applied.

3.1.5 Data Fusion

The problem of data-fusion is extensively taken on in the academic [46] and industrial [20] literature, often making reference to automotive applications [47]. Researchers in this field agree that the most accepted definition of data fusion was provided by the Joint Directors of Laboratories (JDL) workshop [49]:

Definition. *Data Fusion* A multi-level process dealing with the association, correlation, combination of data and information from single and multiple sources to achieve refined position, identify estimates and complete and timely assessments of situations, threats and their significance.

The combination of multiple sources that ensues from the use of data fusion allows to obtain improved information; in this context, improved information means less expensive, higher quality, or more relevant information [48].

In order to achieve this, data coming from different sensors is mixed accordingly to the relationship between the different streams of data. Now, this relationship is not always the same; it is in fact possible to distinguish three types of relation between the sources [48]:

- *Complementary*: when the information coming from the input sources is representative of different parts of the global framework. In this case data fusion can be used to achieve a more comprehensive assessment of the situation. An example of complimentary information is given by the video streams produced by two cameras pointed on the same target but with different fields of view;

- *Redundant*: when the input sources provide information about the same situation. In theory, no *new* information is gathered. This circumstance is analogous to the repetition of the measurement of a certain quantity varying only the measurement device. The objective of exploiting redundant information is to increment the confidence we can attribute to the understanding of the situation and to provide a fall-back in case one of the input sources is impaired;

- *Cooperative*: when the input information is fused into new information, more complex than the original. The combination of audio and video could be considered cooperative.

In our case study data fusion will be the fundamental technique exploited to obtain an enhanced understanding of the driving scenario, compensating for the lack of one of the sensors (the LRC).

Data fusion can be classified according to different criteria (the aforementioned relationship between input sources is one of them) [20]. The Joint Directors of Laboratories (JDL) Data Fusion Model distinguishes *low level* processing and *high level* processing of data:

- Low Level
  - Track Estimation, i.e. the evaluation of states (e.g. position, velocity,...) used for control loops;
  - Object Discrimination
    - Detection of objects;
    - Classification of objects into predefined classes;
- High Level

  Algorithms for situation assessment and control of the data fusion process (e.g. allocating resources).

Another possible classification of data fusion can be drawn considering the kind of information that is processed [20]:

- *Raw data* represent minimally preprocessed data (e.g. pixels of an image). When raw data fusion is performed the whole information provided by the single sensors is used. On the other hand large amount of data have to be transmitted and processed and, because of the particular nature of the handled data, fusion algorithms will be rather inflexible;
- *Feature level* data are data in which some features of interest have been extracted from raw data. The fusion of this kind of data has the advantage of involving smaller information quantities. Furthermore, fusion algorithms will be more modular and easily extended. The drawback is that the full information coming from the sensors is no more available;
- *Decision level* approach uses decisions made in earlier steps and fuses them to achieve a more exhaustive verdict. This kind of approach is flexible and well suited to situation assessments.

An example of feature-level data fusion applied to ADAS is offered by N. Kaempchen and K.C.J. Dietmayer who developed a fusion algorithm to associate information coming from a laser-scanner and a videocamera installed on a vehicle [19]. Figure 3.8 shows the principle of operation of this data fusion application: thanks to the combination of the two information sources it becomes possible to assign to specific regions of interest (ROI) the corresponding distance from the ego-vehicle. The Laserscanner is exploited to track and classify vehicles (included motorbikes and pedestrians) and the camera is used to estimate the position of the ego-vehicle in the lane, as well as to improve the measurement of the lateral position and of the velocity of the objects detected.

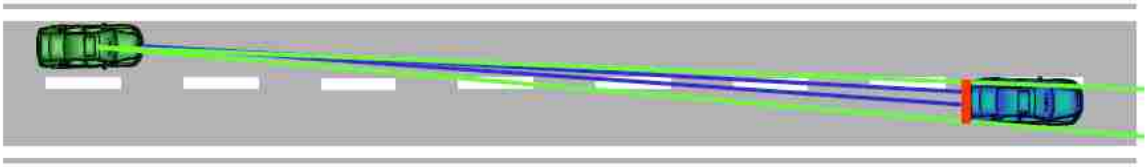


Figure 3.8: *Fusion of Laserscanner (blue) and Videocamera (green)*

The two sensors are used in a complimentary way: in the area immediately in front of the vehicle objects are detected making reference to the very accurate Laserscanner; in the far field the camera attention is controlled by the Laserscanner and it is used to refine the lateral offset estimation. The result is a system capable of identifying with precision objects in front of the vehicle and correlate them with their position in the lanes[4]. This allows the algorithm to accurately predict when a preceding vehicle is about to change lane. The paper is then an interesting example of how data fusion can be exploited in an ADAS application to improve situation understanding.

3.1.6 V2X

Let us conclude the introduction to the general context of ADAS with one of the most relevant and promising developments that is about to enter the automotive world in the near future. As of now, when discussing of communication technology applied to vehicles, it is principally intended *intra-vehicle communication*, i.e. communication within the vehicle

[4] The Laserscanner can be employed only to detect physical objects. As a result, painted lane markings can be only identified if the attention of the camera is dedicated to the region around the detected vehicle.

boundaries between the various sensors, actuators and CPUs (see [50] for an extended review of intra-vehicle networks). While this is a fascinating field exhibiting extremely complex challenges due to ever increasing bandwidth and safety requirements, the area in which the most innovation in terms of communication technology in vehicles is expected is represented by *inter-vehicle communications.* Experts often make reference to this kind of communication as *V2X* - which stands for Vehicle-to-X - to collect under this term all kind of communication between a vehicle and an entity, be it another vehicle (*V2V*) or an infrastructure (*V2I*).

Efforts made in the direction of vehicular communication fall under the framework of *Intelligent Transportation Systems* (ITS). Siemens [51] defines V2X as:

Definition. *Vehicle-to-X* An intelligent transport system where all vehicles and infrastructure systems are interconnected with each other. This connectivity will provide more precise knowledge of the traffic situation across the entire road network which in turn will help:

- Optimize traffic flows;
- Cut accident numbers;
- Reduce congestion;
- Minimize emissions.

This definition captures well the idea behind V2X and it is intended to be applied to all kinds of vehicles, not only cars. The primary objective of this emerging technology is to reduce the number of road deaths.

One of the most important ways in which V2X can accomplish this is by warning drivers of hidden dangers that would be otherwise impossible to discern with traditional sensing technologies. Other crucial advantages reside in the possibility of managing traffic in order to avoid congestions and hence reduce the impact on environment and shorten driving times.

Table 3.1 shows some of the uses for the connected-car technology [52]. The first two columns shortly summarize the advantages to vehicle safety of both declinations of V2X. The third one lists some of the expected advantages that this technology can bring to drivers.

It is interesting to notice that the benefits of V2X are not exclusive to autonomous driving applications only, but can be successfully implemented in standard vehicles, too. The devices that make this possible work using DSRC (Dedicated Short-Range Communications), two-way, short-range wireless communications [53] between vehicles and infrastructures (e.g. traffic lights).

Table 3.1: *Advantages of V2X in terms of safety and convenience to drivers*

V2I Safety	**V2V Safety**	**Convenience**
Red light violation warning	Emergency brake light warning	Eco driving
Curve speed warning	Forward collision warning	Smart cities
Spot weather	Red light violation	Parking information
Work zone safety	Slow traffic ahead	Truck platooning
Bridge height	Aggressive driver warning	Speed harmonization
Pedestrian in crossing signal	Emergency vehicle notification	Queue warning
Stop sign gap assist	Road hazard detection	Insurance pricing

For what concerns automated driving, V2X could prove itself very useful in increasing the amount of data received by a vehicle regarding its surroundings (for instance every pedestrian who carries a smartphone could broadcast its position through low-energy Bluetooth [54]). This allows vehicles to be aware of hidden, out-of-sight obstacles, like cars stopped on the road or trucks approaching around blind corners. Infrastructures could be set up to transmit their position and hence act as stations for DGPS (see section on GPS). *Inter*-vehicles communication could make complex maneuvers (e.g. overtaking) simpler by establishing a “conversation” between the involved vehicles. As a side effect, better managed vehicular flows would also mean lower economical and environmental losses due to congestions.

As of now, obstacles to the diffusion of V2X are represented by safety and privacy concerns. It is in fact paramount, as the network of the vehicle becomes more and more “open”, to suitably protect information from noise-corruption and illegal activities. In this way trust among the consumers can be build and the technology can become accepted and successful.

3.2 Model Components

In this second part of the literature review, a more detailed look at the components of the model of interest shown in figure 1.3 is given.

3.2.1 Electric Power Steering

One of the actuators which were not embedded in the vehicle dynamics block discussed in chapter 2 is the Electric Power Steering (EPS). Electric power assisted steering (EPS/EPAS) systems use the torque generated by an electric motor to help the driver maneuvering the steering wheel (figure 3.9).

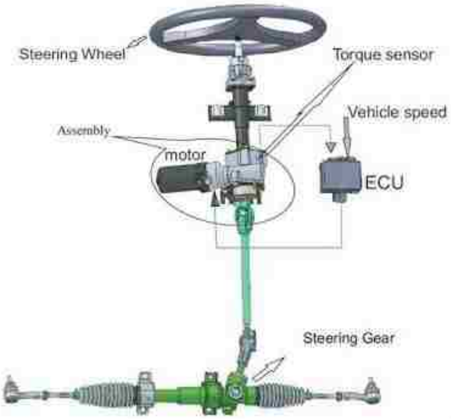


Figure 3.9: *Electric Power Steering actuator*

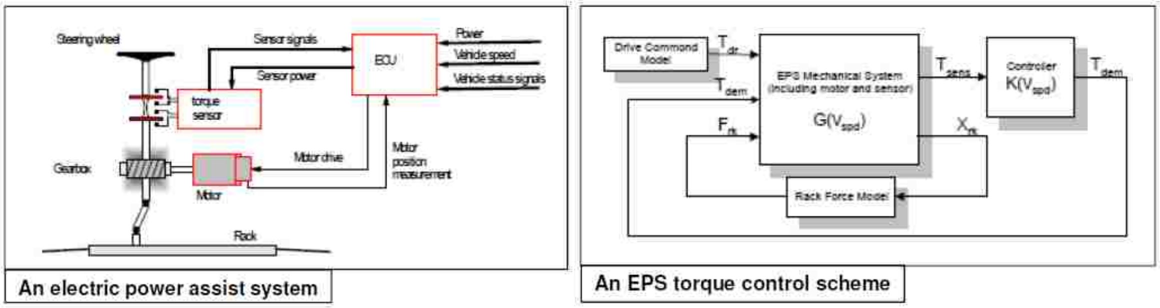
Such systems are rapidly superseding traditional hydraulically-assisted systems because of their greater fuel efficiency (the system operates only when necessary, compared to the oil pump of an hydraulic system which constantly draws power from the engine), simpler manufacturing and maintenance and possibility to embed various assistance systems (e.g.

self-parking).

The operation of the system is as follows: the EPS control system first considers the torque requested by the driver and then accounts for the vehicle speed (with the so-called *boost curves*) in order to modify such requests. On the basis of these inputs, the EPS motor is activated in such a way as to add a torque sufficient to overcome the resistance of the steering rack (friction, tires, *etc.*). The system is highly nonlinear in nature.

More in detail, as described in the paper [59], the system of interest can be represented as in Fig. 3.10. The torque applied by the driver T_{dr} to the steering wheel is provided to the control system's ECU by a sensor based on the measurement of the torsion angle of a deformable bar. This torque is then increased of the quantity T_{dem} necessary to compensate the resistance F_{rk} coming from the steering rack in such a way as to facilitate the movement of the steering wheel.

The control law of the electric power steering can be made dependent on a number of variables in order to tailor system response on driving conditions. Some Fiat vehicles, for instance, offer a second, driver-selectable assist map that is suited to city driving.



(a) EPS layout

(b) EPS control

Figure 3.10: *EPS layout and control loop*

A typical EPS control module is designed to perform the following functions:

- Assistance torque generation, employed to support driver operation of the steering wheel;
- System stabilization, derived from the frequency response of the system;

- So-called *inertia compensation*, used to reproduce the "feeling" of a more traditional hydraulically-assisted system;
- Self-centering, adopted to avoid static residual steering angle after release.

Example of EPS Control

In [60] an example of a two-layer control (figure 3.11) for the unmanned command of an electrically-assisted steering system is proposed. This interesting application uses an high-level fuzzy logic controller to compute the target position of the steering wheel. The fuzzy controller is chosen because it mimics human behaviour. This characteristic allowed system designers to avoid the necessity to determine mathematical models which are either rather complex or not suited to manage the strong non-linearities of the system. At the lower level a traditional PID controller regulates the input of the electric motor to track the reference steering position. This layout, combined with a real-time kinematic differential GPS (RTK-DGPS) providing an exact measurement of the ego-vehicle's position, shows excellent results.

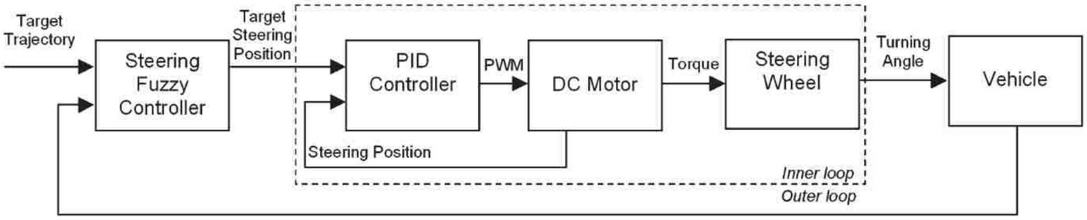


Figure 3.11: *Two-layer steering wheel controller [60]*

3.2.2 Lane Recognition Camera

As previously discussed, look-ahead control approaches require the vehicle to be somehow able to detect features of the road in front. One of the most important functions that must be performed is *lane detection and estimation*. The knowledge of the lane boundaries is crucial for controlling vehicle's lateral position in many ADAS applications: lane departure warning (LDW), lane keeping assistance (LKA), lane change assistance (LCA) and vehicle self-positioning [61] in the lane.

Lane Detection (LD) is typically performed by using cameras as the main source of data [15]. The reason why cameras have such an important role in driving automation is no different from the reason why eyes are so important to all animal species: no other sensor can match the resolution, the details and the vividness of a scene that vision can capture.

Two main types of videocameras are employed in ADAS applications [62]. A *monocular* (i.e. "single-eyed") system employs a single camera sensor to capture the video to be processed. A *stereo*-vision system exploits two cameras, each separated from the other (figure 3.12). Some remarks on the differences between the two systems are given in table 3.2. The most relevant advantage offered my stereo systems is represented by their superior understanding of the 3D features of the world.

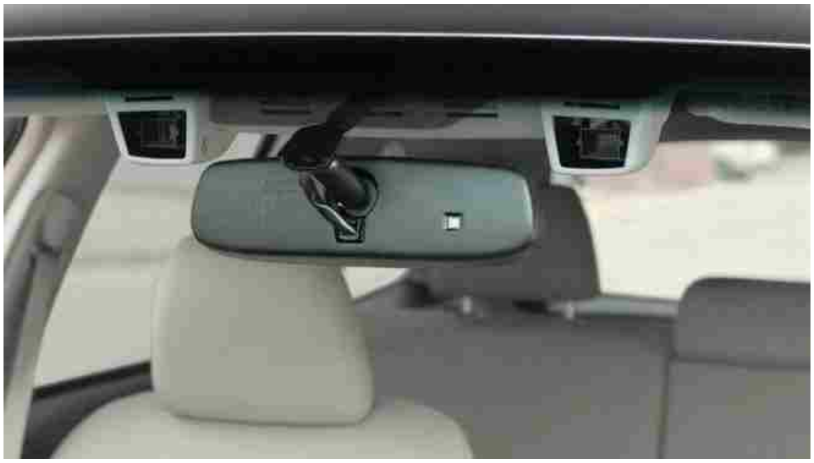


Figure 3.12: *Subaru's "Eyesight" system (Lisa Calvi photo)*

Lane Detection Algorithms

Algorithms and softwares used in image processing are often complex and sophisticated. In the following a brief overview of the main phases implemented in lane detection and road understanding is given. In particular three modules are often recognized as the most important ones in the literature [15] [62]: image pre-processing, feature extraction and model-fitting.

Image pre-processing deals with the enhancement of the frames by making them better suited for the next phases. Noise is reduced and pixel-wise operations are performed to

Table 3.2: *High-level comparison between mono and stereo camera ADAS systems [62]*

	Mono-camera system	**Stereo-camera system**
Number of image sensors, lenses and assemblies	1	2
Physical size of the system	Typically small	Two small assemblies separated by $25 - 30\,cm$ distance
Frame rate	30 to 60 fps	30 fps
Image processing requirements	Medium	High
Reliability of detection and emergency situation assessment	Medium	High
System reliable for	Object detection	Object detection *and* distance-to-object calculation
System cost	1x	1.5x
Software and algorithm complexity	High	Medium

improve image contrast and brightness. The video is often converted to gray-scale, too. Illumination variations, shadows and imagining artifacts make this phase extremely important and difficult.

Once the frame has been suitably pre-processed the image is searched for the features of interest. Considering lane markings, one of the most relevant feature extraction methods is the *Hough Transform* (HT), that employs a voting procedure in order to obtain likely candidates. This method is suited mainly for detecting straight lines. To reach an higher level of performance, the HT can be applied to an *Inverse Perspective Mapping* (IPM) procedure which removes the perspective effect (figure 3.13) from the image, mapping it to a 2D domain. A more conventional approach for LD exploits a combination of gradient magnitude in the edge image and Kalman filters to achieve robust performance. In this case lane markings are detected based on the variations in gradient magnitude in the edge image. Figure 3.14 shows the main idea behind this method: a brightness gradient is expected near every point along the markings of the lanes. The larger the magnitude of the gradient, the higher the probability of detecting a lane marking.

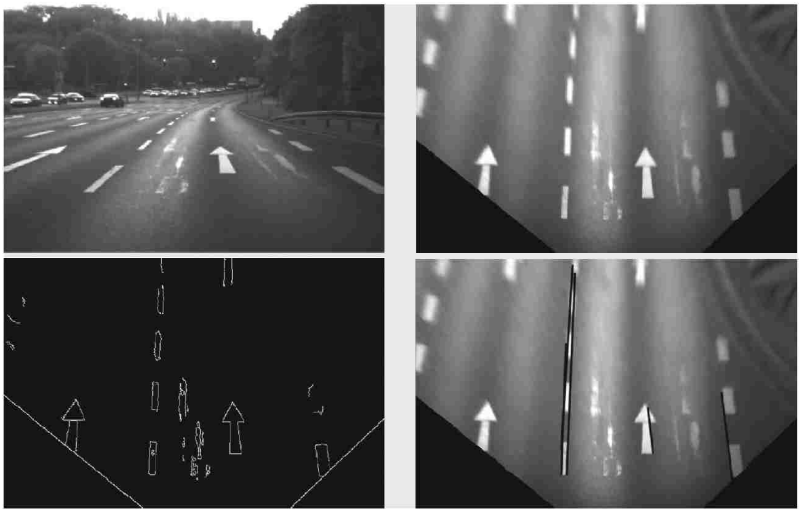


Figure 3.13: *IPM-Hough method for feature extraction. Top left: original image; top right: IPM image; bottom left: Canny edge image; bottom right: Hough lines [15]*

Detected features are then compared to set of models in order to classify them properly. Model-fitting of lane signs is usually performed using straight line models, hyperbolas, least square method, B-snake, B-spline, clothoids[5], and RANSAC (RANdom SAmple Consensus) model fitting. Once lane delimiters have been recognized their tracking is done using Kalman filters that predict the future position of the vehicle in the lane. Machine learning models such as *Support Vector Machine* (SVM) and *Convolutional Neural Networks* (CNNs) can be used as classifiers to label lane markings in real time.

If the most common, currently-available LD methods are proven to work well in simple

[5] Clothoids, or Euler spirals, are curve whose curvature changes linearly with their length. They are widely employed in railroad engineering/highway engineering for modeling transitions between rail/road portions.

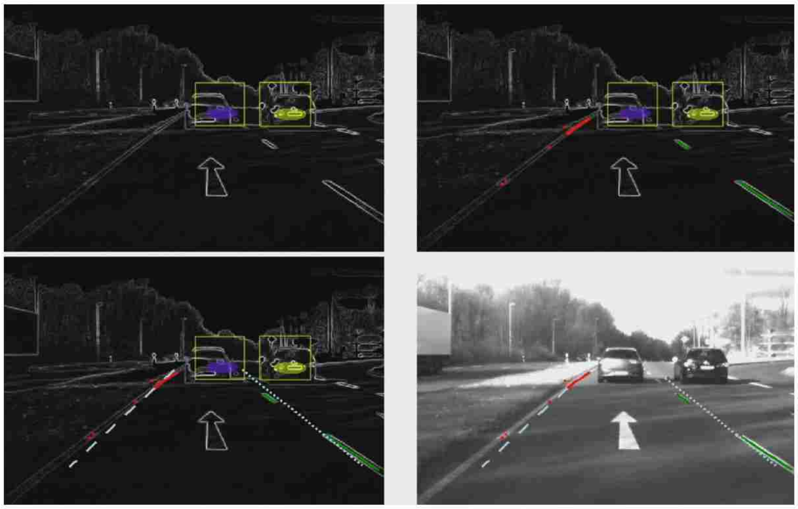


Figure 3.14: *Lane detection through maximum gradient magnitude search. Top left: edge image; top right: looking for gradient magnitude; bottom left: model fitting; bottom right: LD results (detected vehicle regions are not accounted) [15]*

highway environments, problems could arise in more complex situations. Urban scenarios, in particular, remain a challenge. Complex road shapes and faded lane markings severely affect detection performance. Image clarity problems (overexposed frames, shadows or rain/snow on the road) as well as poor visibility (fog, heavy rain or sun reflections) are some of the most demanding issues to LD algorithms. Finally, some systems (the simpler ones) are highly dependent on the *speed* the vehicle is traveling at: the lower the speed the more difficult the correct identification of the road boundaries becomes. This is one of the reason why the algorithm developed here is mainly targeted at speeds higher than 60 km/h.

Examples of LRC Applications

In this last part of the section dedicated to LRC applications to ADAS some examples of possible utilizations are discussed.

M. Aly [63] developed a LD algorithm for urban streets. This sophisticated approach exploits an IPM transform to map the tri-dimensional image generated by the camera to a bi-dimensional top view of the road. The resulting frame is filtered using selective oriented Gaussian filters and fitted using a new and fast RANSAC algorithm. A post-processing step extends and refines the generated splines. Despite some false positives due to stop lines, cross-walks and passing cars are found, the algorithm proves to be an efficient, real-time and robust technique for detecting lanes in urban environments.

Wang, Teoh and Shen developed a LD procedure based on B-Snake lane model [64]. “Snakes” are curves defined within an image domain which can be moved by internal or external forces. Internal “forces” come from the curve itself, whereas external “forces” are determined by image data (figure 3.15). In this application a simplified form was used (B-Spline) to describe lane boundaries. The assumption of parallel road boundaries is used in order to fully recover 3D information from the 2D image. The approach described by the paper proves to be robust against noise and illumination variations. A very interesting outcome is the fact that it can deal with both marked and unmarked roads, with dashed or solid lines.

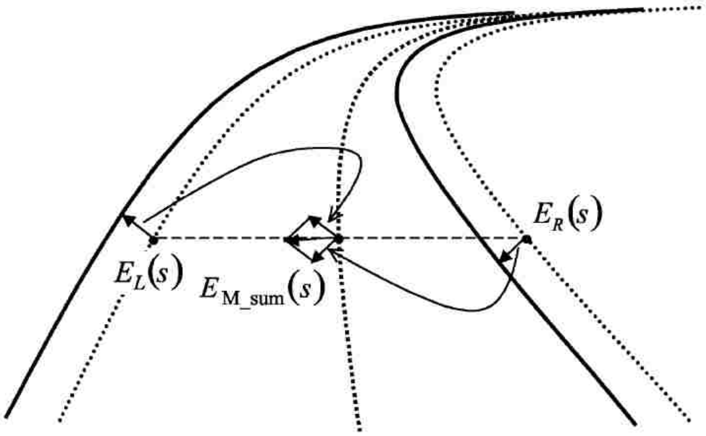


Figure 3.15: *Example of external “forces” causing the lane model (dashed lines) to move towards the real road edges (solid lines) [64]*

Many other interesting applications can be found in the literature (e.g. [65, 66, 67, 68]). Particularly interesting is [69] which defines a vision-based lateral localization algorithm. The system can handle missing lane-markings by storing a matrix of previously detected lane-markings. Finally, in [70] a LOIS (Likelihood Of Image Shape) algorithm is used to develop a LDW system that warns the driver when the measured vehicle offset from the right or left lane markings is falling below a certain threshold too rapidly.

In conclusion, it appears clear how important vision devices are for look-ahead automated driving algorithms. The challenge is then two-fold: on one side, there is the urge to make cameras as reliable as possible in the largest possible set of scenarios and, on the other side, there is the effort to devise systems able to temporarily cope with the lack of information from the vision system.

3.2.3 Global Positioning System

Normal production vehicles are often fitted with GPS-equipped navigation systems. [71] provides the following definition:

Definition. *Global Positioning System (GPS)* The Global Positioning System (GPS) is a space-based navigation system that provides location and time information in all weather conditions, anywhere on or near Earth where there is an unobstructed line of sight to four or more GPS satellites.

The GPS is, in fact, a network of 24 satellites placed into orbit by the U.S. Department of Defense. The system is nowadays available to civilians and its operation is based on *trilateration* of the position of a point on Earth using four satellites.

Theoretically only three satellites could be used to identify the position of a point on the Earth if we reject absurd results: the distance from one satellite defines a sphere of possible positions, a second satellite will narrow down possible locations to the circle defined by the intersection of two spheres and finally a third satellite further reduces the possibilities to two points, of which one can be rejected because of its absurd distance from the ground or traveling speed (see figure 3.16). Distance from a satellite is measured precisely counting how long a radio signal takes to reach the receiver from that satellite. In order to provide accurate results, precise timing is a crucial aspect for both the transmitter (the satellite) and the receiver. This problem is solved at the transmitter end by using sophisticated (and expensive) atomic clocks. Doing the same at the receiver end is simply impossible and it is here that the fourth satellite signal comes into play: because of the lack of perfect synchronization the fourth measurement will *not* intersect the position given by the first three. The receiver can then look for a single correction factor that it can subtract from

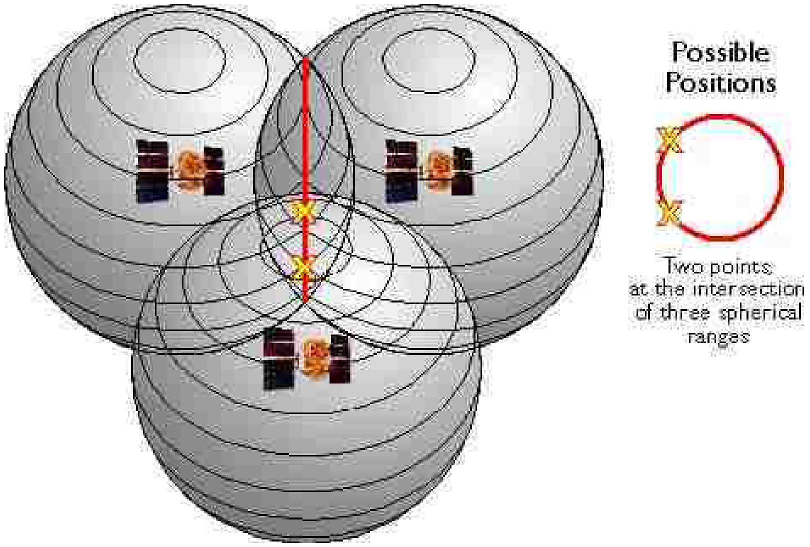


Figure 3.16: *Position estimate through trilateration [72]*

all its timing measurements that would cause them all to intersect at a single point. The remaining issue is that of knowing the exact position of the transmitting satellites in order to finally compute the location of the receiver. This is solved by supplementing the known location on the orbit a satellite should occupy with precise radar measurements.

Despite the level of sophistication of the system, errors could still emerge due to atomic clocks' drift, orbit errors, ionosphere and troposphere disturbances, receiver noise and signal bouncing [73]. As a result a vehicle (or any other civilian GPS-equipped receiver) can only count on a position measurement accurate to around 10 m.

It was already mentioned that in this project, for the purpose of designing the algorithm of the position estimator, far more accurate positioning measurements will be employed. This will be achieved by using a *differential* GPS on the demo vehicle. Such technology can increase ten-fold the accuracy of the system by using the knowledge of the exact position of a fixed receiver. In fact, if such station is sufficiently close to the moving receiver installed on the vehicle, their measurements will be affected by virtually the same errors (excluding those related to the specific receiver hardware and signal bouncing that could occur around the moving receiver). Now, since the absolute position of the fixed station is accurately known *a priori*, its software can calculate how long the traveling time of the signal *should* be. By measuring the difference to the actual signal traveling time, the fixed station can compute an “error correction” factor, which is then broadcast to the moving receivers which are found in the same geographical area. Clearly, using such technology for improving the accuracy of the navigation system installed on commercial vehicles would be a major breakthrough for facilitating and improving autonomous driving systems. Unfortunately the cost and the complexity of such an infrastructure still represent a barrier to the diffusion of differential Global Positioning Systems in vehicles.

As mentioned in the introduction, the GPS receiver will be used to provide the position observer with the following signals:

- Latitude;
- Longitude;
- Elevation;
- Speed;
- Course angle, i.e. the direction of travel;
- HDOP (Horizontal Dilution Of Precision), i.e. position accuracy. A low value is desirable, since it indicates a better positional precision due to a larger angular separation between the satellites used to calculate the receiver position [74].

For easier elaboration of the virtual sensor’s algorithm these quantities will be converted in the X-Y position occupied by the vehicle.

GPS Applications to ADAS

In the literature GPS signals are widely employed to support autonomous driving algorithms. In [75] an automated driving system *without* video-camera is presented. The project shows the feasibility of using a system based on GPS and IMU (Inertial Measurement Unit) only.

3.2.4 Control Systems

In the attempt of improving vehicle dynamic behaviour, automotive companies and research institutes have always looked at control systems as a promising way to enhance vehicle safety

[76]. Well-known examples comprise ABS, traction control and electronic stability program (ESP/ESC). The latter, in particular, is an example of a successful closed-loop automatic control system able to fully govern vehicle lateral dynamics in critical scenarios. Other interesting and more recent examples include active suspensions, active steering and, of course, automated driving.

The growing importance of AVCS (Advanced Vehicle Control Systems) is yet again to be found in the fact that the vast majority of incidents is due to human errors. Trying then to completely take over or at least support the task of the driver is a way to reduce the number of incidents and cut down traffic congestion. An overview of recent developments in automotive controls is offered in [77]. In particular three major areas are investigated:

- *Longitudinal Control*

  The objectives of longitudinal control comprise keeping the safe distance from the vehicle in front, maintaining a constant speed with the least brake use (in order to reduce fuel consumption) and applying the brakes as swiftly as possible in an emergency situation.

  Intelligent (ICC) or adaptive (ACC) cruise controllers expand traditional cruise control functionalities by acting on both accelerator and brake commands. The correct design of these devices must consider a smooth switching between the two in order to guarantee a smooth and fuel efficient ride.

  A promising area of development for longitudinal controls considers the possibility to have groups, or *platoons*, of two or more vehicles traveling on the same lane closely spaced. Vehicle platooning can be realized by devising cruise control systems (AICC - Autonomous Intelligent Cruise Control) able to account for the velocity and acceleration of the vehicle immediately preceding. Even better results can be reached by establishing a V2V communication between the ego-vehicle and the vehicles immediately before and after it as well as the lead vehicle. These CICC (Cooperative Intelligent Cruise Control) systems determine the minimum distance to be kept between vehicles in regards to velocities and braking abilities. In the following, examples of successful applications of the platooning concept will be provided.

- *Lateral Control*

  The tasks of lateral control systems amount to proper vehicle turning and lane keeping, as well as lane changing and obstacle avoidance in extreme conditions. Considering that a third of all U.S. highway fatalities is due to vehicles leaving the road [78], automated steering could significantly reduce road deaths by preventing lane departure and steering overcorrection due to a blown tire.

  As already previously discussed, automated vehicle control can be realized via two different approaches: look-down and look-ahead reference systems.

  Some of the most recent developments in look-ahead technologies perform sophisticated template matching based on features of the roads (lane stripes, signs, tire tracks, oil spots, *etc.*) to determine the vehicle's traveling lane without pavement markings.

- *Integration of Longitudinal and Lateral Controls*

  The development of suitable longitudinal and lateral automated controls represent the first step towards AD. Nonetheless, problems could arise in the case in which the integration between the two is not properly addressed.

  Several scenarios require high collaboration between the two systems for a successful vehicle control. Sharp turns at high speed, for instance, require the vehicle to slow down and apply the right steering angle to lose the least amount of speed. Another critical scenarios is represented by obstacle avoidance, in which the controller needs to determine whether the vehicle should try to stop, go over or swerve around an object in front of it. Finally, if a vehicle looses control the automatic system must wisely evaluate the amount of braking and steering to be applied.

Automated Longitudinal and Lateral Vehicle Controls

The literature offers numerous examples of cruise and lateral controls. [56] presents, for instance, a comprehensive analysis of different controllers which can be used in longitudinal control. In that particular application, PID, PI, sliding mode and fuzzy controllers are all designed to match vehicle dynamics control requirements and then a Fuzzy Supervisory

Expert System is implemented to choose the best option according to the specific dynamic conditions.

The design of traditional Proportional-Integrative-Derivative (PID) and Proportional-Integrative (PI) controllers will be discussed in detail in the next chapter when the longitudinal control used in this project will be discussed. Sliding Mode Control (SMC) is a sophisticated *robust* control approach, i.e. it explicitly deals with the uncertainty in the system to be controlled (generally referred to as the "plant"). This nonlinear control method applies to the plant a discontinuous control signal that forces the plant to "slide" along a cross-section of the system normal behaviour. Fuzzy controllers, on the other hand, rely on fuzzy logic, a branch of mathematical logic which, as opposed to Boolean logic, employs truth variables which can assume any values between 0 and 1. Figure 3.17 shows how different *membership functions* (labeled "cold", "warm" and "hot" in the figure) could be used to assign to the same temperature (vertical line) different levels of truth. So, for instance, the considered temperature could be seen as "not hot", "slightly warm" and "fairly cold". Instead of "crisp" true and false values, fuzzy logic then deals with the concept of "partially true". When this concept is employed for control systems design the biggest advantage is that the solution of the problem can be posed in terms easily understood by human operators, characteristic that makes for an easy, experience-based system tuning.

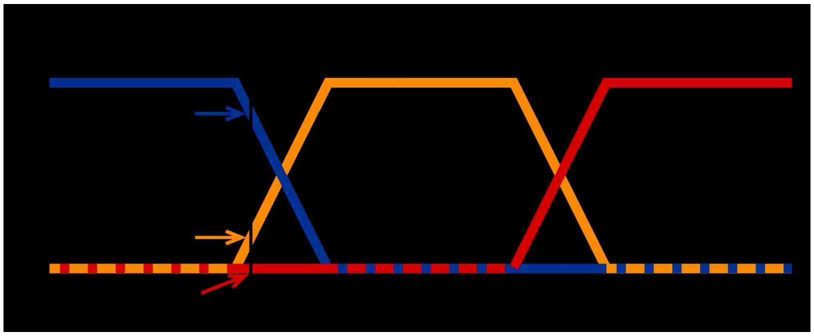


Figure 3.17: *Example of fuzzy logic applied to temperature variables [79]*

Speaking of lateral control approaches, a more complex plant is usually considered. As a result, considering that controllers must be necessarily designed on approximated models of the real plant, it is important to ensure that the control algorithm will perform suitably also in the presence of bounded differences between plant and model. In order to guarantee this, an approach typically used is H-infinity Loop-Shaping, which combines traditional control methods with H_∞ mathematical optimization.

Examples of Lateral Control Design

Naranjo *et al.* [60] and Nishimori *et al.* [80] illustrate two examples of fuzzy control applied to automated driving. Naranjo *et al.* [60], in particular, depicts the combination of two different control algorithms to achieve greater performance. In the first layer a fuzzy controller is implemented to select the adequate position of the steering wheel to negotiate a curve, in the second layer a traditional PID controller moves the steering wheel to track the position targeted by the first layer. This cascade-control paradigm is particularly useful when there is a significant time delay between the variable to be controlled (vehicle turning angle) and the variable upon which the system is acting (steering position). The reason why a fuzzy controller is implemented is that because of its nature it can be more easily tuned to imitate human drivers. Another crucial advantage is that a detailed knowledge of the vehicle dynamics is not needed (much in the way that a driver does not need one). The resulting steering maneuver is very similar to human driving.

The work by Malan, Milanese, Borodani and Gallione [45] was already cited concerning the use of look-down references. Here it is recalled for its interesting control algorithm designed with a combination of feedforward and feedback structures. The feedback action is obtained with three nested loops, of which the outer one is non-linear. The reason why this structure is used is that it provides better performances, it is more robust to plant perturbations and it is better suited for disturbances attenuation. Controllers employed are of the lead, lag or lead-lag types. These so-called “compensators” act on the *root locus* of the closed-loop plant in order to attain desirable characteristics (stability and speed of response for lead compensators and steady-state performance for lag compensators).

3.2.5 Virtual Sensors

In chapter 2 the mathematical justification of virtual sensors (VS) was discussed making reference to the theory developed by Professors M. Milanese and C. Novara from Politecnico di Torino, Italy and by Professors K. Hsu and K. Poolla from University of California at

Berkeley, USA. In this section, some interesting applications of VS to automotive problems are presented.

Applications

In synthesis, the idea behind virtual sensors is that of extracting the information of any physical variable z using only available information y. The values assumed by z may not be available from direct sensors, either because difficult to measure or requiring expensive sensing apparatus. Two scenarios are of interest for virtual sensors applications:

- In *safety critical* control systems it is paramount to properly manage sensor failures. This is usually accomplished through diagnostics and analytical redundancy. This allows the system to both detect and handling errors: if the direct sensors measuring z brake down, it is still possible to achieve a slightly degraded functionality (*limp home*) because the variable can be recovered using available information y;
- The introduction of new competitive functions often urges automakers to look at *cost reduction*. This can be accomplished by using a reduced set of sensors or inexpensive sensors measuring y, when the measurement of z could only be performed by complex or expensive sensors. In this way, the measurement of z is performed only in an initial set of experiments for the proper design of the virtual sensor.

Borodani discussed in [82] one interesting application for each scenario.

In the first application the safety-critical, feedback control loop of ESP (Electronic Stability Program) is considered. Commercially available ESP systems normally use the measured steering wheel angle and vehicle speed to determine the desired response of the vehicle in terms of yaw rate (and sometimes vehicle sideslip angle or sideslip rate). Then it compares the *desired* response with the *measured* (yaw rate) or estimated (sideslip angle) ones. When the discrepancy increases above a certain threshold the system applies the brakes to reduce such a difference. The yaw rate and lateral acceleration sensors' signals are then essential for a proper operation of the ESP. For this reason, in [82], a virtual sensor is designed to perform precise yaw rate signal (z) reconstruction from the velocity signals at each wheel, the steering angle and the lateral acceleration of the vehicle (y).

In the second application the cost reduction possibility for the vertical dynamics control of shock absorbers is investigated. A common control strategy for this kind of application is the "skyhook" approach, which adapts the damping ratio according to the running conditions. The minimum set of physical sensors necessary consists of three vertical accelerometers installed on the vehicle frame and other two, on the front wheel hubs. In the paper a signal reconstruction is carried out to substitute this original set of five sensors with four stroke sensors, mounted on the dampers bodies, with the objective of not modifying the existing control law. This requires that using these four signals the chassis modal velocities (heave $\dot{z}$, roll $\dot{\phi}$ and pitch $\dot{\theta}$) must be dynamically reconstructed. Also in this case, starting from experimental data a virtual sensor can be successfully designed to match required performances.

3.3 Ongoing Research and AD in Urban Scenarios

It was already mentioned how several companies and research institutes are investing heavily in the field of Autonomous Vehicles (AV). As interest for this area of automotive research and development builds up different actors have started to look at possibilities to make this new opportunity profitable. Various venture-capital backed companies have received funding or support from established automotive brands or technology businesses. Consider, for instance, that it has been recently announced that General Motors will acquire the San Francisco-based autonomous vehicle technology developer Cruise Automation for more than $1 billion [84].

Let us review some of the work carried out in the field of autonomous driving, see [85].

Audi [13] has revealed a number of autonomous vehicle prototypes from their A7 and RS7 models. The AD feature developed by the company — referred to as "Audi Piloted Driving" — is set to be commercialized on the next generation of the brand's premium saloons and it is expected to qualify either as Level 2 or Level 3 Automation. Audi is also part of the German consortium — including Daimler and BMW — that bought Nokia's HERE mapping assets in 2015, a fundamental step in the direction of capable AV.

The BMW and Chinese search giant Baidu 2014 partnership has resulted in the production of an autonomous 3 Series-based prototype that drove an 18.6-mile route around Beijing. BMW has recently announced an aggressive strategy for the promotion of electric and automated vehicle under the banner “BMW iNEXT” [86].

The world’s largest automotive supplier, Bosch, has recently largely invested in AD applications [14]. More than 2,000 engineers have been dedicated by the company to driver-assistance systems. Vehicle connectivity is another important area of interest for Bosch, which agrees to the projection that 2020 will see “driverless” cars in action, at least on highways [87].

Several European trucking brands - DAF, Daimler, Iveco, MAN, Scania and Volvo - experimented the already discussed concept of *platooning*, in which multiple trucks form a “train” controlled by a lead truck [88]. Advantages of this driving approach include lower cost (fewer people are required for control) and efficiency (lower space occupied by the convoy on the road and fuel savings provided by the lower aerodynamic drag).

Delphi, a large automotive supplier headquartered in the UK, has developed sensors and softwares packages to transform existing car models into AV. An Audi SQ5 outfitted with Delphi technology completed a 3000-mile trip across the US, performing 99% of the driving by itself [11]. More recently, Delphi showcased a concept of human-machine interface which is designed to bridge the gap before Level 4 Automation is ready. The concept vehicle is designed to encourage consumers to trust the car AD capabilities, while still keeping drivers vigilant so they can take the wheel if necessary [89].

Ford plans for promoting innovation, including vehicle connectivity and autonomous vehicle, culminated in the formation of the subsidiary Ford Smart Mobility LLC in March 2016 [90]. Ford has also pioneered the testing of selfing-driving cars in hostile environments, such as snowy Michigan and complete darkness.

General Motors is one of the most aggressive companies in AD. Besides GM’s acquisition of Cruise Automation, other important investments where made in the companies Sidecar and Lyft. Furthermore, GM has also been developing its own semi-autonomous technology in-house, with its “Super Cruise” technology slated for commercialization on high-end Cadillac models in 2017.

Google’s “X” company has led one of the most high-profile autonomous vehicle programs

[8]. The company rolled out several in-house prototypes, some of which are not fitted with steering wheels [91]. Google has recently opened to collaboration with automotive OEMs, such as Fiat Chrysler Automobiles [92].

Honda is testing autonomous vehicles on Californian public streets and in the GoMentum Station proving ground. This facility is the largest secure bed site for connected and autonomous vehicles in the U.S. [93]. Honda is also offering advanced-driver assistance systems - lane keeping, automatic braking and adaptive cruise control - options even for its entry-level vehicles.

Mercedes-Benz has developed semi-autonomous prototypes fitted with "Intelligent Drive" technology which where tested on German and Californian highways and streets [9] [10]. The company has also showcased its vision for future automobile driving with the F 015 research car. The focus of this project has been the impact of AD on daily life: "anyone who focuses solely on the technology has not yet grasped how autonomous driving will change our society", emphasizes Dr Dieter Zetsche, Chairman of the Board of Management of Daimler AG and Head of Mercedes-Benz Cars. Passengers in self-driving cars can, in fact, use their newly gained free time while traveling for relaxing or working as they please (figure 3.18).

Figure 3.18: *Interior of the Mercedes-Benz F 015 seen as a living space [94]*

The automotive supplier Mobileye focuses on advanced driver assistance systems and its products are used by many vehicle manufacturers. The firm's technology is based on the use of optical vision systems with motion detection algorithms, unlike many other systems which use a combination of visual detection, radar, and laser scanning [95].

Nissan-Renault Alliance is working on the development of AVs [12], with promise of having 10 vehicles on sale by 2020 with “significant autonomous functionality” [96]. Nissan is also collaborating with Toyota in a joint effort to develop standardized “intelligent” maps.

PSA Groupe companies Peugeot, Citroën and DS announced to have reached Level 3 Automation when two Citroën cars had driven “eyes off” from Paris to Amsterdam. While this system is scheduled to arrive by 2021, semi-autonomous “hands off” modes will be available by 2020 [97].

Electric car manufacturer Tesla has been very public about self-driving technology [40] [41], claiming that fully autonomous vehicle are only “two to three years away” [98]. Company’s in-house “Autopilot” features which enable auto steering, lane changing and parking capabilities have already been discussed.

Toyota’s case is an interesting example of how critical autonomous driving is more and more perceived by automakers for their success. In fact, if in 2014 Toyota claimed that it would not have developed a driverless car, one year later the company announced a $1 billion budget for autonomous driving research. Toyota has hired staff from academic institutes (included Standford University and the MIT) and from AV companies [99].

Uber ride-sharing company is strongly investing in autonomous cars and is reportedly working with major automakers to place orders of self-driving cars [100].

Volvo, besides its trucking efforts detailed previously, has also made progress with self-driving passenger vehicles. The ambitious goal of company’s “Intellisafe” system is to make Volvo cars “deathless” by 2020. Interestingly, the company - which has also partnered with Microsoft to further its research efforts in this space - has announced that it will accept full liability when its vehicles are in autonomous mode [101]. The issue represented by legislation (or better, lack thereof) in the field of Autonomous Driving is in fact one of the major challenges that must be confronted before full AV potentiality can be exploited.

Finally, let us mention that also the field of autonomous buses is currently under development. Chinese bus manufacturer Yutong, for instance, has been researching driverless buses since 2012 [102].

3.3.1 Example of ADAS for Unmarked Urban Scenarios

In this last section of the literature review an example of an ADAS system designed for operation in unmarked scenarios is presented. It was mentioned in the introduction that the system to be designed in this project is targeted to be simple and cheap in its implementation. What it will be shown here is, on the contrary, a more complex solution to the problem of missing painted road boundaries.

C. Guo, J. Meguro, Y. Kojima and T. Naito proposed in [18] a stereovision-based multimodal ADAS system devised for expanding usability of ADAS functions to daily urban traffic and, in particular, to unmarked roads. The outcome of this work is an example of how important *comprehensive situational awareness* is for the effectiveness of advanced driver assistance systems. In order to allow proper operation of LKA (Lane-Keeping Assist), ACC (Adaptive Cruise Control) and PCS (PreCrash System) when there are difficulties detecting the targets, such as the drivable roads and the nearby vehicles, it is crucial to understand the targets in the context of the traffic scene. The system designed by Guo *et al.* aims at doing that with the multimodal system in figure 3.19.

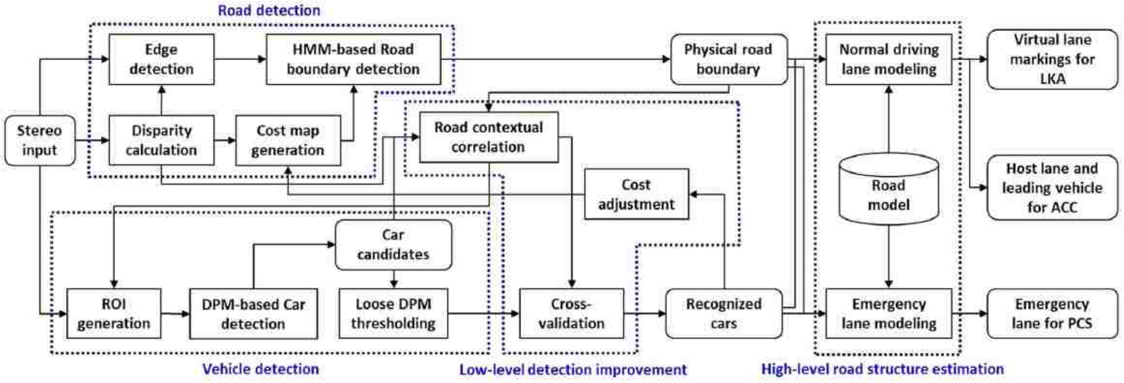


Figure 3.19: *Flow diagram of the system proposed in [18]*

As it can be seen the system consists of four modules: road detection, vehicle detection, low-level detection improvement, and high-level road structure estimation.

Road detection works with different types of roads, weather conditions and time of the day. The system uses, as the exclusive criterion for road detection, the reasonably flat road geometry properties: a *disparity map* is employed, which allows to find the most likely physical road boundary as the one between nearly flat and non-flat regions (figure 3.20).

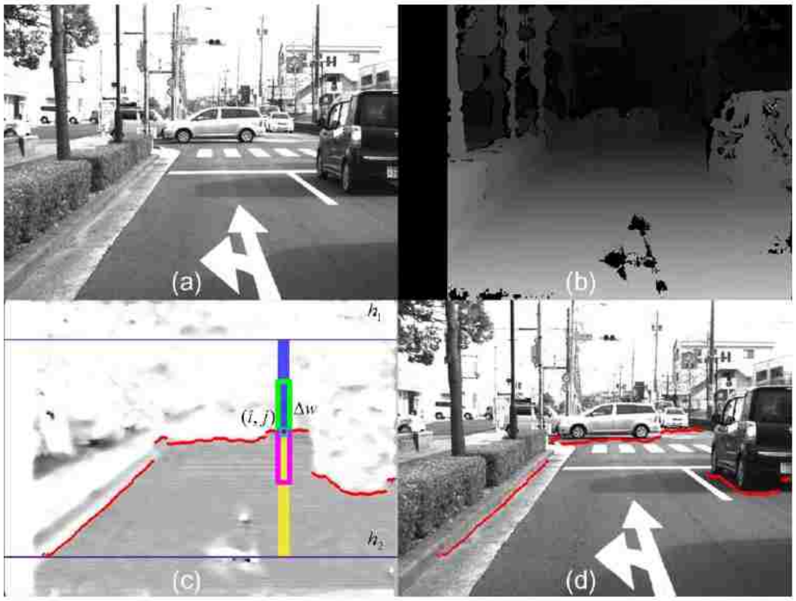


Figure 3.20: *Example of road detection: (a) Reference image; (b) Disparity map; (c) Flatness cost map: higher intensities represent big flatness costs and hence less flatness (e.g. curbs and nearby vehicles); (d) Detection result of the physical road boundary, indicated in red [18]*

Vehicle detection is performed with a *latent SVM* to learn a set of object template, figure 3.21. As shown each template consists of a "root" filter and several "part" filters:

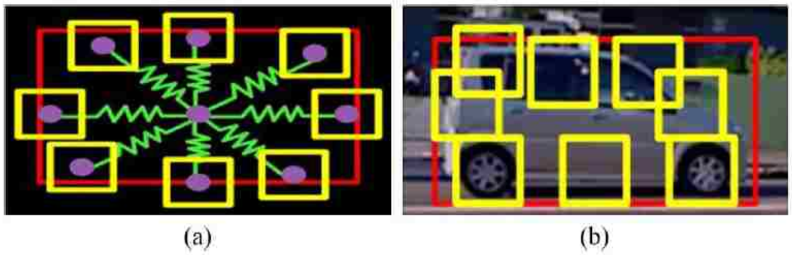


Figure 3.21: *Example of vehicle detection: (a) Model structure: the red box represents the root filter, the yellow boxes the part filters and the springs represent the deformation cost functions; (b) Vehicle detection result [18]*

the former corresponds to the outline of the vehicle, whereas the latter represent different sections of the vehicle.

Supplemental contextual scoring is then employed to reduce false positives. For instance, detected vehicle in unreasonable position or of unrealistic size are rejected. For what concerns the road structure, a clothoid road model is employed and the result is the creation of a virtual driving lane.

In summary, the complete system provides the following four outputs (figure 3.22):

- Physical road boundaries, which are used to distinguish drivable from non-drivable regions;
- Virtual lane markings, which define the driving corridor for vehicle normal driving behaviour;
- Suggested path, which fundamentally reflects the virtual lane markings but with a safety margin offset;
- Virtual PCS emergency lane, which is an obstacle-free route within the physical road boundary used for collision avoidance in emergency situations.

As a result the system designed in [18] is successful in expanding conventional ADAS functions (LKA, ACC, PCS) to normal urban streets. Nonetheless, the system relies on high software complexity and does not account explicitly for the possibility of a camera sensor failure.

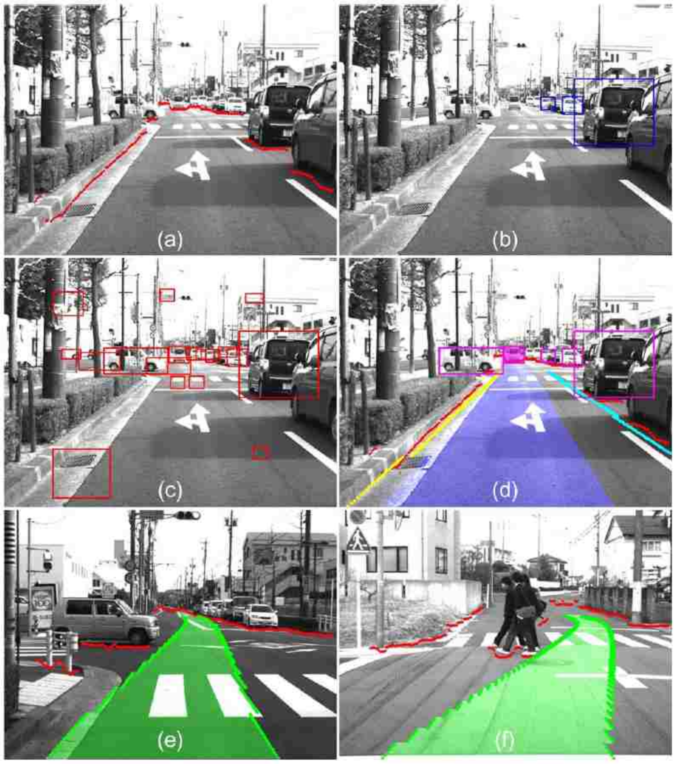


Figure 3.22: *Results provided by the system: (a) Physical road boundary; (b) Vehicle detection with many false negatives due to occlusion and far distance; (c) Vehicle detection with the candidate vehicle objects; (d) Road understanding result: virtual lane markings (yellow/cyan), suggested path (blue), vehicle detection (magenta) and leading vehicle in the suggest path of host lane (magenta region); (e)-(f) Virtual emergency lanes to avoid obstacles that interrupt the suggested path of host lane [18]*

Chapter 4

Description of the Model

In this chapter an in-depth look at the proposed model is presented. In the first part a brief description of the software tools used for the implementation of the model is provided. The remainder of the chapter presents a detailed account of the work done on each of the model components, namely the vehicle dynamics block, the EPS actuator, the LRC sensor, the longitudinal and lateral controllers, and the virtual sensor.

4.1 Blocks Modeling

Let us start this chapter mentioning briefly the software resources employed for the construction of the building blocks of the proposed model (shown again in figure 4.1).

4.1.1 Modeling Softwares

The main software used for the implementation of the blocks constituting vehicle dynamics and the automated driving system is Simulink® by MathWorks.

This software package allows one to model, simulate and analyze multi-domain dynamic systems in a graphical programming environment and it was selected for its great flexibility, the vast library of processing blocks available and the full integration with MATLAB®.

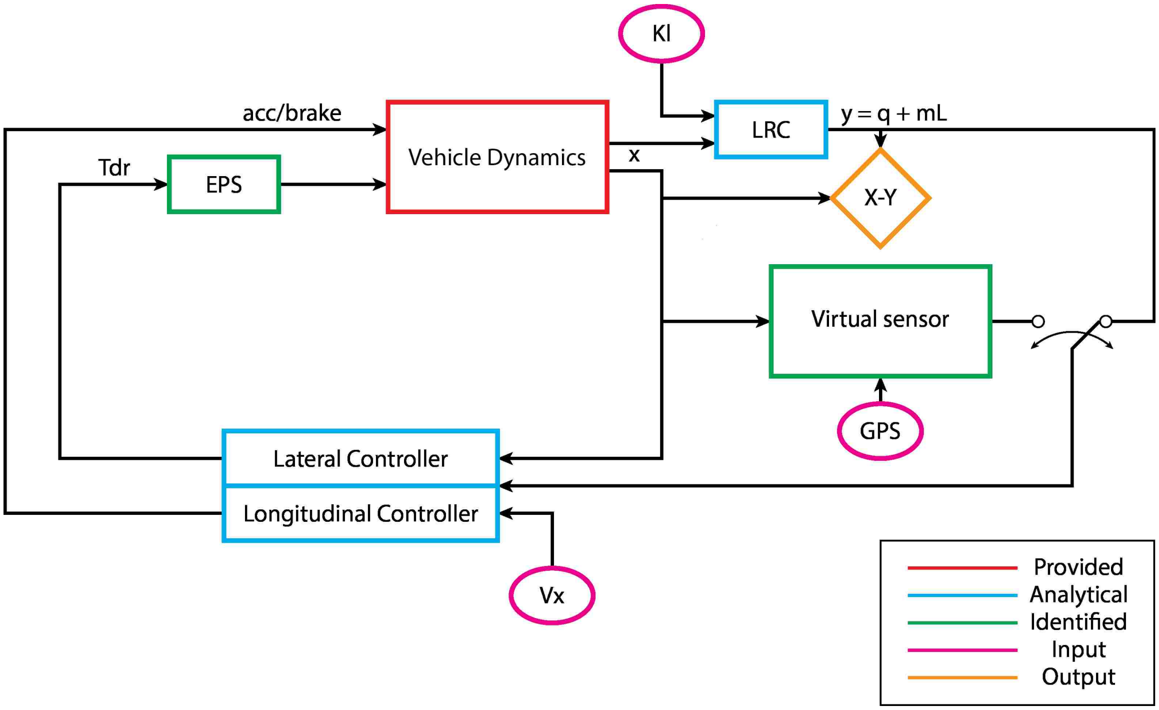


Figure 4.1: *The proposed Control Scheme*

As it is compatible with the requirements imposed by the provided vehicle dynamics S-Functions, the fixed-step "ODE4" solver is employed. This solver implements the classic Runge-Kutta method and allows for an extremely fast and efficient model simulation.

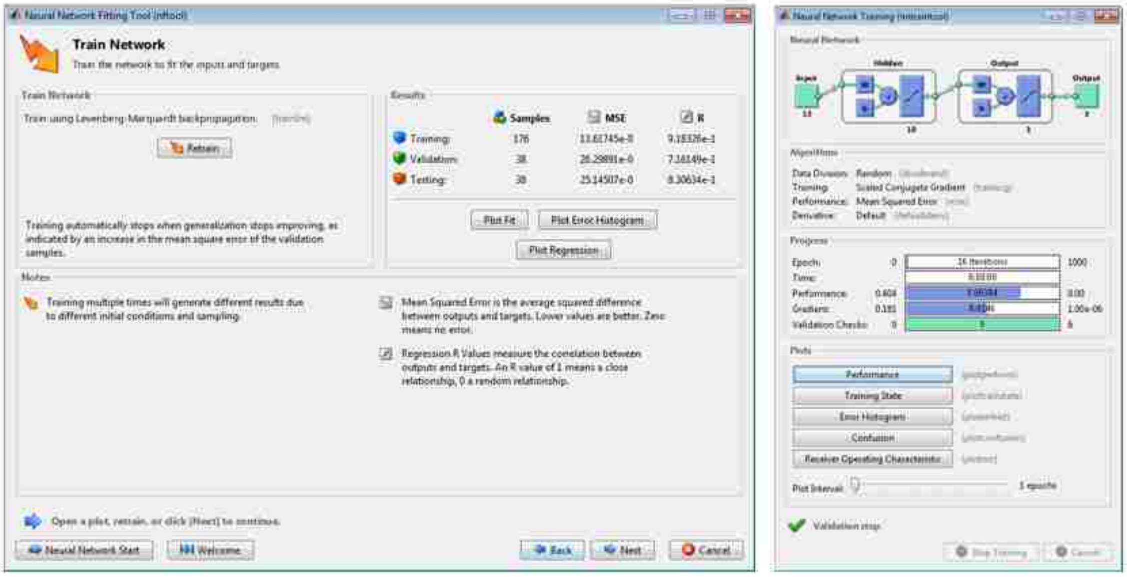
In addition to the Simulink package, two MATLAB toolboxes were used in the modeling of the system's blocks. Let us first briefly introduce the functionalities of these two blocks and then discuss in more detail the technical foundation of each technique supported by the toolboxes. Furthermore, the simulating software package CarMaker® was also employed for the generation of vehicle data useful for the development of the virtual sensor.

Neural Networks Toolbox

This toolbox was employed for the training of neural networks with the objective of modeling the highly non-linear relationship between input and output data.

The interface of the toolbox (figure 4.2) allows for the easy creation of the neural network and subsequent implementation of the model.

The toolbox has advanced functionalities for the creation and training of networks, even



Train using Levenberg-Marquardt backpropagation. (trainlm)

	Samples	MSE	R
Training:	176	13.61745e-0	9.18326e-1
Validation:	38	26.29891e-0	7.38149e-1
Testing:	38	25.14507e-0	8.30634e-1

Training automatically stops when generalization stops improving, as indicated by an increase in the mean square error of the validation samples.

Training multiple times will generate different results due to different initial conditions and sampling.

Mean Squared Error is the average squared difference between outputs and targets. Lower values are better. Zero means no error.

Regression R Values measure the correlation between outputs and targets. An R value of 1 means a close relationship, 0 a random relationship.

Data Division: Random (dividerand)
Training: Scaled Conjugate Gradient (trainscg)
Performance: Mean Squared Error (mse)
Derivative: Default (defaultderiv)

Epoch:	0	16 iterations	1000
Time:		0:00:00	
Performance:	0.604	[illegible]	0.00
Gradient:	0.181	[illegible]	1.00e-06
Validation Checks:	0	6	6

(plotperform)
(plottrainstate)
(ploterrhist)
(plotconfusion)
(plotroc)

Plot Interval: 1 epochs

✓ Validation stop.

Figure 4.2: *Neural Networks Toolbox interface*

those of great complexity. Graphical and quantitative information for an efficient and user-friendly validation of the created network are made available. A vast number of network architectures and training algorithms can be selected, and support of multi-core processing provides fast network training.

System Identification Toolbox

This toolbox provides MATLAB functions, Simulink blocks and an app for constructing mathematical models of dynamic systems from measured input-output data.

The toolbox is particularly useful for those situations in which the derivation of a model from first principles or specifications (a so-called *white-box* or *glass-box* model) is difficult or impossible. When such scenarios arise, *black-box* models are often employed, i.e. the primary interest is in fitting the data regardless of a particular mathematical structure of the model [103]. The toolbox allows one to easily exploit time-domain and frequency-domain input-output data to identify continuous or discrete-time transfer functions, process models and state-space models. For the purpose of this work, the toolbox was mainly employed to derive linear and non-linear ARX models (discussed in the following). Identified models are then easily implemented in Simulink using pre-defined blocks available in the library.

CarMaker

This software commercialized by IPG Automotive is a vehicle simulation tool intended to be applied to every step of the vehicle development process: from early simulations to Hardware-in-the-Loop (HIL) tests on single or multi-ECU systems through to HIL tests on large system test rigs. The most relevant applications for this software are vehicle dynamics simulations, testing and development of chassis control systems, driver assistance systems and system networks where chassis control systems interact with other vehicle areas [104]. This package is equipped with a sophisticated driver model that can perform complex driving maneuvers. CarMaker will be used to generate vehicle data necessary for the identification of the virtual sensor model.

4.2 Vehicle Dynamics

In the section of the Literature Review dedicated to the vehicle dynamics, a description of the main variables involved in both longitudinal and lateral dynamics was provided in addition to the simplified relationship between them. This was achieved by presenting two *linear* models, one for longitudinal dynamics and one (the notorious *bicycle model*) for lateral dynamics.

In the system implemented in Simulink neither of these models is used. A much more complex (and realistic) plant is implemented through two MATLAB S-Functions. Nonetheless, it has to be pointed out that the use of two distinct S-Functions to model separately longitudinal and lateral dynamics limits the amount of interaction between the longitudinal and lateral motion of the vehicle that can be captured.

An S-Function is a computer language description of a Simulink block written in MATLAB, C, C++ or Fortran. In particular, different MEX ("MATLAB Executable") files are called by the two S-Functions in such a way as to make C, C++ or Fortran code readable by Simulink.

The result is two black-box models that, for the purpose of building the overall system, need only be discussed insofar as their inputs and outputs are concerned.

4.2.1 Longitudinal Dynamics

The longitudinal dynamics block is controlled through the following inputs:

- Percentage of throttle pedal [%];
- Pressure of the brake system [bar];
- Initial gear engaged;
- Wind speed [m/s];
- Slope of the road [rad].

The first two inputs are referred to as *command inputs*, i.e. they are the user-controlled inputs employed to control the plant. They will be used as the two inputs manipulated by the controllers in order to have the vehicle speed following the targets.

The longitudinal dynamics S-Function is linked to a MEX file for the automatic shifting of the gears. As a consequence, only a good guess for the initially engaged gear has to be provided as the third input.

The last two inputs are the *disturbances* introduced in the plant by the presence of winds and/or by the inclination of the road. They are not used for control itself, but to impose boundary conditions to the longitudinal motion of the vehicle.

The outputs produced by the block are:

- Longitudinal velocity [m/s];
- Longitudinal acceleration [m/s^2];
- Engine rotational speed [RPM];
- Currently engaged gear.

The last two outputs are not of interest for the project. The longitudinal velocity is the fundamental output we want to control with a dedicated controller and it constitutes one of the inputs of the lateral dynamics block. The longitudinal acceleration, while not explicitly used for control, will be useful for imposing conditions on the maximum values of acceleration and of jerk, in order to ensure a comfortable ride for the passengers.

4.2.2 Lateral Dynamics

Similar considerations hold for the lateral dynamics block. This S-Function is linked to a MEX file which takes in input:

- Longitudinal velocity $[m/s]$;
- Steering wheel angle $[rad]$.

The first input is taken directly from the first output of the longitudinal dynamics block (figure 4.3). The second input is the angle imposed by the driver to the steering wheel.

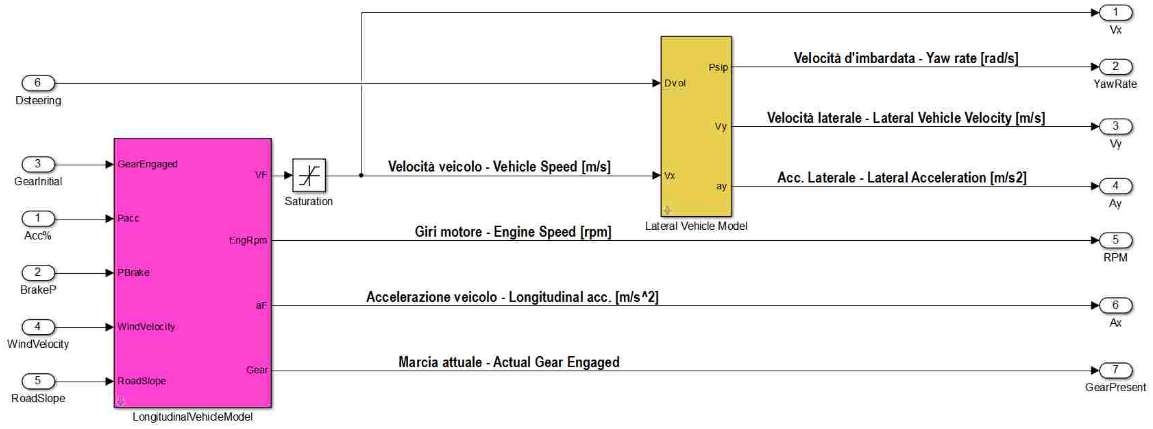

6
Dsteering
3
GearInitial
1
Acc%
2
BrakeP
4
WindVelocity
5
RoadSlope
GearEngaged
Pacc
PBrake
WindVelocity
RoadSlope
VF
EngRpm
aF
Gear
LongitudinalVehicleModel
Saturation
Velocità veicolo - Vehicle Speed [m/s]
Dvol
Vx
Psip
Vy
ay
Lateral Vehicle Model
Velocità d'imbardata - Yaw rate [rad/s]
Velocità laterale - Lateral Vehicle Velocity [m/s]
Acc. Laterale - Lateral Acceleration [m/s2]
Giri motore - Engine Speed [rpm]
Accelerazione veicolo - Longitudinal acc. [m/s^2]
Marcia attuale - Actual Gear Engaged
1
Vx
2
YawRate
3
Vy
4
Ay
5
RPM
6
Ax
7
GearPresent

Figure 4.3: *Longitudinal (magenta) and lateral (yellow) dynamics blocks. The seven outputs are represented in bold font. The saturation block is used to avoid null longitudinal speed being fed to the lateral dynamics block*

Since in a typical lateral dynamics control system the variable acted upon is the torque applied to the steering column, the EPS actuator must be added to the model.

The outputs provided by the lateral dynamics block are:

- Vehicle yaw rate $[rad/s]$;
- Lateral velocity $[m/s]$;
- Lateral acceleration $[m/s^2]$;

The later acceleration is not of particular interest for the control, even though similar considerations to those applied to the longitudinal acceleration and jerk can be made.

Lateral velocity and yaw rate, on the other hand, are the crucial variables determining the lateral dynamics of the vehicle and as such are employed in the control. They will be used as inputs of the analytical model of the Lane Recognition Camera, whose outputs will be the fundamental parameters used for lateral control.

4.3 EPS - Electric Power Steering

The necessity of modeling the Electric Power Steering was justified in the previous section. In the literature review the operating principle of this actuator was explained. The objective is to determine a model as described by equation 4.3.1 able to link the output (steering wheel angle, $\delta_{steering}$ $[deg]$) to the two inputs (torque applied by driver, T_{dr} $[Nm]$ and vehicle velocity, Vel $[km/h]$).

$$\delta_{steering} = f(T_{dr}, Vel) \tag{4.3.1}$$

Now, for the purpose of this study the details of the EPS control loop are supposed to be unknown, hence a *black-box* model is what will be derived and implemented in the overall system. The only aspects about this system that will be considered as known are:

- The system is a low order one;
- The frequency plot of the $\frac{\delta_{steering}}{T_{dr}}$ transfer function is supposed to be known (green curve Fig. 4.4, model linearized for a velocity $\simeq 60$ km/h).

The first point expresses the necessity of identifying models of moderate complexity, i.e. *without* resorting to high orders of the *narx* model or excessively large numbers of nodes in the ANN model. The second result basically confirms the first one: the behaviour is that of a low order model, with a pole clearly identified between 2 and 3 Hz. A resonance between 10 and 20 Hz can also be determined. This resonance is eliminated if the green curve (which deals with all the functions of the EPS actuator implemented) is considered.

4.3.1 Experimental Data

Experimental data have been made available as readings from the CAN network of the vehicle (2015 Fiat *500X*). Tests have been conducted on two days (07-29-2015 and 07-30-2015)

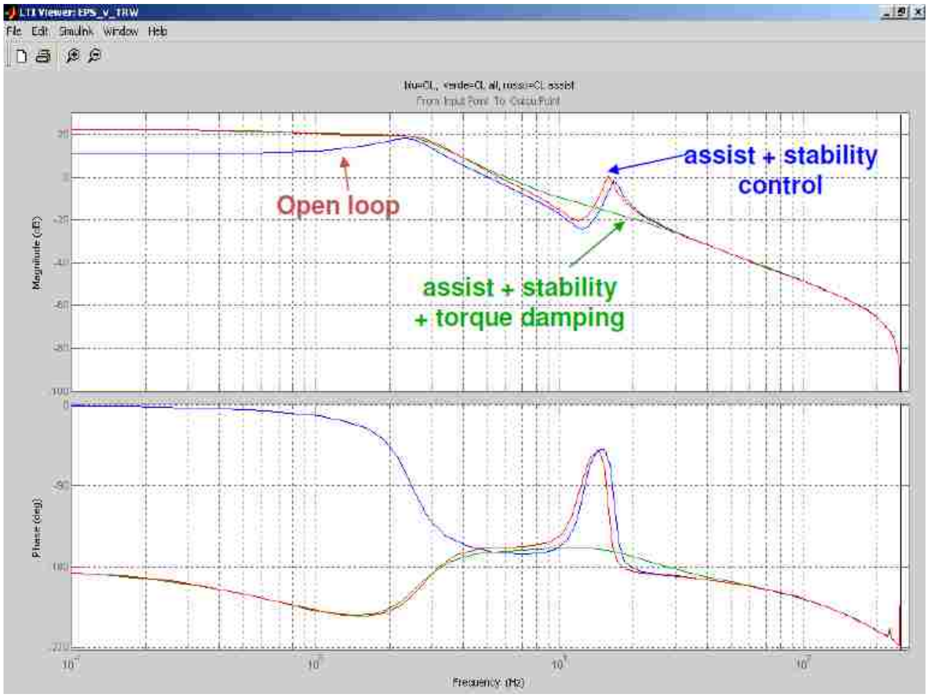

LTI Viewer: EPS_v_TRW
File Edit Simulink Window Help
blu=CL, verde=CL all, rosso=CL assist
From: Input Point To: Output Point
Magnitude (dB)
20
0
-20
-40
-60
-80
-100
Phase (deg)
0
-90
-180
-270
10^{-1}
10^{0}
10^{1}
10^{2}
Frequency (Hz)
Open loop
assist + stability control
assist + stability + torque damping

Figure 4.4: *Bode plot of the transfer function from T_{dr} to $\delta_{steering}$, courtesy of P. Borodani*

at FCA proving-ground “Balocco”. Testing maneuvers simulate sharp steering movements (*steps*) of various amplitude and at different speeds and *sweeping* tests at variable frequency, amplitude and vehicle speeds (see figure 4.5). Maximum driver torque T_{dr} in the tests is limited for safety reasons to ± 3 to 4 Nm.

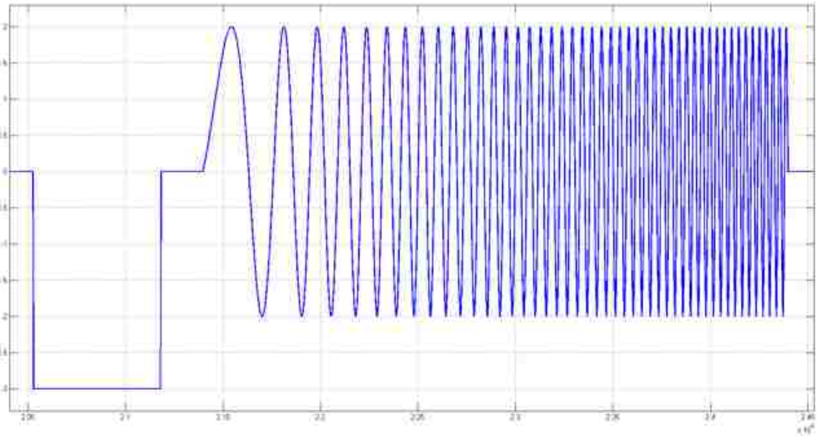


Figure 4.5: *Portion of experimental torque input signal T_{dr} $[Nm]$ vs. time t $[s]$ displaying a step test and a sweep test*

For the analysis of the EPS only 6 of the more than 800 CAN-generated signals are of interest:

- Input signals:
  - *VehicleSpeedVSOSig*: vehicle speed in $[km/h]$;
  - *TorqueOverlaySteeringReq*: driver torque in $[Nm]$:
- Output signals:
  - *LwsAngle*: steering wheel angle in $[deg]$;
- Fault indicator signals:
  - *TorqueOverlayIntActivated*: indicates whether the EPS is actuating the request (must be = 1);
  - *TorqueOverlayFault*: indicates a fault in the system (must be = 0).

Data processing is carried out as follows. First, a MATLAB script examines the entire data collection and builds one input matrix and one output vector only if the values of both fault indicator signals indicate that the system is working properly. Subsequently, data are interpolated according to a common time reference. It is, in fact, necessary to consider that each signal is registered by a different sensor and although the updating frequency (100 Hz) is the same for all of them, they start to register at slightly different instants in time. Network training attempts have been performed without interpolation, but consistently lower performance values were found. Considered time instants are then further reduced, eliminating infinite or non-existent values (`NaN`) and restricting the signals only to the actual test intervals. Finally, as shown in figure 4.6, a shift of the output steering angle $\delta_{steering}$ is performed in order to have null values when the input torque T_{dr} is zero.

4.3.2 Neural Network Models

A large number of models were derived using different kinds of ANNs in order to obtain satisfactory performance values. For brevity, in the following only the base and the final models are presented.

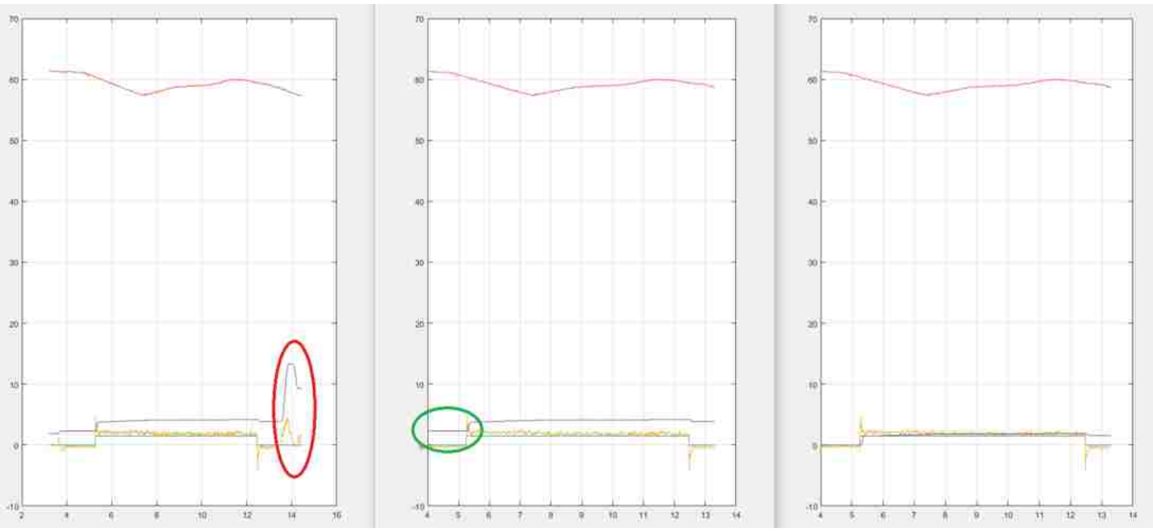
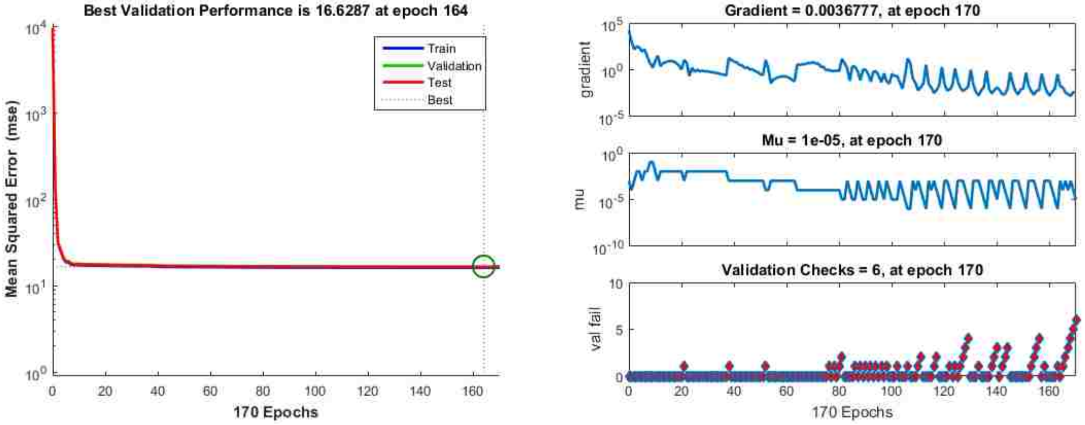


Figure 4.6: *Example of experimental step test. Displayed signals: red, vehicle speed (Vel), blue, input torque (T_{dr}), yellow, assist torque provided by EPS (T_{dem}) and violet, output steering angle ($\delta_{steering}$).* ***Left figure****: the red circle highlights a portion of the test where the torque provided by the EPS and thus the steering angle is different from zero even when T_{dr} is null: this behaviour cannot be explained by our model and it is hence eliminated.* ***Middle figure****: the green circle shows the necessity of shifting the output steering angle $\delta_{steering}$ to zero when the input torque is null.* ***Right figure****: final result.*

Using data from both test days a *feedforward* neural network is trained using all the default settings of the NN Toolbox$^{\mathrm{TM}}$:

- Levenberg-Marquardt training algorithm;
- Randomly divided data into training, validation and test sets;
- 1 hidden layer with 10 neurons;
- Hidden layer transfer function: *tansig*.

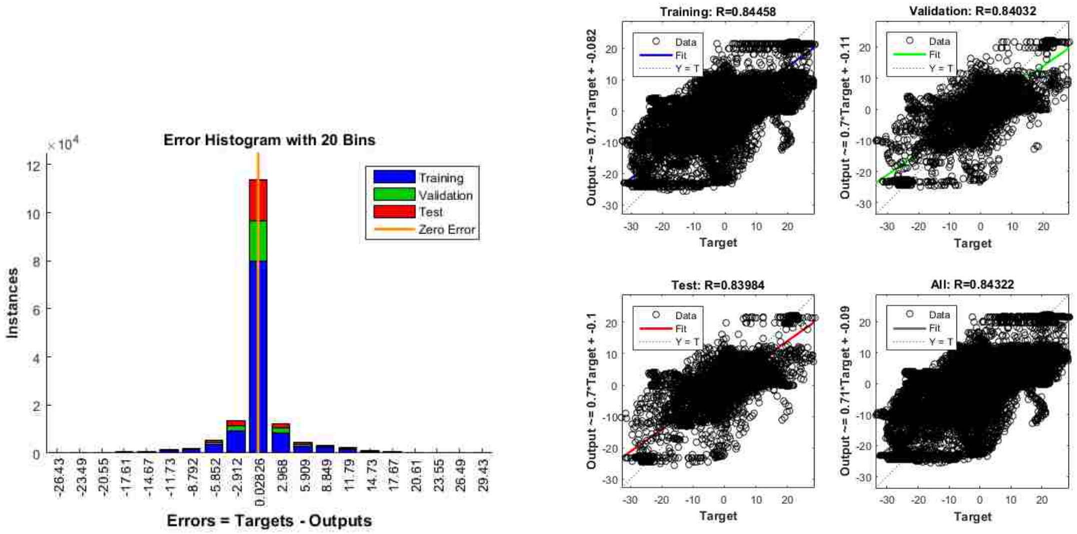
The training requires on average about 160 epochs before coming to a stop when the number of validation checks (introduced in the following) is reached. Performance values of the network are illustrated in figures 4.7 and 4.8. Since data are divided in the training, validation and test data sets (see [108]) randomly, each time a new configuration of the NN parameters is tried the simulation must be run several times in order to ensure that



(a) Validation performance: *MSE*

(b) Training history

Figure 4.7: *Default ANN model results: MSE and training history*



(a) Error histogram

(b) Regression plot

Figure 4.8: *Default ANN model results: error histogram and regression plot*

significantly better or worse than average results are not mistaken for the true behaviour of the network.

What we see in figures 4.7 and 4.8 are the four plots generated by the toolbox:

1. The parameter used to evaluate the performance of the network (in this case the mean

square error [1]) is plotted against each pass of the $(input, output)$ data pairs, called an *epoch*. Ideally training should come to a stop when such parameter has met a value as close as possible to zero (within the limits imposed by *overtraining*);

2. In the second plot the parameters of interest are the gradient of the performance curve (which should be close to zero, as to indicate that we have reached an asymptote) and the number of validation checks. As stated by [108] the number of validation checks represents "the number of successive iterations that the validation performance fails to decrease". When such value is met (before the performance target) the training comes to a stop;

3. The *error histogram* shows how the errors $t_i - y_i$ are statistically distributed (ideally a peak around 0.0 should appear);

4. Finally the *regression plot* provides an insightful information on how well the trained network approximates both the training data and two different data sets that have *not* been used for the training. The correlation coefficient 'R' provides an indication of how well the fit line interpolates the $(target, output)$ data pairs (a value at least equal to 0.75 is advisable to prove that some correlation exists; a value of 1 indicates perfect fit) and the slope of the line tells us how close the fit line is to the perfect agreement between real system and mathematical model represented by $Y = T$.

Compared to preliminary attempts made on crude, non-elaborated data the performances shown by the default neural network appears to be rather good. Interpolation and non-physical data elimination prove to be important phases in the preparation of training and validation datasets. Nonetheless, performances can be further improved by proper tweaking of the network and training parameters.

In general, greater performances (better agreement between model and real system) can be achieved by:

[1] $MSE = \frac{1}{2n} \cdot \sum_{i=1}^{n} E_i^2 = \frac{1}{2n} \cdot \sum_{i=1}^{n} (t_i - y_i)^2$ where t_i is the target value, y_i is the actual output value and n is the number of $(input, output)$ data pairs. Note that MSE is computed referring to un-normalized target and output values, although for enhanced performances the ANN is trained with normalized data.

- Increasing the number of validation checks: an highly non-linear system may have a performance function with many *local minimums.* In order to avoid to remain trapped in one of such minimums and to search for the global minimum a larger number of validation checks can help;

- Increasing the number of nodes in the hidden layer: a network with a larger number of DOFs (degrees of freedom, i.e. the weights and biases of the neurons) is more capable of capturing the non-linearities present in the training data;

- Increasing the number of hidden layers: theoretically, a larger number of hidden layers should provide greater interpolating capabilities to the network. It is however typically assumed that *one hidden layer is sufficient for the large majority of problems* [109];

- Changing the training algorithm: considering the large size of the training data set the Toolbox guide [105] suggests the use of the *Scaled Conjugate Gradient* learning algorithm: `trainscg`. This is related to its smaller memory requirements;

- Changing the nodes' transfer function: when the application of the ANN is — like in this case — model fitting, a good choice is represented by the *Radial Basis* transfer function. When this kind of approach is used the hidden layer neurons are radial effect functions[2] like the Gaussian function. The advantage of this type of function in a neural network application, as stated in [111], is that the output of the hidden layer node will have a large value only when the input signal is near the center of the range which characterizes the utilized function. If such centers are properly chosen, then, RBF networks allow for fast and efficient training and good approximations.

These general guidelines were combined in several attempts in order to improve the quality of the EPS model. In particular, satisfactory results were obtained using a one-layer feedforward network implementing radial basis transfer functions. Many rules of thumb exist for the determination of a starting number of neurons to be implemented in

[2] A RBF (Radial Basis Function) is a real-valued function whose value depends only on the distance from the origin: $\phi(x) = \phi(\|x\|)$[110].

the hidden layer [109]; the one used here is expressed by equation 4.3.2.

$$N_h = \frac{N_s}{(\alpha * (N_i + N_o))} \tag{4.3.2}$$

where N_s is the number of samples in the training data set (default = 70% of all data), N_i and N_o are the number of input and output neurons, respectively, and α is an arbitrary scaling factor (range $[2 \div 10]$). For the data produced during the EPS tests, this results in $N_h = 3000$ approximately using a value of α between 9 and 10.

When such a large network is employed, model fitting of the training data appears good (i.e. the coefficient R is above 0.88, slope coefficient of 0.78 and data compactly organized around the fit line) but lack of generalization is present (the regression parameter is significantly smaller for validation and test data sets). This phenomenon is referred to as *overfitting*.

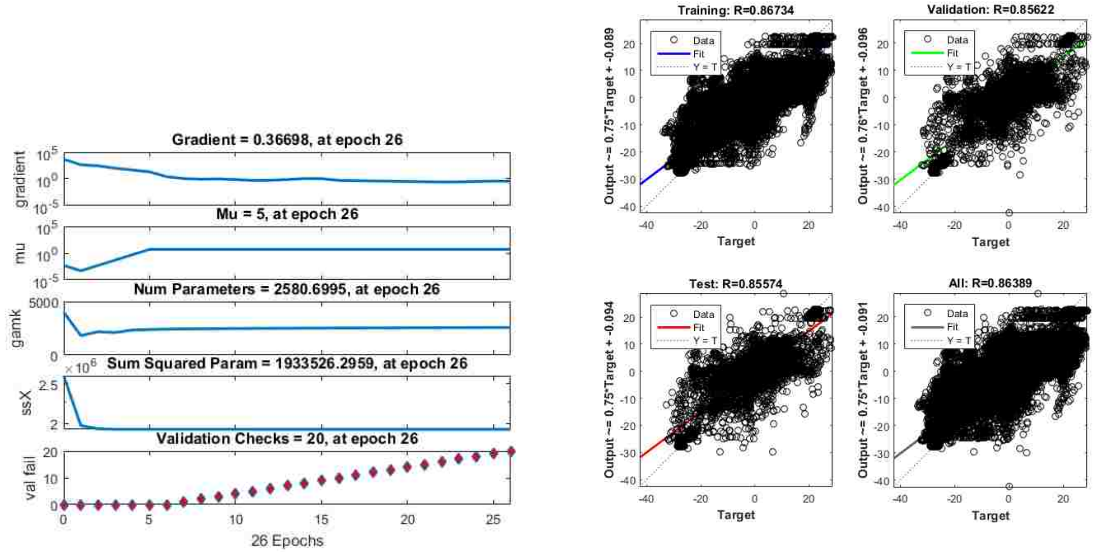
When an overfitting phenomenon occurs it means that the network has learned to model training data but it has lost the capability to generalize to new situations (i.e. other data sets). One could say that the network has learned the training data *too* well. This result could be caused by the use of a too complex network. In fact, when the size of the network increases, also the DOFs of the network increase as well as its ability to generate complex functions. When the network is powerful enough to fit precisely every experimental point it may lose the capability of generalizing since it has incorporated the measurement noise. A first possible solution to avoid overfitting is then exactly that of *reducing* the size of the network, decreasing its “power” or degrees of nonlinearity. Of course, knowing a priori the best dimension of the network for a certain application is not possible. Recognizing this difficulty, the NN toolbox provides two other methods to improve generalization: *regularization* and *early stopping*.

The latter is exactly what the validation dataset is used for: stopping the training when the minimum of the performance index is reached not for the training data set, but for completely independent data, known as the validation set.

The former requires a change in the performance function (default : $F = MSE$). It is possible, in fact, to add a term equivalent to the sum of the network parameters (weights and biases) squared: MSW. The network training will be still aimed at minimizing

$F = \gamma\, MSE + (1-\gamma)\, MSW$[3], but in this case weights and biases will be forced to be smaller, which causes a network response to be smoother and less prone to overfitting. Still, the choice of the parameter γ is not straightforward and hence an algorithm designed to automatically select this regularization parameter would be advisable. One training function made available by MATLAB that implements this is `trainbr`, which incorporates a statistical technique referred to as Bayesian regularization (see [112] for details).

In order to reduce the likelihood of overfitting, the number of nodes is reduced to 1000, the number of validation checks to 50 (from the previous 200) and the training algorithm changed to `trainbr`. The results of such network over experimental data are shown in figure 4.9.



(a) Training parameters

(b) Regression plots

Figure 4.9: *Learning algorithm:* ***`trainbr`***

A possible way of dealing with the presence of noise that is corrupting the data is that of smoothing down the experimental signal. MATLAB offers different tools to cope with the problem. The command that was used here is `smooth`, which can be used to build a

[3] Note that, as previously mentioned, input/output data are normalized prior to NN training. Once this phase has ended the output and F are computed as un-normalized: it is then difficult to determine a target value for the performance function.

moving average filter (figure 4.10).

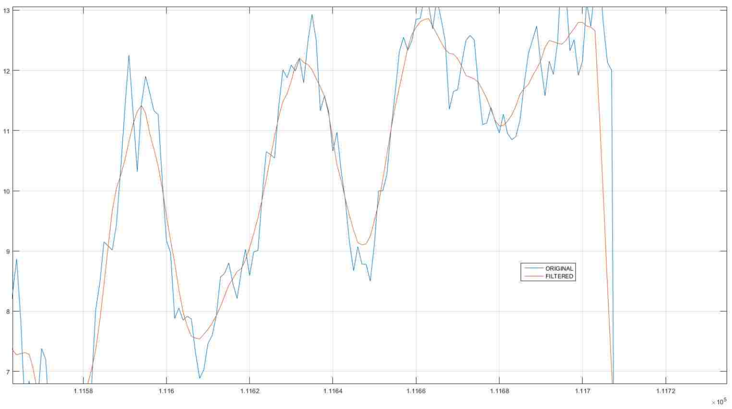
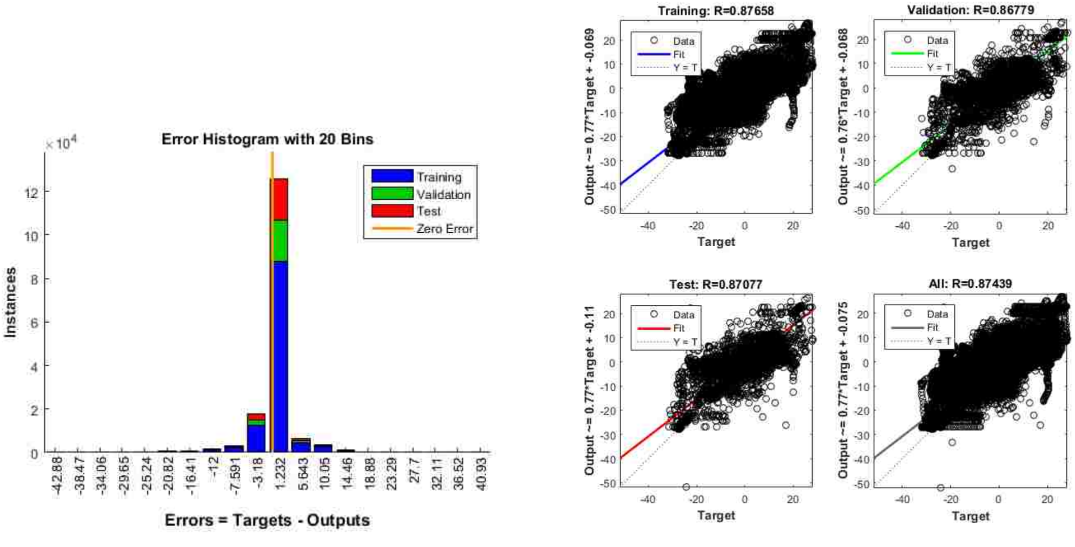


Figure 4.10: *Detail of the T_{dem} signal before (blue) and after (red) filtering*

Data appear to be better interpolated by the fit line (figure 4.11), which has an angular coefficient closer to unity. Further refinements of the networks provided only minor improvements, hence a regression coefficient of 0.87 was accepted has the best result achievable by a standard, feed-forward neural network.



(a) Error histogram

(b) Regression plots

Figure 4.11: *Smoothed data. Learning algorithm:* `trainbr`

4.3.3 Non-linear ARX Models

The same data sets used for NN training are employed for ARX non-linear model estimation. With the help of the `iddata` command two data objects (one for identification, one for validation) are created and then loaded in the System Identification Toolbox.

The approach by means of which the non-linear ARX model is determined is iterative in nature: the objective is to determine the *simplest* model able to suitably describe system dynamics. When performance goals are not met another model structure, model order and/or identification algorithm must be tried until the model is considered good enough. If needed data preprocessing and noise modeling can be used (see [103] and [113]).

The System Identification App (figure 4.12) in the MATLAB environment provides powerful tools to quickly load data and estimate a model. A first attempt is performed considering all the default model and estimation options. Regressors are built with two input and two output terms and the delay is set to one. The model output $y(t)$ is then estimated using the following non-linear autoregressive equation:

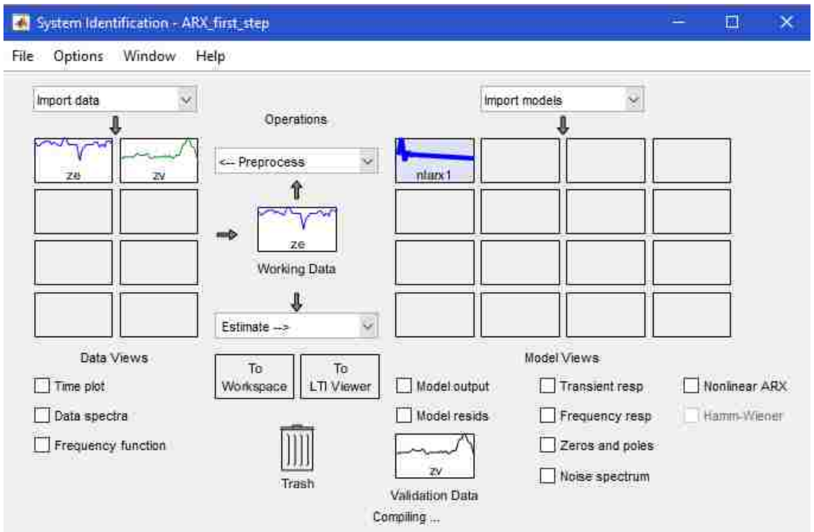


Figure 4.12: *System Identification App interface*

$$y(t) = f\left(y(t-1), y(t-2), u(t-1), u(t-2)\right)$$

where f, the nonlinearity estimator, is `Wavelet Network` by default. Wavelet Networks are actually kinds of Neural Networks in which the processing function corresponds to Wavelet transform. Such transforms closely resemble Fourier transforms in which the test function (the wavelet) allows one to focus on local regions of the signal (see [114]) and decompose it. With a fit of just over 50% the standard *narx* model proves to be inferior to the base ANN model. An extensive trial-and-error procedure is then started with the objective of improving the quality of the identified model.

Because of the highly non-linear characteristics of the EPS operation the derivation of a suitable model requires more complex autoregressive structures. Non-linearities are not only ascribable to the influence of the vehicle speed on the assistance curve, but also to the hysteresis cycle due to self-centering of the steering wheel. This phenomenon is particularly strong for low levels of torque applied to the steering column and it tends to make the model less accurate in this region.

At the end of the identification process, two different *narx* models were identified. The first and best performing one was identified resorting to the following modifications to the base structure:

1. Increased model order. Keeping in mind the objective of employing an order as low as possible while still aiming for the highest performance, a satisfactory result was achieved by resorting to a *third order* model. This means that three past values of both the output ($\delta_{steering}$) and the inputs (T_{dr} and Vel) are accounted for;

2. The selected regressors consider also *custom expressions* between them. For instance, it was noted that at increasing speeds the same level of applied torque produces increasingly small steering angles. As a result a custom regressor, inversely proportional to the past value of vehicle speed, was added. Additionally, to better capture the interaction between the two inputs, a certain number of products between the past values of the inputs were added;

3. Increased number of units in the wavelet network. This parameter is found to be important to increase the DOFs of the model, making it more suitable to approximate highly non-linear relations. Sigmoid non-linearities were also tried, without much success.

The model determined in this way proves to be successful in capturing the complex characteristics of the EPS actuator. Figures 4.13 and 4.14a show the performances of this model. The general behaviour of the system is well captured as it can be seen from the comparison between the model output and the validation data, and the resulting regression coefficient plot which shows (almost) all points compactly organized around the ideal fit line.

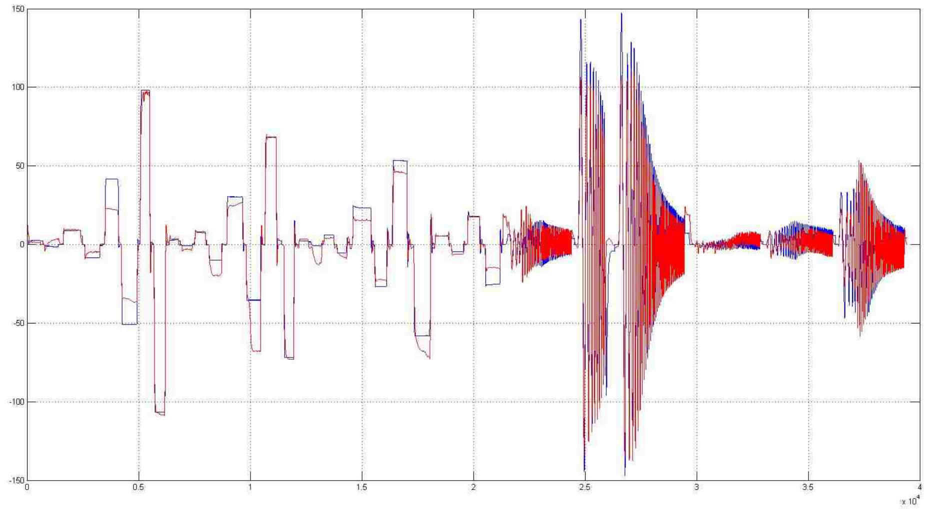
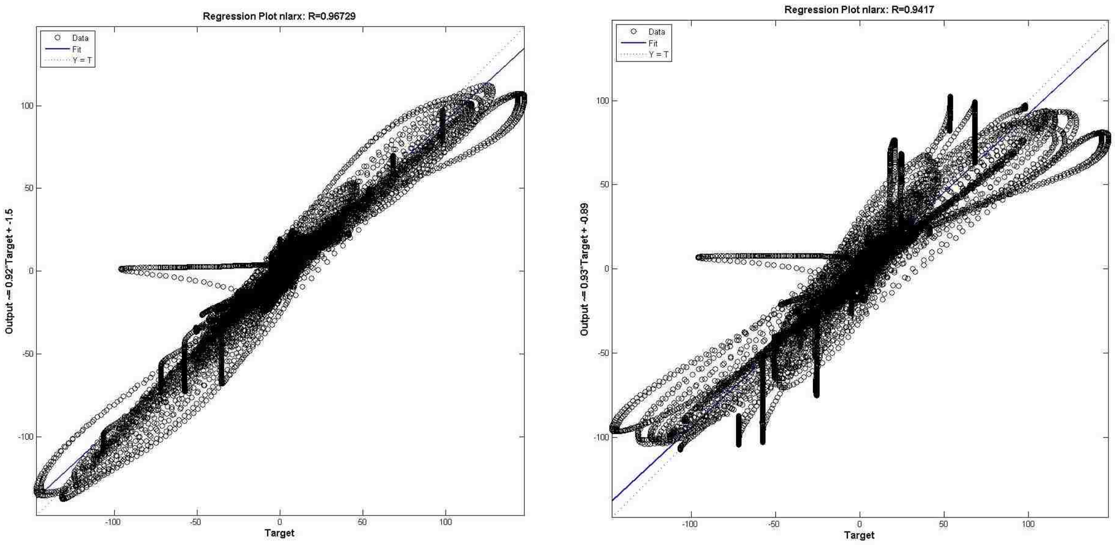


Figure 4.13: *Performance of NARX model with custom regressors: plot superposition between validation data (blue) and model output (red)*

This model matches the requirement of explaining more than 95% of data and it surpasses all models achieved with Neural Networks. Nonetheless, when simulating the model with the validation signal composed of 28 steps and 11 sweeping tests the required CPU time surpasses 20 minutes. When this block is integrated with the other ones composing the system, this time increases even more.

A second model was developed trying to keep the good fit, while reducing the computational cost of the model. A systematic analysis of the influence of the model parameters was carried out and a satisfactory result is achieved when the cross products among the different regressors are removed.

Furthermore, identification data were modified by adding a small portion of signal in which — at different speeds — the steering angle $\delta_{steering}$ and the applied torque T_{dr} are



(a) Regression plot for $NARX$ model with custom regressors: $R = 0.967, slope = 0.92$

(b) Regression plot for $NARX$ model *without* custom regressors: $R = 0.942, slope = 0.93$

Figure 4.14: *Comparison between the two EPS models*

both null. While performances are not affected, the centering of the steering wheel is improved when no torque is applied. This will allow the controller of the EPS to provide less torque to maintain the direction of travel in straight portions of roads, with subsequent lower energy consumption of the system.

Compared with the previously discussed model, the degradation in performance (regression coefficient just below 95%, figure 4.14b) is outmatched by the faster computational times: the same validation signal — when fed to this second model — results in CPU times just above 2 seconds, i.e. 600 times faster than the previous outcome. This will allow the possibility of running a large number of simulations of the overall system during the design of the controllers and the position observer. Once this design phase is completed and greater accuracy sought, the first model can be easily restored in Simulink.

4.4 LRC - Lane Recognition Camera

The importance of the Lane Recognition Camera in look-ahead approaches for automated driving was underlined in the literature review.

In this section the methodological approach for the development of a simple camera model starting from the dynamic states of the car model and the curvature of the road is presented. It was mentioned previously that, among the outputs of the vehicle dynamics blocks, the motion of the ego-vehicle is described in terms of:

- Longitudinal velocity: $v_x(t)$ $[m/s]$;
- Lateral velocity: $v_y(t)$ $[m/s]$;
- Yaw rate: $\dot{\psi}$ $[rad/s]$.

These quantities can be easily obtained also from the linearized dynamics. Information about the geometry of the road is provided by the curvature parameter K_l $[1/m]$, defined as the inverse of the instantaneous radius of curvature R. In [115, 116] these aforementioned quantities are combined into a dynamic system that provides a *linear approximation* of the centerline of the lane through two outputs:

- The distance, measured along the vehicle-fixed y axis, between vehicle's center of gravity and the approximation of the lane centerline (i.e. vehicle *current* position in the lane): q $[m]$;
- The angle between vehicle's longitudinal axis and the approximation of the lane centerline (i.e. vehicle direction in the lane): m $[rad]$.

As figure 4.15 illustrates, this configuration is chosen because the amount of steering angle selected by a driver to negotiate an incoming curve is based on the distance $y_{fb} = mL + q$ between this linear approximation of the centerline and the longitudinal axis of the vehicle (i.e. vehicle *future* position in the lane), L meters ahead from the vehicle center of gravity.

This parameter, termed the *look-ahead distance*, was found to cover an important role in the steering decision by M. F. Land and D. N. Lee by simultaneously recording the steering-wheel angle and the driver's gaze direction in a series of experimental tests [117].

From a mathematical point of view, the relation between the coefficients m and q, and the variables describing vehicle motion can be expressed through the following dynamic

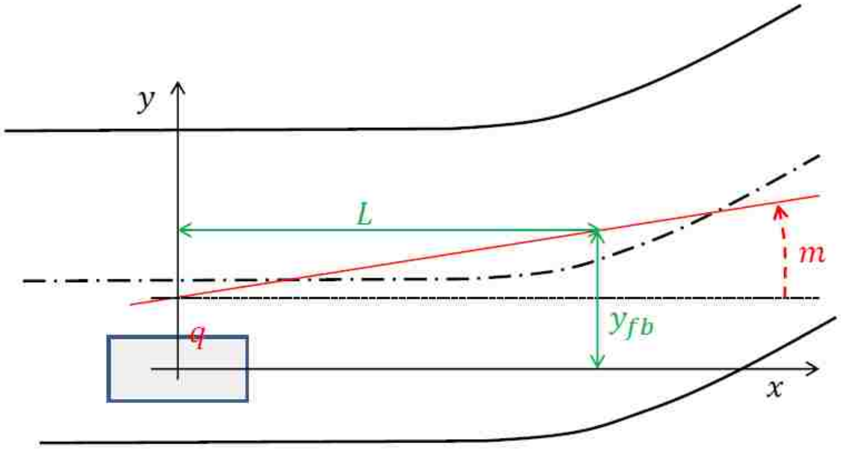


Figure 4.15: *Lane modeled through the linear approximation of its centerline*

equations:

$$\begin{aligned} \dot{q} &= v_x \cdot m - \dot{y} - L \cdot v_x \cdot K_l \\ \dot{m} &= v_x \cdot K_l - \dot{\psi} \end{aligned} \tag{4.4.1}$$

where, under the assumption of small steering angles, it was considered $tan(m) \approx m$ and K_l is measured at the look-ahead point. The validity of this simplified set of equations is guaranteed only under the assumptions of the single track model. The equations correspond to considering the vehicle initially perfectly aligned with the centerline of the lane ($m = q = 0$). Subsequently, the effect of a change in the road curve on the dynamics of m and q is observed.

For constant q, m and y_{fb}, the equations simplify to $m = \dfrac{L}{R}$ which can be geometrically interpreted as the larger the curvature radius R, the smaller m, and vice versa. From the point of view of the implementation of such equations in Simulink, a very simple block scheme can be built using standard elements: figure 4.16 . The task of the controller will be then that of acting on the steering command on the basis of the measurements provided by the vision system. Using y_{fb} as the feedback quantity leads to greater comfort compared to feedback based on position error measured at the center of gravity (*look-down* reference). The parameter q, on the other hand, being more closely related to the position of the vehicle in the lane, is the typical variable used for the definition of system specifications.

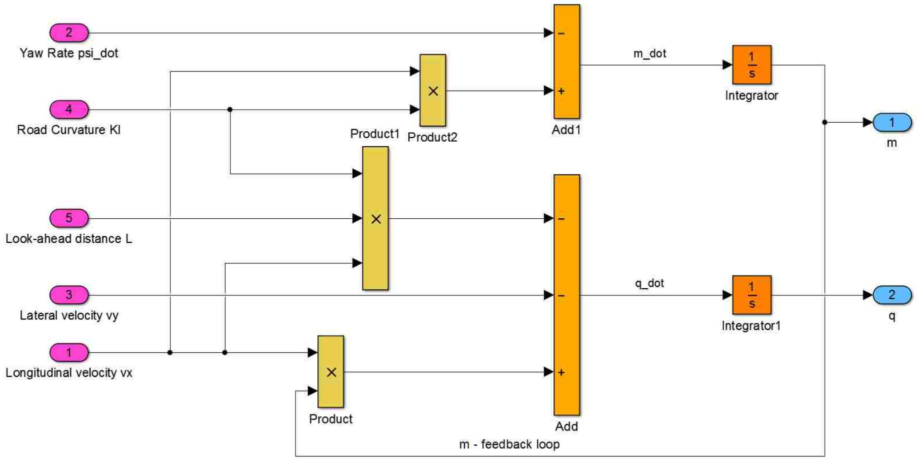


Figure 4.16: *Simplified LRC model implemented in Simulink*

Keeping in mind this observation, [116] underlined how the look-ahead distance parameter was correlated to the comfort/system performance trade-off that can be achieved by the automated driving system: an increase of L improves comfort performances even though it tends to cause a larger lateral position error at the center of gravity (i.e. a larger q) when a curve is approached.

4.5 Controllers Design

In this section, the design procedure for the controllers employed to handle longitudinal and lateral dynamics as desired by the user is presented.

Combining together the blocks discussed to this point, a full vehicle model whose dynamics can be controlled by accelerator and brake pedals, as well as by the torque applied to the steering wheel is obtained. Additionally a simplified camera model provides, given the curvature of the road, the instantaneous distance of the vehicle center of mass from the linear approximation of the centerline of the lane and the instantaneous angle between vehicle longitudinal axis and the aforementioned linear centerline approximation.

Now, as far as control design is considered, an important consideration relies in the fact that the considered system is highly *nonlinear*. This opens two distinct possibilities for the implementation of suitable controllers: either use a non-linear control technique capable of dealing with the non-linearities of the system or implement a linear control technique which, although it will not explicitly account for non-linearities, is able to control the system reasonably well when parameters do not change significantly.

Considering that:

- The speed of the ego-vehicle will vary approximately in the 60 to 130 km/h range;
- The curvature of the road will not exceed typical values of highway scenarios;
- The design of the controller is not the main focus of the project;

the latter control approach will be chosen. The first point, in fact, guarantees that vehicle motion will always be carried out in the two highest gears, with a lower and almost constant inertia of the vehicle; longitudinal vehicle dynamics will then be accurately approximated by a linear model. For the lateral vehicle dynamics, the already mentioned bicycle model will correctly predict vehicle response considering that, as guaranteed by the second point, the steering angles will be small. Finally, given that the control is not the main focus of this work, it is possible to resort to simple linear control techniques to satisfy control requirements.

4.5.1 Longitudinal Control

The task of the longitudinal dynamics controller is that of applying the correct amount of accelerator or brake pedal in order to match the instantaneous speed requested by the user (traditional *Cruise Control* functions, plus brake operation).

The design of a linear controller for this purpose requires the evaluation of a suitable linear model. This can be easily achieved through the System Identification Toolbox and its linear techniques. Data for the identification of the model are obtained through simulations of the provided longitudinal dynamics S-Function: steps of the accelerator and brake commands are used to continuously vary[4] the speed in the considered range.

The resulting linear model is a simple, discrete time, LTI (Linear Time Invariant) model of the first order. This model is MISO (Multiple Inputs Single Output): accelerator and brake pedal signals are the inputs, vehicle speed is the output. The pole of the linear model

[4] For an accurate identification it is of primary importance to avoid sharp deviations of the output(s) which cannot be directly related to variations of the input(s).

has a positive real part and, hence, it is unstable. Feedback-loop control is then required also for stabilization purposes. Figure 4.17 shows the regression plot quantifying the agreement between the provided non-linear model (*truth model*) and the linearized first-order model (*simplified model*) for a free slow-down from 80 km/h. As can be seen in figure 4.17, the agreement is very good at higher speeds, while it starts degrading below $\approx$ 12 m/s (i.e. $\approx$ 45 km/h). This is not a concern since, as already pointed out, considered speeds will always be higher than 60 km/h.

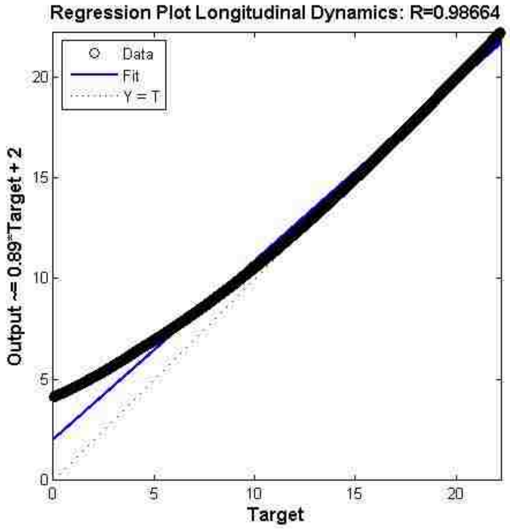


Figure 4.17: *Vehicle slow-down: agreement between truth model and simplified model*

The equivalent mass of the vehicle is given by $M+Mr$, where M is the actual translating mass of the vehicle and Mr is the equivalent mass of the rotating components. The ratio $(M + Mr)/M$ is the *mass factor* and it increases roughly quadratically [58] with the gear ratio, i.e. it is smaller at higher gears. This means that a faster dynamics of the vehicle is observed at higher gears.

Once a properly defined linear model is derived, a linear controller can be designed according to the procedures discussed in the classical control theory. Of course, when the designed controller will be transfered on the truth model, the added complexity (the so-called *un-modeled dynamics*) of this model compared to the linearized one will require a fine (re-)tuning of the controller's parameters.

As stated previously, the controller selected for realizing this cruise control is a PI controller. A PI controller is a particular case of the well-know and widely-employed PID control technique. As figure 4.18 illustrates, the name of this controller (Proportional-Integral-Derivative) infers from how the *error* signal (difference between the desired target for the process variable (so-called *set point*) and the variable itself) is handled.

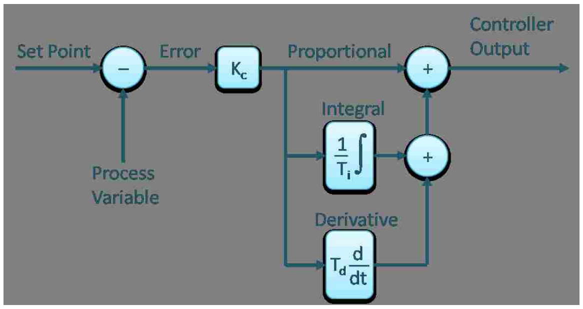


Figure 4.18: *Structure of a PID controller*

In the proportional path the error is multiplied by a constant K_p, in the integral path the error is multiplied by a constant K_i and then integrated and in the derivative path it is multiplied by K_d and then differentiated. The three results are then summed together to produce the controller output. The K terms are called *gains* and can be adjusted (i.e. *tuned*) to a particular plant with a specific set of requirements. The tuning of these parameters changes how sensitive the system is to each of these different paths.

The error term is, in fact, a *signal*, i.e. a function of a time. The proportional path will then produce an output identical to the error signal, except for the scaling factor K_p. The integral path will, on the other hand, produce an output proportional to the *area* of the error signal. This term is then particularly important to remove constant errors: no matter how small the constant error, eventually the summation of that error will be significant enough to adjust the controller output[5]. The derivative path, finally, will contribute to the

[5] This result can be seen as a simple explanation of the famous *Internal Model Principle*, a fundamental result of the modern control theory. This principle states that, in order to achieve a null steady state error, the loop function $L = PC$ of a control system composed of a plant P and a controller C must have a number of poles at the origin k (i.e. of integrators $\frac{1}{s}$) equal to $m + 1$, where m is the exponent of the polynomial input to the system (e.g. for a step input, $m = 0$ and hence one integrator in either the plant P or in the controller C is enough to *guarantee* a null steady state error).

controller output proportionally to the rate of change of the error signal. All three paths are not always needed, and as a consequence simpler layout can be used, with advantages in terms of implementation, tuning and conceptual complexity. Tables 4.1 summarizes the effect of increasing each gain on the performance of the overall system.

Table 4.1: *Effects of increasing a parameter independently*

Parameter	**Rise time**	**Overshoot**	**Steady-state error**	**Stability**
K_p	Decrease	Increase	Decrease	Degrade
K_i	Decrease	Increase	Eliminate	Degrade
K_d	Minor change	Decrease	No effect in theory	Improve if "small"

A PI control block is readily designed in Simulink and applied to the linearized longitudinal dynamics (figure 4.19).

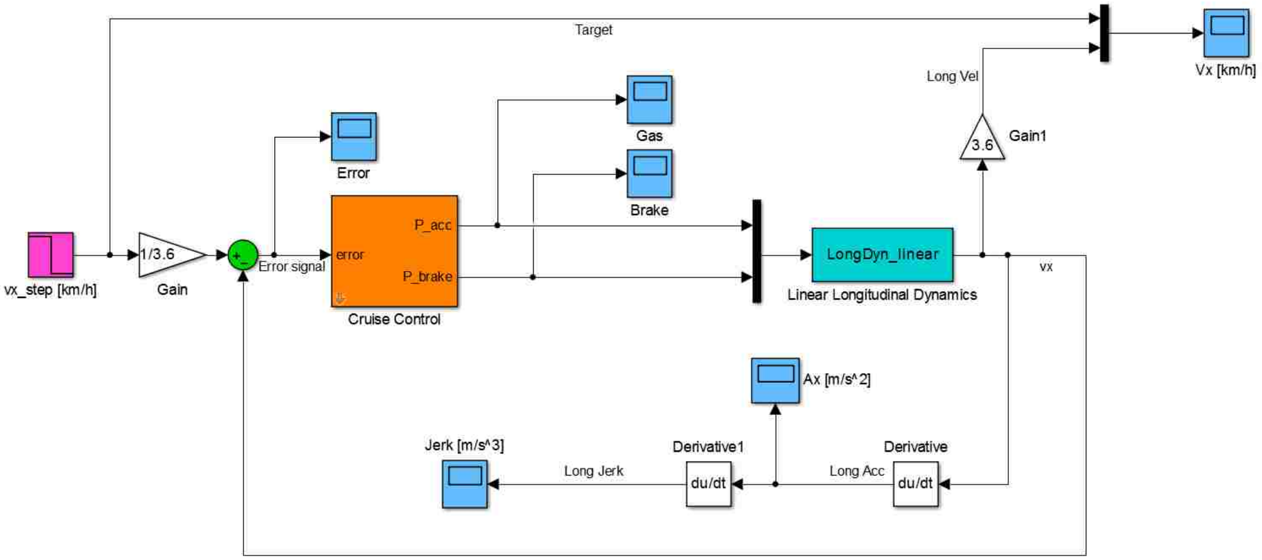


Figure 4.19: *Simulink design of the longitudinal dynamics controller*

The input to the system is the desired vehicle speed (magenta), the output the actual longitudinal speed of the vehicle. The PID gains are initially tuned with the Ziegler-Nichols tuning method: the proportional gain K_p is progressively increased up to the "ultimate" value K_u at which the closed-loop system oscillates with a constant amplitude and an oscillation period T_u. The gains are then set according to table 4.2.

Table 4.2: *Ziegler-Nichols' heuristic tuning technique for PID controllers and related variants*

Controller	K_p	K_i	K_d
P	$0.5K_u$	–	–
PI	$0.45K_u$	$1.2K_p/T_u$	–
PD	$0.8K_u$	–	$K_pT_u/8$
PID	$0.6K_u$	$2K_p/T_u$	$K_pT_u/8$

Gains values determined in this way are only a starting point. A trial-and-error procedure is then carried out in order to improve the performances of the system. In particular, limits on the acceleration and the jerk are imposed in order to improve system comfort for the passengers. For the internal model principle the presence of an integrator in the closed loop guarantees for a step input a zero steady state error.

Given the similarity between linearized and truth models, the PI controller requires only marginal re-tuning of its parameters to keep the same level of performance when applied on the actual system. Performance results will be shown in the next chapters.

4.5.2 Lateral Control

What was shown as the design procedure for the longitudinal controller holds also in the case of the lateral controller. Some particular considerations must nonetheless be made due to the larger complexity of the lateral dynamics subsystem.

The design of a lateral controller is carried out by first tuning the system on a linearized model of the interested dynamics. The model of the LRC presented in the previous chapter is — if parametrized according to v_x — already linear and hence it does not require modifications. On the contrary, particular attention must be devoted to the modeling of the lateral dynamics and, above all, of the EPS actuator.

In the literature several examples of combined lateral dynamics-LRC linear models can be found [115, 116], and here only the final result is reported:

$$\left\{\begin{array}{c}\dot{x}_1\\ \dot{x}_2\\ \dot{x}_3\\ \dot{x}_4\end{array}\right\}=\left\{\begin{array}{c}\dot{v}_y\\ \ddot{\psi}\\ \dot{q}\\ \dot{m}\end{array}\right\}=[A]\left\{\begin{array}{c}v_y\\ \dot{\psi}\\ q\\ m\end{array}\right\}+[B]\left\{\begin{array}{c}\delta_{steer}\\ K_l\end{array}\right\} \tag{4.5.1}$$

$$\left\{\begin{array}{c}y_1\\ y_2\end{array}\right\}=\left\{\begin{array}{c}y_{fb}\\ q\end{array}\right\}=[C]\left\{\begin{array}{c}v_y\\ \dot{\psi}\\ q\\ m\end{array}\right\}+[D]\left\{\begin{array}{c}\delta_{steer}\\ K_l\end{array}\right\} \tag{4.5.2}$$

where

$$[A]=\left[\begin{array}{cccc}-\frac{C_f+C_r}{Mv_x} & -(v_x+\frac{C_fl_f-C_rl_r}{Mv_x}) & 0 & 0\\ -\frac{C_fl_f-C_rl_r}{J_zv_x} & -\frac{C_fl_f^2+C_rl_r^2}{J_zv_x} & 0 & 0\\ \text{-1} & 0 & 0 & v_x\\ 0 & \text{-1} & 0 & 0\end{array}\right],$$

$$[B]=\left[\begin{array}{cc}\frac{kC_f}{M} & 0\\ \frac{kC_fl_f}{J_z} & 0\\ 0 & -Lv_x\\ 0 & v_x\end{array}\right],$$

$$[C]=\left[\begin{array}{cccc}0 & 0 & 1 & \mathrm{L}\\ 0 & 0 & 1 & 0\end{array}\right],$$

and $[D]$ is a 2x2 null matrix. In this analytical model M is the mass of the vehicle, J_z its inertial moment around center of gravity referred to the vertical axis, C_f and C_r are the cornering stiffnesses of the front and rear axles, respectively, l_f and l_r are the distances of the center of gravity from the front and rear axles, respectively, and k is the front wheels angle/steering-wheel ratio (expressed in $[rad/deg]$). The first two equations of 4.5.1 can be easily identified as the bicycle model, whereas the last two are the usual equations for the linear approximation of the centerline provided by the LRC.

Of course, as an alternative to this 4^{th} order, linear, analytical model, an identified model - as in the case of the longitudinal dynamics - could be used. However, given the excellent accuracy (regression coefficient R equal or above 99%) with which the dynamic behaviour of the vehicle is reproduced in the region of interest, there is no reason to do that.

The analysis of the eigenvalues of matrix $[A]$ highlights the instability of the system: two integrators (i.e. poles at the origin) can be found in the transfer functions from both inputs δ_{steer} and K_l to the outputs y_{fb} and q. Again, then, feedback control is required for guaranteeing the stability of the lateral dynamic motion of the vehicle.

The problem with the model introduced here is that the variable the controller will act upon is not the steering wheel angle, but the torque applied to the steering column. It is then necessary to provide a linear model for the EPS actuator, too. This is readily accomplished by identifying with the System Identification Toolbox a linear, second-order state space model starting from data coming from simulations of the truth model. Given the strong non-linearity of the latter, only a small region of the system operation could be accurately modeled. As a consequence the linear model is optimized only for the most common operating scenarios in our system (60 to 130 km/h speed range and 2 to 5 Nm torque range).

Control From Future Vehicle Position

In order to simplify the discussion on the lateral controller, let us first discuss its implementation on the linearized lateral dynamics model discussed above.

As underlined multiple times, two procedures exist in order to realize a feedback loop controller acting on the vehicle's lateral dynamics:

1. Exploit the *future* vehicle position y_{fb} to direct the steering action;

2. Use the *current* vehicle position q to implement a suitable steering control.

In order to take advantage of the camera output and to mimic driver's behaviour [117], the basic approach that will be followed is the first one, known as look-ahead control.

Now, an important consideration has to be made. While y_{fb} (i.e. $q+mL$) is the preferred variable to serve as a starting point for the control action, it is generally q which imposes a constraint on the system. In fact, while y_{fb} can assume large instantaneous values without affecting the quality of the control, q must be *necessarily* kept close to zero. Failure to do so will result in the risk of crossing the boundaries of the lane with potentially catastrophic results. American and European codes generally prescribe lane widths ranging from 2.7 to 3.6 m [118]. In the worst case scenario, considering a vehicle width of 1.8 m, this leaves to the driver roughly 0.4 m per side of displacement with respect to the lane centerline before a wheel could exit the lane. A general guideline could then be that of halving this value: $|q| \leq 0.2\ m$ in all conditions.

If the variable used for feedback (i.e. the variable the controller will force to be zero) is y_{fb} we can predict the value that q will assume in the following two conditions:

- Steady state motion on a *straight* road.

  We mentioned in the section dedicated to the LRC that in steady state condition (i.e. when q, m and hence y_{fb} are constant) equations 3 and 4 in 4.5.1 simplify to $m \approx \dfrac{L}{R}$. On a straight road $R \rightarrow \infty$ and thus $m \rightarrow 0$. In such condition $q = y_{fb} - mL \approx 0 - 0 = 0$, which is the desired outcome;

- Steady state motion in a *curve*.

  Similar considerations hold and they lead to $q = y_{fb} - mL \approx 0 - mL = -\dfrac{L^2}{R}$. Assuming $L = 11.5\ m$[6] and $R = 200\ m$, q would be approximately $-0.66\ m$ (result confirmed in simulations).

This means that in a curve of $R = 200\ m$, if feedback is performed from y_{fb} the only way to achieve $|q| \leq 0.2\ m$ is to use $L \leq 6.3\ m$: while this is theoretically possible, such a small value of L would most certainly negatively impact driving comfort. This consideration rules out the possibility to realize a simple controller from y_{fb}, similarly to what was done for

[6] This value of L is selected because it guarantees a good trade-off between performance and comfort requirements.

longitudinal dynamics. The approach that was followed to meet the requirements is inspired by the LTR (Loop Transfer Recovery).

Loop Transfer Recovery

According to this approach the output of the camera y_{fb} is selected as the output from which feedback is performed, but this variable is first used to reconstruct the parameter q which is then minimized by the controller. Traditionally this is accomplished by designing a Kalman Filter (KF) for state estimation and then a Linear Quadratic Regulator (LQR) for control (LQG design approach).

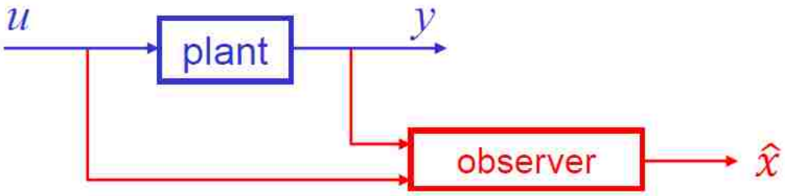


Figure 4.20: *Closed-loop observer scheme*

For simplicity let us first consider the system without the EPS actuator. This system is conveniently described by the A, B, C, D matrices previously introduced and it plays the role of the plant (figure 4.20) whose state must be reconstructed for control purposes. In this work, the implementation of the observer was done according to Luenberger's theory: an LTI dynamic system with inputs equal to the plant's input u and output y is built in order to obtain an estimate ($\hat{x}$) of the plant's state. If A, B, C, D are the matrices of the plant, then the differential equation describing the observer is:

$$\dot{\hat{x}} = (A - LC)\hat{x} + Ly + (B - LD)u. \tag{4.5.3}$$

In equation 4.5.3 the matrix L[7] is chosen according to a traditional eigenvalue placement technique in such a way as to stabilize the dynamics of the error $e = x - \hat{x}$ and to bring its value rapidly to zero.

The reason why a Luenberger observer is designed is that now for control purposes both a complete estimate of the states describing vehicle lateral dynamics and an estimate of

[7] Not to be confused with the *scalar* L, the look-ahead distance.

the parameter q (i.e. the third state x_3) are made available. Both these quantities will be employed in the design of an LQI controller.

An LQI controller is an extension of a standard LQR control. LQR control is a well-known, linear control technique which allows one to define the control law considering an optimality criteria. A cost functional accounting for both the energy (i.e. the L_2 norm) of the states x and of the input u is created:

$$J(u,x) = ||x||^2_{Q,2} + ||u||^2_{R,2} = \int_0^{\infty} x(t)^T Q x(t)\, dt + \int_0^{\infty} u(t)^T R u(t)\, dt$$

where matrices Q and R are selected so that to determine the best trade-off between performances (minimization of $||x||^2$) and command activity (minimization of $||u||^2$). A third contribution $\int_0^{\infty} 2x(t)^T N u(t)\, dt$ can also be added as an additional degree of freedom. Closed-loop *stability* is ensured by the minimization of $||x||^2$. The solution of this optimization problem is the control law $u^*(t) = -Kx(t)$, i.e. state feedback. The matrix K is determined by solving the Algebraic Riccati Equation, which is automatically done by MATLAB.

Now, the LQI control technique retains the same structure with a change: an additional state is obtained by integrating the tracking error (difference between the desired q and the actual q) and added to the system. Once this *augmented system* is built, the matrix K defining the state feedback controller can be computed in the same way. The resulting structure can be found in figure 4.21.

As can be seen, the Luenberger observer is used for reconstructing the four states of the linear model of the lateral dynamics and LRC, and providing an accurate estimate of the parameter q (which is nothing but one of these states). This information is then fed back, subtracted from the reference ($q = 0$) and integrated in order to build the fifth state used for control. The reason why the integral of the tracking error is added as one of the states of the feedback-loop is that in this way constant errors in steady-state can be eliminated. Results pertaining this system will be shown in chapter 6.

Once the matrices Q, R and N have been selected, the model is expanded by adding the identified, linear EPS model. In order to retain the LQI structure, another Luenberger observer is added to reconstruct the states of the EPS model. The computation of the matrix K is now repeated for the overall system given by the series connection of the EPS model and the lateral dynamics (and LRC) model. As a consequence a total of seven states is now considered: four from the model of lateral dynamics, two from that of the EPS and one from the integration of the tracking error.

As with longitudinal dynamics, once proper tuning is achieved on the linearized model, the controller is implemented on the non-linear system ("truth model"). Unavoidably a slight decrease in performances is to be expected, even if something can be recovered with proper fine-tuning. Again, more detailed representations of performance and stability results will be given in the proper section of chapter 6.

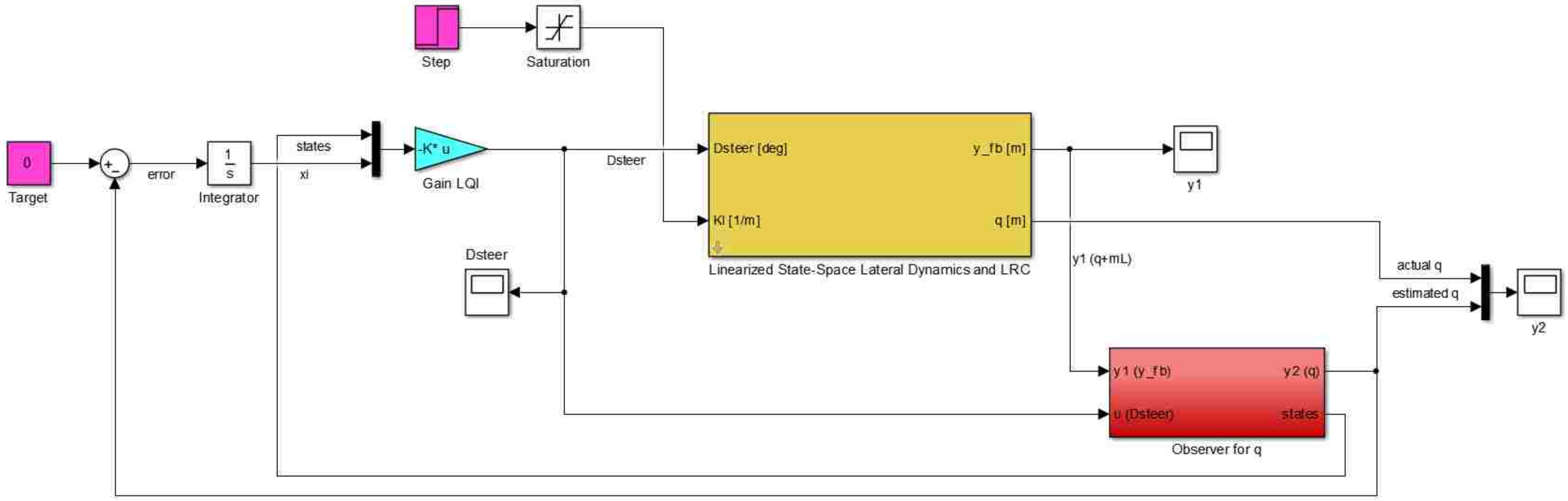


Figure 4.21: *LQI control implementation on the linearized system (without EPS)*

4.6 Virtual Sensor Design

In chapter 2 the mathematical treatment of virtual sensors was considered. In this section the set of data and the procedure used to build this component are reported. Finally the implementation of this block in the overall model is discussed.

4.6.1 Used Data

Let us first consider the data necessary for the identification of the virtual sensor. Considering the objective of reconstructing the *precise vehicle position* starting from the information provided by the standard GPS, it is clear that first and foremost it will be fundamental to record simultaneously these two quantities. As a consequence, testing on a vehicle equipped with *both* standard and differential GPS should be conducted. Furthermore, vehicle data (in particular, velocity v_x, longitudinal and lateral acceleration a_x and a_y, steering angle δ_{steer} and yaw rate $\dot{\psi}$) must be available to have sufficient inputs for the non-linear identification process that allows the creation of the virtual sensor block. These signals are, in fact, essential to provide the additional information about the vehicle dynamics necessary to correct and improve the position provided by the standard GPS.

Considering this necessity, CRF made available the following data sets:

1. GPS acquisitions performed on a vehicle equipped with both differential[8] and standard positioning system (figure 4.22);
2. Vehicle simulations of the 2015 Fiat *500X* with complete data recordings.

Both data sets are needed for the following reason. The first set provides GPS acquisitions from a test on the "Centro Sicurezza" track in Orbassano (Turin) and hence satisfies the requirement of having standard and differential GPS data recorded simultaneously. While position data from this test are available, no information considering the dynamics of the vehicle was recorded. The second set, on the contrary, consists of full vehicle data generated through accurate simulations of the vehicle model in the CarMaker software package. In this second set, the vehicle closely follows a precise trajectory (corresponding

[8] Differential GPS data are transmitted trough VSAT technology.

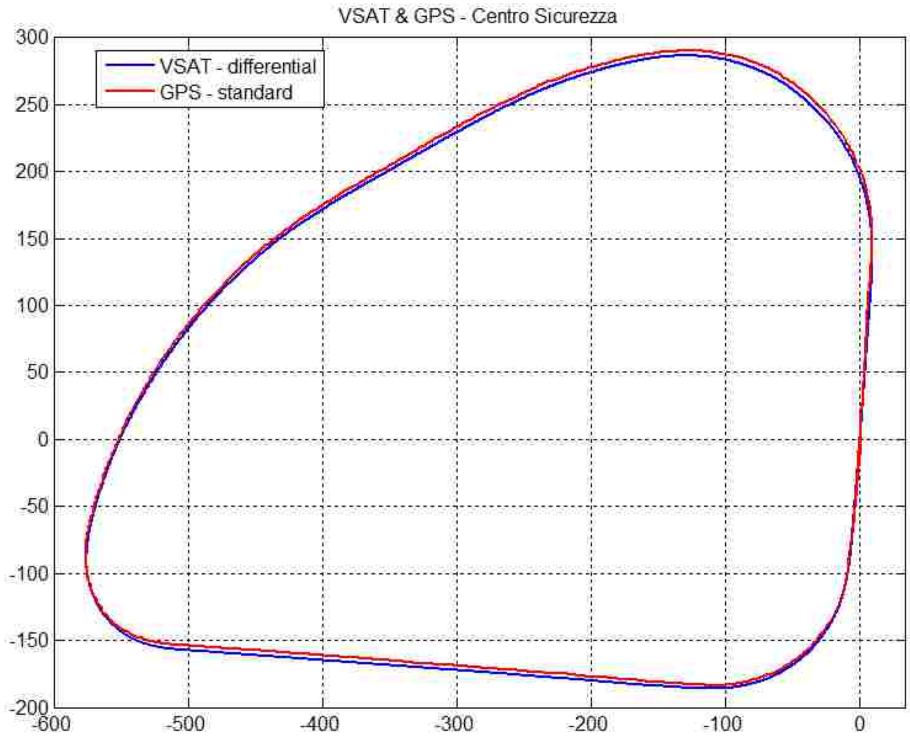


Figure 4.22: Differential GPS (VSAT, blue) and standard GPS (red) acquisitions made at the "Centro Sicurezza" track in Orbassano. X and Y axes dimensions are expressed in meters

again to the "Centro Sicurezza" track) that can be considered the equivalent of the precise vehicle position (i.e. the signal corresponding to the differential GPS), but no standard GPS data is available. As explained in the following, the complementarity between these two sets of data will be exploited to identify the virtual sensor.

4.6.2 Procedure

The procedure to be followed in order to develop the Virtual Sensor (VS) requires that one first elaborates available data and then identifies a suitable non-linear relationship between input and output data.

Data Elaboration

Track acquisitions consist of longitude and latitude measurements provided by differential and standard GPS. The first elaboration that should then be performed is to transform longitude and latitude coordinates in the trajectory followed in the X-Y plane. This transformation is performed in order to obtain more easily interpreted data.

The conversion is readily accomplished by first converting the “degrees.minutes” format in degrees and then by applying the following equations:

$$\begin{aligned} X &= (long - long_0)\, K\, cos(lat) \\ Y &= (lat - lat_0)\, K \end{aligned} \tag{4.6.1}$$

where $long$ and lat are the longitude and latitude in degrees, respectively, and $K = 111317.1\ m/deg$ is the coefficient used to convert degrees in meters assuming a spheric shape for the earth. The parameters $long_0$ and lat_0 represent the origin selected for the X and Y trajectories (the same origin is exploited for all acquisition data, see figure 4.22).

Once GPS trajectories have been converted, they can be used in order to create data for the development of the virtual sensor. The identification of the virtual sensor will be performed considering the second set mentioned in section 4.6.1. This is of course linked to the necessity of having suitable vehicle data for the identification itself. The lack of standard GPS signal in this set is overcome by extracting from the first set a model linking differential and standard GPS.

It is in fact possible to compute the instantaneous difference between the two GPS signals in the first set. This difference is then summed to the trajectory followed by the vehicle in the CarMaker simulations (considered equivalent to the position signal as provided by the differential GPS) in order to obtain the corresponding position provided by a standard GPS. Particular attention must be given to the correct sampling of the considered signals and to the matching between their starting positions. Once this operation is complete, it is now made available a full set of vehicle and position signals to be used for the identification of the virtual sensor.

Virtual Sensor Identification

As was shown in chapter 2, a VS is built when, given a non-linear system of input u and outputs $[y, z]$, it is of interest to determine z_t for $t > T_m$, considering that noise corrupted measurements of $\tilde{u}_t, \tilde{y}_t$ are available for all times t and that z_t is measured only for $t < T_m$.

As already discussed, a functional F_0 and integers n_u, n_y must exist such that the variable of interest z_t may be calculated as:

$$\tilde{z}_t = F_0(\tilde{Y}_t, \tilde{U}_t) + d_t, \quad t = 0, 1, 2, ..., T_m$$
$$\tilde{Y}_t = [\tilde{y}_t, \tilde{y}_{t-1}, ..., \tilde{y}_{t-ny+1}]$$
$$\tilde{U}_t = [\tilde{u}_t, \tilde{u}_{t-1}, ..., \tilde{u}_{t-nu+1}]$$

As a result, the problem of developing the virtual sensor becomes the problem of *estimating* a functional $F_A(Y, U)$ as a parametric approximation of F_0, computed using any desired nonlinear identification method.

Now, considering the problem at hand, the quantity z is represented by the precise vehicle position provided by a differential GPS. The inputs u which can be used to reconstruct z are:

- Vehicle position as provided by a standard GPS;
- Vehicle dynamics data:
  - Velocity v_x;
  - Yaw rate $\dot{\psi}$;
  - Steering angle δ_{steer};
  - Longitudinal and lateral accelerations a_x and a_y.

The technique that will be employed in order to perform this identification is the non-linear ARX, which proved successful in the development of a model for the EPS actuator. As already discussed, the *nARX* structure can be expressed as follows:

$$y(t) = f(y(t-1), y(t-2), \ldots, y(t-n_a), u(t-n_k), u(t-1-n_k), \ldots, u(t-n_b-n_k+1))$$

The variables that can be managed to improve the fit on experimental data are then the numbers of past values of inputs and outputs (orders n_a and n_b), the delays n_k and the non-linearity f.

Considering that the information provided by the GPS is converted in X-Y format before being processed, the VS block will be built by identifying *two* non-linear models (one for the X coordinate and one for the Y coordinate) in the System Identification Toolbox.

A large number of different configurations are tested in the attempt to get the highest level of precision for the sensor. The best results are achieved by means of the following two expressions:

$$X_{VSAT} = f(X_{GPS}, v_x, \dot{\psi}, a_x) \tag{4.6.2a}$$

$$Y_{VSAT} = f(Y_{GPS}, v_x, \dot{\psi}, \delta_{steer}) \tag{4.6.2b}$$

where the subscript $VSAT$ makes reference to the precise vehicle coordinates provided by the differential GPS and the subscript GPS to the standard GPS coordinates. The reason why in equation 4.6.2 the lateral acceleration a_y is not used is that numerical errors are present in this signal. This does not represent a problem because with vehicle speed, yaw rate and steering angle a full picture of the lateral dynamics is obtained anyway.

The non-linearity f is a Wavelet Network and the orders employed are in the range 2 to 3 for all variables involved.

Now, as already discussed, the objective of the virtual sensor is to reconstruct the position of the vehicle in the lane when the output of the LRC is degraded or unavailable. This device is then intended to provide the precise vehicle position for *short* distances, in such a way as to allow the control systems to continue their operation waiting for the signal from the LRC to be restored or the driver to intervene. As a consequence, the VS is identified considering short portions of the test track. Results pertaining the identification and implementation of the virtual sensor are discussed in chapter 6.

4.6.3 Implementation

In order to easily implement the virtual sensor in the overall model, two already existing blocks are exploited. Figure 4.23 shows how the Virtual Sensor block (orange) is added to the existing model. The VS takes as input the position provided by the GPS in X and Y form (magenta) as well as vehicle data necessary for the two $nARX$ models embedded in the virtual sensor, and gives as output the corrected position.

The precise vehicle position is then elaborated by the Input Coordinate Transformation block (dark green, see chapter 5) which converts the X and Y coordinate in vehicle speed v_x and road curvature K_l. The vehicle speed is then sent to the longitudinal controller.

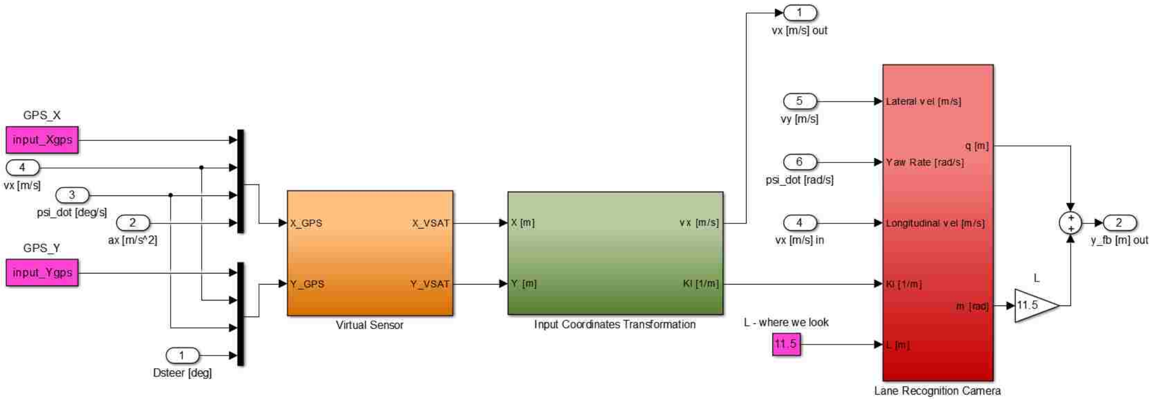


Figure 4.23: *Schematic implementation of the Virtual Sensor block (orange) into the existing model exploiting the Input Coordinate Transformation block (dark green) and the Lane Recognition Camera block (red)*

Now, considering that the lateral controller is designed to work with (an estimate of) the parameter q, the easiest way to implement the virtual sensor into the overall model is to retain the Lane Recognition Camera block (red). However, this *does not* mean that the camera is working. Instead, the LRC block should be considered in this case as nothing more than an algebraic processing unit used to convert the road curvature K_l into an output (the distance $y_f b$, and hence the parameter q) that the lateral controller can use in order to guide vehicle motion in the lane.

An advantage brought by this kind of implementation is the ease with which it is possible to switch from the LRC-driven to the VS-driven system. This can be readily accomplished by implementing in Simulink a time-dependent switch swapping the source of road curvature K_l as necessary.

Chapter 5

Simulation Procedures

In this brief chapter an overview of the simulation and validation procedures used to analyze the system performance is given. Numerical and graphical results coming from the simulations will be provided in chapter 6.

5.1 Simulations Description

The development of this project required the completion of two milestones: first, the design of a suitable control strategy for the autonomous operation of the vehicle and, second, the design of a device — the virtual sensor — able to overcome the loss of data ensuing from a failure or degradation of the camera sensor.

Clearly, the unique contribution of this research comes from the latter part of the project, but it is nonetheless important to quantify the system performances also for the "standard" layout. This is the reason why in the next chapter results will be shown for the following three conditions:

1. Uncontrolled system.

   Considerations on the dynamics of the vehicle under purely human operator control can be made as a benchmark for further discussion;

2. System controlled via LRC.

   Simulations of this system layout will allow one to achieve a double objective: show the need for a suitable controller to manage the unstable dynamics of the car and provide a baseline for comparison with the system equipped with the virtual sensor.

3. System controlled via virtual sensor.

   Testing the performance of the system in which vehicle position is reconstructed by the previously described position observer will illustrate the usefulness of this solution and will serve as basis for the assessment of the value added by the implementation of this system on an automated vehicle.

The testing missions will either represent worst case scenarios in terms of combinations of vehicle speed and road curvature or will correspond to actual road profile such as the CRF proving ground where GPS data have been acquired.

5.2 Model Validation

Given the nature of the designed control systems designed the most important parameters that will be considered in evaluating the performance of the system in the three aforementioned cases are the speed at which the vehicle is traveling and the distance of the vehicle's center of gravity from the centerline of the lane.

Both these parameters are evaluated against their target values (the prescribed longitudinal speed and a null distance from the centerline, respectively). In order to quantify the quality of the fit the regression plot will be often employed. Several examples of its use were presented in chapter 4. Additionally, a validation metric will be used to rapidly quantify the error, i.e. the difference between the expected and the actual values. This metric is considered necessary by W.L. Oberkampf and T.G. Trucano, who, in their paper [119], recognize the importance of numerically quantifying the agreement between the computational and the experimental data. Such metric, V, is computed as:

$$V = 1 - \frac{1}{t_{end} - t_0} \cdot \int_{t_0}^{t_{end}} tanh \left| \frac{N(t) - T(t)}{T(t)} \right| dt \tag{5.2.1}$$

where N represents the experimental data and T the target data.

In equation 5.2.1, because of the linearity of the hyperbolic tangent near the origin, the validation metric V essentially represents the complement to 1 of the average error. A value $V = 1$ would mean exact agreement between numerical and analytical results. When the error increases, the hyperbolic tangent saturates to 1 causing V to approach zero.

Both the regression plot and the validation metric offer interesting information about the data they are applied to. The former quantifies how much better the model is at capturing the trend of data compared to just taking their average. The latter gives a fast way to assess how large the average error is compared to the magnitude of the working data.

5.3 Vehicle Trajectory in the X-Y Plane

As seen, GPS data are presented as X-Y coordinates. Moreover, expressing the position of the vehicle in terms of coordinates with respect to a fixed point helps one to better visualize movements in the lane. It is then convenient to add two blocks to the system in such a way as to change the inputs and the outputs to the X and Y coordinates when needed.

5.3.1 Input Coordinates Transformation

The inputs of the controlled system are the desired longitudinal vehicle speed (v_x) and the curvature of the road (K_l). In this case it is then of interest to implement a transformation of these two quantities to X and Y .

This is readily accomplished by implementing the two following expressions:

$$v_x = \sqrt{\dot{X}^2 + \dot{Y}^2} \tag{5.3.1a}$$

$$K_l = \frac{\ddot{Y}\dot{X} - \dot{Y}\ddot{X}}{(\dot{X}^2 + \dot{Y}^2)^{3/2}} \tag{5.3.1b}$$

where derivatives are taken with respect to time.

Equation 5.3.1a is intuitive. An explanation behind equation 5.3.1b can be found in [120]. In this way it will be possible to provide to the system the time signals of the X and Y coordinates of the centerline of the lane without modifying existing components.

5.3.2 Output Coordinates Transformation

The designed system is based on body fixed coordinates. This is a solution suitable for control system design, since the controller must use body fixed measurements (m and q) of the position error.

On the other side, in order to better appreciate the trajectory followed by the vehicle, it is convenient to plot its position at each instant in an inertial frame of reference.

Rajamani proposed in [121] a simple way of obtaining the actual position of the vehicle starting from the desired one:

$$\begin{aligned} X &= X_{des} - q\, sin(\psi) \\ Y &= Y_{des} + q\, cos(\psi) \end{aligned} \tag{5.3.2}$$

where X_{des} and Y_{des} are the global coordinates of the point on the road centerline which lies on a line along the lateral axis of the vehicle. Now, in our case, Rajamani's expressions will lead to an approximate result. The reason is that q, because of the simplified model of the camera, is not the distance from the actual lane centerline, but from its linear approximation. Nonetheless, being $R >> L$ in all conditions, it is possible to assume that the error introduced is small.

Chapter 6

Results and Discussion

This chapter reports the results of the simulations of the system. As already pointed out, the chronological order with which the model was developed is going to be followed in the representation of the results. First, some results pertaining to the pure, uncontrolled vehicle dynamics will be given in order to show the need for control. Then, the implementation of the longitudinal and lateral controllers will offer the possibility to show the degree of performance reached with the simple, linear controllers discussed in chapter 4.

Finally, a comparison will be drawn between the system controlled using the output of the Lane Recognition Camera and that based on the position reconstruction provided by the Virtual Sensor.

6.1 Uncontrolled Model Performance

When the linear model for the lateral vehicle dynamics and LRC was introduced, it was mentioned that the system was unstable with respect to both the steering angle δ_{steer} and the road curvature K_l.

This result can also be easily confirmed for the non-linear, "truth" model by providing suitable step inputs. Figure 6.1 shows how a small step applied to the steering angle (2 *deg* at 30 *s*) produces a diverging response of the distance between vehicle's C.G. and road

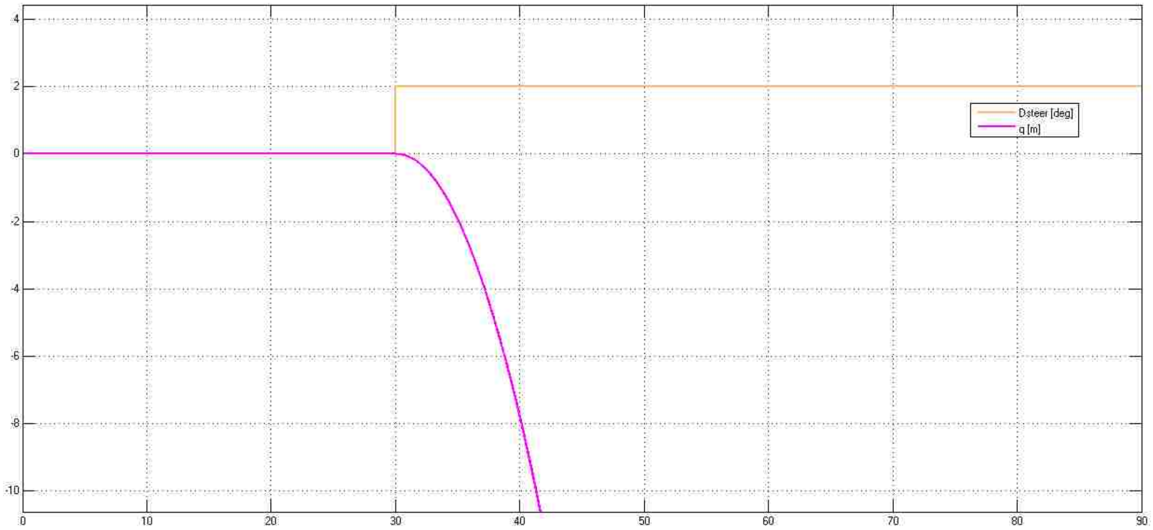


Figure 6.1: *Effect of a step steering angle input (orange) on the distance from the centerline (magenta) as a function of time* $[s]$

centerline.

This effect is even more pronounced when considering the effect of the road curvature. Figure 6.2 clearly illustrates that a change in the curvature of the road determines an

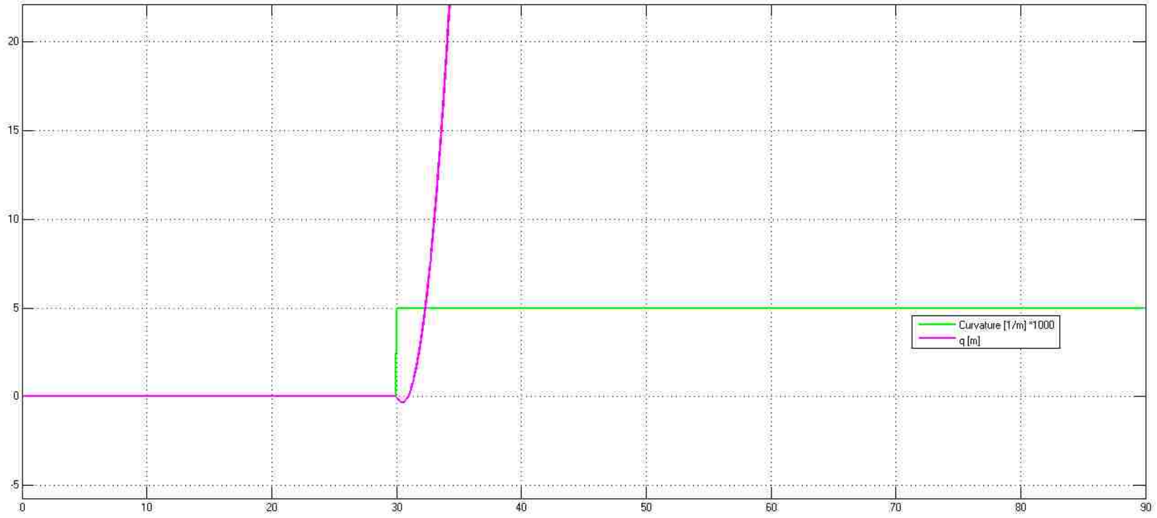


Figure 6.2: *Effect of a step steering road curvature (green) on the distance from the centerline (magenta) as a function of time* $[s]$

unstable response of the parameter q, which rapidly diverges from zero.

These results confirm what was determined analytically by analyzing the eigenvalues of the linearized model and underline the necessity of a controller to stabilize the plant. The stabilizing task is typically performed by the human driver, who, on the basis of the instantaneous variation of the look-ahead position y_{fb}, corrects the value of the steering angle in order to keep the vehicle centered on the lane. This behaviour will be imitated by the lateral controller. If this device is designed using fuzzy control techniques (such as in [60]), then the controller will copy closely the reasoning of the driver. In the case presented here, in which the knowledge of the plant is complete, better performances can be achieved by using traditional control techniques which exploit existing information on the system to be controlled.

6.2 Controlled Model with LRC

Let us now consider the performance of the controlled system. The information of the vehicle position in the lane comes from the LRC sensor. For simplicity, the effect of the longitudinal controller is evaluated first.

6.2.1 Longitudinal Control

The instability of uncontrolled longitudinal dynamics is illustrated by applying the Nyquist Criterion[1]. The application of this criterion to the linearized longitudinal model shows that the closed loop system in which the output (v_x) is fed back and subtracted to the reference in order to obtain the input of the longitudinal dynamics, is unstable ($Z = 1$ for the accelerator input, $Z = 2$ for the brake input). The design of a suitable PI control for both inputs stabilizes the closed loop plant ($Z = 0$ for both input pedals). Moreover, given the already cited Internal Model Principle, the integral path will ensure null steady state error.

Let us now see how this translates graphically. Figure 6.3 shows how the actual speed of the vehicle tracks the target value as a function of time. The limits imposed on the

[1] The Nyquist Criterion states that the number Z of poles of the closed loop function $T(s)$ with real part ≥ 0 is given by the sum of the number N of clockwise encirclements around $(-1, 0)$ of the Nyquist diagram of the open loop function $L(s)$ plus the number P of poles of $L(s)$ with real part > 0.

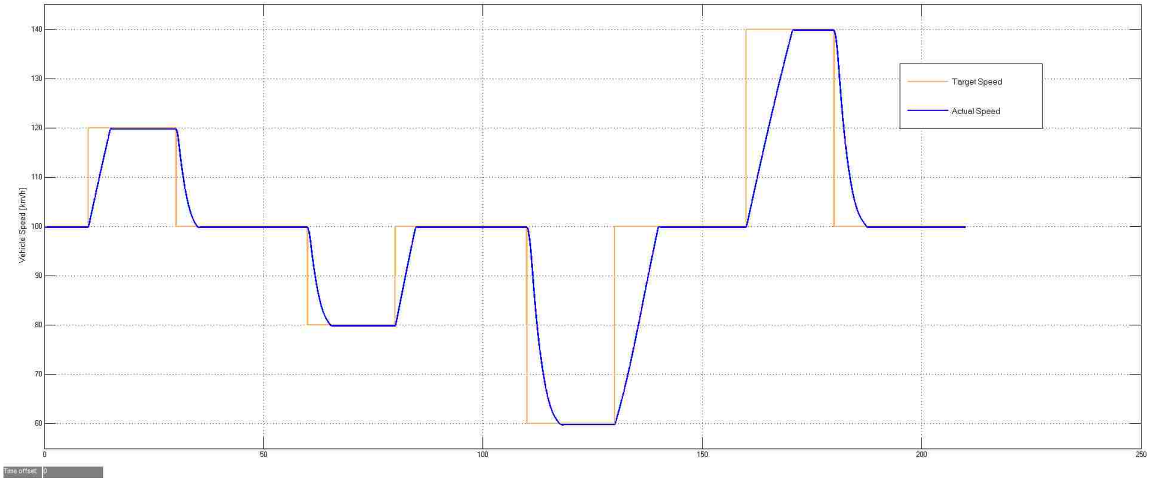


Figure 6.3: *Comparison between targeted longitudinal speed (orange) and actual speed (blue) as a function of time* $[s]$

derivatives of the speed result in a smooth operation of the pedals and a comfortable ride (maximum acceleration $1.5\,m/s^2$, maximum deceleration $-3\,m/s^2$).

The system, as designed here, meets the minimum requirements for testing the overall layout with and without the operation of the position observer. Thanks to the modularity of the system, however, the cruise control presented here can be easily replaced by a more sophisticated unit embedding ADAS functions. For instance, an ACC system could be implemented that regulates the speed in order to keep a safe distance from the vehicle ahead.

6.2.2 Lateral Control

The need of a lateral controller was already highlighted by the necessity of stabilizing the dynamics of the vehicle. Stabilization, however, is not enough: the required system performance is very high and must be met even in worst case scenarios. Safety concerns, in fact, require the vehicle to stay inside the limits of the lane in all operating conditions. Failure to do so could mean that the vehicle is entering the lane of incoming traffic or exiting from the tarmac.

Let us first consider the performances of the first implementation of the LQI controller on the linearized lateral dynamics. In figure 6.4 an abrupt change in road curvature at

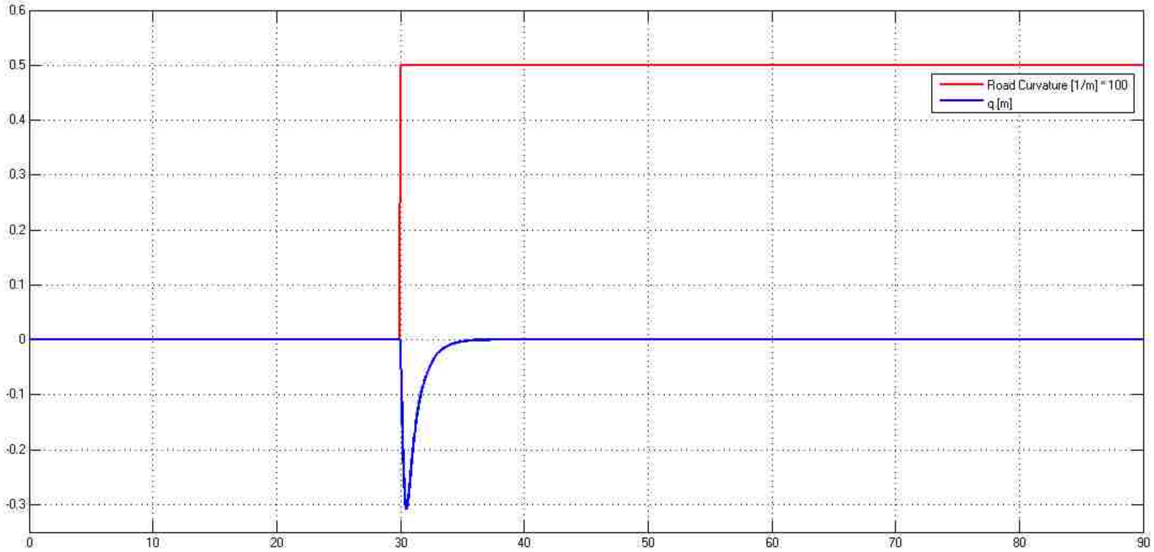


Figure 6.4: *Effect of a step road curvature (red) on the distance from the centerline (blue) as a function of time* [*s*]*; linearized model, no EPS actuator*

$t = 30\,s$ occurs. The speed imposed here is $90\,km/h$ and the curve radius is $200\,m$: a rather demanding condition is then considered. The blue signal illustrates the variation of the distance q between vehicle center of gravity and road centerline. Despite the harsh driving condition, as it can be seen, the distance from the road centerline never overcomes $0.3\,m$ before being smoothly brought to zero in steady state.

This results confirms the quality of the implemented LQI control, compared to the use of a PID control fed directly by the future vehicle position y_{fb} (let us recall that that scheme cannot reduce q below $0.66\,m$ in the considered situation). A more aggressive tuning of the matrices Q and R allows to further reduce the maximum value of q to just above $20\,cm$. The drawback is in that case a less smooth operation of the steering wheel.

Now, when the lateral vehicle model is replaced by the actual non-linear model and, even more so, when the EPS model is added, it is expected some amount of degradation of the performances of the control system. On the other hand, there are no worries regarding robustness and stability:

- *Stability* of lateral control.

The Separation Principle holds, i.e. if the controller *and* the observer are stable when considered alone (which is, as said, an automatic result for the LQR controller and a consequence of the chosen eigenvalues for the observer), then also the overall system is stable;

- *Robustness*[2] of lateral control.

  It is known from classical control theory that the Linear Quadratic Regulator (when combined, as in this case as it will be shown, with a “fast” observer) is robust.

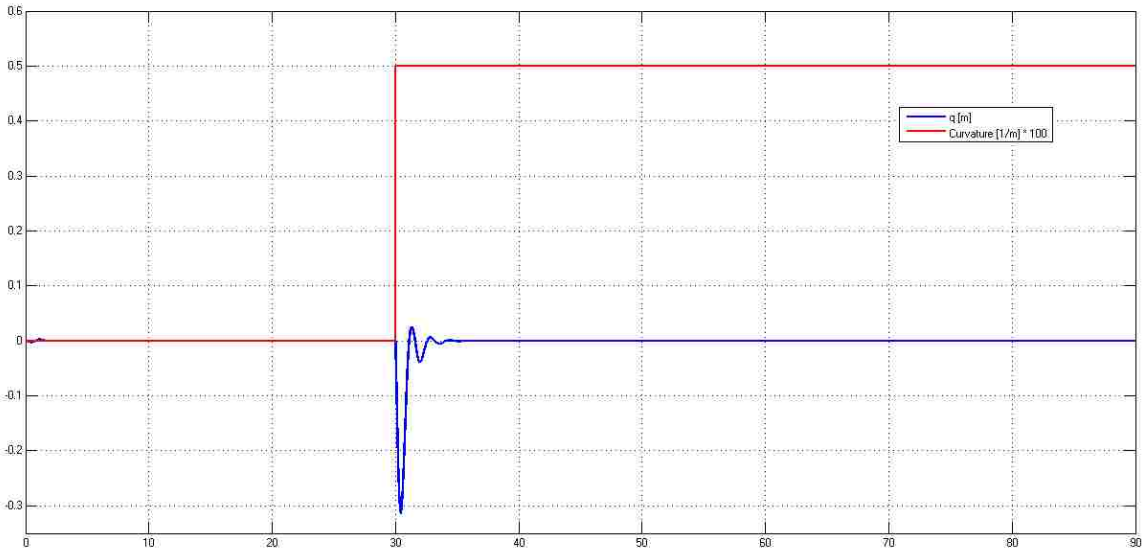


Figure 6.5: *Effect of a step road curvature (red) on the distance from the centerline (blue); truth model and EPS actuator*

In figure 6.5 the same vehicle speed of 90 km/h and the same curve radius of 200 m are considered for the actual vehicle model. As it can be seen, with a fine tuning of the parameters of the LQI controller, the same level of performance expressed in figure 6.4 can be achieved. A certain amount of oscillation is present but their effect is barely noticeable on the trace of the lateral acceleration.

Figure 6.6 shows the variation of the steering wheel torque T_{dr} referred to the aforementioned maneuver in the time range 28 to 38 s. As can be seen, the torque (and hence the

[2] Robustness is the property of a controller by which it can control all the models of the actual plant inside a specific uncertainty set.

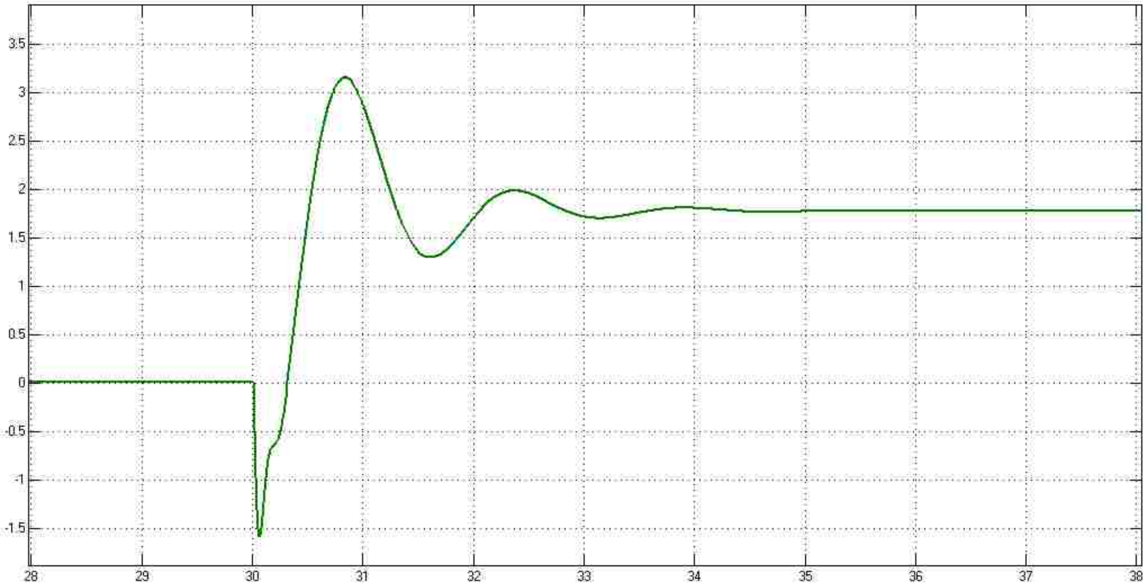


Figure 6.6: *Steering wheel torque T_{dr} as a function of time*

steering wheel angle) varies opposite to the expected direction for a brief amount of time. After that the torque approaches the steady state value with an overshoot that is rapidly damped down.

What has been presented here allows one to see the variation of the controlled variable (q) as a function of time. Let us recall that the LQI controller acts on an *estimate* of q provided by the relative Luenberger observer. Figure 6.7 shows that the estimate and the actual value of q are always very close. The state observer can then be considered "fast" enough for control purposes and the LQI controller can be considered robust.

Trajectory in the X-Y Plane

In chapter 5 the possibility of graphically visualizing the motion of the vehicle in the X-Y plane (i.e. from the top) was introduced. By implementing the equations reported in that chapter we can plot the desired path of the vehicle C.G. versus its actual trajectory.

In figure 6.8 the trajectory that corresponds to a sudden curvature variation from 0 to $\frac{1}{200}\,m^{-1}$ is plotted in blue. In the same figure, the red trace corresponds to the actual path followed by the vehicle. The two signals appear well superimposed. The red box is used to represent the dimension of the vehicle and allows one to better understand the dimension

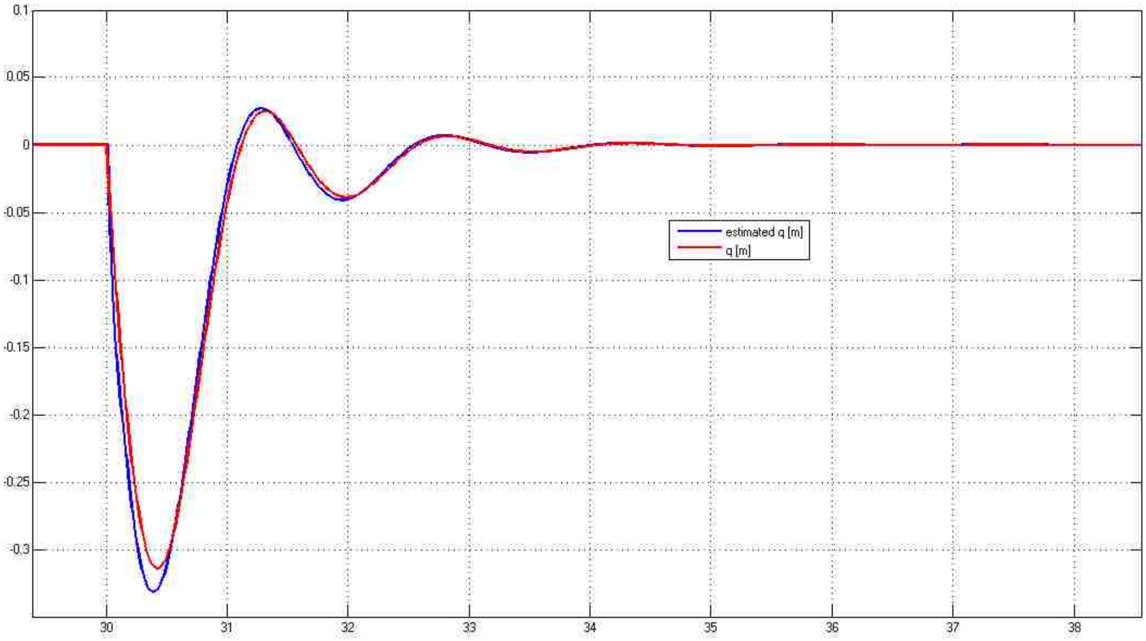


Figure 6.7: *Comparison between the actual value of q as provided by the LRC model (red) and the value estimated by the state observer (blue) as a function of time*

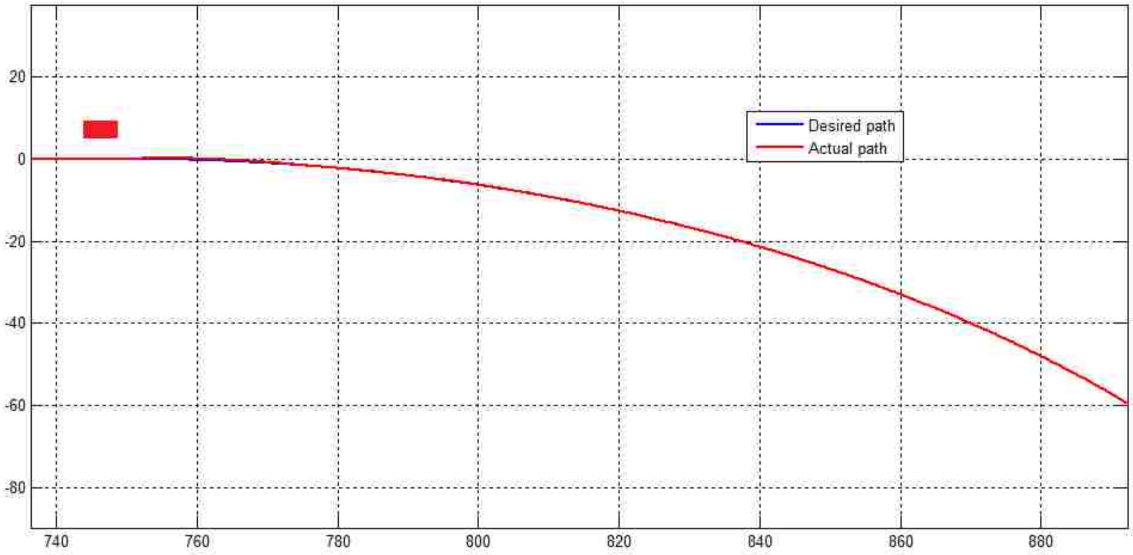


Figure 6.8: *Desired trajectory (blue) and actual vehicle path (red) in the X-Y plane*

of the curve.

The most critical area of the curve is clearly the sharp transition between the straight and the curved part of the road. A zoomed image of the first 30 m (in the X direction) is shown in figure 6.9.

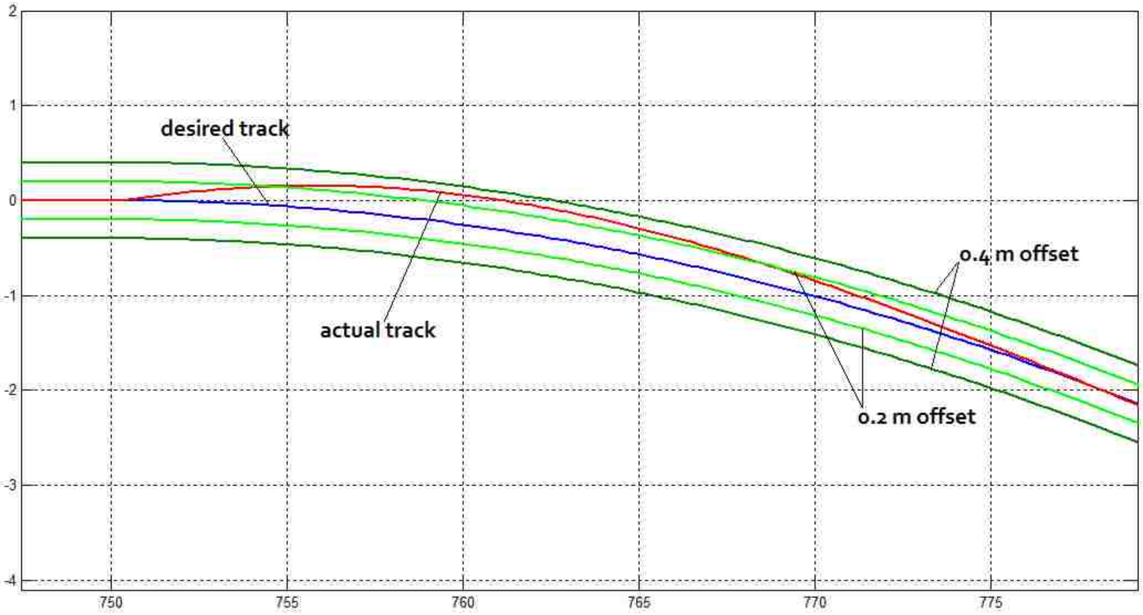


Figure 6.9: *Vehicle trajectory in X-Y plane (90 km/h). Blue: desired track for vehicle's C.G. Light green: $\pm 0.2\,m$ offset from the desired track. Dark green: $\pm 0.4\,m$ offset from the desired track. Red: actual path followed by the C.G. of the vehicle*

As it can be seen, the vehicle does not immediately track the desired path but even steers away briefly from it (see torque plot in figure 6.6). After this first brief phase, however, the controller will provide a large amount of steering angle which brings the actual trajectory closer to the requested one. After about three seconds from the start of the change in curvature the tracking error is reduced to zero and kept that way for the remainder of the curve.

It was mentioned in chapter 4 that, in the worst case scenario, the center of gravity must not displace more than roughly $\pm 0.4\,m$ from the centerline, otherwise the vehicle could face the risk of having one or more wheels outside the traveling lane.

This requirement appears *satisfied* in a variety of test maneuvers, including high speed ($> 130\ km/h$) cornering with small radius ($< 200\ m$). This range of values is extremely unrealistic in actual driving of the vehicle since they bring the tires to the limit of their grip capabilities and cause very uncomfortable levels of lateral acceleration (almost 1 g).

With regard to the second target of keeping the tracking error below $\pm 0.2\ m$ (which means a safety factor of *at least* 2 in all conditions), a standard tuning of the LQI controller does not appear to be able to satisfy it in demanding conditions. On the other hand, when more reasonable level of speeds and of curvature radius are considered, the requirement is easily satisfied: figure 6.10.

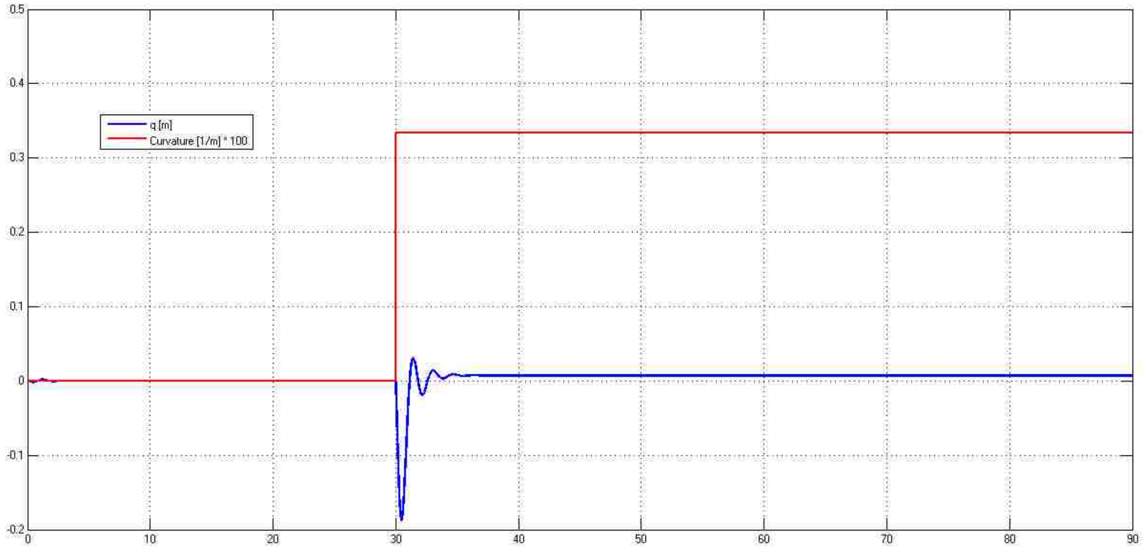


Figure 6.10: *Evolution of position error q considering a speed of $70\,km/h$ and a radius of curvature of $300\,m$*

In order to understand better this point let us make reference to figure 6.11, in which the values of q are plotted as a function of v_x and R. As expected, the higher the vehicle speed v_x the larger the minimum radius of curvature R that can be followed keeping the magnitude of q below $0.2\ m$. At 60 km/h a radius of 280 m or larger can be managed while keeping $|q| \leq 0.2\ m$. At 130 km/h the same requirement can be respected only for curves of radius of 360 m or larger. Figure 6.12 graphically illustrates the combinations of vehicle speed and road curvature that grant $|q_{max}| \leq 0.2\ m$.

In conclusion, the designed LQI controller appears to provide suitable performances when the considered driving scenarios exclude high speed cornering. Besides the physical limits of how fast and accurately the steering wheel can be maneuvered and how much traction the tires can develop, the reason behind this behaviour is found in the linear

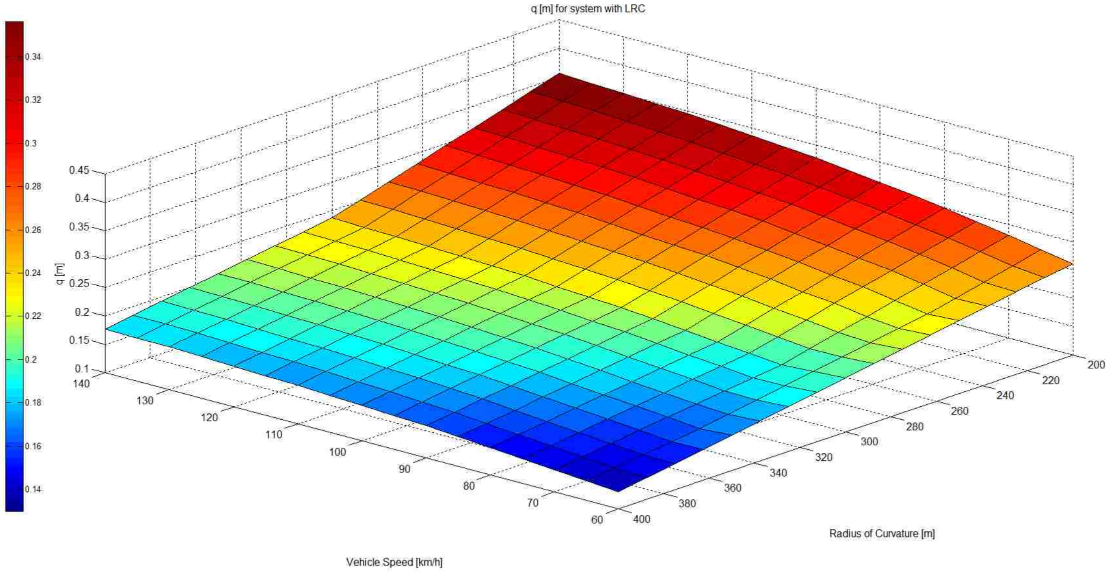


Figure 6.11: *Values of the distance q $[m]$ between vehicle's C.G. and lane centerline as a function of the vehicle longitudinal speed v_x $[km/h]$ and the radius of curvature R $[m]$*

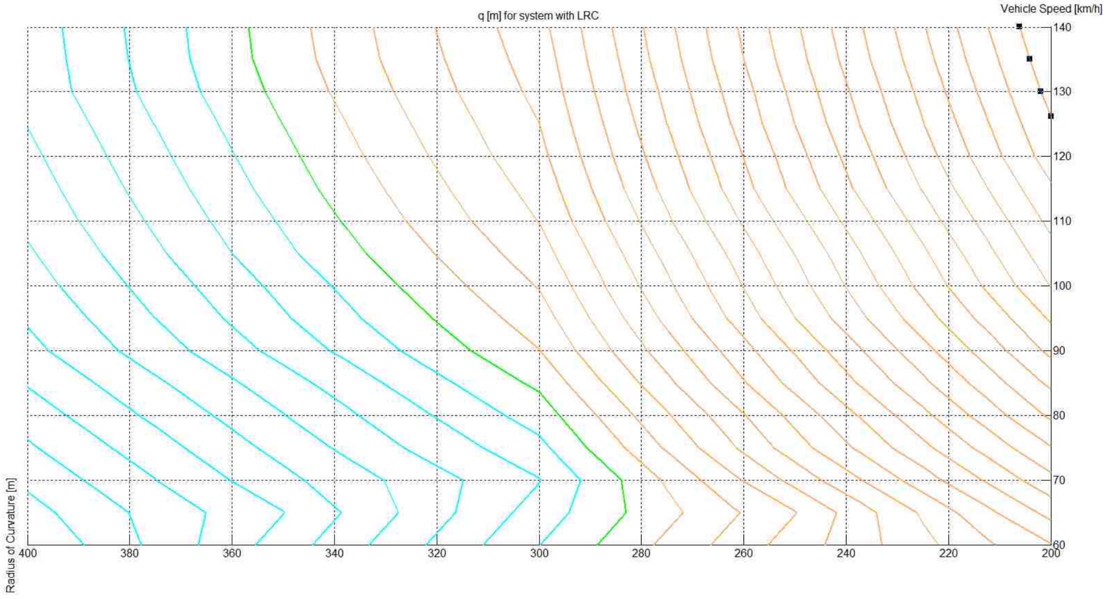


Figure 6.12: *Contour plot of q $[m]$ as a function of v_x $[km/h]$ (ordinate) and R $[m]$ (abscissa). The green line represents $|q_{max}| = 0.2$ m; the blue lines $|q_{max}| < 0.2$ m; the orange lines $|q_{max}| > 0.2$ m*

nature of the employed controller. When the tires, in fact, approach large slip angles, their characteristics depart from the linear region and the simplified model becomes more and more different from the “truth” model. At the same time previously made assumptions (e.g. $tan(m) \approx m$) become less and less realistic. All these factors contribute to deteriorate control system performance in high speed cornering scenarios. Nonetheless, even when “extreme” situations are considered (speeds above $130\, km/h$, curvature radius below $200\, m$ and lane width of $2.7\, m$), the difference between the lane centerline and the path followed by the vehicle’s center of gravity never surpasses $0.4\, m$, preventing vehicle departure from the tarmac and potentially hazardous situations.

Let us conclude this part with a remark on the robustness of the system. In chapter 4 it was mentioned how the vehicle dynamics block can be influenced by wind speed and road slope. By changing these parameters, their influence on the overall system can be studied and the result is that, as long as the engine has sufficient power to maintain the required longitudinal speed, the control systems are able to quickly compensate for any external disturbance with a minimum effect on the dynamics of the vehicle.

6.3 Controlled Model with Virtual Sensor

In section 6.2 general considerations on the robustness and performance of the control algorithm were made and they will not be repeated here. In this part, the attention will be concentrated on illustrating the performance difference between the system with the LRC and that with the Virtual Sensor (VS).

The validation of the VS is performed on a curved portion of the testing track (figure 6.13). As the figure shows, the agreement between VS output and the ideal path to be followed (considered equivalent to what the differential GPS would provide) is very good. The validation metrics V for the X and Y coordinates are 98.28% and 99.92%, respectively. It is important to mention, however, that when the VS is employed along longer paths and/or tighter curves the goodness of fit rapidly deteriorates. If the sensor is identified on longer portions of road (e.g. several laps of the Centro Sicurezza track) then its behaviour appears more generalized, even if less precise.

It is then confirmed that the operation of the VS is suitable to substitute for the LRC only for brief periods. Let us now evaluate the performance of the virtual sensor and then

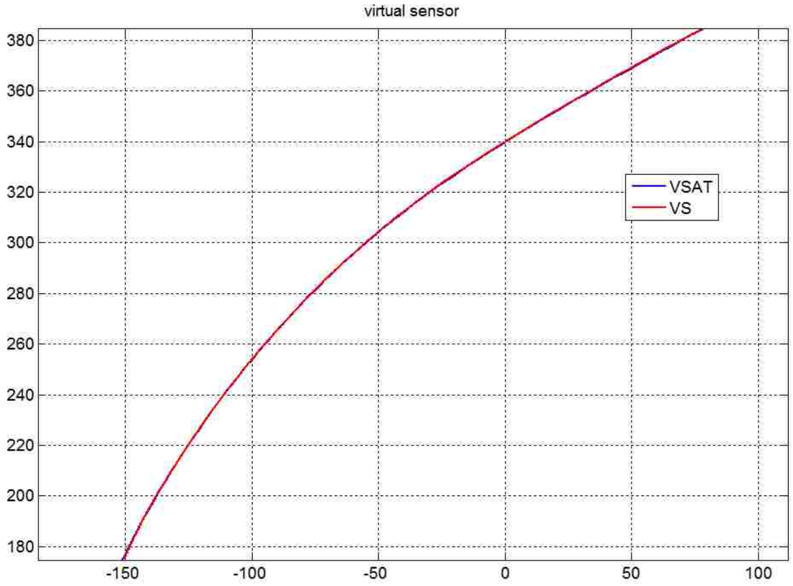


Figure 6.13: *Validation route for the virtual sensor in X-Y plane. The blue curve represents the ideal path to be followed, the red curve the path generated by the virtual sensor*

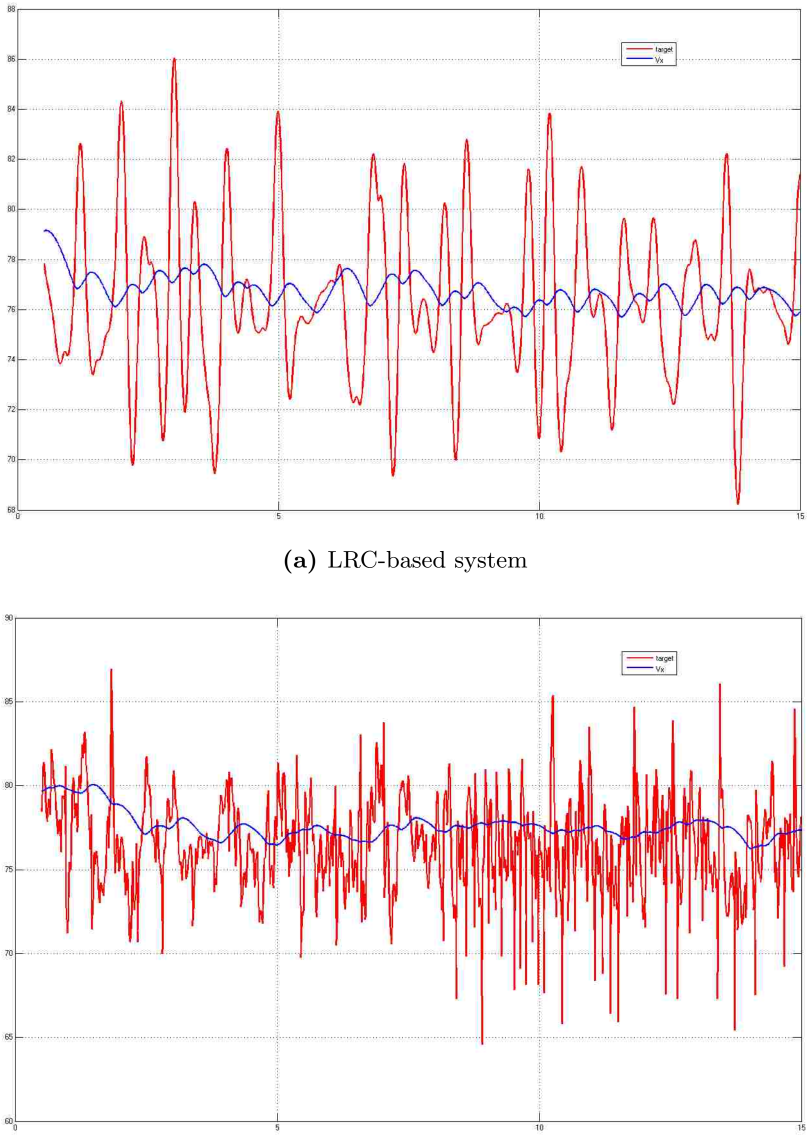
compare it with the system equipped with LRC.

Remark on X and Y Input Coordinates

The first consideration that should be made when dealing with input data coming from differential GPS (and even more so from standard GPS) is that the behaviour of the system modifies with respect to the situation in which the path to be followed was artificially created with a speed v_x and a road curvature K_l. In this latter case, in fact, the profile of the road is very smooth and the variation of the various vehicle states continuous and gradual. On the other hand, when considering X and Y input coordinates, the speed and the curvature imposed to the controllers have a more irregular profile. The effect of this difference is present both when the VS is implemented and when the camera is implemented. As will be discussed later, the vehicle dynamics acts as a filter that avoids erratic behaviors of the vehicle.

6.3.1 Longitudinal Control

In order to compare the performance of the model with either the LRC-based controller or the VS-based controller, the portion of the Centro Sicurezza track already used for VS validation and shown in figure 6.13 will be employed. The standard system (LRC-based) will be directly fed with the differential GPS (i.e. exact) position coordinates. The system with the VS will be of course supplied with the coordinates of the standard GPS.



(a) LRC-based system

(b) VS-based system

Figure 6.14: *Comparison between input speed (red) and actual vehicle speed (blue)*

Let us now consider the two plots of figure 6.14. As previously stated, the input speed v_x that results from the GPS coordinates is very erratic in nature. Nonetheless the combined effect of the controller (which limits the values of longitudinal acceleration and jerk) and of the inertia of the vehicle dynamics tends to filter out high frequency components of the input speed providing more constant outputs.

The variability is definitely higher in the case in which the VS is employed even if the higher frequency of the signal means that the output is actually more constant. In both cases the mean is very similar (around 77 km/h) and close to the mean of the speed at which vehicle simulations were carried out (around 75 km/h).

6.3.2 Lateral Control

Again the model is tested considering precise vehicle position for the system with LRC (so as to simulate the camera extracting the curvature information from the actual road) and the approximate vehicle position for the system with VS. Let us consider first the former system.

As shown for the instantaneous value of v_x, also the road curvature K_l has a strongly variable behavior. In figure 6.15 this parameter is plotted together with the variable q, i.e. the instantaneous distance between vehicle center of gravity and linear approximation of the road centerline.

Figure 6.15 highlights again that most of the variability of the input signal (K_l) is canceled by the filtering action of the controller and the inertia of the vehicle. Nonetheless, at first sight the values assumed by q appear extremely large (figure 6.16).

From a purely mathematical standpoint this is due to the very high peaks of road curvature generated by the Input Coordinate Transformation block (which correspond to curvature radii of about 20 m). As a result the distance q reaches values of 1 m while maximum acceptable levels are less than the half of that.

As it can be seen by figure 6.17, however, the trajectory followed by the vehicle in the X-Y plane appears well inside the ±0.4 m boundaries built around the exact position signal.

Occasionally, the position of the vehicle oscillates around the desired path (figure 6.18) and briefly “touches” the limits at ±0.4 m from the desired path.

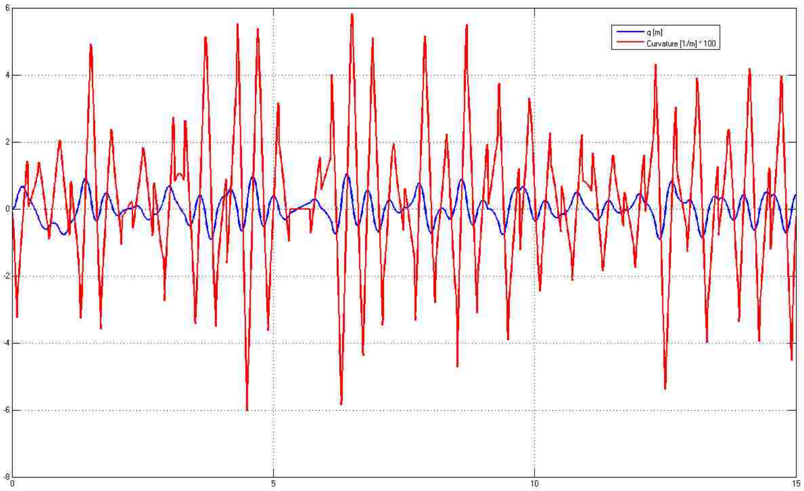


Figure 6.15: *Road curvature Kl (red, magnified 100 times) and distance CG-lane centerline q (blue) (LRC-based system)*

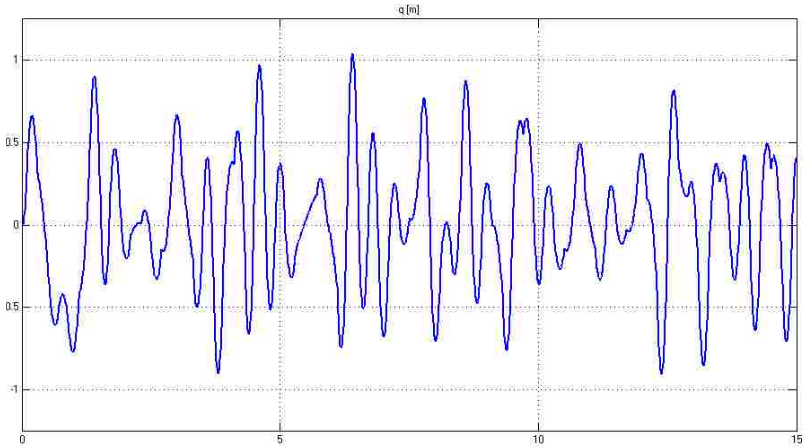


Figure 6.16: *Distance q between vehicle's CG and lane centerline (LRC-based system)*

An apparent contradiction between figures 6.16 and 6.18 arises. The former indicates that the vehicle should be (often) found at a distance larger than 0.4 m from the lane centerline, whereas the latter clearly shows that this is not the case. In order to understand the reason of this mismatch the following considerations should be made:

- Each component of the model has a certain level of *inertia*.

  This is true not only for the longitudinal and lateral vehicle dynamics, but also for the other dynamic blocks in the model (such as the Luenberger observers). All these

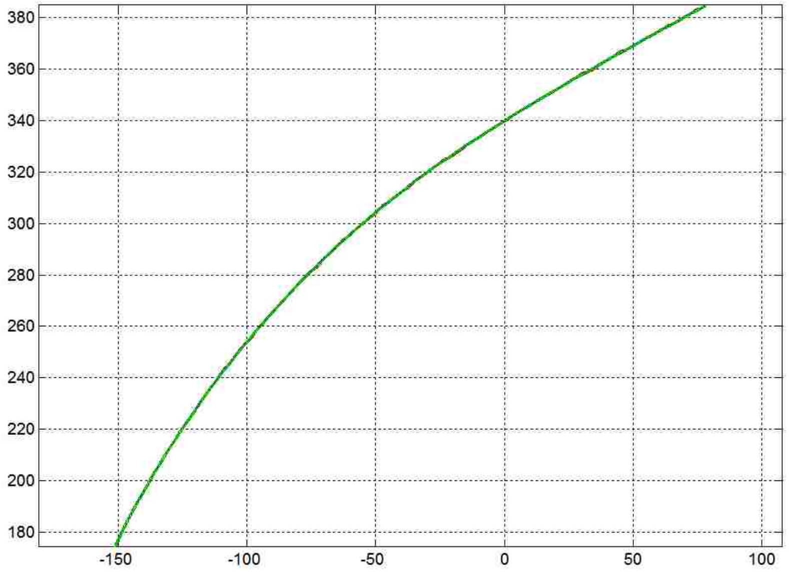


Figure 6.17: *X-Y trajectory of the vehicle (red) and* ±0.4 *m boundaries built around VSAT position signal (dark green) (LRC-based system)*

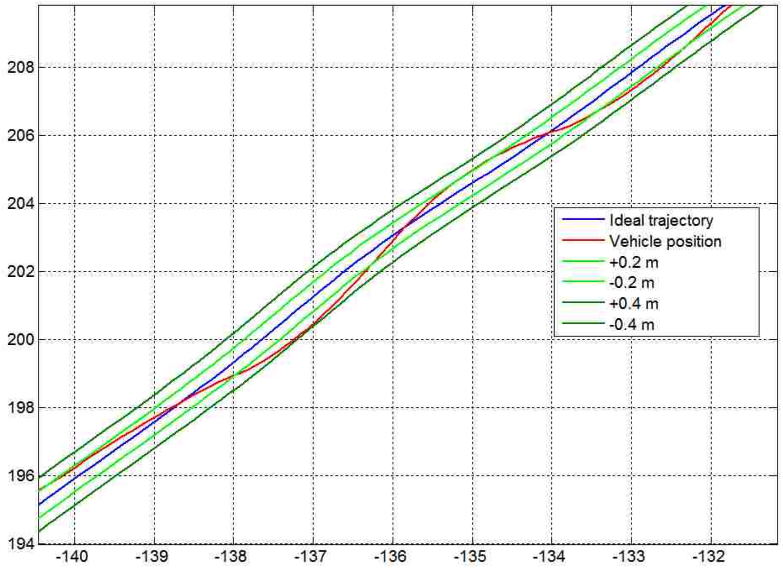


Figure 6.18: *Portion of the X-Y trajectory of the vehicle (red) plotted together with the desired path (blue) and the limits at* 0.2 *and* 0.4 *m from the centerline (light and dark green, respectively) (LRC-based system)*

components will cause the vehicle model to respond more gradually to the inputs, making the vehicle follow an average signal;

- The *average* value of q is below 1 cm.

  As a consequence, even if at times the vehicle does not follow exactly the lane centerline, it never distances itself significantly from it;

- The parameter q expresses the distance between vehicle's center of gravity and the linear *approximation* of the lane centerline.

  The rapidly varying nature of q and m indicates that the linear approximation of the lane centerline varies substantially on an instantaneous basis, whereas the actual centerline has a much smoother profile. It can be concluded that while the vehicle actually *has* a large distance from the instantaneous approximation of the centerline, on average, its distance from the actual centerline is much smaller.

Let us now repeat the analysis considering the vehicle speed and the road curvature as provided by the virtual sensor.

Similarly to what happened to v_x, whose behavior was degraded compared to the situation in which the LRC drives the lateral controller, also the road curvature K_l resulting from the VS is more variable and has larger peak values.

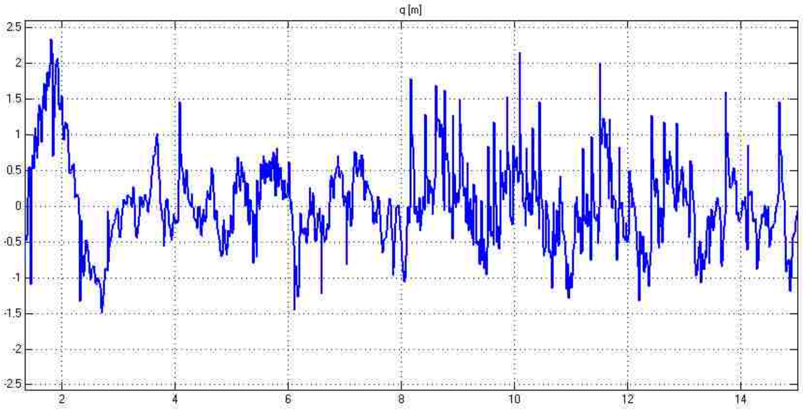


Figure 6.19: *Distance q between vehicle's CG and lane centerline (VS-based system)*

Figure 6.19 shows a strongly variable behavior and large peaks of more than 2 m. Compared to what is shown in figure 6.16 there is a clear degradation in terms of performance.

Nonetheless, once again figure 6.20 shows that the vehicle always stays within the requested boundaries of ±0.4 m from the lane centerline. The first few meters, corresponding to roughly 2 s of vehicle traveling, show the vehicle exiting from the desired region in the X-Y plane. This appears to be caused by large values of the derivatives in the Input Coordinate Transformation block.

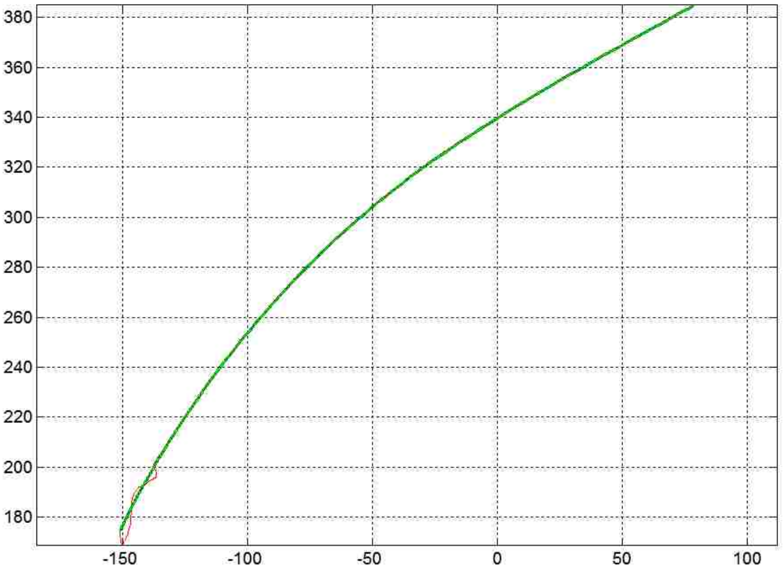


Figure 6.20: *X-Y trajectory of the vehicle (red) and ±0.4 m boundaries built around VSAT position signal (dark green) (VS-based system)*

After this first phase, vehicle motion remains well confined inside the desired boundaries. As figure 6.21 shows, even in the most demanding portion of the track, the system is able to maintain the distance from the lane centerline below (or equal to) 0.4 m.

The apparent contradiction between figures 6.19 and 6.21 can be understood considering the discussion on block inertia, average value and centerline approximation previously carried out.

It is then possible to conclude that, when the Virtual Sensor substitutes the Lane Recognition Camera in providing to the lateral controller a reference path to be followed, the overall control system retains its ability to keep the vehicle inside the lane.

The limited amount of suitable GPS data does not allow more extensive verification of the performance of the model in more demanding conditions. Nonetheless, it is clear

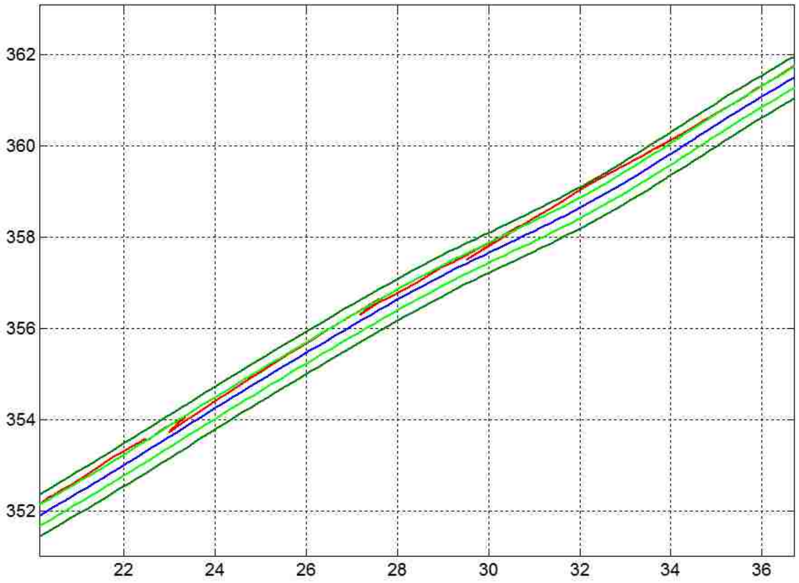


Figure 6.21: Portion of the X-Y trajectory of the vehicle (red) plotted together with the desired path (blue) and the limits at 0.2 *and* 0.4 *m from the centerline (light and dark green, respectively) (VS-based system)*

from the analysis carried out on the data generated at the Centro Sicurezza track, that a decrease in performance has to be expected when the VS is employed. The following points, in particular, can be underlined:

1. The reconstructed vehicle position signal is significantly less smooth than that provided by the differential GPS and this translates into more variability in the v_x and K_l signals. This can lead to numerical problems (e.g. high values of position derivatives in the Input Coordinate Transformation block);

2. The average distance from the actual centerline is, in the system implemented with the VS, above 4 *cm*. This is an indication of the general degradation in performance experienced when the LRC is replaced with the VS;

3. The movement of the vehicle in the lane appears more erratic and larger steering angles are used. In general, a decrease in the comfort level has to be expected. This effect was noticed also when the inputs of the model are changed from v_x and K_l to the exact X and Y coordinates of the lane centerline. Nonetheless, this issue gets even

more relevant when the coordinates are estimated through the virtual sensor.

In the attempt to quantify the amount of performance degradation experienced when implementing the Virtual Sensor, it is possible to refer to the mathematical agreement between the desired vehicle path (lane centerline as provided by the differential GPS) and actual vehicle path. Considering the system based on the LRC, for the portion of the track considered above, the validation metric V equals 99.33 and 99.89% for the X and Y coordinates, respectively. When making reference to the system based on the VS, these values reduce to 98.24 and 99.74%, respectively. The amount of degradation is then minimum.

In order to further improve the performance of the VS-based system, a possibility is to decrease vehicle speed. When this occurs the movement of the vehicle appears slightly less jerky. A possible strategy could then be that of reducing vehicle speed when the main source of position information switches from LRC to VS.

Chapter 7

Conclusion and Recommendations

In this last chapter of this thesis a brief conclusion is drawn and some recommendations for future work are offered.

In chapter 1 the general context in which this project is inserted was introduced. All major automakers and top tier suppliers are looking at the promising area of autonomous drive with the important objective of reducing the number of deaths still occurring on the road. It was in fact pointed out how, in the vast majority of the cases, the largest cause of automotive accidents is linked to improper driver behavior. It is then extremely important to try to support the driver in his/her task and, where possible, to automatize the driving duty.

The systems dedicated to this purpose, collected under the term Advanced Driver Assistance Systems (ADAS), are expected to flood the market in the next few years and some interesting examples are already on the road. In the vast majority of the cases these systems rely on the so-called *look-ahead approach*, i.e. they are based on video sensors (Lane Recognition Camera, LRC) used to capture useful information on the portion of road ahead of the vehicle. Usually the information to be captured is the relative position of the vehicle with respect to the lane markings, which mark the boundaries of the drivable portion of the road. It may occur that this crucial information is lacking, either because road markings are not present or because their visibility is somehow reduced.

In order to overcome this difficulty and to sustain — at least for a brief amount of time (until look-ahead information is restored or driver active intervention is resumed) — autonomous driving functionalities, a simple and economic way of dealing with the lack of data provided by the LRC was presented in this project. Two fundamental outputs were then provided in the previous chapters:

1. An autonomous driving vehicle model built in Simulink. This model embeds all the most relevant components necessary to simulate the behaviour of a real automated vehicle: accurate longitudinal and lateral dynamics to simulate vehicle motion, sensors and actuators to allow the correct interaction of the vehicle with the surroundings and a suitable control system capable of dealing with the constraints imposed both on the longitudinal and lateral dynamics;

2. A Virtual Sensor designed to substitute the LRC when its operation is degraded or impaired. This device, whose mathematical foundations and practical implementation were presented in chapters 2 and 4, allows the reconstruction of the precise vehicle position by *fusing* together different data: GPS coordinates and vehicle dynamic states.

The development of the overall model required the application of several techniques of increasing interest in the engineering field: Neural Networks, ARX and the already-cited Virtual Sensor. The performance of the system was evaluated considering three distinct implementations:

- *Uncontrolled* vehicle dynamics.

  Simulations of the vehicle not subject to the action of the driver or of control devices were carried out with the task of showing the instability of the system and the need of control;

- Controlled dynamics via *Lane Recognition Camera*.

  Testing the model controlled through look-ahead approach allowed to appreciate its stabilizing action and its effectiveness with which tight performance requirements were satisfied in a broad spectrum of situations;

- Controlled dynamics via *Virtual Sensor*.

  The system implementing the VS is successful at substituting the video sensor in those situations where road markings are unavailable. The GPS signal is combined with vehicle data in order to reconstruct the desired path which is then followed by the system only with a slight degradation in performance.

As mentioned in chapter 6, though, the Virtual Sensor as designed here shows limitations in its application: because of the deterioration of the GPS signal due to external factors, the VS does not always correctly generalize its behavior to situations different from those used for its identification. As a result, the Virtual Sensor as presented here is to be intended as useful at reconstructing vehicle position for brief periods in which the camera sensor output is unavailable, rather than a system intended to completely eliminate the need of the LRC.

Considering this limitation and some possible areas of improvement met in the development of this project it is possible to highlight some recommendations for future work:

- The functionalities of the presented system can be expanded exploiting its modularity. ADAS systems could be implemented in the overall model in order to enhance its appeal for on-vehicle implementation. Adaptive Cruise Control and Lane Departure Warning are two simple examples. As a result, the presented model could obtain a greater understanding of the driving scenario and hence get closer to level three automation;

- The performance of the controllers could be increased by resorting to non-linear control devices, better suited to the characteristics of the vehicle dynamics. Alternatively, more configurations of the LQI controller could be studied in order to improve the trade-off between comfort and performance. Suitable strategies to switch between different controller tunings could be implemented in order to adapt to the different driving conditions;

- Due to the lack of time, some parts of the model were implemented according to the simplest approach available. As a result, the coordinate transformation blocks could be improved and replaced with more sophisticated units. Filtering of the GPS coordinates would allow for less erratic behavior of the input signals;

- Similarly, the LRC block could be designed in such a way as to provide higher order approximations of the lane centerline in order to better suit the design of the road;

- Concerning the crucial implementation of the Virtual Sensor, a larger dataset would allow for better identification and hence higher precision of the system in different situations;

- The Virtual Sensor design should be expanded in such a way as to consider a *wider set of inputs*, including those related to the degradation of performance of a standard GPS (e.g. synchronization errors, atmospheric disturbances, *etc.*). The identification and validation sets used in this project were, in fact, portions of the same track. This means that they were recorded at similar time instants and similar geographical locations, making them much less sensitive to external factors than what would be expected for real position data acquired in different days and different locations. Furthermore, an analytical model computing the *expected* vehicle position could be added so as to be compared with the GPS signal and further refine the estimate;

- The switch between the LRC-based and the VS-based solution should be analyzed to identify and solve possible negative effects on the control of the vehicle. More in detail, a strategy should be elaborated in such a way as to identify the conditions that should determine the switch between the two configurations of the model. A possible solution could be that of having the VS-based implementation always running in parallel to the main module driven by the LRC; then, when the information provided by the LRC becomes inconsistent with the position computed by the VS, the switch is performed;

- Finally, testing the system on actual driving conditions and actual hardware would allow for a more meaningful validation.

Bibliography

[1] M. Richter, H.-C. Pape, D. Otte and C. Krettek. *Improvements in passive car safety led to decreased injury severity - a comparison between the 1970s and 1990s.* Injury, Int. J Care Injured (2005) 36, 484-488.

[2] *SAE 2015 Active Safety Systems Symposium.* Plymouth, Michigan, USA. November 4-5, 2015.

[3] *Road safety in the European Union. Trends, statistics and main challenges.* March 2015.

[4] SAE International's standard J3016. *Taxonomy and Definitions for Terms Related to On-Road Motor Vehicle Automated Driving Systems.* Available online at: `http://standards.sae.org/j3016_201401/`

[5] A. Lindgren and F. Chen. *State of the Art Analysis: An Overview of Advanced Driver Assistance Systems (ADAS) and Possible Human Factors Issues.* Human Factors and Economic Aspects on Safety. In " Proceedings of the Swedish Human Factors Network (HFN) Conference, April 5 ? 7, 2006, Linkping, Sweden ". Edited by Clemens Weikert HFN report 2007-1; pp. 38-50

[6] R. Bishop. *Intelligent Vehicle Technology and Trends.* Norwood, MA: Artech House Inc. 2005.

[7] *Driving Safely Into the Future with Applied Technology.* Informational Brochure, U.S. DOT ITS Joint Program Office, Publication number FHWA-OP-99-034.

[8] Google Self-Driving Car Project. `https://www.google.com/selfdrivingcar/`

[9] Autonomous Driving - Mercedes-Benz. `https://www.mercedes-benz.com/en/mercedes-benz/innovation/autonomous-driving/`

[10] Autonomous long-distance drive - Mercedes-Benz. `https://www.mercedes-benz.com/en/mercedes-benz/innovation/autonomous-long-distance-drive/`

[11] *Delphi Successfully Completes First Coast-to Coast Automated Drive.* `http://investor.delphi.com/investors/press-releases/press-release-details/2015/Delphi-Successfully-Completes-First-Coast-to-Coast-Automated-Drive/default.aspx` Press Release; April 2nd, 2015.

[12] *The Future is Now: Nissan advances plans to have commercially viable autonomous drive vehicles on the road by 2020.* http://www.nissanusa.com/blog/autonomous-drive-car March 15, 2014.

[13] Audi piloted driving. http://www.audi.com/com/brand/en/vorsprung_durch_technik/content/2014/10/piloted-driving.html

[14] *Bosch: In 2020, cars should be driving themselves on the freeway.* By Fortune: http://fortune.com/2015/07/16/bosch-self-driving-car/

[15] Y. Alkhorsid, K. Aryafar, G. Wanielik and A. Shokoufandeh. *Camera-Based Lane Marking Detection for ADAS and Autonomous Driving.* Springer International Publishing Switzerland 2015, M. Kamel and A. Campilho (Eds.): ICIAR 2015, LNCS 9164, pp. 514-519, 2015.

[16] Honda Accord, 2016. Brochure available online at: https://www.honda.ca/Content/honda.ca/1b51e125-0d04-4588-a89a-627ea9674325/ModelPage_Downloads/HON16887_02_2016_Accord_Brochure_online.pdf

[17] Active Lane Keeping Assist - Mercedes-Benz. http://www.mercedes-benz.ca/content/canada/mpc/mpc_canada_website/en/home_mpc/passengercars/home/new_cars/corporate_sales_programs/advantages___benefits/safety.html

[18] 3. C. Guo, J. Meguro, Y. Kojima and T. Naito. *A Multimodal ADAS System for Unmarked Urban Scenarios Based on Road Context Understanding.* In IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS, VOL. 16, NO. 4, AUGUST 2015, pp. 1690-1704.

[19] N. Kaempchen and K.C.J. Dietmayer. *Fusion of Laserscanner and Video for Advanced Driver Assistance Systems.* Germany, University of Ulm, Department of Measurement, Control and Microtechnology.

[20] M. Darms, F. Foelster, J. Schmidt, D. Froehlich and A. Eckert. *Data Fusion Strategies in Advanced Driver Assistance Systems.* Continental AG, in SAE Int. Journal Passeng. Cars - Electron. Electrc. Systems, Volume 3, Issue 2, pp. 176-182.

[21] Fiat 500X brochure. Available online at: https://www.fiatusa.com/assets/pdf/brochures/fiat_500x.pdf

[22] M. Milanese, C. Novara, K. Hsu, K. Poolla, *Nonlinear virtual sensors design from data*, 14th IFAC Symposium on System Identification, Newcastle, Australia, 2006

[23] World Health Organization. Global Health Observatory data repository. *Road traffic deaths. Data by country.* Available online at http://apps.who.int/gho/data/node.main.A997

[24] World Health Organization. *Global Status Report on Road Safety 2013: supporting a decade of action.* Geneva, Switzerland, 2013.

[25] World Health Organization. *Global Status Report on Road Safety 2015.* Geneva, Switzerland, 2015.

[26] Barbara E. Sabey, H. Taylor. *The known risks we run: the highway.* Transport and Road Research Laboratory, 1980

[27] U.S. Department of Transportation. National Highway Traffic Safety Administration. *National Motor Vehicle Crash Causation Survey. Report to Congress.* July 2008.

[28] UK Government. Statistical data set. *Contributory factors for reported road accidents (RAS50).* Available online at `https://www.gov.uk/government/statistical-data-sets/ras50-contributory-factors` Last updated: 24 September 2015.

[29] Treat, J. R.; Tumbas, N. S.; McDonald, S. T.; Shinar, D.; Hume, R. D.; Mayer, R. E.; Stansifer, R. L.; Castellan, N. J. *Tri-level study of the causes of traffic accidents: final report. Executive summary.* Report covers the period Aug 1972-June 1977. Released Oct 1979.

[30] Society of Automotive Engineers. *The future of vehicle safety*, by J. Gaus. April 2005.

[31] U.S. Department of Transportation. National Highway Traffic Safety Administration. Safety Assurance, Office of Vehicle Safety Compliance. *Federal Motor Vehicle Safety Standards and Regulations.* See e.g. FMVSS 208 for vehicle crashworthiness against frontal crashes.

[32] IMS. *Electronic Stability Control Systems Saving Lives, Report Finds.* Available online at: `http://www.intellimec.com/myconnectedcar/files/2012/12/withwithout_esp3.jpg` [accessed June 3, 2016]

[33] Wikipedia contributors. *Advanced driver assistance systems.* Wikipedia, The Free Encyclopedia, March 11, 2016, 08:55 UTC, `https://en.wikipedia.org/w/index.php?title=Special:CiteThisPage&page=Advanced_driver_assistance_systems&id=709505099` [accessed June 3, 2016]

[34] Karel A. Brookhuis, Dick de Waard and Wiel H. Janssen. *Behavioural impacts of Advanced Driver Assistance Systemsan overview.* EJTIR, no. 3 (2001), pp. 245-253.

[35] Vallet, M. *Les dispositifs de maintien de la vigilance des conducteurs de voiture.* In: "Le maintien de la vigilance dans kes transports". M. Vallet (Ed.). Caen: Paradigm. 1991.

[36] European Union. *Dedicated Road Infrastructure for Vehicle Safety in Europe.* CORDIS Archive available at: `http://cordis.europa.eu/telematics/tap_transport/research/16.html` [accessed June 4, 2016]

[37] Maple Shade Mazda. *LDWS (Lane Departure Warning System).* Available online at: `http://www.msmazda.com/mrcc-ldws` [accessed June 10, 2016]

[38] Cranky Driver. *What the Hell: Is Adaptive Cruise Control?.* Available online at: `http://www.crankydriver.com/word/what-the-hell-is-adaptive-cruise-control/` [accessed June 10, 2016]

[39] How Stuff Works Auto. *How Self-parking Cars Work*. Available online at: http://www.auto.howstuffworks.com/car-driving-safety/safety-regulatory-devices/self-parking-car2.htm/ [accessed June 10, 2016]

[40] MIT Technology Review. *10 Breakthrough Technologies* (2016). Available online at: https://www.technologyreview.com/s/600772/10-breakthrough-technologies-2016-tesla-autopilot/ [accessed June 12, 2016]

[41] Tesla Motors. *Model S, Autopilot*. Available online at: https://www.teslamotors.com/en_CA/models [accessed June 12,2016]

[42] Wikipedia contributors. *Electronic throttle control*. Wikipedia, The Free Encyclopedia, April 21, 2016, 06:40 UTC, https://en.wikipedia.org/w/index.php?title=Electronic_throttle_control&olid=716345615 [accessed June 14, 2016]

[43] T. Y. Shin, S. Y. Kim, J. Y. Choi, K. S. Yoon and M. H. Lee. *Modified Lateral Control of an Autonomous Vehicle By Look-Ahead and Look-Down Sensing*. In International Journal of Automotive Technology, Vol. 12, No.1, pp. 103-110 (2001).

[44] Nico Kaempchen, Klaus C.J. Dietmayer. *Fusion of Laserscanner and Video for Advanced Driver Assistance Systems*. University of Ulm, Department of Measurement, Control and Microtechnology.

[45] Stefano Malan, Mario Milanese, Pandeli Borodani, and Alessandro Gallione. *Lateral Control of Autonomous Electric Cars for Relocation of Public Urban Mobility Fleet*. In IEEE TRANSACTIONS ON CONTROL SYSTEMS TECHNOLOGY, VOL. 15, NO. 3, MAY 2007.

[46] David L. Hall and James Llinas. *Handbook of Multisensor Data Fusion*. CRC Press LLC. 2001

[47] Thomas Herpel, Christoph Lauer, Reinhard German and Johannes Salzberger. *Multisensor Data Fusion in Automotive Applications*. In 3rd International Conference on Sensing Technology, Nov. 30 Dec. 3, 2008, Tainan, Taiwan.

[48] Federico Castanedo. *A Review of Data Fusion Techniques*. In The Scientific World Journal, Volume 2013 (2013), Article ID 704504, 19 pages.

[49] JDL. *Data Fusion Lexicon. Technical Panel For C3*. F.E. White, San Diego, Calif, USA, Code 4^{20}, 1991.

[50] S. Tuohy, M. Glavin, C. Hughes, E. Jones, M. Trivedi, L. Kilmartin. *Intra-Vehicle Networks: A Review*. In IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS, VOL. 16, NO. 2, APRIL 2015.

[51] Siemens AG. *Vehicle-to-X (V2X) communication technology*, 2015. Infographic available online at: https://www.mobility.siemens.com/mobility/global/SiteCollectionDocuments/en/road-solutions/urban/trends/siemens-vehicle-to-x-communication-technology-infographic.pdf [accessed June 15, 2016]

[52] Embedded Computing Design. *V2X technology makes cars safer*, 2015. Available online at `http://embedded-computing.com/guest-blogs/v2x-technology-makes-cars-safer/` [accessed June 15, 2016]

[53] Denso Dynamics. *All About V2X: Talking Car Technology.* Available online at: `http://www.densodynamics.com/all-about-v2x-talking-car-technology/` [accessed June 15, 2016]

[54] BROADCOM. *The Next Frontier of Car Technology: Connecting to Everything*, 2015. Available online at: `http://www.broadcom.com/blog/automotive-technology-2/the-next-frontier-of-car-technology-connecting-to-everything/` [accessed June 15, 2016]

[55] Khalid El Majdoub, Fouad Giri, Hamid Ouadi, Luc Dugard, Fatima Zara Chaoui. *Vehicle longitudinal motion modeling for nonlinear control.* Control Engineering Practice, Elsevier, 2012, 20 (1), pp.69 - 81.

[56] Pushkin Kachroo, Masayoshi Tomizuka and Alice M. Agogino. *A Comprehensive Strategy for Longitudinal Vehicle Control with Fuzzy Supervisory Expert System.* 0-7803-2559-1/9. IEEE, 1995.

[57] CAFB ACBA. *Road Performance Tires - The Theory.* Available online at: `http://www.cafb-acba.ca/education/road-performance-tires-the-theory.html` [accessed June 18, 2016]

[58] Thomas D. Gillespie. *Fundamentals of Vehicle Dynamics.* Published by Society of Automotive Engineers, Inc. 1992.

[59] Xiao-Dong Sun*, Angel Andres**, Tony Burton** and Peter Scotson*. *Application of robust loop-sharing control to EPS torque control augmentation.* *TRW Conekt, Solihull B90 4GW England, **TRW Electric Steering Ltd. Birmingham, B6 7AY, England.

[60] Jse E. Naranjo, Carlos Gonzlez, Ricardo Garca, Teresa de Pedro, and Rodolfo E. Haber. *Power-Steering Control Architecture for Automatic Driving.* In IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS, VOL. 6, NO. 4, DECEMBER 2005.

[61] P.S. Chandakkar, Y. Wang, B. Li. *Improving vision-based self-positioning in intelligent transportation systems via integrated lane and vehicle detection.* In: 2015 IEEE Winter Conference on Applications of Computer Vision (WACV), pp. 404411, January 2015

[62] Aish Dubey. *The challenges and opportunities for ADAS stereo vision applications, part I.* EDN Network. October 14, 2015.

[63] Mohamed Aly. *Real time Detection of Lane Markers in Urban Streets.* California Institute of Technology.

[64] Yue Wang, Eam Khwang Teoh and Dinggang Shen. *Lane detection and tracking using B-Snake.* Elsevier. Image and Vision Computing 22 (2004), 269280.

[65] Young Uk Yim and Se-Young Oh. *Three-Feature Based Automatic Lane Detection Algorithm (TFALDA) for Autonomous Driving*. In IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS, Vol. 4, No. 4, December 2003.

[66] Jens Goldbeck and Bernd Huertgen. *Lane Detection and Tracking by Video Sensors*. IEEE 0-7803-4975-X/98. 1999.

[67] Mario Bellino, Yuri Lopez de Meneses, Peter Ryser and Jacques Jacot. *Lane detection algorithm for an onboard camera*. Photonics in the Automobile, edited by Thomas P. Pearsall, Proceedings of SPIE Vol. 5663 (SPIE, Bellingham, WA, 2005).

[68] Y. Chen and M. Y. He. *Vehicle-Mounted Camera Optimal Setting and Lane Projective Model for Lane Detection System*. Trans Tech Publications. Applied Mechanics and Materials ISSN: 1662-7482, Vols. 220-223,pp 1362-1367.

[69] Young-Woo Seo. *Tracking and Estimation of Ego-Vehicles States for Lateral Localization*. The Robotics Institute Carnegie Mellon University Pittsburgh, Pennsylvania 15213. May 2014.

[70] Chris Kreucher, Sridhar Lakshmanan and Karl Kluge. *A Driver Warning System Based on the LOIS Lane Detection Algorithm*. University of Michigan.

[71] Wikipedia contributors. *Global Positioning System*. Wikipedia, The Free Encyclopedia, 15 June 2016, 09:02 UTC, `https://en.wikipedia.org/w/index.php?title=Global_Positioning_System&oldid=725379705` [accessed 20 June 2016]

[72] Penn State University. GEOG 482. The Nature of Geographic Information. *18. Satellite Ranging*. Available online at: `https://www.e-education.psu.edu/natureofgeoinfo/c5_p18.html` [accessed 24 June 2016]

[73] Trimble. *Trimble GPS Tutorial*. Available online at: `www.trimble.com/gps_tutorial/whygps.aspx` [accessed 24 June 2016]

[74] Wikipedia contributors. *Dilution of precision (GPS)*. Wikipedia, The Free Encyclopedia, 18 February 2016, 16:23 UTC, `https://en.wikipedia.org/w/index.php?title=Dilution_of_precision_(GPS)&oldid=705625608` [accessed 16 March 2016]

[75] W. Rahiman and Z. Zainal. *An overview of development GPS navigation for autonomous car*. In 2013 IEEE 8th Conference on Industrial Electronics and Applications (ICIEA), 19-21 June 2013. pp 1112-1118.

[76] Anton T. van Zanten. *Evolution of Electronic Control Systems for Improving the Vehicle Dynamic Behavior*. Robert Bosch GmbH.

[77] Steven A. Nobe and Fei-Yue Wang. *An Overview of Recent Developments in Automated Lateral and Longitudinal Vehicle Controls*. IEEE 0-7803-7807-2/01. 2001.

[78] D. J . LeBlanc, G. E. Johnson, P. J. T. Venhovens, G. Gerber, R. DeSonia, R. D. Ervin, C. Lin, A. G. Ulsoy and T. E. Pilutti. *CAPC: A Road-Departure Prevention System*. IEEE Conntrol Systems, Vol. 16, No. 6, pp. 61-71. December, 1996.

[79] Wikipedia contributors. *Fuzzy logic*. Wikipedia, The Free Encyclopedia, 15 June 2016, 15:01 UTC, `https://en.wikipedia.org/w/index.php?title=Fuzzy_logic&oldid=725419248` [accessed 25 June 2016]

[80] K. Nishimori, M. Okada and N. Ishihara. *Simulation of fuzzy control using vision system in driving an automatic vehicle.* Dept. of Electrical and Electronic Engineering, Tottori University. Koyama, Tottori 680, Japan.

[81] Wikipedia contributors. *State observer*. Wikipedia, The Free Encyclopedia, 6 June 2016, 08:29 UTC, `https://en.wikipedia.org/w/index.php?title=State_observer&oldid=723958593` [accessed 25 June 2016]

[82] Pandeli Borodani*. *Virtual Sensors: an Original Approach for Dynamic Variables Estimation in Automotive Control Systems.* * Centro Ricerche Fiat - Vehicle Systems. AVEC '08. 9th International Symposium on Advanced Vehicle Control. Kobe International Conference Center, Kobe, Japan.

[83] M. Milanese, C. Novara, K. Hsu, K. Poolla. *Filter design from data: direct vs. two-step approaches.* ACC 2006, Minneapolis, Minnesota, USA, June 2006.

[84] Fortune. *GM Buying Self-Driving Tech Startup for More Than $1 Billion* by Dan Primack, Kirsten Korosec. Available online at: `http://fortune.com/2016/03/11/gm-buying-self-driving-tech-startup-for-more-than-1-billion/` [accessed 27 June 2016]

[85] CB INSIGHTS. *30 Corporations Working On Autonomous Vehicles.* April 18, 2016. Available online at: `https://www.cbinsights.com/blog/autonomous-driverless-vehicles-corporations-list/` [accessed 27 June 2016]

[86] BMW Group. BMW Group driving the transformation of individual mobility with its Strategy NUMBER ONE ¿ Next. Press Release, March 16, 2016. Available online at: `https://www.press.bmwgroup.com/global/article/detail/T0258269EN/bmw-group-driving-the-transformation-of-individual-mobility-with-its-strategy-number-` [accessed 27 June 2016]

[87] Fortune. *Bosch: In 2020, cars should be driving themselves on the freeway* by Kirsten Korosec. Available online at: `http://fortune.com/2015/07/16/bosch-self-driving-car/` [accessed 27 June 2016]

[88] The Guardian. *Convoy of self-driving trucks completes first European cross-border trip.* Agence France-Presse, April 7 2015. Available online at: `https://www.theguardian.com/technology/2016/apr/07/convoy-self-driving-trucks-completes-first-european-cross-border-trip` [accessed 27 June 2016]

[89] WIRED. *Delphi's new self-driving car teaches us to give up the wheel* by Alex Davies, January 4, 2016. Available online at: `https://www.wired.com/2016/01/delphis-new-self-driving-car-teaches-us-to-give-up-the-wheel/` [accessed 27 June 2016]

[90] The Ford Motor Company, MediaCenter. *Ford Smart Mobility LLC established to develop, invest in mobility services; Jim Hackett named subsidiary chairman.* Available online at: `https://media.ford.com/content/fordmedia/fna/us/en/news/2016/03/11/ford-smart-mobility-llc-established--jim-hackett-named-chairman.html` [accessed 27 June 2016]

[91] Forbes. *Google's Gargantuan Push For Cars With No Steering Wheel By 2020* by Leo King, March 19, 2015. Available online at: `http://www.forbes.com/sites/leoking/2015/03/19/googles-gargantuan-push-for-cars-with-no-steering-wheel-by-2020/#ab1650f7f7db` [accessed 27 June 2016]

[92] Bloomberg. *Fiat, Google Plan Partnership on Self-Driving Minivans*, by Tommaso Ebhardt, May 3, 2016. Available online at: `http://www.bloomberg.com/news/articles/2016-05-03/fiat-google-said-to-plan-partnership-on-self-driving-minivans` [accessed 27 June 2016]

[93] GoMentum Station. *Autonomous Vehicle (AV) Program.* Available online at: `http://gomentumstation.net/project/av/` [accessed 27 June 2016]

[94] Mercedes-Benz. *The Mercedes-Benz F 015 Luxury in Motion.* Available online at: `https://www.mercedes-benz.com/en/mercedes-benz/innovation/research-vehicle-f-015-luxury-in-motion/` [accessed 27 June 2016]

[95] Wikipedia contributors. *Mobileye.* Wikipedia, The Free Encyclopedia, 25 June 2016, 11:09 UTC. Available online at `https://en.wikipedia.org/w/index.php?title=Mobileye&oldid=726930786` [accessed 27 June 2016]

[96] Road Show by CNET. *Nissan CEO Carlos Ghosn talks self-driving cars, EVs at New York show* by Antuan Goodwin. Available online at: `http://www.cnet.com/roadshow/news/nissan-ceo-carlos-ghosn-talks-self-driving-cars-evs/` [accessed 27 June 2016]

[97] Digital Trends. *Peugeot-Citroen announces a huge model offensive, and plan to return to the U.S.* by Ronan Glon, April 5, 2016. Available online at: `http://www.digitaltrends.com/business/peugeot-citroen-push-to-pass-plan-news-details/` [accessed 27 June 2016]

[98] Recode. *Elon Musk: Self-Driving Cars Are Coming Sooner Than You Think* by Mark Bergen, October 6, 2015. Available online at: `http://www.recode.net/2015/10/6/11619270/elon-musk-self-driving-cars-are-coming-sooner-than-you-think` [accessed 27 June 2016]

[99] The Wall Street Journal. *Toyota Hires Entire Staff of Autonomous-Vehicle Firm* by Mike Ramsey, updated March 9, 2016. Available online at: `http://www.wsj.com/articles/toyota-grabs-tech-talent-by-hiring-entire-jaybridge-staff-1457553870` [accessed 27 June 2016]

[100] REUTERS. *Uber seeking to buy self-driving cars: source* by E. Taylor and H. Ten Wolde, March 18, 2016. Available online at: `http://www.reuters.com/article/us-daimler-uber-idUSKCN0WK1C8` [accessed 27 June 2016]

[101] Fortune. *Volvo CEO: We will accept all liability when our cars are in autonomous mode* by K. Korosec, October 7, 2015. Available online at: `http://fortune.com/2015/10/07/volvo-liability-self-driving-cars/` [accessed 27 June 2016]

[102] WIRED. *China's self-driving bus shows autonomous tech's real potential* by Alex Davies, October 7, 2015. Available online at: `https://www.wired.com/2015/10/chinas-self-driving-bus-shows-autonomous-techs-real-potential/` [accessed 27 June 2016]

[103] Lennart Ljung. *System Identification Toolbox^{TM}User's Guide*. MathWorks. MATLAB R2015b.

[104] IPG AUTOMOTIVE. CarMaker®. Your open integration and test platform for virtual test driving. Available online at: `http://ipg.de/simulation-software/carmaker/` [accessed 14 August 2016]

[105] S. Samarasinghe. *Neural Networks for Applied Sciences and Engineering*. Auerbach Publications, 2006.

[106] A. Niu, D. Li, L. Wan, W. G. Wee, C. W. Carrier. *A comparative study of model fitting by using neural network and regression*. SPIE Vol. 1469 Applications of Artificial Neural Networks II (1991), pp. 495-505.

[107] Wikipedia contributors. *Nonlinear autoregressive exogenous model*. Wikipedia, The Free Encyclopedia. 3 October 2014. Web. Retrieved 7 March 2016, from `https://en.wikipedia.org/w/index.php?title=Nonlinear_autoregressive_exogenous_model&oldid=628118103` [accessed 6 July 2016]

[108] M. Hudson Beale, M. T. Hagan, H. B. Demuth. *Neural Network Toolbox^{TM}User's Guide*. MATLAB, MathWorks, R2015b.

[109] Gaurang Panchal, Amit Ganatra, Y P Kosta and Devyani Panchal. *Behaviour Analysis of Multilayer Perceptrons with Multiple Hidden Neurons and Hidden Layers*. International Journal of Computer Theory and Engineering, Vol. 3, No. 2, April 2011, ISSN:1793-8201.

[110] Wikipedia contributors. *Radial basis function*. Wikipedia, The Free Encyclopedia. 6 March 2016. Web. Retrieved 26 March 2016, from `https://en.wikipedia.org/w/index.php?title=Radial_basis_function&oldid=708598629` [accessed 6 July 2016]

[111] Zhongbiao Sheng and Xiaorong Tong. *The Application of RBF Neural Networks in Curve Fitting*. Advanced Materials Research. 2012 Trans Tech Publications, Switzerland. ISSN:1662-8985, Vols. 490-495, pp. 688-692.

[112] F. Burden, D. Winkler. *Bayesian regularization of neural networks*. Methods Mol Biol. 2008; 458:25-44.

[113] Lennart Ljung. *System Identification Toolbox^{TM}Getting Started Guide.* MATLAB, MathWorks, R2015b.

[114] Eduardo Martin Moraud. *Wavelet Networks.* February 6, 2009.

[115] V. Cerone, A. Chinu and D. Regruto. *Experimental results in vision based lane keeping for highway vehicles.* In Proceedings of the American Control Conference, Anchorage, AK. May 8-10, 2002.

[116] V. Cerone, M. Milanese and D. Regruto. *Combined Automatic Lane-Keeping and Drivers Steering Through a 2-DOF Control Strategy.* In IEEE TRANSACTIONS ON CONTROL SYSTEMS TECHNOLOGY, VOL. 17, NO. 1, JANUARY 2009.

[117] M. F. Land and D. Lee. *Where we look when we steer.* Article in Nature, July 1994.

[118] U.S. Department of Transportation. Federal Highway Administration. Safety, Lane Width. Available online at: `http://safety.fhwa.dot.gov/geometric/pubs/mitigationstrategies/chapter3/3_lanewidth.cfm` [accessed 19 July 2016]

[119] W. L. Oberkampf and T. G. Trucano. *Verification and validation in computational fluid dynamics.* Progress in Aerospace Sciences 38 (2002), pp. 209-272.

[120] Willard Miller. *The Formula for Curvature.* October 26, 2007.

[121] Rajesh Rajamani. *Vehicle Dynamics and Control.* Springer. Mechanical Engineering Series. Second edition, 2012. pp. 39-40.

Vita Auctoris

NAME:	Jerome Blanc
PLACE OF BIRTH:	Aosta, Italy
DATE OF BIRTH:	1992
EDUCATION:	Politecnico di Torino, B.Sc. in Automotive Engineering, Torino, Italy, 2014
	Politecnico di Torino, M.Sc. in Automotive Engineering, Torino, Italy, 2016
	University of Windsor, International M.A.Sc. in Mechanical Engineering, Windsor, Canada, 2016