

POLITECNICO DI TORINO

Master of Science Degree in Automotive Engineering

MASTER'S DEGREE THESIS

Longitudinal speed estimation and road condition identification using Artificial Neural Networks



Supervisors: Prof. Andrea Tonoli
Prof. Nicola Amati
Dr. Angelo Bonfitto
Dr. Stefano Feraco

Candidate: Alessandro Allisiardi

December 2018

To my parents Elvira and Claudio, who constantly supported me with endless love and blind confidence: you are the best.

To myself, since I know the work and the sacrifices I had to do to conquer this target. Stop thinking to your limits and start focusing on your strengths and power: you can be limitless, start living.

Acknowledgments

I would like to thank all the people that supported me during the development of this thesis work, giving advices on the procedures to be adopted, supporting me with the software employed and encouraging me during the tough times.

First of all I would like to thank professor Andrea Tonoli, who gave me the opportunity to participate to this project. Without his experience and wise advices, this thesis work would never start.

Then I want to thank my supervisors Dott. Angelo Bonfitto and Dott. Stefano Feraco, who supported me every single day with patience and professional competence.

Furthermore, I would also thank Professor Nicola Amati, who constantly followed the project and gave professional suggestions on how to improve the work.

Last but not least, I would like to thank my parents, Claudio and Elvira, who constantly gave me support and encouraged me during my life: you are the best.

Declaration

I formally declare that the present thesis work “Longitudinal speed estimation and road condition identification using Artificial Neural Networks” has been composed solely by myself and that it has not been submitted, in whole or in part, in any previous application for a degree. Furthermore, except where stated otherwise by reference or acknowledgment, the work presented here is entirely my own.

Allisiardi Alessandro
Torino, December 2018

Abstract

The longitudinal speed in vehicles is a fundamental dynamic parameter that is widely used to understand the behaviour of the vehicle: it is employed in order to control the dynamics of the vehicle and for this reason is one of the main inputs for Anti-Blocking System (ABS), Electronic Stability Program (ESP) and other intelligent devices, that are fundamental to guarantee safe driving conditions. Thus, the correct and precise estimation of the longitudinal speed is fundamental for the correct working conditions of these active control systems.

The vehicle speed estimation is generally computed using only data coming from the speedometer mounted on-board, which represents the speed computed as the mean speed of the vehicle's wheels. This method is not sufficient, because of the possibility to produce an error in the speed estimation is really high. Actually, it is common that in some driving conditions the vehicle can lock one of the four wheels or vice versa one of them can start spinning. During these occurrences, the behaviour of the longitudinal speed of a single wheel can corrupt the vehicle speed estimation, causing a wrong input for the safety systems of the vehicle. Thus, there is the need to improve our estimation capability.

The state of the art gives us different tools that can be used in order to improve the estimation of the longitudinal speed, such as cameras, GPS-based systems, Extended Kalman Filters (EKF) or Artificial Neural Networks (ANNs). The last solution is investigated in the present work.

The use of ANNs for vehicle speed estimation represents a new method, which exploits Machine Learning (ML) techniques. ANNs have been chosen to perform the vehicle speed estimation, since they have a lower cost of implementation and, at the same time, can guarantee a highly reliable and accurate results.

The present thesis work consists of two different and interconnected stages. The first stage is the design of a Nonlinear Autoregressive Neural Network with exogenous input (NARX) for the estimation of the longitudinal speed of the vehicle. This is performed using inputs parameters computed by means of sensors already implemented on the vehicle such as accelerometers (lateral and longitudinal), yaw rate sensor, steering sensors and wheel speed sensors. Then, the target for the estimation process is provided by the vehicle speed

signal, measured by means of an optical sensor. The estimation has been carried out for dry, wet and icy road conditions. The second stage is the design of a pattern recognition neural network, that must classify in real time the road condition. Thus, receiving this information, the Electronic Control Unit (ECU) can select the most accurate estimation among the three designed in the previous step for dry, wet and icy conditions.

The investigated system results to be very accurate, proving the reduction in the estimation error with respect to other systems commonly adopted for the vehicle speed estimation.

Contents

| | |
|--|-----|
| Acknowledgments..... | i |
| Declaration..... | iii |
| Abstract..... | v |
| Contents..... | vii |
| 1 Introduction | 1 |
| 1.1 Thesis motivation..... | 1 |
| 1.2 Thesis outline | 3 |
| 1.3 Longitudinal speed Estimation – State of the Art | 4 |
| 1.4 Road Condition Identification – State of the Art..... | 11 |
| 1.5 Machine learning basics..... | 18 |
| 1.5.1 Artificial Neural Networks..... | 21 |
| 2 Overall system layout..... | 27 |
| 2.1 Test conditions and dataset | 28 |
| 3 Longitudinal speed estimation..... | 31 |
| 3.1 NARX architecture..... | 31 |
| 3.2 Training and validation – parameter setting..... | 33 |
| 3.2.1 DRY conditions | 36 |
| 3.2.2 WET conditions | 44 |
| 3.2.3 ICE conditions..... | 50 |
| 4 Road condition identification..... | 57 |
| 4.1 Architecture of the classifier | 58 |
| 4.2 Dataset – feature extraction | 59 |

| | | |
|-----|-------------------------------|----|
| 4.3 | Training and Validation | 61 |
| 5 | Conclusions..... | 67 |
| 5.1 | Comments on the results | 67 |
| 5.2 | Future works | 68 |
| | List of Figures | 69 |
| | List of Tables..... | 71 |
| | References..... | 73 |
| | Appendix A | 75 |
| | Appendix B | 83 |

1 Introduction

1.1 Thesis motivation

In these years, one of the main research and development fields about automotive engineering is related to the safety, concerning both passenger and pedestrian safety. For this reason, nowadays we can notice that on vehicles there are a lot of electronic devices aimed to control the stability of the vehicle.

They are mainly divided in two big families:

- Reactive safety devices: those are tools that become active when emergency happens. Typical examples are the ABS, ESP, TCS. The ABS for instance, starts working when the wheel enters in the unstable working region, so the risk is that it locks and does not allow to perform correctly the breaking action. The ABS avoids the locking of the wheels, by reducing the pressure on the brake calliper and so allowing the wheel to rotate, instead of being locked.
- Active safety devices: their purpose is to prevent the dangerous events, by continuously adapting to the changing environment. We can list active suspensions systems, active aerodynamics devices and the Adaptive Cruise Control (ACC) systems which is implemented to keep a safe distance from the preceding vehicle and, if necessary, to brake in order to avoid collisions.

The common denominator of these devices is that their functions are all based on the analysis of dynamic parameters of the vehicle, and typically one of the most important is the longitudinal speed of the vehicle. The ABS functionality is based on the relationship between friction coefficient and slip σ of the tire, where σ is defined as [17]:

$$\sigma = \frac{v}{V} \tag{1}$$

where V is the vehicle speed.

Thus, it is easy to understand that the ability to compute or at least to estimate these dynamic parameters in real time, is becoming of paramount importance today, in order to guarantee the right safety level to the pedestrians and the passengers.

Of course this cannot be done just once for all the conditions: it is necessary to assure the correct working of these devices in all the possible driving conditions, that simply means to check the performance of those devices in dry, wet and icy conditions.

The point is that today many of these devices are already implemented on the vehicles and this means that we have already the possibility to compute the longitudinal speed of the vehicle (see next chapter for more details), but the challenge is related to the precision of the computation: since we speak about safety devices, a small mistake could lead to really dangerous consequences.

As a result, even if the longitudinal speed could be easily computed using already available sensors implemented on the vehicle, the problem is that it is very easy to make mistakes in the real time computation.

Thus to overcome this problem we can try not to compute the speed but to estimate it, basing our estimation on other signals that are directly measured with sensors. To do that, we speak about Virtual Sensing: we estimate a given quantity using as input other variables correctly computed.

Virtual Sensing is divided in [2]:

- *Analytical Techniques*: Estimation performed relying on approximated physical laws, which characterize the relationship of the direct measured quantities and the quantity to estimate. (Model-Based)
- *Empirical Techniques*: Estimation performed relying on *historical measurements and correlation* of the quantity to estimate with direct measured quantities. (Model-Less). The historical data of the interested quantity can be derived from actual measurement campaigns with a temporarily installed sensor.

This thesis work is aimed to the estimation of the longitudinal speed in high performance vehicles using a Virtual Sensing technique, using a specific Machine learning algorithm known as Neural Network, in order to improve the already existing techniques that are mainly too expensive or imprecise.

1.2 Thesis outline

The present work is organized as follows:

- **Chapter 1:** it is the introduction chapter in which information about the thesis motivation and thesis structure are provided to the reader. Furthermore is described the state of the art of Longitudinal speed estimation and road condition identification technologies implemented up to now, and is also present a section in which some information about machine learning and artificial neural networks are given, in order to introduce the reader to the main topic of this thesis.
- **Chapter 2:** in this chapter the work is presented from the operative point of view. It is described the overall system layout implemented to carry out both the longitudinal speed estimation and the road condition identification algorithms. In addition to this, it is also deeply explained how the different networks have been designed, with focus on the inputs used for the analysis and the different manoeuvres performed in DRY, WET and ICY conditions.
- **Chapter 3:** here starts the core of the thesis with the detailed description on how the longitudinal speed estimation has been performed. It begins with the characterization of the NARX network architecture employed for the purpose, and then continues with a presentation of the three networks used for the estimation, highlighting the hyperparameters employed and providing some pictures with the obtained results.
- **Chapter 4:** this chapter is organized similarly to the previous one. It presents the classification network employed for the road condition identification. First of all, the network structure is described, then the dataset used and the procedure for the feature extraction that has been implemented for the classification are presented. As a final step, the results obtained are reported.
- **Chapter 5:** this is the concluding chapter in which the results are summarized and some suggestions for future works and applications are reported.

1.3 Longitudinal speed Estimation – State of the Art

Every time we study a moving object, its velocity becomes an important parameter to be known if we want to understand deeply how it behaves in many different conditions, changing and defining as many boundary conditions as possible.

Knowing that, it is easy to understand that the more precise we are in evaluating the velocity of our system, the better will be our knowledge about it and its behaviours. Unfortunately, at the moment to obtain a precise and reliable measurement of the velocity we must use costly and bulky instruments, such as Kistler sensors, or very complex control systems that are not suitable for a mass production object, such a vehicle.

Anyway, many techniques have been developed in order to try to improve the measurement of the velocity without increasing too much the cost.

Here I list some of the most important and used methods.

Global Positioning System based.

GPS based systems are widely employed for their capability in providing information about the velocity heading of a vehicle. Of course the GPS system first of all provides the body's position anywhere on the globe, with the only disadvantage that its signal can be corrupted by atmospheric conditions, injected noise (called Selective Availability SA) and other noises. Anyway these problems can be easily overcome analysing signals coming from more satellites, using the so called Differential GPS (DGPS), that analyses at least three satellites signals to compute the body's position, in order to reduce the error. Furthermore the DGPS is able to compute the Doppler shift in the carrier-phase ranging signal between two different samples of time, that provides good velocity values.

GPS provides a three dimensional velocity vector of the vehicle considered, with an accuracy of 5 cm/s in each axis, considering a 1σ interval on a Gaussian distribution. These values are computed considering the error induced by SA, but applying some corrective actions such as Wide Area Augmentation System (WAAS) or coast guard differential corrections, the SA can be eliminated, except for some signal latency, improving the performances of the system [3].

To obtain the information about velocity a possible layout to be implemented on the vehicle is a couple of GPS antennas, as it is used in [3] and in many applications for aircraft heading computation.

The limitation of this approach is mainly related to the already mentioned problem of signal latency during broadcasting the corrections of coast guard differential or WAAS. Typically this latency is in the range of 5-10 seconds, which results in an additional 5 mm/s error in the velocity computation. Other error sources are the time delays due to the signal moving across the Ionosphere and Troposphere, but this error can be easily cancelled when two consecutive measurements are used to determine the rate of change of the phase of the carrier wave.

Anyway those errors can be easily overcome with new and stronger GPS signals, that are improving constantly, and this leads to the facts that better GPS signals associated with other on board sensors, can provide better vehicle states such as the velocity.

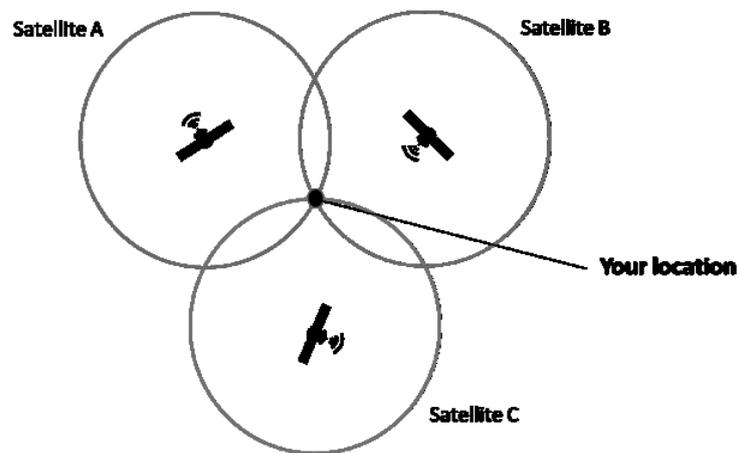


Figure 1. Schematic description on GPS working principle

Kalman Filtering

A proposed solution for the estimation of lateral and longitudinal velocity of the vehicle is the one proposed by Chu in [4] based on the online identification of the cornering stiffness using the Recursive Least Squares algorithm (RLS) and an Extended Kalman Filter algorithm (EKF).

Generally, a Kalman filtering is an algorithm able to retrieve an accurate estimation of certain variables, starting from measurements with noise estimating a joint probability distribution over the variables for each timeframe [24]. The Extended Kalman Filter is the nonlinear version of the Kalman filter, which typically linearizes about an estimate of the current mean and covariance [22].

Kalman filter is widely employed in many fields, such as econometrics, signal processing and navigation of vehicles. This algorithm works in two steps. The first one, also called

prediction step, consist in estimating the current state variables together with their uncertainties. Then we perform the measurement in the next time instant, and so the estimates are updated using a weighted average criterion: higher weight is given to the estimation with higher certainty. This process is recursive and one of its strong points is the computational speed: the Kalman algorithm only compares the estimation in instant t and in instant $t-1$, and no further information is needed.

The recursive least squares is an adaptive filter algorithm that recursively finds the coefficients that minimize a weighted linear least squares cost function relating to the input signals. This approach is in contrast to other algorithms such as the least mean squares (LMS) that aim to reduce the mean square error [23].

In [4] these 2 algorithms are implemented: EKF to estimate the lateral and longitudinal speed, RLS to estimate the tyre's cornering stiffness.

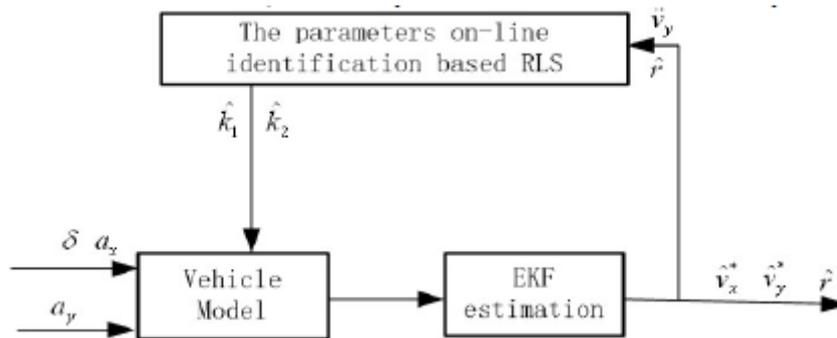


Figure 2. System used for the analysis in [4].

It is clear the role of the EKF: it is used to estimate in real time longitudinal, lateral velocity and yaw rate. This is one of the most common applications of a EKF for real time navigation and control of the vehicles.

Kalman filter is still one of the most powerful tools that can be used in order to estimate some uncertain parameters of the system we are studying, being a car, an aircraft or whatever. The difficulty of this approach is mainly related to the capacity of defining correctly the state equations, inputs and outputs of the system.

Another possibility to compute the longitudinal and lateral components of the speed, is given by the Adaptive Kalman filter, as it is presented in [25].

The drawback of using a common Kalman filter or an EKF is that the estimation of the vehicle states are highly accurate only in the case of constant noise. However, while the car is moving, a complex driving environment can bring uncertain noise disturbance over time (such as road roughness and wind disturbance), so there is the need to use an adaptive

algorithm with a certain tracking capability to estimate real-time. For this reason a good solution could be the application of an Adaptive Kalman Filter.

In classical Kalman filtering theory, the formulation of a filtering problem is defined knowing a priori the noises and disturbances affecting the system. Since it is very difficult to know precisely those information, using a Kalman filter with a wrong a priori information can lead to large estimation errors or even divergence of errors [25]. To solve these errors an Adaptive Kalman Filter can be used, adapting the Kalman filtering to the real data.

For a given nonlinear system, the state equations are always defined as:

$$\dot{x} = f(x, u) + w \quad (2)$$

$$y = h(x) + v \quad (3)$$

Where x is the state vector, y is the measurement vector, w and v are process noise and measurement noise respectively [25].

The typical adaptive Kalman filter procedure is divided in three steps:

- Time update;
- Measurement update;
- Noise estimation.

In the end, the application of an Adaptive Kalman filter can improve a lot the accuracy of the models [26], as it is possible to be seen also from the results presented in [25], where the estimation of the velocity is obtained with an high degree of precision.

Modular Nonlinear Observers

Nonlinear observers are used to take into account the nonlinear dynamics (mainly due to highly non-linear friction forces), and to obtain simple designs with few tuning knobs (as opposed to extended Kalman filter designs). Furthermore, the Riccati equation is avoided, so that we can implement the systems in low cost embedded computer units. Thus, this kind of solution for the longitudinal speed estimation is derived from the Kalman filtering algorithm: in this case the system works mainly with nonlinear quantities, that are computed using dedicated models [5].

To estimate the longitudinal velocity, a model that has the longitudinal acceleration as input is used in [5]. Then the velocity is computed knowing the rotational speed of the wheels and from this an observer that helps in the computation of the longitudinal speed is derived.

The problem is that if we base our computations on the wheel velocities, the risk of error is really high: the friction between tires and road could be highly nonlinear or also could happen that the wheel starts to slip or vice versa to lock, and this leads to possible errors in the estimation of the overall vehicle speed, increasing the risk of a wrong estimation.

Furthermore, a possible solution to the problem of measuring highly nonlinear quantities, such as tyre's friction forces, can be solved relying only on the measurement of an Inertial Measurement Unit (IMU). But, as it is stated by Hashemi in [27], relying only on the measured acceleration from the IMU and using a slip detection algorithm along with a proper stochastic estimator is not sufficient to guarantee the effectiveness of the kinematic-based velocity estimation approaches due to sensor noise (or bias) and uncertainties in the model. The main problem trying to estimate the longitudinal speed from the analysis of tire forces is that it is required to deal with road friction and many tire parameters which are time varying.

Thus, using vehicle kinematics and accelerations with implementation of an observer on tire forces is a reliable approach to tackle this problem and to estimate a vehicle speed.

This observer can be implemented considering the unknown road condition as a bounded uncertainty, then employing the estimated friction-independent tire forces for correcting the estimates, as presented in [27].

Model Predictive Control (MPC)

Using methods such as Kalman filters, EKF, AKF and observers implemented with nonlinear tire models, such as the LuGre friction model, is possible only under the hypothesis of not considering high-slip cases and non-Gaussian process and measurement noises, because those conditions can lead to mistakes in the estimation of the longitudinal speed. Therefore developing a robust measurement also in these conditions is required. A possible solution is presented in [28].

Recently, due to large improvements in the computational power, the popularity of some control algorithms is increased, such as the Model Predictive Control algorithm. It consists of an optimization in-the-loop algorithm, that results in highly optimized solutions. The drawback of the MPC is that it is computationally expensive, in particular if compared to some traditional algorithms such as PID or LQR techniques [28].

MPC models predict the change in the dependent variables of the modelled system that will be caused by changes in the independent variables. Independent variables that cannot be adjusted by the controller are used as disturbances. Dependent variables in these processes are other measurements that represent either control objectives or process constraints.

MPC uses the current plant measurements, the current dynamic state of the process, the MPC models, and the process variable targets and limits to calculate future changes in the dependent variables. These changes are calculated to hold the dependent variables close to target while honouring constraints on both independent and dependent variables. The MPC typically sends out only the first change in each independent variable to be implemented, and repeats the calculation when the next change is required [29].

Artificial Neural Networks

To solve all the problems we listed before, starting from the estimation error up to the computational effort required, we can use a Machine Learning algorithm. Those algorithms are much more powerful because they do not require any kind of vehicle or tire model but they just need to be fed with the signals coming from the sensors already implemented on the vehicle.

The working principle of those learning algorithms is very similar to the human learning process, as suggested by the name “Neural Network”. We, as humans, learn by experience: we try to do something and doing it we understand how we can improve it. Neural Networks work in the same way: we feed them with a huge quantity of inputs and we try to train them in the most widespread range of conditions possible, so that once they are implemented in real scenarios they can easily estimate the correct value of the velocity.

It is clear that exist different kinds of networks that can be implemented: for instance in this paper I will focus mainly on a Regressive and on a Classification Neural Network, because they are the best choice for our purpose. Anyway the base principle behind neural networks is the same for all the different kind.

We can say that ANNs can be compared to a professional rally driver [1]: they are very skilled in understanding the behaviour of the vehicle in real time. This because they have a really great experience in driving in all the possible conditions on tarmac, mud, snow, etc. As a consequence of that, they immediately understand if the vehicle is approaching the nonlinear conditions or not, and adapt their driving skills as a consequence.

Thus the rally drivers can be seen as black box models, in which we give some inputs like acceleration, wheel speed, yaw rate and others, and it provides us with an output that (in general) is the control of the vehicle.

This is the same done by neural networks, that, in our case, give us an output (the longitudinal velocity) that is required to better control the vehicle dynamics.

Before going on with the description of this estimation method, could be very useful to give a description of the behaviour of the speed in vehicles.

In vehicles the overall velocity of the body is given by composition of the four wheels velocities: this means that the module and the direction of the body velocity vector, indicated with \vec{V} and centred in the centre of mass of the vehicle, is given by the vectorial sum of the four wheels velocity vectors, as depicted in picture 1.1.1:

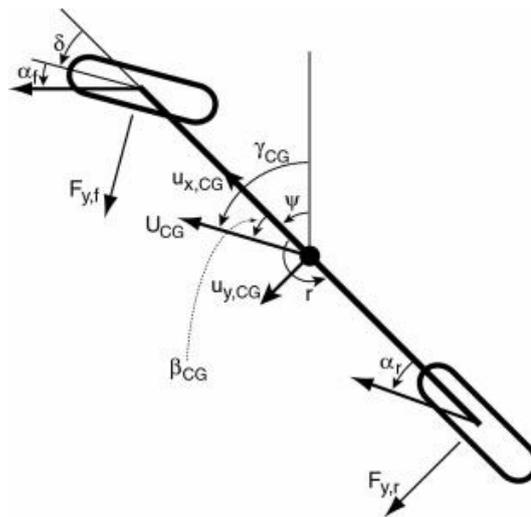


Figure 3. Bicycle model of a car

where U_{CG} is the centre of gravity velocity, u_x and u_y its components in the inertial reference frame, δ is the steering angle of the wheels, β_{CG} represents the sideslip angle of the vehicle and α the sideslip angle of the wheels. Just to recall, the sideslip angle is the angle between the velocity vector direction and the longitudinal direction of the body.

For sake of simplicity, in this picture we have used the bicycle model of a vehicle according to the simplified hypothesis given by Genta in [17], but anyway it is possible to see the two velocity vectors of the wheels and, computing the vectorial sum, retrieve the velocity vector of the CG.

Another assumption that we have considered during this work is that we are considering a four wheeled vehicle.

Obviously, this focus on the velocity of a vehicle does not want to be an exhaustive explanation of the topic, but only a brief recall of the main notions. Anyway, specifying these concepts was fundamental in order to introduce my thesis and to understand which are the most influencing factors to be considered when choosing the inputs to be used in the neural network: of course the four wheel speeds will be at the basis of my analysis.

1.4 Road Condition Identification – State of the Art

Nowadays, vehicles are equipped with a wide variety of sensors that adopted to improve the stability of the vehicle itself and at the same time allow to help the driver in having a better control during some difficult conditions. One of the typical examples is given by the TCS control system: it helps the driver in keeping the control of the car in difficult scenarios, like starting on a slope, or starting the vehicle when one wheel is on an icy patch of the road and so on. The secret of the TCS, is the possibility to rely on sensors that allow a precise and reliable estimation of the tire road friction coefficient and of the torque applied on each single wheel. In this way the systems recognises a possible dangerous situation and is able to react properly, avoiding the loss of control of the vehicle.

As described by Gustafsson [15], four different approaches to the tire-road friction estimation have been developed:

- Using the wheel slip: analysis of difference in wheel speed between driven and non-driven wheels.
- Use of optical sensors: they are installed at the very front of the vehicle, in order to be able to detect the reflections of the road and so to be able to detect (and slightly predict) the road condition or the presence of lubricants.
- Acoustic sensors to detect the noise of tiers.
- Strain and stress sensors inserted in the tire's tread. This is technique is very complex and expensive.

We can distinguish the Road condition identification techniques in two big families [30]:

The cause based: try to detect factors that affect the tyre–road friction coefficient and then attempt to predict what μ_{max} will be using, a tyre model or analytical theory. Although these approaches demonstrated high accuracy in experiments, they have three

main disadvantages. First, they require extra dedicated sensors such as lubricant or optical sensors. Second, they require accurate tyre models and special training software. Third, they tend to lose accuracy when conditions deviate from the conditions under which they were trained.

The effect based: focused on the effects on the vehicle or tyre due to the change of tyre-road friction. They include: acoustic approaches, tyre-tread deformation approaches, and slip-based approaches. Besides, the so-called model-based friction estimation can be classified as effect-based approach.

Here I list some of these systems and explain briefly how they work.

Slip-based tyre-road friction estimation

This is one of the main research field which scientist have deeply analysed in order to derive a robust estimation of the road conditions.

One interesting application is given by Gustafsson in [15]. Its work is based on the relationship between friction coefficient of the ground versus the tire slip, which definition has been already given by (1). A plot of friction coefficient versus slip allow to better understand how their relationship can change according to the ground conditions.

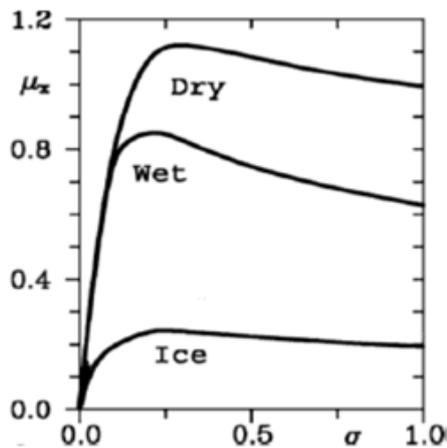


Figure 4. Friction coefficient versus slip.

This plot highlights the relationship among these two parameters in different road surface conditions. As it is expected the friction coefficient is decreasing in module moving from dry conditions to icy conditions. As a remark, the values reported in the picture refer to standard tires used on passenger cars [17]. If we consider high performances tires, such as the ones used in Formula 1 competitions, the friction coefficient can reach peak values of 1.8.

Figure 4 represents the relationship $\mu_x(\sigma)$ for different road conditions. It is important to underline that the values of this function depend on a different number of parameters such as type of tire, road conditions, speed and many others. Furthermore the values reported refer to standard tires used on common passenger cars, while if we consider tires employed in Formula 1 cars or, more in general, in sport cars, $\mu_x(\sigma)$ can reach peak values of 1.8 [17].

The idea behind the slip based road condition identification is to predict the maximum value of μ_x based on previous collected data in low friction and low slip conditions, when the acceleration is $< 0.2g$. However, two main problems with this approach. First, to measure small slips, especially during braking, in a practical setting is not very easy and requires appropriate filtering techniques. Second, the shape of slip curve at low friction demand is mainly determined by the tyre carcass stiffness rather than the road condition according to accepted tyre theory [30].

The innovation introduced by Gustafsson is that there is a relationship among the slope of tire characteristic in the linear region and the friction coefficient μ_x .

The curves present on the diagram are computed using the Pacejka model, also called *magic formula*. Assuming that we are working in the linear range of the curves we can compute the slope C (slip stiffness) of the curves at different vertical loads. Furthermore it is possible to notice that there is an offset δ near the origin of the plot, that means the slip is not zero when the driving force is zero. Estimating also a γ parameter for the detection of gravel soil (see [15] for more detailed explanation), we can rely on three different coefficients for the computation of μ_x : C , δ and γ . Then designing a classifier, we can distinguish three working ranges based on different values of the previous parameters:

Gravel ($\mu = 0,5$) if $\gamma > 0.027$.

Asphalt ($\mu > 0.8$) if $\gamma < 0.027$ and $C > 30$.

Snow/Ice ($\mu < 0.3$) if $\gamma < 0.027$ and $C < 30$.

After the classifier designed by Gustafsson, other scientist have tried to improve his work, such as computing the slope in braking conditions, in order to improve the accuracy of the estimation by using more friction and more slip. But this method introduced more uncertainties due to the need of correctly measure the brake pressure in the braking system of the vehicle.

Driving force Observer

The road friction coefficient can be computed if the driving force of the wheel is observable [14]. If we consider the wheel model depicted in figure 1.2.1, the equation of motion of the driven wheel is:

$$F_d = \frac{1}{r} (T - J_n \frac{d\omega}{dt}) \quad (4)$$

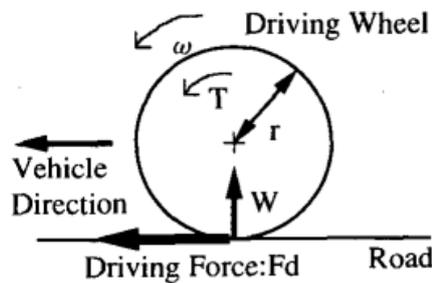


Figure 5. Single wheel model

where T , ω and J_n , are the motor's torque, wheel rotating velocity and wheel inertia, respectively. Knowing this equation the driving force can be computed by means of an observer, where typically a low pass filter is used in order to cut out the noise caused by taking the differential of wheel velocity signal. Once the driving force is known, knowing the vertical force that acts on the wheel we can retrieve the road friction coefficient, that is the proportionality coefficient between F_d and W [14].

Of course this is an *effect based* method, since it analyses the relationship between road friction and driving force of the vehicle.

Another example of application of the driving force observer method is provided by [14].

The actual friction coefficient is again computed using a driving force observer and applying

$$F_d = \mu N \quad (5)$$

Then, for the road condition identification it is also required the computation of the slip, and it is performed evaluating the slope of the $\mu_x(\sigma)$ plot, in order to evaluate the maximum friction applicable in the given road condition, to maximize the traction force exploitable.

The computation of the slope A is performed using:

$$\frac{\delta\mu}{\delta t} = A \frac{\delta\sigma}{\delta t} \quad (6)$$

To do so two algorithms can be used: the recursive least squares, already explained in 1.3, or the fixed trace (FT) algorithm. The results have proven that the FT algorithm is able to provide better results when the variation of the slip ratio is small, while in that condition the RLS method is less robust.

Vision systems on vehicles

During several years research has been focused on new methods to classify road conditions in an attempt to decrease accidents caused by slippery roads. There are nowadays many non-contact prototype sensors utilizing optical characteristics to make classifications of different road conditions. These prototype sensors can be classified into two distinct categories; those mounted on the vehicle and those mounted on a stationary weather station, i.e. on a road weather information stations (RWIS) [31].

The most common solutions using optical technologies based on non-contact sensors comprise cameras and Near InfraRed (NIR) techniques.

As it is reported in [31], to classify different road conditions with an optical non-contact sensor the variation in intensity of light scattered from the surface is explored. The scattering of light is dependent on the roughness of the illuminated surface and the absorption of the illuminated material. This makes it possible to distinguish the rough surfaces, dry asphalt and asphalt covered with snow, from the smoother surfaces, asphalt covered with water or clear ice solely based on the intensity. To distinguish between the two rough surfaces and the two smooth surfaces, respectively, the scattering taking place below the surface is considered. This scattering results in a spectral response which means that the amount of scattered light from the road material and the material on the road, i.e. water, ice and snow, depends on the wavelength of the light.

Considering this technology, one of the most important problems to be solved is the disturbance introduced by the external light that can interfere with the light used by the sensor for the measurement.

To handle such disturbances in [31] it is proposed a solution based on the design of a classification algorithm to analyse the variations in the scattered light intensity. There are two things that are of importance, the intensity of the disturbing light and the intensity distribution of the illuminating source. By knowing these it is possible to calculate the

amount of illuminating light that is reflected and then make a classification of the present road condition.

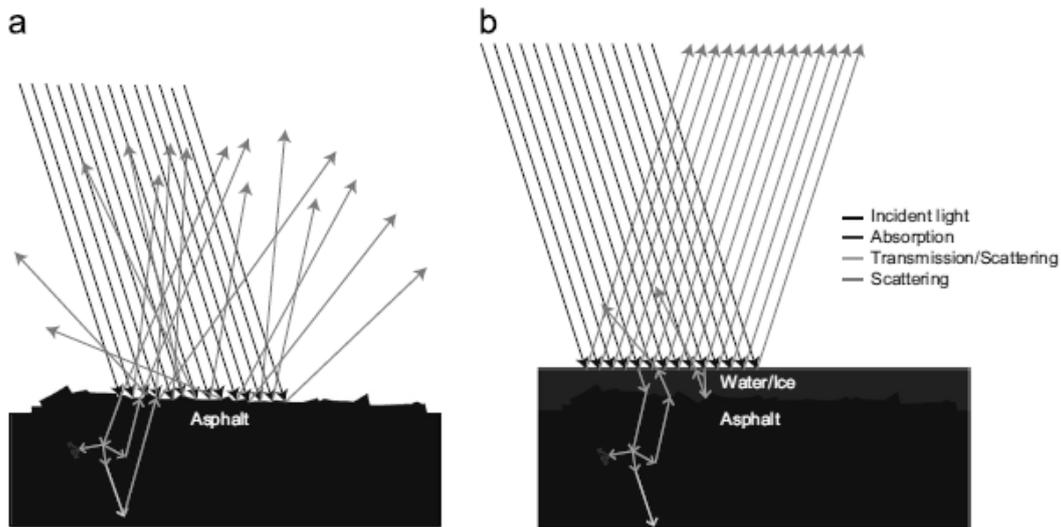


Figure 6. working principle of light scattering

To perform the classification, the light intensities of the analysed wavelengths are compared, and are used to compute some parameters useful to perform a fast image classification.

Besides NIR sensors, other techniques involving the spectroscopy have been developed.

The sensors employed emit light towards the road at different wavelengths, generally in the infrared range. According to the road status the light is reflected back and a detector catches this signal, that is processed and analysed in order to determine if the road is dry, wet or icy.

An interesting work about spectroscopy analysis has been presented in [32], where the authors have analysed the latent heat of water during freezing and melting, in order to detect icy patches on the road. Detecting these patches is very difficult, also because one of the variables that should be considered is the water freezing point temperature, which is unknown or uncertain due to the use of salt on the road, which has the ability to lower the freezing point temperature. For ice formation to occur on a wet road surface, the road surface temperature has to sink below the freezing point temperature. The phase transformation of water to ice is exothermic, which means heat energy is released when water freezes. This is referred to as the latent heat of fusion and is equivalent to 333.55 J/g at $0 \text{ }^\circ\text{C}$, which represents the amount of energy required to heat 1 g of water from 0 to 80

°C [32]. The heat release occurring during this exothermic reaction, can be measured by the infrared radiation analysis. The infrared is the part of the wavelength spectrum comprised between visible lights and microwaves, that means between 0.7 and 1000 μm . Typically, the warmer the object we are considering, the more is the radiation emitted, and the law that describes this physical phenomenon is the Stefan–Boltzmann law:

$$R = \varepsilon\sigma T^4 \quad (7)$$

Where R is the heat transfer ($W * m^{-2}$), ε is the emissivity, σ is the Boltzaman constant ($5.67*10^{-8} W * m^{-2} * k^{-4}$), and T is the temperature (K).

To carry out the measurement, An infrared thermometer receives thermal radiation and measures temperature on a detection element. A large variety of detection elements and techniques is currently available, such as photon and mosaic detectors. The objective is to absorb as much of the incoming radiation as possible. The detection element is often protected by a lens, which must not absorb heat radiation within the specific measurement wavelength spectrum. The lens is also critical for determining the angles at which incoming radiation is projected onto the detection element, and thereby determines the size of the measurement spot and field of view [32].

Radar systems on vehicles

Nowadays, with the raise of new technologies such as autonomous driving and driver-assistance features, there is the need of devices with a good range and rapidity of detection. For this reasons, the application of radar devices in vehicles is increased a lot, mainly to aid in detecting possible obstacles, and so improving the effectiveness of some new safety devices, like the Adaptive Cruise Control (ACC).

Thus, radar systems already implemented on vehicles can be a good solution to be implemented also for road condition identification. This solution has been the topic of researchers.

The problem of road condition identification using radar backscatter is faced in [33], where different test with radars mounted on stationary and moving platforms are performed.

The fundamental point is represented by the backscattering phenomenon. For a patch of a scattered road the power received by the radar is given by:

$$P_r = \frac{1}{(4\pi)^2} \int \frac{P_t G_t A_e \sigma_s}{R^4} dA \quad (8)$$

Where P_t is the radar transmit power, G_t is gain of the transmit antenna, A_e is the receiving antenna's effective area, σ_s is the elemental scattering cross section, and R is the scatterer-radar radial distance. G_t , A_e , σ_s are typically functions of elevation and azimuthal angles. The road condition's influence on P_r can be observed through σ_s , which is further impacted by properties such as complex permittivity, surface roughness, and inhomogeneities of the top-cover and subsurface [33].

Typically, radar scattering measurements are reported at 76 and 24 GHz, as it is reported in [34]. The results at 24 GHz show that snow, ice and water can change the scattering properties of the road. The measurement of backscattering ratios for different polarizations turned out to be the best method to detect low friction spots. This kind of differential measurement enables the elimination or reduction of the effects of most unknown parameters such as measurement distance, asphalt properties, weather conditions, etc., which affect absolute backscattering measurements. It was found that water changes the backscattering properties of asphalt more than there may be differences caused by various asphalt types. Therefore, water can be reliably detected [34].

1.5 Machine learning basics

Machine Learning (ML) is a computer science branch, which targets the possibility for machines (that means computers) to learn by themselves from the dataset that they have as input.

The first man who used this name was an American scientist, Arthur Lee Samuel in 1959, but nowadays the most accredited definition is the one given by Tom Michael Mitchell:

<< a program is able to learn from experience E with reference to some tasks T and to measurement of the performance P , if its performance, measured by P , in completing the task T , is increasing with E >>

Simply put, this means that a Machine Learning algorithms use computational methods to "learn" information directly from data without relying on a predetermined equation as a model.

This is translated in the fact that we do not need to develop many equations to say to the machine "what to do", but it is enough to give a set of inputs to the machine and it is able to work out the required task and furthermore it is able to improve its performance as the number of samples available for learning increases.

Machine Learning is based on two working principles[6]:

- Supervised approach: the aim of supervised machine learning is to build a model that makes predictions based on evidence in the presence of uncertainty. A supervised learning algorithm takes a known set of input data and known responses to the data (output) and trains a model to generate reasonable predictions for the response to new data.

To develop predictive models, it uses:

- Classification techniques: predicts discrete responses. This is the technique I used to classify if the road condition is dry, wet or icy.
 - Regression techniques: predicts continuous responses. The longitudinal velocity signal has been estimated using this approach.
- Unsupervised approach: the main difference with the previous approach lies in the fact that here we don't have any known output to take as reference. The main target of this approach is to find some hidden patterns or intrinsic structures in data. Clustering is the most common technique.

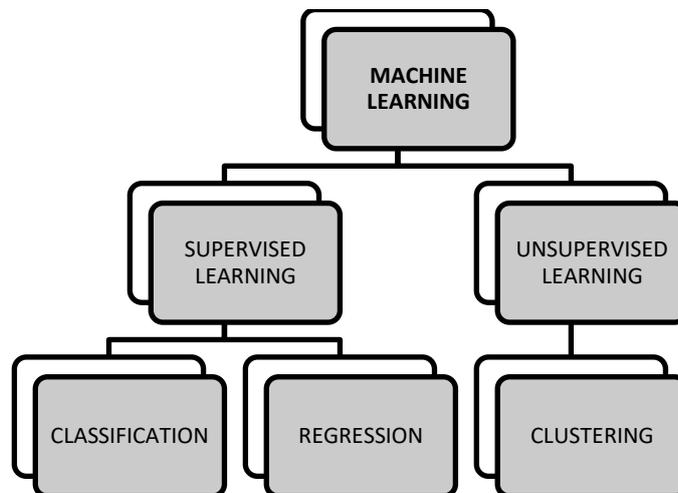


Figure 7. Schematic summary of the different machine learning algorithms

Another main advantage of the Machine Learning technique is that it can be used for many different purposes in different fields. And as a consequence, in the last years this method started to be studied and analysed also for automotive applications.

In particular, for what concern automotive field, ML can be used for improving the performances of the vehicle itself, or to focus some indirect aspect of the vehicle such as the aftermarket or the development process.

Just to give some examples, for what concern the improvement of the vehicle the main topics are Autonomous driving and Advanced Driving Assistance Systems (ADAS), and in this fields the pattern recognition capability of the Machine Learning is playing a key role.

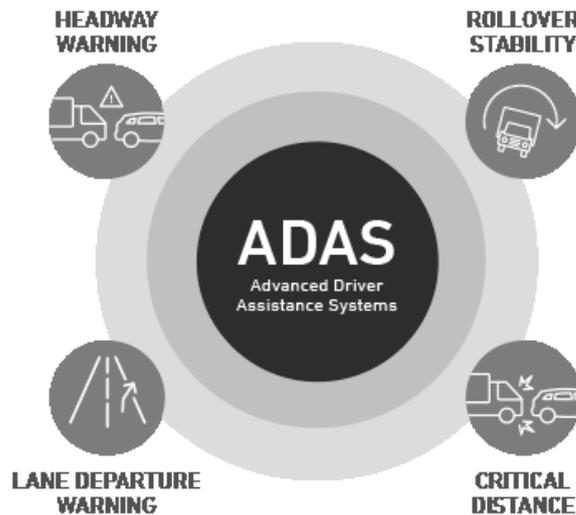


Figure 8. Some examples of ADAS applications

On the other side, for what concern the aftermarket and development innovation, we can focus on three main topics [2]:

- *Predictive maintenance*, suggesting how to adjust maintenance interval through monitoring and making failure prediction about different components like batteries, fuel pump or starter motor.
- *Customer feedback analysis*, suggesting how drive future vehicle designs through the analysis of social media activities, recording rewards and disappointments.
- *Enhancing vehicle user experience*, through the creation of user-specific profiles allowing personalization and suited personal assistance.

Thus, it is easy to understand that Machine Learning is a very powerful tool that can be used in different fields of application. The last question is: how can we decide which is the best algorithm to be implemented for our given purpose? In general we can say that there is no precise rule to be followed.

Finding the right algorithm is just trial and error—even highly experienced data scientists can't tell whether an algorithm will work without trying it out. But algorithm selection also

depends on the size and type of data you're working with, the insights you want to get from the data, and how those insights will be used.

1.5.1 Artificial Neural Networks

Artificial Neural Networks (ANNs) is an innovative architecture that is based on the same principle of neural connections present in animal brains: we have a net of many simple parallel processors that are strongly interconnected and this results in a distributed computational model that has an adaptive capacity.

As said before, Neural Networks are a typical application of supervised learning [6]: it is trained by iteratively modifying the strengths of the connections so that given inputs map to the correct response.

They can be used:

- For modeling highly nonlinear systems
- When data is available incrementally and you wish to constantly update the model
- When there could be unexpected changes in your input data
- When model interpretability is not a key concern

ANNs, as said before, can be used for many different applications, but whichever application we are considering, this algorithm will always be based on an elementary unit which is called neuron, the same as our brain.

This artificial neuron concept was proposed for the first time by W.S. McCulloch and Walter Pitts in *"A logical calculus of the ideas immanent in nervous activity"* (1943), where they managed to describe a linear combiner which was fed with multiple binary data in input and it was able to give single binary data in output. With the correct amount of data this machine was able to compute simple Boolean functions.

Anyway this machine was still not able to learn from experience, that is the key feature of ANNs.

The first hypothesis of learning algorithms were introduced by D.O. Webb in its work *"The organization of behaviour"* (1949), but the first real learning algorithm was developed in 1958 by F. Rosenblatt, who described in his book *"Psychological review"* the first scheme of Neural Network, that was called *"Perceptron"*.

The main difference between the Perceptron and the combiner of McCulloch, was that the Perceptron was characterized by variable synaptic weights, that allowed the Perceptron to learn .

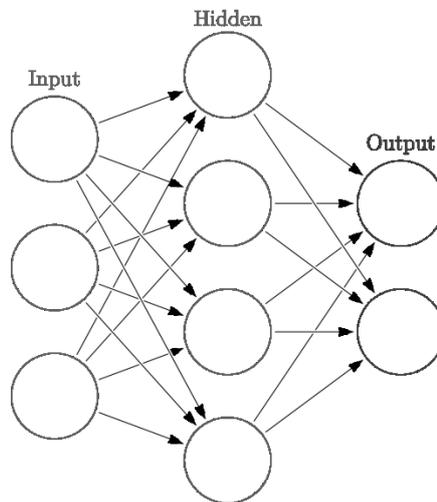


Figure 9. Layered structure of a network

The results obtained by Rosenblatt spread a lot of optimism in the scientific community, but the research was not able to go too much further in this field and we need to wait the last decades of the twentieth century to see again an increase in the development of the artificial intelligence field. In this period (1986) scientists were able to improve the capabilities of neural networks thanks to the introduction of the *back propagation* algorithm (*BP*).

It consist in a learning technique based on examples, which is a generalization of the Perceptron.

Thus, moving to a more technical description, Artificial Neural Networks are one of the available algorithms of Machine Learning. [2] Their structure is composed by a connection of many processors that have an elementary computational capability, called *artificial neurons* or *nodes*.

With the exception of the input layer, each neuron receives signals from the neurons of the previous layer linearly weighted by the interconnect values between neurons. The neuron then produces its output signal by passing the summed signal through a sigmoid function.

To each single node is assigned a weight, that is fundamental to understand how much that neuron can influence the neurons connected to it according to the strength of their

connection, that is the same process we can notice in neurobiology, analysing the communication between animal neurons.

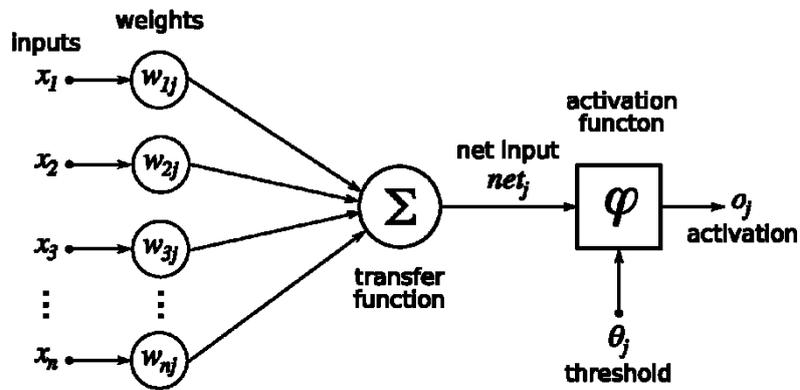


Figure 10. Schematic representation of an artificial neuron

Signals flow into the input layer, pass through the hidden layers, and arrive at the output layer.

[2] Artificial neurons are composed by two different elements:

- The *Propagation Function*: It combines the inputs coming from the previous layers accordingly to their weights and determines the excitation level of the neuron ;

$$net_j = \sum_{i=1}^I (x_i w_{i,j}) \quad (9)$$

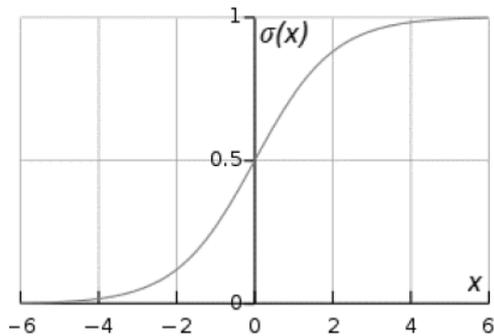
- The *Activation Function*: it activates the neuron when the propagation function exceeds a defined threshold, and doing so generates and limits the output of the neuron, that usually is comprised in $[0, 1]$ or $[-1, +1]$ range of values.

$$out_j = f\left(\sum_{i=1}^I (x_i w_{i,j}) + b_j\right) \quad (10)$$

So the activation function makes the neuron work as a chip: if the excitation computed by the propagation function overcomes a certain value, then the activation function gives output “1” and the neuron can compute the output. Vice versa, the result is “0” andso the neuron does not compute the output.

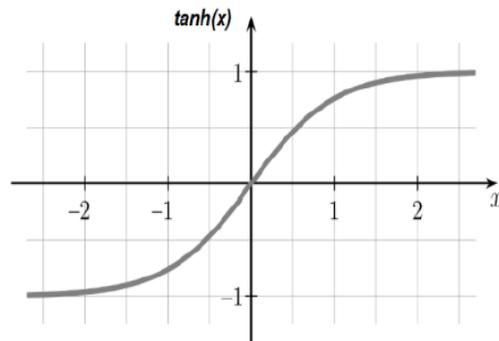
It is a binary function: the neuron is firing or not.

Where I number of neurons in the previous layer, i the neuron in the previous layer, j the considered neuron, x_i the input from i -th neuron in the previous layer, w the connection weight, net the output of propagation function, out the output of the neuron and b the bias. The last parameter, the *bias*, accordingly with its value, can increase or decrease the net input to the activation Function.



$$\sigma(x) = \frac{1}{1 + e^{-x}}$$

Figure 7a. Sigmoid function and its equation.
Range [0,1]



$$\tanh(x) = 2\sigma(2x) - 1$$

Figure 7b. Tanh function and its equation.
Range [-1, 1]

Figure 11. Examples of activation functions

It is worth to highlight that the Activation function must be a *nonlinear* function, because only in that case the network is able to solve non trivial problems using a small amount of neurons.

Once the structure of the network and the working principle of neurons are known, it is necessary to focus on the learning algorithms.

Generically speaking, there is not a single possible algorithm that fits all the scenarios, but it is necessary to choose the correct one for our purpose.

Anyway we can distinguish two big families of learning algorithms:

- *Feedforward ANN* [8][9]:

They are often called *Multilayer Perceptron (MLPs)*, too. With this structure, the signal is propagated from the input layer up to the output layer without any cyclic or traverse connection, with each neuron of a layer that is connected to every neuron of the following layer. This explains why it is called “feed forward”. One of the most used algorithms in a feedforward structure is the *Backpropagation algorithm*. With this

technique, the net compares the output with a certain desired value and computes the error as difference of the two values. The purpose is to reduce the error.

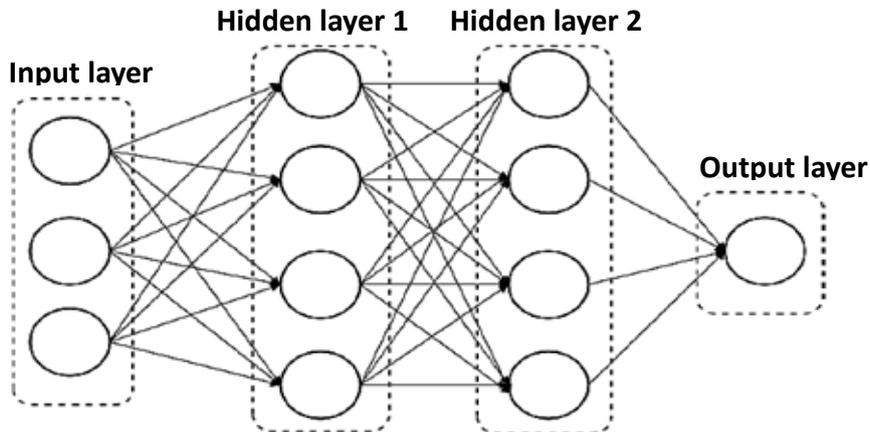


Figure 12. Feedforward ANN

- *Recursive ANN [10]:*

With *recursive* we indicate a kind of network that works on a structure, like a grid, string, graph or sequence. It operates on a sequence of vectors x_1, x_2, \dots, x_t , and this introduces a notion of time in the model, as each vector is located in a slot of time space. At each time step, the recurrent neural network computes a function of the current input and the activation or the output from the previous time step.

The basic configuration of RNN is the so called Simple Recurrent Network, that was firstly used by Jeff Elman in 1990 [11]. It has a very simple structure: a three layered NN, with a set of context units. The hidden layer is connected to the context units with a fixed unitary weight. At each time step, the input is fed-forward and the learning rule is applied, so the fixed back connections can save a copy of the previous values in the hidden units to the context units, since they propagate over the connections before the learning rule is applied. To improve the performances of Elman's RNN, we can find other architectures like *Long Short Term Memory (LSTM) Neural Network*, which are often very effective but very computationally expensive at the same time, or other more convenient structures like *Recurrent Multi-Layer Perceptron (RMLP)*. Another possible solution are the *Nonlinear Auto-Regressive network with exogenous input (NARX)*. This is the architecture chosen for this work.

NARX ANNs can be effectively used in the modeling of nonlinear dynamic systems. In fact, the output of the NARX networks can be the estimate of the output of the considered nonlinear dynamic system.

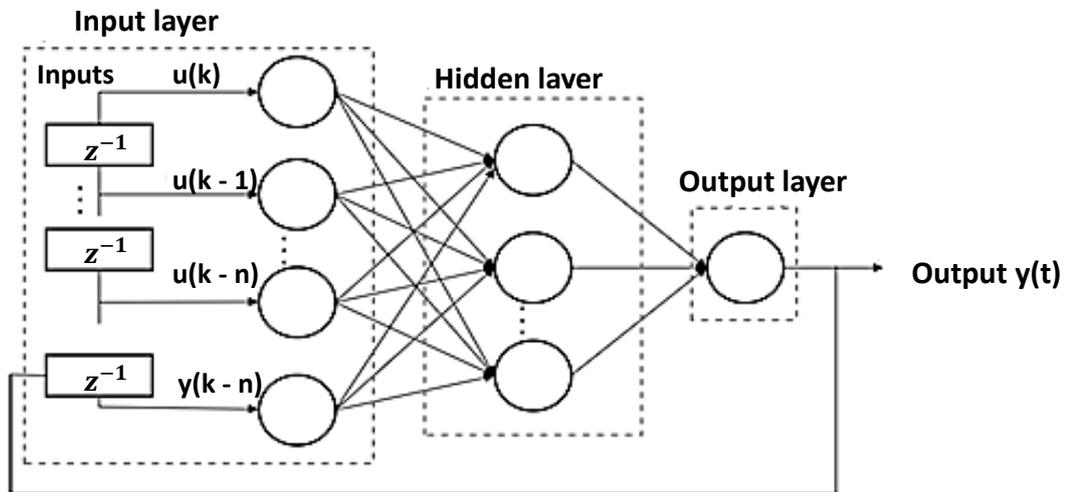


Figure 13. NARX ANN structure

Once the network is designed in terms of number of layers, neurons and architecture, the last step to be done is to train the network. To do this we can choose among many different training processes.

The one used in this work is the Levenberg-Marquardt (LM) backpropagation.

As it is described by Levenberg and Marquardt in their original works [12], [13] this process is able to update weights and bias values according to the Levenberg-Marquardt optimization, also known as the damped least-squares (DLS) method, which is used to solve non-linear least squares problems.

In the same articles, it is possible to read a detailed application of this process.

2 Overall system layout

In this section the structure of the system used for the estimation is presented and explained.

The system is divided in two main parts, as depicted in **Figure 14**

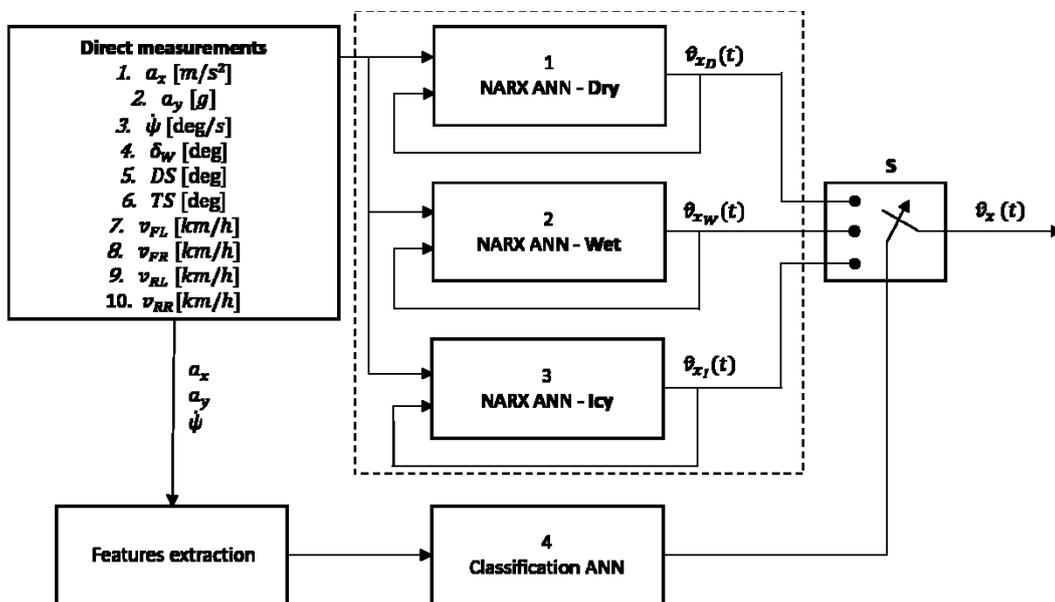


Figure 14. System layout

As it is possible to notice from the Figure 14, the system is composed by two sections:

1. The first one is characterized by the presence of Recursive ANNs: there are three networks, each one of them estimating in real time the longitudinal velocity in a different road condition (dry, wet, icy).
2. The second group is dedicated to the classification, that means understanding the actual road condition and so deciding which of three networks use for the real time estimation. The classification is performed working on seven main inputs, longitudinal and lateral acceleration, yaw rate and the four wheel speeds. They are subdivided in buffers of 10 seconds and for each slot we apply a Feature Extraction, retrieving information about RMS, Average, Root Mean Square Value and others.

2.1 Test conditions and dataset

As it was said in the introduction, this is a project developed by Politecnico of Turin in partnership with Automobili Lamborghini S.p.A, so the dataset available was provided by the engineers of the car company.

In order to obtain reliable results with the train of the ANN, there is the need to give as input as much as possible driving cases, in order to cover the maximum possible range of manoeuvres.

To do so, the engineers have made a lot of test with different drivers, including professional drivers, testing engineers, common drivers and so on. Furthermore all test have been performed using summer and winter tires, enabling and disabling the control systems, choosing different driving modes of the vehicle, new and worn tires.

They have carried out a lot of manoeuvres in different road conditions, and below is possible to find some of the most important [18], [19]:

- *Acceleration and breaking while cornering:*
Helpful to understand the behaviour of the vehicle and its yaw stability when it is submitted to heavy load transfer
- *Double lane change:*
To analyse road holding capabilities of the vehicle
- *Steady state cornering:*
To study the understeering and oversteering dynamics, having the steering as function of lateral acceleration. Generally this manoeuvre can be performed in different ways, like at constant lateral acceleration or at constant steer with throttle ramp
- *Sine sweep:*
To analyse the resonance frequency of the vehicle. This is done applying steer input at different frequencies
- *Steering ramp and Step steer:*
To study the transient response of lateral acceleration and yaw rate while steering
- *Laps along handling circuits*
Mainly performed in Nardò, Vizzola and Rosengarten circuits.

Lamborghini gave us also the possibility to use the required instrumentation and the vehicles provided with all the sensors for gathering the data.

The inputs for our estimation were collected by Data Logger, that recorded the ten input signals from the CAN-Bus of the vehicle. The ten input signal are listed in the table below:

| # | Type | Task | Parameter | Symbol | unit |
|----|-------|------|---------------------------|--------------|---------------------|
| 1 | Input | R&C | Longitudinal acceleration | a_x | [m/s ²] |
| 2 | Input | R&C | Lateral acceleration | a_y | [G] |
| 3 | Input | R&C | Yaw rate | $\dot{\psi}$ | [deg/s] |
| 4 | Input | R | Steering wheel angle | SW | [deg] |
| 5 | Input | R | Dynamic steer | DS | [deg] |
| 6 | Input | R | Total steer | TS | [deg] |
| 7 | Input | R&C | FL wheel speed | v_{FL} | [Km/h] |
| 8 | Input | R&C | FR wheel speed | v_{FR} | [Km/h] |
| 9 | Input | R&C | RL wheel speed | v_{RL} | [Km/h] |
| 10 | Input | R&C | FH wheel speed | v_{RR} | [Km/h] |

Figure 15. 10 inputs

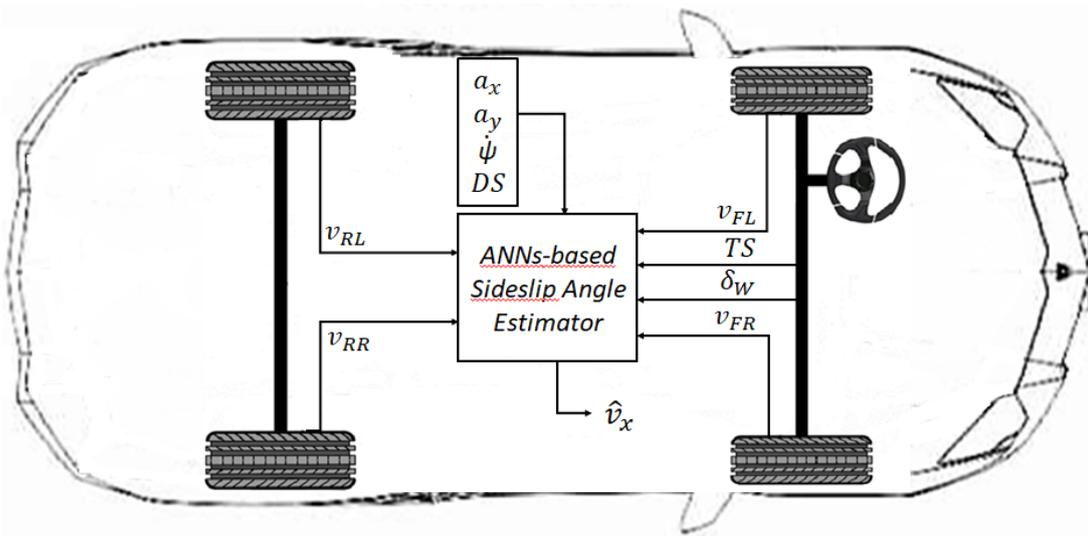


Figure 16. Schematic representation of the on-board layout

The signals transmitted by the CAN-Bus where obtained by standard sensors:

- Longitudinal accelerometer (from ESP)
- Lateral accelerometer (from ESP)
- Yaw rate sensor (from ESP)
- Vehicle speed sensor (from ESP)
- Front steering angle sensor

- Rear steering angle sensor
- Wheels speeds sensors (from ESP)

RACELOGIC electronic system was employed to measure, record and display the data from the moving vehicle [2]. Those signals need to be filtered and sampled at 10 Hz. This job is described in [1].

When the data are ready, it is possible to start with the training phase of the neural network, which is fed by the 10 inputs listed before plus the Longitudinal velocity signal measured with the Kistler sensor, that is used as reference by the network.

Then, the following step is to start the Validation phase: in this step we test the ability of the ANN to estimate the velocity, so we feed it only with the 10 inputs and without the Kistler reference.

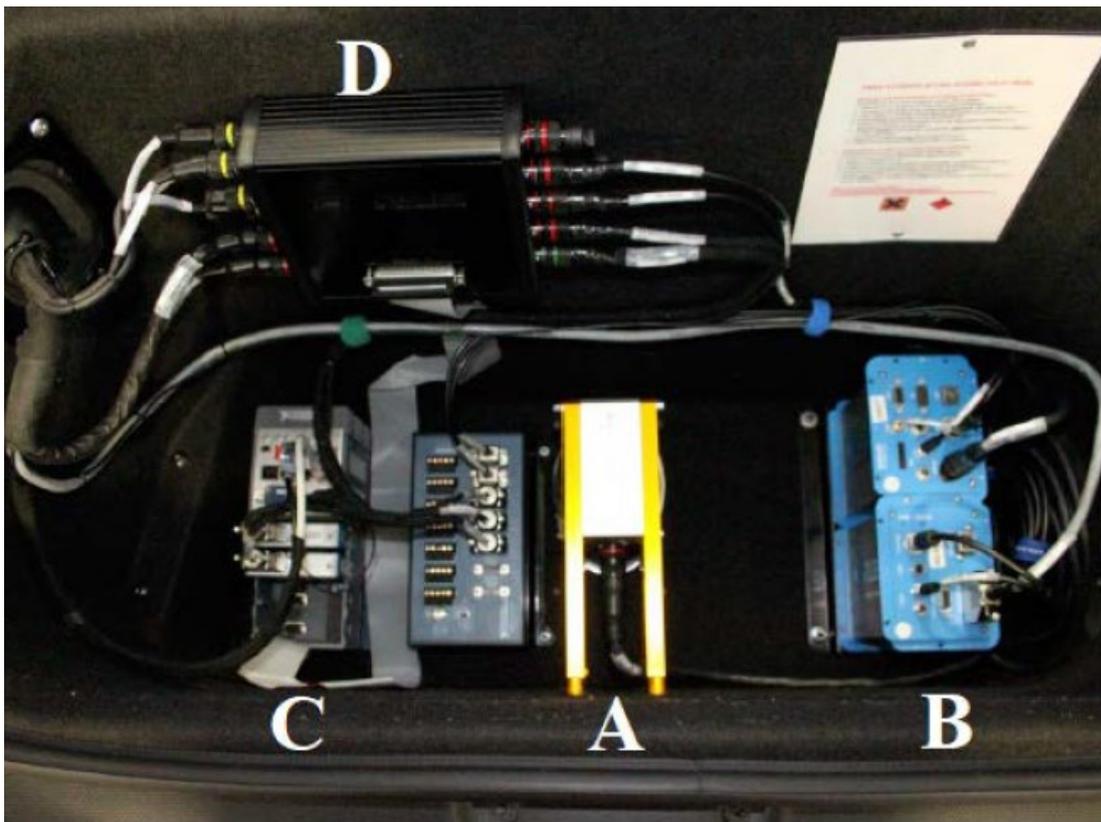


Figure 17. Longitudinal velocity sensor layout. A) Optical sensor and lamp. B) Image correlation processor. C) Rapid prototyping ECU.

In Appendix A are listed the different manoeuvres considered in this work.

3 Longitudinal speed estimation

The target of the present work is to improve the reliability of the longitudinal speed estimation, in order to overcome the typical issue of all the methods implemented up to now: the difficulty in estimating the speed in manoeuvres that are not present in the input dataset.

This solution has been carried out using a *Recurrent ANN* because it has the advantage of considering not only the input signals in a given time instant, but also the past ones. This is really helpful in recognizing particular manoeuvres, because they are the result of a certain sequence of events.

3.1 NARX architecture

NARX is the acronym of Non-linear Auto-Regressive network with Exogenous input, where exogenous means external. This is a kind of recursive network which is mainly used for predictions of dynamic systems. To do so, it is based on a recursive structure where the output of a given time instant is used as input for the following time instant.

Mathematically speaking we can describe it in this way:

$$y(t)=f(y(t-1),\dots,y(t-n),u(t-1),\dots,u(t-n)) \quad (11)$$

The equation describes very well that the output at a given time instant t is a function of all the input in the previous instants and also of all the output previously computed.

The architecture is a layered structure, and it is composed by:

- a sigmoid function in the hidden layer, that gives a certain weight to the communication between neurons;
- a transfer function in the output layer, which is required to activate the system and compute the output when the sigmoid function overcomes a predefined threshold;

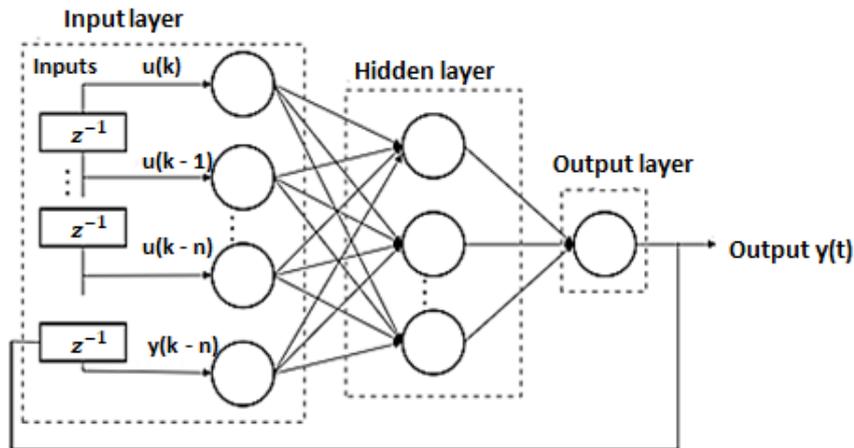


Figure 18. Typical structure of a NARX network

From Figure 18 it is also possible to notice the presence of buffers, denoted by the letter “n”: they represent the quantity of outputs and inputs of the previous instants need to be stored in order to obtain the desired output.

This structure can be used in two different configurations.

Typically, during training phase, it is used in open loop configuration, as depicted in Figure 19.

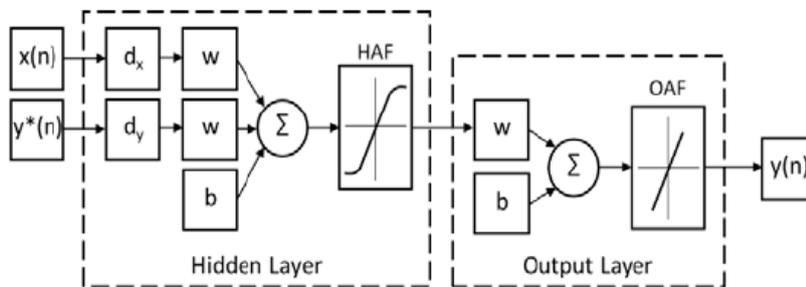


Figure 19. Open loop configuration

This structure is preferred during training because the output is not fed back, but it is used as a reference for the following computations, so that the network can literally train itself in estimating correctly the longitudinal speed, also with the help of the reference output.

Then, after this starting phase of training, we can perform the *Prediction* phase. It is generally done using a closed loop configuration, so that the net cannot rely anymore on the reference output, but uses it as input for the correct estimation.

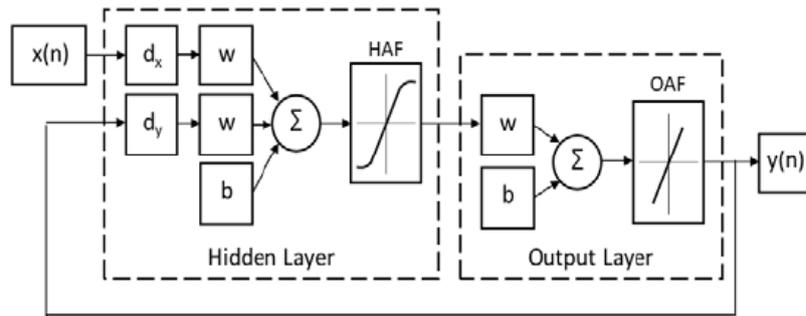


Figure 20. Closed loop configuration

The typical working steps are the train, validation and test performed in open loop and then the prediction phase in closed loop.

3.2 Training and validation – parameter setting

As previously described in chapter 3.1, the procedure followed to carry out the estimation of the longitudinal speed is divided in Training in open loop configuration and Validation in closed loop. This has been done using Matlab and Simulink softwares.

The parameter setting is the key step to be done, in which our network is designed. Parameter setting means choosing the number of layers, the number of neurons, the training algorithm and the number of delays to be used. This is a fundamental phase of the design, because we need to design correctly the network in order to avoid to use too much computational energy and obtaining anyway a weak estimation of the speed. Another issue that could happen is the so called “*overfitting*” or “*overlearning*”[20]: this is a phenomenon by which our network reaches really good values in performances, but anyway the estimation is not satisfactory. This is mainly due to the fact that the network memorizes the training points more and more, while its test response is becoming worse. This problem can be solved using the *early stopping* technique, which suggest to stop the training once we reach the minimum of the learning curve of the network, that is usually indicated with a circle on the figure, as it can be seen in Figure 21.

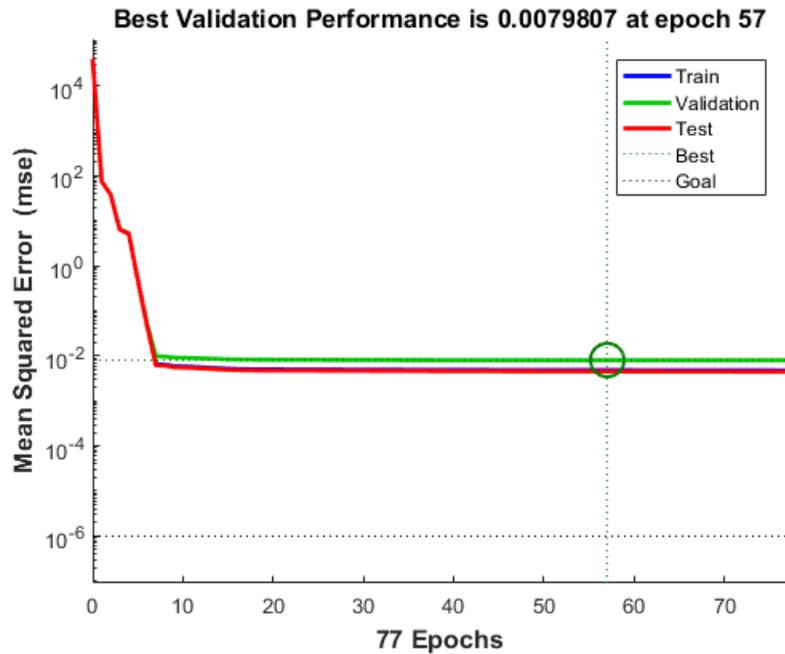


Figure 21. Example of learning curve.

As it can be seen, the best validation performance is indicated by a circle.

To go more in depth with those Hyperparameters previously defined, we start describing the training functions. As said before we can choose among different training functions:

- “*Trainscg*” updates values according to Scaled Conjugate Gradient method. It is a backpropagation training function that uses gradient derivatives. [2]
- “*Trainlm*” updates values according to Levenberg-Marquadt optimization. This is the fastest algorithm but requires more memory than the others. Is a backpropagation training function that uses Jacobian derivatives. [2]
- “*Trainbr*” updates values according to Levenberg-Marquadt optimization. It minimizes a combination of squared errors and weights and so determine the better combination to produce a network able to generalize. This process is called Bayesian regularization. is a backpropagation training function that uses Jacobian derivatives. [2]

Another important aspect to be considered is the *Network performance function*. This is the function we need to use in order to compute the error and so to evaluate the behaviour of the network. Generally speaking you can choose among different performance functions or even customize some functions for your specific applications. In this work we used the Mean Squared Error (MSE) function.

In statistics, the mean squared error (MSE) or mean squared deviation (MSD) of an estimator measures the average of the squares of the errors—that is, the average squared difference between the estimated values and what is estimated [21].

In other words, computes the squared error of our estimation with respect to a certain reference target and then computes the mean. It is similar to the regression problem in statistics.

For what concern the data, we have also to manage the subdivision of the dataset for the Training phase. For this work, it was chosen to use 70% - 15% - 15% subdivision:

- The *Training set* will be used for training, specifically to compute gradient and updates weights and bias. This correspond to 70% of data available in the dataset.
- The *Validation set* will be used to assure that network is generalizing and to stop training before Overfitting (15%). The error on the validation set is monitored during training process; it normally decreases during the initial phase of training, as does training set error. However, when the network begins to overfit the data its value typically increases.
- The *Test set (15%)* will be used as a completely independent test of network generalization. The error on the test set is not used during training but is employed to compare different models. However, it can be useful to monitor its error during training because if it reaches a minimum a significantly different iteration respect to validation error, this might indicate a poor division of data set. [2]

This is about the hyperparameters selection and definition. But there is still one element to be correctly defined, which is the dataset for training.

Of course it is of paramount importance to obtain good results in the estimation, the problem is which manoeuvres should be used for training and which not.

Ideally we should use all the manoeuvres at our disposal, but the risk is to increase too much the computational power required. So typically it is better to start with few manoeuvres in input and then if it is necessary to increase the number until we reach satisfactory results.

As it can be understood, both the hyperparameters setting and the training dataset are procedures that must be carried out in a trial and error procedure that is performed iteratively. Furthermore the same network should be trained more than once, because the

preliminary random assignment of weights and bias, could result in different performances of our ANN.

For what concerns the present work, it has been chosen to train and validate three different neural networks, one for each road condition (DRY, WET, ICE). It could also be possible to train one single network for all the three conditions, but the complexity of the network would be too high and furthermore the results obtained were not satisfactory, so this possibility was abandoned.

The following paragraphs will show in detail the characteristics of each one of the three networks.

3.2.1 DRY conditions

In this paragraph the first section of the overall system layout presented in chapter 2 is presented. In Figure 22 is depicted a visual correspondence of the DRY network location inside the overall system described in this work

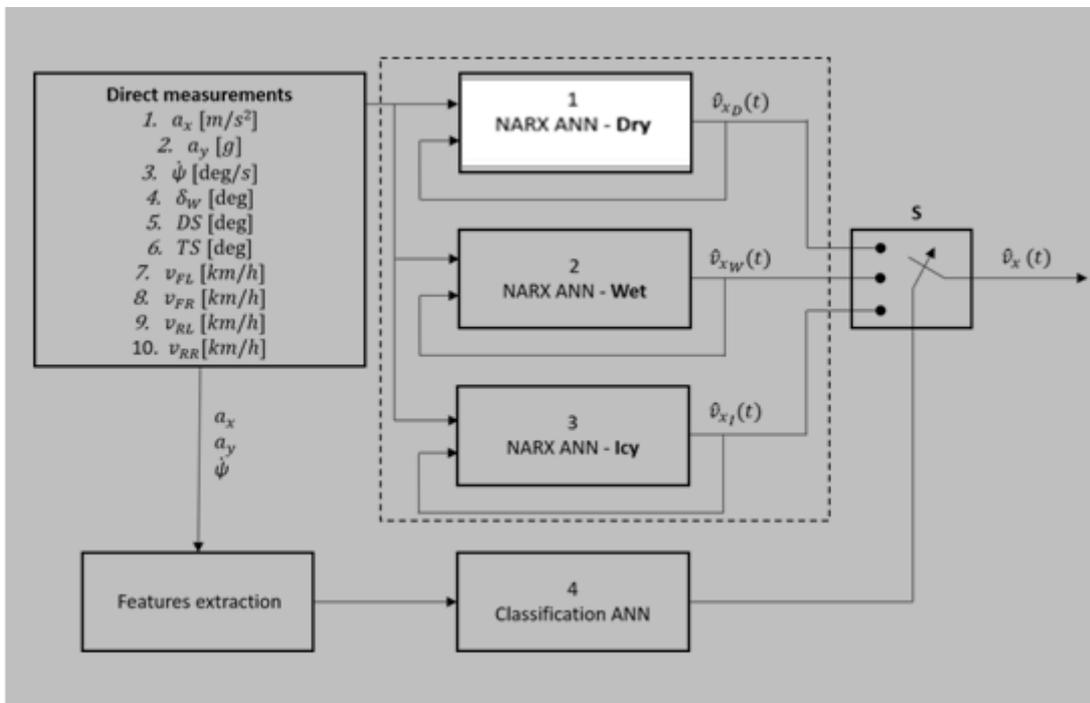


Figure 22. NARX ANN for DRY conditions in the overall system.

The network structure used in this testing conditions are summarized in Table 1

| | |
|---------------------|---------|
| Input delays | 2 |
| Feedback delays | 2 |
| Hidden layer size | 40 |
| Training function | trainlm |
| Total manoeuvres | 45 |
| Training manoeuvres | 7 |

Table 1. Hyperparameters defining the structure of the network used for the speed estimation in DRY conditions

This structure has been chosen after a trial and error phase, in which different number of neurons, delays and training functions were tested. This one was the synthesis of the previous tests, and of course it is the one that provided the best results.

Here below it is illustrated the results for every manoeuvre and also some of them in single plot in order to allow the reader to have a better understanding of the precision in estimation capability of our network.

In Figure 23 are represented the results of the estimation performed using the neural network, represented by the red line, compared to the reference measurement given by the Kistler camera, represented by the blue line.

In order to better understand the performance of the estimation with respect to the Kistler measurement, above the plots are listed the errors computed. In the list is also present the error between the Kistler and the ESP measurement, which is the old estimation technique. In this way we can highlight the improvement in the error estimation given by the new ANN technology.

For what concern the errors in the picture below, it is represented the average error in the first row, computed as difference between our estimation and the reference one divided by the duration time of the manoeuvre. This is computed both for the ANN (left in the row) and the ESP estimation (right in the row). The formulas used for the computation of the errors are:

$$\text{Average error ANN} = \frac{1}{T} \times \sum_i (estVel_i - velX_i) \quad (12)$$

$$\text{Average error ESP} = \frac{1}{T} \times \sum_i (Esp_V_signal_i - velX_i) \quad (13)$$

where, T represents the duration in time (s) of the manoeuvre, and the summation is performed analysing the difference in value between the two measurement in every i -th second of the manoeuvre.

In the second row, is written the maximum error computed during the estimation of every manoeuvre, again organised as ANN on the left and ESP on the right. The formulas used for the computation of the maximum error are:

$$\text{Maximum Error ANN} = \max[\text{estVel}_i - \text{velX}_i] \quad (14)$$

$$\text{Maximum Error ESP} = \max[\text{Esp}_V\text{signal}_i - \text{velX}_i] \quad (15)$$

As a further remark, is useful to say that for both the average and the maximum error we have:

estVel = the speed estimated by the ANN;

velX = the speed measured by the reference Kistler camera;

$\text{Esp}_V\text{Signal}$ = the speed measured from the ESP signal.

From the results depicted in Figure 23 is possible to see that in the majority of the manoeuvres the improvements in terms of average and maximum error are remarkable, reaching peaks of improvement higher than 50% in some profiles. On the other side, it is also possible to find some profiles in which the error obtained using the ANN is a little bit worse than the one obtained using the ESP signal. This is explained considering that typically these profiles correspond to manoeuvres in which the speed goes to zero in some instants. Since it is a system that receives ten inputs per time and these inputs are very different one from the other, if the vehicle is stopped, that means the speed is zero, it could happen that one (or more) of the inputs is different from zero, even slightly, and this can lead to wrong information going to the network and so can lead to an estimation of the speed different from zero.

This typically happens when the vehicle stops at the paddock of the racetrack, without shutting off the engine, during the record of a certain manoeuvre. The result is the typical profile with a lot of peaks that correspond to the manoeuvre performed by the pilot, separated by intervals in which the speed is zero, because the vehicle is stopped in the paddock.

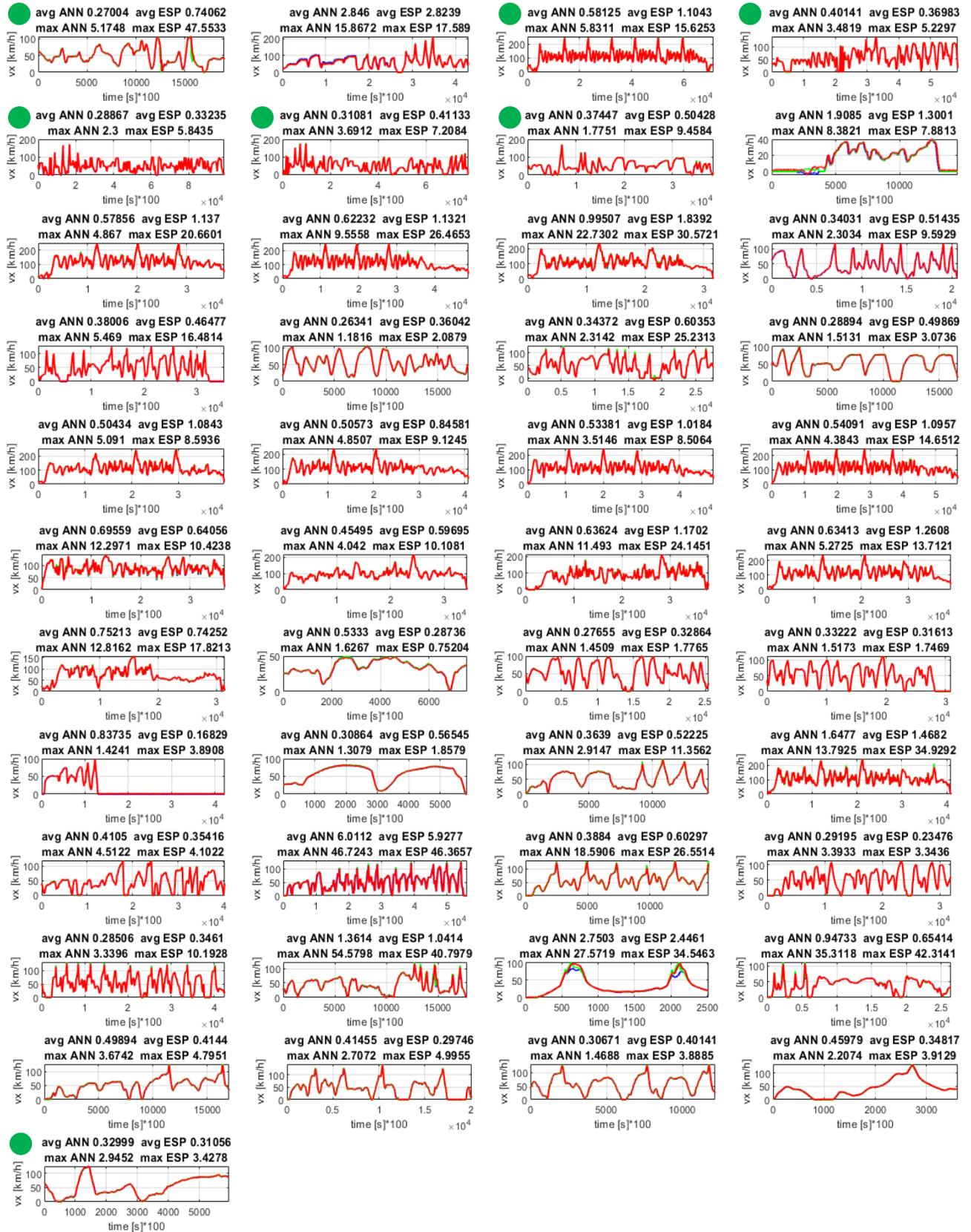


Figure 23. Results obtained in DRY conditions.

As it is possible to see from the picture the difference between Kistler and ANN is minimal, that means that the error in the ANN's estimation is really low. On the left column above every plot is represented the average error and maximum error of ANN and ESP, compared to the Kistler reference.

This is one of the main problems encountered during the simulations performed and one possible solution (and suggestion for future works) to improve the estimation is to filter the data in input and allow the network to recognize these events and impose that the speed should be zero when certain input parameters go to zero.

The last useful parameters to be given are the Total error and the maximum average error computed considering all the profiles in dry conditions. The results obtained are again compared among ANN and ESP, as previously done. The values are:

(ANN) | MAX_AVGERROR = 6.0112 km/h | TOT_ERROR= 0.7735 km/h

(ESP) | MAX_AVGERROR = 5.9277 km/h | TOT_ERROR= 0.8806 km/h

In this case, the formulas used are:

$$\text{MAX_AVGERROR} = \max(\text{Average error ANN}_i) \quad (16)$$

$$\text{TOT_ERROR} = \frac{1}{\text{number of profiles}} \times (\sum_i \text{Average error ANN}_i) \quad (17)$$

In the following, some pictures of relevant manoeuvres in dry conditions are reported.

Together with the results obtained with the recursive ANN in the estimation of the longitudinal speed, compared with the kistler reference speed and the ESP computation, the inputs used for the estimation are also reported.

The first plot represents manoeuvre "AS". This manoeuvre has been performed in sport driving mode, with the electronic stability control switched off.

In this profile are represented three testing phases. The first one is done at 80 km/h, steering < 90 degrees both to the left and to the right, with second gear engaged.

The second one is equal to the first one but the steering is at 45 degrees.

The third phase is performed at 100 km/h, steering < 30 degrees, with second and third gears engaged. As last remark, a new set of tires have been installed before starting this testing manoeuvre.

The second plot represents manoeuvre "A" in DRY conditions. This manoeuvre has been performed in corsa driving mode, with the electronic stability control switched off. This profile represents a sine sweep manoeuvre performed at different speeds: 50, 80, 100 and 120 km/h.

The third plot depicts manoeuvre "W". This manoeuvre has been performed in strada driving mode, with the electronic stability control switched off. In this picture it is represented a lap performed with an aggressive driving style, as it is clearly noticeable from the plots of the yaw rate and lateral accelerations. Anyway, even if the manoeuvre is quite complex it is possible to appreciate the precision of the ANN estimator

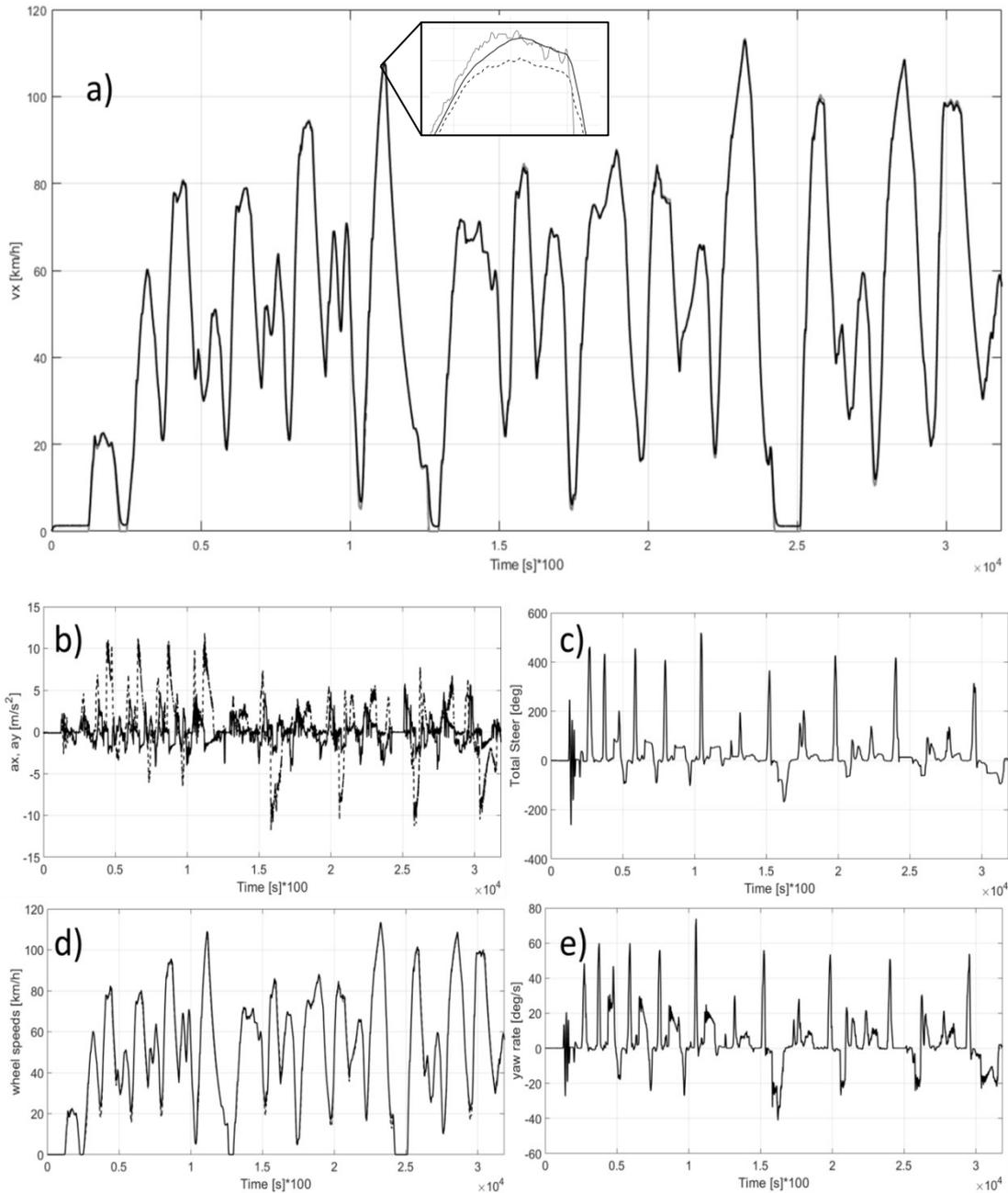


Figure 24. Profile "AS".

In picture a) are depicted the estimated speed using ANN (black line), the Kistler reference (black dashed line) and the ESP measurement (grey line). In b) there are the longitudinal acceleration (black line) and lateral acceleration (black dashed line). In c) is represented the total steer. In d) are reported the four wheel speed. In e) is present the yaw rate.

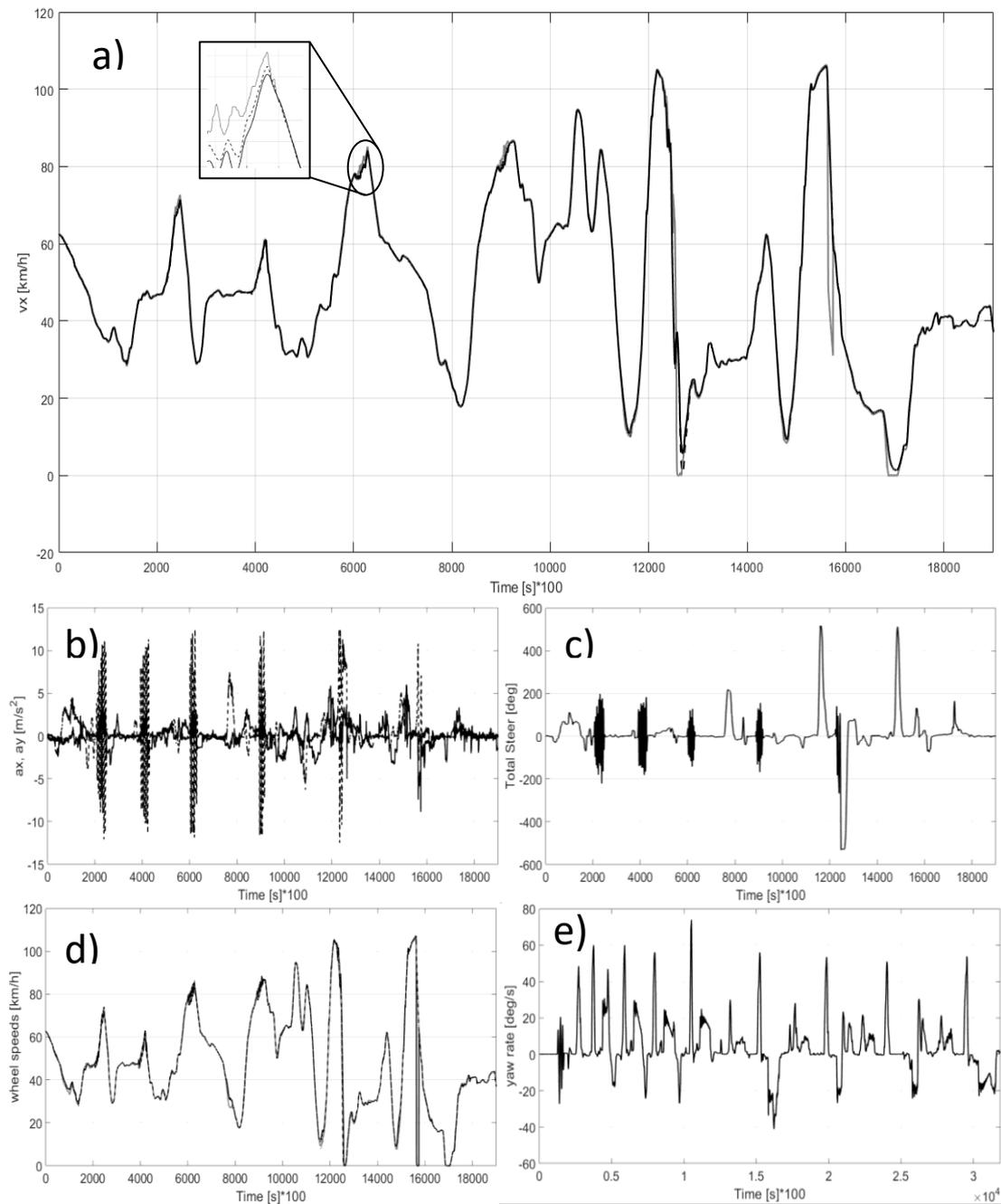


Figure 25. Manoeuvre A

This manoeuvre has been performed in corsa driving mode, with the electronic stability control switched off. This profile represents a sine sweep manoeuvre performed at different speeds: 50, 80, 100 and 120 km/h.

In picture a) are depicted the estimated speed using ANN (black line), the Kistler reference (black dashed line) and the ESP measurement (grey line). In b) there are the longitudinal acceleration (black line) and lateral acceleration (black dashed line). In c) is represented the total steer. In d) are reported the four wheel speed. In e) is present the yaw rate.

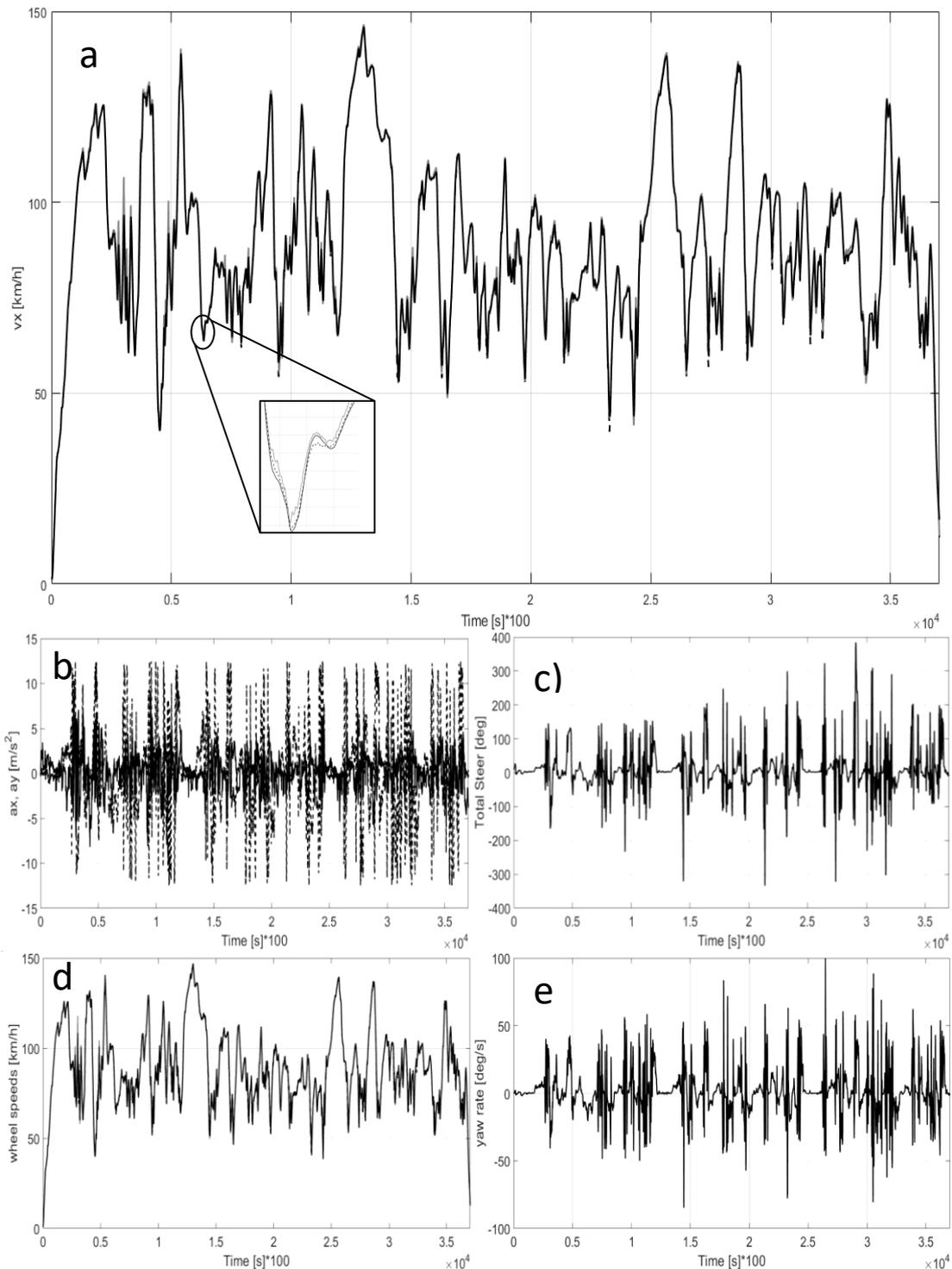


Figure 26. Manoeuvre W.

This manoeuvre has been performed in strada driving mode, with the electronic stability control switched off. In this picture it is represented a lap performed with an aggressive driving style, as it is clearly noticeable from the plots of the yaw rate and lateral accelerations.

In picture a) are depicted the estimated speed using ANN (black line), the Kistler reference (black dashed line) and the ESP measurement (grey line). In b) there are the longitudinal acceleration (black line) and lateral acceleration (black dashed line). In c) is represented the total steer. In d) are reported the four wheel speed. In e) is present the yaw rate.

3.2.2 WET conditions

In this section it is described the structure used and the results obtained for the speed estimation in WET conditions. In Figure 27 the visual location of the network used for the estimation in WET conditions in the designed layout is represented.

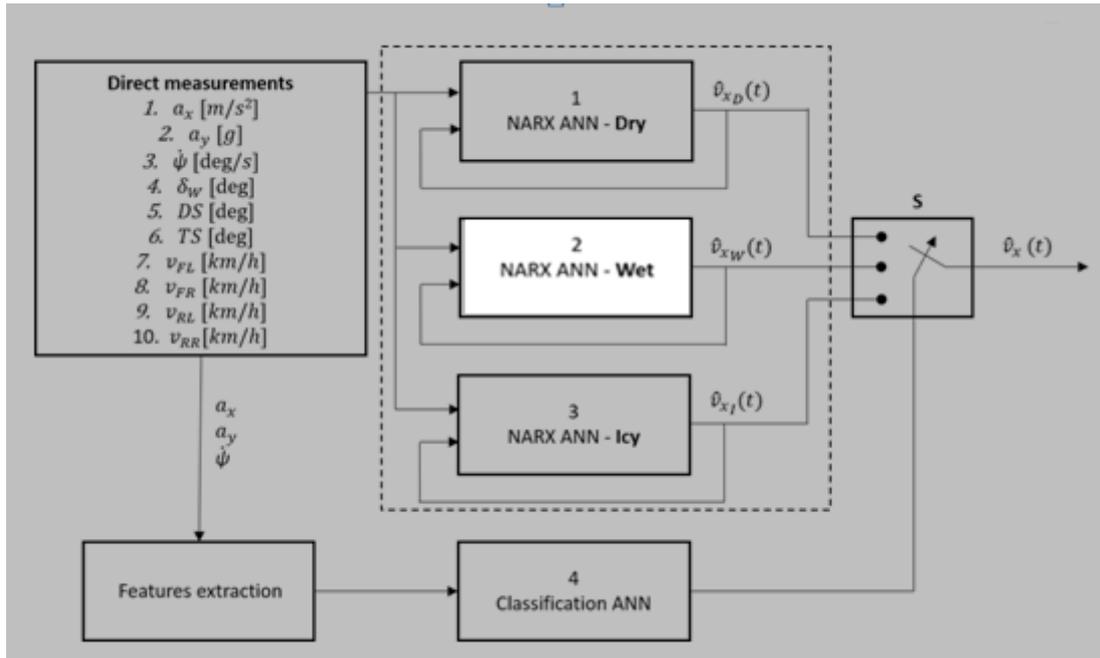


Figure 27. NARX ANN for WET conditions in the overall system.

In

Table 2 the structure of the recursive network used for the estimation is represented:

| | |
|---------------------|---------|
| Input delays | 3 |
| Feedback delays | 3 |
| Hidden layer size | 60 |
| Training function | trainlm |
| Total manoeuvres | 40 |
| Training manoeuvres | 6 |

Table 2. Hyperparameters defining the structure of the network used for the speed estimation in WET conditions

For what concern the WET conditions, they are the most difficult conditions in which the estimation can be performed, because the manoeuvres are characterized by the presence

of many instants in which the speed goes to zero, that is the condition in which the network has more difficulties in estimating the speed, as already explained in chapter 3.2.1, since it is a system that receives ten inputs per time and these inputs are very different one from the other. If the vehicle is stopped, that means the speed is zero, it could happen that one of the inputs is different from zero, even slightly, and this can lead to wrong information going to the network, thus can lead to an estimation of the speed different from zero.

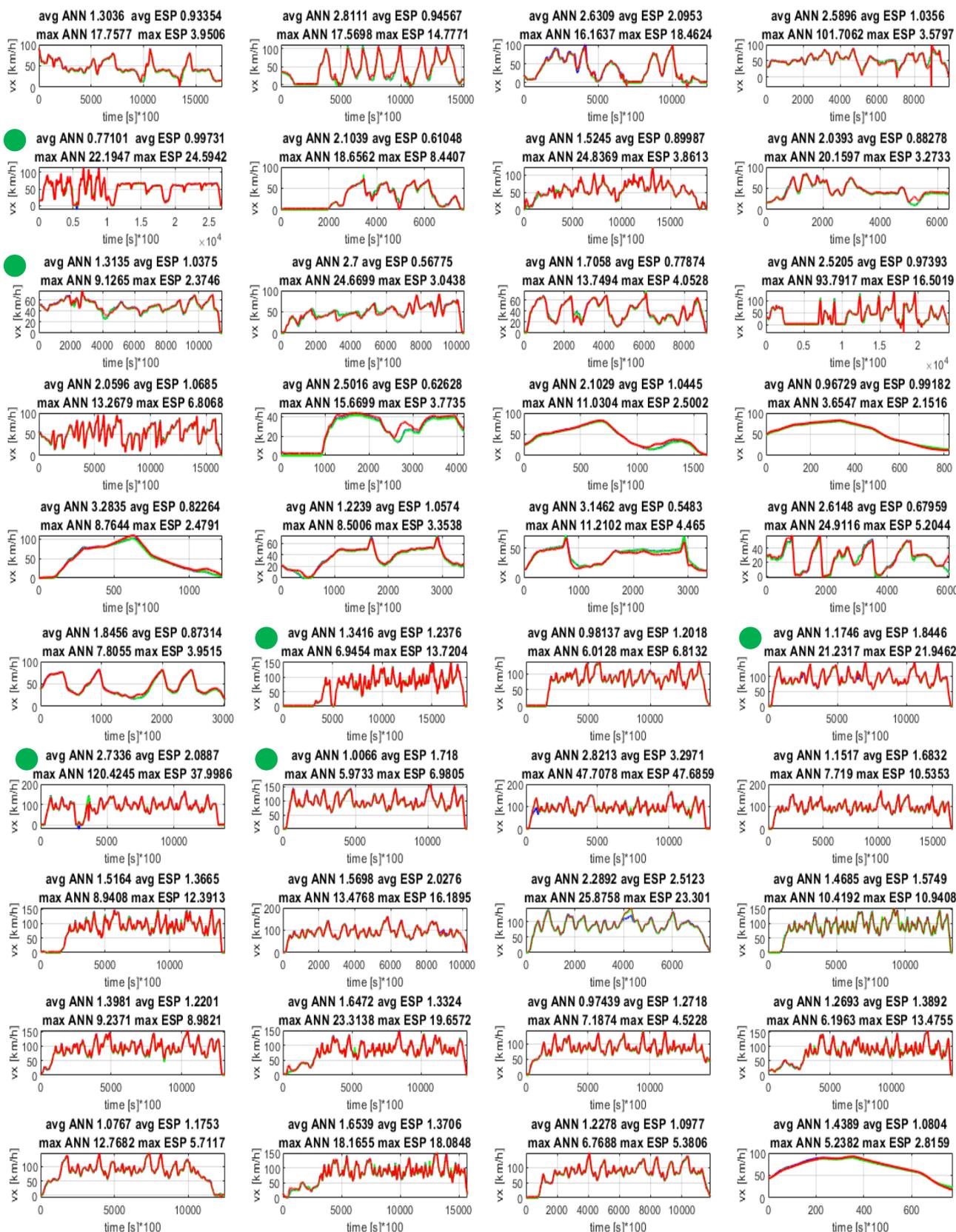
For this reason the WET conditions are a little bit more tricky to be estimated with respect to DRY and ICY conditions. Anyway the results obtained by our network are still satisfactory even if compared to the ESP estimation.

In Figure 28 are depicted the overall WET manoeuvres and the results obtained in each one of them, and furthermore are represented some of the profiles in order to have a clearer picture of the results obtained in every single manoeuvre.

As always the blue line is the reference signal of the Kistler measurement, while the red line represents the estimation performed by our network.

In the following are also plotted some significant manoeuvres in WET conditions, with focus on the inputs analysed. The first plot represents manoeuvre “DN”. This manoeuvre has been performed in strada driving mode, electronic stability control switched on, with a smooth driving style. The tires used were not heated up before the test.

The second picture is about manoeuvre “CY”. This manoeuvre has been performed in corsa driving mode, electronic stability control switched on, with an aggressive driving style. Also in this case the tires have not been heated up before the test.



3

Figure 28. Overview of the speed estimation in WET conditions. The blue line represents the speed computed by the Kistler camera, while the red line represents the ANN estimation. It is easy to understand from the picture that WET conditions were the most difficult to be estimated. Anyway the error is still acceptable in most of the profiles.

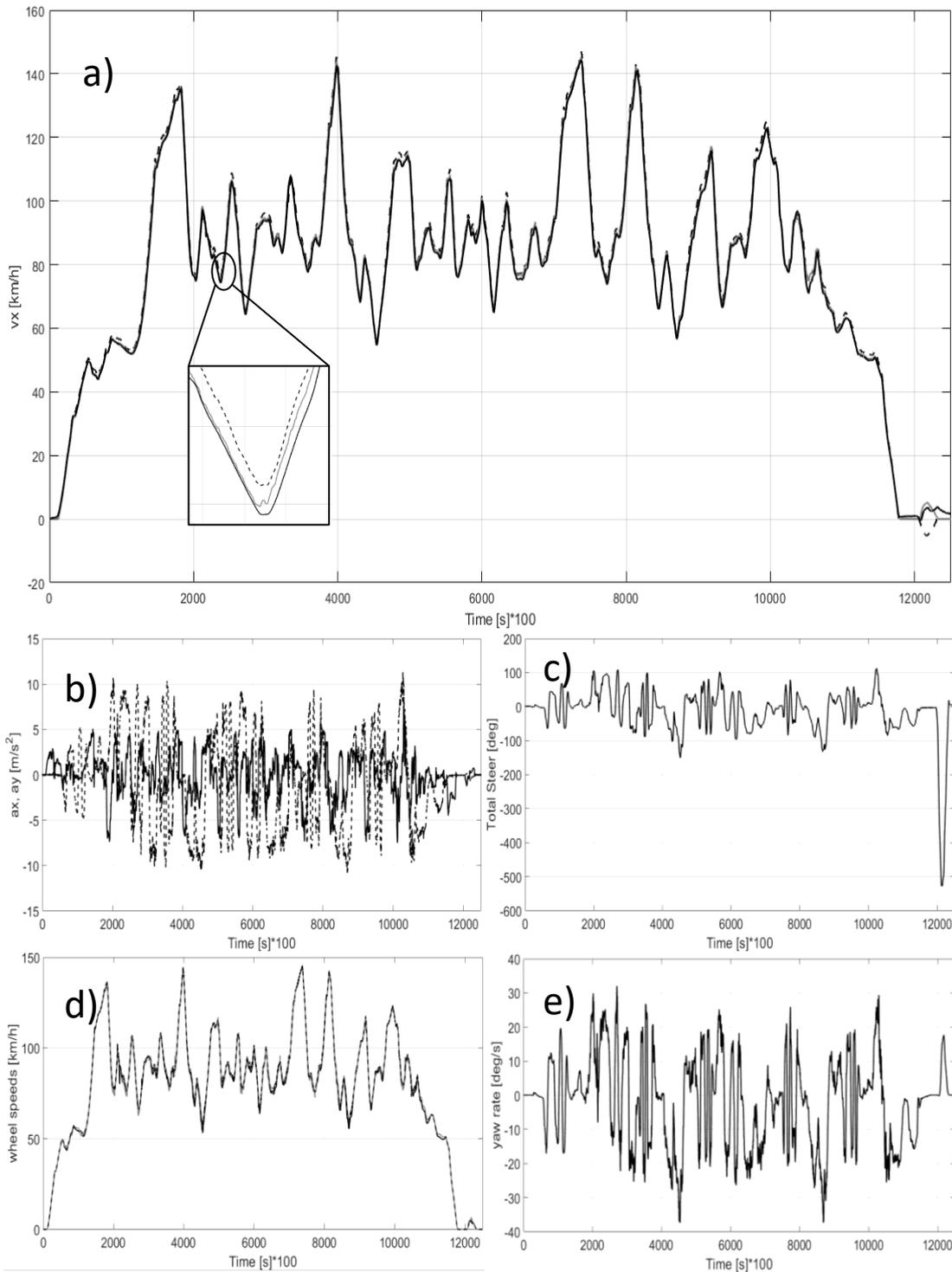


Figure 29. Manoeuvre DN in WET conditions.

This manoeuvre has been performed in strada driving mode, electronic stability control switched on, with a smooth driving style. The tires used were not heated up before the test. It is worth notice how the ANN estimator is able to copy much better the Kistler reference speed with respect to the ESP when, around instant 12, the driver gives a last steering to the vehicle. In picture a) are depicted the estimated speed using ANN (black line), the Kistler reference (black dashed line) and the ESP measurement (grey line). In b) there are the longitudinal acceleration (black line) and lateral acceleration (black dashed line). In c) is represented the total steer. In d) are reported the four wheel speed. In e) is present the yaw rate.

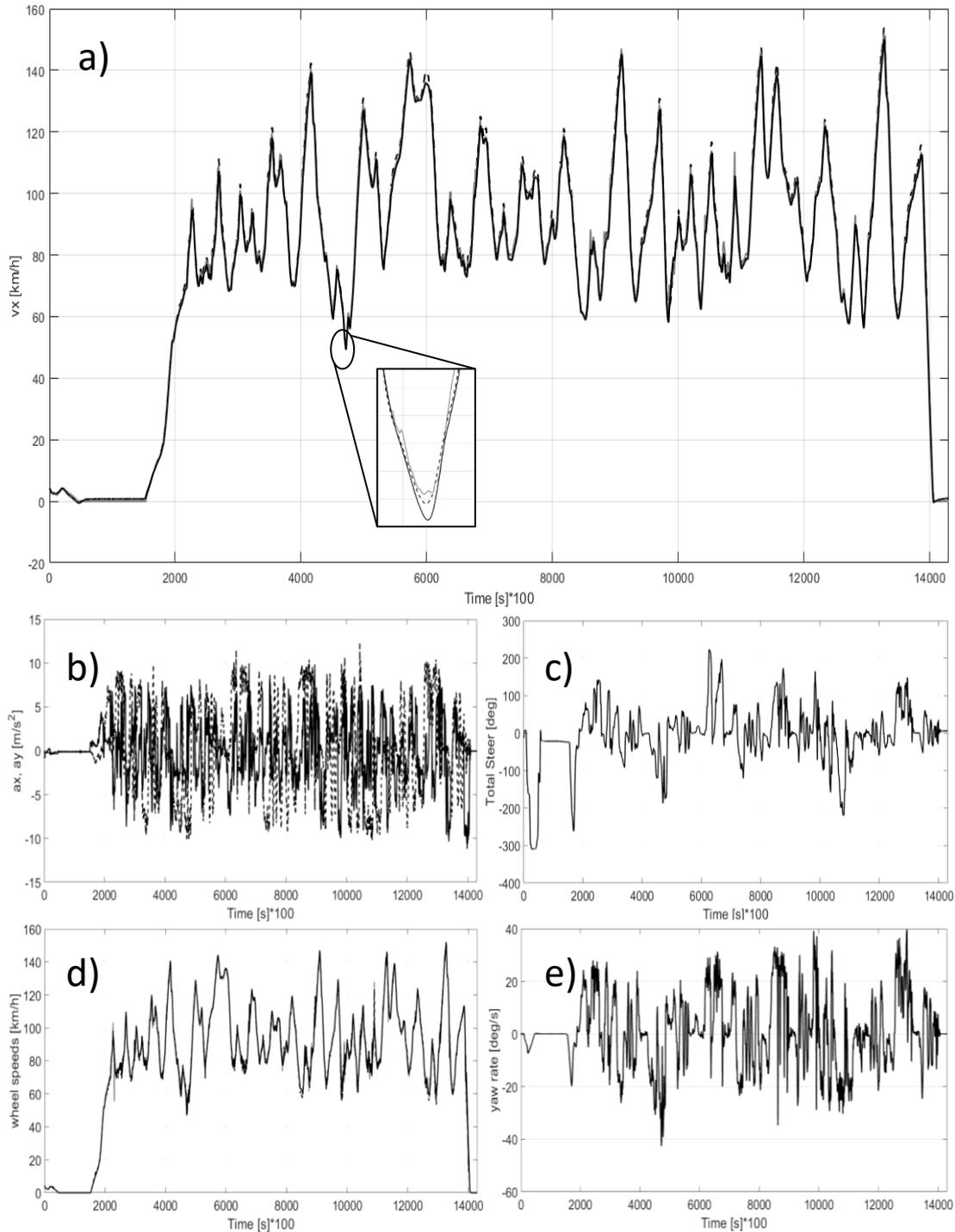


Figure 30. Manoeuvre CY in WET conditions.

This manoeuvre has been performed in corsa driving mode, electronic stability control switched on, with an aggressive driving style. Also in this case the tires have not been heated up before the test.

In picture a) are depicted the estimated speed using ANN (black line), the Kistler reference (black dashed line) and the ESP measurement (grey line). In b) there are the longitudinal acceleration (black line) and lateral acceleration (black dashed line). In c) is represented the total steer. In d) are reported the four wheel speed. In e) is present the yaw rate.

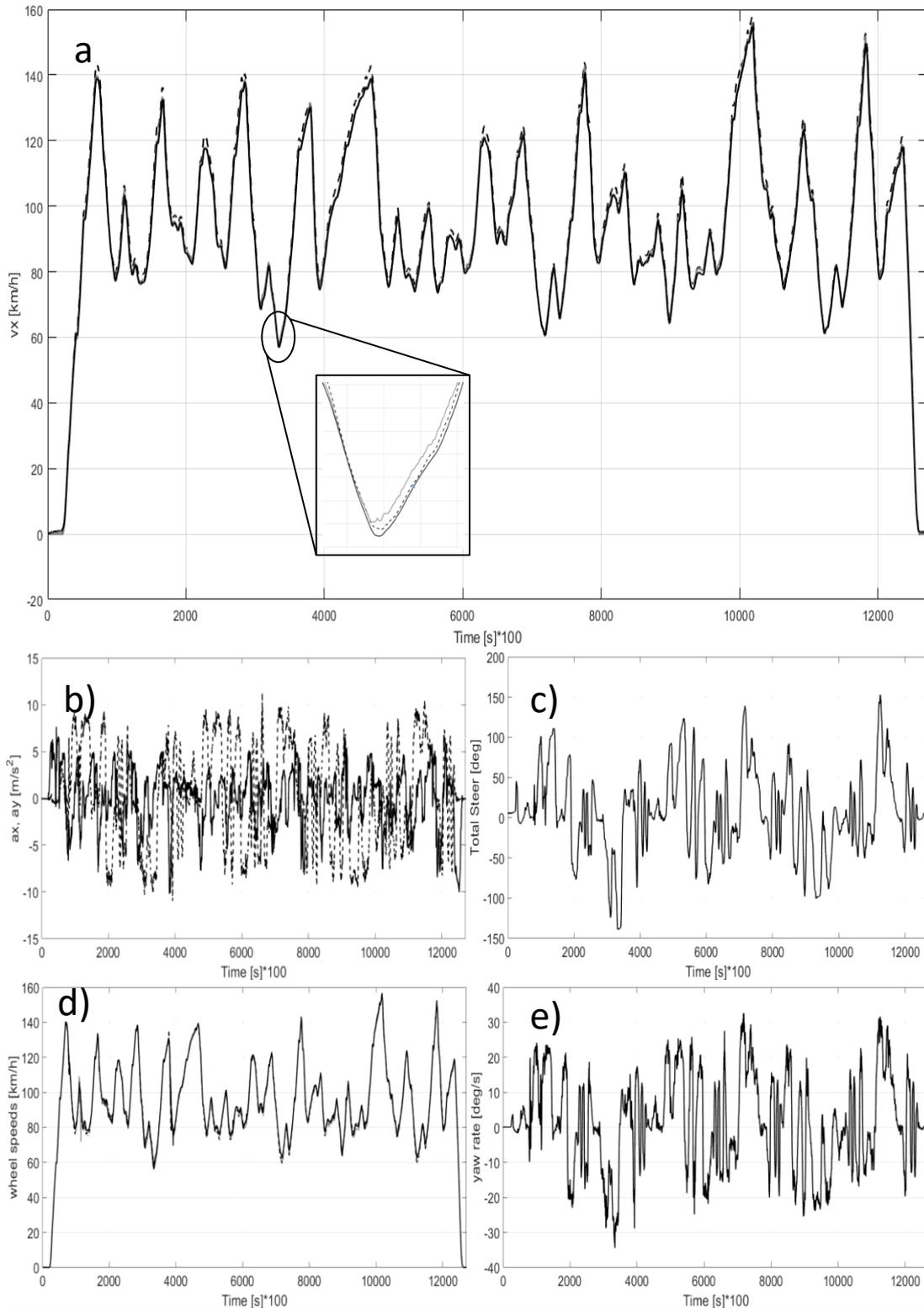


Figure 31. Manoeuvre CV

This manoeuvre has been performed in sport driving mode, electronic stability control switched off, with a smooth driving style. In this case the tires have been heated up before the test.

In picture a) are depicted the estimated speed using ANN (black line), the Kistler reference (black dashed line) and the ESP measurement (grey line). In b) there are the longitudinal acceleration (black line) and lateral acceleration (black dashed line). In c) is represented the total steer. In d) are reported the four wheel speed. In e) is present the yaw rate.

3.2.3 ICE conditions

In this paragraph the results obtained in ICY conditions are presented.

Table 3 lists the main properties of the regression NARX.

| | |
|---------------------|---------|
| Input delays | 2 |
| Feedback delays | 2 |
| Hidden layer size | 40 |
| Training function | trainlm |
| Total manoeuvres | 40 |
| Training manoeuvres | 14 |

Table 3. Hyperparameters defining the structure of the network used for the speed estimation in ICY conditions.

Figure 32 gives a visual positioning of the network used for the longitudinal speed estimation in ICY conditions inside the overall system layout, divided in regression and classification branches.

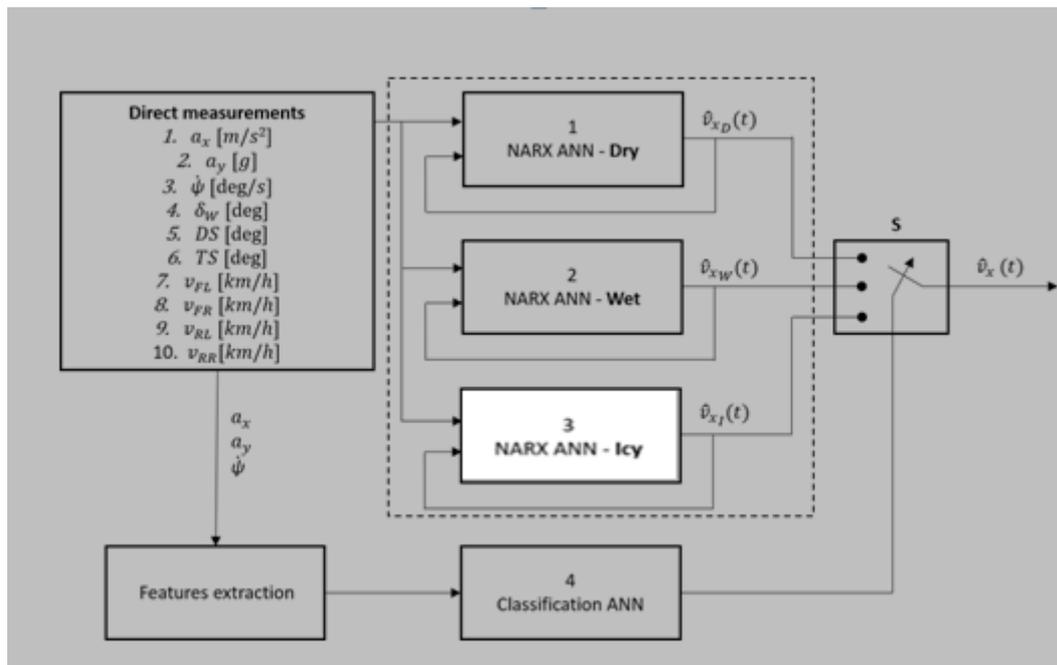


Figure 32. NARX ANN for ICY conditions in the overall system.

The estimation carried out in ICY conditions was really difficult to be obtained and required a lot of effort in terms of number of trials in order to obtain a network architecture able to provide satisfactory results.

Obtaining high performances in the estimation was possible re-training the closed loop network resulting from the first training phase.

As it was explained, typically a recursive ANN is firstly trained in open loop layout, when some of the inputs are used as reference for the network in order to train it, and in a second phase it is tested in closed loop, using 15% of the input dataset to check the results of the previous training without using any reference value.

Since it was very difficult to obtain satisfactory results using this common approach in ICY conditions, the solution was to train again the closed loop network in order to further reduce the error in the estimation.

Here below it is reported the piece of Matlab code used for this purpose:

```
netc1 = closeloop(net);  
[xc,xic,aic,tc] = preparets(netc1,Xt,{},Tt);  
yc1 = netc1(xc,xic,aic);  
estVelCL1 = cell2mat(yc1);  
[netc2,tr] = train(netc1,xc,tc,xic,aic);  
yc2 = netc2(xc,xic,aic);  
estVelCL2 = cell2mat(yc2);
```

From the code above, it is clear which are the steps required for this purpose. First of all we retrieve a first closed loop network, called *netc1*, that will be trained again in the following step to obtain another network, *netc2*, in which the error in the estimated speed is much lower, thus increasing the performances of our ANN.

Just to give the idea of the improvements, in some plots the maximum error obtained with the ANN was 10 time lower than the one obtained using the ESP for the computation: this is highlighted in Figure 34.

The only drawback of this solution relies on the fact that since we are training a closed loop network, the computational time and effort required are increasing.

This is mainly explained by the fact that a closed loop has a more complex structure, due to the feedback branch, and training this network requires more time if compared to a simpler open loop layout.

Anyway, this solution is really helpful and despite of the increased training time the results obtained improved a lot: typically ICY conditions are the most demanding scenarios, because the signals are very complex and disturbed, but using the re-training of the network these difficulties can be overcome.

Below are represented the overall profile plot and some single plots, in which it is possible to better notice the precision of the estimation.

The first picture represents manoeuvre “GB” performed in a daily driving mode, neutral driving conditions on a handling track. It is easy to be shown that the estimation capability of the ANN is quite high all along the duration of the manoeuvre. Notice also the difference among the estimation performed with the ANN and the one made by the ESP: the error of the ESP has been heavily reduced.

The second picture depicts manoeuvre “FO”: it is again a profile performed on the handling circuit, but with an aggressive driving style, as it is possible to be noticed from the values of accelerations and yaw rate. Again is worth to notice the improvements in the estimation made by the ANN with respect to the ESP.

The last figure represents manoeuvre “HO”. This is a manoeuvre performed in an urban environment, using a strada driving mode. It is useful to understand the behaviour of the vehicle and of the estimator in a condition outside of the track, in a “everyday” environment. Also in this conditions the ANN performs much better with respect to the oldestimation technique based on the ESP.

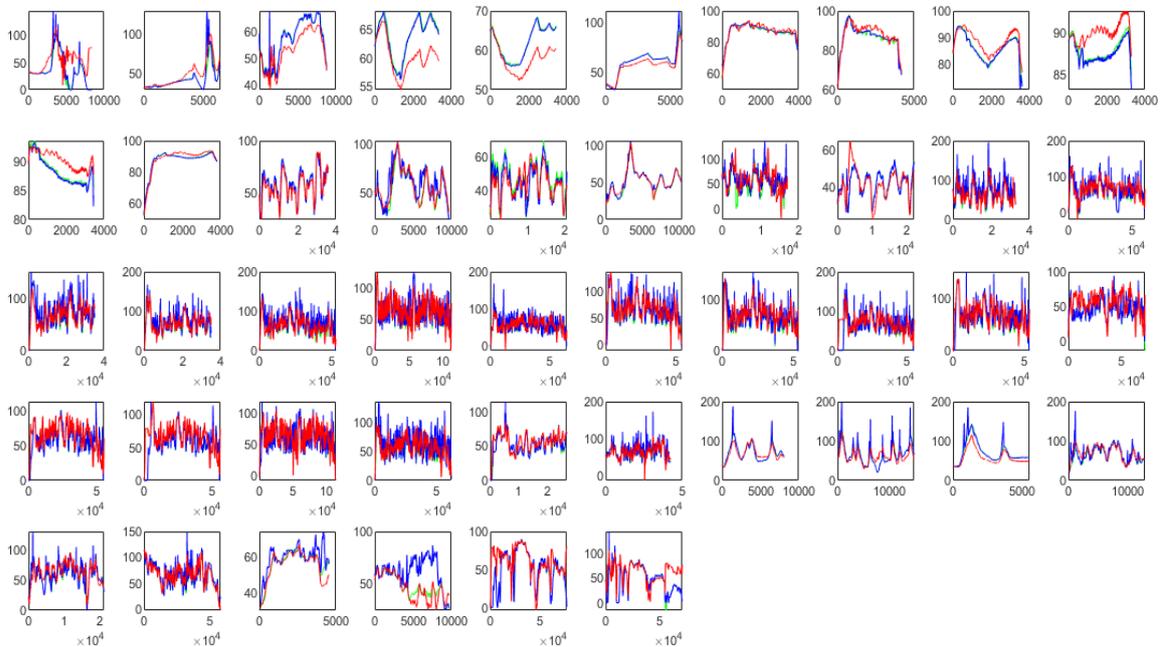


Figure 33. Example of results before using re-training technique.

This picture represents one of the first attempt in estimating the speed in ICY condition, without using the re-training technique. In this testing conditions even the simplest manoeuvres were really difficult to be estimated. It is clear the improvement done if we compare this picture with the results of Figure 34.

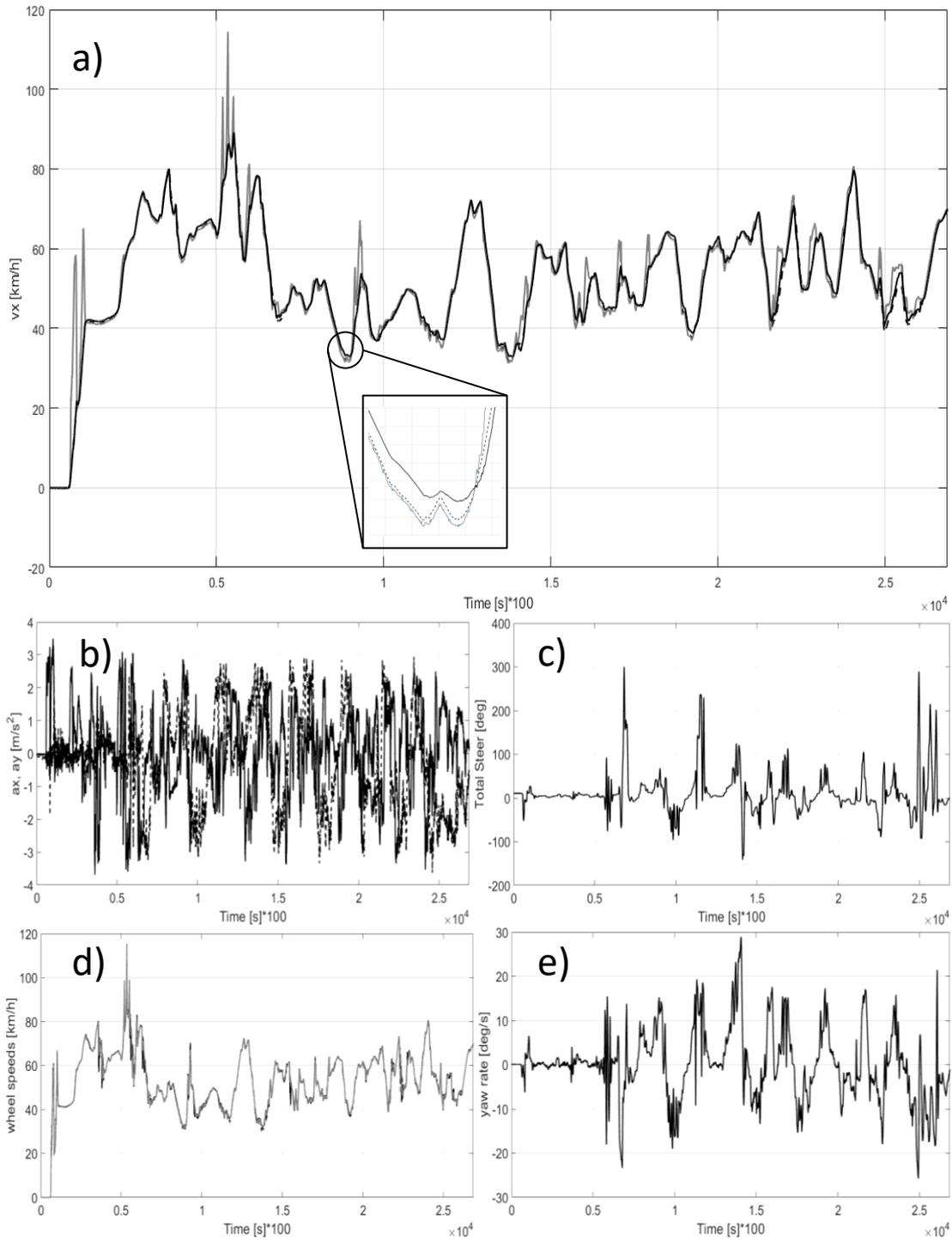


Figure 35. Manoeuvre GB in ICY conditions.

This profile represents a daily driving mode, performed in neutral driving conditions on a handling track.

It is easy to be shown that the estimation capability of the ANN is quite high all along the duration of the manoeuvre. Notice also the difference among the estimation performed with the ANN and the one made by the ESP: the error of the ESP has been heavily reduced. In picture a) are depicted the estimated speed using ANN (black line), the Kistler reference (black dashed line) and the ESP measurement (grey line). In b) there are the longitudinal acceleration (black line) and lateral acceleration (black dashed line). In c) is represented the total steer. In d) are reported the four wheel speed. In e) is present the yaw rate.

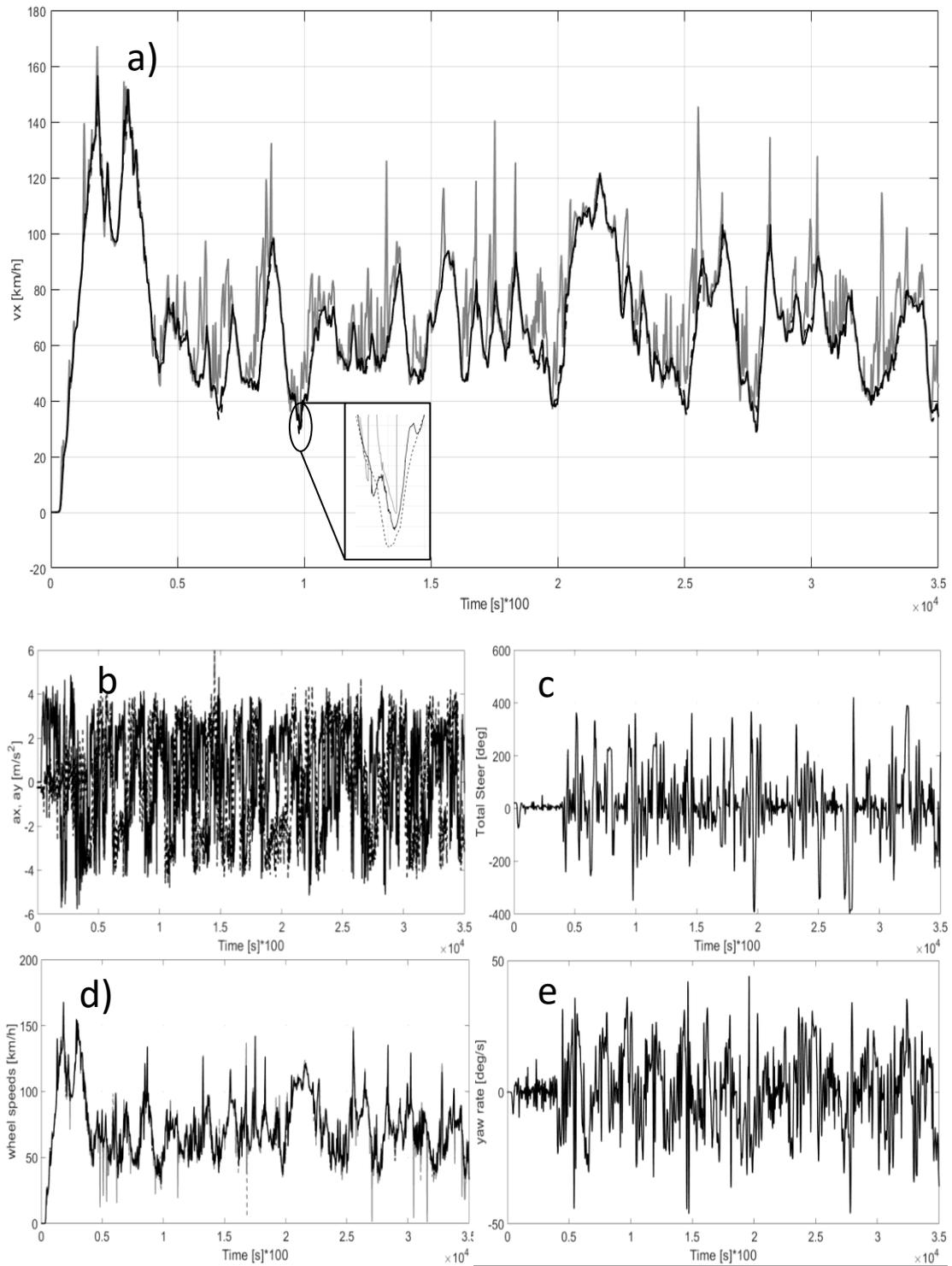


Figure 36. Manoeuvre FO in ICY conditions.

This is again a profile performed on the handling circuit, but with an aggressive driving style, as it is possible to be noticed from the values of accelerations and yaw rate. Again is worth to notice the improvements in the estimation made by the ANN with respect to the ESP.

In picture a) are depicted the estimated speed using ANN (black line), the Kistler reference (black dashed line) and the ESP measurement (grey line). In b) there are the longitudinal acceleration (black line) and lateral acceleration (black dashed line). In c) is represented the total steer. In d) are reported the four wheel speed. In e) is present the yaw rate.

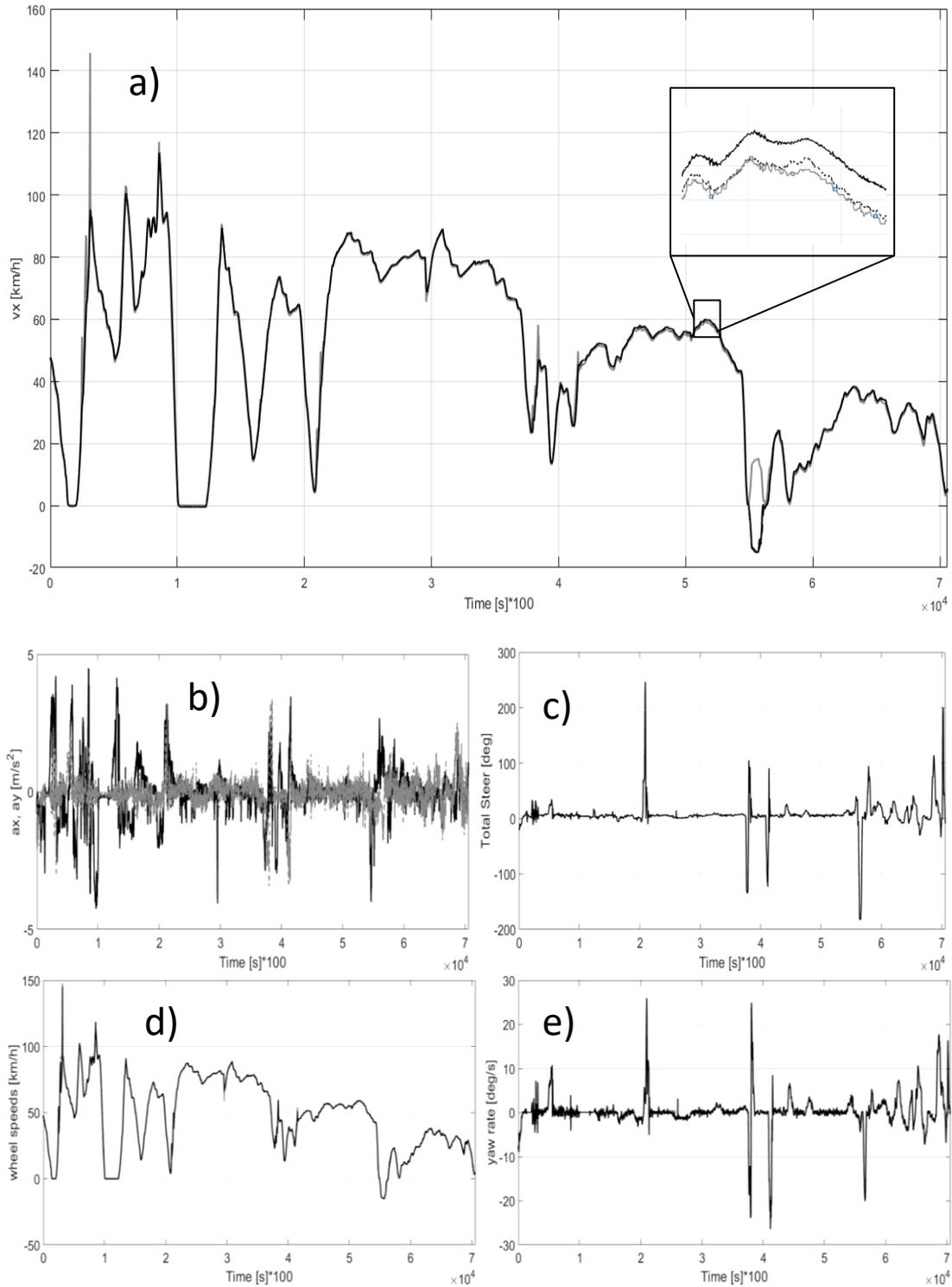


Figure 37. Manoeuvre HO in ICY conditions.

This is a manoeuvre performed in an urban environment, using a strada driving mode. It is useful to understand the behaviour of the vehicle and of the estimator in a condition outside of the track, in a “everyday” environment. Also in this conditions the ANN performs much better with respect to the oldestimation technique based on the ESP.

In picture a) are depicted the estimated speed using ANN (black line), the Kistler reference (black dashed line) and the ESP measurement (grey line). In b) there are the longitudinal acceleration (black line) and lateral acceleration (black dashed line). In c) is represented the total steer. In d) are reported the four wheel speed. In e) is present the yaw rate.

4 Road condition identification

In chapter 1.2 the state of the art of road condition identification techniques was described, and in this section the same topic with the application of artificial neural networks is analysed more in depth.

The problem of understanding in real time the road conditions is extremely important in order to improve the reaction time of all the active systems in vehicle, and as a consequence to improve driver and passengers' safety.

To accomplish this target, a *Classification* neural network has been designed. Its working principle is based on the analysis of some features extracted from longitudinal and lateral acceleration, yaw rate and the four wheel's longitudinal speeds signals.

This kind of network requires less computational power with respect to the *Recursive* networks, since its work is based only on the analysis of those features.

Figure 39 represents the overall system layout highlighting how it is used the classification network.

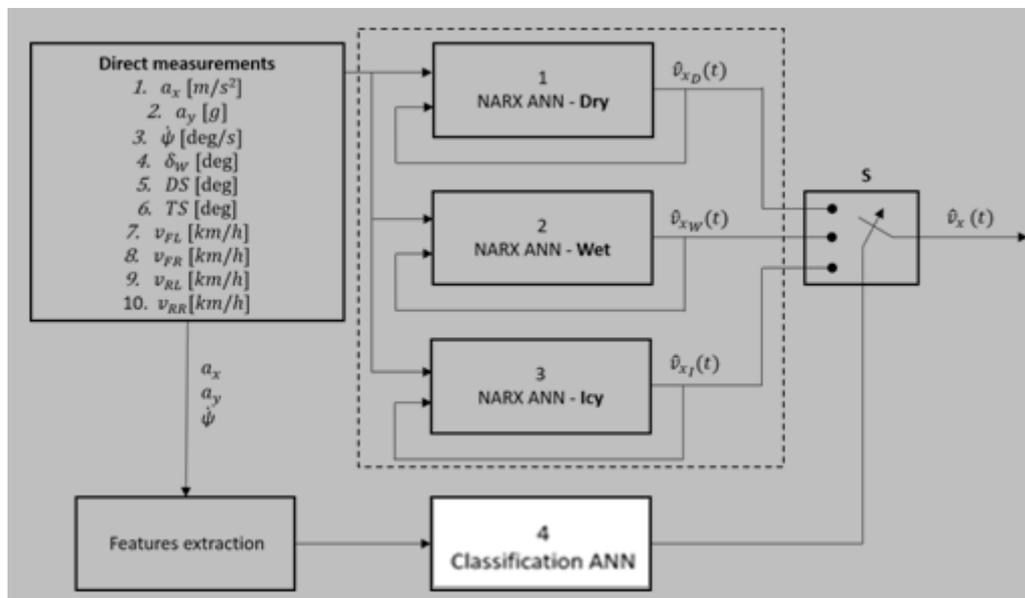


Figure 39. Pattern recognition network for road condition classification in the overall system layout

4.1 Architecture of the classifier

The architecture employed for this network is the so called *Pattern Recognition network (Patternet)*. It is a feedforward network for input classification in different classes, and for this reason it is trained using a supervised learning algorithm able to let this network assign some inputs to given target classes. Those target classes are represented by vectors of zeros except for the position that represent that class, which is indicated by ones.

For its nature, the classification network is giving a discrete response instead of a continuous response, that is typical of a regression network.

The architecture of this network is also characterized by a layered structure. There is an input layer, in which the number of neurons is given by the number of features we give in input, an hidden layer similar to the hidden layer present in recursive networks, and an output layer, characterized by as many neurons as the number of output classes we want to map.

The values of the output layer represent the level of confidence in assigning the correct class.

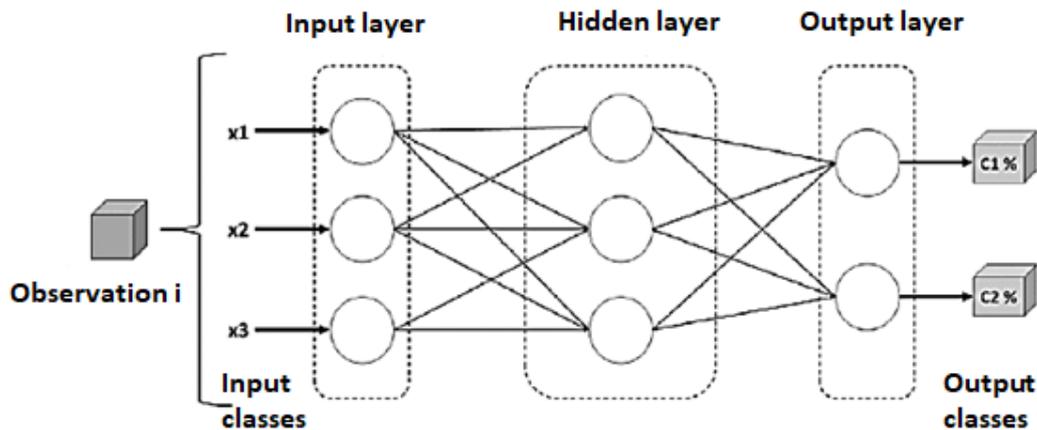


Figure 40. Example of Patternet, with 3 input features

Notice that in the picture are represented only three input classes, while in the classification network employed for the present work are employed 7 input classes.

4.2 Dataset – feature extraction

The dataset used for this section of the work is the same we have used for the recursive network described before.

The only difference with respect to the first part of this thesis is that the number of inputs we are using are different. In this case we consider as inputs the longitudinal and lateral acceleration, the yaw rate and the four wheel speeds.

The choice of using those inputs is related to prior considerations about vehicle dynamics and vehicle behaviours on different surfaces. As a result of these considerations, those seven inputs were chosen as best parameters that, when analysed, can give the best indication about the road conditions on which the vehicle is moving.

Classification networks work with discrete inputs that are represented by the i -th observation, and returns discrete output, that are the different classes we need to predict.

Typically the procedure adopted for this kind of network is divided in two steps.

The first one is committed to the preparation of the data to be given in input. Those input signals are firstly selected among all the dataset, according to the considerations explained before, and then buffered in slots of 10 seconds each, that means 1000 Hz.

This phase is called *Signal selection and Buffering*.

The buffering is a mandatory step, in order to prepare the following phase of classification: *Feature extraction*. In this second part, for each single slot of 10 seconds the network retrieves the features (also called predictors) required for the analysis, that will be used in order to understand which road surface we are travelling on.

The signal predictors used for this work are:

- Spectral peak: computes the peaks present in the spectrum and identifies the highest one, for each signal present in the considered buffer.
- Spectral power: computes the integral of the power spectral density (PSD) of the features. The PSD describes the power distribution of a signal in the frequency domain, so integrating in the frequency domain we obtain the total signal power in W .

$$\int PSD df \quad (18)$$

where f is the frequency, and PSD is computed in $\frac{J}{s Hz}$.

- Peak magnitude to root mean square value: computes the relationship among the peak magnitude and the root mean square (RMS) value of the signal. This is helpful in

order to understand the type of signal we are considering. For instance if this ratio is equal to $\sqrt{2}$, then we can recognize a sine wave signal.

The formula employed for the RMS value is:

$$x_{rms} = \sqrt{\frac{1}{n}(x_1^2 + x_2^2 + \dots + x_n^2)} \quad (19)$$

Where n is the total number of elements and x_i^2 represents the squared value of each element.

And in general the Peak to RMS ratio of a variable is given by:

$$\frac{V_{Peak}}{V_{RMS}} = \sqrt{2} \quad (20)$$

This equation represents an example of Peak to RMS ratio of a voltage signal

- Standard deviation: indicates how much is the dispersion of the data considered. Typically low values of standard deviation implies that the values are located near the mean of the distribution, that is also the expected value. Vice versa, if the standard deviation is high the values are spread out over a wider range.

It is computed using:

$$\sigma = \sqrt{\frac{\sum(x_i - \mu)^2}{n}} \quad (21)$$

Where x_i indicates the i -th component, μ is the true mean of the distribution and n is the total number of elements considered in the population.

- Variance: "is the expectation of the squared deviation of a random variable from its mean" [35]. Strictly speaking, it measures how far a set of numbers are from their average value. It is equal to the squared value of the standard deviation.

It is computed as the squared value of the standard deviation, and it is typically represented as:

$$\sigma^2 \text{ or } Var(x) \quad (22)$$

The network gives as output a *feature vector* composed by 140 element. This will be the input vector for our network.

This procedure is typical for Classification networks, where generally the input dataset is very large and sometimes characterized by redundant information. So to overcome these

difficulties, the network uses the feature vector as input, which has a lower dimension, and this improves the computational velocity of the network without sacrificing the estimation result. Here below it is possible to see a schematic representation of this architecture.

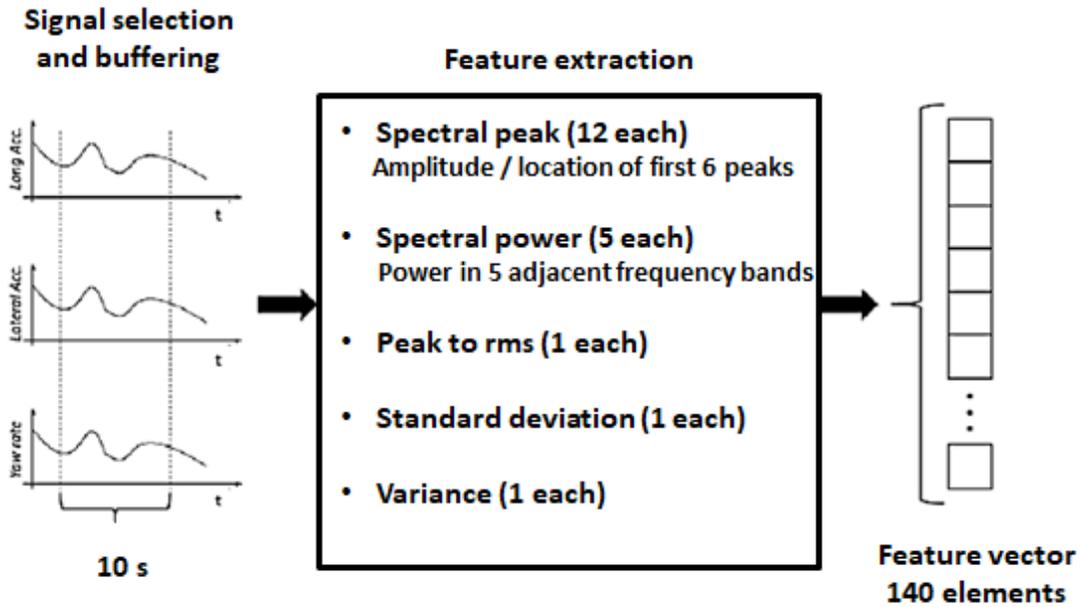


Figure 41. Example of classification procedure

To be complete, in Appendix B are reported all the features used for the analysis during the classification.

4.3 Training and Validation

The design and implementation of the classification network was carried out using MATLAB and Simulink softwares and the corresponding neural network toolbox, called Signal Processing Toolbox™.

For what concern the network architecture, it is characterized by a single hidden layer, with a Sigmoid function, and a single output layer, with a Softmax function, which is typically used for classification problems.

Figure 42 represents the typical architecture of a classification neural network, where it is possible to notice the two layers with the two functions used for processing the data.

The Sigmoid function is the same that is described in chapter 1.4, while the softmax function is typically employed in classification neural networks. It is a generalization of the

logistic function that “squashes” a k-dimensional vector of arbitrary real values in another k-dimensional vector of values comprised among (0, 1) and which the sum is equal to one.

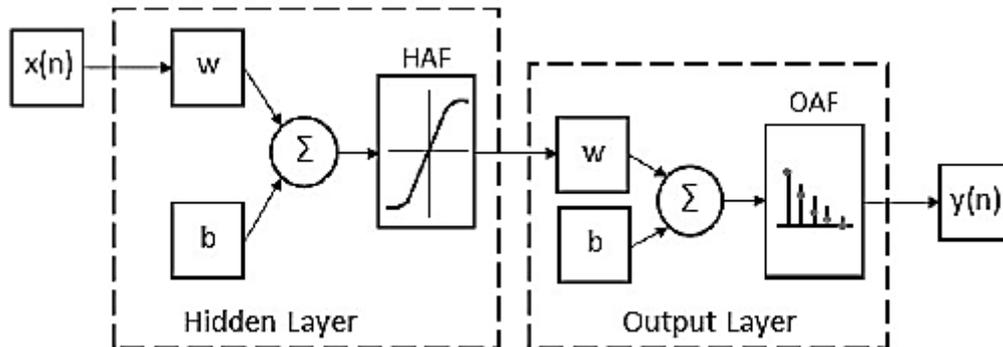


Figure 42. Classification neural network architecture.

The results obtained during the classification are reported in terms of ROC plot and classification matrix.

The Receiver Operating Characteristic (ROC) plot is an useful tool to study the relationships among real alarm and fake alarm. It is a plot organized on two axes: on the abscissa axis are present the *False Positives* (FP), while on the y axis are reported the *True Positives* (TP). These two concepts can be easily explained with an example. Let's consider a binary classifier, that has to classify, which means “recognize”, two different signals. The two considered signals can be represented by two Gaussian distributions. Once chosen a threshold value that allows us to distinguish among the two signals, since the two Gaussian curves are a little bit superimposed, four different results can be obtained with the classifier:

- If the value predicted by the classifier is *positive* and also the true value is *positive*, then it is called *true positive* (TP);
- If the value predicted by the classifier is *negative* but the true value is *positive*, then it is called *false positive* (FP);
- On the contrary, if the value predicted by the classifier is *negative* and also the true value is *negative*, then it is called *true negative* (TN);
- A *false negative* (FN) is instead obtained when the result of the classifier is *negative* and the true value is *positive*.

With the analysis of a ROC plot it is possible to evaluate the ability of the classifier to discern simply by analysing the area underneath the plot, which is called *Area Under Curve* (AUC). The values of AUC are comprised between (0, 1), indicating the probability of the classifier to distinguish and recognize correctly the two classes.

On a plot, the ROC curves pass through points (0, 0) and (1, 1) and have two limit conditions:

- A curve with 45 degree of slope, that connects the origin to point (1, 1). This is the curve which represents the casual classifier, with the AUC equal to 0,5
- A curve that is composed by the segment which connects the origin to point (0, 1) and by the segment connecting (0, 1) to (1, 1). This curve has the AUC equal to one and represents the perfect classifier.

The other useful tool that is possible to use in order to evaluate and analyse the performance of a classifier, is the so called *confusion matrix* in which *true positives*, *true negatives*, *false positives* and *false negatives* are reported.

As the name suggests, it is a matrix that allow to understand how much our classifier is confused relatively to the parameters it has to distinguish.

In this matrix, the row represent the *Predicted classes* while the columns represent the *true classes*. On the diagonal are present the classes that the classifier was able to recognize correctly.

For what concern the procedure of a classification network, it is the same used for a recursive one. This means that also in this case the work is divided in a training phase and in a validation phase.

The network employed for the training phase is defined by this parameter setting:

- Input Layer Size: 140.
The dimensions of the hidden layer are defined by the number of features to be computed for observation.
- Hidden Layer Size: 150.
- Output Layer Size: 3.
The dimension of the output layer are corresponding to the number of output classes. (DRY - WET - ICY).
- Performance Function: crossentropy.

Computes net performance given targets and outputs. It penalizes a lot the outputs which are very inaccurate, while does not penalize the correct outputs. In this way is possible to obtain better classifiers.

- Training Function: trainscg

- Training buffered manoeuvres: 69

| | | | | | |
|--------------|-----|---------------|---------------|---------------|---------------|
| Output class | DRY | 951 43,0% | 22 1,0% | 6 0,3% | 97,4% 2,6% |
| | WET | 10 0,5% | 273 12,4% | 4 0,2% | 95,1% 4,9% |
| | ICE | 9 0,4% | 8 0,4% | 927 41,9% | 98,2% 1,8% |
| | | 98,0% 2,0% | 90,1% 9,9% | 98,9% 1,1% | 97,3% 2,7% |
| | DRY | WET | ICE | Target class | |

Figure 43 are reported the results obtained in the training phase.

| | | | | | |
|--------------|-----|---------------|---------------|---------------|---------------|
| Output class | DRY | 951 43,0% | 22 1,0% | 6 0,3% | 97,4% 2,6% |
| | WET | 10 0,5% | 273 12,4% | 4 0,2% | 95,1% 4,9% |
| | ICE | 9 0,4% | 8 0,4% | 927 41,9% | 98,2% 1,8% |
| | | 98,0% 2,0% | 90,1% 9,9% | 98,9% 1,1% | 97,3% 2,7% |
| | DRY | WET | ICE | Target class | |

Figure 43. Confusion matrix for training phase.

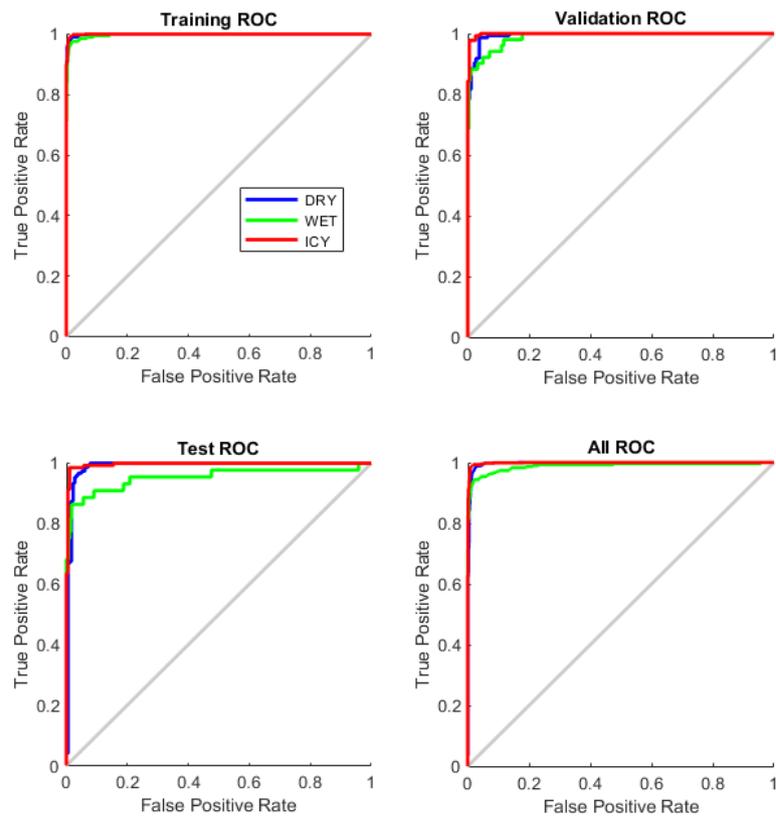


Figure 44. ROC plot of training phase

After the training phase, it was necessary to design also the network for the validation phase. For this purpose, the parameter setting adopted was the following:

- Input Layer Size: 140.
- Hidden Layer Size: 150.
- Output Layer Size: 3.
- Performance Function: crossentropy.
- Training Function: trainscg
- Training buffered manoeuvres: 56

The results obtained during the validation phase are reported in Figure 45.

| | | | | | |
|--------------|-----|---------------|---------------|---------------|----------------|
| Output class | DRY | 388 42,5% | 2 0,2% | 9 1,0% | 97,2% 2,8% |
| | WET | 4 0,4% | 62 6,8% | 7 0,8% | 84,9% 15,1% |
| | ICE | 3 0,3% | 0 0,0% | 438 48,0% | 99,3% 0,7% |
| | | 98,2% 1,8% | 96,9% 3,1% | 96,5% 3,5% | 97,3% 2,7% |
| | DRY | WET | ICE | Target class | |

Figure 45. Confusion matrix for validation phase

As it is possible to be noticed from the confusion matrix of the validation phase, DRY and ICY conditions can be classified with a really high percentage of success, while WET conditions are the most difficult to be correctly recognized. This is probably due to the fact that WET conditions are a sort of mid-way with respect to the other two, and for this reason it is easier to confuse a wet road with a dry or icy one, also depending on the quantity of water on the ground.

5 Conclusions

This chapter deals with some peculiar critical aspects of the thesis work performed and summarizes the results obtained, with a focus on the possible future works useful in order to improve the vehicle stability and, as direct consequence, the safety of passengers and occupants.

5.1 Comments on the results

From the results obtained it is possible to understand that the NARX architecture in closed loop is able to provide good estimation results with low computational capacity, and this is the real strong point of this technology with respect to other previous adopted solutions like Kalman filtering or GPS based systems, which are very popular in literature.

Of course the results we have obtained are derived from the analysis of a given dataset of manoeuvres, that even if composed by 125 manoeuvres performed in different conditions, anyway need to be implemented with other testing manoeuvres to obtain a complete and precise estimation in every conditions. For this reason could happen that some profiles can present errors of some kilometres per hour in the speed estimation, simply because that given profile is particularly complex or induces some dynamics in the vehicle which cannot be easily detected in the training phase, introducing some errors in the estimation as a consequence.

The condition which caused more difficulties was the WET road conditions, both in the longitudinal speed estimator and in the road condition classifier. This is probably due to the fact that WET conditions are a sort of hybrid condition, a mid way between DRY and ICY conditions that is more difficult to be correctly detected. Imagine for instance a scenario in which it rains very slowly and the tarmac absorbs a lot the water. In this condition the tarmac is wet due to the humidity, but it is not so wet to cause aquaplaning or to have a thin water meatus on it. This is a difficult problem for our classifier because it is not easy to classify in the road is in WET or DRY conditions and as a consequence also the inputs for the longitudinal speed estimator can be a little bit corrupted.

Anyway we can say that the estimation results obtained are very satisfactory and can be the starting point for future applications for further studies on the vehicle dynamics on different road conditions.

To obtain such a results, the networks have been trained using different layouts and performing different train and testing phases, in order to obtain the best possible network. In the end it was clear that a network with few neurons and few layers was the best option for our purpose.

5.2 Future works

The present work is a good starting point for future applications of a longitudinal speed estimator and of a road condition classifier on a real vehicle. The advantages introduced by an estimator based on artificial neural network are multiple. The first one is for sure the ability to estimate and if necessary adjust in real time the dynamics of the vehicle in order to improve its stability and also to increase the safety of driver and passengers. The second important advantage is related to the implementation cost. Typically this kind of devices can be implemented on the vehicle using an Arduino board, which is characterized by good computational capacity, small dimensions and really low cost. This means that improving the reliability of this technology can open new possibilities of application also in lower segment of the market, without remaining only a priority of premium vehicles.

Thus, starting from this work the possibilities of implementation are a lot, also due to the continuously increasing trend of inserting on the vehicle electronic devices and sensors, that can improve more and more the collection of data.

List of Figures

| | |
|---|----|
| Figure 1. Schematic description on GPS working principle | 5 |
| Figure 2. System used for the analysis in [4]. | 6 |
| Figure 3. Bicycle model of a car | 10 |
| Figure 4. Friction coefficient versus slip. | 12 |
| Figure 5. Single wheel model..... | 14 |
| Figure 6. working principle of light scattering | 16 |
| Figure 7. Schematic summary of the different machine learning algorithms | 19 |
| Figure 8. Some examples of ADAS applications..... | 20 |
| Figure 9. Layered structure of a network | 22 |
| Figure 10. Schematic representation of an artificial neuron..... | 23 |
| Figure 11. Examples of activation functions | 24 |
| Figure 12. Feedforward ANN | 25 |
| Figure 13. NARX ANN structure | 26 |
| Figure 14. System layout..... | 27 |
| Figure 15. 10 inputs | 29 |
| Figure 16. Schematic representation of the on-board layout | 29 |
| Figure 17. Longitudinal velocity sensor layout. A) Optical sensor and lamp. B) Image correlation processor. C) Rapid prototyping ECU. | 30 |
| Figure 18. Typical structure of a NARX network..... | 32 |
| Figure 19. Open loop configuration..... | 32 |
| Figure 20. Closed loop configuration..... | 33 |
| Figure 21. Example of learning curve. | 34 |
| Figure 22. NARX ANN for DRY conditions in the overall system..... | 36 |
| Figure 23. Results obtained in DRY conditions. | 39 |
| Figure 24. Profile "AS" | 41 |
| Figure 25. Manoeuvre A | 43 |
| Figure 26. Manoeuvre W. | 43 |
| Figure 27. NARX ANN for WET conditions in the overall system..... | 44 |
| Figure 28. Overview of the speed estimation in WET conditions..... | 46 |

| | |
|--|----|
| Figure 29. Manoeuvre DN in WET conditions..... | 47 |
| Figure 30. Manoeuvre CY in WET conditions..... | 48 |
| Figure 31. Manoeuvre CV | 49 |
| Figure 32. NARX ANN for ICY conditions in the overall system. | 50 |
| Figure 33. Example of results before using re-training technique. | 52 |
| Figure 34. Global overview of the profiles in ICY conditions. | 53 |
| Figure 35. Manoeuvre GB in ICY conditions..... | 54 |
| Figure 36. Manoeuvre FO in ICY conditions..... | 55 |
| Figure 37. Manoeuvre HO in ICY conditions. | 56 |
| Figure 38. Manoeuvre HO..... | 56 |
| Figure 39. Pattern recognition network for road condition classification in the overall system layout | 57 |
| Figure 40. Example of Patternet, with 3 input features | 58 |
| Figure 41. Example of classification procedure | 61 |
| Figure 42. Classification neural network architecture..... | 62 |
| Figure 43. Confusion matrix for training phase. | 64 |
| Figure 44. ROC plot of training phase | 65 |
| Figure 45. Confusion matrix for validation phase..... | 66 |

List of Tables

| | |
|---|----|
| Table 1. Hyperparameters defining the structure of the network used for the speed estimation in DRY conditions | 37 |
| Table 2. Hyperparameters defining the structure of the network used for the speed estimation in WET conditions | 44 |
| Table 3. Hyperparameters defining the structure of the network used for the speed estimation in ICY conditions..... | 50 |
| Table 4. DRY manoeuvres | 77 |
| Table 5. WET manoeuvres | 79 |
| Table 6. ICY manoeuvres..... | 82 |
| Table 7. List of features computed for the road condition identification. | 84 |

References

- [1] Jyotishman Ghosh - State Estimation and Control of Active Systems for High Performance Vehicles. 2017.
- [2] Alessandro Gianfrate - Sideslip Angle Estimation and Road Condition Identification by Artificial Neural Networks. 2017.
- [3] David M. Bevly - The Use of GPS Based Velocity Measurements for Improved Vehicle State Estimation. 2000
- [4] Liang Chu - Vehicle Lateral and Longitudinal Velocity Estimation Using Coupled EKF and RLS Methods. 2010.
- [5] Lars Imsland - Vehicle Velocity Estimation using Modular Nonlinear Observers
- [6] Introducing Machine Learning – Mathworks ebook.
- [7] D.C. Park - Electric Load Forecasting Using An Artificial Neural Network (1991)
- [8] “Deep Learning: Feedforward Neural Network” – [www. towardsdatascience.com](http://www.towardsdatascience.com) (last access December 2018)
- [9] “Neural Networks” - cs.stanford.edu (last access December 2018)
- [10]H. Nozer - Recursive Neural Networks
- [11]Stefano Feraco - State of Charge estimation in electrical vehicles Lithium batteries using Artificial Neural Networks
- [12]K. Levenberg, "A Method for the Solution of Certain Non-Linear Problems in Least Squares". Quarterly of Applied Mathematics, 1944
- [13]D. Marquardt, "An Algorithm for Least-Squares Estimation of Nonlinear Parameters". SIAM Journal on Applied Mathematics, 1963
- [14]H.Sado - Road Condition Estimation for Traction Control in Electric Vehicle
- [15]F. Gustafsson - Slip-Based Tire-Road Friction Estimation
- [16]Patent US6040916 – Process and apparatus determining the condition of a road surface
- [17]G. Genta – The Automotive Chassis
- [18]Influence of Chassis Control Systems on Vehicle Handling and Rollover Stability. Alrik L. Svenson, Aleksander Hac. 2007.
- [19]Test Procedures and Evaluation Tools. Karlsson, Anders. 2014.
- [20]Neural Networks in System Identification. G. Horvath

- [21] Mean squared error. en.wikipedia.org/wiki/Mean_squared_error (last access December 2018)
- [22] https://en.wikipedia.org/wiki/Extended_Kalman_filter (last access December 2018)
- [23] https://en.wikipedia.org/wiki/Recursive_least_squares_filter (last access December 2018)
- [24] https://en.wikipedia.org/wiki/Kalman_filter (last access December 2018)
- [25] Vehicle Lateral and Longitudinal Velocity Estimation Based on Adaptive Kalman Filter. L.Chu
- [26] Adaptive Kalman Filtering. Steven D. Brown
- [27] Longitudinal vehicle state estimation using nonlinear and parameter-varying observers. Ehsan Hashemi.
- [28] Integrated model predictive control and velocity estimation of electric vehicles. Milad Jalali.
- [29] https://en.wikipedia.org/wiki/Model_predictive_control#Overview (last access December 2018)
- [30] On-board road condition monitoring system using slip-based tyre-road friction estimation and wheel speed signal analysis. K. Li.
- [31] Road condition analysis using NIR illumination and compensating for surrounding light. J. Casselgren.
- [32] Ice formation detection on road surfaces using infrared thermometry. M. Riehm.
- [33] Road Condition Identification from Millimeter-wave Radar Backscatter Measurements. P. Asuzu.
- [34] Road-Condition Recognition Using 24-GHz Automotive Radar. V.V. Viikari.
- [35] <https://en.wikipedia.org/wiki/Variance> (last access December 2018)

Appendix A

In this section, a more detailed description of the manoeuvres analysed in DRY, WET and ICY condition is given. The profiles with a tick are the one employed in training phase.

| DRY | | |
|-----|------------------------------------|---|
| ID | PROFILE | DESCRIPTION |
| ✓ A | corsa_esc_off_sinesweep | Corsa/ESC off/sinesweep at 50,80,100,120 km/h |
| B | ayconst_steerconst_throttleramp | Ay constant at 50,80,100 km/h, constant steer/ throttle ramp |
| ✓ C | ESConcorse_HDL_hottires | ESC on/ handling circuit/ corsa |
| ✓ D | corsa_esc_off_ramp_acc_brake | Corsa/ esc off /slow ramp /steer at 50,80,100,120 [km/h] /accelerations and breaking 0 100 km/h |
| ✓ E | acc_brk_steerconst_sine_ramp | Sport /ESC off/ accelerations and breaking 0 100 km/h/ constant steer/ ramp throttle/ sine sweep at 50,80,100 km/h |
| ✓ F | ramp_acc_brk_sine_ayconst_step_abs | Strada/esc off /slow ramp/steer at 50, 80 , 100 [km/h] /sine sweep a 50, 80, 100 [km/h] /Ay costante 100, 80, 50 [km/h] /steer cost ramp throttle/ step steer while cornering a 80, 100 [km/h] /ABS while cornering |
| ✓ G | ayconst_steerconst_thrramp | Ay costante at 50, 80 e 100 [km/h]/ constant steer and ramp throttle |
| AA | warmup | Aquisition test |
| H | sport_esc_on | Sport/ ESC on |
| I | sport_esc_off_clean | Sport/ESC off/clean |
| J | sport_esc_off_dirty | Sport /ESC off/dirty |
| K | corsa_esc_off_used_tires_steer | Corsa/ESC off/constant steer/ worn tires |
| L | sport_esc_off_steer_usedtires | Sport/ESC off/constant steer/ worn tires |

| DRY | | |
|-----|------------------------------------|--|
| ID | PROFILE | DESCRIPTION |
| M | corsa_esc_off_ steer | Corsa/ESC off/constant steer |
| N | strada_esc_off_ steer_usedtires | Strada/ESC off/constant steer/ worn tires |
| O | strada_esc_off_ steer_ay_ | Strada/ESC off/constant steer/ ay constant |
| S | corsa_esc_off_ HDL | Corsa/ESC off/ handling circuit |
| T | corsa_esc_on_ cleanHDL | Corsa/ESC on/ clean / handling circuit |
| U | _strada_escon_cleanHDL | Strada/ESC on /clean /Handling circuit |
| V | strada_escoff_ HDL | Strada/ESC off/clean /handling circuit |
| W | strada_escoff_ dirty | Strada/ESC off/dirty |
| X | _escon_dirty | Strada/ESC on/ dirty |
| Y | corse_escoff_ dirty | Corsa/ESC off/dirty |
| Z | corsa_escon_ dirty | Corsa/ESC on/dirty |
| AD | sport_esc_on_ dirty | Sport/ESC on/ dirty |
| AE | test_acquisition | Test aquisition |
| AF | sport_esc_off_ steer_usedtires | Sport/ESC off/steer < 45°/ steer = 80°/ used tires |
| AG | strada_esc_off_ usedtires | Sport/ESC off/steer < 45°/ steer < 80°/ |
| AI | sport_esc_off_ ayconst_steer | Sport/ESC off/ Ay constant /steer constant/ ramp throttle |
| AJ | sport_esc_off_ ramp | Sport/ ESC off/ramp steer at constant speed |
| AK | corsa_esc_off_ slowramp | Corsa/ESC off/slow ramp steer/ steer cost/ ramp throttle |

| DRY | | |
|------|---------------------------------------|--|
| ID | PROFILE | DESCRIPTION |
| AO | sport_esc_on_ outtrack | Sport/ESC on/dirty / out of track at second 107 |
| AP | corsa_esc_ off | Corsa/ESC off/constant speed 100-120 km/h |
| AQ | strada_esc_off_ steer | Strada/ESC off/ constant speed 80 [km/h] steer < 90° / constant speed 80 [km/h] steer 45°/ constant speed 100 [km/h] steer < 30° |
| BC | corsa_esc_off_ steer | Corsa/ ESC off/ constant speed 80 [km/h] steer < 90°/ constant speed 80 [km/h] steer 45°/ constant speed 100 [km/h] steer < 30° |
| AS | sport_esc_off_ steer | Sport/ESC OFF/ constant speed 80 [km/h] steer < 90°/ constant speed 80 [km/h] steer 45°/ constant speed 100 [km/h] steer < 30° |
| AT | sport_esc_off_ steer | Sport/ESC OFF/ constant speed 80 [km/h] steer < 90°/ constant speed 80 [km/h] steer 45°/ constant speed 100 [km/h] steer < 30° |
| AU | sport_esc_off_ drifiting | Sport/ ESC off/ drifting |
| AV | sport_esc_off_ drifiting | Sport/ ESC off/ drifting |
| AW | corsa_esc_off_ drifiting | Corsa/ ESC off/ drifting /aquisition also at low speed |
| AX | strada_esc_off_ steer | Strada/ESC OFF /constant speed 100 [km/h] steer 30° |
| AY | corsa_esc_off_ steer | Corsa/ESC OFF /constant speed 100 [km/h] steer 30° |
| AZ | sport_esc_off_ steer | Sport/ESC OFF /constant speed 100 [km/h] steer 30° |
| BA | sport_esc_off_ steer | Sport/ESC OFF /constant speed 100 [km/h] steer 30° |
| ✓ BB | corsa_esc_off_ abs_while_cornering | Corsa/ESC off/ ABS on while cornering |

Table 4. DRY manoeuvres

| WET | | |
|------|---------------------------|---|
| ID | PROFILE | DESCRIPTION |
| BE | calibration_racelogic | calibration |
| BF | acc_brak_II | accelerations [0,100] [km/h] brakings [100,0] [km/h] |
| BG | slalom_sinesweep | Slalom/sine sweep at 80 km/h |
| BH | stepsteer | step steer while cornering |
| ✓ BI | heatuptire_ayconst | Heating up tires while ay constant |
| BJ | steer_const_throttle_ramp | Constant steer/ throttle ramp |
| BK | warmup | warmup |
| BL | racelogic_calib_II | calibration |
| ✓ BQ | slowramp_sx | Slow ramp/steer SX |
| BS | slow_ramp_dx | Slow ramp/steer DX |
| BT | steerconst_throttleramp | Steer constant/throttle ramp |
| BV | acc_brake_III | Acceleration and braking |
| BW | abs_while_turn | ABS while cornering |
| BZ | calib_racelogic | calibration |
| CA | gas_release | Gas release in second gear at 80 km/h |
| CB | gas_release | Gas release in second gear at 80 km/h |
| CG | sine_sweep_80 | Sine sweep at 80 [km/h] |
| CK | steer_const | Constant steer/throttle ramp |
| CL | steer_const | Constant steer/throttle ramp |
| CM | ABS_esc_sport | ABS while cornering/ESC on/ Sport (50 [km/h]) |
| CN | abs_esc_sport | ABS while cornering / ESC on/ Sport (80 [km/h]) |
| ✓ CP | ESC_off_sporco | Corsa/ESC off /dirty |

| WET | | |
|------|-------------------------|--------------------------|
| ID | PROFILE | DESCRIPTION |
| CQ | ESC_off_pulito | Corsa/ESC off /clean |
| ✓ CR | corsa_esc_off_sporco | Corsa/ESC off /dirty |
| ✓ CS | esc_sport_pulito | Sport/ESC on /clean |
| ✓ CV | sport_esc_off_sporco | Sport/ESC off /dirty |
| CW | sport_esc_on | Sport/ESC off /dirty |
| CX | sport_esc_on_pulito | Sport/ESC on /clean |
| CY | corsa_esc_on_sporco | Corsa/ESC on /dirty |
| DA | strada_esc_on_sporco | Strada/ESC on /dirty |
| DD | strada_esc_off_pulito | Strada/ESC off /clean |
| DE | strada_esc_off_sporco | Strada/ESC off /dirty |
| DI | strada_esc_on_sporco | Strada/ESC on /dirty |
| DK | strada_esc_off_drifting | Strada/ESC off /drifting |
| DL | strada_esc_off_clean | Strada/ESC off /clean |
| DM | strada_esc_on_dirty | Strada/ESC on /dirty |
| DN | strada_esc_on_clean | Strada/ESC on /clean |
| DO | strada_esc_off_dirty | Strada/ESC off /dirty |
| DP | strada_esc_off_clean | Strada/ESC off /clean |
| CH | sine_sweep_80 | Sine sweep at 80 [km/h] |

Table 5. WET manoeuvres

| ICY | | |
|------|--|---|
| ID | PROFILE | DESCRIPTION |
| EG | 30SSI_2017_1_21_9_18_18 | Sine sweep 30 km/h |
| ✓ EK | 60SKP_2017_1_20_14_46_48 | Skidpad 60 km/h |
| ET | 60SSI_2017_1_21_9_19_54 | Sinesweep 60 km/h |
| ✓ EV | 90LC_2017_1_21_9_49_42 | Lane change 90 km/h |
| ✓ EW | 90LC_2017_1_21_9_52_31 | Lane change 90 km/h |
| ✓ EX | 90SSI_2017_1_20_14_23_48 | Sine sweep 90 km/h |
| ✓ FA | 90SSI_2017_1_21_9_21_20 | Sine sweep 90 km/h |
| FD | HDL_2017_1_21_9_35_16 | Handling circuit |
| FG | HDL_AGG_2017_1_21_15_1_25 | Handling circuit/dirty |
| ✓ FH | HDL_Neutral_2017_1_20_14_40_29 | Handling circuit/ clean |
| FI | HDL_neutral_2017_1_21_14_59_39 | Handling circuit/ clean |
| ✓ HM | SKP_2017_1_21_15_6_4 | skidpad |
| FK | HDL_strada_2017_1_21_9_25_10 | Handling circuit/strada |
| ✓ FL | HDL_strada_AGG_2017_1_21_9_41_18 | Handling circuit/strada/dirty |
| ✓ FM | Land_hdl_sport_agg_RWS_2017_1_23_16_18_49 | Handling circuit/sport/dirty/ rear wheels steering |
| ✓ FN | Land_hdl_sport_agg_RWS_2017_1_23_16_29_7 | Handling circuit/sport/dirty/ rear wheels steering |
| FO | land_hdl_sport_aggressive_2017_1_23_11_31_24 | Handling circuit/sport/dirty |
| FP | land_hdl_sport_aggressive_2017_1_23_11_41_29 | Handling circuit/sport/dirty |
| FQ | Land_HDL_Sport_neutral_RWS_2017_1_23_17_0_3 3 | Handling circuit/sport/clean/ rear wheels steering |
| ✓ FR | LAND_HDL_strada_AGG_1_21_16_52_57 | Handling circuit/strada/dirty |
| FS | Land_hdl_strada_agg_RWS_2017_1_23_15_59_39 | Handling circuit/strada/dirty/ rear wheels steering |

| ICY | | |
|------|--|--|
| ID | PROFILE | DESCRIPTION |
| FT | Land_hdl_strada_agg_RWS_2017_1_23_16_9_11 | Handling circuit/strada/dirty/ rear wheels steering |
| FU | land_hdl_strada_aggressive_2017_1_23_10_55_50 | Handling circuit/strada/dirty |
| FV | land_hdl_strada_aggressive_2017_1_23_11_20_54 | Handling circuit/strada/dirty |
| FW | LAND_HDL_strada_neutral_2017_1_21_17_19_57 | Handling circuit/strada/clean |
| FX | land_hdl_strada_neutral_2017_1_23_10_32_22 | Handling circuit/strada/clean |
| FY | land_hdl_strada_neutral_2017_1_23_10_44_52 | Handling circuit/strada/clean |
| FZ | Land_HDL_strada_neutral_RWS_2017_1_23_16_41_12 | Handling circuit/strada/clean/ rear wheels steering |
| ✓ GA | LandHDL_corsa_AGG_2017_1_21_17_4_28 | Handling circuit/corsa/dirty |
| ✓ GB | LandHDL_strada_2017_1_21_16_37_57 | Handling circuit/strada/ |
| ✓ HE | long_acc_brk_corsa_2017_1_24_12_4_50 | Longitudinal acceleration/braking/ corsa |
| HF | long_acc_brk_sport_2017_1_24_12_3_12 | Longitudinal acceleration/braking/ sport |
| HG | long_acc_brk_strada_2017_1_24_12_0_37 | Longitudinal acceleration/braking/ strada |
| HH | long_acc_brk_strada_2017_1_24_12_12_0 | Longitudinal acceleration/braking/ strada |
| HI | rose_corsa_agg_rws_2017_1_24_11_58_19 | Rosengarten/corsa/dirt y/ rear wheels steering |
| HJ | rose_hdl_sport_agg_rws_2017_1_24_11_54_37 | Rosengarten/handling circuit/sport/dirty/ rear wheels steering |

| ICY | | |
|-----|--|--|
| ID | PROFILE | DESCRIPTION |
| HK | rose_hdl_strada_agg_rws_2017_1_24_11_34_22 | Rosengarten/handling circuit/strada/dirty/rear wheels steering |
| HL | SKP_2017_1_21_15_5_13 | skidpad |
| HN | Urban_strada_2017_1_24_11_13_1 | Urban/strada |
| HO | urban_strada_2017_1_24_12_16_25 | Urban/strada |

Table 6. ICY manoeuvres

Driving Mode [2]:

- “Strada” This is the setting designed for daily driving. The exhaust baffles only open at high engine revolutions, the transmission shifts smoothly, and the power split is 30% front and 70% rear.
- “Sport” The exhaust baffles open at lower engine revolutions, the throttle has more prompt response. and the power split is 10% front and 90% rear.
- “Corsa” This is the setting designed for circuit driving. Shift times further reduced. And power split is 20% front, 80% rear.

Driving style [2]:

- Dirty: aggressive
- Clean: neutral
- Drifting

Appendix B

In this section a complete list of all the features computed for the Road condition identification is presented. Consider that some of them are repeated twice in the computation, so here are present 70 features instead of 140.

| # | Feature | Source | Unit |
|----|---|--------|--------|
| 1 | Height and position of first peak (2 each) | a_x | [W/Hz] |
| 2 | Height and position of second peak(2 each) | a_x | [W/Hz] |
| 3 | Height and position of third peak (2 each) | a_x | [W/Hz] |
| 4 | Height and position of fourth peak (2 each) | a_x | [W/Hz] |
| 5 | Height and position of fifth peak (2 each) | a_x | [W/Hz] |
| 6 | Height and position of sixth peak (2 each) | a_x | [W/Hz] |
| 7 | Height and position of first peak (2 each) | a_y | [W/Hz] |
| 8 | Height and position of second peak (2 each) | a_y | [W/Hz] |
| 9 | Height and position of third peak (2 each) | a_y | [W/Hz] |
| 10 | Height and position of fourth peak (2 each) | a_y | [W/Hz] |
| 11 | Height and position of fifth peak (2 each) | a_y | [W/Hz] |
| 12 | Height and position of sixth peak (2 each) | a_y | [W/Hz] |
| 13 | Height and position of first peak (2 each) | ψ | [W/Hz] |
| 14 | Height and position of second peak(2 each) | ψ | [W/Hz] |
| 15 | Height and position of third peak (2 each) | ψ | [W/Hz] |
| 16 | Height and position of fourth peak (2 each) | ψ | [W/Hz] |
| 17 | Height and position of fifth peak (2 each) | ψ | [W/Hz] |
| 18 | Height and position of sixth peak (2 each) | ψ | [W/Hz] |
| 19 | Height and position of first peak (2 each) | HL | [W/Hz] |
| 20 | Height and position of second peak(2 each) | HL | [W/Hz] |
| 21 | Height and position of third peak (2 each) | HL | [W/Hz] |
| 22 | Height and position of fourth peak (2 each) | HL | [W/Hz] |
| 23 | Height and position of fifth peak (2 each) | HL | [W/Hz] |
| 24 | Height and position of sixth peak (2 each) | HL | [W/Hz] |
| 25 | Height and position of first peak (2 each) | HR | [W/Hz] |
| 26 | Height and position of second peak(2 each) | HR | [W/Hz] |
| 27 | Height and position of third peak (2 each) | HR | [W/Hz] |
| 28 | Height and position of fourth peak (2 each) | HR | [W/Hz] |
| 29 | Height and position of fifth peak (2 each) | HR | [W/Hz] |
| 30 | Height and position of sixth peak (2 each) | HR | [W/Hz] |
| 31 | Height and position of first peak (2 each) | VL | [W/Hz] |
| 32 | Height and position of second peak(2 each) | VL | [W/Hz] |
| 33 | Height and position of third peak (2 each) | VL | [W/Hz] |
| 34 | Height and position of fourth peak (2 each) | VL | [W/Hz] |

| # | Feature | Source | Unit |
|----|--|--------------|-------------------------------------|
| 35 | Height and position of fifth peak (2 each) | <i>VL</i> | [W/Hz] |
| 36 | Height and position of sixth peak (2 each) | <i>VL</i> | [W/Hz] |
| 37 | Height and position of first peak (2 each) | <i>VR</i> | [W/Hz] |
| 38 | Height and position of second peak (2 each) | <i>VR</i> | [W/Hz] |
| 39 | Height and position of third peak (2 each) | <i>VR</i> | [W/Hz] |
| 40 | Height and position of fourth peak (2 each) | <i>VR</i> | [W/Hz] |
| 41 | Height and position of fifth peak (2 each) | <i>VR</i> | [W/Hz] |
| 42 | Height and position of sixth peak (2 each) | <i>VR</i> | [W/Hz] |
| 43 | Total power in 5 adjacent frequency bands (5 each) | a_x | [W] |
| 44 | Total power in 5 adjacent frequency bands (5 each) | a_y | [W] |
| 45 | Total power in 5 adjacent frequency bands (5 each) | $\dot{\psi}$ | [W] |
| 46 | Total power in 5 adjacent frequency bands (5 each) | <i>HL</i> | [W] |
| 47 | Total power in 5 adjacent frequency bands (5 each) | <i>HR</i> | [W] |
| 48 | Total power in 5 adjacent frequency bands (5 each) | <i>VL</i> | [W] |
| 49 | Total power in 5 adjacent frequency bands (5 each) | <i>VR</i> | [W] |
| 50 | Peak to RMS | a_x | [-] |
| 51 | Peak to RMS | a_y | [-] |
| 52 | Peak to RMS | $\dot{\psi}$ | [-] |
| 53 | Peak to RMS | <i>HL</i> | [-] |
| 54 | Peak to RMS | <i>HR</i> | [-] |
| 55 | Peak to RMS | <i>VL</i> | [-] |
| 56 | Peak to RMS | <i>VR</i> | [-] |
| 57 | Standard deviation | a_x | [m/s ²] |
| 58 | Standard deviation | a_y | [m/s ²] |
| 59 | Standard deviation | $\dot{\psi}$ | [deg/s] |
| 60 | Standard deviation | <i>HL</i> | [km/h] |
| 61 | Standard deviation | <i>HR</i> | [km/h] |
| 62 | Standard deviation | <i>VL</i> | [km/h] |
| 63 | Standard deviation | <i>VR</i> | [km/h] |
| 64 | Variance | a_x | [m ² /s ⁴] |
| 65 | Variance | a_y | [m ² /s ⁴] |
| 66 | Variance | $\dot{\psi}$ | [deg ² /s ²] |
| 67 | Variance | <i>HL</i> | [km ² /h ²] |
| 68 | Variance | <i>HR</i> | [km ² /h ²] |
| 69 | Variance | <i>VL</i> | [km ² /h ²] |
| 70 | Variance | <i>VR</i> | [km ² /h ²] |

Table 7. List of features computed for the road condition identification.

In this table are present 70 elements, instead of 140 as stated in the text, because the features that require more than one computation for each source have been considered together, just to simplify a little bit the table. The sources are a_x , longitudinal acceleration, a_y , lateral acceleration, $\dot{\psi}$, yaw rate, *VL*, speed of the front left wheel, *VR* speed of the front right wheel, *HR* speed of the rear right wheel and *HL* speed of the rear left wheel.