

Soft sensors in automotive applications

Marco Carratù

UNIVERSITY OF SALERNO



DEPARTMENT OF INDUSTRIAL ENGINEERING

*Ph.D. Course in Industrial Engineering
Curriculum in Electronic Engineering - XXXI Cycle*

SOFT SENSORS IN AUTOMOTIVE APPLICATIONS

Supervisor

Prof. Consolatina Liguori

Ph.D. student

Marco Carratù

Ph.D. Course Coordinator

Prof. Ernesto Reverchon

Publications resulting from this work

Capriglione, D., Carratu, M., Pietrosanto, A., & Sommella, P. (2019). A Virtual ANN-Based Sensor for IFD in Two-Wheeled Vehicle. In book: Sensors - Proceedings of the Fourth National Conference on Sensors, February 21- 23, 2018, Catania, Italy Publisher: Springer, DOI: 10.1007/978-3-030-04324-7_55

Capriglione, D., Carratu, M., Pietrosanto, A., & Sommella, P. (2018). Analytically redundancy based IFDI scheme for semi-active suspension systems in motorcycle. Paper presented at the Journal of Physics: Conference Series, 1065(10) doi:10.1088/1742-6596/1065/10/102011.

Carratu, M., Pietrosanto, A., Sommella, P., & Paciello, V. (2018). Measuring suspension velocity from acceleration integration. Paper presented at the Proceedings - IEEE 16th International Conference on Industrial Informatics, INDIN 2018, 933-938. doi:10.1109/INDIN.2018.8472039.

Carratù, M., Capriglione, D., Liguori, C., Sommella, P., Paciello, V., & Pietrosanto, A. (2018). Virtual sensors for two-wheeled vehicles control. Congresso Nazionale GMEE 2018, Padova. ISBN 978-88-31901-06-2 (pp. 201-202).

Carratù, M., Pietrosanto, A., Sommella, P. & Paciello, P. (2018). Semi-active suspension system for motorcycles: From the idea to the industrial product. IEEE International Instrumentation and Measurement Technology Conference: Discovering New Horizons in Instrumentation and

Measurement Proceedings (I2MTC), Houston. DOI: 10.1109/I2MTC.2018.8409829 (pp.423-428).

Capriglione, D., Carratù, M., Pietrosanto, A., & Sommella, P. (2018). Real-time implementation of an IFD scheme for motorcycle sensors. IEEE International Instrumentation and Measurement Technology Conference: Discovering New Horizons in Instrumentation and Measurement Proceedings (I2MTC2018), Houston. DOI: 10.1109/I2MTC.2018.8409833 (pp.215-220).

Capriglione, D., Carratù, M., Pietrosanto, A., & Sommella, P. (2018). NARX ANN-based instrument fault detection in motorcycle. Measurement: Journal of the International Measurement Confederation, 117, 304-311. doi:10.1016/j.measurement.2017.12.026

Carratù, M., Pietrosanto, A., Paciello, V., & Sommella, P. (2017). Velocity estimation from acceleration measurements in motorcycle suspensions. Congresso Nazionale GMEE 2017, Modena. ISBN 978-88-903149-9-5 (pp 225-226).

Capriglione, D., Carratù, M., Sommella, P., & Pietrosanto, A. (2017). ANN-based IFD in Motorcycle Rear Suspension. 15th IMEKO TC10 Workshop on Technical Diagnostics 2017 - "Technical Diagnostics in Cyber-Physical Era" (IMEKO TC10), Budapest. EID: 2-s2.0-85041209418, ISBN: 978-1-5108-4491-9 (pp. 22-27).

Capriglione, D., Carratù, M., Liguori, C., Paciello, V., & Sommella, P. (2017). A Soft Stroke Sensor for Motorcycle Rear Suspension. Measurement: Journal of the International Measurement Confederation. DOI: 10.1016/j.measurement.2017.04.011 (Volume 106, 1 August 2017, pp. 46-52).

Carratù, M., Pietrosanto, A., Sommella, P. & Paciello, P. (2017). Velocity Prediction from Acceleration Measurement for Two Wheels Vehicle. I2MTC 2017 - 2017 IEEE International Instrumentation and Measurement Technology Conference,

Proceedings, Torino. DOI: 10.1109/I2MTC.2017.7969943 (pp. 1619-1624).

Capriglione, D., Carratù, M., Liguori, C., Paciello, V., & Sommella, P. (2016). Prediction of Motorcycle Suspension Stroke through Dynamic Neural Networks. WSEAS Transactions on Systems and Control. ISSN / E-ISSN: 1991-8763 / 2224-2856, Volume 11, 2016, Art. #41, (pp. 376-383).

Acocella, G., Carratù, M., Paciello, V., Pietrosanto, A., & Sommella, P. (2016). Ricostruzione della velocità di una sospensione di un motoveicolo basata sull'uso di accelerometri. Congresso Nazionale GMEE 2016, Benevento. ISBN 978-88-940453-6-9 (pp 39-40). Contributo in Atti di convegno.

Pietrosanto, A., Sommella, M., Carratù, M., De Santo and L. Feo. (2016). Suspension Stroke Reconstruction from Acceleration. Proceedings of International Conference on Technology Management (ICTM 2016), Chicago. ISBN: 978-0-9853540-9-

Contents

Contents.....	I
List of figures	V
List of tables	IX
Abstract	XI
Introduction	XIII
Chapter I.....	1
The motorcycle suspension systems.....	1
I.1 Physical model of the suspension systems.....	2
I.1.1 The spring	6
I.1.2 The damper	7
I.2 Passive suspension systems.....	9
I.3 Active suspension systems.....	10
I.4 Semi-active suspension systems	11
I.5 A real case: The MAGNETO system.....	15
I.5.1 Description of the MAGNETO control system.....	17
I.5.2 Description of the MAGNETO damping system.....	19
Chapter II.....	21
Hard and soft sensors for the control of a semi-active suspension system...	21
II.1 Physical dynamics in a motorcycle system.....	21
II.1.1 Center of gravity in a motorcycle body	22
II-I.2 The pitch motions.....	23
II.1.3 The roll motions.....	23
II.1.4 The yaw motions.....	24
II.1.5 Role of suspensions	25

II.2 Pitch driven control strategy	25
II.2.1 Typology of control scheme	26
II.3 Set of sensors needed for the control strategy	27
II.3.1 Stroke sensors	28
II.3.2 Gyroscopes	29
II.3.3 Accelerometers	30
II.3.4 Longitudinal speed sensors	31
II.4 Soft sensors	32
II.5 The test bed	35
Chapter III	37
Rear stroke soft sensor based on NARX network	37
III.1 The rear stroke soft sensor	37
III.2 The measurement setup for the rear stroke soft sensor	39
III.3 Tuning of the NARX network	40
III.4 Performance of the rear stroke soft sensor	49
Chapter IV	53
Front stroke velocity soft sensors based on accelerometers	53
IV.1 The measurement setup	53
IV.2 Front stroke velocity soft sensor based on digital filtering	54
IV.3 Tuning of the digital filters parameters	56
IV.4 Performance of the front stroke soft sensor based on digital filtering ..	57
IV.5 Front stroke soft sensor based on NARX network	59
IV.6 Tuning of the NARX network	59
IV.7 Performance of the front stroke velocity soft sensor based on NARX network	60
IV.8 Comparison of the proposed front stroke velocity soft sensors	61
Chapter V	65
IFD scheme for the rear stroke sensor	65
V.1 The IFD scheme	66
V.1.1 Soft sensor used	67
V.1.2 Residual generator	68
V.1.3 Decision Maker	71

V.2 Validation of the IFD scheme.....	71
V.2.1 Short and open circuit faults.....	71
V.2.2 Hold fault.....	72
V.2.3 Losing calibration fault.....	73
V.3 Experimental Results.....	73
V.4 On-line fault detection scheme.....	79
V.5 The realized prototype.....	79
V.5.1 Hardware.....	80
V.5.2 Firmware.....	81
V.6 Experimental results on the road.....	88
V.6.1 Faults introduced in the control loop.....	88
V.6.2 Diagnostic and dynamic performance.....	88
Chapter VI.....	91
Conclusions.....	91
References.....	93
Appendix I.....	99
NARX network.....	99

List of figures

Chapter I: The motorcycle suspension systems	
Figure I.1 <i>Main blocks of a vehicle</i>	3
Figure I.2 <i>Main motions of a vehicle</i>	3
Figure I.3 <i>The quarter-car model</i>	4
Figure I.4 <i>The half-car model</i>	4
Figure I.5 <i>The full-car model</i>	5
Figure I.6 <i>Acceleration Power Spectrum of the motorcycle vibrations</i>	5
Figure I.7 <i>The spring</i>	6
Figure I.8 <i>Simple harmonic motion</i>	7
Figure I.9 <i>The damper</i>	7
Figure I.10 <i>Muted harmonic motion</i>	8
Figure I.11 <i>Damped harmonic motion in critical damping condition</i>	8
Figure I.12 <i>Passive suspension</i>	9
Figure I.13 <i>Model of a passive suspension</i>	9
Figure I.14 <i>Model of an active suspension</i>	10
Figure I.15 <i>Semi-active suspension scheme</i>	12
Figure I.16 <i>CDC damper</i>	13
Figure I.17 <i>The magnetorheological fluid</i>	14
Figure I.18 <i>Magnetorheological suspension</i>	15
Figure I.19 <i>The Magneto kit</i>	16
Figure I.20 <i>ECU of the Magneto system</i>	17
Figure I.21 <i>The Magneto suspensions</i>	19
Figure I.22 <i>Block Diagram of the MAGENTO driver system</i>	19

Chapter II: Hard and soft sensors for the control of a semi-active suspension system	
Figure II.1 <i>Coordinates of the motorcycle center of gravity</i>	22
Figure II.2 <i>Motorcycle pitch motion</i>	23
Figure II.3 <i>Motorcycle roll motion</i>	24
Figure II.4 <i>Motorcycle yaw motion</i>	24
Figure II.5 <i>Example of a closed loop strategy</i>	27
Figure II.6 <i>Sensing process</i>	27
Figure II.7 <i>Linear potentiometer</i>	28
Figure II.8 <i>The gyroscope model</i>	30
Figure II.9 <i>Operation principle of an accelerometer</i>	30
Figure II.10 <i>Speed sensor</i>	31
Figure II.11 <i>Pitch motion during acceleration</i>	33
Figure II.12 <i>Pitch motion during breaking</i>	34
Figure II.13 <i>Test Bed</i>	36
Chapter III: Rear stroke soft sensor based on NARX network	
Figure III.1 <i>Scheme of the rear stroke soft sensor</i>	39
Figure III.2 <i>The trial and error procedure</i>	39
Figure III.3 <i>Example of training dataset</i>	41
Figure III.4 <i>Magnification of the undersized NARX output (red line) versus the Training set (blu line)</i>	43
Figure III.5 <i>Conversion from Series-Parallel to Parallel-Parallel scheme</i> .	44
Figure III.6 <i>Prediction of the Training set</i>	44
Figure III.7 <i>Example of a REC curve</i>	46
Figure III.8 <i>REC: Comparison among the best 10 neural networks on Training set</i>	47
Figure III.9 <i>REC: Comparison among different tests set using $d_{in}=10$ and Neurons =15</i>	47
Figure III.10 <i>REC: Comparison among different test sets using $d_{in}=10$ and Neurons =20</i>	48
Figure III.11 <i>REC: Comparison among different test sets using $d_{in}=10$ and Neurons =25</i>	48

Figure III.12 REC: Comparison among test set 1 and 3 using the three best neural networks achieved	49
Figure III. 13 Prediction of the rear suspension stroke.....	50
Figure III. 14 Magnification of Figure III.13	50
Figure III.15 SOE curve	51
Figure III.16 SOE: Comparison among HCM soft sensor and NARX soft sensor	52
Chapter IV: Front stroke velocity soft sensors based on accelerometers	
Figure IV.1 Sensors used for the front stroke velocity soft sensor	54
Figure IV. 2 Front stroke velocity soft sensor scheme based on digital filters	55
Figure IV.3 Spectrum of the suspension acceleration during a ride.....	55
Figure IV.4 Sensitivity analysis for the front stroke velocity soft sensor based on digital filtering.....	57
Figure IV.5 Prediction of the front suspension stroke using the digital filter	57
Figure IV.6 Magnification of Figure IV.5 with respect to high suspension velocities;.....	58
Figure IV.7 Magnification of Figure IV.5 with respect to low suspension velocities.....	58
Figure IV.8 Front stroke velocity soft sensor scheme based on NARX.....	59
Figure IV.9 Predicted front stroke velocity vs real front stroke velocity	60
Figure IV.10 Magnification of Fig. IV.9 with respect to high suspension velocities.....	61
Figure IV.11 Magnification of Fig. IV.9 with respect to low suspension velocities.....	61
Figure IV.12 Relative prediction error per velocity class.....	62
Figure IV.13 Distribution of relative error percentage with respect to the suspension velocity classes.....	63
Figure IV.14 Histogram of suspension velocity for the Testing dataset.....	63
Figure IV. 15 Example of semi-active suspension control: stroke velocity and controlled current.....	64
Chapter V: IFD scheme for the rear stroke sensor	
Figure V.1 Block-diagram of the proposed IFD scheme.....	67

Figure V.2 Prediction of the rear suspension stroke by the NARX Network for dataset.....	68
Figure V.3 Magnification of figure V.2 with reference to the highest peaks	69
Figure V.4 Measured and predicted normalized rear stroke.	69
Figure V.5 Moving averaged residual versus L_s	69
Figure V.6 SOE curves for NN as function of the window length L	70
Figure V.7 Magnification of Figure V.6	70
Figure V.8 The proposed detection rules	71
Figure V.9 Open circuit fault	72
Figure V.10 Short circuit fault	72
Figure V.11 Hold fault	73
Figure V.12 Uncalibration fault (10%).....	73
Figure V.13 The realized IFD ECU	80
Figure V.14 Flow-chart of the proposed five-step procedure for the MCU-implementation of the IFD scheme.....	81
Figure V.15 Some of the main practices adopted for final code optimization (STEP 4 of the procedure): Loop Jamming.....	85
Figure V.16 Some of the main practices adopted for final code optimization (STEP 4 of the procedure): Loop Unrolling.	85
Figure V.17 Some of the main practices adopted for final code optimization (STEP 4 of the procedure): Use of FPU.	85
Figure V.18 Percentage error between the reference outputs (provided by Matlab code) and the firmware ones (provided by the Final optimized code).	87

List of tables

Chapter II: Hard and soft sensors for the control of a semi-active suspension system	
Table II.1 <i>Sensors used on the test bed</i>	35
Chapter III: Rear stroke soft sensor based on NARX network	
Table III.1 <i>Main sensors used for the rear stroke soft sensor</i>	40
Table III.2 <i>Comparison between 25 neural networks</i>	45
Chapter IV: Front stroke velocity soft sensors based on accelerometers	
Table IV.1 <i>Main parameters of the stroke velocity soft sensor based on digital filtering approach</i>	56
Chapter V: IFD scheme for the rear stroke sensor	
Table V.1 <i>Detection of losing calibration faults: False Alarm percentage</i> .	75
Table V.2 <i>Detection of losing calibration faults: Correct Detection percentage</i>	75
Table V.3 <i>Detection of losing calibration faults: Missed Detection percentage</i>	76
Table V.4 <i>Detection of Open Circuit faults: False Alarm percentage</i>	76
Table V.5 <i>Detection of Open Circuit faults: Correct Detection percentage</i>	76
Table V.6 <i>Detection of Open Circuit faults: Missed Detection percentage</i> .	77
Table V.7 <i>Detection of Short Circuit faults: False Alarm percentage</i>	77
Table V.8 <i>Detection of Short Circuit faults: Correct Detection percentage</i>	77
Table V.9 <i>Detection of Short Circuit faults: Missed Detection percentage</i> .	78
Table V.10 <i>Detection of Hold faults: False Alarm percentage</i>	78
Table V.11 <i>Detection of Hold faults: Correct Detection percentage</i>	78
Table V.12 <i>Detection of Hold faults: Missed Detection percentage</i>	79

Table V.13 <i>Execution times measured with experimental tests (A frequency clock of 22 MHz is involved) and memory required for the implementation of the IFD scheme.</i>	86
Table V.14 <i>Diagnostic performance of the IFD scheme</i>	88
Table V.15 <i>Dynamic performance of the IFD scheme (μ is the mean value, σ is the standard deviation).</i>	89

Abstract

In this work, design and validation techniques of two soft sensors for the estimation of the motorcycle vertical dynamic have been proposed. The aim of this work is to develop soft sensors able to predict the rear and front stroke of a motorcycle suspension. This kind of information are typically used in the control loop of semi-active or active suspension systems. Replacing the hard sensor with a soft sensor, enable to reduce cost and improve reliability of the system. An analysis of the motorcycle physical model has been carried out to analyze the correlation existing among motorcycle vertical dynamic quantities in order to determine which of them are necessary for the development of a suspension stroke soft sensor. More in details, a first soft sensor for the rear stroke has been developed using a Nonlinear Auto-Regressive with eXogenous inputs (NARX) neural network. A second soft sensor for the front suspension stroke velocity has been designed using two different techniques based respectively on Digital filtering and NARX neural network. As an example of application, an Instrument Fault Detection (IFD) scheme, based on the rear stroke soft sensor, has been shown. Experimental results have demonstrated the good reliability and promptness of the scheme in detecting different typologies of faults as losing calibration faults, hold-faults, and open/short circuit faults thanks to the soft sensor developed. Finally, the scheme has been successfully implemented and tested on an ARM microcontroller, to confirm the feasibility of a real-time implementation on actual processing units used in such context.

Introduction

Today we can assist to an increasing adoption of sensors and electronic devices inside automobiles and motorcycles. They are widely employed in different types of applications like Electronic fuel injection (EFI), Antilock-braking system (ABS), Electronic stability program (ESP), Active or Semi- Active suspension system and so on.

The correct operation of each subsystem strongly depends on the reliability and accuracy of the sensors output involved in the control loop, therefore, it becomes fundamental the introduction of strategies aimed to reduce the number of sensors involved in it.

As an example, one of the new most critical subsystems in the motorcycle context is represented by the semi-active or active suspension system, which compared to classical suspension systems, can change the damping coefficient as a function of the suspension stroke velocity, the pitch rate, and/or other measurements about the vertical dynamic of the motorcycle. A suspension system directly influences comfort and stability, ensuring the contact between tires and road, and at the same time isolating the vehicle frame from the road roughness; for these reasons, the control-strategy of a semi-active or active suspensions system need to be performant and reliable.

The quantities that must be measured to have an optimal control strategy come out from a set of sensors typically composed by biaxial/triaxial gyroscopes, relative stroke sensors and longitudinal velocity sensor. More in details, the biaxial/triaxial gyroscopes are used to measure yaw, roll, and pitch; the stroke sensors are used to measure the vertical extensions and vertical compressions of the rear and front suspensions and the longitudinal velocity sensors are used to measure the motorcycle speed. The correct operation of all the sensors involved in the control-loop, influence the performance of a semi-active or active suspension system, mainly in terms of response time, control accuracy and stability.

For example, the linear potentiometers are the most used sensors for linearity and simplicity as stroke sensors, but they suffer of wear, tear and aging higher than the other sensors involved in the control loop, reducing in the long run the performance of the system.

To save the efficiency and the effectiveness of the suspension control strategy a “Soft Sensor” could be used.

More in details, “Soft Sensor” generally means the process of estimating any system or process variable by adopting mathematical models, replacing some physical devices, and using data acquired from some other available sensors. It is possible to differentiate two principal categories of soft sensors: Model-driven and Data-driven. The first typology of soft sensors is commonly based on First Principle Models (FPM), they are also known as phenomenological models or white-box model (Fortuna, Graziani, Rizzo, & Xibilia, 2007). More in details, the FPMs describe the physical background of phenomena, calculating the value of interest thanks to these equations. This approach does not consider any disturbances of the ideal conditions, and it works only considering the ideal steady-state conditions of the model. Due to this motivation, other kind of soft sensor devoted to the data driven models, gained on popularity. The data-driven models are fully based on the data recorded during the phenomena and they are also called black-box models.

For a data-driven approach, different predictive techniques are available ranging from statistical methods such as Principle Component Regression or Support Vector Machine (Zhang, Yin, & Wang, 2017) to soft computing methods like Artificial Neural Networks (Capriglione D. , Carratù, Pietrosanto, & Sommella, 2017) or Neuro-Fuzzy Systems.

Theoretically, no explicit knowledge is needed for the development and management of a soft sensor based on this type of models, however, in some practical scenarios there is an undeniable amount of knowledge required to achieve adequate performance. This knowledge is usually applied to data pre-processing and model selection. This kind of soft sensing approaches are also named grey-box models and it is located between the phenomenological models and the data-driven approaches. Wide applications can be accomplished by soft sensors. One of the dominant application areas of soft sensors is the online prediction of a quantities which cannot be measured directly from the sensors available in the system. For these reasons it is of huge interest to obtain prediction about these quantities at higher sampling rate and/or at lower financial burden, which is exactly the role of the soft sensors. This kind of soft sensor application field is commonly named On-line Prediction. Also, in Process Monitoring and Instrument Fault Detection (IFD) techniques, the use of soft sensors is important. These tasks mean to the detection of abnormal conditions of the observed variable and identification of the deviation source. The need of IFD schemes for assuring the correct operation of devices devoted to passenger comfort and safety is today urgent also in the motorcycle context.

Soft sensors could represent a good solution successfully applied to solve different problems varying from an implementation of a diagnostic algorithm for MOSFET fault detection and identification in a DC-AC converter Class-E2 (Catelani, et al., 2018) to the evaluation of elongation speed in a

motorcycle electro-hydraulic suspension using a couple of accelerometers (Delvecchio, Spelta, Perico, & Savaresi, 2010).

The development of a soft sensor in the motorcycle suspension systems field, may be useful for multipurpose, first of all, the inferential model intended to reduce the measuring hardware requirements may result into a significant source of budget saving and increasing the control loop reliability, indeed, the fault probability is strongly influenced by the amount of the devices operating in the harsh environment. Probably, the main application of the soft sensor is the Sensor Validation, a particular kind of fault detection, in which the system to be monitored is a sensor or a set of sensors, following the physical/analytical redundancy-based approach typically exploited in the automotive safety.

In order to estimate vertical dynamic quantities of a motorcycle, many authors have proposed different schemes with different aims, all based on the main hypothesis of the Half-Car model (Spelta, Delvecchio, & Savaresi, 2010). Although this model is simple and easily used, it does not allow the steering and the linkage nonlinear effects to be correctly estimated in terms of the corresponding varying wheel base and transfer load.

Following the black and gray box approach, in this thesis two types of soft sensors regarding motorcycle suspension system will be presented taking into account the nonlinear effects previously presented: a first one, useful for the reconstruction of the rear suspension stroke uses a Nonlinear Auto-Regressive with eXogenous inputs (NARX) network and a second one, based on accelerometers able to estimate the front suspension stroke velocity through different methods.

An IFD scheme able to detect faults occurring on the rear stroke sensor in a semi-active suspension system, will be also presented. The IFD is done by comparing the actual sensor output with the expected one provided by the soft sensor developed.

The IFD scheme will be characterized in order to evaluate the promptness and reliability in detecting faults introduced by open/short circuit, mechanical locking (hold faults) and uncalibrated sensors (due to slight variations of the input/output sensor curve).

On the other hand, such application's scenario is very challenging for both real time execution and on-board effective implementation of such schemes. Indeed, hard constraints are imposed on hardware/software solutions to be used mainly in terms of costs, compactness, weights, current consumption, electromagnetic interference issues, even if smartness and reactivity are demanded by control systems. For these reasons, the Thesis ends with the implementation on a low-cost general-purpose ARM-based microcontroller of the developed IFD scheme for the rear stroke sensor.

Chapter I

The motorcycle suspension systems

Today all vehicles are equipped with a suspension system responsible to ensure the contact between tires and road, and at the same time isolates the vehicle frame from the road roughness (Cossalter V. , 2002).

For economical reason and minor complexity, most of the commercial vehicles are equipped with passive suspensions; they are composed of two parallel elements: a shock absorber (K_s spring) and a viscous damper (C). The parameters about the passive suspension are fixed and chosen by the manufacturer in order to obtain a compromise value between the comfort and the handling of the vehicle. Generally, a “soft” setup aims to improve comfort, absorbing (and thus compensating) the roughness of the road but reducing the road holding due to the wide vertical oscillations of the vehicle and the consequent large fluctuations in the contact force between tire and road. On the other hand, a “hard” setup aims to guarantee better adherence reducing the passenger comfort due to an increase of the vertical stresses on the vehicle body (Gobbi & Mastinu, 2001) .

While improving the ride comfort is achievable with minimizing the vertical mass accelerations, the handling (sometimes referred to as "buoyancy control") can be maximized with the minimization of the fluctuations of contact force between tire and road, keeping the height of the sprung masses with respect to the ground constant, against forces of various kinds (aerodynamics, load transfers, etc.) that act on the body during braking, accelerating and cornering actions. It would be desired, that the suspensions were simultaneously “soft” towards the roughness of the road and “rigid” towards external inertial forces, but it seems clear that there are two conflicting specifications. A good behavior could be achieved by the adoption of a compromise among the two specifications. In motorsport competitions, cars have very stiff suspension, offering performance in terms of trim and road holding, but useless in everyday life. The performances reachable with passive suspensions are limited by the simplicity of the actuation devices and by the

limited number of freedom degrees. In order to obtain both the performances, a passive suspension system could be replaced with a semi-active or active suspension system (Sun & Yang, 2009). Rarely they are used on regular vehicle, due to the not easy control strategy and not negligible cost.

The semi-active suspensions systems are composed, in the same way of the passive suspensions, by the two classical elements, but with the difference that there is an actuator and a related control system able to suitably change the damping constant coefficients of the damper.

In the active suspension systems, in addition to the spring and the damper, there is a third element named actuator, able to generate an internal time-variable force $F(t)$ between the vehicle chassis and the wheel.

These systems allow, through a right regulation of the impressed force, to stabilize the movement of the sprung masses and obtain higher performances with respect to passive suspension systems. The regulation in both the semi-actives and actives suspension systems is based on the measures regarding the vertical dynamic of a motorcycle and a suitable control strategy (Hrovat, 1997).

I.1 Physical model of the suspension systems

In this section various physical models for passive, active and semi-active suspension systems have been reported. All the models presented belong to the class of linear systems, characterized by a low complexity usually required for the design of suitable control system. More complex and accurate models, generally strongly non-linear, can be developed and are used in practice as an off-line analysis and verification tool. The determination of parameter values is extremely problematic in complex models and, in the most cases, the calibration is carried out with data-driven approaches.

Considering the class of linear systems, there are different categories of mathematical models and the choice depends on the purpose (analysis, synthesis) and on the information needed from it. As well we know, a vehicle is composed by a huge and complex system with a very high number of interacting components. (see Figure I.1).

The physical models are divided into three main categories:

- "Quarter-Car" models
- "Half-Car" models
- "Full-Car" models

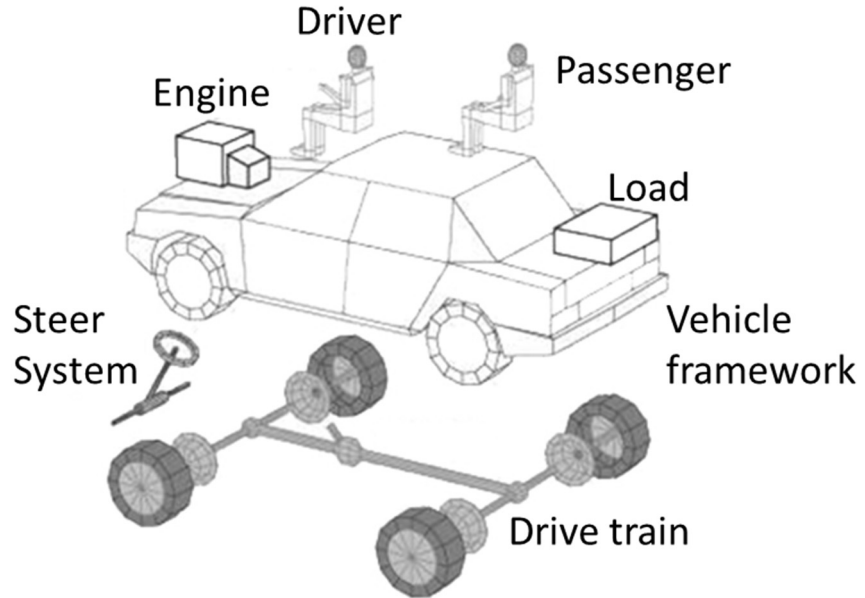


Figure I.1 *Main blocks of a vehicle*

The quarter-car model describes the vertical dynamics of a quarter of the entire vehicle, concentrating the analysis on a single wheel and on the relative suspension system. The vehicle is essentially divided into four sections that are modeled separately, neglecting mutual interactions. By this way, it is possible to study only the vertical dynamic (heave) without any possibility to characterize other dynamics as the roll, pitch, and yaw (see Figure I.2) (Yao & Zheng, 2006). In Figure I.3 a dynamic model representative of the quarter-car scheme of a passive suspension system is depicted.

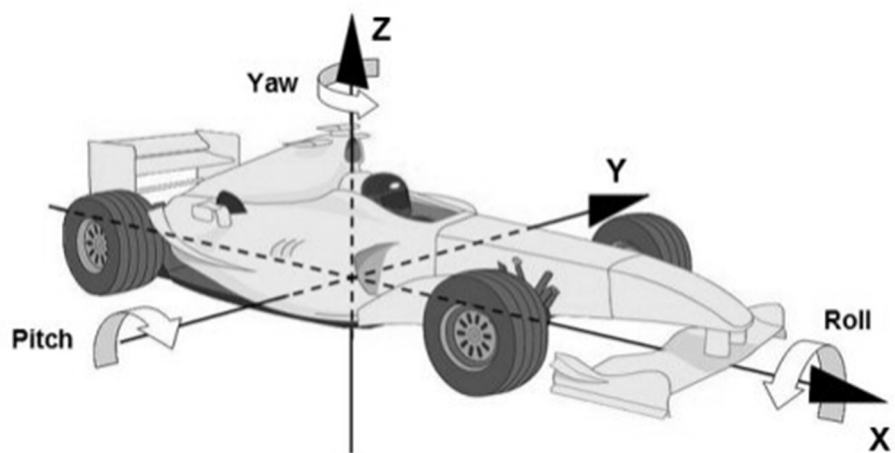


Figure I.2 *Main motions of a vehicle*

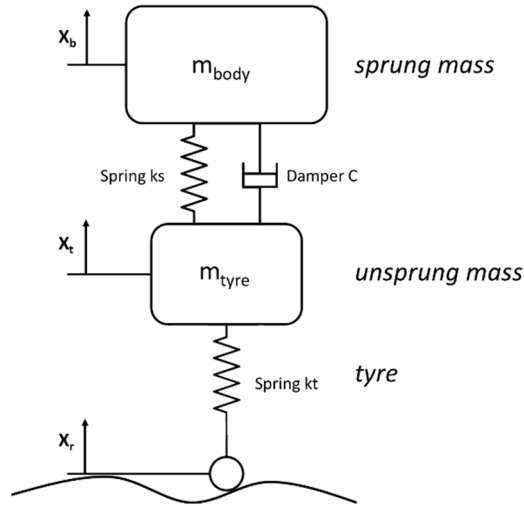


Figure I.3 *The quarter-car model*

In the half-car model, the car is seen "sideways" (see Figure I 4). The front wheel and the rear wheel, with the relative suspensions, are modeled in a coupled manner. This model allows to represent the pitch motions in addition to the vertical dynamic of the front and rear parts of the vehicle.

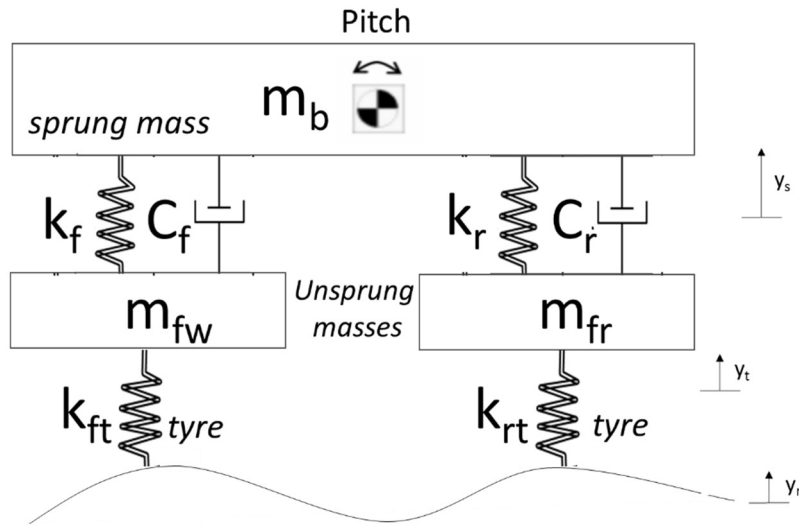


Figure I.4 *The half-car model*

In the full-car model the entire vehicle is considered. All possible vehicle dynamics, including roll and yaw, can be analyzed. The body is represented as a non-deformable parallelepiped with 6 degrees of freedom. (Figure I.5).

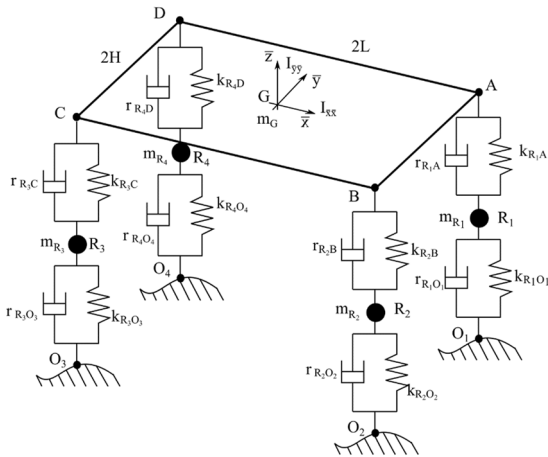


Figure I.5 *The full-car model*

Considering the quarter-car model, where the single suspension is analyzed, it is possible to find different separated elements:

- Elastic part;
- Damper;
- Mechanism that regulates their movements.

Also, the tires exhibit a damper behavior, with respect to the vibrations characterized by a high frequency and low amplitude, while the suspensions contribute to the damping oscillations characterized by a lower frequency and wider amplitude as depicted in Figure I.6 (Liguori, Paciello, Paolillo, Pietrosanto, & Sommella, 2013).

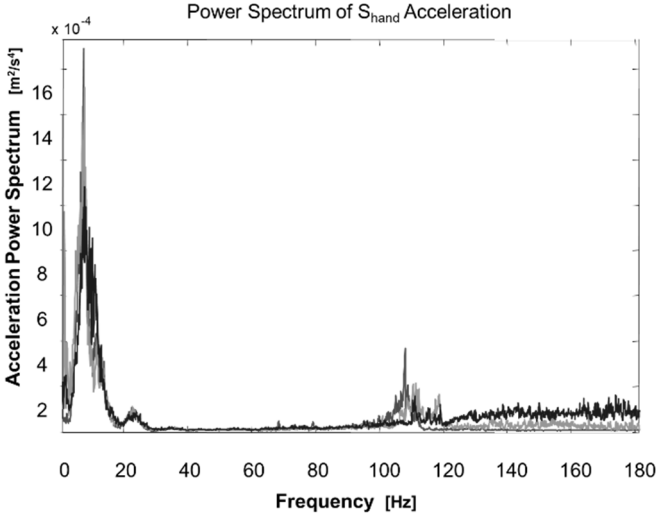


Figure I.6 *Acceleration Power Spectrum of the motorcycle vibrations*

About the motorcycle context, the unsprung masses are typically composed by all those things connected to the ground, while the sprung masses are all the elements placed above the suspension and rigidly fixed between them.

1.1.1 The spring

One of the elements that composes the suspension is the elastic part, also named spring (see Figure I.7). Its function is to store the energy coming from the impact against an asperity of the road surface.

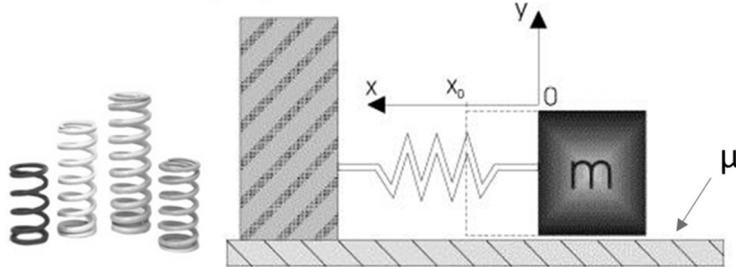


Figure I.7 The spring

Considering Figure I.7 and assuming no friction between the mass and the plane, the equation that regulates the movements of the mass m is:

$$F = kx \quad (\text{I.1})$$

where k is the stiffness constant of the spring and x is the stroke of m . Putting the mass m in a position different from the equilibrium, the spring will be charged with potential energy given by the following equation:

$$E_p = \frac{kx^2}{2} \quad (\text{I.2})$$

When the mass m is left free, it will tend to return to the equilibrium position; during the movement, the potential energy is transformed into kinetic energy whose expression is

$$E_c = \frac{mv^2}{2} \quad (\text{I.3})$$

where v is the speed of the mass.

Returning to the original position, all the potential energy has been transformed into kinetic energy and therefore the system is not yet in equilibrium due to inertia that will continue move the mass.

This process, under ideal conditions, would go indefinitely following the following law.

$$E_c + E_p = \frac{mv^2}{2} + \frac{kx^2}{2} = \text{cost} \quad (\text{I.4})$$

This phenomenon reported in Figure I.8 is an example of the "simple harmonic motion" whose descriptive law is:

$$x = x_0 \sin(\omega t) \quad (I.5)$$

in which x_0 is the amplitude of the oscillation and ω is the natural pulsation of the system that is related to the elastic constant of the spring and to the mass.

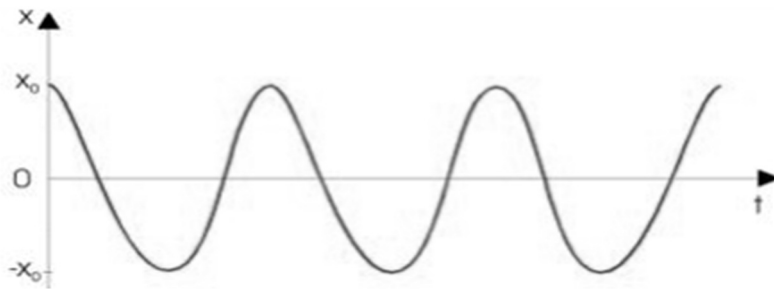


Figure I.8 *Simple harmonic motion*

If the spring is stressed with a repeated force overtime with pulsation Ω , there would be oscillations of different amplitudes. When the pulsation Ω matches with the natural pulsation ω of the system, critical conditions are reached: a positive trend of the oscillations amplitude leads to break the system. For the reason previously mentioned, a suspension composed by only the elastic element cannot be used.

Therefore, an element able to reduce the number of oscillations in the slight time as possible, must be introduced. This aim is typically assigned to the damper.

I.1.2 The damper

The main function of the damper is to dissipate the energy transmitted from the road surface, in order to reduce the oscillation of the elastic element. In Figure I.9 the mass-spring system with the addition of the damper is shown.

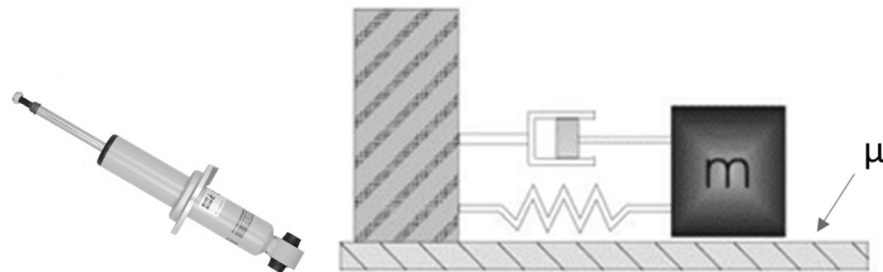


Figure I.9 *The damper*

The force impressed by the damper is proportional to the stroke velocity of the mass m according to the law:

$$F = cv = c\dot{x} \tag{I.6}$$

where F is the damping force, c is the damping constant and \dot{x} is the velocity of the mass m .

The introduction of the damper in the system, is able to transform the "simple harmonic motion" into a "damped harmonic motion", that is an oscillating motion in which the amplitude of the oscillations tends to decrease over time as shown in Figure I.10.

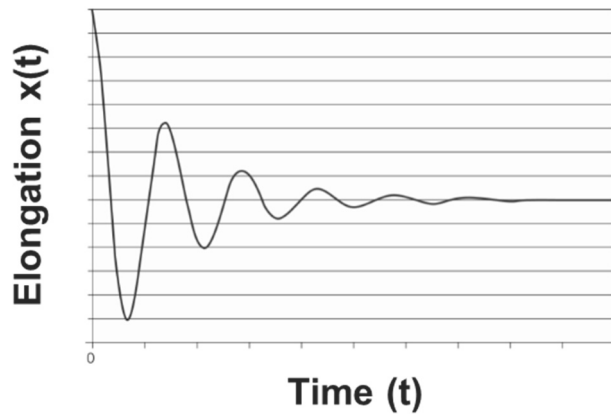


Figure I.10 *Muted harmonic motion*

The ideal condition would be one in which, the system stops after a single oscillation: this target can be achieved with a precise value of c , named critical damping (see Figure I.11).

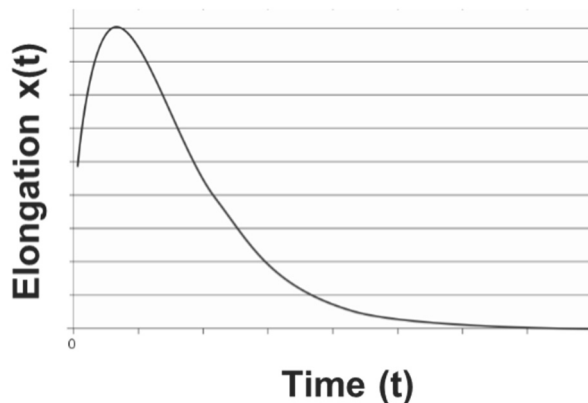


Figure I.11 *Damped harmonic motion in critical damping condition*

I.2 Passive suspension systems

Conventional or passive suspension systems (see Figure I.12) are composed by a coil spring steel and a fluid damper (oil or gas); the former has the task of temporarily accumulate energy while the latter to dissipates it.



Figure I.12 *Passive suspension*

The "force-displacement" or "force-speed" characteristics of these types of suspensions are not adjustable by the user, so once fixed in the design phase, they remain unchangeable (spring constant k_s and damper value C , see Figure I.13). For this reason, a trade-off between a rigid suspension which ensures greater road holding but paying in terms of comfort, and a soft suspension that reduces the vibrations coming from the road surface but reducing tightness especially in turns, must be selected.

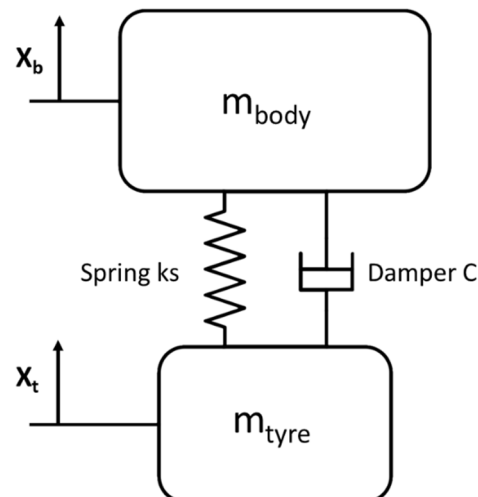


Figure I.13 *Model of a passive suspension*

It is important to specify that the suspension is never stressed by traction, indeed, the springs are mounted in order to be always more or less compressed since it is desired that the suspension does not work due to small stresses (typically managed by the tire). This initial compression, called preload, allows a degree of freedom in the vehicle setting: adjusting the preload, it is possible to choose between a harder or softer behavior. Note that a preload adjustment does not change the spring stiffness constant, therefore, with this adjustment, can be set the threshold after which the suspension starts to work. The stiffness k has been assumed constant, obtaining a relationship between load and linear displacement, however the springs for the suspensions are often fabricated in order to have an increasing k with the growth of the load. The preload is also important for setting the static height of the vehicle from the ground.

I.3 Active suspension systems

In the active suspensions, in addition to the spring and the damper, there is also an actuator able to generate a force $F(t)$ variables over time among the sprung and the unsprung masses (see Figure I.14).

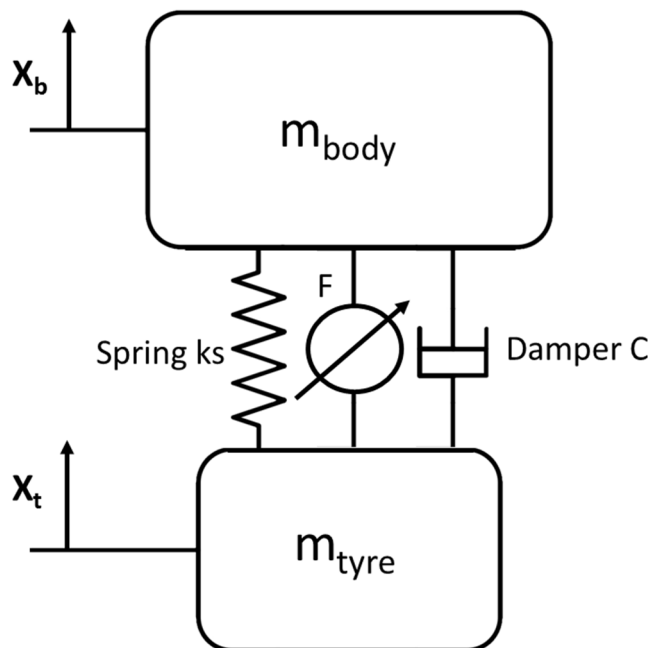


Figure I.14 Model of an active suspension

An active suspension needs several sensors to measure the dynamic of the vehicle, in order to give to the control system information to establish the

intensity of the force $F(t)$ that must be applied to obtain the desired damping. This “modulation” of the damping force makes the performance of the active suspension better than the passive suspension, although, a perfectly accurate and performant actuator is required. Therefore, an active suspension is able to dissipate energy and inject it into the system (hence the name of "active suspension") through the actuator; the benefits in terms of performance are considerable, but there are some negative aspects:

- Need of an electro-mechanical system that drives the actuators, with the relative accessories (collecting tanks, servo valves, amplifiers, etc.) (Audino, 2007).
- Need of an actuator able to provide to the sprung mass the "calculated" force.
- Need of sensors (accelerometers, speed sensors, stroke sensors, etc.)
- Need of hardware and software for the suspensions system control

It follows an increase of the vehicle weight, larger consumption (also due to the energy need by the actuator system), higher cost and lower reliability due to the large number of components. Theoretically, the performance of an active suspensions system is limited only by the necessary power. As an example, it can be assumed that an active suspension system for cars needs an increase in power between 5 and 15 kW for each wheel and an increase in weight between 20 and 30 kg for each wheel (Guglielmino, Sireteanu, Stammers, Ghita, & Giuclea, 2008). A semi-active suspension system could be a compromise solution between the simplicity and low cost of a passive suspension system and the high performance with high cost of an active suspension system.

I.4 Semi-active suspension systems

Semi-active suspension system represents the best compromise among costs and performance, achievable from a suspension. The controllability of these systems come from the presence of a damper capable of modifying their damping characteristic in real-time. The damping change is possible modifying the opening of the valves, through which the oil flows, or changing the physical properties of the latter (viscosity). For this reason, these types of suspensions are named "semi-active", i.e. they do not require active power for the control.

An active damper in parallel with the conventional spring, allows a controlled energy dissipation in real-time (see Figure I.15).

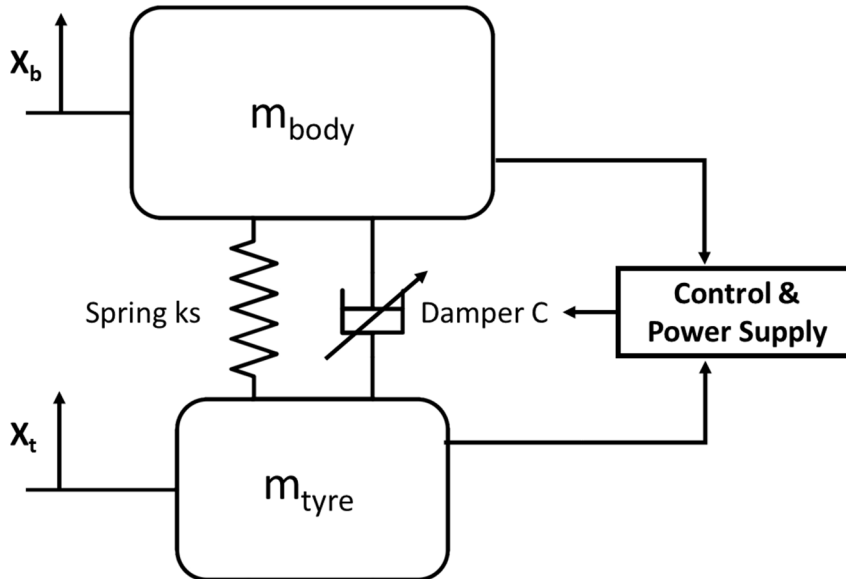


Figure I.15 *Semi-active suspension scheme*

The main advantage of this type of suspension system is the possibility to change the damping without the presence of an actuator, indeed, the active damper is often made by a mechanical, electrical, or magnetical system that reacts to a feedback control system.

It is possible to differentiate three types of semi-active suspensions, according to the type of phenomenon that regulates the variation of the damping coefficient (Acocella, Anchini, Paciello, Pietrosanto, & Sommella, 2010) :

- Continuously Damping Control suspensions (CDC).
- Electrorheological suspensions (ER).
- Magnetorheological suspensions (MR).

Continuously Damping Control dampers (CDC)

The CDC damper is based on the variation of the orifices size, connecting the upper and lower chamber of the damper piston.

When the suspension is excited by a road bump, the oil is forced to pass through the holes in the piston. The variation of the orifices section leads with an alteration of the viscous friction and a consequent change of the suspension speed. In this way, a variable stiffness of the suspension is obtained. A section of a CDC damper is shown in Figure I.16, where it is possible to notice the mechanism responsible the orifices variation. The response time for a CDC

suspension is approximately 30-40ms. The mechanism that acts on the orifices is driven by a solenoid valves, that require currents ranging from 0 to 1.5 A. The power required by this type of suspension is estimated at around 10 W.

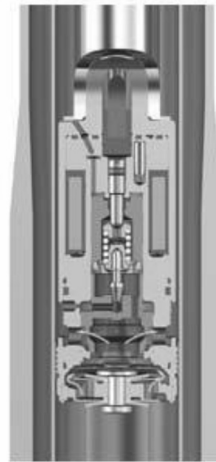


Figure I.16 *CDC damper*

Electrorheological suspensions (ER).

This type of suspension is characterized by the use of an electrorheological fluids, categorized as colloidal fluids able to change their viscosity with the application of an electric field perpendicular to the flow direction. The low cost and the fast response time make ER systems very interesting in the field of semi-active suspensions.

Under the influence of a suitable electric field, the fluid changes its viscosity from fluid to almost solid; this phenomenon is possible thanks to the presence of polarizable particles immersed in a non-conducting oil. With respect to the CDC suspension, the electrorheological fluid can change its characteristics in above 15ms. This type of technology results to be very functional, however there are some modeling problems with the electrorheological fluid.

Magnetorheological suspensions (MR)

The main characteristic of a magnetorheological suspension is the particular fluid present inside it. This type of fluid is composed by an oil with rheological properties that can be altered by applying a magnetic field: a variable damping can be done changing this field. (Paciello & Sommella, Smart sensing and smart material for smart automotive damping, 2013)

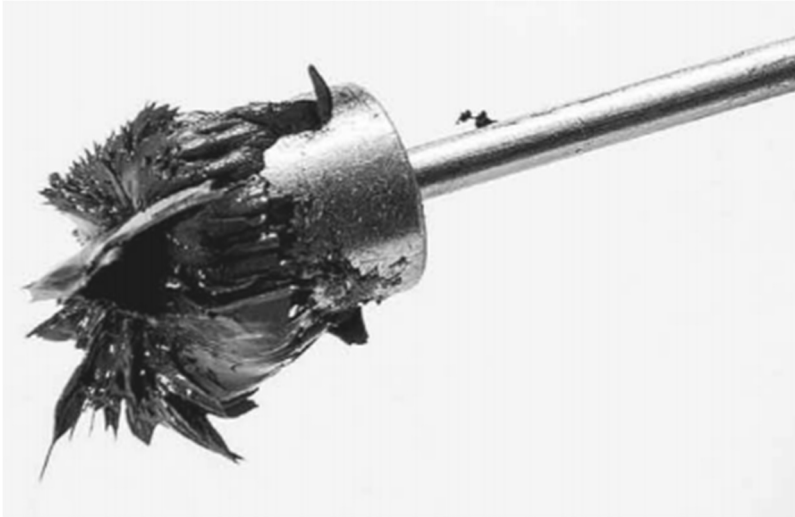


Figure I.17 *The magnetorheological fluid*

Magnetorheological fluids belong to the same family of electrorheological fluids and consist in a normal oil, enhanced with ferromagnetic microscopic (a few micron) particles, able to modify the internal viscosity reversibly and suddenly proportionally to the applied field. Without the application of a magnetic field, the ferromagnetic particles do not modify the classical newtonian behavior of the fluid; by the application of a magnetic field, the ferromagnetic particles arrange themselves in an ordered manner forming chains perpendicular to the direction of the fluid flow, opposing it and producing a behavior of a semi-solid fluid, able to change the stiffness of the suspension. With respect to the other types of suspensions, the magnetorheological fluid can change its characteristics in above 30 ns. The magnetorheological suspension is composed by a cylinder containing a floating piston, which separates the area with the magnetorheological fluid by the area with a gas.

The electromagnetic coil located inside the piston is able to generate the magnetic field by applying an electric current ranging from 0.1 A to 2 A. Figure I.18 shows a double tube shock absorber with a blind piston (without holes) and an electromagnetic coil positioned in the upper part of the figure. The coil is integral with the two cylinders and when the magnetic field varies, then limits the passage of the magnetorheological fluid from an external chamber to the internal one (extension) and vice versa (compression). The gas present in this suspension is necessary, as well as in all suspensions, to compensate the volume variation due to the movement of the piston.

Although both electrorheological and magnetorheological fluids are potentially excellent tools for a dynamic regulation of the stiffness in a suspension system, in recent years studies have focused on MR fluids since

they have better electro-physical properties (Paciello & Pietrosanto, *Magnetorheological Dampers: A New Approach of Characterization*, 2011).

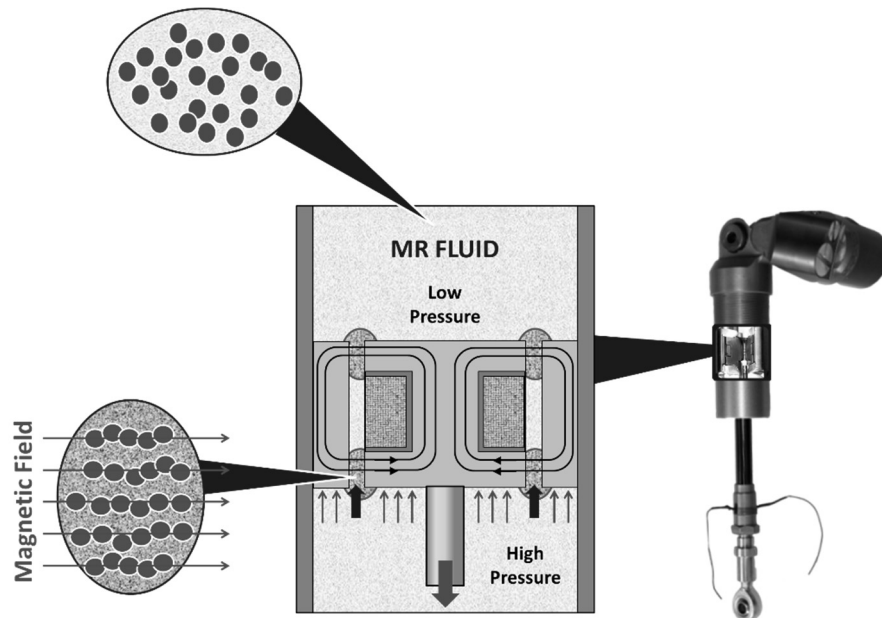


Figure I.18 *Magnetorheological suspension*

I.5 A real case: The MAGNETO system

During the Ph.D. , in order to the develop, test, and evaluate different soft-sensors and an IFD scheme, a semi-active suspension system named *Magneto System* was used. *Magneto* was developed by SPRING OFF s.r.l., a spinoff of the University of Salerno in collaboration with MUPO s.r.l.

The *Magneto System* was born from an idea of Gerardo Acocella, track engineer in the Superstock 1000 World Championship (2007-2009 with the Ducati Junior Team, 2010-2015 with the BMW Motorrad Italy Team, from 2015 with Honda SBK).

The experience in the field of electronics and digital signal processing, developed by the measurement laboratory of the University of Salerno, together with the experience gained in international motorcycle competitions (WSBK), has allowed the development of an electronically controlled suspension system for motorcycles able to adapt both to the road profile and to the rider's driving style (European Patent No. 2250038, 2012) .

The *Magneto system* is showed in Figure I.19 and it is principally composed by:

- Electronic fork and Mono-shock (based on magnetological fluid)
- Control unit
- Linear potentiometers

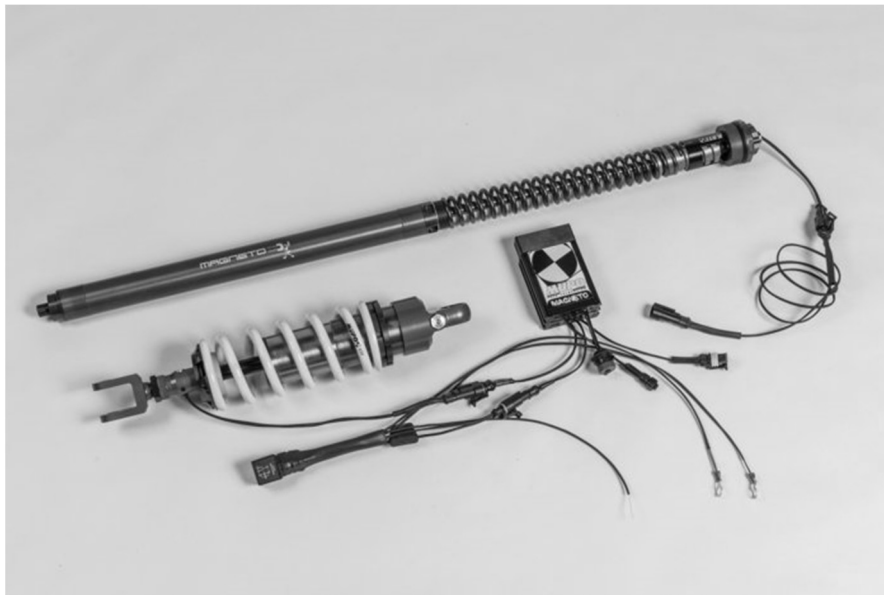


Figure I.19 *The Magneto kit*

The innovative *Magneto system* is able to modify in real time the response of the suspensions using the measurement information related to the vertical dynamics of the vehicle.

The control unit represents the technological solution, able to regulate in real time the damping characteristics of both the magnetorheological fork and monoshock. The control unit, thanks to the data achievable by suitable sensors, controls the suspension system ensuring the instant adaptation to the road profile and to the dynamic conditions of the bike (acceleration, braking, curves), in order to ensure the best set-up allowed by the pilot driving style in all drive conditions.

Actually, the system developed by Spring Off is available as an aftermarket product for all enthusiasts who want the highest level of performance from the suspensions while respecting maximum safety. Furthermore, the system is designed to allow the continuous updating of the strategy, keeping the safety characteristics in the guide and reliability of the product unchanged.

1.5.1 Description of the MAGNETO control system

The suspension control system is composed by several hardware elements. The main component is represented by a microcontroller located in the electronic control unit (ECU) that deals with: management and acquisition of the signals coming from a set of sensors and the CAN-BUS, the elaboration of the control strategy, and the generation of the signals for the actuator.

There is also a data logger suitable for sniffing the traffic over CAN-BUS, able to store all the data exchanged among the sensors in the motorcycle system. The presence of a data-logger is fundamental for the storage of data necessary for the post-analysis.

Only the pushed integration of all these tools, makes it possible to obtain a performing, innovative, and at the same time robust and reliable system.

Electronic Control Unit (ECU)

The control unit of the *Magneto system* (see Figure I.20) is a microcontroller-based system responsible of the digital signal processing functionality. The firmware manages:

- The acquisition from the sensors;
- The management of the signals present on CAN bus;
- The selection of the setup parameter;
- The execution of the control strategy;
- The generation of the control signal able to drive the dampers



Figure I.20 *ECU of the Magneto system*

More in details, about the acquisition of the analog signals, each sensor is connected to an analog input of the microcontroller which use a sampling frequency of 1 kHz and a quantization with a variable number of bits, while the digital sensors are directly connected to the digital bus available on the microcontroller.

The control unit is able to load the configuration setup into the control strategy firmware, which contain information on:

- the acquisition channels (if analog, is possible to set the number of bits and the sampling frequency), the type of coding used, and the CAN channel on which this acquisition will be identified;
- the calibration coefficients of the sensors and the relative units of measurement;
- the characteristic parameters of the control strategy

The processing of the control strategy consists in a suitably mixing of the signals coming from sensors in order to obtain an output value, which represents the type of damping desired.

Data logger

The MDLog (Motorbike Datalogger) is a professional microcontroller-based unit, able to sniff all the traffic present on the CAN bus network and acquiring both digital and analog signals. The main characteristics of the MDLog are:

- 8 GB of memory
- 4 analog channels
- 4 digital channels
- CAN-BUS port
- Sampling frequency up to 1 kHz.

The MDLog is equipped with an ad-hoc software for managing and analyzing data. It is possible to configure which signals must be stored, the resolution (number of bits), the type of encoding (range and unit of measure) and the sampling rate.

Once the acquisition parameters have been programmed, data capture starts when the control unit is switched on.

A suitable software allows to download the contents of the memory and organizes the data according to the acquisition settings; it can also mask and reorder data in a database and export them in different formats.

I.5.2 Description of the MAGNETO damping system

The *magneto system* uses the magnetorheological dampers as depicted in Figure I.21 and previously described. The variation of the damping coefficient of the suspensions is obtained thanks to the application of a magnetic field. The magnetic field is applied through a solenoid immersed in the magnetorheological fluid.



Figure I.21 *The Magneto suspensions*

The electrical current is provided to the coils of the two dampers thanks to a power amplifier, which supply power to the system. The choice of the right current value to be applied, depends on the model that binds the damping force to the injected current and to the suspension stroke velocity. The control strategy instantly calculates the optimal value of the damping force, which is converted into the current to be injected through the amplifiers (see Figure I.22) (Acocella, Anchini, Paciello, Pietrosanto, & Sommella, 2010).

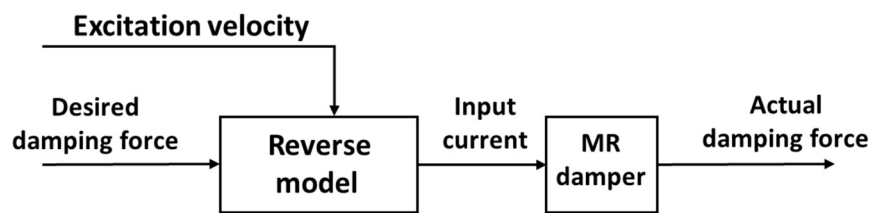


Figure I.22 *Block Diagram of the MAGNETO driver system*

Chapter II

Hard and soft sensors for the control of a semi-active suspension system

A semi-active suspension system is a typical example of an electronic subsystem in the motorcycle sector; it replaces the operation of a mechanical subsystem with an electromechanical equivalent that can be controlled.

To understand the basilar rules needed for the dynamic control of a semi-active suspensions system, in the following paragraphs a description of the control strategy and the sensors set involved are shown.

Searching the best behavior of the motorcycle attitude, different signals, and parameters regarding the status of the vertical dynamic must be monitored. To determine the useful information, it is important to highlight which and how many sensors are necessary for the acquisition of information. The model underlying the control rules with their relative used quantities will be shown in the next paragraph.

II.1 Physical dynamics in a motorcycle system

The motorcycle and the rider are subjected to a series of motions during the driving phase. More in details, during the steering, these motions allowing the curve to be made without falling. The motorcycle, indeed, differently from the car, presents only a dynamic balance without a static balance. The motorcycle stays in the vertical position thanks to the gyroscopic effect of the wheels and the corrections made by the pilot especially at low speed. The fundamental motions regarding the attitude of a motorcycle that allow the steering are mainly three: the pitch, the roll, and the yaw motions. These motions are generated in consequence of a phenomenon named load transfer. This phenomenon occurs during both the acceleration and deceleration phases of the motorcycle, since the braking and accelerating forces are generated at

ground level, therefore distant from the point where the center of gravity is located. So, during the phases of speed variation, moments are generated from the forces mentioned which tend to rotate the body of the motorcycle.

II.1.1 Center of gravity in a motorcycle body

The body center of gravity is defined as that point in which, all the weight of the object in question is acting. Clearly, it is an ideal concept for a motorcycle, since its weight is distributed in all the space occupied. However, it is an approximation particularly useful for obtaining good practical results. Being a three-dimensional body, is it possible to consider the motorcycle as a solid body and therefore consider, due to the good symmetry of the vehicle, the position of the center of gravity located approximately on the vertical longitudinal plane passing through the vehicle centerline. The two other coordinates of the center of gravity position, can be defined within the previously mentioned plane considering the height from the road surface h and the distance b from the projection, on the road surface, of the center of the rear wheel (see Figure II.1).

The center of gravity position influences significantly the dynamic behavior of the motorcycle, particularly in the acceleration and braking phases.

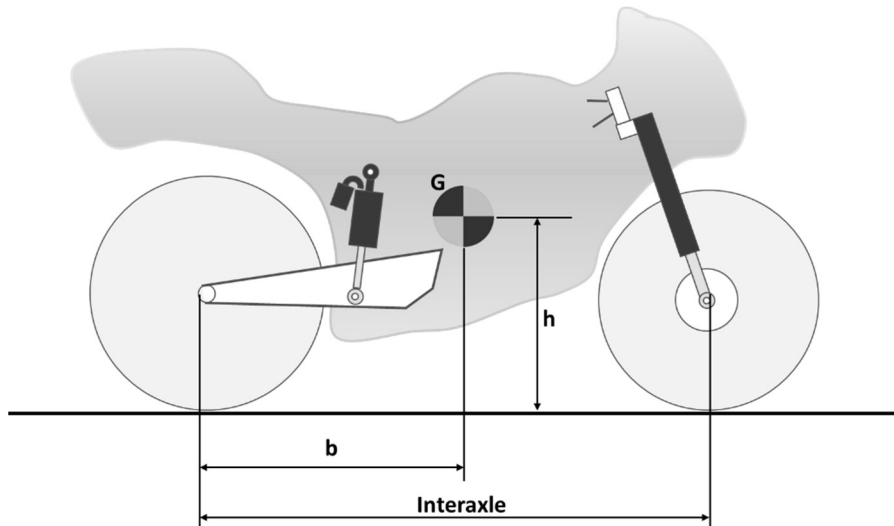


Figure II.1 *Coordinates of the motorcycle center of gravity*

II-1.2 The pitch motions

This type of motion consists in the longitudinal lowering of the motorcycle front part during braking and in its consequent lifting in the acceleration phase as it is possible to see in Figure II.2. Indeed, a rotation is generated around the transverse axis passing through the vehicle center of gravity. It is easy to understand how this behavior, although mediated by the actions of the suspensions, is caused by the load transfer from one axis to another as consequence of a motorcycle speed variation.

In addition to the load transfer phenomenon, the pitching is also caused by the steering: the rotation of the steering causes (keeping ideally the center of the wheel fixed) the detachment of the tire from the ground. Clearly the wheel will keep contact with the ground, causing a slight pitching movement.

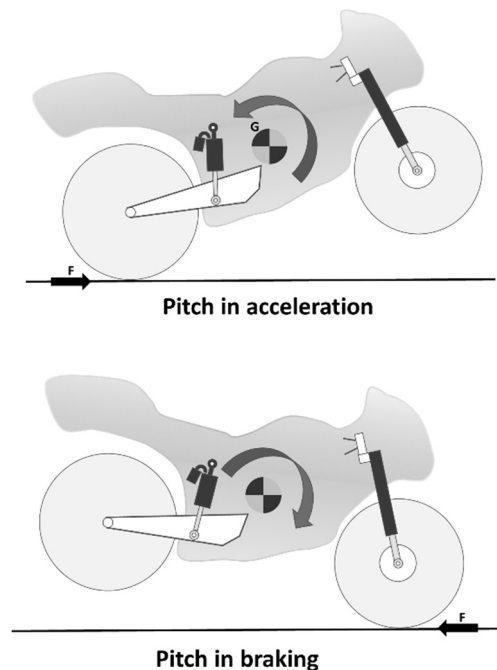


Figure II.2 *Motorcycle pitch motion*

II.1.3 The roll motions

The roll motion consists in the motorcycle rotation around the straight line connecting the contact points of the tires with the road surface (longitudinal axis) (see Figure II.3). The roll motion is fundamental because it allows to oppose at the centrifugal force when the motorcycle cornering. It is often

possible see in competitions, the behavior of the driver which tilts the motorcycle as much as possible, moving the body towards the center of the curve, with the intent of opposite the centrifugal force. Must be noticed that a roll motion cause also, a slight pitching motion.

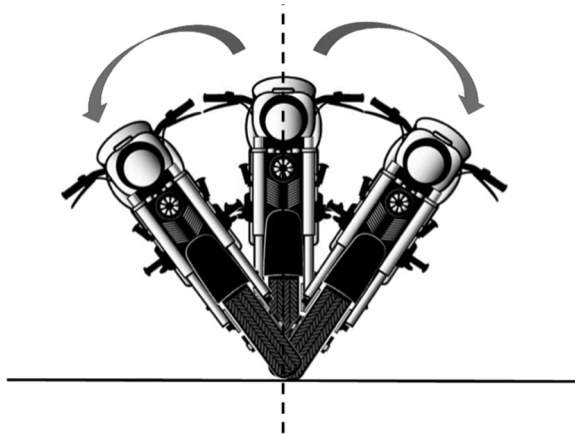


Figure II.3 *Motorcycle roll motion*

II.1.4 *The yaw motions*

The yaw motion determines the rotation of the motorcycle around its vertical barycentric axis during the steering phase (see Figure II.4). Also, this motion generates an inertial moment, caused by the disposition trough the longitudinal side of the vehicle of the masses, placed in a far position from the center of gravity.



Figure II.4 *Motorcycle yaw motion*

II.1.5 Role of suspensions

The suspensions have the function of absorbing the irregularities of the road surface, keeping the wheels in contact with the ground, minimizing the motorcycle vertical displacements and compensating the load variations due to the distribution of the "sprung" and the "unsprung" masses. The aim is to keep the motorcycle's attitude as constant as possible, i.e. the wheelbase measurements, the front chassis, and the position of the center of gravity in all conditions, in order to compensate all the motions of the motorcycle.

From the efficiency of the suspension system, therefore, depends the comfort and the quality of the road holding.

The suspensions, which elastically connect the wheels to the frame, must have, as reported in Chapter I, two separate but complementary functions: the springing and the damping ones. The former ensures the absorption of irregularities (and any load variation acting according to the plane having the center of gravity and the tire contact points to the ground). The latter dampens the movement in both directions (compression and extension) of the spring, avoiding rebounds and reducing oscillations; influence, in other words, the stroke velocity of the suspensions.

II.2 Pitch driven control strategy

As said previously, the suspension must keep the motorcycle's attitude constant, in any condition. Considering a motorcycle suspension, they can compensate the vertical movements of the suspended mass, varying the pitch angle.

Controlling the pitch angle value in a short period to the nominal value could be defined as the best-work condition, but the nominal value mentioned, however, is not constant and it depends on many factors as: the distribution of loads (one passenger, two passengers, one passenger with luggage and so on), the speed of the motorcycle, the acceleration, the slope of the road surface, etc. . Therefore, it is preferable to control the pitch angle derivative, i.e. the pitch (Liguori, Paciello, Paolillo, Pietrosanto, & Sommella, 2014). The task of the suspensions must behave to oppose the pitch motion.

A primitive control strategy could use as feedback the measure of the difference of the pitch from zero; the difference, being the error achieved by the system, can be evaluated in different ways, as:

- Maximum of the absolute instantaneous error
- Maximum of the relative instantaneous error
- Integral of the maximum error in a certain time interval

- Integral of the relative error in a certain time interval

The choice of the right criterion for the feedback calculation, depends significantly on the vehicle use context. In a city driving context, if the absolute maximum pitch error and the vertical velocity of the motorcycle are limited, the comfort is preferred; on the other hand, if it would be highlighted the sport performance, it is essential that the motorcycle keeps stable its attitude using as parameter for the control strategy, the integral of the relative error in a given time interval (Guglielmino, Sireteanu, Stammers, Ghita, & Giuclea, 2008) .

However, the pitch stabilization is not always enough; for example, if the aim is to get a good comfort, it is important to avoid rapid vertical variations of the front and rear side of the motorcycle, so the strategy based on the only correction of the pitch in these situations could be no longer correct (Cossalter, Doria, Garbin, & Lot, 2006). Therefore, it would be necessary recognize different events and manages it appropriately, considering for example the velocity stroke of the suspension as parameter to be used in the control strategy.

II.2.1 Typology of control scheme

The suspension system can be modelled as a dynamic Multiple Inputs Multiple Outputs (MIMO) system in which there are many inputs and outputs. Considering the easy pitch strategy previously presented, the system can be seen as a Single Input Single Output (SISO) model: the output variable to be controlled is the pitch for which a measure of this parameter is required; as input, a mechanical force for the active suspension, or the shock absorber damping coefficient for the semi-active suspension, must be considered.

The pitch control strategy can be achieved with a classic control scheme, as is well known from the theory of automatic controls. There are two principal schemes of control: open loop or closed loop; the closed loop is the widespread in the vast majority of systems because it has important advantages:

- Less sensitivity to the parametric variations of the system.
- It does not require a very deep knowledge of the system.
- Immunity to disturbances.
- Finer control
- Improving of the system stability.

A simple schematic of closed loop control is reported in Figure II.5

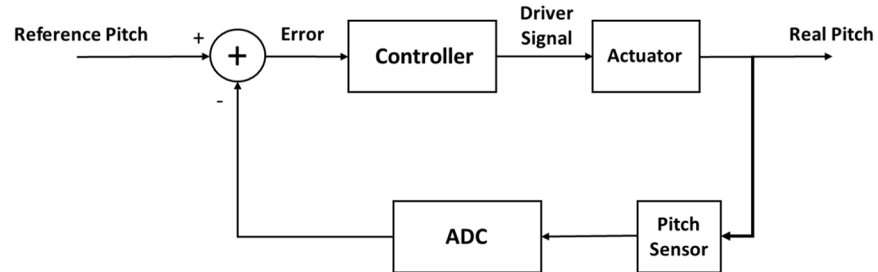


Figure II.5 Example of a closed loop strategy

Instead, in order to avoid large variations of the vertical suspension velocity, an open loop control that measures the front and rear vertical speeds and decides when bypass the pitch controller, forcing a softening of the suspension, could be used.

For any control scheme a set of quantities about the vertical dynamic of the motorcycle must be measured. The principal quantities are: pitch or pitch angle, front, and rear suspension stroke (necessary to calculate the stroke velocity of the suspension and reveal when the wheel rises from the ground). However, the roll signal, the yaw angle, the longitudinal speed of the motorcycle may be useful, to detect situations of poor balance of the vehicle and to make the grip of the motorcycle safer.

II.3 Set of sensors needed for the control strategy

A fundamental aspect to provide a good suspension control is the measure of the motorcycle dynamics. These measures take place through the use of several sensors (Marek, 2011), which follow the characteristic motions of a motorcycle. A sensor converts analog quantities into electrical signals (see Figure II.6).



Figure II.6 Sensing process

In particular, the pitch, the roll and the yaw motions are measured with a gyroscopic sensor, while stroke sensors are used to measure the vertical extensions of the front and rear suspended masses (Liguori, Paciello, Paolillo, Pietrosanto, & Sommella, 2015). Therefore, each specific sensor installed, provides the acquisition of each movements of interest. In the following a brief description of the sensors involved will be shown.

II.3.1 Stroke sensors

There are many devices that can be used to measure the stroke of an object. For example, a modulation can be used:

- The capacity between two armatures of a capacitor (capacitive sensors);
- The mutual inductance between two windings (inductive sensors);
- The partition factor of a potentiometer (resistive sensors)
- Hall voltage in a sensor coupled to a permanent magnet if the object is ferromagnetic (magnetic sensors);
- Light interception by a rotating or translating grating (optical sensors).

More in details, resistive potentiometer sensors are the most used to measure the vertical extensions and compressions of the front and rear suspensions. The potentiometer is the simplest electrical position sensor. A movable cursor is placed on a fixed resistor in this kind of device. The resistance of the circuit among any of the two terminals and the cursor, obviously depends on the position of the latter. It is clear that the potentiometer is a "modulating" sensor, since in order to obtain useful electrical information, it is necessary to put it into a circuit which supply an auxiliary power source. Using the Ohm's law to analyze the resistive potentiometer, the output voltage comes out from a fraction of the supply voltage; therefore, the mechanical action (the quantity to be measured) acts by modulating the supply voltage in output.

These sensors are generally used to obtain an electrical output from elastic elements used in mechanical devices and are constituted by a resistive element (cylinder), on which a moving contact (mobile part) is able, driven by the mechanical parts, to measure the motions (see Figure II.7).



Figure II.7 *Linear potentiometer*

In this way, thanks to the position of the slider led by the suspension stroke, the potentiometer will have different voltage level in output.

The characteristics, which influence the choice of a potentiometer are various, among which could be highlighted:

- The maximum applicable voltage;
- The electric stroke, which determines the maximum distance detectable;
- The stroke speed, which expresses how many meters the sensor is able to travel in a second;
- The precision referred to the repeatability of the measure;
- The output characteristic, which shows in percentage how much the output values follow a linearity law;
- The resolution, which indicates the smallest variation of values that the sensor is able to detect.

The measurement of the vertical stroke of a vehicle does not require a very high accuracy, while its important to have: mechanical and electrical strength, ease of use and low cost. Linear potentiometer sensors are easy to use, have a low cost and provide a high value of the output signal; however, they are affected by friction, inertial effects and wear which cause a premature aging.

II.3.2 Gyroscopes

Gyroscopes are used for angular velocity measurements. The gyroscope is a body (cylinder, disk, etc.) with rotation symmetry with respect to an axis (said gyroscopic axis), supported by a frame (suspension) and which can be putted in rapid rotation (see Figure II.8).

The gyroscope is therefore a rotating physical device which, due to the law of conservation of angular momentum, tends to keep its axis of rotation oriented in a fixed direction. A homogeneous cylindrical mass, which rapidly turns around its barycentric axis (stable axis of rotation), has centrifugal forces that develop perfectly balanced with each other. In this way, no one internal force can influence the rotary motion or generate vibrations changing the rotary motion. There are several types of gyroscopes: laser gyroscope, hemispherical resonator gyroscope, vibrating gyroscope.

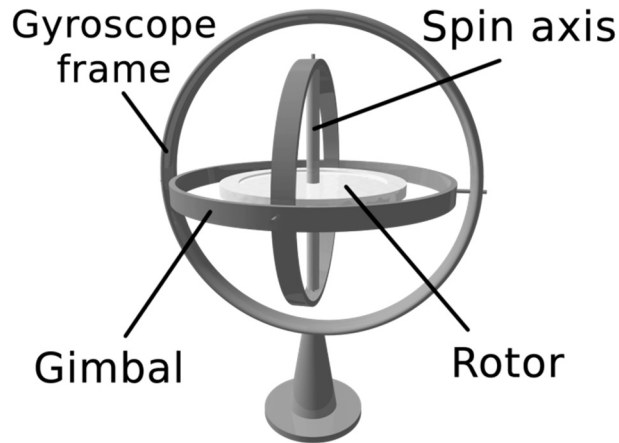


Figure II.8 *The gyroscope model*

Recently, most of the gyroscopes are developed in Micro Electro-Mechanical Systems (MEMS) technology. A MEMS gyroscope is composed by a vibrating system consisting of a suspended mass through an elastic system that allow it to move in the x, y, and z directions.

II.3.3 Accelerometers

An accelerometer is a measuring instrument capable of detecting and/or measuring acceleration, by performing the calculation of the force measured against the mass of the object.

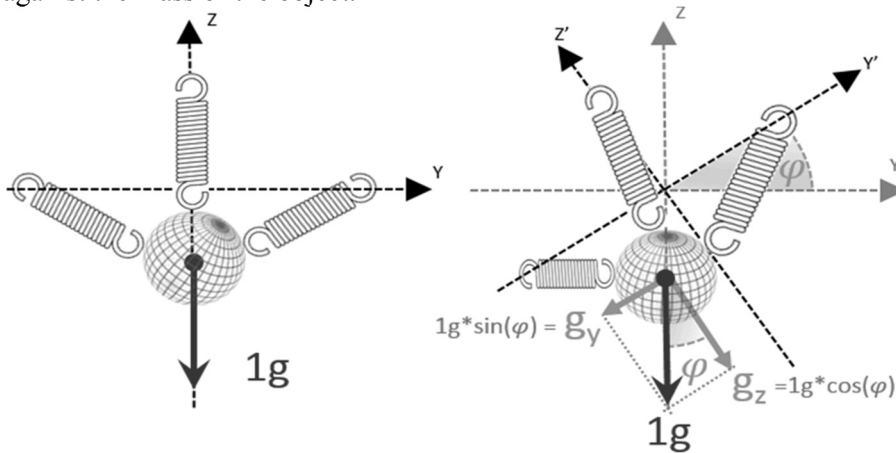


Figure II.9 *Operation principle of an accelerometer*

In most accelerometers, the principle of operation is the same: they are based on the detection of the inertia of a mass when it is subjected to an acceleration as shown in Figure II.9.

The mass is suspended from an elastic element, while some type of sensor detects its displacement from the fixed structure of the device. In the presence of an acceleration, the mass (which is equipped with its own inertia) moves from its resting position proportionally to the acceleration detected. The sensor transforms this displacement into an electrical signal.

II.3.4 Longitudinal speed sensors

The speed sensors are typically based on the Hall effect. They are integrated units having the transducer, magnet, and electronics that are used to provide the speed sensing. Usually the speed sensor uses a gear wheel added to the part in rotation (see Figure II.10).



Figure II.10 *Speed sensor*

For a motorcycle semi-active suspension system, typically can be needed up to five sensors.

The use of many sensors poses some problems:

- The acquisition system must acquire many input channels, thus increasing its complexity, processing time, and therefore reducing the control band;
- Each sensor is exposed to the possibility of failure; therefore, the reliability of the system is reduced;
- Each sensor has a cost, not only the unit cost of the devices, but also the cost of development both in terms of time and human resources;
- The sensors have a weight and a size that can be a not negligible

problem in a motorcycle.

On the other hand, having many sensors has the advantage of:

- Accurately monitor the condition of the motorcycle;
- Exploit analytical redundancy to perform diagnostics pursue on the system;

However, the set of sensors can be reduced if there are redundancy information among the quantities analyzed: a particular measure could be done without a direct measure but using other sensors in order to implement a “Soft Sensor” (Ruhm, 2007).

II.4 Soft sensors

More in details, “Soft Sensor” means the process of estimating any system or process variable by adopting mathematical models, replacing some physical devices, and using data acquired from some other available sensors. It is possible to differentiate two principal categories of soft sensors: Model-driven and Data-driven. The first typology of soft sensors is based on First Principle Models (FPM). More in details, the FPMs describe the physical background of phenomena, calculating the value of interest thanks to these equations. This approach does not consider any disturbances of the ideal conditions, and it works only considering the ideal conditions of the model without non-linearity. The data-driven models are fully based on the data recorded during the phenomena.

For a data-driven approach, different predictive techniques are available and range from statistical methods such as Principle Component Regression or Support Vector Machines to soft computing methods like Artificial Neural Networks or Neuro-Fuzzy Systems.

Soft sensors can carry out wide applications as in (Norgia, Boniolo, Tanelli, Savaresi, & Svelto, 2009) and (Liguori, Paciello, Pietrosanto, & Sommella, 2014). One of the dominant application areas of soft sensors is the online prediction of a quantities, which cannot be measured directly from the sensors available in the system.

The idea to adopt a soft sensor in a semi-active suspension system like the Magneto system described in Chapter I can gives different advantages as:

- Reduction of the system cost due to the minor number of cables and sensors.
- Improvement of the reliability of the system.

- Improvement of the safety through an implementation of an IFD scheme based on an analytical redundancy.

More in details, the attention has been focused on the first and third points: a reduction of the set of sensors used in the previously presented Magneto system and an IFD scheme for the rear linear potentiometer will be shown. The reduction of the number of sensors represent a significant economic saving, both for its own cost but also for the integration and installation costs. On the other hand, the reduction of the number of sensors tends to improve safety, since less sensors make the system more reliable, and since IFD scheme, on the critical sensors like the potentiometers, could be implemented thanks to the redundancy introduced (Capriglione, Carratù, Sommella, & Pietrosanto, 2018).

The model used for the study of motorcycle motion is the half-car explained in Chapter I; since we are interested in analyzing the pitch motion, whose analysis is satisfactory and quite simple.

This model is composed by the rigid suspended masses, which are connected, at the two ends with two spring-damper systems (spring and shock absorber) through two articulated joints. In sequence, they are attached to the two unsprung masses (wheels and all that is rigidly connected to them), which rest connected to the ground through a spring (tires).

An external stress causes a variation of the pitch, it may come from a different distribution of the suspended mass, due to an acceleration variation, or to an irregularity of the road surface. In both cases the suspensions and the tires contribute to tone down these variations. The most significant variations occur in the suspension, while the tire is slightly deformed. According to the model, the control strategy has to measure the pitch angle and the suspension motions in terms of stroke. Neglecting the presence of the tires, the model is simplified further; as reported in Figure II.11 and Figure II.12, the pitch angle is geometrically related to the vertical heights of the suspended mass in both the extremity.

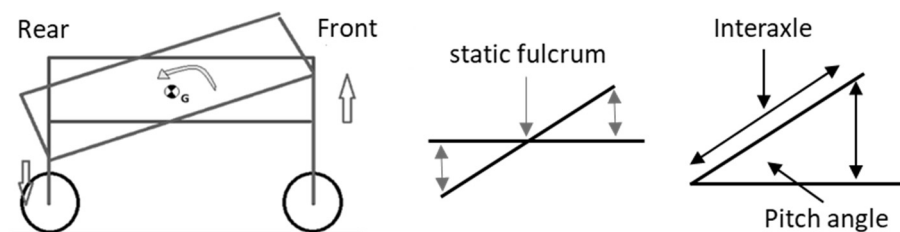


Figure II.11 Pitch motion during acceleration

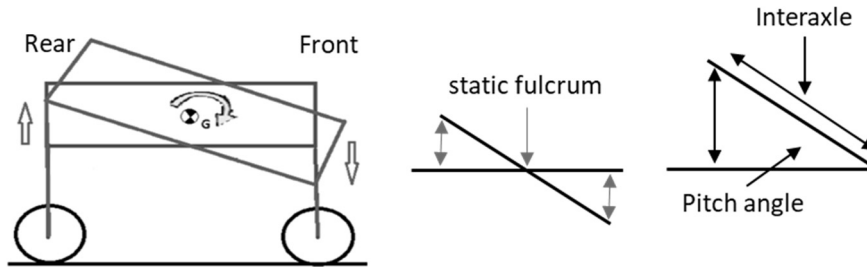


Figure II.12 Pitch motion during braking

Then measuring two of the three parameters available among pitch angle, front, and rear height, the third parameter can be obtained indirectly. The three signals are redundant among them: for example, measuring the two heights, the pitch angle is equal to:

$$\varphi = \sin^{-1} \left(\frac{P_{\text{rear}} - P_{\text{front}}}{\text{Interaxle}} \right) \quad (\text{II.1})$$

where φ is the pitch angle with positive reference in braking, P_{rear} is the stroke of the rear suspension, P_{front} is the stroke of the front suspension and interaxle is the distance between the centers of the two wheels. Reversing the eq. (II.1) it is possible to calculate:

$$P_{\text{front}} = P_{\text{rear}} - \text{Interaxle} * \sin \varphi \quad P_{\text{rear}} = \text{Interaxle} * \sin \varphi + \text{Ant}$$

$$\text{where } \varphi = \int \text{pitch dt.} \quad (\text{II.2})$$

It is also possible to calculate the pitch from the φ angle, making a derivative with eq. (II.3):

$$\text{pitch} = \frac{d\varphi}{dt} = \frac{d}{dt} \left[\sin^{-1} \left(\frac{P_{\text{rear}} - P_{\text{front}}}{\text{Interaxle}} \right) \right] \quad (\text{II.3})$$

Unfortunately, this model although simple, does not accurately replicate the physic of the motorcycle. In particular, the three sensors are not enough to accurately describe the vertical dynamics of the motorcycle, indeed:

- It is not possible to perfectly detect the vertical height of the suspended mass on both the front and rear extremities, because it also depends on the oscillations of the rubber that are not directly detectable; however, these variations are quite limited in amplitude and frequency.
- The pitch sensor is a gyroscope that detects the pitch angle variation with respect to the axis perpendicular to the weight force; if the road surface is rising or falling, the pitch detected will no longer refer to the road surface, but to the horizon axis, whereby

the geometric link of the three parameters is altered; also in this case this effect is rather limited in amplitude.

- The load variation due to the number of passengers or luggage change the interaxle distance.
- The interaxle distance is not constant and it depends on the leverage of the rear wheel and on load variations.

In order to exploit a more precise model of the motorcycle dynamic to develop a more accurate soft sensors, methods based on data driven approach could be adopted. In the following Chapters innovative modelling methods will be shown paying attention on the Neural Networks and Digital filtering techniques.

II.5 The test bed

It is now reported a description of the test bed used for the aim of this Thesis. A SUZUKI GSX-1000 motorcycle, equipped with an experimental set of sensors and a Magneto Kit described in Chapter I, has been used. The sensors have been located in different places on the motorcycle (see Figure II.13) in order to obtain information about the vertical dynamic of the vehicle. The sensors mounted on the motorcycle are reported in table II.1. To develop the Soft Sensors and an IFD scheme presented in the following Chapters, different measurement campaigns based on real data acquired on the field was performed. In each Chapters a detail of the particular motorcycle riding will be shown. The data logging achieved thanks to the MDLog presented in Chapter I, has been done with reference to the following signals: fork stroke, pitch rate, vehicle speed, rear shock stroke, front wheel vertical acceleration, rear wheel vertical acceleration and body acceleration. Data recording was carried out at the sampling frequency of 1 kHz.

Table II.1 *Sensors used on the test bed*

Sensor Type	Model	Manufacturer	Symbol
<i>Linear stroke sensor</i>	<i>SLS130</i>	<i>Penny & Giles</i>	<i>S_{front}</i>
			<i>S_{rear}</i>
<i>Magnetic encoder</i>	<i>970-011</i>	<i>Dorman</i>	<i>V_{bike}</i>
<i>Gyroscope</i>	<i>L3GD20</i>	<i>ST Microelectronics</i>	<i>Gyro</i>
<i>Accelerometer</i>	<i>LIS331H</i>	<i>ST Microelectronics</i>	<i>ACC_{front}</i>
			<i>ACC_{rear}</i>

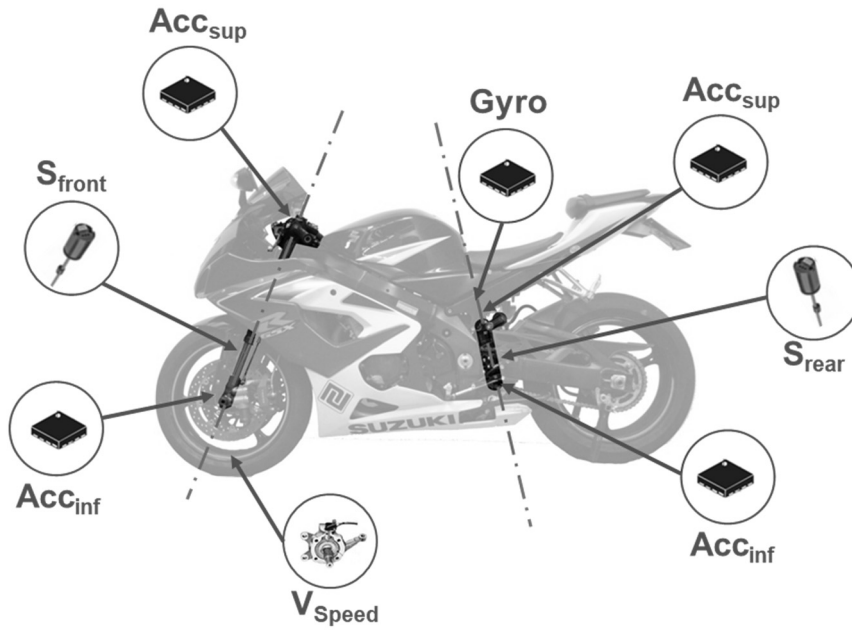


Figure II.13 *Test Bed*

Chapter III

Rear stroke soft sensor based on NARX network

This chapter describe the soft sensing of the rear suspension stroke through a suitable exploitation of the analytical redundancy existing among the vertical dynamics quantities of a two-wheeled vehicles. Indeed, as reported in Chapter II the behaviour of a motorcycle may be firstly modelled by a rigid system, where the rear suspension stroke, although greatly dependent from the road profile, also considers the heavy movement of the front suspension and the pitch of the vehicle frame.

More in details, a data-driven approach is applied to provide the mathematical model which allows the rear suspension stroke to be inferred from a set of influential variables during the motorcycle riding also described in Chapter II.

III.1 The rear stroke soft sensor

The aim is the selection of a model structure among different (candidate) representations, wherein a set of dependent variables, i.e. the system outputs are the consequence of a set of independent variables, i.e. the system inputs. Since the main hypothesis of the Half-Car Model reported in Chapter II (i.e. a steady state condition for the motorcycle dynamics) does not allow the steering and the linkage nonlinear effects to be correctly estimated in terms of the corresponding varying wheel base and transfer load, the Nonlinear Auto-Regressive with eXogenous inputs (NARX) model appeared as the most straightforward choice.

This kind of solution is generally more efficient than Nonlinear Moving Average (NMA) models; nevertheless, the corresponding prediction capability is limited to small steps since the error propagation of the system output through the closed loop. Then, the approach based on exogenous inputs was performed by considering observation intervals proportional to the system

dynamics in order to limit the model error propagation when a great number of predicted output samples is considered.

Following the research trend (Capriglione D. , Carratù, Pietrosanto, & Sommella, 2017), according to which NARX Artificial Neural Network (ANN) has emerged as an effective solution to the problems of both system prediction (estimation of the next value of the output signal) and nonlinear filtering (when the system output is a noise-free version of the input signal), the design of the soft stroke sensor was based on a recurrent network, with feedback connections enclosing several layers according to the equation:

$$y(t) = f(y(t-1), \dots, y(t-n_y), u(t-1), \dots, u(t-n_u)) \quad (\text{III.1})$$

where the next value of the dependent output $y(t)$ is regressed on previous values of both the output signal and an independent (exogenous) input signal $u(t)$, whilst a Feed-Forward NN is adopted to approximate the function f .

Model selection also includes the identification of the relevant model inputs from the candidate set of independent variables (regression selection). It was performed through the analysis (in terms of magnitude) of the estimated cross-correlation between each candidate input and the dependent variable of interest. As a result, only the fork stroke, the pitch rate and the vehicle speed appear to be the relevant inputs and included in the considered model structure (see Figure III.1). For the model estimation, data samples corresponding to about 60 minutes of the motorcycle riding was considered as the Training Set, whereas the remaining data were used as the Test Set for the next model validation.

In details, the soft sensor for the rear suspension stroke was modelled by exploiting the Neural Network Toolbox included in MathWorks MATLAB™ and considering the Serial Parallel scheme for the Network Training: considering as the model feedback the true output instead of the estimated sample allows the Static Back-Propagation to be adopted for training the network and more accurate results to be achieved (More details in Appendix I).

The model estimation of NARX Network was performed by varying the number N of neurons in the hidden layer (ranging from 5 to 25) and the tapped delay d_{in} for inputs and output signal (ranging from 10 ms to 100 ms), thus resulting in a total of 25 combinations. More in details, the training of the soft sensor was performed by adopting the Levenberg-Marquardt algorithm and a maximum of 1000 epochs for each parameter's configuration.

As a result, the NARX Network characterized by $N = 15$ neurons in the hidden layer, $d_{in} = 100$ ms, showed the best performance in terms of minimum prediction error about the training dataset and was considered for the development of the Soft Sensor. More details about the development of the NARX network, the training and validation process will be given in section III.3. In Appendix I there are more information about the NARX models.

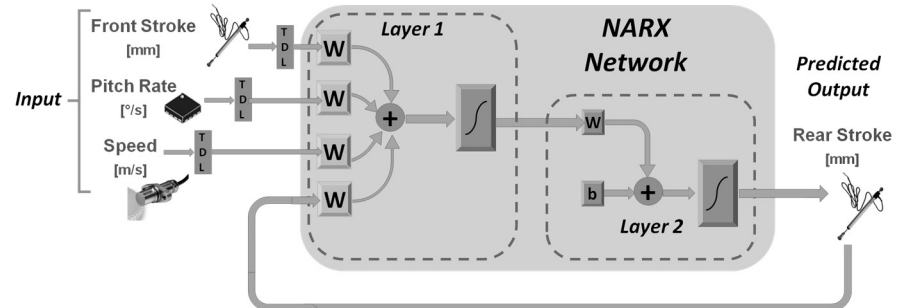


Figure III.1 Scheme of the rear stroke soft sensor

III.2 The measurement setup for the rear stroke soft sensor

The designer of a soft sensor should typically adopt the trial and error procedure depicted in Figure III.2, which includes three main steps: Data Collection and Filtering, the definition and estimation of a Model Structure, and the Model Validation.

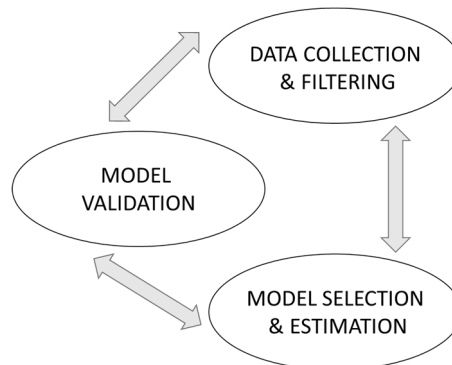


Figure III.2 The trial and error procedure

When the model fails the validation phase, the designer should go back to any of the previous steps and use all available insight to critically try different choices until the success of the model validation indicates that the procedure can stop. The data driven approach has been applied for the soft sensor design and validation of the rear suspension stroke by considering the SUZUKI GSX-1000 model as test motorcycle (described In Chapter II). From the available sensors on the test bed, for the rear stroke soft sensor only the sensors included in Table III.1 have been used. The basis of any data driven approach is a measurement campaign which may give a suitable insight into the relevant variables, the system order, the delays, the sampling time, the operating range and the nonlinearity about the process of interest. Then, the soft sensor

designer should perform the critical analysis of the available data to choose the candidate influential variables and the main events.

Table III.1 *Main sensors used for the rear stroke soft sensor*

Sensor Type	Model	Manufacturer	Symbol	Mounting notes
Linear stroke sensor	SLS130	Penny & Giles	S_{front}	fixed to the fork and measuring the front suspension stroke
			S_{rear}	mounted between the frame and rear wheel and measuring the rear suspension stroke
Magnetic encoder	970-011	Dorman	V_{bike}	fixed to the front wheel and measuring the motorcycle (longitudinal) speed
Gyroscope	L3GD20	ST	$Gyro$	fixed to the frame and measuring the motorcycle pitch and roll velocities

Moreover, in order to remove offsets, mitigate high frequency noise, and prevent the larger magnitude variables to be dominant over smaller ones, the designer should perform the identification process by including suitable digital data filtering and scaling. Finally, to prevent the influence of inconsistent data, the designer should apply suitable outlier identification strategies.

About the data collection for the system of interest, the motorcycle riding refers to a test lap which includes various profiles (cobblestone stretch, urban and extra-urban road, concentrated obstacles) in order to introduce different excitation modes of the suspension system.

As a result, a data logging was achieved with reference to the following signals: fork stroke, pitch rate, vehicle speed, as independent variables and the rear shock stroke as dependent variable. Data recording was carried out at the sampling frequency of 1 kHz, then a resampling at 100 Hz was performed to match the dynamics of the semi-active suspension and the loop frequency typically adopted by the control strategies.

The min-max normalization method was adopted for data scaling, whereas the detection of outliers was performed according to the 3σ edit rule with a robust scaling proposed in (Bi & Berrett, 2003), so that the mean and standard deviation of each variable of interest are replaced respectively by the median and the median absolute deviation from the median.

III.3 Tuning of the NARX network

In order to obtain good performance with the use of a NARX network, it is important to dedicate particularly attention on the tuning parameters related to the network achievable with the training process. The network has been trained to predict the output signal of the rear stroke sensor in a motorcycle

suspension system, starting from measurements done by the sensors set introduced in paragraph 2. More in details, the training process aims, thanks to a training dataset, to tune the parameters of the neural network, necessary to obtain the right reconstruction of the output. The NN toolbox of MATLAB was used to carry out the mathematical analysis and the simulation of the network created.

Firstly, an accurate analysis regarding the input data for the training process must be done; for this reason, dozen files containing sets of acquisitions performed riding on different types of road have been investigated. It is a good point the selection of a training dataset containing a high dynamic of the analysed quantity. For the purpose, the training set on which the network was trained, has been chosen without excessive anomalous trends and with an excursion of all possible conditions (see Figure III.3). The importance to choose the correct training set help to exalt the generalization properties of the neural network, indeed the use of a training set without a high dynamic excursion, could introduce errors in the reconstruction of signals in which there are more excursions than the training set.

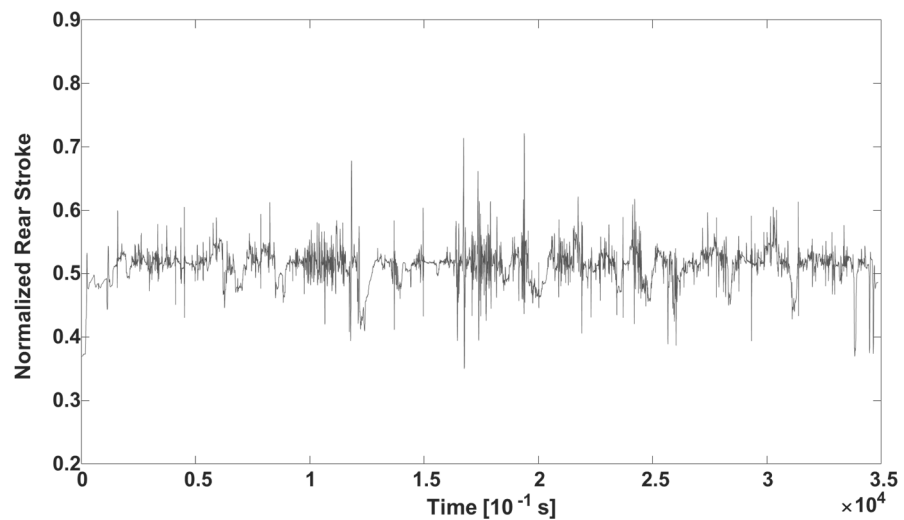


Figure III.3 *Example of training dataset*

The rear suspension stroke will be delayed by an asperity of the road compared to the front suspension stroke. The time delay can be calculated thanks to the motorcycle wheelbase and a fixed longitudinal speed; it would be more correct to implement a delay calculation related to the speed of the motorcycle, but it will be left to future implementations (De Luca & Doria, 2007). For this reason, a forward shift of 5 samples has been operated on the signal measured by the rear stroke sensor and used as training dataset, in order to synchronize

the rear stroke signal with the front stroke signal which is above 50ms in advance.

Also, a normalization operation has been done on the data used as input for the neural network referring on each sensor range: 0-150 [mm] for the front suspension stroke, -80, 80 [°/s] for the gyroscope, 0-200 [km/h] for the longitudinal speed and 0-110 [mm] for the rear suspension stroke.

Remembering that for a NARX networks, the output is given by a non-linear function related to the value of the output considered in the previous instants and from the value of the exogenous variable also observed in the past moments, therefore it is necessary to choose the number of instants corresponding to the samples observed in the past moments; this type of parameter will be named d_{in} .

For the training process the Levenberg-Marquardt algorithm as training function has been selected. The training function is responsible to update the weight and link values according to the training grade. The training algorithms of nonlinear systems can normally be divided in two principal groups: gradient descent algorithms and Gauss-Newton algorithms.

The Levenberg-Marquardt (LM) training algorithm has been developed to work on the strengths of the two groups in order to take advantage of both. More in details, the LM algorithm is an iterative regression technique, now considered standard for solving multivariable nonlinear problems. The algorithm can be described as composed of a slow, but converging gradient descent phase, followed by a faster Gauss-Newton resolver. Unlike the classic Error Back Propagation algorithm, the LM algorithm does not suffer of a rather slow convergence with the consequent risk of occurring in local minima. Moreover, the LM algorithm is particularly fast in learning with respect to Error Back Propagation (EBP) algorithm, especially when the number of inputs is particularly high. Other training functions were also evaluated, but the LM algorithm has demonstrate the best generalization properties. The LM algorithm is also called as "Bayesian regularization process" (Chaki, 2019).

Also, the number of epochs in which the training was performed represent an important parameter: increasing the minimum number of training periods has a positive effect on the error committed by the network. For the aim a maximum of 1000 epochs have been selected after the evaluation of the mean epochs required for a good training process (less than 1000).

Another important parameter to choose for the training process is the number of neurons in the hidden layer N which is directly proportional to the complexity of the neural network. There is not a theory to indicate the correct number of neurons N , but some indications could be achieved from the behaviour of the training set reconstruction during the training process.

Indeed, if the size of the neural network is too small, they will be unable to learn the relevant information from the training set and therefore they will have a limited capacity for generalization, vice versa if the size is too big, the

network will begin to memorize the details of the training set and it will fail to learn the basic rules, obtaining however a little generalization.

In Figure III.4 is reported the output (blue line) of an undersized network unable to predict the signal on which it was trained (red line).

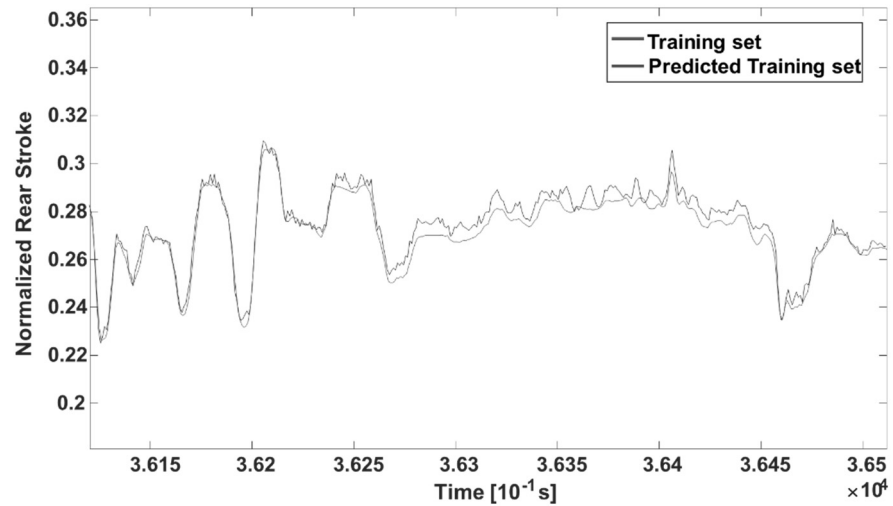


Figure III.4 Magnification of the undersized NARX output (red line) versus the Training set (blue line)

Oversizing the neural network by increasing the number of neurons does not help in terms of results, but simply worsens the functioning of the network, causing in some cases resonance phenomena.

As reported in paragraph III.1 the training process has been done on a series-parallel scheme. The series-parallel scheme is important for the training of the NARX network since it has as input also the real signal of the rear stroke sensor, enabling a better reconstruction of the same.

Clearly, the necessity to give as input the rear stroke sensor corresponding to the desired output, excludes the use of the series-parallel scheme in a context different from the training process. For this reason, the series-parallel scheme must be converted into a parallel-parallel architecture, removing as input the data coming from the rear stroke sensor and putting in feedback as input the predicted output (see Figure III.5).

One of the best practices is to not underestimate the performance of the series-parallel network since it represents the best theoretical performance achievable with a parallel-parallel network. An example of prediction of the Training set with the series-parallel scheme is reported in Figure III.6

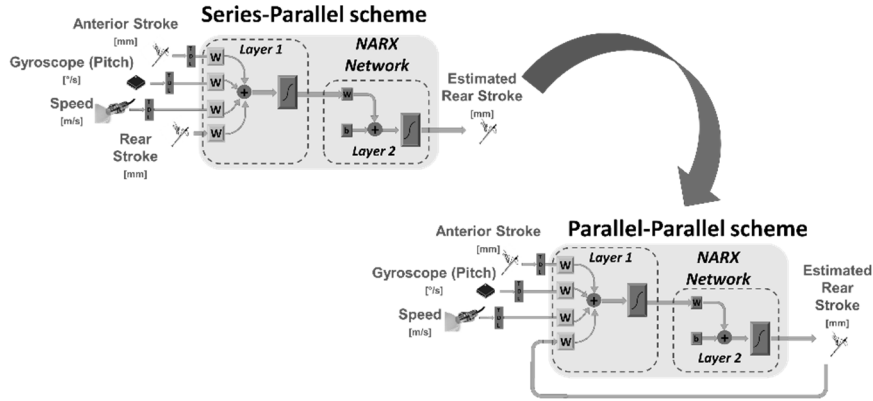


Figure III.5 Conversion from Series-Parallel to Parallel-Parallel scheme

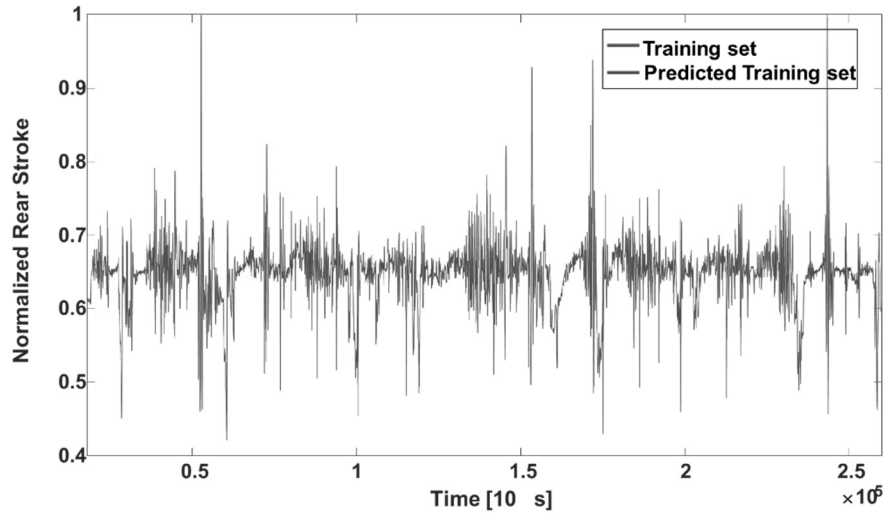


Figure III.6 Prediction of the Training set

Considering the tunable parameter previously presented, in order to find the best architecture, the number of neurons N and the number of samples d_{in} were varied. With a data-driven approach, table III.2 was created to perform a first skimming, using as threshold of acceptance the $E_{r,mean} \% < 10\%$ defined as:

$$E_{r,\%} = 100 * \text{mean}_i \left(\frac{|Y_p - Y_t|}{|Y_t|} \right) \quad (III.2)$$

Where y_p is the predicted output and y_t is the reference output coming from the rear stroke sensor.

Table III.2 Comparison between 25 neural networks

<i>N</i> number of neurons	<i>d_{in}</i> number of samples	$E_{r,mean}\%$ <10% ?
5	1	No
5	3	No
5	5	No
5	7	No
5	10	Yes
10	1	No
10	3	No
10	5	Yes
10	7	Yes
10	10	No
15	1	No
15	3	No
15	5	No
15	7	Yes
15	10	Yes
20	1	No
20	3	Yes
20	5	No
20	7	Yes
20	10	Yes
25	1	No
25	3	Yes
25	5	No
25	7	No
25	10	No

The table shows a set of 25 analyzed neural networks, but only 10 of them present a prediction error $E_{r,\%}$ lower the 10% and will be considered in the next steps.

In order to compare the performance of different (ANNs) architectures, rather than use synthetic indexes like the $E_{r,\%}$ that provides information only on the overall performance of a neural network, a graphical tools useful for providing both synthetic and detailed indications on the actual performance of the network have been adopted and will be presented: the Regression Error Characteristics (REC).

The Regression Error Characteristics (REC) curve plots on the y-axis the relative occurrences of the regression function outputs that are within a given error range (tolerance, on the x-axis). Thus, the resulting curve is the (estimated) cumulative distribution function (CDF) of the relative regression

error E_r , defined as the relative difference between the model prediction y_p (at each sampling point) and the corresponding true output y_t measured by the actual sensor:

$$E_r = \left| \frac{y_p - y_t}{y_t} \right| \quad (\text{III.3})$$

The area over the curve (AOC) represents the expected mean error (biased estimation) and provides a measure of the mean accuracy: the closer curve to the y-axis represents the better performance exhibited by the estimated model (see Figure III.7).

It is easy to observe, the integral information provided by the REC curve disregards the time evolution of the regression error. On the other hand, the 'local accuracy' represented by the regression errors corresponding to consecutive (predicted) output samples is fundamental within the context of the sensor validation (Betta, Capriglione, Pietrosanto, & Sommella, 2011).

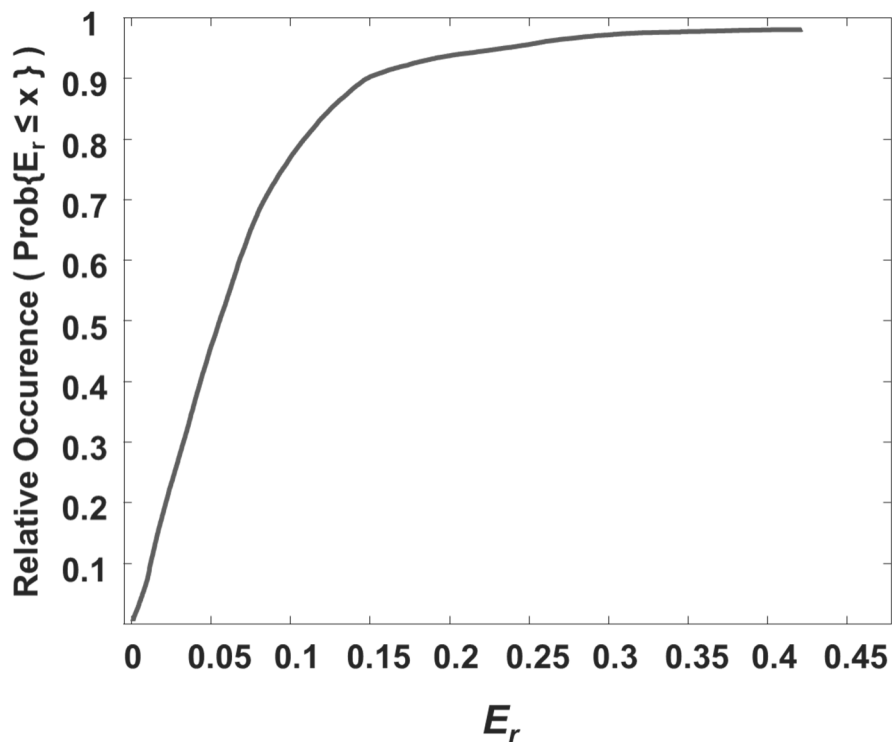


Figure III.7 Example of a REC curve

The REC reported in Figure III.8 shows a comparison between the 10 neural networks achieved by the observation of table III.2. The figure shows that the networks with $N=15$, 20 and 25 neurons and $d_{in}=10$ respectively, present the best performance in term of REC. The last choice among the tree last neural

network architecture has been made evaluating each architecture on 5 test set, in order to estimate the behavior on new sets of data never used before.

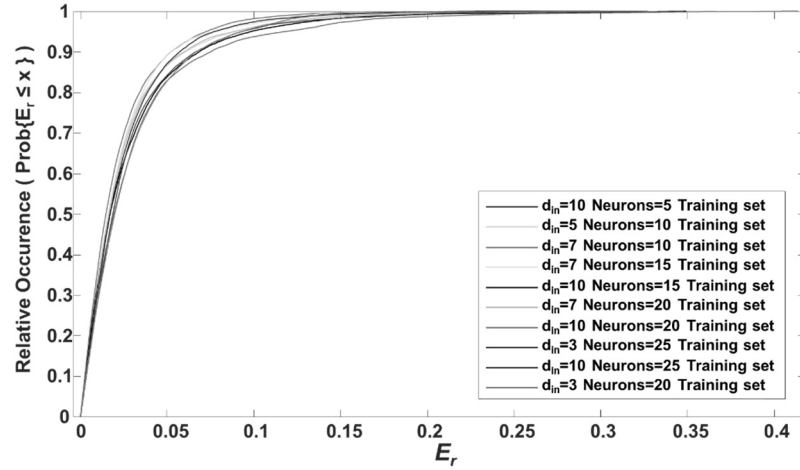


Figure III.8 REC: Comparison among the best 10 neural networks on Training set

In the figures below (Figure III.9 to Figure III.11) are reported the Regression Error Characteristics of the three best neural networks obtained on different Testing sets not related to each other.

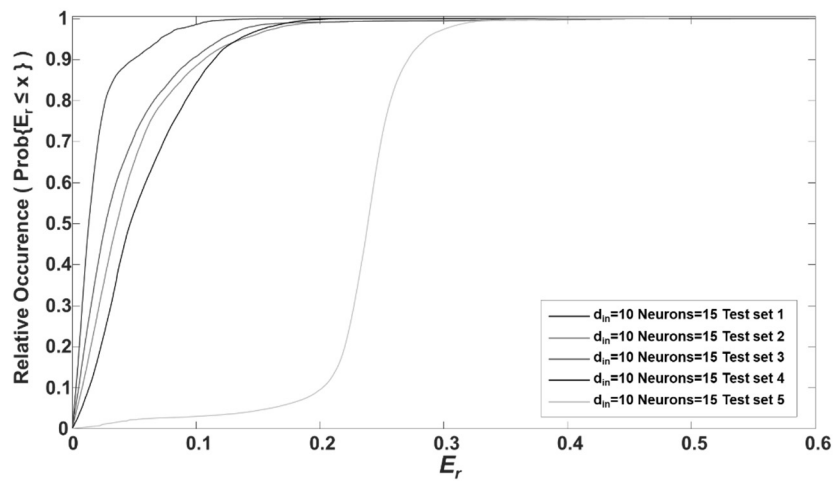


Figure III.9 REC: Comparison among different tests set using $d_{in}=10$ and Neurons = 15.

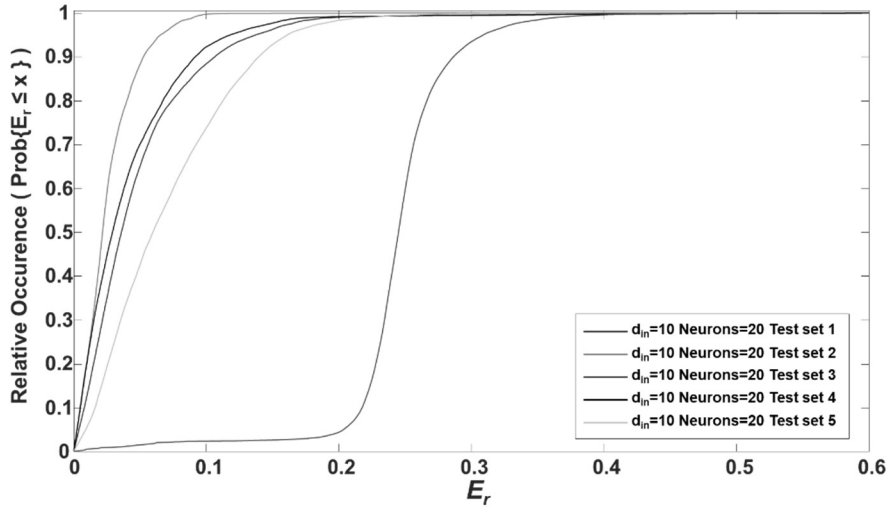


Figure III.10 REC: Comparison among different test sets using $d_{in}=10$ and Neurons =20

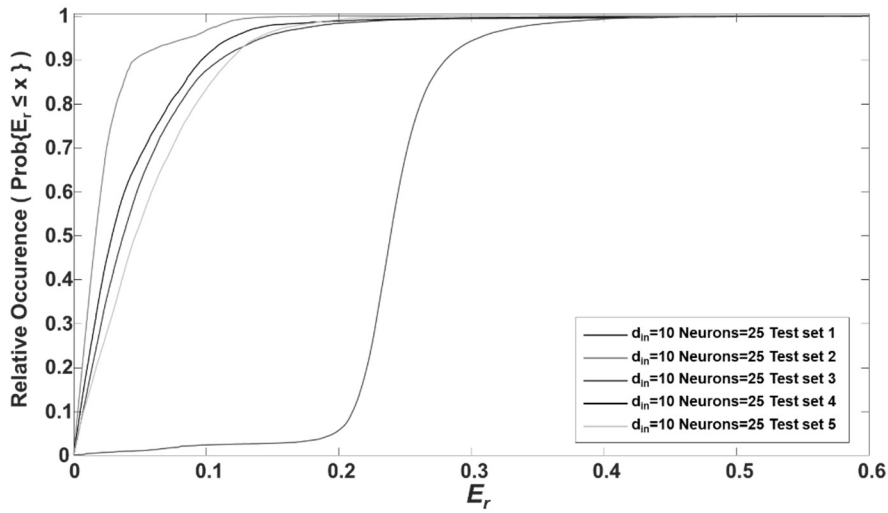


Figure III.11 REC: Comparison among different test sets using $d_{in}=10$ and Neurons =25

It is possible to observe that regardless the architecture of the network, the performance varies depending on the test set used. The neural network does not reconstruct different sets with the same error.

In order to determine which of the three neural networks presents the best performance, the Regression Error Characteristics obtained with the Test set 1 and 3 has been plotted on the same graph. These Test sets have been chosen for the comparison since they cover almost all the real situations in which the motorcycle can be gather.

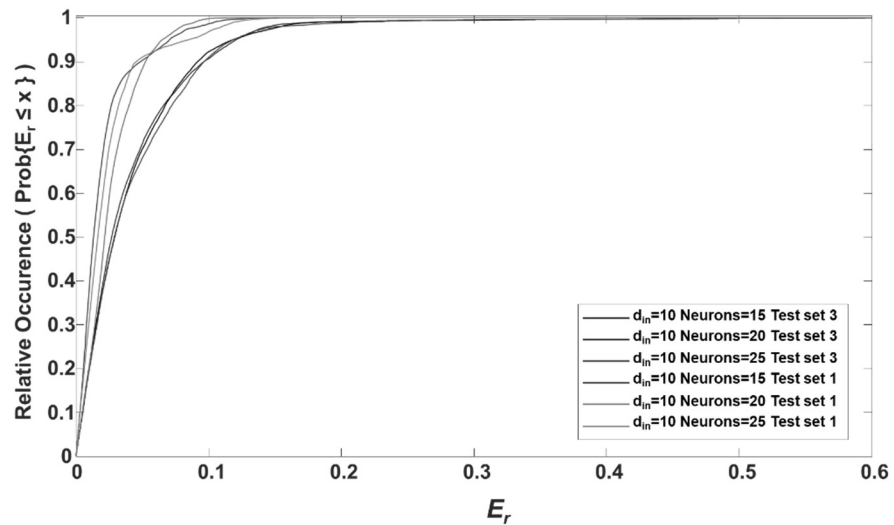


Figure III.12 REC: Comparison among test set 1 and 3 using the three best neural networks achieved

The Figure III.12 above shows a slightly difference between the three networks, but the one with 15 neurons and $d_{in}=10$ presents the best performance in both the test sets and has been chosen for the further analysis.

III.4 Performance of the rear stroke soft sensor

The NARX Network achieved, has been used to predict the rear suspension stroke using as input a Test set and considering the parallel-parallel scheme. The results shown in Figure III.13 (entire Test set) and Figure III.14 (magnification), highlight the particularly good capability of predicting the reference (measured) rear stroke (blue line). In details, the mean value for the relative regression error E_r according with Eq. III.1 is equal to 4%.

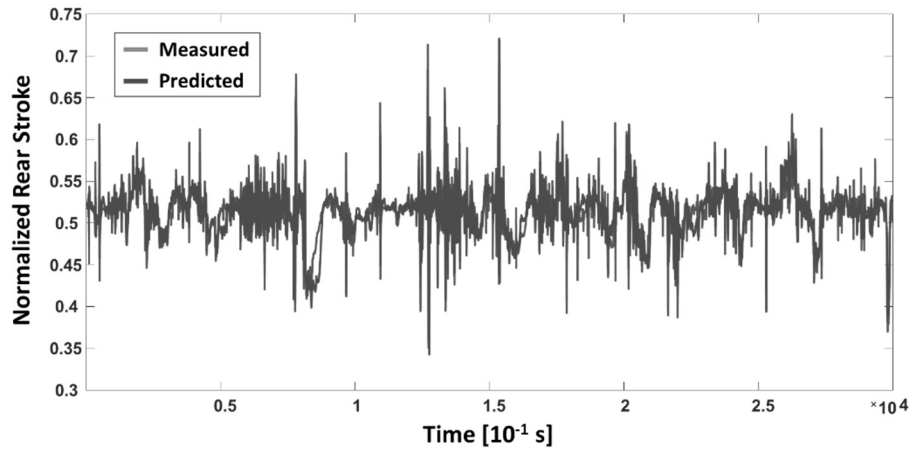


Figure III. 13 Prediction of the rear suspension stroke

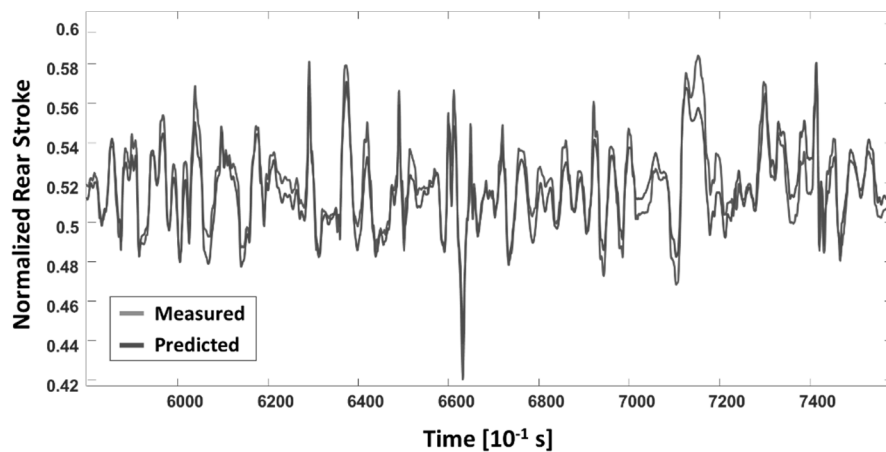


Figure III. 14 Magnification of Figure III.13

However, the “local accuracy” represented by the regression errors corresponding to consecutive (predicted) output samples is fundamental within the context of the sensor validation. As an example, when the focus is devoted to the fault detection performance, once a suitable threshold has been selected as tolerable error, the ANN able to guarantee a small percentage of error exceeding the threshold during a fixed time interval may be preferred to a different model which assures the lowest mean error (on the whole data set) but it is characterized by many time intervals when a higher percentage of threshold overcoming occurs.

This feature may be highlighted by the Sliding Occurrence Error (SOE) curve (Figure III.15): with reference to a moving window constituted by L_s

successive samples, it plots the error tolerance (defined as the mean relative deviation $E_{mean,L}$) on the x-axis and the corresponding relative occurrences in the moving window of the regression error on the y-axis. In details, at each sample point, the maximum absolute deviation $E_{mean,L}$ is defined according to:

$$E_{mean,L}(i) = \frac{1}{L_s} \sum_{k=0}^{L_s-1} \left| \frac{y_p(i-k) - y_m(i-k)}{y_m(i-k)} \right| \quad (III.4)$$

where L_s is the number of samples included in the window length L . In other words, the SOE curve represents the survivor function of the error tolerance.

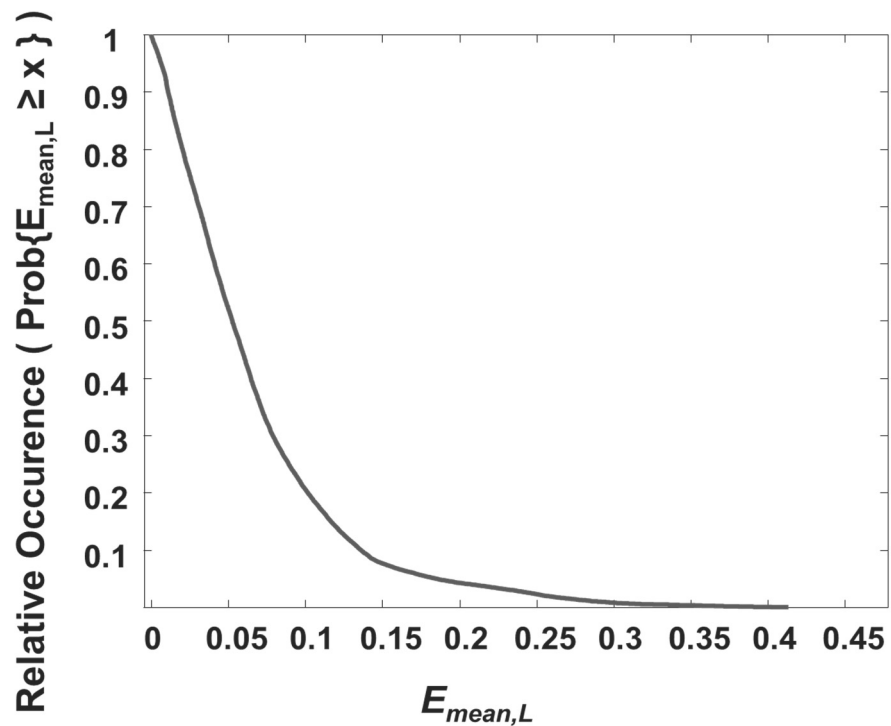


Figure III.15 *SOE curve*

A further comparison between the rear stroke sensor achieved with the NARX Network soft sensor and the rear stroke achieved with the Half Car Model (HCM) equations (presented in Chapter II, section 2) has been performed in terms of the provided local accuracy by considering the corresponding SOE curves, which are reported in Fig III.16.

For each sensor, the results about the mean relative deviation of the predicted output have been determined for two values of the sliding window length L (10 ms and 0.5 s). The opposite behaviors showed by the soft sensors, as the window length increasing, highlight the NARX Network is more effective in

the signal tracking of the reference rear stroke during short time observation intervals. As an example, about the worst predicted cases (ten percent of the Test set), the minimum value for the relative deviation $E_{mean,L}$ decreases from 5% to 4% for the NARX Network, whereas the analogous amount for the HCM stroke sensor increases from 20% to 21%.

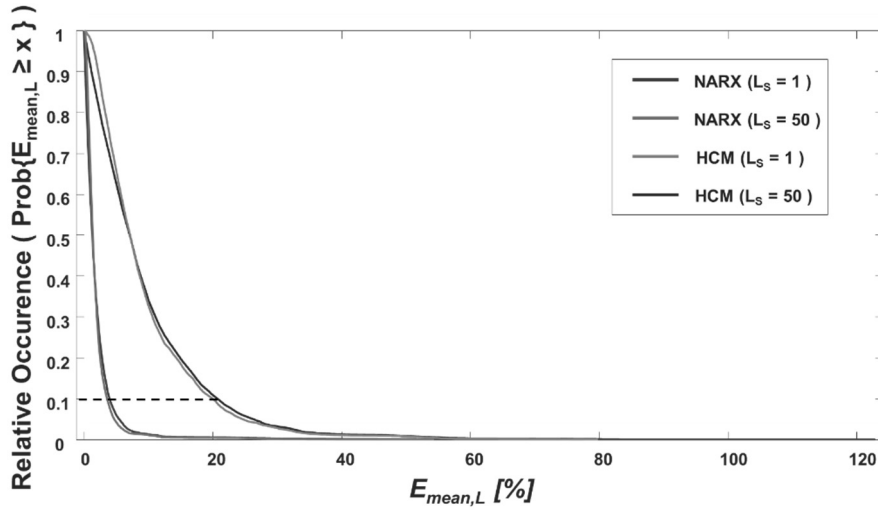


Figure III.16 SOE: Comparison among HCM soft sensor and NARX soft sensor

Chapter IV

Front stroke velocity soft sensors based on accelerometers

As previously reported in chapter II, the vertical dynamic of a motorcycle should be measured from a set of sensors, which may include gyroscope, accelerometers, stroke sensors and/or new sensors.

This last topic is strategic for massive exploitation of semi-active suspensions because the adopted sensors (as external devices to the embedded system controller) are crucial as for both reliability and cost of the whole system (Carratù, Pietrosanto, Sommella, & Paciello).

In this chapter, two different soft sensors have been developed to predict the suspension stroke velocity from acceleration measurement: a former based on digital filtering techniques and latter based on an Artificial Neural Networks (ANN). Both the approaches rely to the signals acquired by accelerometers installed on body and wheels of a motorcycle.

IV.1 The measurement setup

Both the front soft sensors developed to estimate the measurement of the suspension stroke velocity have been evaluated with reference to a Suzuki GSX-1000 motorcycle adopted as experimental test bed and described in Chapter II. More in details, the sensors used for the aim are (see Figure IV.1):

- a linear potentiometer P, measuring the front suspension stroke;
- a triaxial accelerometer S1, fixed to the motorcycle steering;
- a triaxial accelerometer S2, fixed to the front wheel;
- a triaxial accelerometer S3, fixed to the motorcycle frame;
- a triaxial accelerometer S4, fixed to the rear wheel;
- a MDLog.

For a detailed description of the set of sensors used refer to Chapter II.

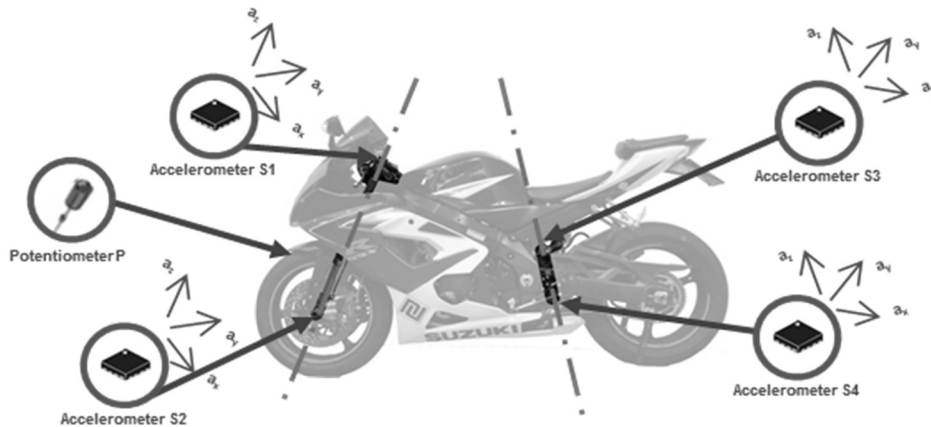


Figure IV.1 Sensors used for the front stroke velocity soft sensor

Data collection was carried out by completing different test lap of 300 seconds, which encompasses different operational conditions, including:

- a cobblestone for a length of about 50 m, which excites the suspension response to the pitch;
- a stretch of rough urban road, negotiated at average speed equal to 50 km/h, that introduce a mixed pitch-heavy movement of the motorcycle on a broad spectrum;
- a stretch of extra-urban road, negotiated at average speed equal to 90 km/h, that typically introduces a pure heavy movement;
- a region with 4 speed bumps (30 m equidistant from each other), which allows the suspension behavior to be verified against sudden load transfer.

Both the approaches introduced have been adopted for estimating the velocity of the front suspension on the basis of the analytical redundancy among the measured acceleration signals. As a reference for the velocity signal, the derivative of the front suspension stroke measured by the linear potentiometer has been considered.

IV.2 Front stroke velocity soft sensor based on digital filtering

The approach is based on the digital filtering and integration of the frame (S1) and front (S2) wheel acceleration signals according to the scheme depicted in Figure IV.2.

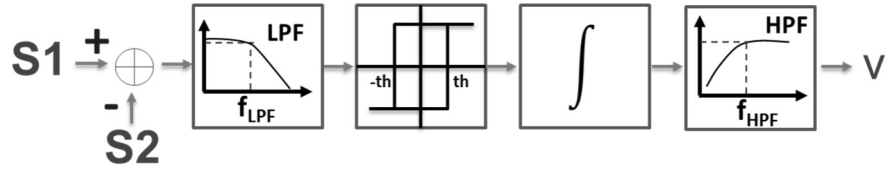


Figure IV. 2 *Front stroke velocity soft sensor scheme based on digital filters*

It includes a second order low pass filter (LPF) to remove high frequency noise, a hysteresis block, a filter like integrator (FLI), a first order high pass filter (HPF) in order to solve the acceleration integration drift typically caused by offset and noise at lower and high frequencies (Gavin, Morales, & Reilly, 1998). The model estimation of the soft sensor leads to the choice of the cut-off filter frequencies, which should consider both the signal frequency range and the delay introduced in the original signals by the filters.

In detail, the cut-off frequency f_{LP} for the LPF should be designed with the aim of preserving the frequency band of the physical phenomena of interest but limiting the phase delay. The frequency range of the vibration experimented by the motorcycle dampers is typically 1-10 Hz (see Figure IV.3), thus, f_{LP} should be investigated in the range [20÷60] Hz so that:

- the noise at high frequencies may be reduced.
- a slight phase delay is added in the band of interest.

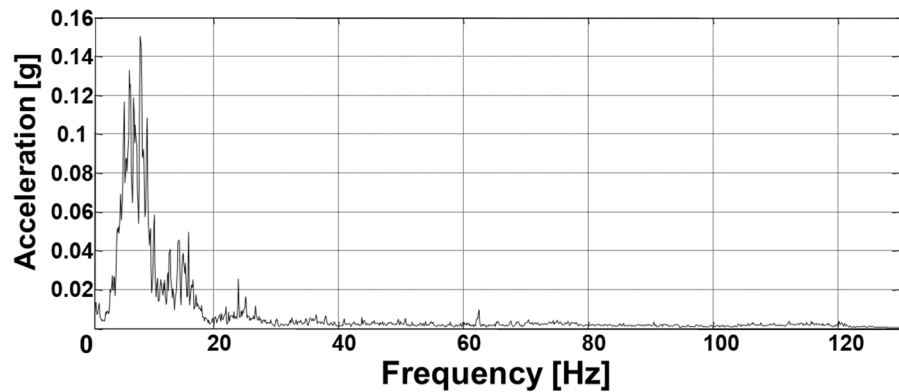


Figure IV.3 *Spectrum of the suspension acceleration during a ride*

The hysteresis block (characterized by the threshold th) is introduced to reduce the broad band noise which should be emphasized by the next FLI (implementing the trapezoid approximation).

The HPF (characterized by the cut-off frequency f_{HP}) is finally adopted in order to both remove the offset emphasized by the integrator and limit the drift caused by the unknown initial condition of accelerometers.

IV.3 Tuning of the digital filters parameters

A sensitivity analysis has been performed in order to give an indication for the selectable parameters of the filtering chain. In details, the examined cost function is represented by the relative (percentage) error $E_{r,\%}$ between the predicted and expected suspension velocity, averaged on the whole dataset according to the following equation:

$$E_{r,\%} = 100 * \text{mean}_i \left(\frac{|V_{\text{pred},i} - V_{\text{ref},i}|}{|V_{\text{ref},i}|} \right) \quad \forall |V_{\text{ref},i}| > 400 \text{ mm/s} \quad (\text{IV.1})$$

A threshold of 400 mm/s was imposed in the evaluation of the $E_{r,\%}$ since a worst prediction of the velocity suspension stroke under that threshold, does not influence a control strategy scheme for a semi-active suspensions system (Acocella, Anchini, Paciello, Pietrosanto, & Sommella, 2010).

The reference suspension velocity as previously reported, has been measured by the linear potentiometer mounted on the motorcycle. Table IV.1 reports the analyzed range of the parameters introduced in each block of the corresponding scheme.

Table IV.1 Main parameters of the stroke velocity soft sensor based on digital filtering approach

Symbol	Description	Analyzed Range	
		min	MAX
f_{LP}	(First order) Low Pass Filter cut-off frequency	20 Hz	60 Hz
th	Hysteresis threshold	0.1 m/s ²	0.5 m/s ²
f_{HP}	(First order) High Pass Filter cut-off frequency	0.5Hz	1.0 Hz

As results of the performed sensitivity analysis, the cut-off frequencies for the Low and High Pass Filters appears as the most significant parameters to be tuned in order to achieve the lowest $E_{r,\%}$. In detail, the three-dimensional surface depicted in Fig. IV.4, report the estimated dependence of the prediction error on both the parameters: the best combination represented by $f_{LP} = 45$ Hz and $f_{HP} = 0.7$ Hz allows (in conjunction with the threshold $th = 1$ m/s²) to maintain $E_{r,\%}$ lower than 25%.

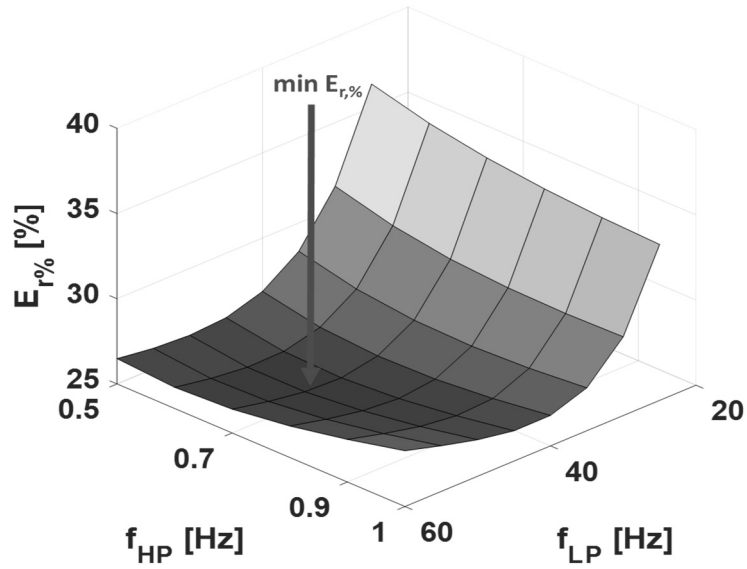


Figure IV.4 Sensitivity analysis for the front stroke velocity soft sensor based on digital filtering

IV.4 Performance of the front stroke soft sensor based on digital filtering

About the model validation of the soft sensors developed, Figure IV.5 shows the predicted (front) suspension velocity (red line) when the Testing dataset is considered for the model input (blue line).

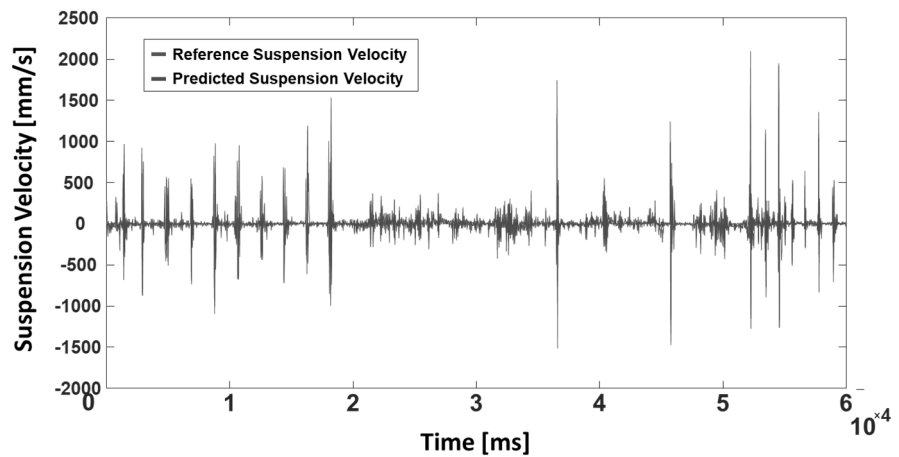


Figure IV.5 Prediction of the front suspension stroke using the digital filter

More in details, the soft sensor is able to satisfyingly predict the velocity when the suspension experiments significant compressions and rebounds strictly correlated to the motorcycle riding over bumps and/or holes (Figure IV.6) whereas, the prediction error increases when lower suspension velocities are observed (Figure IV.7).

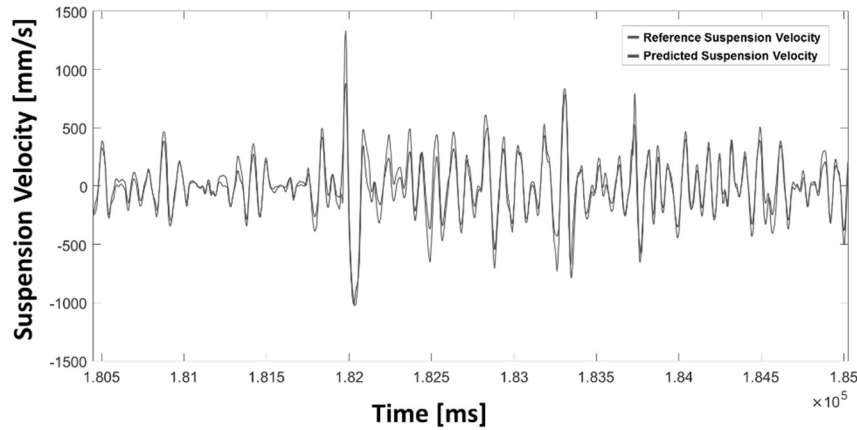


Figure IV.6 Magnification of Figure IV.5 with respect to high suspension velocities;

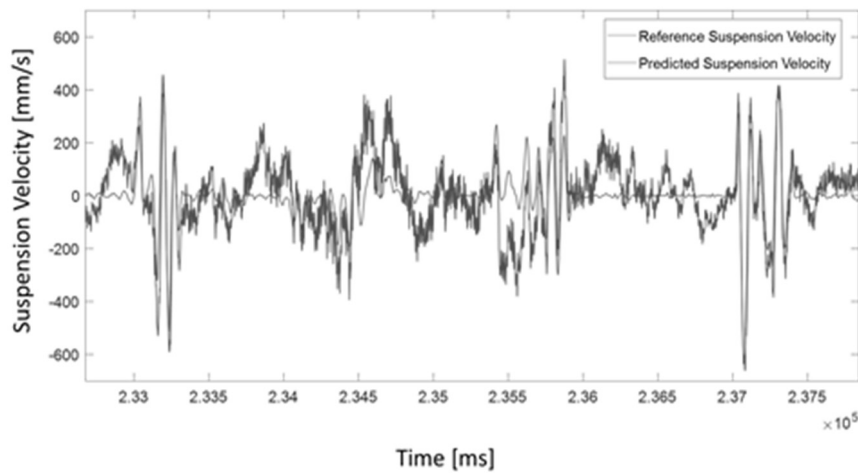


Figure IV.7 Magnification of Figure IV.5 with respect to low suspension velocities

The front stroke velocity soft sensor based on the digital filtering is able to predict the suspension velocity greater than ± 400 mm/s with a relative error

percentage lower than 20% that could be enough for a pitch control strategy presented in Chapter II.

IV.5 Front stroke soft sensor based on NARX network

An alternative method to develop a front stroke velocity soft sensor using accelerometers, is to use a black-box approach with an Artificial Neural Network (ANN) as reported in Chapter III for the rear stroke soft sensor. In the analysis and design of nonlinear systems, artificial neural networks have become standard tools due to the good performance obtained for a large number of real-world application.

The Figure IV.8 shows the developed soft sensor based on a particular type of neural network named Nonlinear AutoRegressive network with eXogenous inputs (NARX).

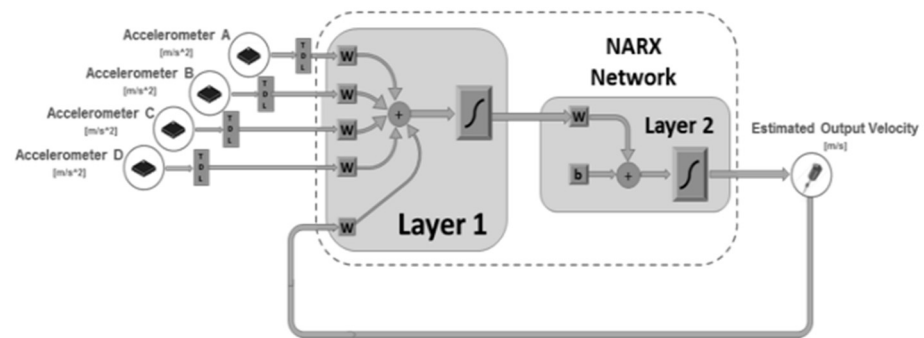


Figure IV.8 Front stroke velocity soft sensor scheme based on NARX

IV.6 Tuning of the NARX network

The model estimation of NARX network was carried out with reference to the following parameters:

- input topology (combination of triaxial front and rear acceleration signals);
- number N of neurons in the hidden layer (ranging from 5 to 25).
- the tapped delay d_{in} and d_{out} for inputs and output signal (range from 1 ms to 20 ms).

More in details, the training of the soft sensor was performed by adopting the Neural Network Toolbox included in MathWorks MATLAB considering the Levenberg-Marquardt algorithm and a maximum of 1000 epochs for each parameters configuration.

The procedures used to carry out all the parameter of the training process are the same of Chapter III.

As a result, the NARX network, which takes into account the six front acceleration signals, and is characterized by $N = 16$ neurons in the hidden layer, $d_{in} = 7$ ms, showed the best performance in terms of minimum prediction error about the training dataset. Figure IV.9 depicts the predicted suspension velocity (red line) from the best NARX model (when the closed loop is considered) together with the reference values (blue line) derived by the linear potentiometer signal.

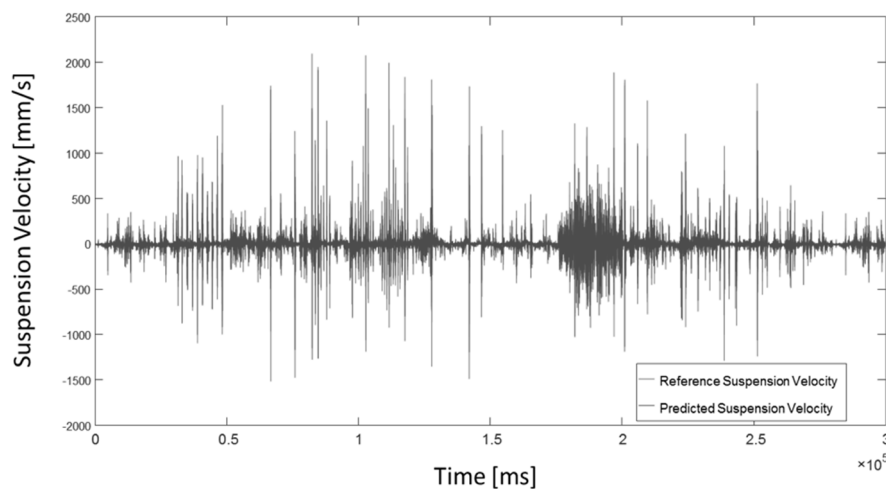


Figure IV.9 Predicted front stroke velocity vs real front stroke velocity

IV.7 Performance of the front stroke velocity soft sensor based on NARX network

According to the Eq. IV.1. the $E_r\%$ achieved is typically lower than 20%, which could be an interesting result for the use of the soft sensor in the loop control of a motorcycle suspension systems.

The behavior showed by the soft sensor based on the NARX Network, allows to best predict the low suspension velocity during a motorcycle riding (see Figure IV.10), whereas, it typically underestimates the velocity peaks observed in correspondence of the bumps and/or holes (see Figure IV.11).

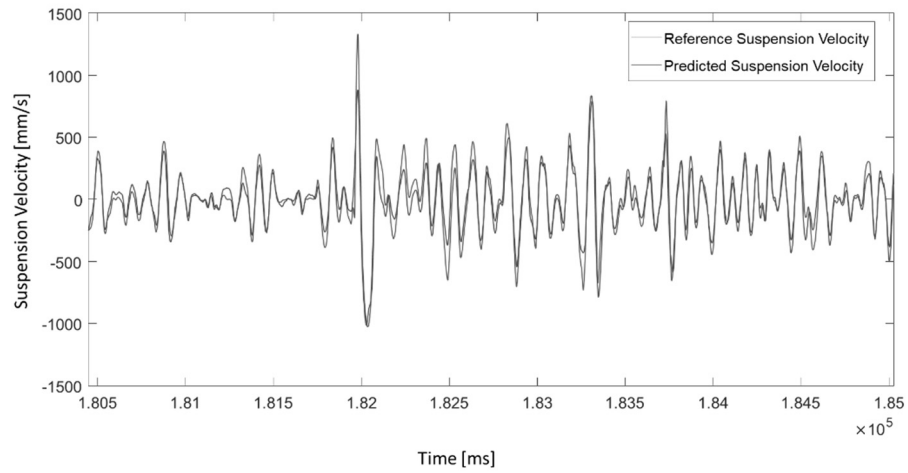


Figure IV.10 Magnification of Fig. IV.9 with respect to high suspension velocities

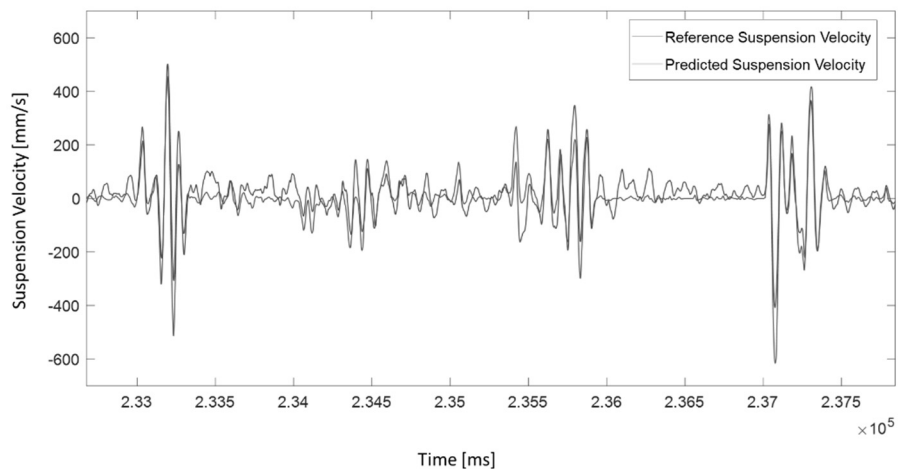


Figure IV.11 Magnification of Fig. IV.9 with respect to low suspension velocities.

IV.8 Comparison of the proposed front stroke velocity soft sensors

Considering the soft sensor based on the digital filtering, a complementary behavior is exhibited by the sensor based on the NARX Network, which is able to predict the low suspension velocity during the motorcycle riding,

whereas, it typically underestimates the velocity peaks observed in correspondence of the bumps and/or holes.

As a summary of the experimental comparison between the analyzed software sensors, the relative prediction error per velocity class $E_{r,V_{class}}$ according to eq. IV.2

$$E_{r,V_{class}} = \frac{|V_{pred,i} - V_{ref,i}|}{|V_{ref,i}|} \quad \forall |V_{ref,i}| \in V_{class} \quad (IV.2)$$

is showed in Figure IV.12 for different suspension velocity classes together with the corresponding occurrences when the distribution of the testing dataset is considered.

The software sensor based on the digital filtering of the front suspension acceleration is able to predict the suspension velocity greater than 400 mm/s with a relative error percentage lower than 20% (as well as an error distribution characterized by low variance, see Figure IV.13 and the corresponding boxplots) whereas, the prediction error percentage exhibited by the NARX Network may be preferred in the range [0÷400] mm/s, which includes the most frequently observed classes for the (reference) suspension velocity.

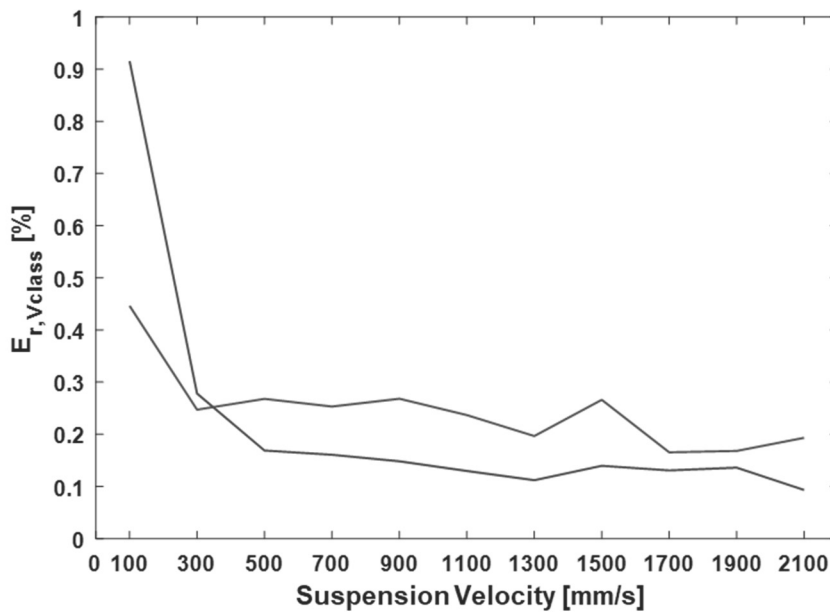


Figure IV.12 Relative prediction error per velocity class

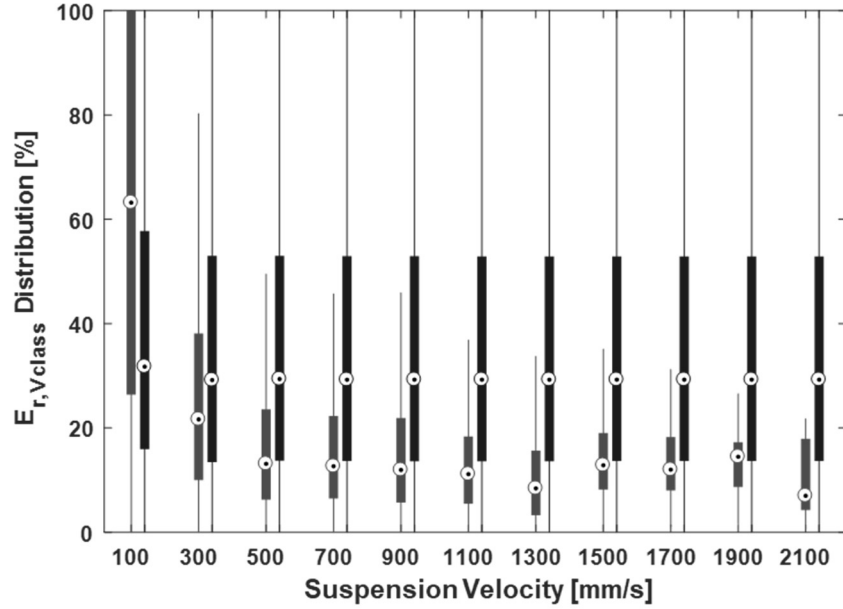


Figure IV.13 Distribution of relative error percentage with respect to the suspension velocity classes

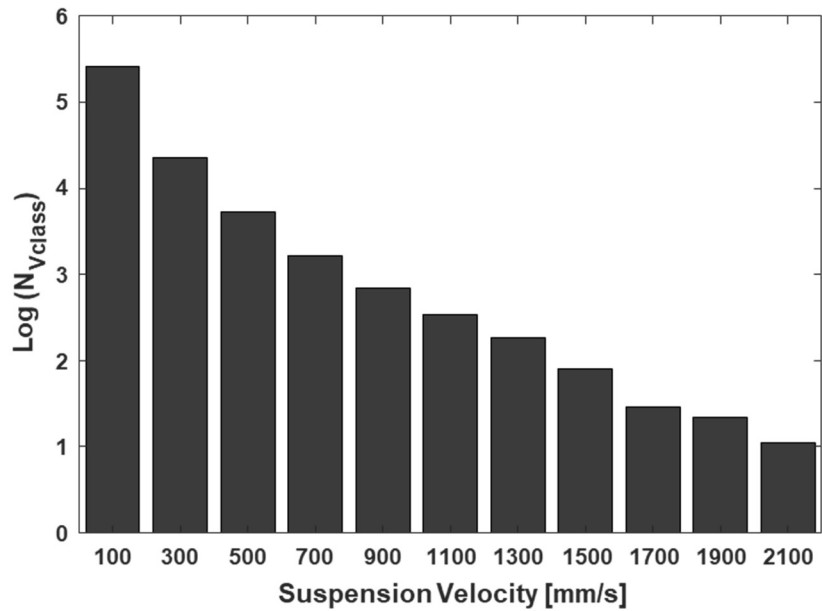


Figure IV.14 Histogram of suspension velocity for the Testing dataset

On the basis of the previous model validation, the soft sensor based on digital filtering of the frame and front wheel acceleration is expected to be suitable in cost-effective applications such as the semi-active suspension control systems because it may allow replacing the linear potentiometer. As an example, Figure IV.15 shows the controlled electric current typical for adjustable magneto-rheological fork as function of the suspension velocity corresponding to the motorcycle riding over a bump. As you may note, the control algorithm is more sensitive to the high range of the suspension velocity (in correspondence of which the greater demanded damper stiffness is assured by the modulated intensity of the electrical current).

On the other hand, the adoption of the soft sensor based on the NARX model should be preferred in order to develop effective IFD (Instrument Fault Diagnosis) schemes, which are necessary to improve the fault-tolerance property of the adjustable suspension system. Indeed, it can be paralleled with the linear potentiometer (hardware sensor), and faults can be detected by the comparison between the outputs of the hard and soft sensors. Moreover, the soft sensor can be exploited to provide an estimate of the hardware sensor output in case of sensor fault. Therefore, it can be used as a back-up device until the hardware sensor is not replaced during the service.

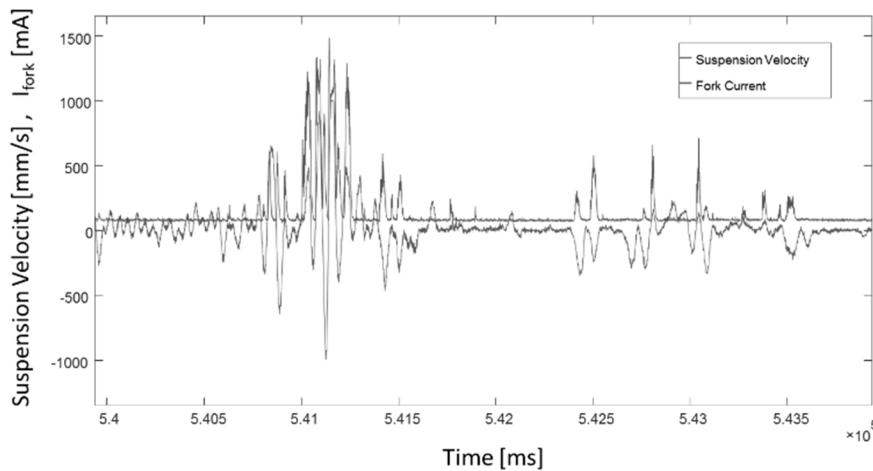


Figure IV. 15 Example of semi-active suspension control: stroke velocity and controlled current

Chapter V

IFD scheme for the rear stroke sensor

Improvements in safety, comfort, and performance of the current transportation applications, have been also allowed by the increasing adoption of new sensors and electronic devices inside automobile and motorcycle contexts. As expected, the reliability and effective operating of such important systems strongly depend on the reliability and the accuracy of data coming from sensors involved in the measurement and control chain (Poussot-Vassal, Spelta, Sename, Savaresi, & Dugard, 2012). Thus, fault tolerant systems should be employed considering that a fault tolerant control may be achieved only if very sensitive and fast Instrument Fault Detection (IFD) schemes are included in the control loops (Betta & Pietrosanto, 2000). These schemes should be able to identify in real time the faults that could occur also on the sensors, in order to readily trigger suitable recovery strategies for managing (or accommodating if possible) an occurred fault, and avoiding that wrong data coming from faulty sensors could affect the correct working of the monitored system (Capriglione, Liguori, & Pietrosanto, 2007).

In the motorcycle framework, a challenging topic is represented by the development of semi active suspension systems electronically controlled as reported in Chapter I and mainly devoted to guarantee an effective contact between tires and road with the aims of improving the passenger's safety and comfort. As said in Chapter II, all the strategies proposed in literature for controlling the damping coefficient of the suspension are based on the measurement information about the vehicle dynamics outputted from a set of different sensors among which linear potentiometers, accelerometers, stroke sensors, gyroscope, and magnetic encoders. Among these sensors, the one used to measure the vertical extensions and vertical compressions of the rear suspension plays a fundamental role in the control strategy. Due to their simplicity, low costs, and good performance in terms of linearity, the most used sensors are linear potentiometers. Nevertheless, they could be unreliable in the long run.

As a consequence, the fault detection of such sensor is strongly recommended to avoid wrong, and in some cases, dangerous motorcycle behaviors. Thus, the employment of IFD scheme for sensors monitoring such a quantity is crucial (Capriglione D. , Carratù, Pietrosanto, & Sommella, 2018) .

The strong constraints of space existing in the motorcycle context, address to avoid duplication or triplication of sensors and related cabling, thus preferring the development of IFD scheme based on analytical redundancy. To this aim, due to the correlations existing among the quantities measured by the sensors involved in the control loop, an analytical redundancy-based IFD scheme will be presented during this Chapter.

More in details, the rear stroke soft sensor presented in Chapter III has been adopted to generate the residuals for the rear suspension stroke sensor. In particular, the proposed solution employs a Nonlinear Auto-Regressive with eXogenous inputs (NARX) network because its attractive feature in effectively consider for the nonlinearities of the system under test. The main goal is the experimental verification of both diagnostic performance and promptness against typical kind of faults that could be experienced, including also “small faults”, “hold” and “losing calibration” (e.g. due to slight variations of the input/output sensor curve). Then, a first challenge is in verifying the feasibility of real-time implementation of such IFD scheme on a general-purpose microcontroller typically employed in motorcycle context as STM32F4. That is not a trivial result since in many applications involving ANNs, special purpose DSP based, or FPGA based systems are required for assuring a real-time operating. Furthermore, several challenges in terms of hardware/software platform to be used with particular reference to costs, compactness and weights, current consumption, and electromagnetic interference issues have to be faced as well.

The validation tests and analysis have shown that the proposed IFD scheme can be successfully developed on this kind of architectures by assuring the real-time operating. The last section is dedicated to the experimental results related to the on-line validation of the realized IFD system on the test bed presented in Chapter II and showing diagnostic and dynamic performance achieved.

V.1 The IFD scheme

The simplified block diagram of the IFD scheme is reported in Figure V.1. It is based on the following main blocks:

- Data acquisition: this stage samples and stores the output of 4 sensors: the rear suspension stroke S_{rear} , the front suspension stroke S_{front} , the velocity of the motorcycle V_{bike} and the pitch $Gyro$.

- Soft sensor: this stage, based on the suitably designed and tuned neural network presented in Chapter III, provides the predicted value of S_{rear} at a given time instant as a function of the S_{rear} values measured at previous instants and of S_{front} , V_{bike} , $Gyro$;
- Residual generator: it computes the difference between the measured and the predicted values of S_{rear} in order to highlight the symptom of the faults;
- Decision maker: it implements the rules needed to correctly detect different types of faults.

In the following the Soft sensor, the Residual generator and the Decision maker are described in detail.

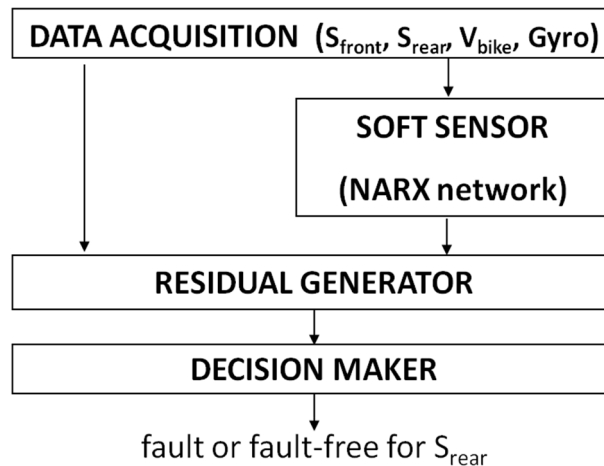


Figure V.1 Block-diagram of the proposed IFD scheme

V.1.1 Soft sensor used

The soft sensor for prediction of the rear stroke suspension has been developed by adopting a *Nonlinear Auto-Regressive with eXogenous inputs (NARX) Neural Network*, which considers the front suspension position, the pitch rate of the motorcycle body and the longitudinal speed of the vehicle. These quantities are strictly correlated to the rear suspension behavior as highlighted by the half car model proposed to predict the steady state conditions for the in-plane motorcycle dynamics. However, the simplified model does not allow the steering and the linkage nonlinear effects to be correctly estimated in terms of the corresponding varying wheelbase and transfer load. Following the trend

in the literature (Capriglione, Carratù, Liguori, Paciello, & Sommella, 2017) about the prediction of non-linear dynamic systems, the choice of the NARX neural networks seem to be the most promising solution thanks to the good capability of noise filtering typically exhibited. Different NARX Networks have been analyzed by adopting the Matlab Neural Toolbox. The development of the rear stroke soft sensor has described in Chapter III.

V.1.2 Residual generator

The second block of the proposed IFD scheme compares the prediction of the rear stroke soft sensor based on NARX neural network with the output of the rear stroke sensor.

As previously reported, the instantaneous prediction of the soft sensor is satisfying for most of the experimental dataset (see Figure V.2). However, some conditions remain, where the percentage difference between the ground-truth and the predicted position are significant (see the highest peaks in Figure V.3).

Thus, a strategy based on moving average is employed for computing a more accurate residual.

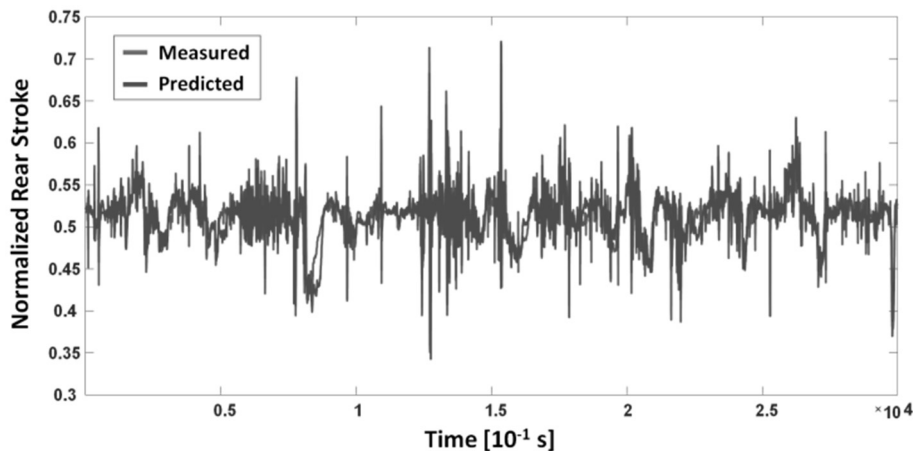


Figure V.2 Prediction of the rear suspension stroke by the NARX Network for dataset

As an example, Figure V.4 report the comparison between the output of the soft and hard stroke sensors, with reference to a poor local prediction of the rear suspension position (percentage error greater than 40%). Figure V.5 shows the residual $E_{mean,L}$ computed by the proposed block according to:

$$E_{\text{mean},L}(i) = \frac{1}{L_s} \sum_{k=0}^{L_s-1} \left| \frac{y_p(i-k) - y_m(i-k)}{y_m(i-k)} \right| \quad (\text{V.1})$$

where y_p and y_m are the predicted and measured stroke, L_s is the number of samples included in the moving window length L .

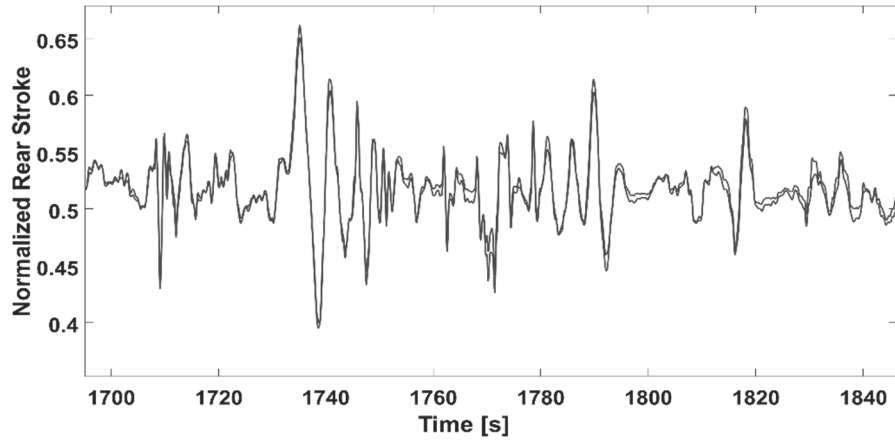


Figure V.3 Magnification of figure V.2 with reference to the highest peaks

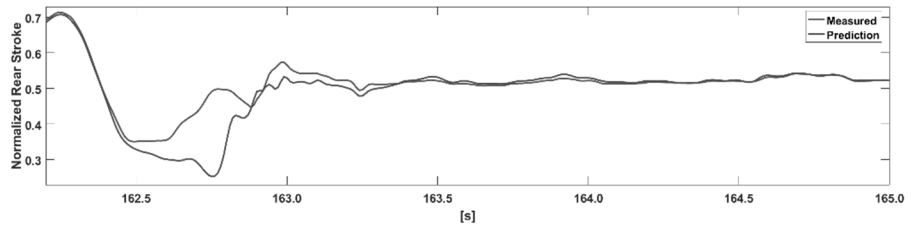


Figure V.4 Measured and predicted normalized rear stroke.

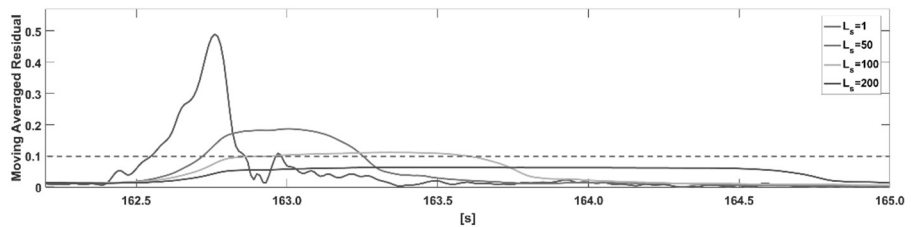


Figure V.5 Moving averaged residual versus L_s

As expected, greater values for L allow limiting the height of the local peaks introduced in the residual signal by the poor prediction. On the other hand,

exceeding in the moving average leads to obtain a not accurate prediction of the error when long observation periods are considered. Indeed, the local accuracy of the NARX Network with reference to the experimental dataset may be revealed by the graphical tool described in Chapter III. The *Sliding Occurrence Error (SOE)* curve plots the mean relative deviation $E_{mean,L}$ on the x-axis and the corresponding relative occurrences in the moving window of the regression error on the y-axis. Thus, the SOE curve may be interpreted as the survivor function of the error tolerance. As depicted in Figure V.6 and Figure V.7, about the worst predicted cases by the NARX Network (ten percent of the experimental dataset), the minimum value for the relative deviation $E_{mean,L}$ is less than 5%, when L equal to 500 ms is considered. Thus, a compromise value for L should be selected according to the method described in the following.

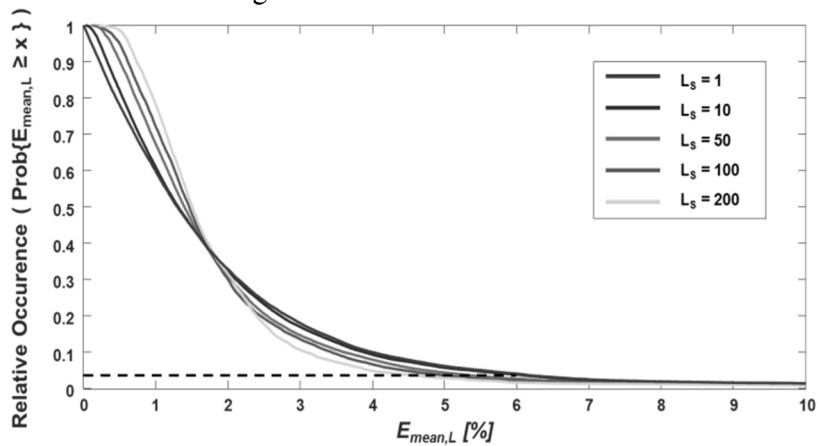


Figure V.6 SOE curves for NN as function of the window length L

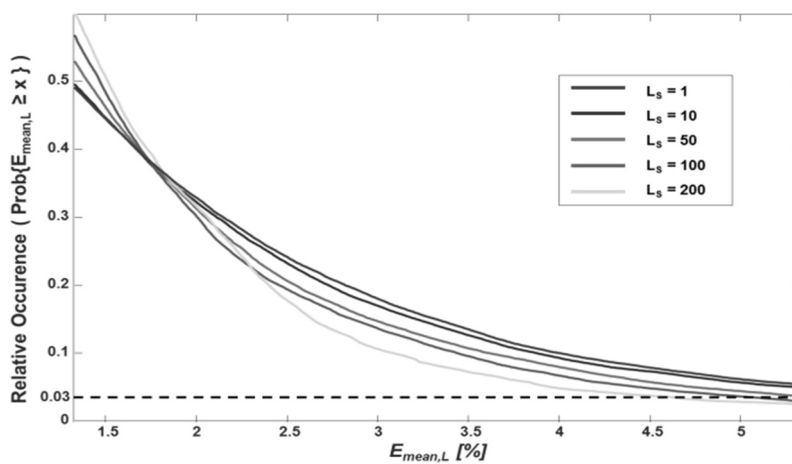


Figure V.7 Magnification of Figure V.6

V.1.3 Decision Maker

The proposed IFD scheme for the rear stroke sensor aims to reveal firstly the small faults, also known as “losing calibration faults”, mainly due to the device wear, tear and aging, or to other influence factors as the variation of the sensor power supply and which results as changing of the input/output curve of the sensor.

Such a kind of fault generally appears as slight amplitude deviation from the expected behavior and could be detected through the plausibility checks typically implemented in automotive ECUs only after hours or days from the occurrence, when the performance degradation implies unacceptable risk levels.

Moreover, the proposed scheme is devoted to also detect the open, short-circuit and hold faults. According to the proposed strategy schemed in Figure V.8, a fault is detected when the residual computed by the corresponding block exceeds a fixed threshold $T\%$ longer than an integer multiple n of the sliding window L .

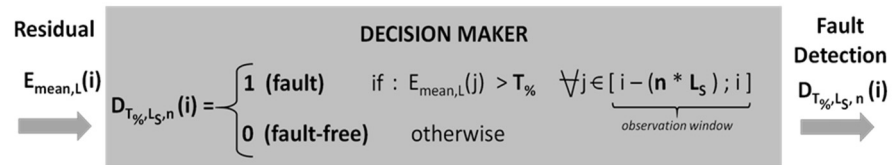


Figure V.8 The proposed detection rules

V.2 Validation of the IFD scheme

As far as the generation of faults, open circuit, short circuit, hold and losing calibration have been emulated starting by fault free signal. A short description of each fault condition is reported in the following.

V.2.1 Short and open circuit faults

Open circuit: from the time instant in which the fault occurs, the sensor output is driven to the maximum possible value (full scale).

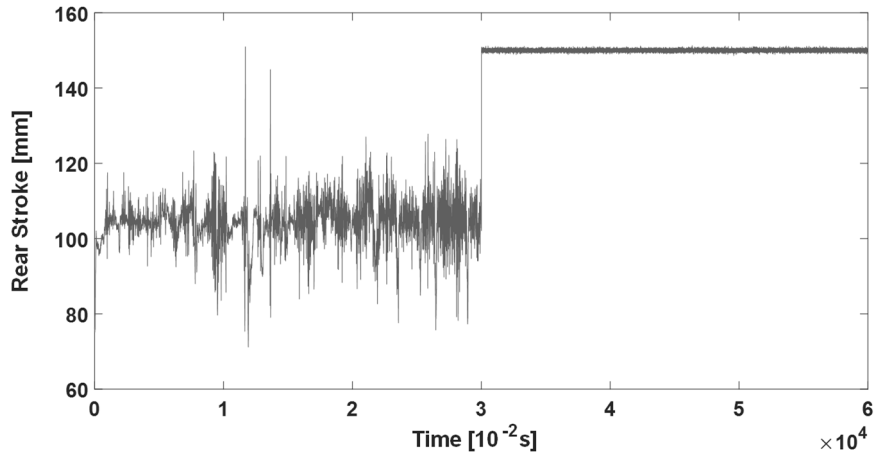


Figure V.9 *Open circuit fault*

Short circuit: from the time instant in which the fault occurs, the sensor output is driven to minimum possible value;

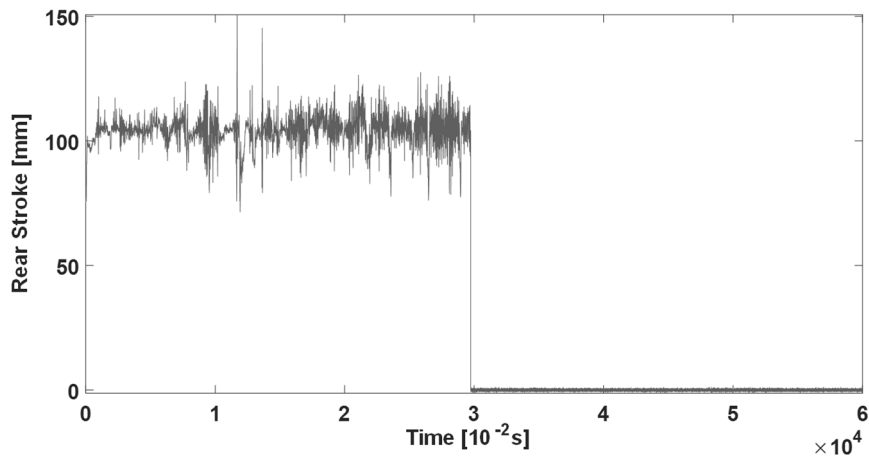


Figure V.10 *Short circuit fault*

V.2.2 Hold fault

From the time instant in which the fault occurs, the sensor output is kept constant to the last fault-free value;

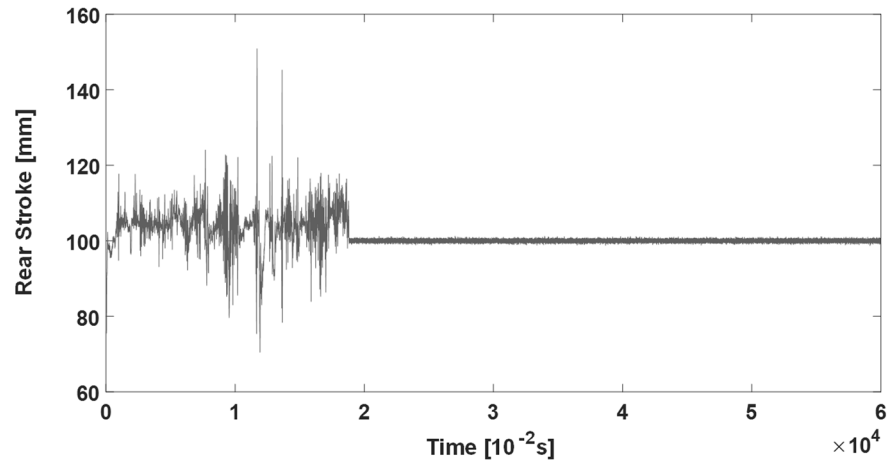


Figure V.11 *Hold fault*

V.2.3 *Losing calibration fault*

From the time instant in which the fault occurs, the sensor output is multiplied by a constant for changing the slope of its Input/Output curve;

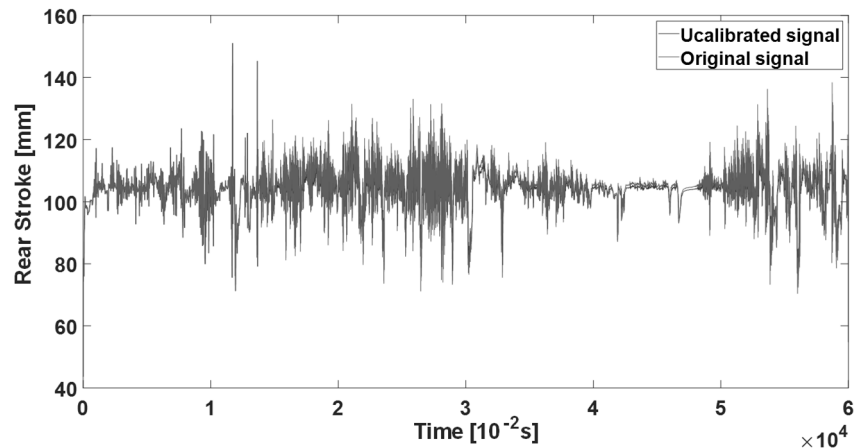


Figure V.12 *Uncalibration fault (10%)*

V.3 Experimental Results

Focus has been devoted to analyzing the best value for the window length L and the integer n when the most accurate NARX model and the level of the losing calibration $T\%$ are fixed. For each class of interest (losing calibration faults not lower than $T\%=10\%$, as well as the open, short-circuit and hold

faults), the instrument fault detection scheme has been verified against $N_{\text{faults}}=1000$ faults randomly introduced in the (measured) rear stroke samples of dataset, by considering the following performance indexes:

- the percentage FA% of false alarms, when threshold is exceeded for predicted samples corresponding to faulty-free sensor output;
- the percentage MD% of missed detections, when either threshold is not exceeded for predicted samples corresponding to faulty sensor output or threshold is exceeded after a maximum delay $t_{d,\text{max}}$ with respect to the fault insertion time;
- the percentage CD% of correct fault detections, when threshold is exceeded for predicted samples corresponding to unhealthy sensor output by the maximum observation time $t_{d,\text{max}}$.

The test results are summarized in Tables V.1-V.12 for L and n varying in the ranges [10÷200] ms and [1÷5] respectively, when $t_{\text{max}} = 120$ s is considered. Considering that the sum of the proposed indexes is equal to 100% for each combination of the L and n parameter, a satisfying performance may be obtained for all the fault types:

- very low values for the FA%, and MD% indexes (not greater than 1.0% and 2.0% respectively) are achieved when the proposed fault detection scheme is adopted by considering the moving observation window Lobs ($n \cdot L_S$ consecutive samples) in the range [40÷200] ms (see the gray regions highlighted in the corresponding Tables);
- the adoption of shorter sliding windows for residual generation ($L < 50$ ms) typically leads to poor performance in terms of FA% (see the values in the top-left corner of the False Alarm fault Tables) because of the prediction limits exhibited by the NARX model about the signal tracking for 10% of the Test set samples (as previously observed in Figure V.7);
- a larger sliding window ($L > 150$ ms) leads to poor performance in terms of MD% (see the values in the bottom-right corner of the Missed detection fault table) because the threshold exceeding is not completely satisfied for all the output samples within the observation time window (as depicted in the corresponding examples of Figure V.7);

- the open and short faults represent the easiest conditions to be detected (as shown by the large gray areas highlighted in the corresponding Tables) because of the significant residual achieved in correspondence of the fault insertion due to the extreme values for the measured signal;
- reasonably satisfying values of CD% and MD% may be achieved for the hold faults through shorter observation windows (see the small gray area highlighted in the corresponding Tables). Indeed, the measured suspension stroke is near the balance position for the greatest part of the dataset. Thus, further research could be addressed to include other detection rules based on the analysis of the derivative signal for the measured rear suspension stroke.

Moreover, a very good promptness of the decision maker has been achieved: for all types of the faults, the corrected detection is obtained after a mean time delay $t_{d,mean}$ lower than $(L_{obs} + 2 \text{ seconds})$.

Table V.1 *Detection of losing calibration faults: False Alarm percentage*

		L [ms]					
		10	20	50	100	150	200
n	1	97.7	97.6	54.0	0.2	0.0	0.0
	2	97.6	0.6	0.7	0.0	0.0	0.0
	3	97.6	0.4	0.0	0.0	0.0	0.0
	4	0.5	0.4	0.0	0.0	0.0	0.0
	5	0.3	0.0	0.0	0.0	0.0	0.0

Table V.2 *Detection of losing calibration faults: Correct Detection percentage*

		L [ms]					
		10	20	50	100	150	200
n	1	2.3	2.4	46.0	99.7	99.1	99.1
	2	2.4	99.3	98.9	98.1	89.7	97.2
	3	2.4	98.8	98.2	50.6	0.0	0.0
	4	99.4	98.7	98.1	0.0	0.0	0.0
	5	99.2	98.6	0.0	0.0	0.0	0.0

Table V.3 Detection of losing calibration faults: Missed Detection percentage

		L [ms]					
		10	20	50	100	150	200
<i>n</i>	1	0.0	0.0	0.0	0.1	0.9	0.9
	2	0.0	0.1	0.4	1.9	10.3	2.8
	3	0.0	0.8	1.8	49.4	100.0	100.0
	4	0.1	0.9	1.9	100.0	100.0	100.0
	5	0.5	1.4	100.0	100.0	100.0	100.0

Table V.4 Detection of Open Circuit faults: False Alarm percentage

		L [ms]					
		10	20	50	100	150	200
<i>n</i>	1	99.0	98.9	53.1	0.5	0.0	0.0
	2	99.0	0.9	0.6	0.1	0.0	0.0
	3	99.0	0.9	0.6	0.1	0.0	0.0
	4	1.1	0.9	0.2	0.1	0.0	0.0
	5	1.1	0.9	0.2	0.1	0.0	0.0

Table V.5 Detection of Open Circuit faults: Correct Detection percentage

		L [ms]					
		10	20	50	100	150	200
<i>n</i>	1	1.0	1.1	46.9	99.4	99.5	99.2
	2	1.0	99.0	99.2	99.1	98.8	98.7
	3	1.0	98.9	99.2	98.7	98.5	97.9
	4	98.8	98.9	99.0	98.6	97.9	96.9
	5	98.8	98.8	98.9	98.4	97.0	96.5

Table V.6 *Detection of Open Circuit faults: Missed Detection percentage*

		L [ms]					
		10	20	50	100	150	200
n	1	0.0	0.0	0.0	0.1	0.5	0.8
	2	0.0	0.1	0.2	0.8	1.2	1.3
	3	0.0	0.2	0.2	1.2	1.5	2.1
	4	0.1	0.2	0.8	1.3	2.1	3.1
	5	0.1	0.3	0.9	1.5	3.0	3.5

Table V.7 *Detection of Short Circuit faults: False Alarm percentage*

		L [ms]					
		10	20	50	100	150	200
n	1	98.6	98.6	53.9	0.4	0.0	0.0
	2	98.6	0.7	0.7	0.2	0.0	0.0
	3	98.6	0.7	0.7	0.2	0.0	0.0
	4	0.9	0.7	0.7	0.2	0.0	0.0
	5	0.9	0.7	0.7	0.2	0.0	0.0

Table V.8 *Detection of Short Circuit faults: Correct Detection percentage*

		L [ms]					
		10	20	50	100	150	200
n	1	1.4	1.4	46.1	99.6	99.7	99.5
	2	1.4	99.3	99.2	99.3	99.2	98.6
	3	1.4	99.3	99.2	99.0	98.6	98.6
	4	99.1	99.2	98.8	98.4	98.6	98.2
	5	99.1	99.1	98.7	98.4	98.4	98.0

Table V.9 *Detection of Short Circuit faults: Missed Detection percentage*

		L [ms]					
		10	20	50	100	150	200
<i>n</i>	1	0.0	0.0	0.0	0.0	0.3	0.5
	2	0.0	0.0	0.1	0.5	0.8	1.4
	3	0.0	0.0	0.1	0.8	1.4	1.4
	4	0.0	0.1	0.5	1.4	1.4	1.8
	5	0.0	0.2	0.6	1.4	1.6	2.0

Table V.10 *Detection of Hold faults: False Alarm percentage*

		L [ms]					
		10	20	50	100	150	200
<i>n</i>	1	98.2	97.9	51.3	0.0	0.0	0.0
	2	98.0	0.2	0.0	0.0	0.0	0.0
	3	98.0	0.2	0.0	0.0	0.0	0.0
	4	0.2	0.2	0.0	0.0	0.0	0.0
	5	0.1	0.2	0.0	0.0	0.0	0.0

Table V.11 *Detection of Hold faults: Correct Detection percentage*

		L [ms]					
		10	20	50	100	150	200
<i>n</i>	1	1.8	2.1	48.7	92.2	88.2	84.4
	2	2.0	99.4	90.8	76.6	67.0	42.1
	3	2.0	97.8	86.6	65.2	28.6	13.7
	4	99.2	96.8	74.9	38.1	12.5	7.9
	5	98.1	89.4	58.1	22.1	7.6	7.1

Table V.12 *Detection of Hold faults: Missed Detection percentage*

		L [ms]					
		10	20	50	100	150	200
<i>n</i>	1	0.0	0.0	0.0	7.8	11.8	15.6
	2	0.0	0.4	9.2	23.4	33.0	57.9
	3	0.0	2.0	13.4	34.8	71.4	86.3
	4	0.6	3.0	25.1	61.9	87.5	92.1
	5	1.8	10.4	41.9	77.9	92.4	92.9

V.4 On-line fault detection scheme

Will be now described the on-line implementation and operation, on a general-purpose microcontroller typically employed in motorcycle context, of the Instrument Fault Detection scheme for the rear stroke sensor.

The main goal is the experimental verification of both diagnostic performance and promptness against typical kind of faults presented in section V.2 A brief description of the realized Electronic Control Unit (ECU) based on the microcontroller unit (MCU) ARM-M4 STM32F4 (ST, 2018) is detailed in Section V.7. Most efforts are directed to firmware optimization in order to assure real-time operating through an architecture characterized by costs, clock frequencies, memory resources, arithmetic logic units less powerful than one achievable with DSP based and FPGA based systems (Samragh, Ghasemzadeh, & Koushanfar, 2017).

The experimental results related the on-line validation of the realized IFD system by highlighting diagnostic and dynamic performance achieved on the field will be shown.

V.5 The realized prototype

In order to on-line implement the proposed IFD scheme for the rear suspension stroke of a motorcycle, the system under test used is the SUZUKI GSX-1000 motorcycle equipped with several kind of sensors and Electronic Control Units (ECUs) devoted to accomplishing different tasks as engine management and automatically adapt the semi active suspension system of assuring safety and comfort in all motorcycle driving and running conditions described in Chapter II. Furthermore, to test the described IFD scheme, another ECU described in the following, specifically devoted to accomplishing such a task has been designed and realized.

V.5.1 Hardware

The realized IFD ECU is principally composed by three main functional blocks:

- Power converter;
- Transceiver CAN;
- Microcontroller STM32F405RGT6.

The first block contains the power converter and it is used to adapting the input voltage in the range of 10-16 V (typically provided by the motorcycle battery) to the 3.3 V necessary for the Microcontroller.

The second block is composed by a CAN transceiver and it is used for connecting the IFD ECU with the CAN bus network, thus allowing the acquisition of the data coming from the sensors of interest through the datalogger.

The third block is composed by the Microcontroller (MCU) STM32F4 and it is responsible of the execution of the IFD scheme. The selected MCU belongs to a family based on the general-purpose ARM®Cortex®-M4 which covers a wide range of applications in the fields of automotive, medical equipment, industrial, motor drive, home audio, alarm systems.

It is based on a 32-bit RISC-based core architecture and it can work up to a 168 MHz clock frequency.

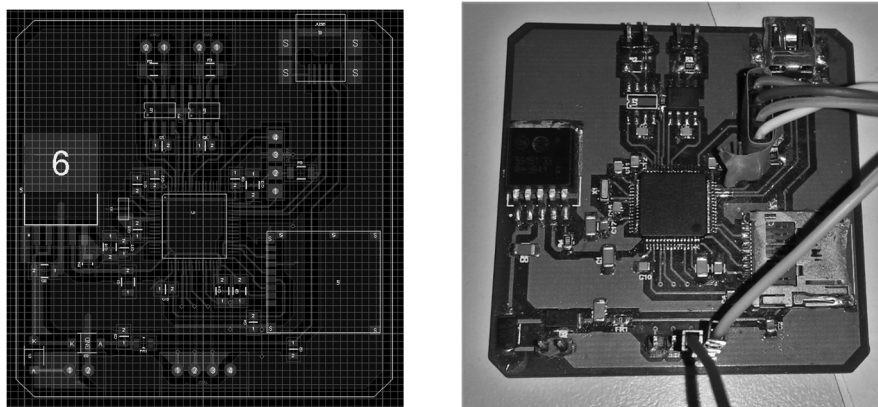


Figure V.13 *The realized IFD ECU*

The Cortex M4 core has a built-in Floating-point unit (FPU) single precision, which supports all ARM single-precision data-processing instructions and data types and implements a full set of DSP instructions. The architecture

incorporates high-speed embedded memories (Flash memory RAM up to 1 Mbyte, SRAM up to 192 Kbytes, backup SRAM up to 4 Kbytes). The device is equipped with three 12-bit ADCs, two DACs, a low-power Real-Time-Clock (RTC), twelve general-purpose 16-bit timers including and two general purpose 32-bit timers. Figure V.13 reports the realized prototype board hosting the above described blocks.

V.5.2 Firmware

This section describes the application of the Five STEP procedure (see Figure V.14) proposed in (Capriglione D. , Carratù, Pietrosanto, & Sommella, 2018) for generating and validating the implemented firmware. Starting from the Matlab™ script (m file) which implements the IFD scheme, the procedure allows achieving a code suitably arranged and optimized for fulfill the MCU and real time requirements.

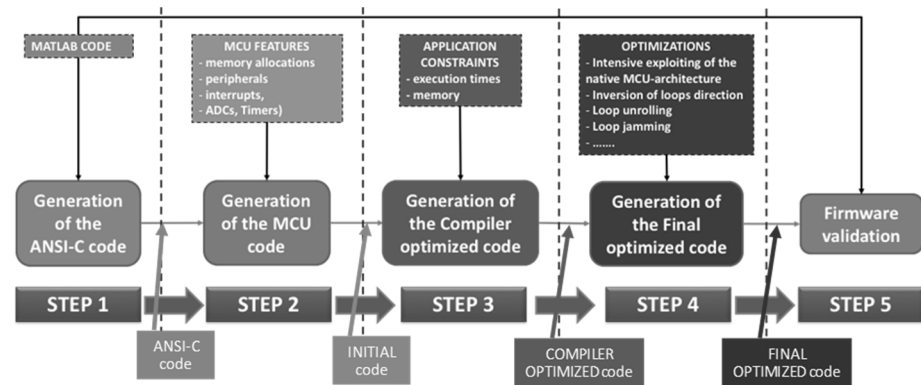


Figure V.14 Flow-chart of the proposed five-step procedure for the MCU-implementation of the IFD scheme

To develop the firmware on the MCU, the MDK Keil μ Vision (IDE) (KEIL, 2018) has been used. In the following, each step of the procedure and the most important phases and code optimizations are described.

STEP 1: Generation of the ANSI-C code

Starting from the Matlab™ script (m file), the corresponding ANSI-C code was achieved by means of the Matlab Compiler R2018a. This code version makes use of mathematical libraries data representation, I/O interfaces that cannot be directly employable on an MCU, so the ANSI-C code has to be suitably revised before to be developed on the MCU adopted (i.e. next STEP 2).

STEP 2: Generation of the INITIAL code

As for the generation of the INITIAL Code, several operations are made, among which the most important are:

- Allocating data structures coherently to the memory organizations and data representations available on the MCU used. In particular, 32 kB and 40 kB of FlashRAM and SRAM, respectively, were used.
- Selecting the most suitable clock sources and frequencies for each peripheral. Such a value has to be selected on the basis of a suitable trade-off among execution time, power consumption, frequency clock values (available on the MCU used) and for minimizing the probability of electromagnetic interference with nearby electronic devices and cables. As matter of fact, in motorcycle context the restricted spaces available bring to have high density of cables and electronic devices in small areas, so electromagnetic interference and heat dissipation problems cannot be neglected. From these points of view lower clock frequencies are preferable but such choice bring to a general increasing of the execution time.
In particular, it has been verified that the minimum possible clock frequency able to satisfy real-time constraints (i.e. 1 ms for this application) was equal to 22 MHz if all the optimizations described in the following steps C-D are adopted.
- Configuring and enabling line and events of interrupts as well as the use of Direct Memory Access (DMA) for each peripheral adopted. In particular, CAN BUS and a Timer were employed and configured for driving interrupt lines and regulate the flow of the main program.

STEP 3: Generation of the COMPILER OPTIMIZED code

The MDK Keil μ Vision (IDE) ARM Compilation Tools offers a range of options that can be combined to optimize the output code for best performance, for smallest code size, or for any performance point in the middle of these two requirements.

More in details the available compiler options are:

- Cross-Module Optimization:
This function is able to remove the not used functions to achieve a reduced code size.
- MicroLIB library:

It is a subgroup of the ISO standard C runtime library and offers a tradeoff between functionality and code size.

- **Link-Time Code Generation:**
A utility that trains the compiler to create objects in an intermediate format so that the linker can execute further code optimizations. Link-time code generation can reduce code size and allow the application to run faster.
- **Three optimization levels:**
The different levels of optimization let the programmer to achieve a trade-off between the level of debug information available in the compiled code and the performance of the code in term of time execution. In the following the possible optimization levels are briefly described.
 - “O0-level”: most of the optimizations are switched off and the code generated has the best debug view.
 - “O1-level”: a set of restricted optimizations is applied. More in details, the compiler applies automatic optimizations like removing redundant code and re-ordering instructions to avoid an interlock situation. The code generated is reasonably optimized, with a good debug view.
 - “O2-level”: optimizations applied at this level get advantage of ARM’s in-depth knowledge of the processor architecture in order to exploit processor-specific behavior of the given target. It generates well optimized code, but with limited debug view.
 - “O3-level”: applies the most powerful optimization. The optimization is in accordance with the user intention in order to obtain a space-optimized or time-optimized code. The code generated in this level is not useful for debug view.
- **Optimize for time:**
Teaches the compiler to optimize the code for the fastest execution time, at the risk of an increase of code size.

As a consequence, several optimization options could be selected and the combination of options to apply will depend on the optimization goals, for example in favoring either the smallest code size, or the best performance in terms of execution times (these are contrasting needs, therefore a suitable trade-off should be considered).

Since the memory required (32 kB and 40 kB of FlashRAM and SRAM, respectively) for the firmware implementation was very limited with respect to the one available for the MCU used, the options O1 level and Optimizing for time were enabled with the aim of reducing the execution time. By this way, for the selected clock frequency (i.e. 22 MHz) we have achieved an execution time equal to 1.6 ms. Therefore, in order to constraint the execution time to be less than 1 ms further optimizations are needed.

STEP 4: Generation of the FINAL OPTIMIZED code

The compiler optimizations allowed to improve the performance with respect to the INITIAL code, but further straightforward techniques have been employed to significantly increase the code efficiency in terms of both execution time and memory usage.

In particular, at first, all data structures were converted in 32-bit data type with aim of intensively exploiting the native MCU architecture.

Secondly, as suggested in literature (Srivastava & Wall, 1999), (Ullman, 1986) and (Calder, Grunwald, & Zorn, 1994), the following main practices have been adopted:

- Loop Jamming: to exploit the redundancy operations and to eliminate the overhead in one loop;
- Inversion of loops direction: the use of a decrement counting and of an unconditional jump from zero flag allows reducing the use of general compare instructions;
- Loops unrolling: the replication of the body loop in order to exploit the pipeline architecture;
- Invariant code motion: statements or expressions are moved outside the body of a loop without affecting the semantics of the program;
- Strength reduction: Expensive operations are replaced with equivalent but less expensive operations;
- Exploiting floating-point unit: the floating-point unit available on the considered MCU has been enabled for improving the efficiency and speed on floating point operations.

```

ak = 0;
for (ck = 0; ck <= 28; ck += 3) {
    for (k = 0; k < 3; k++) {
        c[ck + k] = x[ak + k] - settings_xoffset[k];
    }
    ak += 3;
}

ak = 0;
for (ck = 0; ck <= 28; ck += 3) {
    for (k = 0; k < 3; k++) {
        b_c[ck + k] = c[ak + k] * settings_gain[k];
    }
    ak += 3;
}

ak = 0;
for (ck = 0; ck <= 28; ck += 3) {
    for (k = 0; k < 3; k++) {
        y[ck + k] = b_c[ak + k] + settings_ymin;
    }
    ak += 3;
}
    
```

Loop Jamming →

```

ak = 0;
for (ck = 0; ck <= 28; ck += 3) {
    for (k = 0; k < 3; k++) {
        c[ck + k] = x[ak + k] - settings_xoffset[k];
        b_c[ck + k] = c[ak + k] * settings_gain[k];
        y[ck + k] = b_c[ak + k] + settings_ymin;
    }
    ak += 3;
}
    
```

Figure V.15 Some of the main practices adopted for final code optimization (STEP 4 of the procedure): Loop Jamming.

```

for( int i = 0; i < 100; i++)
v[ i ] = x[ i ] * 4 ;
    
```

Loop Unrolling →

```

for( int i = 0; i < 100; i += 2 ) {
v[ i ] = x[ i ] * 4 ;
v[ i+1 ] = x[ i+1 ] * 4 ; }
    
```

Figure V.16 Some of the main practices adopted for final code optimization (STEP 4 of the procedure): Loop Unrolling.

```

c_c += c_a[ak] * ((2.0F / (1.0F +
(float)exp(-2.0F * ((fv0[ak] + fo) + a[ak]))))
- 1.0F);
}
*b_y1 = ((1.20426226F + c_c) - 1.0F) / 0.0489715971F + 39.25F;
    
```

FPU use →

```

c_c += c_a[ak] * ((2.0F / (1.0F
+ expf (-2.0F * ((fv0[ak] + fo) + a[ak])))
- 1.0F));
}
*b_y1 = ((Num_4 + c_c) + 1.0F) * Num_6 + Num_1;
    
```

Figure V.17 Some of the main practices adopted for final code optimization (STEP 4 of the procedure): Use of FPU.

To better show the application of such rules, Figure V.15 to Figure V.17 shows some examples of the practices above described and Table V.13 highlights the effects of steps 3 and 4 on the execution times and memory needed for the implementation of the IFD scheme. To measure the execution times, the transition of a digital line at the start and the stop of the main function was included. Then, 1000 repetitions of the developed firmware were run and each time the time interval between the start and stop of the main function through a digital scope LECROY 104Xs (10 ppm Time interval accuracy) was estimated.

Table V.13 Execution times measured with experimental tests (A frequency clock of 22 MHz is involved) and memory required for the implementation of the IFD scheme.

Execution times [μs]					
Initial code (STEP 2)		Compiler optimized code (STEP 3)		Final optimized code (STEP 4)	
Mean value	Standard deviation	Mean value	Standard deviation	Mean value	Standard deviation
1851.6	7.0	1628.0	6.0	686.6	0.3
502.3	2.1	448.4	1.8	181.3	0.1
314.4	1.3	282.7	0.9	111.8	0.1

Memory occupation [kB]					
Initial code (STEP 2)		Compiler optimized code (STEP 3)		Final optimized code (STEP 4)	
Flash RAM	SRAM	Flash RAM	SRAM	Flash RAM	SRAM
33	40	25	40	24	40

As for the execution times the following main considerations can be drawn:

- the use of the compiler optimizations (STEP 3) alone generally does not guarantee significant improvements in the code efficiency of the proposed IFD scheme. Indeed, it allows reducing the mean execution time of about 10 % (compare performance of Initial optimized code and Compiler optimized code);
- thanks to the adoption of the proposed guidelines and intensive use of MCU architecture (STEP 4), it has been possible to significantly reduce the mean execution time of about 60 % (compare performance of Compiler optimized
- thanks to the adopted code optimizations the time constraint for the real-time operating (1ms) is satisfied.

As for the memory, as expected SRAM occupancy is not affected by the code optimizations whereas the Flash RAM occupancy is reduced when the code optimizations are adopted.

The achieved values are in any case fully compatible with the memory resources available on the considered MCU (compare values of Table V.13 with MCU features).

STEP 5: Firmware validation

A firmware verification and validation are required for confirming that the implemented code really provides the expected outputs. This kind of analysis allows:

- verifying general correctness of the firmware implementation;
- evaluating the effects of data approximations introduced by finite arithmetic representation of the adopted MCU.

This phase was made by comparing the outputs of the Final optimized code (provided by STEP 4) with ones achieved by the Matlab m-file (used as reference).

Since the most complex operations involve the NARX, such a validation is made by comparing the corresponding outputs provided by MCU versus ones provided by Matlab. Figure V.18, reports the percentage deviation between the MCU outputs and Matlab ones for a data subset considered. The absolute maximum percentage error was always less than $2 \cdot 10^{-5}$ and similar results were achieved for other data subsets.

As a consequence, the firmware implementation is validated. Moreover, these values suggest that the errors due to the MCU data approximation can be considered negligible with respect to the typical measurement uncertainties involved (1 % 5 % of the reading), also considering a different data representation among MCU (32-bit) and Matlab (64-bit).

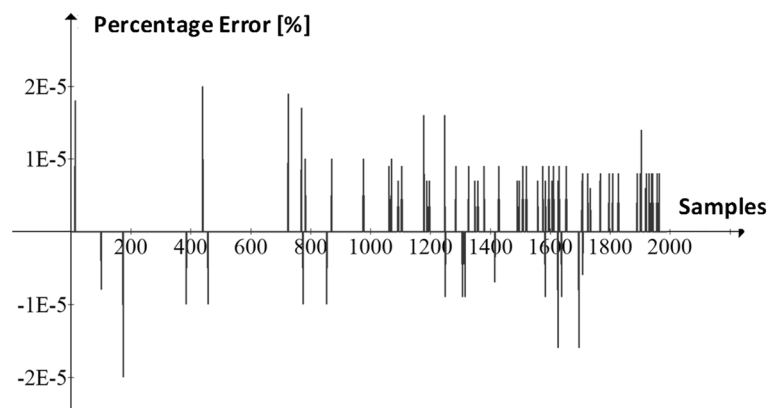


Figure V.18 Percentage error between the reference outputs (provided by Matlab code) and the firmware ones (provided by the Final optimized code).

V.6 Experimental results on the road

In this section the experimental results for testing the performance of the realized IFD system on the motorcycle are reported. In particular, a suitable experimental campaign has been carried out for analyzing the diagnostic and dynamic performance during real tests.

V.6.1 Faults introduced in the control loop

As far as the generation of faults, open circuit, short circuit, hold and losing calibration have been emulated starting by fault free signals.

In a more detail, the emulation of the faults was achieved thanks to capability of the CAN-BUS data logger to be programmed for putting on the CAN-BUS network suitably altered (in a controlled way) versions, of the sensor real outputs. In particular, for each of the considered faults to be emulated, the CAN-BUS data logger was programmed in order to introduce the desired fault after a suitable time instant. By this way, it will be possible to estimate the above defined figures of merit in a systematic way.

V.6.2 Diagnostic and dynamic performance

To evaluate the diagnostic performance of the realized IFD system, experimental campaigns were conducted for evaluating the previously described figures of merit. In particular, the motorcycle run along a mixed route about 5 km-length with the following scheme:

- 20 laps for the evaluation of FA % when the sensor is fault free;
- 20 laps (for each one of the considered faults) for the evaluation of CD % and TD when one of the above described faults is present.

As for FA %, it was less equal to 0 %. As for CD % it was approaching 100 % for each considered fault with except for the Losing calibration fault (see Table V.14).

Table V.14 *Diagnostic performance of the IFD scheme*

Type of Fault	Open circuit	Short circuit	Hold	Losing calibration
CD %	100 %	100 %	100 %	95 %

Regarding TD, Table V.15 reports the achieved experimental results in terms of mean value and standard deviation.

Table V.15 *Dynamic performance of the IFD scheme (μ is the mean value, σ is the standard deviation).*

Type of Fault	Open circuit		Short circuit		Hold		Losing calibration	
	μ [s]	σ [s]	μ [s]	σ [s]	μ [s]	σ [s]	μ [s]	σ [s]
TD	0.4	0.1	0.5	0.2	1.4	0.2	26.4	3.8

From the analyses of such results some considerations can be drawn:

- Open and short circuit faults are easily (CD % equal to 100 %) and quickly detected (TD mean value < 1 s) and also the observed value of σ confirms very stable and robustness features of the realized IFD scheme with respect to this kind of faults. Similar considerations can be made for hold faults which are detected in the 100 % of cases and with TD mean value < 2 s.
- Losing calibration faults, as expected, they require more time to be detected and in one case (i.e. one of the 20 lap) we had a missed detection.

Chapter VI

Conclusions

In this work, design and validation techniques of a soft sensors for the motorcycle vertical dynamics have been presented. Also, their application in an IFD scheme with a validation on a real motorcycle has been carried out. More in details, the attention has been devoted to the front and rear stroke suspension sensors since the quantities measured from them are very important for the control strategy of a semi-active or active suspension system. The most used stroke sensors are the linear potentiometers and represents a weak point of the control loop in a semi-active or active suspension system due to the high cost and premature aging. For this reason, two different soft sensors regarding a semi-active suspension system have been presented: a front stroke velocity soft sensor and a rear stroke soft sensor.

The first soft sensor presented face to the problem to measure the suspension velocity from the acceleration signals output from two low-cost accelerometers connected respectively to the sprung mass and the front wheel of a motorcycle using two different techniques.

The former technique based on the digital filtering of accelerations intends to improve the numerical integration through the adoption of the cascade of suitable low and high pass filters. Experimental results confirm the corresponding soft sensor is suitable for the needs of a semi-active or active suspensions control system in order to both reduce the negative effects of low frequencies and to follow with precision the widest oscillations of the suspension velocity signal.

The latter technique, based on the adoption of recurrent Artificial Neural Network, lead to develop a soft sensor able to correctly predict the suspension velocity in lower ranges.

The second soft sensor is mainly focused on the prediction of the rear stroke sensor using a NARX Neural Networks. The measurement campaign and post-processing analysis concerning with the suspension stroke sensor have highlighted the validity of the proposed solution in terms of both static and dynamical behavior (represented respectively by the mean error and the sliding mean deviation in the output prediction).

Experimental results lead to the adoption of the NARX model as a useful benchmark for the implementation of IFD strategies for motorcycle rear stroke sensors.

The deep analysis performed in terms of REC and SOE curves has proved that the prediction error can be limited to 5-6%, which allows to identify also very small losing calibration faults, thus enabling the capability of quickly triggering recovery or predictive maintenance procedures with the aims of improving the driver safety and holding high the real effectiveness of the suspension systems.

Thanks to the adoption of suitable programming rules and code optimization the whole IFD procedure has been developed on a commercial low-cost general purpose STM32F4 MCU belonging to the class of ARM-M4 architecture. Experimental results show that the implemented scheme is able to identify several types of faults that could occur on linear potentiometers which are the most used sensors employed in such context. In particular, short circuit, open circuit and hold faults are always detected ($CD = 100\%$) and very quickly (time needed for the detection is less than 2 s) whereas losing calibration fault requires larger time interval (about 30 s) to provide reliable diagnostic results. However, since the effects of this last fault is not so critic for passenger safety and motorcycle handling, the time required for their diagnosis is suitable for the purpose.

References

Acocella, G., Anchini, R., Paciello, V., Pietrosanto, A., & Sommella, P. (2010). A new approach to magnetorheological damping control. *Proc. of 2010 IEEE International Instrumentation and Measurement Technology Conference, IMTC Conference*, (p. 908-912).

Audino, D. (2007). A Perspective on In-Motorcycle Electronic Systems. IEEE.

Betta, G., & Pietrosanto, P. (2000). Instrument fault detection and isolation: state of the art and new research trends. *IEEE Transactions on Instrumentation and Measurement*, 100-107.

Betta, G., Capriglione, D., Pietrosanto, A., & Sommella, P. (2011). ANN-based sensor fault accommodation techniques. *IEEE International Symposium on Diagnostics for Electric Machines, Power Electronics & Drives (SDEMPED)*. Bologna.

Bi, J., & Berrett, K. (2003). Regression Error Characteristic Curves. *Proceedings of the Twentieth International Conference on Machine Learning (ICML-2003)*. Washington DC.

Calder, Grunwald, & Zorn. (1994). *Quantifying Behavioral Differences Between C and C++ Programs*. Colorado (USA): University of Colorado, Department of Computer Science.

Capriglione, D., Carratù, M., Liguori, C., Paciello, V., & Sommella, P. (2017). A soft stroke sensor for motorcycle rear suspension. *Measurement*, 46-52.

- Capriglione, D., Carratù, M., Pietrosanto, A., & Sommella, P. (2018). Real-Time implementation of an IFD scheme for motorcycle sensors. *Proc. of IEEE International Instrumentation and Measurement Technology Conference (I2MTC)*. Houston: IEEE.
- Capriglione, D., Carratù, M., Pietrosanto, P., & Sommella, P. (2017). ANN-based IFD in Motorcycle Rear Suspension. *Proc. of 15th IMEKO TC10 Workshop on Technical Diagnostics*. Budapest.
- Capriglione, D., Carratù, M., Sommella, P., & Pietrosanto, A. (2018). NARX ANN-based instrument fault detection in motorcycle. *Measurement: Journal of the International Measurement Confederation*, 304-311.
- Capriglione, D., Liguori, C., & Pietrosanto, A. (2007). Real-time implementation of IFDIA scheme in automotive system. *IEEE Transactions on Instrumentation and Measurement*, 824 – 830.
- Carratù, M., Pietrosanto, A., Sommella, P., & Paciello, V. (s.d.). Suspension Velocity Prediction From Acceleration Measurement for Two Wheels Vehicle. *I2MTC*. 2017.
- Catelani, M., Ciani, L., Reatti, A., Corti, F., Sorrentino, V., Ayachit, A., & Kazimierzuk, M. (2018). Reliability analysis and electrical characterization of a Class-E resonant inverter. *IEEE International Instrumentation and Measurement Technology Conference (I2MTC)*, (p. 1-6).
- Chaki, S., & Ghosal, S. (2019). Integrated soft computing based methodologies for modelling and optimisation.
- Cossalter, Doria, Garbin, & Lot. (2006). Frequency-domain method for evaluating the ride comfort of a motorcycle.
- Cossalter, V. (2002). Suspension overview in Motorcycle Dynamics. (p. 140-156). WI: Race Dynamics Inc.
- De Luca, P., & Doria, A. (2007). Numerical analysis of a scooter passing over a road bump.
- Delvecchio, D., Spelta, C., Perico, G., & Savaresi, S. (2010). Accelerometer-based estimation of the elongation speed in a motorcycle suspension via Kalman-filter techniques. *Proc. of 49th IEEE Conference on Decision and Control*, (p. 5566-5571).

Fortuna, L., Graziani, S., Rizzo, A., & Xibilia, M. (2007). *Soft Sensors for Monitoring and Control of Industrial Processes*. Springer Book.

Gavin, H., Morales, R., & Reilly, K. (1998). Drift-free integrators. *Review of Scientific Instruments*, 2171-2175.

Gobbi, M., & Mastinu, G. (2001). Analytical description and optimization of the dynamic behavior of passively suspended road vehicles. *Journal of Sound and Vibration*, 457-481.

Guglielmino, E., Sireteanu, T., Stammers, C., Ghita, G., & Giuclea, M. (2008). Semi-active Suspension Control.

Hrovat, D. (1997). Survey of advanced suspension developments and related optimal control applications. *Automatica*, (p. 1781–1817).

KEIL. (2018, August 15). Tratto da KEIL: <http://www.keil.com/appnotes/files/apnt202.pdf>

Liguori, C., Paciello, V., Paolillo, A., Pietrosanto, A., & Sommella, P. (2013). Characterization of motorcycle suspension systems: comfort and handling performance evaluation. *Proc. of 2013 I²MTC Conference*, (p. 444-449).

Liguori, C., Paciello, V., Paolillo, A., Pietrosanto, A., & Sommella, P. (2014). On road testing of control strategies for semi-Active suspensions. *Proc. of 2014 IEEE International Instrumentation and Measurement Technology Conference*. Montevideo.

Liguori, C., Paciello, V., Paolillo, A., Pietrosanto, A., & Sommella, P. (2015). ISO/IEC/IEEE 21451 smart sensor network for the evaluation of motorcycle suspension systems. *IEEE Sensors Journal*, 1549-2558.

Liguori, C., Paciello, V., Pietrosanto, A., & Sommella, P. (2014). A software sensor for motorcycle suspension stroke. *Proc. of 20th IMEKO TC4 Symposium on Measurements of Electrical Quantities*, (p. 1110-1115). Benevento.

Marek, J. (2011). Automotive MEMS sensors — Trends and applications. *VLSI Technology, Systems and Applications (VLSI-TSA)*.

- Norgia, M., Boniolo, I., Tanelli, M., Savaresi, S., & Svelto, C. (2009). Optical Sensors for Real-Time Measurement of Motorcycle Tilt Angle. *IEEE Transactions on Instrumentation and Measurement*, 1118-1129.
- Paciello, V., & Pietrosanto, A. (2011). Magnetorheological Dampers: A New Approach of Characterization. *IEEE Transaction on Instrumentation and Measurement*, 1718 – 1723.
- Paciello, V., & Sommella, P. (2013). Smart sensing and smart material for smart automotive damping. *IEEE Instrumentation and Measurement Magazine*, 24-30.
- Poussot-Vassal, C., Spelta, C., Sename, O., Savaresi, S., & Dugard, L. (2012). Survey and performance evaluation on some automotive semi-active suspension control methods: A comparative study on a single-corner model. *Annual Reviews in Control*.
- Ruhm, K. (2007). Sensor fusion and data fusion – Mapping and reconstruction. *Measurement*, 145-157.
- Samragh, M., Ghasemzadeh, M., & Koushanfar, F. (2017). Customizing Neural Networks for Efficient FPGA Implementation. *IEEE 25th Annual International Symposium on Field-Programmable Custom Computing Machines*, (p. 1-8).
- Spelta, C., Delvecchio, D., & Savaresi, S. (2010). A comfort oriented control strategy for semi-active suspensions based on half car model. *ASME Conference DSCC20102*, (p. 835-840).
- SPRINGOFF. (2012). *European Patent No. 2250038*.
- Srivastava, & Wall. (1999). A practical system for intermodule code optimization at link-time. *Journal of Programming Languages*.
- ST. (2018, August 15). Tratto da ST: <http://www.st.com/en.DM00037051.pdf>.
- Sun, J., & Yang, Q. (2009). Compare and Analysis of Passive and Active Suspensions under Random Road Excitation. *Proceedings of the IEEE International Conference on Automation and Logistics*.
- Ullman, S. (1986). *Compilers: Principles, Techniques, and Tools*. Addison-Wesley.

Yao, J., & Zheng, J. (2006). Semi-active suspension system design for quarter-car model using model reference sliding mode control. *IEEE International Conference on Vehicular Electronics and Safety*, (p. 392-402).

Zhang, J., Yin, Z., & Wang, R. (2017). Nonlinear Dynamic Classification of Momentary Mental Workload Using Physiological Features and NARX-Model-Based Least-Squares Support Vector Machines. *IEEE Transactions on Human-Machine Systems*, 536-549.

Appendix I

NARX network

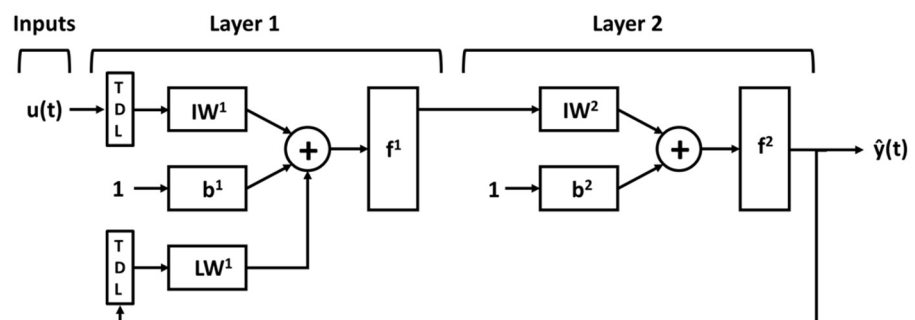
The Non-linear Auto Regressive network with eXogenous inputs (NARX) is a recurrent dynamic network, with feedback connections enclosed between different layers of the network. The NARX model is based on the ARX linear model, which is commonly used in time series modeling. In such situations, the construction of an ARX-type model should only be based on observing the system's behavior by using the input and output variables (external).

The equation that defines a NARX model is:

$$y(t) = f(y(t-1), \dots, y(t-n_y), u(t-1), \dots, u(t-n_u)) \quad (\text{Appendix.1})$$

Where the next value of the output signal function $y(t)$ is regressed to earlier values of the output signal and to earlier values of an independent (exogenous) input signal.

It is possible to implement the NARX model by using an open-loop neural network to approximate a specific function f . A resulting network diagram is shown in Figure Appendix.1, where a two-layer open-loop network is used.



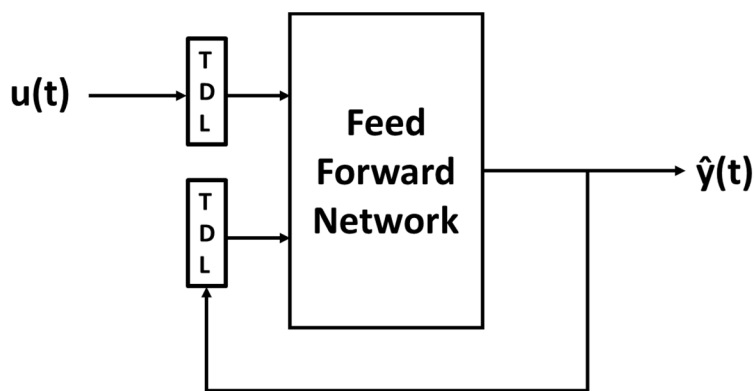
Appendix I.1 Structure of a NARX network

This implementation allows to reach an ARX model in which the input and output can be multidimensional. Non-linear self-regressive models with

nonlinear auto regressive exogenous inputs (NARX) are obtained by considering only regressors: $Y(t-K)$ and $U(t-K)$.

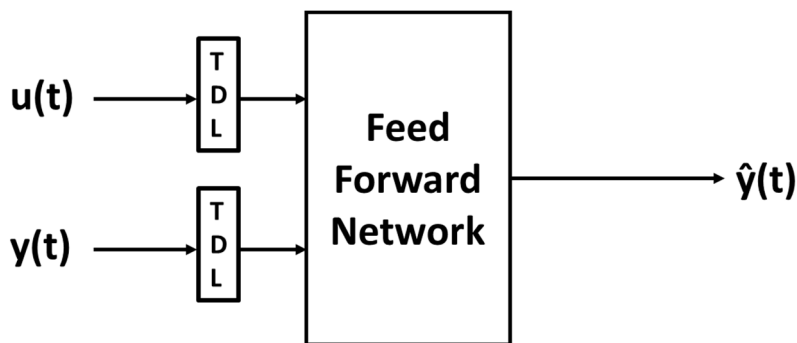
As can be seen from the Eq. Appendix.1 of the NARX model, the output is given by a nonlinear function that refers to the value of the output considered in the earlier moments and the value of the exogenous variable also observed in the past moments.

The output of a NARX network can be described as an estimation of the output variable of a nonlinear dynamic system that will be attempt to model. The latter is reintroduced as input for the open loop neural network as shown in the figure below.



Appendix I.2 NARX Parallel Architecture

Since the true output is available during the training of the network, it is possible to use a series-parallel architecture, in which the true output is used instead of the estimated output, as shown in the figure below on the right.



Appendix I.3 NARX Series-Parallel Architecture

This operation has two advantages: the first one is that the inputs of the open loop network is more accurate, the second one is that the resulting network

has a purely open-loop architecture which allows the use of the static backpropagation algorithm for the learning phase.